The feasibility of an effective data warehousing solution for a tertiary institution

By

Amer Bin Nazir

Submitted in fulfilment of the requirements of the degree

MASTER SCIENCE

In the Faculty of Natural and Agricultural Sciences
Department of Computer Science and Informatics
University of the Free State
Bloemfontein
South Africa

2008

Study leader

Prof Theo McDonald
Department of Computer
Science and Informatics

Acknowledgment

I would like to express my appreciation to my supervisor Prof Theo McDonald at the University of the Free State for his support and involvement in the research. Further, I would like to thank Mr. Willem Malherbe, Registrar of the University of the Free State, for helping me in developing a data warehouse for the University.

Declaration

I declare that the dissertation hereby submitted by me for the M. Sc. Computer Science degree at the University of the Orange Free State is my own independent work and has not previously been submitted by me at another university/faculty. I further more cede copyright of the dissertation in favour of the University of the Free State.

Table of contents

List of tables ____ *vii*

List of figures ____ *viii*

List of abbreviations ____ *ix*

Abstract ____ *1*

Abstrak ____ *3*

Chapter 1 ____ *5*

Introduction ____ *5*

1.1. Introduction ____ **5**

1.2. Problem definition ____ **7**

1.3. Research objectives ____ **8**

1.4. Hypothesis of the study ____ **8**

1.5. Research methodology ____ **9**

1.5.1. Action research ____ 9

1.6. Expected benefits of the study ____ **11**

1.7. Limitations of this study ____ **11**

1.8. Structure of dissertation ____ **12**

1.9. Summary ____ **13**

Chapter 2 ____ *14*

Data warehouse basics and concepts ____ *14*

2.1. Introduction ____ **14**

2.2. Business intelligence ____ **14**

2.2.1. BI tools ____ 14

2.2.2. DW is the basis for BI ____ 15

2.3. What is a DW? ____ **16**

2.4. OLTP systems ____ **16**

2.4.1. Entity relational model ____ 17

2.4.2. ER-limitations in data querying ____ 17

2.5. Difference between OLTP and DW systems ____ **18**

2.5.1. Slowly changing dimensions (SCD) ____ 19

2.6. Why has a separate data warehouse? ____ **19**

2.7. Data warehouse or data mart ____ **19**

2.7.1. Data marts ____ 20

2.7.2. Top-down versus bottom-up approach ____ 21

2.8. DW architecture components ____ **21**

2.9. Data acquisition ____ **21**

2.9.1. Extraction, transformation and loading (ETL) ____ 22

2.9.2. Data extraction ____ 23

2.9.3. Data transformation / data cleansing ____ 24

2.9.4. Data loading ____ 25

2.10. Data storage 25
2.10.1. Dimensional model 26
2.10.2. Fact table 26
2.10.3. Dimension table 27

2.11. Information delivery 27
2.11.1. Online analytical processing (OLAP) 27
2.11.2. Types of OLAP tools 27
2.11.3. Data mining 29

2.12. What to put in the DW? 29

2.13. Metadata 29

2.14. Summary 30

Chapter 3 *31*

Problems in the existing OLTP system *31*

3.1. Introduction 31

3.2. History of student record systems at the UFS 31

3.3. Legacy system 31

3.4. IBM system 31

3.5. PeopleSoft OLTP system 32

3.6. Problems in the current OLTP system 32
3.6.1. Size and complexity of the database 33
3.6.2. Lack of data standardisation/interaction 33
3.6.3. Redundant tables 33
3.6.4. No referential integrity 34
3.6.5. Unique key violation 34
3.6.6. Data capturing problems 36
3.6.7. Typographical errors 36
3.6.8. Missing academic programme and academic plans 37
3.6.9. Missing links 37
3.6.10. Inconsistencies in data 38
3.6.11. Spaces in mandatory columns 39
3.6.12. Dropped academic plans 39
3.6.13. Changed academic plans 40
3.6.14. Missing enrolment dates 40
3.6.15. Year and semester module conflicts 41
3.6.16. Product customization issues 41

3.7. Summary 41

Chapter 4 *42*

Higher education management information system (HEMIS) *42*

4.1. Introduction 42

4.2. National Database 42
4.2.1. Returns to be sent to the DoE via the VALPAC2 system 42
4.2.2. Submission dates for student and staff data 43
4.2.3. Funding groups 43
4.2.4. Classification of educational subject matter (CESM) 44
4.2.5. Study time 44
4.2.6. Course credit 44
4.2.7. Full-time equivalent (FTE) 46
4.2.8. Teaching input unit (TIU) 46
4.2.9. Funding formula 47

4.3. VALPAC2 file structure 48
4.3.1. VALPAC2 system and its limitations 49
4.3.2. VALPAC2 reporting 49

4.4. Third party solutions 50

4.5. Benefits in using HEMIS 50
4.5.1. HEMIS database software 51
4.5.2. Database structure 51
4.5.3. Lack of expertise and skills in TIs 51
4.5.4. HEMIS technical support 51
4.5.5. Efficient customization and modification process 52
4.5.6. Existing hardware and networks support 52
4.5.7. Data loads from VALPAC2 validated files 52
4.5.8. Better data validation process than VALPAC2 52
4.5.9. Auditing reports 53
4.5.10. Ad-hoc reporting 53
4.5.11. Microsoft Excel analytical capabilities 53

4.6. HEMIS limitations and pitfalls 53
4.6.1. Cannot provide the complete picture 53
4.6.2. Extraction, transformation and loading (ETL) 54
4.6.3. Replicated data loads 54
4.6.4. Non descriptive database fields 54
4.6.5. Complex joins 54
4.6.6. Table joins using textual attributes 55
4.6.7. Database views for disintegrated data 56
4.6.8. End-user layer for reporting 56
4.6.9. Delay reporting 56
4.6.10. Management requirement changes very rapidly 56
4.6.11. Access to HEMIS data 56
4.6.12. The HEMIS system provides information not knowledge 57
4.6.13. HEMIS and strategic reporting 57
4.6.14. Cost of HEMIS systems 57

4.7. Summary 57

Chapter 5 59

Star models of the student data mart 59

5.1. Introduction 59

5.2. Theoretical framework (TF) 59
5.2.1. Theoretical framework for HEMIS 59
5.2.2. Theoretical framework for student data mart 60

5.3. Student data mart 61
5.3.1. Naming conventions 62
5.3.2. Grain of the student data mart 63
5.3.3. Summarisation of the SDM 64
5.3.4. Conformed dimensions 64
5.3.5. Junk dimensions for filtering HEMIS data 65

5.4. Student dimension 66
5.4.1. Student address dimension 68

5.5. Enrolment model 68

5.6. Course registration model 71

5.7. Admission applications model 74

5.8. Admission applications snapshot model 76

5.9. Undergraduate longitudinal studies model 78

5.10. Output subsidy model 79
5.11. Summary 81
Chapter 6 *82*
Extraction transformation and loading issues in the student data mart *82*
6.1. Introduction 82
6.2. The ETL process 82
6.2.1. Incremental loads 82
6.2.2. Staging tables 83
6.2.3. Staging layer history tables 83
6.3. ETL for the student data mart 84
6.4. The student dimension 84
6.4.1. Address 84
6.4.2. Primary nationality and native language 85
6.4.3. Typographical errors 86
6.5. Programme dimension 86
6.5.1. Academic plans with more than one academic programme 87
6.5.2. Academic career and education level of academic plans 88
6.6. Student courses and end-term registrations 89
6.6.1. Dropped academic plans 90
6.6.2. Next year's enrolment record 90
6.6.3. Changed academic plans 91
6.6.4. Academic programme primary location indicator 92
6.6.5. Missing enrolled dates 93
6.6.6. Student FTE 93
6.6.7. Year census flag 94
6.6.8. Staging table enrolment 95
6.6.9. Historical data 95
6.6.10. Primary academic plan 96
6.6.11. Entrance category 97
6.6.12. Subsidy student status 98
6.6.13. Enrolment flags dimension 99
6.7. Loading of enrolment facts 100
6.7.1. Nationality lookup failure 100
6.7.2. Duplicate race or gender 100
6.7.3. Matric rating 100
6.8. Data auditing package 101
6.9. Summary 101
Chapter 7 *102*
Comparisons between HEMIS and SDM systems *102*
7.1. Introduction 102
7.2. Similarities between the two systems 102
7.2.1. Separate hardware and software 102
7.2.2. Ongoing maintenance 102
7.2.3. Historical data 103
7.2.4. Data correction at source side 103
7.2.5. Web interface and security 103
7.2.6. Costing 103
7.3. Differences 103
7.3.1. Data coverage 104

7.3.2. Data load frequency 104
7.3.3. Extraction, transformation and loading 104
7.3.4. Incremental loads 105
7.3.5. Data auditing 106
7.3.6. Database structure 106
7.3.7. Course credit value and subsidy example 107
7.3.8. Slowly changing dimensions (SCD) 109
7.3.9. Report writing skills 110
7.3.10. Front-end reporting tools 110

7.4. Comparison summary 112

7.5. Summary 112

Chapter 8 113

Survey results and researcher's experiences 113

8.1. Introduction 113

8.2. Email survey 113
8.2.1. Data warehouse efforts 113
8.2.2. Data warehouse data 114
8.2.3. Executive sponsorship 115
8.2.4. Data dictionary software 116
8.2.5. HEMIS system 116

8.3. Usability testing 117
8.3.1. Academic information student unit (AISU group) 117
8.3.2. BI user group (BI group) 118
8.3.3. Head of schools 118
8.3.4. Portfolio executives 118

8.4. Usability testing results 118

8.5. Researcher's experiences and research findings 124
8.5.1. Tertiary institutions' survival 124
8.5.2. The OLTP system is not suitable for strategic information 125
8.5.3. Third party software is not suitable for strategic information 125
8.5.4. DW simplifies database structures 125
8.5.5. Problems in the OLTP database can be solved in the staging area 126
8.5.6. Limitations of MOLAP storage and data retrieval 126
8.5.7. Usability testing results 127
8.5.8. The tendency for other TIs to adopt data warehousing 127

8.6. Recommendations 127
8.6.1. Start small 128
8.6.2. Use what is available 128
8.6.3. Make use of summarised granularity 128
8.6.4. Derived and calculated fields and their effectiveness 129
8.6.5. Make use of junk dimensions 129
8.6.6. Get management involved 130
8.6.7. Major subjects or topics as tables in the warehouse 130
8.6.8. Get end-user support 131

8.7. Summary 131

Chapter 9 133

Conclusion 133

9.1. Introduction 133

9.2. Motivations for this study 133

9.3. Factors enforcing TIs for seeking help 133

9.4. Why the UFS was chosen for this study? 135

9.5. Research design 135

9.6. Feasibility of the DW solution 135

- Low budgets 136
- DW dimensional modelling 136
- Data problems at OLTP data can be fixed from the ETL 137
- ROLAP model in querying and analysis 137

9.7. Proposal for future studies 137

9.8. Summary 138

References 139

Appendixes 145

Appendix A Student file 145
Appendix B Course registration file 147
Appendix C Credit file 147
Appendix D Qualification file 148
Appendix E Qualification CESM file 148
Appendix F Course file 148
Appendix G Staging_Table_Courses 150
Appendix H Staging_Table_Enrolment 151
Appendix I Data warehouse & metadata Email 153
Appendix J Questionnaire output 154
Appendix K Usability testing questionnaire 155
Appendix L Usability testing results 159
Appendix M Program student address 166
Appendix N Program student nationality 170
Appendix O Program native language 173
Appendix P Program primary location 175
Appendix Q Program plan priority level 178
Appendix R Program HEMIS subsidy status 180
Appendix S Program entrance category 186
Appendix T Published papers 198

List of tables

Table 2-1: Difference between OLTP and DW systems **18**
Table 2-2: Data warehouse versus data mart **20**
Table 2-3: Duplicate customer records **25**
Table 2-4: Difference between MOLAP and ROLAP models **28**
Table 3-1: Column with different names and width **33**
Table 3-2: Subject table **34**
Table 3-3: Academic organisation table **34**
Table 3-4: Department table **34**
Table 3-5: ACAD_PLAN_TBL with no primary key **35**
Table 3-6: Modules having more than one course identification number **35**
Table 3-7: Inconsistent city names stored in PERSONAL_DATA table **36**
Table 3-8: Past academic programme and plan combinations **37**
Table 3-9: Graduates having no direct link with PS_STDNT_ENRL_TBL table **38**
Table 3-10: Academic career in STDNT_ENRL table **39**
Table 3-11: Academic career in CLASS_TBL table where classes are scheduled **39**
Table 3-12: STDNT_ENRL table **39**
Table 3-13: Unknown dropped plans **40**
Table 3-14: Student plan replacement with new one without keeping history **40**
Table 4-1: Funding groups: 2006/07 to 2008/09 ..... **44**
Table 4-2: Study time in universities **45**
Table 4-3: Input subsidies **47**
Table 4-4: Output subsidy excluded honours degree **48**
Table 4-5: Output subsidy higher degrees **48**
Table 4-6: Student collections files **48**
Table 4-7: Staff profile files **48**
Table 5-1: SDM business areas **62**
Table 5-2: Snapshot dimension **76**
Table 5-3: Pivot table for admission application snapshot facts **78**
Table 5-4: Pivot table for UG longitudinal students facts **79**
Table 5-5: Pivot table for output subsidy **80**
Table 6-1: STAGING_BATCH_ID table **83**
Table 6-2: Student Addresses table **85**
Table 6-3: OLTP nationality table **85**
Table 6-4: Student native language **86**
Table 6-5: PS_ACAD_PLAN_TBL **88**
Table 6-6: Mismatch of academic careers with degree level **88**
Table 6-7: DoE definitions of education levels **89**
Table 6-8: Program dimension **89**
Table 6-9: Student changed plan **91**
Table 6-10: Student course location **92**
Table 6-11: Campus location and priority **93**
Table 6-12: Enrolment status in staging area courses **95**
Table 6-13: Staging table enrolment **95**
Table 6-14: Academic career priority **96**
Table 6-15: Primary plan **97**
Table 6-16: Entrance category **97**
Table 7-1: Query response time **109**
Table 7-2: SCD type change 2 in program dimension **110**
Table 7-3: Comparison between HEMIS and SDM **111**

List of figures

Figure 1-1: The research interest in action research **10**
Figure 2-1: BI Tools **15**
Figure 2-2: ER diagram of OLTP system **17**
Figure 2-3: Data warehouse architecture **22**
Figure 2-4: Student enrolment star **26**
Figure 2-5: Metadata essential for end-users and IT **29**
Figure 4-1: HEMIS submission dates **43**
Figure 4-2: Calculation of the course credit for teaching **45**
Figure 4-3: Calculation of the course credit for research offerings **46**
Figure 4-4: VALPAC2 ER diagram **49**
Figure 4-5: HEMIS table's complex join paths **55**
Figure 5-1: Theoretical framework for third party HEMIS system **60**
Figure 5-2: Theoretical framework for SDM **61**
Figure 5-3: Factless fact table **64**
Figure 5-4: Student enrolment flags dimension **66**
Figure 5-5: Student dimension **67**
Figure 5-6: Enrolment star **69**
Figure 5-7: Enrolment fact table **70**
Figure 5-8: Pivot table from student enrolment star **71**
Figure 5-9: Course registration star **72**
Figure 5-10: Student course fact table **73**
Figure 5-11: Pivot table from student course registration star **74**
Figure 5-12: Admission applications star **75**
Figure 5-13: Application fact table **75**
Figure 5-14: Pivot table from admission application star **76**
Figure 5-15: Admission applications snapshot star **77**
Figure 5-16: UG longitudinal studies star **78**
Figure 5-17: Output subsidy star **80**
Figure 6-1: ETL for student dimension **84**
Figure 6-2: ETL for the program dimension **87**
Figure 6-3: Enrolment ETL package 1 ..... **90**
Figure 6-4: Enrolment ETL package 2 ..... **92**
Figure 6-5: Enrolment ETL package 3 ..... **94**
Figure 6-6: Enrolment ETL package 4 ..... **96**
Figure 6-7: Subsidy student definition **98**
Figure 6-8: Enrolment flags dimensions **99**
Figure 6-9: Loading of Enrolment Flags dimension **99**
Figure 7-1: OLTP: database structure for course credit values **107**
Figure 7-2: SQL for generating course credit value **108**
Figure 8-1: Data warehouse effort **114**
Figure 8-2: Data warehouse data **115**
Figure 8-3: Executive sponsorship **115**
Figure 8-4: Data dictionary software **116**
Figure 8-5: HEMIS system **117**
Figure 8-6: Report writing capabilities **119**
Figure 8-7: Underlying database structure **120**
Figure 8-8: Report writing assistance **121**
Figure 8-9: Reporting requirements **122**
Figure 8-10: DW reporting **123**
Figure 8-11: DW reporting environment **124**
Figure 8-12: MOLAP Cube maintenance **127**

List of abbreviations

AR	Action research
ASCII	American Standard Code for Information Interchange
BI	Business intelligence
CESM	Classification of educational subject matter
DBMS	Database management system
DoE	Department of education
DTS	Data transformation services
DW	Data warehouse, Data warehouses, Data warehousing
EIS	Enterprise information system
ER	Entity relationship modelling
ERP	Enterprise resource planning
ETL	Extraction transformation and loading
EUL	End user layer
FTE	Full time equivalent
HEDA	Higher education data analyzer
HEMIS	Higher education management information system
IT	Information technology
MDDB	Multiple-dimension database
MIS	Management information system
MOLAP	Multidimensional online analytical processing
OLAP	Online analytical processing
OLTP	Online transaction processing
RDBMS	Relational database management systems
ROLAP	Relational online analytical processing
SA	South Africa, South African
SCD	Slowly changing dimensions
SDM	Student data mart
TF	Theoretical framework
TI	Tertiary institution
TIs	Tertiary institutions
TIU	Teaching input unit
UFS	University of the Free State

Abstract

Even though industry in South Africa has utilized data warehousing technologies successfully for a number of years, tertiary institutions have lagged behind. This can in part be attributed to the high costs involved, many failures in the past and the fact that the decision makers of these institutions are unaware of what data warehousing is and the advantages it can bring. Several factors, however, are forcing tertiary institutions in the direction of data warehousing. They need all the help they can get to make this process as easy as possible.

Most of the tertiary institutions that still survive today came through periods of tough rationalizations and mergers. In order to stay alive and competitive, they have grown through the years and have developed into large businesses in and of themselves. On the one hand they had to make ends meet with subsidies from government that became less and less and on the other hand they had to provide more and more detailed statistics to the government. This change has resulted in a more business-like management of these institutions. Strategic decision making has now become of the utmost importance to tertiary institutions to meet the frequent changes in the government funding structure.

The University of the Free State initially tried to accomplish that with an online transaction processing system developed in-house. These systems, however, are designed to optimize transactional processing and the features which increase the efficiency of these systems are generally those which also make it difficult to extract information. When that did not work, a new online transaction processing system was bought from an international company at a huge cost. During the course of data transfer from the old to the new system (with a different database design) numerous data conversion errors generated anomalies and a lack of integrity in the database. The new system also proved inadequate to provide the necessary statistics required by the Department of Education. A system was subsequently purchased that utilized ASCII files prepared by the online transaction processing system which generated fixed reports according to the Department of Education requirements. This system provided a workable solution, but with changes in requirements, new reports need to be

developed continuously. It was also worthless for institutional planning and forecasting.

This study reported the advantages and disadvantages of the current systems in use to provide statistics to the Department of Education. It then proposes a new system based on data warehousing principles. The dimensional star schema design for a data warehouse is provided. The methods used to transfer, load and extract data are discussed in detail. The data warehouse solution is then compared to the current solutions. The conclusion is that a data warehouse is a feasible solution for the strategic information problems tertiary institutions are facing today. An effective management information system using data warehousing can be developed in-house with low budgets, institutional data can be fitted into dimensional modelling star schemas, and error free data can be provided to end-users by developing proper extraction, transformation and loading packages. The data surfaced to end-users from relational online analytical processing can provide statistics to government and can be used for general planning and forecasting purposes.

Keywords: Tertiary institution, data warehousing, student data mart, star schema, dimensional modelling, extraction, transformation, and loading, action research, comparisons, forecasting and planning

Abstrak

Alhoewel die industrie in Suid-Afrika datapakhuistegnologie vir ‘n aantal jare reeds suksesvol aangewend het, het tersiêre inrigtings agtergebly. Dit kan deels toegeskryf word aan hoë kostes, die vele mislukkings in die verlede en die feit dat die besluitnemers in hierdie inrigtings onbewus is van wat datapakhuise behels en die voordele wat dit inhou. Tans dwing verskeie faktore tersiêre inrigtings egter in die rigting van datapakhuise. Hulle benodig al die hulp wat hulle kan kry om hierdie proses so maklik as moontlik te maak.

Die meeste van die tersiêre inrigtings wat vandag nog oorleef, het deur tye van moeilike rasionalisersings en saamvoegings gekom. Om te oorleef en kompeterend te bly, moes hulle oor die jare groei en ontwikkel in groot besighede. Aan die eenkant moes hulle gate toestop met subsidies wat minder en minder word en aan die anderkant moes hulle meer en meer statistieke aan die regering verskaf. Hierdie verandering het meer van ‘n besigheidsbenadering tot die bestuur van die inrigting tot gevolg gehad. Strategiese besluitneming het nou van die allergrootste belang geword om die gereelde veranderinge in die regering se befondsingstruktuur die hoof te bied.

Die Universiteit van die Vrystaat het probeer om hierdie uitdaging oorspronklik aan te pak met ‘n transaksieverwerkingstelsel wat intern ontwikkel is. Hierdie stelsels is egter ontwikkel om transaksieverwerking te optimaliseer en die eienskappe wat die doeltreffendheid van hierdie stelsels verhoog, is gewoonlik ook verantwoordelik om die onttrekking van inligting te bemoeilik. Toe hierdie stelsel misluk, is ‘n nuwe stelsel teen hoë koste vanaf ‘n internasionale maatskappy aangekoop. Gedurende die oordragproses van die data vanaf die ou na die nuwe stelsel (met ‘n verskillende databasisontwerp) het verskeie data-omskakelingsfoute anomalieë en ‘n gebrek aan integriteit in die databasis tot gevolg gehad. Die he took geblyk dat die nuwe stelsel onvoldoende was om die nodige statistieke aan die Departement van Onderwys te verskaf. ‘n Stelsel is gevolglik aangekoop wat ASCII-lêers gebruik wat deur die transaksieverwerkingstelsel gegenereer is en wat vaste verslae lewer volgens die vereistes van die Departement van Onderwys. Hierdie stelsel was ‘n werkbare oplossing, maar met veranderinge in vereistes moes nuwe verslae voortdurend ontwikkel word. Dit was ook waardeloos vir beplannings- en voorspellingdoeleindes van die inrigting.

Hierdie studie doen verslag oor die voor- en nadele van die huidige stelsels om statistieke aan die Departement van Onderwys te verskaf. Dit stel dan ‘n nuwe stelsel voor wat gebaseer is op datapakuisbeginsels. Die dimensionele sterskema vir ‘n datapakhuis word gevolglik verskaf. Die metodes wat gebruik word om die data oor te dra, te laai en te onttrek word breedvoerig bespreek. Die datapakhuisoplossing word dan vergelyk met die huidige oplossings. Die gevolgtrekking is dat ‘n datapakhuis ‘n geskikte oplossing is vir die strategiese inligtingsprobleme wat tersiêre inrigtings vandag in die gesig staar. ‘n Doeltreffende bestuursinligtingstelsel wat datapakhuise gebruik kan intern met ‘n lae begroting ontwikkel word, inrigtingdata kan getransformeer word na dimensionele sterskemas en foutvrye data kan verskaf word aan eindgebruikers deur die gebruik van geskikte Onttrek-, Transformeer- en Laai-pakkette. Die data verskaf aan eindgebruikers vanaf ROLAP kan die statistieke aan die regering verskaf en dit kan gebruik word vir algemene beplanning en voorspelling vir die inrigting.

Sleutelwoorde: Tersiêre inrigtings, datapakhuise, sterskema, dimensionele modellering, onttrekking, transformasie en laai, aksienavorsing, vergelykings, voorspelling en beplanning.

Chapter 1
Introduction

1.1. Introduction

Business Intelligence (BI) has nowadays become an important part of the enterprises around the world. According to Lokken (2001) "business intelligence technologies attempt to help people understand data more quickly so that they can make better and faster decisions and, ultimately, better move toward business objectives". The key drivers behind BI objectives are to increase organisational efficiency and effectiveness (Eckerson, 2003a). Information is required to identify where the organisation has been, where it is now, and where it needs or wants to be in the future (Wierschem, McMillen, and McBroom, 2003). BI depends on BI applications having access to properly prepared data. A transactional database is not well suited for BI. For effective BI applications, database programmers need to create data warehouses (DW) or data marts, which are properly formatted amalgamations of all the key enterprise data. BI applications generate analytics or reports by querying and interpreting the contents of those data marts or DW (Zeichick, 2005). The DW essentially holds the BI for the enterprise to enable strategic decision making (Ponniah, 2001, p.12).

Wierschem et al. (2003) indicates that a DW requires millions of dollars to develop, plus significant hardware and personnel investment. Hammond (1998) cited on Gray & Israel (1999) quotes a Meta Group survey that the average cost for an enterprise warehouse is $3 million out of which $1 million is for professional services. This huge cost can be a major obstacle in developing countries. Wagner, Cheung, Lee, and Ipiscw (2003) stated that the budgets of developing countries are not even sufficient to pay for the knowledge management enabling information technology (IT) architecture. He indicated that less developed countries like the Philippines and Pakistan spend a smaller percentage of their budgets on IT as compared to developed countries like the United States and United Kingdom. For example the Philippines spends only 0.8 percent of its budget on IT as compared to the United States that spends 13 percent on IT. From the above statistics there is a great challenge to find

ways for applying DW technology in developing countries with their limited budgets for IT.

Industry in developed countries is experiencing a dramatic increase in the use of DW techniques and businesses have been using DW since the 1970s (Wierschem et al., 2003). Major users of DW include credit card companies, retailers, financial services, banks, airlines, manufacturing companies, telephone companies and insurance companies. Markedly absent from the above list of high profile users are academic institutions. Academic institutions, however, only recently have begun to identify and explore the possibilities and benefits that DW offers (Wierschem, et al., 2003).

Tertiary institutions (TIs) have grown through the years and educational institutions have developed into large businesses in and of themselves (Desruisseaux, 2000). This change has resulted in a more business-like management of these institutions as well (Lazerson, Wagener & Moneta, 2000). Koch and Fisher (1998) indicate that the truly significant problems facing TIs today relate to the nature of the curriculum, uses of faculty time, how to restrain cost increases, distance learning and the use of technology, cooperative relationships with business, and governance and leadership arrangements.

Before the advent of democracy in 1994, the South African (SA) government's tertiary education funding policies mirrored apartheid's divisions and the different governance models which it imposed on the tertiary system (Bunting, 2002). For the new government that came into power in 1994, the focus was to address the imbalances of the past, especially health, housing and primary education. The result was that the subsidies allocated to universities (their primary source of income) have drastically been cut. Most universities still surviving today had to go through a period of tough rationalizations and mergers. Even though the universities in SA now run on limited budgets, they have grown through the years and currently they need all the help they can get to properly manage their businesses.

SA tertiary institutions are financed principally by government subsidy and fee recovery from 1995 (Subotzky, p545-562, 2003). The Department of Education (DoE) needs unit record statistics of students and staff quarterly or yearly from all TIs for planning purposes and for allocating subsidies to them. The responsibility for

ensuring the accuracy and completeness of the data in the returns submitted to the Department rests with the institutions and they must be confident about the reasonableness and accuracy of the data prior to sending it to the Department.

1.2. Problem definition

As mentioned above, the DoE needs unit record statistics of students and staff quarterly or yearly from all TIs for planning and subsidy purposes. Top management of every tertiary institution (TI) also requires strategic information of their students and staff for utilizing the subsidies and developing future plans. Most of the institutions are using online transaction processing (OLTP) systems for record keeping and querying their institutional data. These systems are not really suitable for BI purposes, because data is fragmented in different OLTP systems e.g., Student Record Systems, Human Resources Systems, Library Systems, Inventory Systems, etc. The situation gets even worse when a TI replaces the old legacy system with a new OLTP system for record keeping. During data transfer from the old to the new system (with a different database design) numerous data conversion errors generate anomalies and a lack of integrity in the database. This really creates problems and difficulties for the management information system (MIS) department in the development and generation of strategic reports.

The MIS staff of these institutions remains busy throughout the year in generating statistics from their OLTP systems that are sitting with disintegrated and dirty data. In most of the cases it is impossible for the MIS staff to clean data properly which eventually results in inaccurate and incomplete submissions. To overcome these data issues more and more institutions are seeking help by purchasing enterprise resource planning (ERP) and third party higher education management information systems (HEMIS). The new implementation, especially in the case of an OLTP system, comes with thousands of tables with inconsistencies in columns and table names with no proper referential integrity and making data extraction more challenging. On the other hand HEMIS system come with their own pitfalls and deficiencies.

To date, little work has been done on fitting student and staff data to the dimensional model star schema. Allan (2000) indicated that the smaller client base for student

record systems has meant that less effort has gone into the development and standardisation of student record systems than is the case for accounting systems.

The problem, therefore, is, given a low budget and scarce BI resources and, given an OLTP system with data integrity problems, can a cost-effective DW solution be achieved that will solve the dirty data problem, that will replace the third party solutions and will provide both the required statistics for the DoE and strategic information for the institution internal needs.

1.3. Research objectives

The primary objective of this study is:

- To come up with empirical evidences that with given budget and resource constraints, that a DW is a better solution than OLTP or HEMIS systems, in order to provide strategic information for the DoE and institutional internal needs.

The secondary objectives of the study are:

- To come up with a proposed set of star model diagrams that can fit institutional data for both DoE and internal reporting.
- To point out the efforts required in the extraction, transformation, and loading (ETL) of data in the DW from an OLTP system that has anomalies and a lack of data integrity.
- To investigate which DW model is suitable for querying and reporting according to the size of TI data

1.4. Hypothesis of the study

The following specific research hypothesis is proposed:

An effective data warehousing solution can successfully be implemented to provide management information for a tertiary institution despite difficult challenges involved in its development.

1.5. Research methodology

The research was conducted at the University of the Free State (UFS) by developing a DW for planning and forecasting future needs. The DW output was more focused on MIS. The research methodology chosen for this research was action research (AR) and in the following sections motivation and benefits of choosing AR are highlighted.

1.5.1. Action research

AR can be described as a family of research methodologies which pursue action (or change) and research (or understanding) at the same time (Dick, p.1, 1999). It is characterised by the cyclic revision of action followed by reflection, often culminating in the refinement of the understanding using methods such as modelling. The iterative nature of the methodology promotes convergence to a greater understanding (Dick, 1999). Baskerville & Wood-Harper (1996a) applied AR in the methodology of systems development in information systems.

Marshall & McKay (n.d) reiterated some of the features of AR that make it particularly apposite for application to many facets of research in information systems. He further stated that the usual representation of the action research process is as a single cycle as shown in Figure 1-1 (with possible iterations), no matter which depiction of AR is used. This cycle can be passed through once in an AR study (referred to by Baskerville and Wood-Harper, 1998 as linear AR), or it can be repeated in the same context until satisfactory outcomes have been achieved, or a similar process can be applied in a number of different sites (called multiple iterations of AR (Koch, McQueen, & Scott, 1998).

Figure 1-1 is a representation of the researcher's problem solving interest. The researcher has a particular idea, or objective, or research question of interest which he/she wishes to pursue. Having identified some initial area of interest, the researcher will engage in the relevant literature, clarify issues and identify existing theoretical frameworks of relevance.

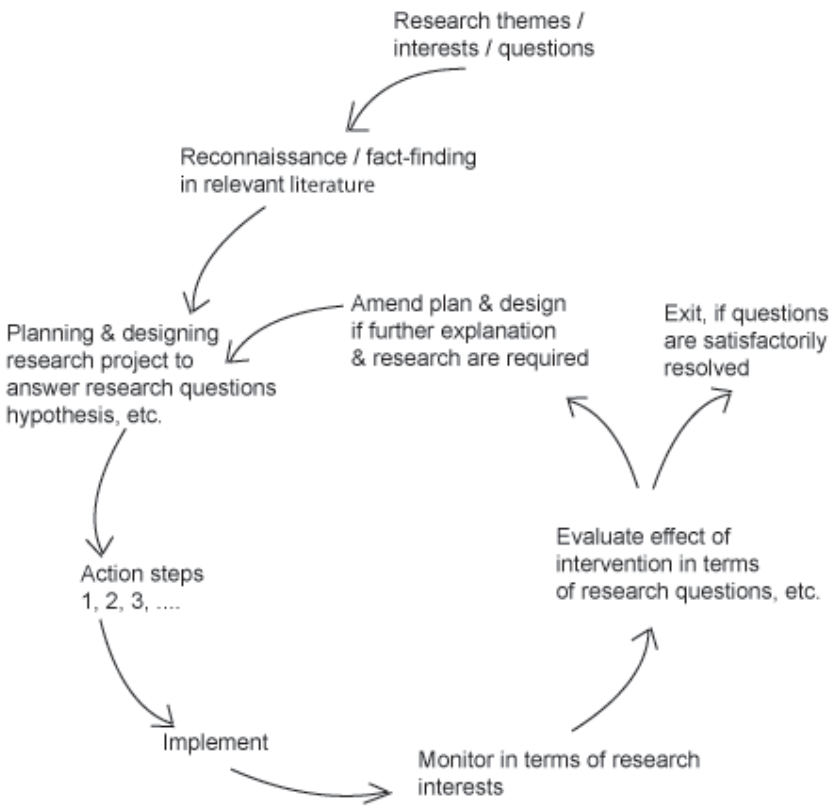

Figure 1-1: The research interest in action research

A theoretical framework from which to investigate the research interest will be adopted. From there, the researcher plans and designs a research project with the express purpose of enabling him/her to find answers to research questions, themes, or objectives, and so on. Action is taken, the researcher remaining cognisant of his/her particular theoretical perspective. These actions are monitored in terms of research interests, and evaluated for the effect the intervention has had in terms of the research questions. If the research questions can be answered or satisfactorily resolved, or in some way illuminated or even reframed, the researcher exists from the organisational settings. Otherwise, the researcher will amend his/her plans and designs to seek further explanations.

According to Marshall et al. (n.d) AR offers many positive features thus rendering it a powerful tool for researchers who are interested in finding out about the interplay between humans, technology, information, and socio-cultural contexts. One distinguishing feature of AR is, therefore, the active and deliberate self-involvement

of the researcher in the context of his/her investigation. Due to the fact that the author of the thesis is actively involved in the development of the DW, the action research methodology fits perfectly.

1.6. Expected benefits of the study

Tertiary institutions (TIs) in SA are just now entering the DW arena and this technology is therefore still evolving and new in this cultural diverse country. This study will be helpful for TIs which are planning for DW implementation in the following ways:

- It will be helpful to TIs in demonstrating that DW will definitely off-load the efforts from management that is currently spending a lot of time in the collection of data for students and staff submissions to the DoE, as well as reports required by top management for current and future statistics.

- It will be helpful for decision makers to realize that OLTP systems are not suitable for data analysis.

- It will provide eye opener evidence on the quality of data that has numerous errors and inconsistencies, as a result of disintegration between different OLTP and file systems throughout the institution.

- It will provide guidelines in fitting student and staff data into star diagrams that is the core database design for DW.

- The guidelines, methods, and scripts provided in the dissertation will be useful in setting up proper ETL processes for DW.

- It will be helpful in generating certain statistics like teaching input units, staff and student ratios, etc. from the DW that was never simple or possible from traditional OLTP systems and reporting tools.

1.7. Limitations of this study

It was stated previously that this study will provide benefits, and new knowledge to other SA tertiary institutions, in understanding the value of clean and integrated data

using dimensional modelling. In spite of all these benefits, the study has some limitations due to the following reasons:

- The study is limited to the student data mart (SDM) only. It is not possible to build the complete DW due to project time span constraints, hardware and development cost.

- It is not possible to conduct the feasibility study in other SA tertiary institutions which are sitting with the same data issues and problems.

1.8. Structure of dissertation

The dissertation will proceed according to the following outline.

Chapter 2 will provide DW basics that include definitions of DW, differences in OLTP and DW, DW architecture components and dimensional modelling in order to give the reader insight in DW technology.

Chapter 3 will provide problems in the existing OLTP systems by highlighting numerous data errors by quoting examples.

Chapter 4 will cover HEMIS systems that a TI is using for providing staff and student unit record statistics to DoE and institutional internal usage. Limitations of DoE VALPAC2 and third party HEMIS systems are highlighted in this chapter.

Chapter 5 will cover dimensional modelling with star diagrams of the student data mart for fitting institutional data.

Chapter 6 will cover the ETL process that is the heart of DW. This chapter provides examples of numerous efforts that were invested in order to provide quality data by using data cleansing and custom built programs.

Chapter 7 will provide comparisons between HEMIS and DW systems by giving similarities and differences between the two systems.

Chapter 8 will provide research findings by narrating researcher experiences and survey results conducted during the research.

Chapter 9 will cover conclusions by sharing knowledge to the reader about researcher experiences and recommendations in setting up a BI infrastructure for a tertiary institution. Recommendations for future studies that can be of interest to other researchers are also presented in this chapter.

1.9. Summary

This chapter first of all provided a background for the research. It was also the first activity, “research question of interest”, in the AR cycle. Several major challenges like cost, corrupted OLTP systems, scarce DW resources and statistics both for government and internal planning were highlighted. The main purpose of the research was stated as to determine if an effective DW solution can be developed irrespective of the challenges. The research methodology followed in this dissertation was also explained. The next chapter is devoted to DW basics that will provide enough knowledge to the reader about this technology in order to understand the rest of the dissertation.

Chapter 2

Data warehouse basics and concepts

2.1. Introduction

In the previous chapter the background was provided for the research. This chapter is purely focused on DW basics and concepts and is the second activity, "Fact-finding in relevant literature", in the AR cycle. The chapter starts by defining and explaining where BI fits in terms of OLTP and DW systems and that is followed by architectural components of DW. The chapter also provides detail on metadata definition and concepts which is an essential part of DW.

2.2. Business intelligence

"…Business intelligence is the process of getting enough of the right information in a timely manner and usable form and analyzing it so that it can have a positive impact on business strategy, tactics or operations" (Wally, 2003).

2.2.1. BI tools

To setup proper BI infrastructure a series of BI tools are required. Lokken (2001) stated that "...BI tools are back-end, infrastructure tools that deal with extracting data, cleaning it up, transforming it, re-organizing it, and optimizing it for use in decision making. These back-end tools include data warehouses, data marts, online analytical processing (OLAP) servers, and ETL tools. Other BI tools are designed to extract knowledge and insight from the data once it has been prepared. These tools include reporting, query, on-line analysis and exploration, visualization, decision modelling and planning, and data mining tools. Portals, dashboards, and scorecards are also pieces of the puzzle that help further organize information for easy consumption", as shown in Figure 2-1.

Figure 2-1: BI Tools

2.2.2. DW is the basis for BI

Successful knowledge management needs to integrate databases, information systems, and knowledge base systems. A DW can connect these three kinds of systems. DW provides a wide basis of integrated data and this data can be presented via MIS or enterprise information system (EIS). It could be interpreted as knowledge if analysis algorithms discover currently unknown patterns in the large amounts of DW data. According to Erdmann (1997) "newly derived knowledge or visualized information may be incorporated into the management's decision making process".

A DW is the basis of BI and a DW itself does not create value, value comes from the use of the data in the warehouse (List, Bruckner, Machaczek, and Schiefer, 2002). The greatest potential benefits of the DW occur when it is used to redesign business processes and to support strategic business objectives (Watson and Haley, 1998). According to Donhardt and Keel (2001) the DW empowers institutional decision makers by placing inquiry and analysis tools at their fingertips by providing the following benefits:

- Users can produce customized reports anytime, anywhere.
- Easy access and quick information delivery support administrative decisions at all levels and improve the way the university does business.

- Giving users the ability to generate their own reports greatly reduces the effort once spent by the MIS department developing ad-hoc programs and answering questions.

- Clients can passively view static reports or interact with dynamic analyses that help them develop their own customized reports.

2.3. What is a DW?

A data warehouse is a copy of transaction data specifically structured for querying and analysis (Kimball, 1996). Ponniah (2001, p.13) indicates that the data warehouse is an informational environment that:

- Provides an integrated and total view of the enterprise.

- Makes the enterprise's current and historical information easily available for decision making.

- Makes decision-support transactions possible without hindering operational systems.

- Renders the organisation's information consistent.

- Presents a flexible and interactive source of strategic information.

The five key defining features of DW are subject-oriented data, integrated data, time-variant data, non-volatile data, and data granularity (Ponniah, 2001, p.20). Before further exploring DW systems, OLTP systems and why they are different from DW systems will be explored.

2.4. OLTP systems

OLTP systems are the systems that are used to run the day-to-day core business of the company (Ponniah, 2001, p.10). An OLTP system supports the basic business processes of the company. These systems typically get the data into the database.

2.4.1. Entity relational model

OLTP systems are based on entity-relational (ER) modelling, starting with a conceptual ER design, translating the ER schema into a relational schema, and then normalizing the relational schema. ER-modelling works by dividing the data into many discrete entities as shown in Figure 2-2, each of which becomes a table in the OLTP database (Kimball 1996, p.8). ER modelling seeks to drive all the redundancy out of the data. If there is no redundancy in the data, then a transaction that changes, only needs to touch the database in one place (Kirpekar, 2005).

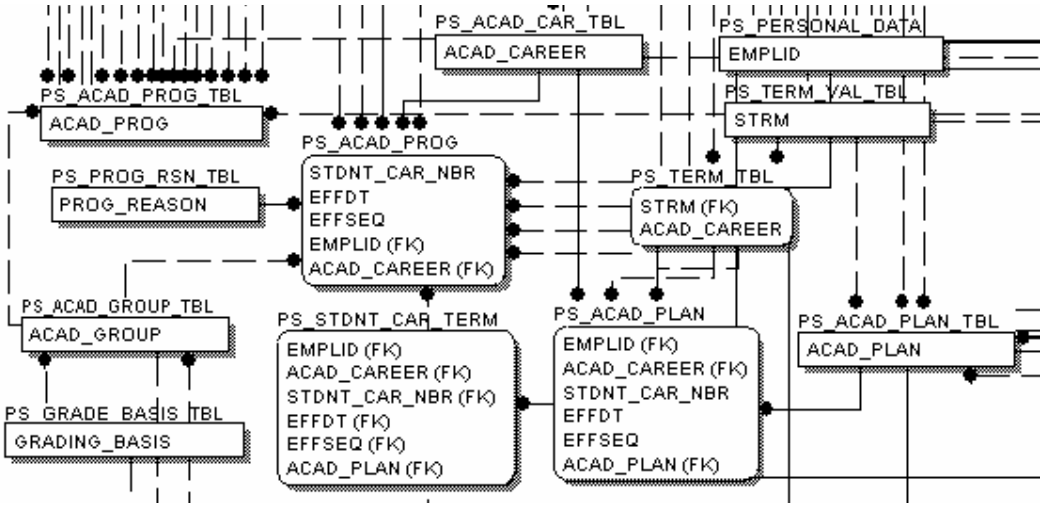

Figure 2-2: ER diagram of OLTP system

2.4.2. ER-limitations in data querying

Stevenson (1997) research shows that "…in order to optimize update operations in a transactional system, data redundancy is minimised and this makes extracting data complex". Kimball (1996, p.9) stated that ER diagrams are very symmetric and all the tables look the same. There is no way to tell which table is the most important, the largest, contain numerical measurements of the business and which tables hold static or near-static descriptions of objects. ER-models should, therefore, not be used as the basis for enterprise DW. ER data models are a disaster for querying and cannot be understood by users and they cannot be navigated usefully by database management system (DBMS) software.

2.5. Difference between OLTP and DW systems

OLTP and DW systems are two entirely separate environments. According to Kirpekar's (2005) research, "OLTP users are different, the data content is different, the data structures are different, the hardware is different, the software is different, the administration is different, the management of the system is different, and the daily rhythms are different". The design techniques and design instinct appropriate for transaction processing are inappropriate and even destructive for information warehousing. To further provide better understanding between OLTP and DW system, Ponniah (2001, p.11) summarised the following main differences between the two systems as shown in Table 2-1.

Table 2-1: Difference between OLTP and DW systems

	OLTP System	**Data Warehousing**
Data Content	Current Values	Archived, derived, summarised
Data Structure	Optimized for transaction	Optimized for complex queries
Access Frequency	High	Medium to low
Access Type	Read, update, delete	Read
Usage	Predictable, repetitive	Ad-hoc, random, heuristic
Response Time	Sub-seconds	Several seconds to minutes
Users	Large number	Relatively small number

2.5.1. Slowly changing dimensions (SCD)

One major difference between an OLTP and a DW system is the ability to accurately describe the past (Kimball, 1996). The large volume of data in an OLTP system is typically purged every 90 to 180 days. Business analysts need to track changes in dimensional attributes (Ross and Kimball, 2005). The DW must accept the responsibility of accurately describing the past and this feature is managed in DW by using SCD. Ross et al., (2005) further stated that SCD can be implemented in DW by choosing Type 1, Type 2, and Type 3 methods:

- Type 1 is most appropriate when processing corrections; this technique won't preserve historically accurate associations. The changed attribute is simply updated (overwritten) to reflect the most current value.

- With a type 2 change, a new row with a new surrogate primary key is inserted into the dimension table to capture changes. Both the prior and new rows contain as attributes the natural key (or durable identifier), the most-recent-row flag and the row effective and expiration dates.

- With type 3, another attribute is added to the existing dimension row to support analysis based on either the new or prior attribute value. This is the least commonly needed technique.

2.6. Why has a separate data warehouse?

An OLTP configured database server could not be used as a basis for DW. An OLTP system is designed and tuned from known tasks and workloads, such as indexing and hashing using primary keys, searching for particular records, and optimizing queries. On the other hand DW queries are often complex. They involve the computation of large groups of data at summarised levels, and may require the use of special data organisations. Processing OLAP queries in operational databases would substantially degrade the performance of operational tasks (Han and Kamber, 2001, p.44).

2.7. Data warehouse or data mart

It is always a big challenge to decide what to build first, a DW or a data mart? Ponniah (2002, p.25) stated that before deciding to build a DW the following basic and fundamental questions need to be addressed:

- Top-down or bottom-up approach?
- Enterprise-wide or departmental?
- Which first—data warehouse or data mart?
- Build pilot or go with a full-fledged implementation?
- Dependent or independent data marts?

Table 2-2: Data warehouse versus data mart

Data Warehouse	Data Mart
Corporate/Enterprise-wide	Departmental
Union of all data marts	A single business process
Data received from staging area	Star-join (facts & dimensions)
Queries in presentation resource	Technology optimal for data access and analysis
Structure for corporate view of data	Structure to suit the departmental view of data
Organisational on ER model	

2.7.1. Data marts

A data mart is a collection of subject areas organized for decision support based on the needs of a given department (Inmon, 1999). There are two kinds of data marts, dependent and independent. A dependent data mart is one whose source is a DW. An independent data mart is one whose source is the legacy applications environment.

Ponniah (2002, p.26) in Table 2-2 provided brief differences between a DW and a data mart.

2.7.2. Top-down versus bottom-up approach

In the top-down approach, the overall corporate-wide data repository is using the ER modelling technique. The enterprise data warehouse feeds the departmental data marts that are designed using the dimensional modelling technique. The bottom-up approach, starts building several data marts using the dimensional modelling technique and the collection of these data marts forms the DW environment.

2.8. DW architecture components

According to Ponniah, (2002, p.29) the three major components of a DW as shown in Figure 2-3 are:

- Data acquisition
- Data storage
- Information delivery.

As can be seen in the Figure 2-3 a data mart is a subset of a DW for use by a single department or function.

2.9. Data acquisition

In this area data from different sources as shown in Figure 2-3 are extracted and moved to the staging area. In the staging area ETL is performed. During ETL each file is extracted by performing various transformations like sort, merge, resolving inconsistencies, and cleansing of the data. After transformation and integration, data is prepared for loading into the DW storage.

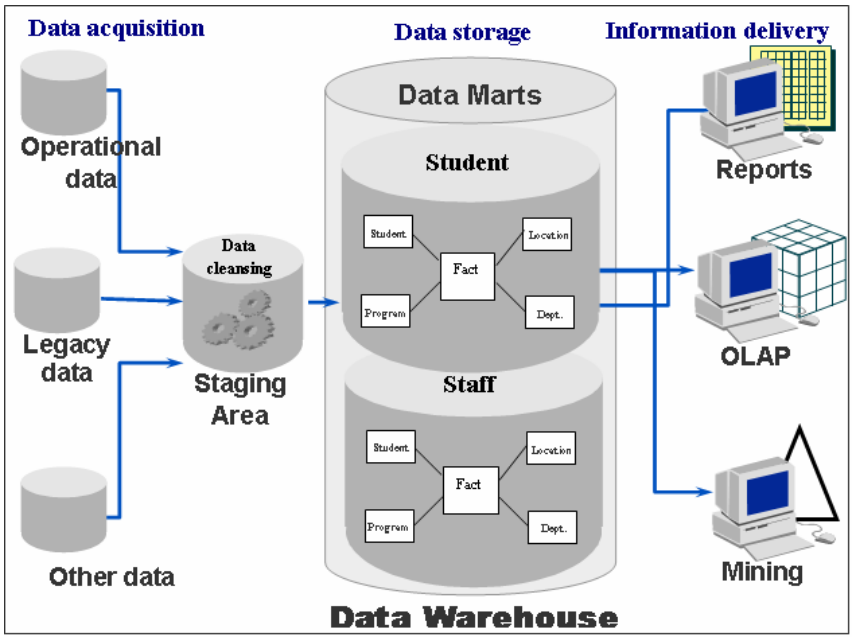

Figure 2-3: Data warehouse architecture

2.9.1. Extraction, transformation and loading (ETL)

ETL is the heart and soul of BI. ETL processes bring together and combine data from multiple source systems into a DW for providing a single version of the truth. Eckerson (2003b) research shows that "ETL design and development work consumes 60 to 80 percent of an entire BI project". ETL functions reshape the relevant data from the source systems into useful information to be stored in the DW. Without these functions there would be no strategic information in the DW. According to Gatziu and Vavouras (1999) "…the fundamental reason for building a data warehouse is to improve the quality of information in the organisation". If the source data is not extracted correctly, cleansed, and integrated in the proper formats, query processing, the backbone of the DW, could not happen. Ponniah (2002, p.259) describes the difficulties in ETL functions due to:

- Source systems are very diverse and disparate.
- There is usually a need to deal with source systems on multiple platforms and different operating systems.
- Many source systems are older legacy applications running on obsolete database technologies.

- Generally, historical data or changes in values are not preserved in source operational systems. Historical information is critical in a DW.
- Quality of data is dubious in many old source systems that have evolved over time.
- Source system structures keep changing over time because of new business conditions. ETL functions must also be modified accordingly.
- Gross lack of consistency among source systems is commonly prevalent. The same data is likely to be represented differently in the various source systems. For example, data on salary may be represented as monthly salary, weekly salary, and bimonthly salary in different source payroll systems.
- Even when inconsistent data is detected among disparate source systems, lack of a means for resolving mismatches escalates the problem of inconsistency.
- Most source systems do not represent data or formats that are meaningful to the users. Many representations are cryptic and ambiguous.

2.9.2. Data extraction

Effective data extraction is the key to success. Here is a list of data extraction issues:

- Source identification: Identify source applications and source structures.
- Method of extraction: for each data source, define whether the extraction process is manual or tool-based.
- Time window: for each data source, denote the time window for the extraction process.
- Job sequencing: determine whether the beginning of one job in an extraction job stream has to wait until the previous job has finished successfully.
- Exception handling: determine how to handle input records that cannot be extracted.

2.9.3. Data transformation / data cleansing

Data cleansing is an important task for DW specialists, database administrators, and developers. Usually, data extracted from OLTP systems contains lots of errors, and must be first transformed and cleaned before it goes into the DW (Gatziu et al., 1999). Data values from OLTP systems can be incorrect, inconsistent, unreadable or incomplete. Furthermore, different formats and representations may be used in the various OLTP systems. Gatziu et al., (1999) stated that data cleansing is an essential task in order to get correct and qualitative data into the DW, and includes the following tasks:

- Convert data to the common, internal warehouse format from a variety of external representations.
- Identify and eliminate duplicate and irrelevant data.
- Transform and enrich data to correct values (e.g., by checking the membership of an attribute in a list).
- Reconcile differences between multiple sources, due to the use of homonyms (same name for different things), synonyms (different names for same things) or different units of measurement.

After cleansing, data that comes from different sources and stored in the same warehouse table must be merged and possibly set into a common level of detail. Data-cleansing techniques come in several forms including de-duplication, validation, and house-holding. De-duplication ensures that one accurate record exists for each business entity, represented in a transactional or analytic database. Validation ensures that each attribute maintained for a particular record is correct. Addresses are a good candidate for validation procedures where cleanup and confirmation procedures are performed as shown in Table 2-3. House-holding is the technique of grouping individual customers by the household or organisation of which they are a member.

Table 2-3: Duplicate customer records

First Name	Last Name	Address1	Address2	City	State	Zip Code
John	Doe	112 Sunny Vale Ln.	Apt. #23	Anytown	NC	28227
John	Do	112 Sunny Vail Lane	Apt 23	Anytowne	NC	28227-5410

2.9.4. Data loading

The whole process of moving data into the DW repository is referred to in several ways. Because loading the DW may take an inordinate amount of time, loads are generally a cause of great concern. During the loads, the DW has to be offline. For loading into the DW a window of time is required without affecting the DW users. Therefore, consider dividing up the whole load process into smaller chunks and populating a few files at a time. This will provide two benefits, parallel load can be applied and a part of the DW will be up and running while loading the other parts. The loading process is generally three of types:

- Initial load: populating all the DW tables for the very first time
- Incremental load: applying ongoing changes as necessary in a periodic manner
- Full refresh: completely erasing the contents of one or more tables and reloading with fresh data (initial load is a refresh of all tables).

2.10. Data storage

The cleansed data from the staging area is stored in the DW using a star schema or also called a dimensional model (Ponniah, 2002). This is the fundamental data design technique for the DW.

2.10.1.Dimensional model

The dimensional data model places all relevant data fields into one of two types of tables, fact tables and dimension tables, as shown in Figure 2-4. There is one large dominant table in the centre of the schema. It is the only table in the schema with multiple joins connecting it to other tables. The other tables are connected to the central table with a single join. The central table is called the fact table and the other tables are called dimension tables (Kimball, 1996, p.11).

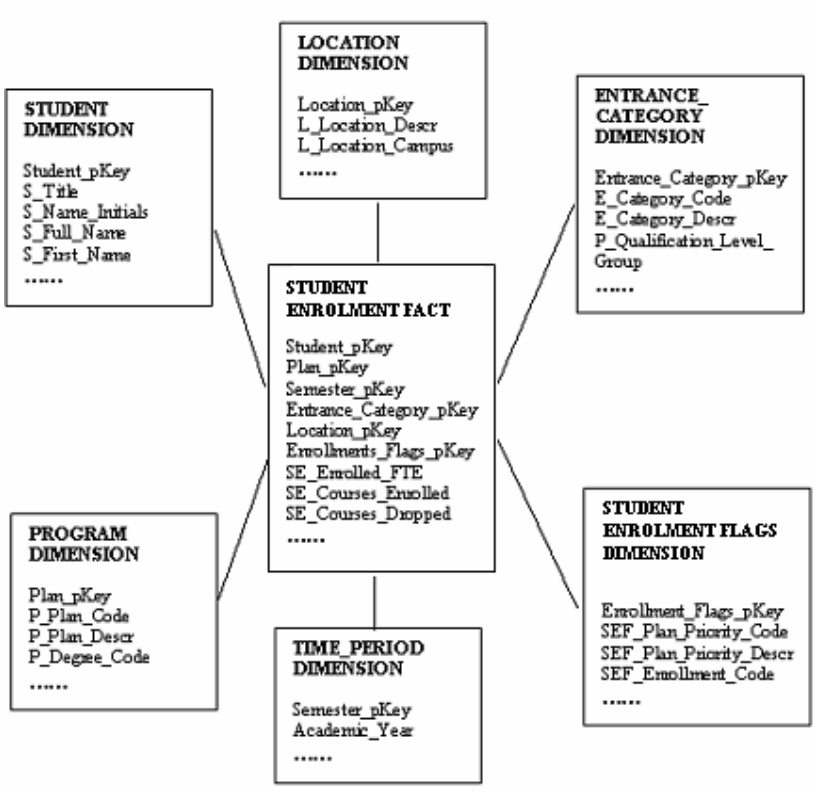

Figure 2-4: Student enrolment star

2.10.2.Fact table

The fact table stores the measures of the business. The best and most useful facts are numeric, continuously valued, and additive. In general, facts are those attributes, usually quantitative, that users wish to measure about a subject. The data grain is an important characteristic of the fact table and is the level of detail for the measurements or metrics (Ponniah, 2001). Due to the data grain (summarisation or aggregation) the large number of records will be compressed into a few dozen rows of the user's answer set. If the measurements are numbers and if they are additive it is

very easy to build the answer set (Kimball, 1999, p.12). Some of the business facts are semi-additive and non-additive. Semi-additive facts can be added along only some of the dimensions, and non-additive facts simply can't be added at all. For non-additive facts the only options is to summarise the records using a count. Some fact tables may just contain summary data and are called aggregate fact tables.

2.10.3.Dimension table

The dimension tables are where the textual descriptions of the dimensions of the business are stored. The best attributes are textual, discrete, and used as the source of constraints and row headers in the user's answer set (Kimball, 1996, p.13). A key role for dimension table attributes is to serve as the source of constraints in a query or to serve as row headers in the user's answer set. Dimension tables consist of sets of highly correlated descriptive attributes that can be placed in an obvious category (Allan, 2000).

2.11. Information delivery

The information delivery component makes it easy for the users to access the information either directly from the DW or from the dependent data marts. The DW can satisfy users of all levels of management by providing data in the form of reports, Excel pivot tables, OLAP and data mining technologies.

2.11.1.Online analytical processing (OLAP)

OLAP technology organizes data in multidimensional tables (also called cubes) and provides access to the DW through an interactive graphical user interface (Gorla, 2003). OLAP tools are for users who require intensive data analysis capabilities.

2.11.2.Types of OLAP tools

The business problem that OLAP tools solve is the need for users to "drill" seamlessly into information when additional details are required. OLAP tools provide users with ad- hoc access to data on an as needed basis. These tools insulate users from the details surrounding the retrieval of information from the DW (Trepte, 1997). Trepte (1997) further stated that the driving factor behind selecting a particular OLAP tool is the size and volume of data that needs to be analyzed. More importantly, the different types of OLAP tools solve different business problems. Two major types of OLAP are

multidimensional online analytical processing (MOLAP) and relational online analytical processing (ROLAP). In MOLAP data is cleaned, aggregated in a multiple-dimension database (MDDB), and uploaded into a data cube periodically. It is used for large departments or groups because it supports large amounts of data and users. In ROLAP data is aggregated and stored along with relational databases management systems (RDBMS). Gorla (2003) indicated that MOLAP tools make the DW system easy to use but not useful. ROLAP tools make the DW useful but not easy to use. Gorla (2003) further provided differences between MOLAP and ROLAP technologies as described in Table 2-4.

Table 2-4: Difference between MOLAP and ROLAP models

MOLAP	ROLAP
MOLAP for non-sophisticated computer users.	ROLAP for the sophisticated users.
Users who use only preset reports and have no need to monitor the daily transaction.	Users that need to analyze the market information regularly.
If users need consistent information over a period of time.	If the requirements change frequently.
MOLAP should be used where data is relatively non-volatile.	ROLAP should be used for volatile data environment.
Model is recommended in the initial stages of adoption.	Model is recommended after considerable experience.

2.11.3.Data mining

"…Data mining refers to extracting or mining knowledge from large amounts of data" (Han, 2001, p.5). The benefits of data mining are its ability to gain deeper understanding of the patterns previously unseen, using current available reporting capabilities. The enterprise DW, either as a centralized repository feeding dependent data marts or as a conglomerate of conformed data marts on a bus structure, forms a very useful source of data for data mining.

2.12. What to put in the DW?

A DW must deliver the right data to the right people. However, the DW cannot deliver all the data people want. People are always asking new questions, so predicting what data they need is difficult (Porter and Rome, 1995). According to best practices it is not a good idea to put everything in the DW. Firstly, it increases the length of the DW implementation, in most of the cases time spans to more than 3 years. Secondly, after three years user will only be able to see how the reporting from DW differs from OLTP and what value user can get from DW.

2.13. Metadata

Metadata is a key architectural component of the DW and describes all the pertinent aspects of the data in the DW fully and precisely (Ponniah, 2002, p.35, 174).

METADATA	
for End-Users	**for IT**
Data content	Source data structures
Summary data	Source platforms
Business dimensions	Data extraction methods
Business metrics	External data
Navigation paths	Data transformation rules
Source systems	Data cleansing rules
External data	Staging area structures
Data trasnformation rules	Dimensional models
Last update dates	Initial loads
Data load/update cycles	Incremental loads
Query templates	Data summarization
Report formats	OLAP system
Predefined queries/reports	Web-enabling
OLAP data	Query/report design

Figure 2-5: Metadata essential for end-users and IT

2.14. Summary

This chapter has covered the second activity of the AR cycle on "fact finding in relevant literature". DW definitions and concepts were explained in this chapter to provide enough knowledge to the reader in order to understand the rest of the dissertation. First of all, the term BI was explained followed by the importance of DW in establishing BI infrastructure. The comparisons between DW and OLTP systems made it clear where OLTP stands. DW architectural components that are data acquisition, data storage and information delivery were explained, followed by metadata that plays an important role in DW architecture. The next chapter will highlight numerous data issues of OLTP systems to give the reader insight as to what happens when an organisation replaces its old legacy system with a new OLTP implementation in order to get quality and integrated data.

Chapter 3

Problems in the existing OLTP system

3.1. Introduction

In the previous chapter an overview of DW concepts was provided. OLTP and DW systems were compared, and the place of each in an organisation was indicated. This chapter is purely focused on the OLTP portion of the research and is the third activity, "Planning & Designing", in the AR cycle. The focus will be on numerous data errors and inconsistencies that exist in the OLTP system, which can be the result, when a TI implements a commercial product for an integrated solution. Being aware of these problems is important because to have quality data in the DW, the extract, transform and load process must fix these problems.

3.2. History of student record systems at the UFS

The University of the Free State (UFS), established in 1904, is among one of the oldest universities in SA. It has computerized student records from 1946 up to today. The University wanted to preserve this historical data because of its value for business trend analysis (Gatziu et al., 1999). "...Trend analysis expands the University's ability to understand student progress, achievement patterns, core effectiveness, spending patterns, and staffing trends and other topics that have implications for its core missions" (Evans, n.d.).

3.3. Legacy system

The first system designed for keeping student records was developed using COBOL programming language and a hierarchical database that ran on the Sperry Univac machine. To load data onto the main frame, a Mapper package was purchased that ran on an Ontel minicomputer. With the passage of time the system became obsolete in order to meet the growing requirements of the University. The limitation of the system forced the University to seek other solutions.

3.4. IBM system

In 1986 the University moved to an IBM relational database. The Cross System Product (CSP) programming language was used to develop an in-house system. The

data was transported from the previous COBOL legacy system by writing in-house transformation programs. Data was exported into flat files and afterwards data from flat files was imported into the SQL database. Initially the developers were happy with this in-house application due to the fact that it was easy for them to update the system according to policies of the University, which changed every year. Over time, however, the system failed because of programming languages that became obsolete, developers having left the University and the system being designed in patches and phases, which resulted in a lack of data integrity and inconsistency in the system interface design.

3.5. PeopleSoft OLTP system

After the bad experience with the IBM in-house development, the University decided to buy an ERP system from an international company. The fact that more and more universities were opting to implement integrated software packages from this company helped in making the decision. In 2003 the University purchased this new system, to be used as the OLTP system, at a huge cost by considering the following factors: a complete package with design consistency, analytical and strategic reporting capability, new technology and full technical and maintenance support.

3.6. Problems in the current OLTP system

Within one year after the installation of the new OLTP system, the University faced a number of new challenges that they had never considered when they purchased this commercial product. During the course of data transfer from the old to the new system (with a different database design) numerous data conversion errors generated anomalies and a lack of integrity in the database. The new system also proved inadequate to provide the necessary statistics. Another problem was the lack of customization of the product and the University is now not in a position to afford the customization costs. Swartz and Orgill (2001) stated that one of the biggest problems in ERP project implementations arises when the institutions attempts to customize the new system to fit every existing business practice.

In the following sections details of the problems that the UFS encountered with the new OLTP system will be provided.

3.6.1. Size and complexity of the database

The PeopleSoft database holds more than 16,000 tables for storing Student, Human Resources, Financials and Accounts data. Not all of these tables are kept up to date. There are a number of tables having no data because of the commercial design of the product that is installed in number of other universities around the World. Another big factor is the complexity of the product due to multiple joins as shown in Figure 2-2.

3.6.2. Lack of data standardisation/interaction

In a relational database the size and data types of columns must be the same throughout the database because this is essential for enforcing referential integrity or joining tables. In the OLTP system there are a number of cases where the columns do not have the same size, for example the Acad_Plan column has different column widths in the tables listed in Table 3.1.

In most of the cases, the information that can be shared in different systems was entered in different formats. For example, the staff-id in one Human Resources database was entered with fifteen characters 00000000000012 and in the Financial or General Ledger database it was entered as a 12 character value. Similarly, one of the major problems that were identified in the OLTP system is the lack of standardization in column names. For example the academic plan is named Acad_Plan in the master table PS_ACAD_PLAN_TBL and in other tables it is named Enrl_Actn_Rsn_Last in PS_STDNT_ENRL_TBL (keeps record of student module registration and grades) and Enrol_Action_Reason in PS_ENRL_REQ_DETAIL_TBL (keeps record of plan that student has dropped) as shown in Table 3-1.

Table 3-1: Column with different names and width

Table name	Column name	Size
PS_ACAD_PLAN_TBL	Acad_Plan	Varchar2(10)
PS_ACAD_PLAN_TBL	Acad_Plan	Varchar2(10)
PS_STDNT_ENRL_TBL	Enrl_Actn_Rsn_Last	Varchar2(4)
PS_ENRL_REQ_DETAIL_TBL	Enrol_Action_Reason	Varchar2(4)

3.6.3. Redundant tables

More than one table was created in the database for storing the same information. Tables 3-2 to 3-4 are listing examples of such duplication where department-ids were entered in three different tables having different descriptions.

Table 3-2: Subject table

Subject	Effdt	Descr	Acad_Org
014	31/08/2004	School of Allied Health Prof.	014
017	01/01/1900	SHOOL FOR MEDICAL SCIENCES	017
075	01/01/1900	MUCPP	075

Table 3-3: Academic organisation table

Acad_Org	Effdt	Descr
014	01/01/1900	School of Allied Health Prof.
017	01/01/1900	School for Medical sciences
075	01/01/1900	MUCPP

Table 3-4: Department table

Deptid	Effdt	Descr
014		
017	01/01/1910	Health sciences general
075		

From the above tables it is possible to see that the description for academic organisation 017 was misspelled in Table 3-2 whereas a different description was found in Table 3-4 for the same organisation. In addition, the department table as in Table 3-4 does not contain any information for ids 014 and 075. Consequently it became difficult for the end-user to decide which tables to use for collecting departmental information.

3.6.4. No referential integrity

The primary job of a relational database is to enforce referential integrity between tables. A major problem with ERP systems is that they never come up with referential integrity because whenever there is an upgrade it overwrites all of the previous referential integrity constraints. Validation is mostly provided from front-end screens, but it is very easy for data entry operators to bypass these front-end validations while entering data.

3.6.5. Unique key violation

An academic plan belongs to only one academic programme. From Table 3-5 it can be seen that there were multiple entries for academic plan 7513 in the Acad_Plan

column, with different academic programmes, having different effective dates which are a violation of business rules.

Table 3-5: ACAD_PLAN_TBL with no primary key

Acad_Plan	Effdt	Acad_Prog	Degree
7513	1/1/1901	M7131	7513
7513	1/1/1902	Q7D3	7513
7513	1/1/2004	M7131	7513

There were 42 other different academic plans having the same business violation. Therefore, special queries need to be written, using the maximum effective date, in order to find the latest or accurate academic programme, for the corresponding academic plan. Table 3-6 lists another example where a module or course (Module ECO221) was entered two or more times with different course-ids. There were 37 different modules having more than one course-id.

Table 3-6: Modules having more than one course identification number

Crse_Id	Catalog_Nbr	Acad_Group	Effdt	Descr	Students_Enrolled
003452	ECO221	MHUM	1/1/1901	Eco221 Macro Economics	
003452	ECO221	MHUM	1/1/2001	Eco221 Macro Economics	
003452	ECO221	MHUM	1/1/2002	Eco221 Macro Economics	
003452	ECO221	MHUM	1/1/2003	Eco221 Macro Economics	
020033	ECO221	QEMS	1/1/2004	ECO221 Economics	1
003452	ECO221	MEMS	1/1/2004	Eco221 Macro Economics	7
020026	ECO221	QEMS	10/20/2004	Economics	

From the above table it can be seen that module EC0221 has three different course-ids and that course-id 003452 was entered five times with different effective dates as indicated in the Effdt column (see Table 3-6). In the previous example of Table 3-5 it was mentioned that using the maximum effective date, the last entry can be identified but, what is currently happening in the system is that even in 2006, enrolments were made using old course-ids i.e., seven students were registered with 003452 and one with 020033 where no entry was found for the latest course-id with effective date 10/20/2004 as shown in Table 3-6. In the same way duplicate entries can be made in

any of the master and child tables in the OLTP system and the database will never give an error.

3.6.6. Data capturing problems

The navigation between data entry screens has made it difficult for data entry operators to capture data. For example, while entering student demographic information the data entry operators have to navigate through a set of screens to enter all the required information. For capturing student demographic information there is a separate screen for name, address, contact numbers, nationality, national-id or passport, ethnicity, etc. This allows the data entry operator to save a student record by entering partial information and the system never prompts or gives an alert for this incomplete record entry.

3.6.7. Typographical errors

Student demographic information is very important for drilling down to country, state/province and city level. In the current OLTP system there is no way for standardizing city or other such values. Table 3-7 lists an example of the city name, "Bloemfontein" that was entered with 16 different spellings while entering student demographic information.

Table 3-7: Inconsistent city names stored in PERSONAL_DATA table

City_Name	Records_ Entered
BOLEMFONTEIN	1
BLOEMFONTEIIN	2
BLOEMFONTEN	38
BLAOEMFONTEIN	1
BLOEMF9NTEIN	1
BLOEMFSIDE	1
BLOEMFONTEIN	38738
BLOEMFONYEIN	2
BLOEMFONTIN	7
BLOEMFONTEIN-	1
BLOEMFONTEON	4
BLOEMFOTEIN	38
BLOEMFONTEIMN	1
BLOEMFONTEION	1
BLOEMFONTIEIN	2
BLOEMFONTEINM	2

3.6.8. Missing academic programme and academic plans

In the UFS a qualification, which is called an academic programme in the OLTP system, can be broken down into sub-qualifications called academic plans. During registration the academic programme and academic plan is captured in the registration tables. When a description of an academic programme and its academic plans is needed, a join is required between the academic programme table PS_ACAD_PROG_TBL and the academic plan PS_ACAD_PLAN_TBL. The data migration from the old system resulted in academic plans having no academic programme entries in the academic programme table. For example, there are 808 combinations of academic programmes and academic plans for which students were enrolled in the past, and now these combinations are no longer valid, or exist in the base academic programme table. In Table 3-8 it is shown that there were a number of history records for students with combinations of academic programme and plans in PS_STDNT_ENRL_TBL having no corresponding entries in the parent PS_ACAD_PLAN_TBL. Due to this reason there is no way to extract the description of academic plans and degrees for which these students were registered.

Table 3-8: Past academic programme and plan combinations

Acad_prog	Acad_plan	Student_ enrolled
0000	5920	7
E3000	3301E	24
R6000	R6313	1
M7D1	7213	243
Q6000	6005	111

3.6.9. Missing links

When a student successfully completes his degree, entries are made in the PS_ACAD_DEGR_PLAN_TBL and the PS_ACAD_DEGREE_TBL tables for graduation purposes. There were 7440 records/students whose entries were found in the PS_ACAD_DEGR_PLAN_TBL and the PS_ACAD_DEGREE_TBL tables, but no corresponding entry was found in the PS_STDNT_ENRL_TBL table, where the registration record is kept. Table 3-9 is listing examples of students with such missing academic plans.

Table 3-9: Graduates having no direct link with PS_STDNT_ENRL_TBL table

Year_of Graduation	Students
1900 – 2003	7073
2004	218
2005	119
2006	30

From Table 3-9 it can be seen that the migrated data from 1900 to 2003 have 7073 such cases where no record exists in the registration tables. Data problems were also shown for the years 2004, 2005 and 2006 when live data was captured. This data problem was noticed when students were awarded degrees other than that for what they were enrolled for. For example, a student registered for a four year bachelors program can request the University to award him a three year bachelor's degree if he does not want to continue his studies. Such requests are processed manually because the OLTP System is not capable to establish these links.

In the same way there were students whose record exists in the PS_STDNT_ENRL_TBL table and other detail tables but, no related record exists in the student demographic table, PS_PERSONAL_DATA_TBL. There were 4618 graduated students before 2004 with no demographic information in the PS_PERSONAL_DATA_TBL table.

3.6.10.Inconsistencies in data

The normalized database structure allows entering or modifying of data in one place. This concept was managed poorly in the current OLTP system. For example, the academic career for an academic plan entered in one table was found entirely different in another table for the same academic plan. Table 3-10 and Table 3-11 are displaying such anomalies in the data. The academic career for the module RIS121 was entered with "HSUG" in the PS_STDNT_ENRL_TBL table that holds student registration information. The same module RIS121 was entered with a "UGRD" academic career in the PS_CLASS_TBL table where class schedules are stored. This academic career column exists in a number of other tables and in most of the cases information does not match.

Table 3-10: Academic career in STDNT_ENRL table

Strm	Enrl_actn_rsn_last	Calatog_nbr	Acad_career
2052	8242	RIS121	HSUG
2041	6500	RIS134	PGRD
2041	9211	SDW701	UGRD

Table 3-11: Academic career in CLASS_TBL table where classes are scheduled

Strm	Calatog_nbr	Acad_career
2052	RIS121	UGRD
2041	RIS134	UGRD
2041	SDW701	PGRD

3.6.11.Spaces in mandatory columns

The only database constraint that can be found in all underlying database tables is the "NOT NULL" constraint which forces users to enter a value for such mandatory columns. This validation was violated during data migration from the IBM machine. Spaces were added in such mandatory columns where no corresponding value was found. Table 3-12 lists an example of mandatory columns having spaces where important information needs to be captured.

Table 3-12: STDNT_ENRL table

Emplid	Strm	Enrl_Status_Reason	Enrl_Actn_Rsn_Last	Grade_Category
1997823289	2001	DROP		
2002139919	2031	ENRL		NONE
2005033366	2061	DROP		

The PS_STDNT_ENRL_TBL table stores important information about student academic records. For instance it stores the student's academic plan for which he/she was enrolled, the grade category, important dates of enrolment, and modules dropped or withdrawn, etc. From Table 3-12 it can be seen that student 1997823289 has no entry for his academic plan in column Enrl_Actn_Rsn_Last that is a mandatory column. Similarly, for student 2005033366 having Enrol_Status_Reason = "DROP", spaces were added where a value of 20 is required.

3.6.12.Dropped academic plans

A student is allowed to move from one academic plan or degree to another degree. In such cases a student can move from their previous registered degree before the census date. Student fee records are maintained against each degree for which the student is registered. To adjust the student tuition fee record, spaces are added in place of the

academic plan that gets dropped, as shown in Table 3-12 under the Enrl_Actn_Rsn_Last column. This solution to fix the tuition fee creates problems for the MIS department in finding the dropped plans when writing complex queries against audit tables. Table 3-13 shows a record count of dropped academic plans where no information was found in audit tables. There is currently no way to see the academic plans of such students from the OLTP systems.

Table 3-13: Unknown dropped plans

Academic_ Year	Unknown dropped plans
2004	2
2005	2
2006	38

3.6.13.Changed academic plans

In a student information system, changes in data happen very frequently and history gets overwritten with the latest information. These changes usually happen close to graduation ceremonies. Student graduation records were identified where the student was awarded degree in academic plans that were different in which they were registered. The OLTP system allows these changes by overwriting the history. These changes in academic plans were fixed only on the customized PS_STDNT_ENRL_TBL table while the admission application tables PS_ADM_APPL_TBL still keeping previous academic plans. This partial data updating removes the link between the student admission application data from the registration data. Table 3-14 is reporting such cases where the number was high in 2006.

Table 3-14: Student plan replacement with new one without keeping history

Academic_ Year	Changed_Plans
2004	27
2005	38
2006	384

3.6.14.Missing enrolment dates

Student module enrolment dates are required to see either the student enrolled in a particular module before or after the census date. In the old IBM system the module enrolment date got overwritten whenever there was a change in the student module

record. For example, when updating modules for adding marks, the date stamp on Module_Enrolment_Date column got updated with the latest date on which the marks were entered. In this way the original date on which the module was registered is no longer available. Conditions became worse during data migration from the IBM system when there were null or wrong enrolments dates.

3.6.15. Year and semester module conflicts

In the UFS there is a difference between the start, end or census dates for semester and year modules. The OLTP database structure allows entries for semester modules only. For example, 2061 and 2062 represents the first and the second semester of the year 2006 respectively. This scheme works for entering semester modules' start, end, and census dates, but fails when entering year modules.

3.6.16. Product customization issues

Customization is always required in tailoring the system according to the organisation's requirements. Each year product vendors launch patches for product enhancement and bug fixes. To remain updated with the latest versions users have to update their systems with the latest releases. This updating has a major impact on customized modules and tables because the new release overwrites the previous customization.

3.7. Summary

This chapter plays an important role in providing a clear picture of the problems that a TI can face with their OLTP Systems. The commercial OLTP databases come with thousands of tables and interfaces that need customization according to the institution's requirements. Numerous errors in data resulted in a lack of data integrity and inconsistency and made it nearly impossible for a TI to provide statistics to the DoE and to top management of the institution. The decision makers of the UFS understood these data problems and are now seeking a solution from a third party HEMIS system. In the next chapter HEMIS Systems will be discussed in detail with their advantages and disadvantages.

Chapter 4

Higher education management information system (HEMIS)

4.1. Introduction

In the previous chapter details were provided regarding the problems that TIs are facing today due to their dirty and decentralised OLTP systems sitting with numerous data errors. From these OLTP systems statistics of students and staff must be generated on a regular basis and submitted to the DoE for subsidy purposes. Providing clean and correct data to the DoE is now becoming a major challenge for the TIs. Possible help comes in the form of a system provided by the DoE, VALPAC2, and HEMIS systems. These systems together with their benefits and shortfalls will be discussed in this chapter. This chapter can also falls under the third activity, "Planning and designing" of the AR cycle.

4.2. National Database

The DoE is maintaining a National Database to ensure quality education and plan proper subsidies for these institutions. From the National Database, the DoE can monitor where TIs are investing their resources in terms of education and expenses. It is also easy for the DoE to monitor Black transformation and the ratio of Black students against White students who may have different qualifications. In the following sections details are provided on the DoE HEMIS system.

4.2.1. Returns to be sent to the DoE via the VALPAC2 system

The biggest challenge facing the TIs is to increase the number and types of graduates. Due to this reason, the DoE needs unit record statistics of students and staff quarterly or yearly from all TIs in order to plan and giving subsidies to them. In August 1999, the Department provided all institutions the technical details about the collection and specifications for file scopes and file structures. The department maintains a PC-based software package "VALPAC2" to import and validate data in ASCII files. The responsibility for ensuring the accuracy and completeness of the data in the returns provided to the Department rests with the institutions. Institutions must therefore be confident about the reasonableness and accuracy of the data summarised prior to

sending the data to the Department. The following sections will provide further information on DoE submissions in order to provide enough knowledge to the reader about the complexity involved in generating these submissions.

4.2.2. Submission dates for student and staff data

The HEMIS submission for students is divided into three submissions over a two year time span and one submission for the HEMIS staff every year (see Figure. 4.1). HEMIS submissions are validated and submitted in the required format before the deadlines which are set by the DoE.

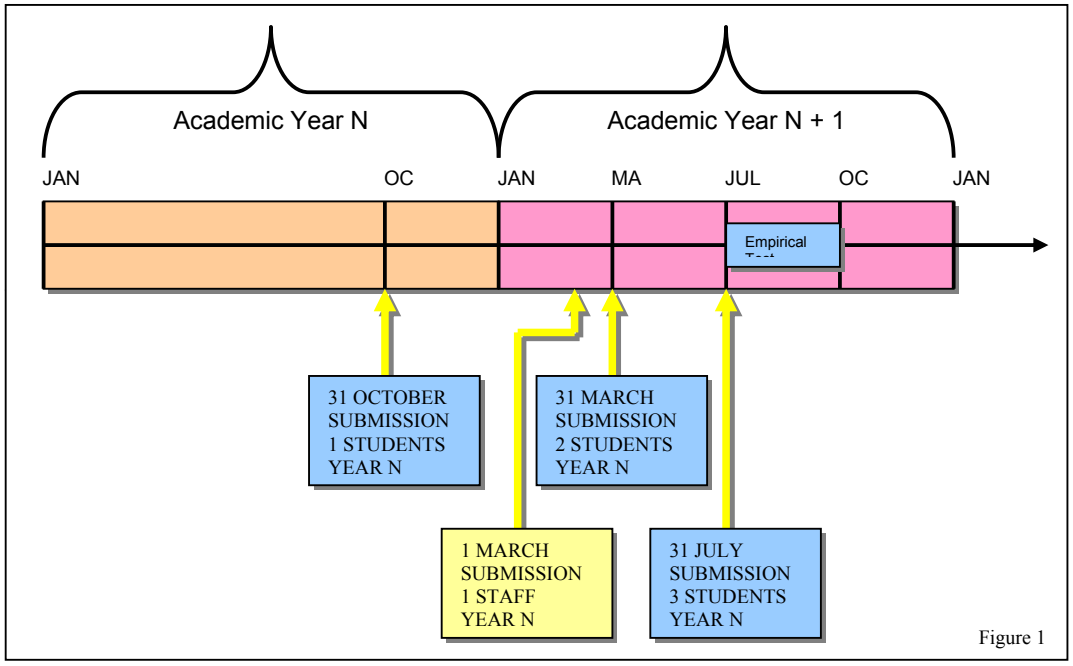

Figure 4-1: HEMIS submission dates

4.2.3. Funding groups

The DoE maintains a funding grid by dividing different qualifications into four different funding groups. Qualifications were divided into these groups by giving high weight to qualifications with a shortage of graduates in the republic. This includes, for example, the Health Sciences and Agriculture. Therefore, the DoE wants TIs to register or attract students in the qualifications that are underprovided in the republic. The funding or subsidies, that are the primary source of income for TIs, are based on the funding group's weight as shown in Table 4-1.

Table 4-1: Funding groups: 2006/07 to 2008/09

Funding group	CESM categories included in funding group
1 Weighting = 1	07 education, 13 law, 14 librarianship, 20 psychology, 21 social services/public administration
2 Weighting = 1.5	04 business/commerce, 05 communication, 06 computer science, 12 languages, 18 philosophy/religion, 22 social sciences
3 Weighting = 2.5	02 architecture/planning, 08 engineering, 10 home economics, 11 industrial arts, 16 mathematical sciences, 19 physical education
4 Weighting = 3.5	01 agriculture, 03 fine and performing arts, 09 health sciences, 15 life and physical sciences

4.2.4. Classification of educational subject matter (CESM)

The CESM was primarily developed as a classification tool that would standardize and facilitate the exchange of subject related data. A CESM must be assigned to each course in order to obtain subsidy for them from the DoE.

4.2.5. Study time

The DoE degrades subsidies for students who never fulfil or complete their studies in the minimum preparatory time as shown in Table 4-2. Such students do not contribute a lot to the institution's subsidy shares. The definition provided for the minimum preparatory time is the minimum number of years (to the nearest 1/10) of full-time post-secondary study that should have been completed before the studying of a particular degree/certificate can commence.

4.2.6. Course credit

Each distinct instructional offering is allocated a weight called course weight. The total credit for all instructional offerings of a full-time student is one per year. From Figure 4-1 the course credit calculation of three years for the B curriculum can be seen.

Table 4-2: Study time in universities

Qualification	Minimum total time (Yrs)	Minimum experiential time (Yrs)	Minimum formal time (Yrs)
B. Com	3.0	0.0	3.0
LLB	4.0	0.0	4.0
Post graduate certificate	1.0	0.0	1.0
Honours B. Com	1.0	0.0	1.0
MBChB	6.0	1.1	4.9
D Phil	2.0	0.0	2.0

For research degrees, in most of the cases, the total credits for Master's degrees are 1.000 and 2.000 for Doctoral degrees. Due to the fact that the research degrees are normally spread over more than one year, a special mechanism has to be devised. In this case, the total credit values are divided by the average time that students take to complete the qualification over the years as shown in Figure 4-2.

Calculation of the Course Credit for Teaching

Undergraduate degree with no experiential time and a fixed curriculum

Example 2: Unisa B Com (new curriculum)

Minimum total time	= 3.0 years		
Minimum experiental time	= 0.0 years		
Minimum formal time	= 3.0 years		
First year	10 modules	1/10 = 0.100 each	
		10 x 0.1000 =	1,000
Second year			
	10 modules	1/10 = 0.100 each	
		10 x 0.1000 =	1,000
Third year	2 courses	1/2 = 0.500 each	
		10 x 0.1000 =	1,000
Total			3,000

Figure 4-2: Calculation of the course credit for teaching

Since students often do not follow the fixed curriculum, the original credit values would then have to be adjusted. The course credits could be adjusted per year for each degree for practical reasons. An adjustment factor is then calculated that will satisfy the 2% (now 0%) test. This is then multiplied with the original course credits to obtain an adjusted course credit that would satisfy the 2% (now 0%) test.

Calculation of the Course Credit for Research Offerings

Number of graduates for the degree or cluster

Number of years the graduates were enrolled

= Average time for completion

Credit value = Total credit value X Average time for completion

Where total credit value would typically be 1.000 for research masters and 2.000 for doctoral degrees.

Figure 4-3: Calculation of the course credit for research offerings

The OLTP systems that TIs are currently using only capture basic information and never support calculations that the DoE asks for. Data is extracted into flat files or spreadsheets for performing a 2% test and the rest of the calculations. On the other hand, the DoE has provided a VALPAC2 system to all TIs for standardising the submission.

4.2.7. Full-time equivalent (FTE)

A FTE student total for a course is determined by this formula:

FTE student enrolments = credit value for course X head count enrolment for course

The head count enrolment for a course is the total of students enrolled for that course on the census day determined by the institution. The credit value of a course is the fraction it constitutes of a standard full-time curriculum.

4.2.8. Teaching input unit (TIU)

The formula in calculating the TIU is given below:

TIU = Total FTE * HEMIS Unit Level Subsidy Level * Funding Factor for the CESM Category/Sub-Category

The values for the HEMIS Unit Level Subsidy Level and the Funding Factor for the CESM Category can be obtained from the grid as shown in Table 4-3 (Funding Formula – input subsidy (enrolments)).

4.2.9. Funding formula

The funding formula for teaching input and output subsidies are described below:

Teaching: (Rand value for 2004 TIU: rec'd for 2006)

a) Input subsidy (enrolments) – currently 64% (Apply per 1 FTE or individual course weight) as shown in Table 4-3.

Table 4-3: Input subsidies

FUNDING GROUP	CESMS	COST FACTOR/ LEVEL:	UG 01,02,04	PG 03,05	M 06,07	D ** 08,09
			1	2	3	4
1	07 Educ 13 Law 14 Libr 20 Psych 21 Pub Adm/Soc Wrk	1.0	1.0 TIU R 6 900.00	2.0 TIU R13 800.00	3.0 TIU R20 700.00	4.0 TIU R27 600.00
2	04 Commerce 05 Communications 06 Computer Science 12 Languages 18 Philosophy 22 SocSci(inclEcon..)	1.5	1.5 TIU R10350.00	3.0 TIU R20 700.00	4.5 TIU R31 050.00	6.0 TIU R41 400.00
3	02 Archi, Building 08 Engineering 10 Home Ec 11 IndArt,Trade,Tech 16 Math Sciences 19 Physl Education	2.5	2.5 TIU R17 250.00	5.0 TIU R34 500.00	7.5 TIU R51 750.00	10.0TIU R69 000.00
4	01 (Agric) 03 Visual & Perf Arts 09 Health Sciences 15 Life & Phys Sci	3.5	3.5 TIU R24 150.00	7.0 TIU R48 300.00	10.5TIU R72 450.00	14.0TIU R96 600.00

** NB PhD is 2FTEs therefore double for full degree TIUs

Human Science & Natural Science

b) Output subsidy (qualifications awarded excluding Honours degrees research) – 16% as shown in Table 4-4.

Table 4-4: Output subsidy excluded honours degree

QUALIFICATION TYPE		WGHT	R AMNT
1st (UG) Dip (<2yrs) / Honours / PG Dip	U H X	0.5	R 6 350.00
1st (UG) Dip (>2yrs) /1st Bach / PG Bach	U B P	1.0	R12 700.00
Prof 1st Bach	F	1.5	R19 050.00
Non Research Masters (100% course work)	M	0.5	R 6 350.00

c) Output subsidy (higher degrees awarded and research publications) – 13% as depicted in Table 4-5.

Table 4-5: Output subsidy higher degrees

QUALIFICATION TYPE		WGHT	R AMNT
Masters (100% Dissertation)	M	1.0	R 81 000.00
Ph D	D	3.0	R243 000.00
Research Publication		1.0	R 81 000.00

4.3. VALPAC2 file structure

The data for the unit record collections are to be sent to the department in the form of a Microsoft Access database which is generated by the VALPAC2 system. This is the only format which is accepted by the department. To import data into this database, TIs generate ASCII files by exporting data from their OLTP systems. The VALPAC2 system consists of the following tables/files (see Tables 4-6 and 4-7) for students and staff statistics respectively (see Appendix A to F for detail table structures):

Table 4-6: Student collections files

	VALPAC2 file name
Student file	STUD
Course registration file	CREG
Credit value file	CRED
Qualification file	QUAL
Qualification CESM file	CESM
Course file	CRSE

Table 4-7: Staff profile files

	VALPAC2 file name
Staff profile file	PROF
Staff FTE file	SFTE

The ER diagram for student files as shown in Figure 4-3 presents, how student data is stored in the VALPAC2 file structure. The information on student demographics, qualifications for which they registered, and CESM subcategories all are kept together

in one student file (STUD). So, if a student registers in more than one program, then student demographic information will get repeated a number of times and that can create anomalies in the database. The main important information on student FTE needs to be calculated by writing complex queries using the course registration (CREG) and course credit values (CRED) files. To extract input and output subsidies TIs have to write their own in-house programs.

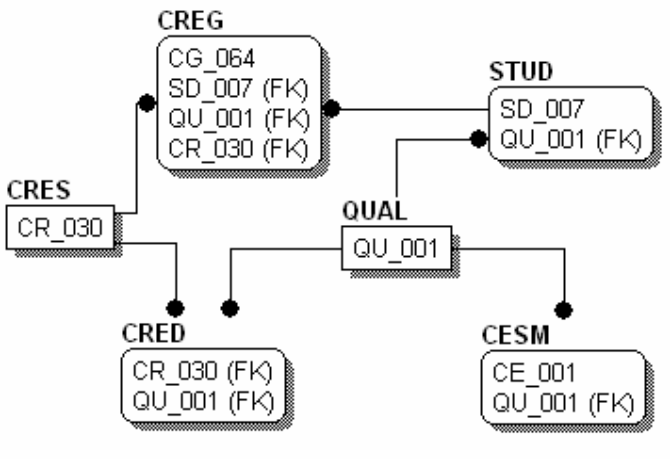

Figure 4-4: VALPAC2 ER diagram

4.3.1. VALPAC2 system and its limitations

The information that is sent to the DoE is limited to only those students and staff who qualified for subsidies. The VALPAC2 systems' validation rejects students who are not registered for qualification (that are not subsidised by the DoE). TI's, however, require such student and staff information to see the complete picture of their institutional data for internal strategic planning (Viljoen, 2006). It was also stated by Viljoen (2006) that the VALPAC2 system is not suitable for institutional internal strategic planning because it preserves data for only two years and that is not enough for forecasting and other predictive modelling.

4.3.2. VALPAC2 reporting

Another big challenge for a TI is to produce effective reporting to top management, faculties and departments. The VALPAC2 system does not support web reporting and therefore the product must be installed for each individual who would like to access the VALPAC2 data. The VALPAC2 system's reporting basically covers data

validation and a few fixed formatted reports. Validation reports are helpful in fixing data while loading.

4.4. Third party solutions

Most of the TIs like UFS, The University of Cape Town, and the University of Pretoria use third party solutions, HEMIS systems, to address issues of decentralised data and user friendly reporting infrastructure. These third party solutions consists of a customised interface by incorporating standard reporting tools like Crystal Reports and Business Objects with their web versions. These third party solutions are using the DoE VAPLAC2 file structures discussed in section 4.3. Data loading in the file structure is based on the ASCII file formats that a TI is generating for the DoE submission. Although the standard reporting tools like Crystal Reports and Business Objects have user friendly interfaces, data extraction is still a big issue. In most of cases, technical staff is required to build and publish reports due to the complex underlying file structures that require a number of joins to retrieve data. In simple terms, data is not ready to use and exists in the form of a transactional file structure that is, in any case, not suitable for reporting. The issue of the enterprise's view of institutional data for internal reporting is also absent. In the following sections benefits and limitations of HEMIS systems are discussed in detail.

4.5. Benefits in using HEMIS

The system's basic purpose is data validation and error reporting in order to fix the data before submissions. In addition, faculties, departments, and top management need information regarding FTEs for planning purposes and this information is available through the VALPAC2 systems. The VALPAC2 system, however, does not have an interactive web interface to publish such information throughout the institution. To fill the gap, the HEMIS system comes into the picture for ad-hoc and interactive web reporting so that it is easy for TI management to centrally publish and maintain the reports.

The HEMIS system works only on the DoE submissions. At any point in time, the TI management can see the exact picture of data that was submitted to the DoE and such data is very useful in calculating future subsidies and in developing certain comparative reports among different previous year submissions.

4.5.1. HEMIS database software

As was mentioned in section 4.3.1, the VALPAC2 system can keep data for only two years. The basic reason why VALPAC2 does not allow data for more than two years is the Microsoft Access database size limitations. Two years of data holds eight submissions and every submission holds more than fifty thousand rows. That exceeds the data handling capacity of Microsoft Access. Alternatively, third party HEMIS systems are designed using better databases, like Microsoft SQL Server that can allow unlimited storage.

4.5.2. Database structure

The HEMIS database follows the ER modelling principles and is similar to the VALPAC2 database structure (see Appendix A to F). The HEMIS database structure can be modified or customised to accommodate institutional data that is not reported or required by the DoE, but plays an important role in the institutional internal reporting. A number of coding schemes like nationality codes, attendance type codes and entrance category codes can be used for an efficient method of storage and fewer efforts can be invested for efficient methods of data retrievals.

4.5.3. Lack of expertise and skills in TIs

TIs are currently lacking in terms of qualified and experienced staff in the field of DW and BI. The existing MIS staff has only been exposed to traditional reporting tools. They struggle to understand and differentiate between traditional and BI reporting environments. The sales representatives of the HEMIS systems can easily convince such staff by showing web technologies for publishing traditional reports using traditional databases.

4.5.4. HEMIS technical support

The existing MIS staff is not capable of running and developing special reports by themselves. They always need assistance from the HEMIS technical support. In most of cases, a TI that is using the HEMIS system has a yearly contract for technical support with HEMIS vendors.

4.5.5. Efficient customization and modification process

The DoE introduced major changes in the HEMIS submission during the HEMIS institute seminar in 2006. Room and space information were also required. Additionally, several changes were made in the STUDENT, QUALIFICATION, COURSE and CREDIT files. This new information is required by the DoE for the 2007 submissions. Use of the HEMIS systems overcomes these necessary changes from the MIS staff because they will get updates from HEMIS vendors for such changes.

4.5.6. Existing hardware and networks support

Another point that motivates a TI to use the HEMIS systems is cheap hardware and networking technologies. All BI tools from Oracle, Microsoft, Teradata, IBM, etc. are resource hungry and require expensive hardware, software and networks. It is not an easy job for the TIs to upgrade as compared to other businesses like Banks, Cell phone companies and Insurance companies.

4.5.7. Data loads from VALPAC2 validated files

Another point that can convince the MIS staff easily to purchasing HEMIS system is the data files that a TI is submitting to DoE. TI MIS staff feels more comfortable when they are analysing the same data that was sent to the DoE. The TI management can then generate internal statistics in determining how much subsidy each faculty or department is going to generate even before the DoE will provides them with such statistics.

4.5.8. Better data validation process than VALPAC2

Some of the HEMIS vendors, like higher education data analyzer (HEDA), have better data validation processes than VALPAC2. Loading data from ASCII files into the HEMIS system helps in identifying errors by sending emails to data owners for fixing errors at the source side. This process even helps in many cases to implement business constraints and to prevent data entry operators from entering incomplete and invalid entries.

4.5.9. Auditing reports

After every five years, a TI gets audited by the DoE for the HEMIS information about student and staff unit records and statistics. A TI institution needs to produce forty different reports of their students, staff and other statistics for the auditing processes. The HEMIS system vendors sell their system with these standard reports and can integrate data from the university and other sources that produce the required reports. Writing these complex reports removes a heavy load from the MIS department.

4.5.10. Ad-hoc reporting

In addition to auditing reports, other reports can be provided to faculties, departments and top management as per request. Complex procedures and functions are mostly used to get the HEMIS technical developers to organise data from the HEMIS database in the format that is required for output.

4.5.11. Microsoft Excel analytical capabilities

The HEMIS system comes with export capabilities to export data intro spreadsheets. Microsoft Excel is the most common tool for data formatting, slicing, dicing and low level analytical capabilities. The MIS staff gets motivated with this export capability of the HEMIS system that provides them with an opportunity to play around with the data using their favourite spreadsheet tools.

4.6. HEMIS limitations and pitfalls

The following sections will draw attention to the limitations of the HEMIS system that were never considered when purchasing these products.

4.6.1. Cannot provide the complete picture

As was explained before, the HEMIS system is based on the subset of institutional data that qualify for DoE submissions. Each TI is sitting with a number of systems to run their business and they need a central integrated system from where they can see a complete picture. For example, research degrees like PhDs and Masters get high subsidies and this information is included in the HEMIS system. However, there is no information about research publications that also contributes in obtaining more subsidies as compared to other professional degrees. Currently, in all TIs, research publications data is sitting in research offices in a separate system. This disintegration

is not useful in analysing research publications to find out which publications areas are subsidised more by the DoE.

4.6.2. Extraction, transformation and loading (ETL)

The HEMIS system is loaded from the six student and two staff text files that a TI is generating from their OLTP systems for DoE submissions. HEMIS systems vendors have not invested as much effort into ETL as compared to that which DW vendors are providing in their products. Several of the HEMIS systems, like HEDA, come with light ETL interfaces for loading data from external files or other databases directly into the HEMIS database. While loading data, a user can specify some validation criteria to accept only those records that satisfy the DoE's business rules.

4.6.3. Replicated data loads

The DoE requires three submissions of students from enrolments to graduation and one submission for staff. In this way, the data is replicated three times in student files and one time in staff files each year. The data, with multiple submissions in a year, may be quite useful for the DoE to see enrolments and graduation statistics. But, for a TI, for internal querying and reporting purposes, the latest status of the submission is required. According to the HEMIS system setup the data replication consumes huge space for data storage and has a significant impact on query performance.

4.6.4. Non descriptive database fields

The DoE database file structure naming conventions are difficult to remember and need decoding at the reporting level. The HEMIS vendors also followed the same naming convention. For example, the STUDENT file fields Student_Number and Gender were coded as SD_001 and SD_012 respectively (see Appendix A for more details). These naming conventions may be beneficial for a transactional database optimisation structure, but from a reporting point of view it is frustrating for developers. They have to rename all the fields by creating views or maintaining a catalogue to find the description of the fields while writing reports.

4.6.5. Complex joins

The basic purpose of the HEMIS system is to provide statistics in terms of student and staff FTEs and requires recalculation each time a query hits the database. The data is

organised in such a format that FTE's need to be calculated per student per course by joining three separate tables as indicated in Figure 4-4. The HEMIS data exists in HEMIS tables per collection year and per submission. Whenever a join is required between tables, a number of joining paths need to be established, as shown in Figure 4-4. The report writer has to remember to put a filter on the submission otherwise queries return multiple rows for the same reporting year due to the presence of three submissions for students.

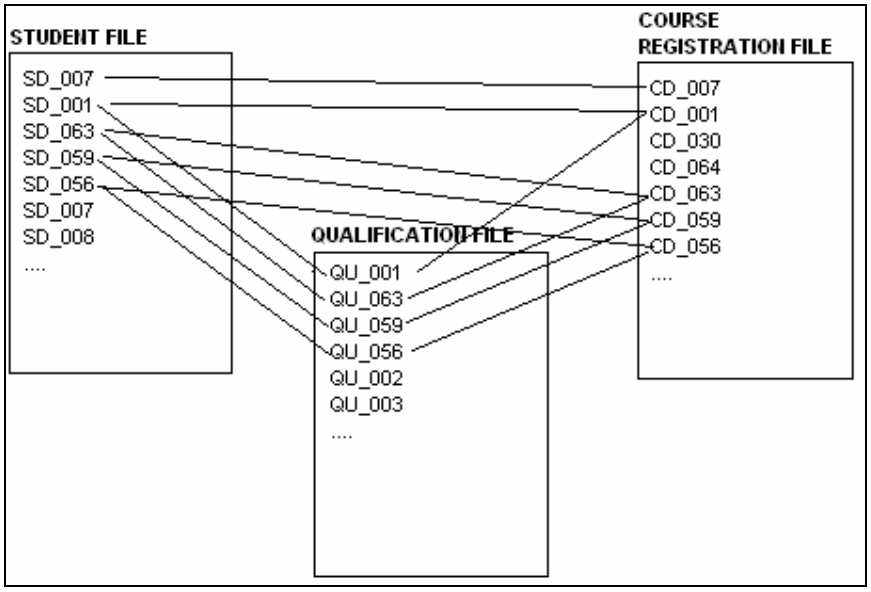

Figure 4-5: HEMIS table's complex join paths

4.6.6. Table joins using textual attributes

Complex joins always results in extremely slow data retrieval that consumes system and staff resources in terms of time and effort. The performance cannot be optimised due to the transactional structure of the database. For example, to see a student course registration with a student name, course name, and qualification name, six joins need to be established between the Course Registration (CREG), Student (STUD), Course (CRSE) and Qualification (QUAL) tables (see Figure 4-3). Another factor that degrades query performance is the joining on textual attributes. This is because no surrogate key concept was implemented in the HEMIS design to reduce the complexity of joins.

4.6.7. Database views for disintegrated data

To reduce efforts for joining different tables views were used on top of the HEMIS tables. This method of creating views ends up with a number of view objects in the database for separate reports. For new reporting requests, MIS staff has to invest a lot of time to find out which existing view will be helpful for the incoming request. If, by mistake, the view gets modified to suite new requirements, it disturbs the previous reports that were based on that view.

4.6.8. End-user layer for reporting

A special end-user layer (EUL) is maintained by the MIS department. This is where all the decoding, transformation and functions are used to enrich the data for an acceptable publishing format. The EUL reporting layer definitely reduces complexity for the users in creating reports, but at the cost of maintenance by developers. Whenever a user runs a report, performance is the biggest issue that cannot be resolved with this type of setup.

4.6.9. Delay reporting

Student registration and demographic information changes very rapidly at the source OLTP system. This change cannot be reflected in the HEMIS tables until a complete load for the next submission is done. In this way, faculties, departments and top management must wait for the next submission for the latest data in order to do important analysis.

4.6.10. Management requirement changes very rapidly

The top management's requirements change very rapidly and they need statistics in the format that is most suitable for their analysis. Each time a new request emerges, the MIS staff has to spend hours to produce data from the HEMIS repository. The biggest challenge is to verify that HEMIS procedures or views that were used to extract data from different tables are providing correct statistics, because complex joins always results in missing or cross join data.

4.6.11. Access to HEMIS data

Throughout the year MIS staff remains busy in producing information for faculties, departments and top management. MIS staff is very scared to give access to the

HEMIS data to business managers and non-technical staff because the database structure is not suitable for non-technical staff and there is always a need to check and verify the join paths and conditions while developing a simple report. It was experienced that non-technical staff cannot understand “EQUI” or “OUTER” joins. They then develop incorrect statistics that create embracement and users start to blame the data for reliability and accuracy.

4.6.12. The HEMIS system provides information not knowledge

Another limitation or disadvantage of the system is its limited capacity for data analysis. Fixed reports or other customised reports that come with the HEMIS system only provide raw data or information. This information is not useful to any of the business managers or staff unless they possess the necessary skills for further analysis. Unfortunately, with the current installation or setup of the HEMIS system, this part is totally ignored.

4.6.13.HEMIS and strategic reporting

Central reporting, data management and transactional reporting are not the only need and requirement of today’s TIs. To survive with tremendous decreases in subsidies, a TI needs to develop their plans and budgets well in advance in order to earn a greater subsidy from the DoE. This can be achieved by generating advance enrolment plans that will be helpful for registering students in areas that generate more subsidies from the DoE. But, unfortunately none of the VALPAC2 and HEMIS systems address this reporting requirement.

4.6.14.Cost of HEMIS systems

Most of the companies, like HEDA, are selling their systems with ongoing maintenance agreements. The cost of this system ranges between R90, 000 to R100, 000 a year. This cost is not much less when compared to the cost of the DW and other BI reporting technologies.

4.7. Summary

This chapter first provided a background on the DoE VALPAC2 system that was handed over to all TIs for maintaining the National Database. The VALPAC2 file structure was explored and it was noted that it is the only acceptable format from the

DoE for data submissions. This was followed by a discussion of the VALPAC2 file structure and its data validations, transformations and shortfalls in using and writing reports. Third party HEMIS system was also discussed in detail that TIs are considering for workable solutions in order to fulfil the reporting requirements of the Institution. Their shortfalls make it clear that HEMIS system is not able to address the issues that a TI is facing today. The main theme behind the HEMIS system is data validation with a fixed set of reports for DoE auditing purposes. This definitely does not solve the problem that TIs seek the most help with. The solution for this is a DW that can integrate and store summarised or detailed data in a suitable format for reporting. The next chapter is on star models for the student data mart.

Chapter 5

Star models of the student data mart

5.1. Introduction

In the previous chapter, the DoE VALPAC2 and the third party HEMIS system were discussed. It was explained that the latter system came into the picture owing to its compatibility with the VALPAC2 file structure. Because of numerous limiting factors, however, TIs are forced to seek help in terms of other solutions. One of the possible ways out is that of a DW that is query centric and capable of answering all the required questions. Designing the DW, star schema diagrams using the dimensional modelling technique, is the most popular methodology. This chapter starts by presenting a theoretical framework of HEMIS and SDM systems. The rest of it is dedicated to presenting star schema diagrams of the student data mart (SDM). This represents the fourth activity, "Action steps", in the AR cycle.

5.2. Theoretical framework (TF)

A TF is a conceptual model of how one theorizes or makes logical sense of the relationships among several factors that have been identified as important to the problem (Sekaran, 2000). In the following sections the researcher will propose theoretical frameworks for both the HEMIS and SDM systems.

5.2.1. Theoretical framework for HEMIS

The TF for HEMIS systems is presented in Figure 5-1. The input to the system comes from OLTP systems. The MIS team generates text files by extracting a subset of student and staff data who qualified for government subsidies by writing complex programs according to the VALPAC2 (Microsoft Access database) file structure. Data is loaded into the VALPAC2 system and the VALPAC2 validation process rejects records that do not satisfy the DoE definitions. Rejected records are rechecked and attempts are made to correct the data on the OLTP side for errors that occurred while capturing the data. Once a load is finalised, the VALPAC2 database in Microsoft Access is submitted to DoE. It was explained in the previous chapter that the VALPAC2 system is not accessible by the TI's faculties, schools and departments for the data that is submitted to the DoE for reporting student FTE's, input and output

units, and subsidies. To fill this gap for the purposes of extracting student and staff FTEs, teaching splits, input and output units, input and output subsidies, third party HEMIS systems are used due to their web interfaces and ease of use in writing reports.

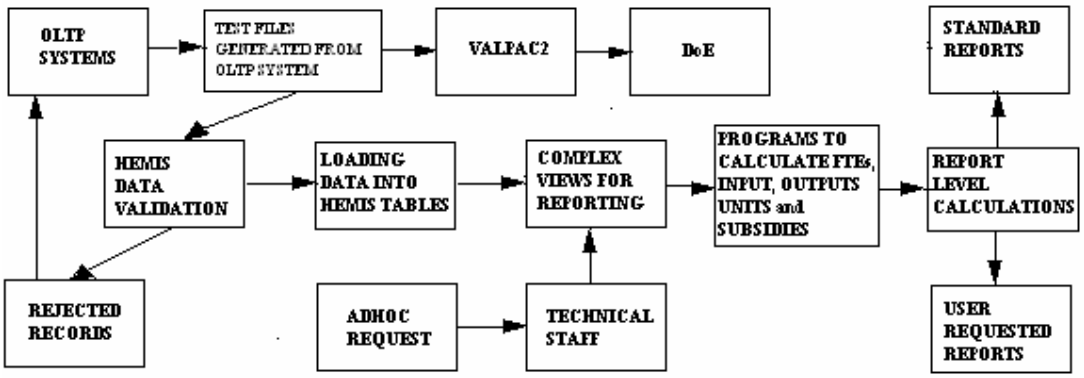

Figure 5-1: Theoretical framework for third party HEMIS system

It is indicated in Figure 5-1 that the third party HEMIS system obtains its loads from the text files that the MIS department generates for the DoE VALPAC2 system instead of from the data that was loaded and validated from the VALPAC2 system. The third party HEMIS system contains its own data validations, data loading rules and setups, together with a mechanism for providing alerts regarding incomplete and invalid data. Owing to these two different setups of data validations with respect to VALPAC2 and the third party HEMIS system, different sets of data results are generated and that create ambiguity in their validity. The end-users consequently lose confidence in the data, because the impression were created that their reports were not reflecting exactly what was submitted to the DoE. It was shown in the Figure 5-1 that complex views need to be written by technical staff for each standard and ad-hoc request. Programs created to calculate FTEs, input and output units and subsidies are used at report level to draw statistics that are required as output. These complex calculations that happen at run time put heavy loads on the reporting server and results in delays in reporting.

5.2.2. Theoretical framework for student data mart

It was demonstrated in the previous section that the HEMIS system operates on the subset of data for students and staff who have qualified for government subsidies. However, faculties, schools, departments, and top management need the complete picture of the institutional data. In the same way other important areas of information

regarding student financial aid, fees, residence, admission applications etc. are totally inaccessible from the HEMIS system. The only sources left for accessing such data are the OLTP systems that contain disintegrated and dirty data. The complexity involved in writing reports from the HEMIS and OLTP systems due to their complex database structure is really a major challenge for the MIS department.

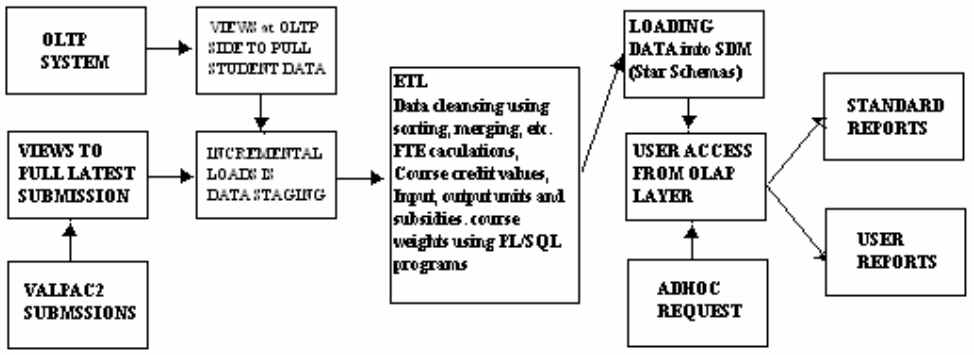

Figure 5-2: Theoretical framework for SDM

These challenges can be overcome by providing access to both HEMIS and OLTP systems' data from one single platform, a SDM. Figure 5-2 illustrates the TF of the SDM. The VALPAC2 system stores a number of definitions for calculating FTE and other important validations according to the DoE definitions. It is not worthwhile to reinvent the wheel in calculating FTE and data validations according to the HEMIS system from the SDM. Therefore, all of the HEMIS facts in the SDM obtain their data from the VALPAC2 system to provide consistent definitions and data. The ETL process collects data from both the OLTP and the VALPAC2 system. Views were written to pull incremental data both from the OLTP system and latest submission from the VALPAC2 system. The ETL processes merge and sort data from both the OLTP and VALPAC2 systems. PL/SQL programs written to calculate FTEs and other facts run during ETL. Finally the data is loaded into the SDM star schemas. End-users are given access to the SDM by OLAP technology which really creates value out of data. In the next sections the SDM will be discussed in detail.

5.3. Student data mart

Poter and Rome's (1995) research indicates that building a DW is extremely complex and takes commitment from both the information technology department and the business analysts of the organisation. In the current SDM it was also a difficult task to identify key business areas and to slot student data records into the star schema to

satisfy the needs of all levels of users. The biggest challenge was the amalgamation of HEMIS and OLTP systems, so that the end-user is able to access both systems on one platform with consistent definitions of codes and descriptions. The presence of HEMIS data in the SDM has increased the value of the data for calculating student FTE's, HEMIS headcounts, and teaching splits between input and output subsidies that generate money for the faculties and departments. After proceeding through the requirement analysis phase and the reports that were generated as per request from top management, faculties, and departments, the SDM was divided into business areas as listed in Table 5-1.

Table 5-1: SDM business areas

Star Model	System support	Description
Enrolment model	HEMIS and institutional queries	Queries about student enrolment/ term registration statistics such as HEMIS and institutional headcounts: which students have dropped courses, qualified, may proceed, research percentages, etc.
Course registration model	HEMIS and institutional queries	Model can answer queries concerning course level statistics such as enrolled, passed, no result, course marks, grades, HEMIS FTE's, HEMIS input, output subsidy, input, output units, etc.
Admission application model	Institutional queries	Application level statistics: applications received, students registered, residence applications, financial aid applications, etc.
Application snapshot model	Institutional queries	Model derived from admission application model and provides information per snapshot.
Undergraduate longitudinal studies model	Institutional queries	This model is derived from the enrolment and course registration models. It is useful to track academic progress or throughput studies of students.
Output subsidy model	HEMIS	Model provides teaching splits as regards teaching and output units and subsidies.

5.3.1. Naming conventions

An entirely different approach is followed in the naming conventions of the OLTP system in contrast to the DW system. For example, the naming convention followed

for Student_Number in the OLTP system is by using SD_0001, whereas in the DW systems descriptive column names were used to provide the report writer with a clear understanding of the data listed in the columns. The naming convention in the SDM was followed using the rule of prefixing each non-prime attribute (columns that are not part of the primary key) with the table initials and in case of HEMIS by adding HM in the column prefix. For example, in the enrolment star as shown in Figure 5-6, columns in the STUDENT_DIMENSION were named by prefixing by S_ while in the same way STUDENT_ENROLMENT_FACT columns were prefixed with SE_.

5.3.2. Grain of the student data mart

According to Allan (2001) the level of detail available in the fact table is referred to as grain. One of the biggest differences between commercial dimensional star models (Accounting, Inventory, etc.) and academic dimensional star models is to be found in the grain of the fact tables. In the case of inventory star models the grain of the fact table can be per month, per product, per customer while the facts columns actually store summarised data according to the product sale per month. The grain of the student fact table is entirely different. In academia the University's top management requires summarised data, while on the other hand business managers, faculties, schools, and departments are interested in detailed data per student. Allan's (2001) research indicates that "….the grain of a student record star schema with dimensions of time (academic year, term), student bio/demo (one record per student), term (one record per student per term) and student matriculation (one record per student per course of study undertaken) would be student per term per course of study". In view of the above definition by Allan (2001) and the requirements of the University's users, the grain of the student fact tables were designed per student, per program, per semester, along with other participating dimensions. Due to the per student grain of fact tables, the student star models' fact table became a factless fact table by storing a value of 1 in the measure column as shown in Figure 5-3.

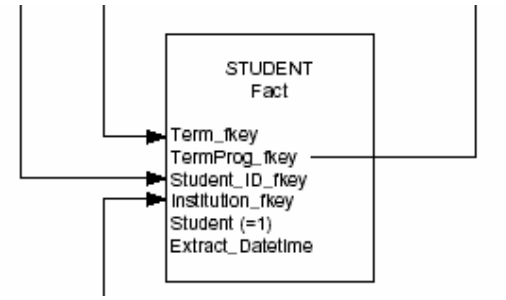

Figure 5-3: Factless fact table

In the current SDM design a fact table contains more than one facts. In the case of the enrolment model, as shown in Figure 5-6, facts are assigned values of 1 and 0. For example, if a student is enrolled in the current academic year the fact SE_Headcount is set to 1 and other facts such as SE_Dropped, SE_Cancelled, etc. are assigned a value of 0.

5.3.3. Summarisation of the SDM

The summarisation of institutional data is different from that in the commercial DW such as Banking, Inventories, etc. In this data mart, course registration is the deepest fact table with millions of rows. The grain of STUDENT_COURSES_FACT as shown in Figure 5-9 is per student, per plan, per semester, and per course in order to gain a course level picture of student records. During the requirement gathering phase it was established that most of the reports used for reporting and planning purposes are based on head counts of students per semester or per year basis. To meet the needs of such reports STUDENT_COURSES_FACT is not the right candidate to pose queries because it will result in slow performance. To resolve this issue, another fact table, STUDENT_ENROLMENT_FACT, is designed in the enrolment model as shown in Figure 5-6. The STUDENT_ENROLMENT_FACT table stores summarised data with grain per student, per semester, and per program and therefore its row size is 13% as compared to that of STUDENT_COURSES_FACT.

5.3.4. Conformed dimensions

Like every data mart, the SDM contains a number of conformed dimensions that simplify management. The following dimensions are such dimensions that are present in most of the dimensional models.

- STUDENT DIMENSION: Student demographic breakdowns per nationality, gender, race etc.

- LOCATION DIMENSION: Primary location (such as the Bloemfontein, Vista, Qwaqwa campuses) from where the student is conducting his/her studies.

- ENTRANCE CATEGORY: To establish the student status of first-time-entering, non-entering, entering, and transfer students.

- TIME PERIOD: DoE requirements regarding data from the TIs are based on the year record in the form of headcounts, FTE, etc., whereas institutional internal reporting is based on semester breakdowns. The TIME_PERIOD dimension is designed to satisfy requirements for both semester and year headcounts.

- PROGRAM DIMENSION: This dimension represents the amalgamation of OLTP system tables PS_ACADEMIC_PLAN_TBL (program details like, program code, title, etc), PS_ACADEMIC_GROUP_TBL (faculty, and department details) into a single picture from a one dimensional table.

5.3.5. Junk dimensions for filtering HEMIS data

A junk dimension is a convenient grouping of flags and indicators. It is helpful, but not absolutely required (Ross et al., 2005). In the current research it was concluded that junk dimensions are very useful in the SDM in order to enrich data for the purpose of providing certain statistics such as separating HEMIS and non-HEMIS students. These statistics are not available in the source OLTP system. In the current case, the STUDENT_ENROLMENT_FLAGS dimension as shown in Figure 5-4 was created. This dimension is then used in the enrolment star (see Figure 5-6). The DoE needs the student's enrolments together with the primary academic plan, but this information is not available in the source data. Similarly a set of filters are used to establish the records of those students who qualify for government subsidies, using filters such as:

- Enrolments before census date (SEF_Census_Descr).

- Barring students who had dropped or withdrawn from their study plans before this date.

- Undergraduate students who fulfil their matriculation requirements (SEF_Matriculation_Descr).

- Barring students who failed or obtained re-assessments in the previous year (SEF_Status_Descr).

This information is processed in the staging area by extracting data from VALPAC2 and OLTP systems, and corresponding flags and indicators were added to the data extracted. A surrogate key was generated in the STUDENT_ENROLMENT_FLAGS dimension by extracting unique combinations of flags and indicators. The size of this junk dimension is 0.035% of the size of the STUDENT_ENROLMENT_FACT rows.

Column Name	Data Type	Length
Enrollment_Flags_Pkey	numeric	9
Enrollment_Flags_Oper_Key	char	8
SEF_Plan_Priority_Code	char	1
SEF_Plan_Priority_Descr	varchar	30
SEF_Matriculation_Code	char	1
SEF_Matriculation_Descr	varchar	30
SEF_Enrollment_Code	char	1
SEF_Enrollment_Descr	varchar	30
SEF_HEMIS_Code	char	1
SEF_HEMIS_Descr	varchar	20
SEF_Census_Code	char	1
SEF_Census_Descr	varchar	30
SEF_Data_Code	char	2
SEF_Data_Descr	varchar	30
SEF_Status_Code	char	1
SEF_Status_Descr	varchar	25
SEF_Extract_Datetime	datetime	8

Figure 5-4: Student enrolment flags dimension

5.4. Student dimension

The student dimension is the widest dimension in this SDM with 59 columns. There are fifteen production database tables, as shown in Figure 5-5, which hold a student record containing many-to-many relationships with the parent OLTP table PERSONAL_DATA.

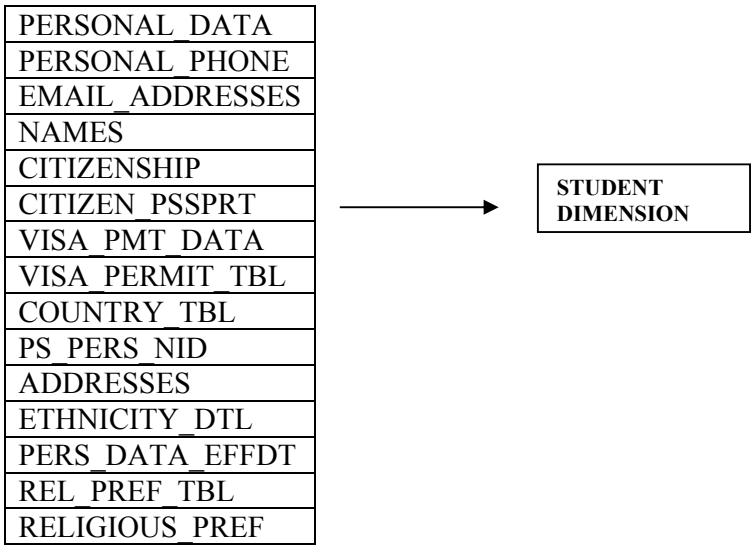

Figure 5-5: Student dimension

To extract data from these fifteen tables, a complex query is required with a number of outer joins and it must establish the maximum effective date. In the SDM it becomes straightforward to use the STUDENT dimension. It is possible to drill down and roll up among different hierarchies in this dimension. The student file (STUD) in HEMIS (see Appendix A) combines student demographics along with qualification, entrance category, qualification attendance mode, CESM categories, and % research for Masters. The student dimension in the SDM holds only student demographics as shown in Figure 5-5. The rest of the information not linked directly to the student demographics is moved into other appropriate dimensions and fact tables as described below:

- Qualification: This information is made available from the PROGRAM dimension.

- Entrance category: A separate dimension, ENTRANCE_CATEGORY, is created and is present in the student enrolment star, course registration, and admission applications star.

- Attendance mode: This information is made available through the ATTENDANCE_MODE dimension in the student enrolment star.

- % Research for Masters: To display the student research area a dimension named RESEARCH_AREA is added in the student enrolment star while % research is provided through STUDENT_ENROLMENT_FACT.

- CESM categories: The information for major CESM categories is made available through the MAJOR_CESM dimension in the student enrolment star.

5.4.1. Student address dimension

In some of the examples of a student data mart in the literature, a separate dimension, STUDENT_ADDRESS, was designed to track changes in the student addresses per term. In the case of the UFS an overall total of 6.49% of students were found in the University data from 1992-2005 who had changed their addresses more than once during their study period. In the 2004 enrolments only 2.8% of the students were found with changed addresses. Owing to these statistics the STUDENT_ADDRESS dimension was excluded from the design because it would only have occupied extra space on the hard disk. Type three changes can be used in the STUDENT_DIMENSION to store students whose addresses have altered.

5.5. Enrolment model

The star model for student enrolments as shown in Figure 5-6 is the most important and query centric model in this SDM to answer both HEMIS and institutional queries concerning student headcounts and other important facts as shown in Figure 5-7. The grain of this star model is per student, per semester, and per program along with other participating dimensions in the fact table. The model is divided into eight dimensions, five of which are conformed along with three other dimensions specific to this model, as described below:

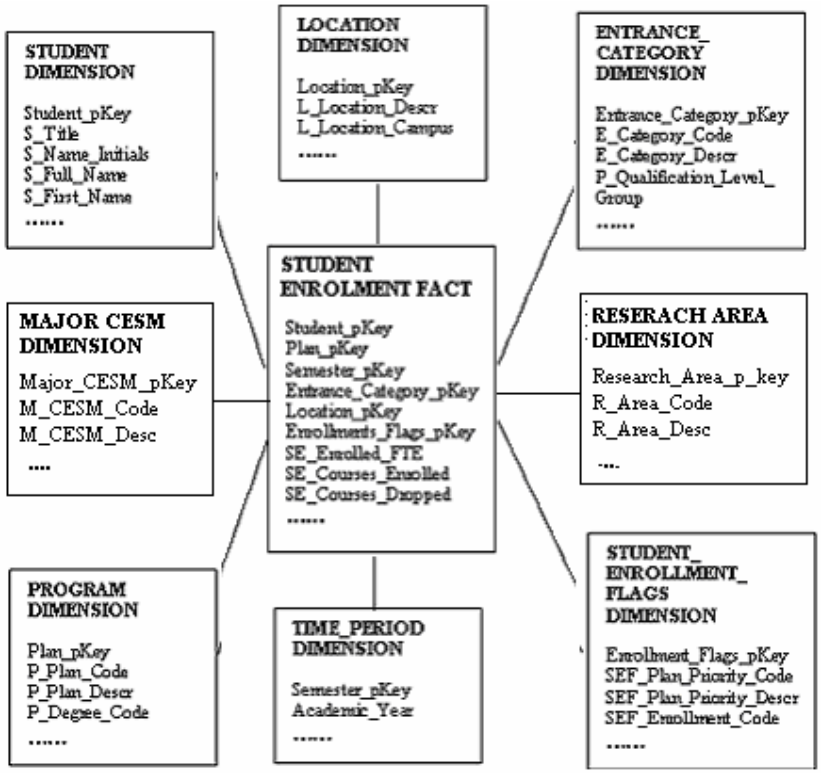

Figure 5-6: Enrolment star

- STUDENT ENROLMENT FLAGS: This most important junk dimension is able to answer queries using a number of flags extracted from the data that is described as below:
    - The student plan priority flag can answer queries in terms of reporting the primary plan if a student is registered during the same academic year in more than one program.
    - Matriculation flags are used to filter students whose matriculation records are missing.
    - Employ subsidy flags are used to segregate students who qualify for subsidies.
    - Data validity flags are very useful to separate the data referring to the VISTA and Qwaqwa campuses from the rest of the University's data.
- RESEARCH AREA: This dimension is useful in providing the Honours and course level Master's research area or field of specialization.

- MAJOR CESM: This dimension is able to provide the second order CESM code which depicts a major field of study approved by the Minister for a particular qualification (if the qualification is not being undertaken by an occasional student).

The presence of features such as student headcount, attendance mode, % research masters, major CESM (which in HEMIS were incorporated in the STUD file (see Appendix A), and the student aggregated program level regarding enrolled and successful FTE students (which in HEMIS is available from the CRED file (see Appendix C)), has greatly increased the value of this model. The facts concerning student FTE's (SE_HM_Enrolled_FTE, SE_HM_Success_FTE), qualified (SE_HM_Qual_Resch_Count) and non-qualified (SE_HM_Qual_No_Resch_Count) research head counts, as depicted in Figure 5-7, are incorporated into this model by deriving this information from HEMIS.

	Column Name	Data Type	Length
🔑	Student_Pkey	varchar	11
🔑	Plan_pKey	varchar	10
🔑	Semester_pKey	char	4
🔑	Entrance_Category_pKey	varchar	4
🔑	Location_pKey	varchar	10
🔑	Enrollment_Flags_pKey	numeric	9
🔑	Age_pKey	numeric	9
	SE_HM_Headcount	numeric	9
	SE_Qual_Headcount	numeric	9
▶	SE_HM_Qual_Resch_Count	numeric	9
	SE_HM_Qual_No_Resch_Count	numeric	9
	SE_HM_Enrolled_FTE	float	8
	SE_HM_Success_FTE	float	8
	SE_Dropped	numeric	9
	SE_Cancelled	numeric	9
	SE_Proceed	numeric	9
	SE_Enrollment_Date	datetime	8
	SE_Grading_Basis_Date	datetime	8
	SE_Changed_With_Plan	varchar	10
	SE_Extract_Datetime	datetime	8

Figure 5-7: Enrolment fact table

Before the SDM, it was very difficult to determine the student's primary academic plans, identify students who qualify for government subsidies, etc. Now, there is no need to write complex queries in order to determine these statistics, because the enrolment star has made this task straightforward. In Figure 5-8 it is demonstrated that, using the STUDENT_ENROLMENT_FLAGS dimension, it is just a matter of

dragging and dropping items into the pivot table so as to generate important statistics for HEMIS (subsidy students) and non HEMIS students.

Count of					Acader			
Plan	Enroll	HEMIS	Locatic	Enti	2003	2004	2005	2006
Primary Plan	Enrolled	Subsidy student	Distance	FU	91	257	210	88
			Distance Total		91	257	210	88
			Main	FU	2,840	3,155	3,310	3,319
			Main Total		2,840	3,155	3,310	3,319
			QwaQwa	FU	267	327	454	474
			QwaQwa Total		267	327	454	474
			Vista	FU	220	59	14	13
			Vista Total		220	59	14	13
		Subsidy student Total			3,418	3,798	3,988	3,894
	Enrolled Total				3,418	3,798	3,988	3,894
Primary Plan Total					3,418	3,798	3,988	3,894

Figure 5-8: Pivot table from student enrolment star

5.6. Course registration model

This is the deepest and densest model, with eleven participating dimensions in this SDM, as shown in Figure 5-9. This model is useful to answer institutional and HEMIS queries concerning student course pass or throughput rates, FTE's, the input subsidy and a number of other facts as shown in Figure 5-10. The grain of the STUDENT_COURSES_FACT table is per student, per program, per semester and per course along with other participating dimensions. This model is based on six conformed and five other dimensions that are specific to this model. The five dimensions specific to this model are described below:

- DEPARTMENT DIMENSION: This dimension is very useful in breaking down input and output subsidies at department level.

- COURSE DIMENSION: Course level descriptions are available from this dimension. This is a static dimension and changes therefore mostly occur at the beginning of the year in terms of adding new courses.

- COURSE OFFER DIMENSION: This dimension holds detailed level information about the date of a course's start, its end date, course education level, attendance type, HEMIS funding status and other important data.

- STUDENT COURSE FLAGS DIMENSION: This junk dimension is based on flags and indicators to answer queries regarding the grade category such as enrolled, dropped, and withdrawn status along with other important parameters.

- FUNDING GROUP CESM DIMENSION: CESM stands for classification of educational subject matters and this dimension provides information on CESM categories, CESM levels, etc.

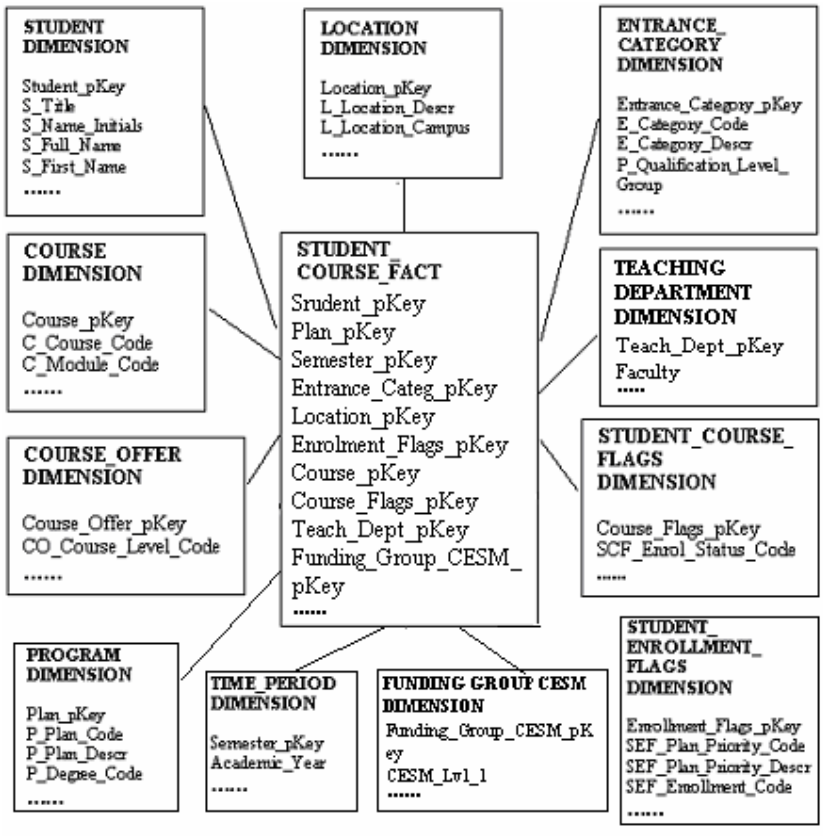

Figure 5-9: Course registration star

In addition to the number of facts that were derived from OLTP systems, certain facts were derived and calculated using HEMIS definitions. The facts SC_HM_Teach_Enroll_FTE and SC_HM_Teach_Credit_FTE are course credit values, calculated using the formulas as described in Figure 4-1 and 4-2. Similarly, the facts SC_HM_Teach_Input_Subsidy, that provides subsidy information against enrolled FTE students, is derived using funding groups weighted as shown in Table 4-

1. The definitions for calculating SC_HM_Teach_Weighted_FTE and SC_HM_Teach_Input_Unit are also taken from HEMIS.

SC_Enrolled	numeric	9
SC_Dropped	numeric	9
SC_Cancelled	numeric	9
SC_Course_Grade	varchar	3
SC_Course_Marks	int	4
SC_Completed_Research_Value	real	4
SC_HM_Teach_Enroll_FTE	float	8
SC_HM_Teach_Credit_FTE	float	8
SC_HM_Teach_Weighted_FTE	float	8
SC_HM_Teach_Input_Unit	float	8
SC_HM_Teach_Input_Subsidy	float	8
SC_Status_Date	datetime	8
Changed_With_Plan	varchar	10
SC_Enrollment_date	datetime	8
SC_Dropped_Date	datetime	8
SC_Grading_Basis_Date	datetime	8
SC_Grade_Date	datetime	8

Figure 5-10: Student course fact table

The trends evident in institutional queries are focused on course marks, grades, number of students enrolled, dropped, and cancelled and enrolment and course outcome dates. However, HEMIS queries concern teach enrolled FTE, teach credit FTE, teach weighted FTE, teach input unit, and teach input subsidy. Previously, complex queries were used to extract FTE and subsidy information by relating all of the six HEMIS tables (STUD, CREG, CRED, QUAL CREM, and CRSE (see Table 4-6)). Using the above course registration star model, all of the required institutional and HEMIS information is made available using one single fact table.

From Figure 5-11 it is evident that, using the course registration star model, it is just a matter of dragging and dropping items on the pivot table to answer course level queries like FTE's by splitting subsidy and non-subsidy students. This was previously a tedious task.

	2006 Sub 3 (FINALIZED)				
Doe Group	Teach Enroll Fte	Teach Credit Fte	Teach Weighted Fte	Teach Input Unit	Teach Input Subsidy
All other Humanities majors	6,250	5,147	8,073	11,859	97,502,236
H04 Business Commerce & Management Sciences	2,767	1,897	3,491	5,237	43,056,541
H07 Education	1,079	915	1,526	1,526	12,545,224
Science Engineering & Technology	8,190	6,572	12,232	37,468	308,062,255

Figure 5-11: Pivot table from student course registration star

5.7. Admission applications model

This model is capable of answering institutional queries with respect to admission applications data that is never reported to the DoE and neither is available from HEMIS. The model contains seven participating dimensions, with four conformed dimensions and three dimensions specific to the model as shown in Figure 5-12. The grain of the STUDENT_APPLICATION_FACT table is per student, per academic plan, per semester. New dimensions in this model are:

- STUDENT LAST SCHOOL DIMENSION: This dimension provides information on the last school attended by the student.
- STUDENT LAST UNIVERSITY DIMENSION: For both entering and non-entering students this dimension provides information on the last university attended by the student.
- STUDENT APPLICATION FLAGS DIMENSION: This junk dimension holds information on application flags and indicators such as the status of financial aid applications, residence applications, application evaluations, applications completed, cancelled, etc.

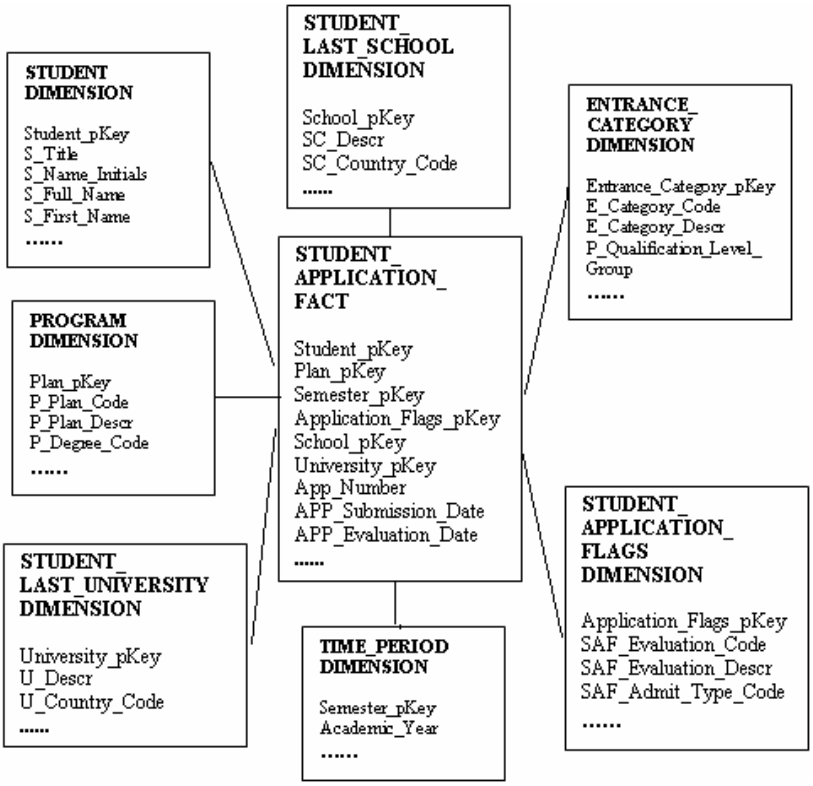

Figure 5-12: Admission applications star

The fact table as depicted in Figure 5-13 stores admission application numbers and application dates, along with other facts.

	Column Name	Data Type	Length
🔑	Student_pKey	varchar	11
🔑	Plan_pKey	varchar	10
🔑	Semester_PKey	char	4
🔑	Application_Flags_pKey	numeric	9
🔑	School_pKey	varchar	11
▶🔑	University_pKey	varchar	11
	APP_Number	numeric	9
	APP_Application_Submission_Date	datetime	8
	APP_Evaluation_Date	datetime	8
	APP_School_Degree_Descr	varchar	30
	APP_School_Graduation_Date	datetime	8
	APP_University_Degree_Descr	varchar	30
	APP_University_Graduation_Date	datetime	8
	APP_Active_Status_Date	datetime	8
	APP_Admit_Status_Date	datetime	8
	APP_Application_Status_Date	datetime	8
	APP_Cancelled_Status_Date	datetime	8
	APP_Completed_Status_Date	datetime	8
	APP_Extract_Datetime	datetime	8

Figure 5-13: Application fact table

From the above admission application star it is really a matter of dragging and dropping items on the pivot table in order to draw important statistics as shown in Figure 5-14.

Count of Stude	Acader						
Cancelled_S	1999	2000	2001	2002	2003	2004	2005
DATA				1			
DENY						3	1,766
NOT CANC	1,908	2,544	3,883	5,100	4,791	5,660	16,977
WADM	2,253	2,981	4,140	4,837	5,109	4,135	52
Grand Total	4,161	5,525	8,023	9,938	9,900	9,798	18,795

Figure 5-14: Pivot table from admission application star

5.8. Admission applications snapshot model

This model is an excellent example of information that was neither possible to obtain or too difficult to extract from the OLTP system. Information is required to track the admission application history as per snapshot, as illustrated in Table 5-2. This snapshot is very helpful in tracking application trends from previous years at a specific snapshot date. The student application snapshot star (see Figure 5-15) is designed to answer such queries.

Table 5-2: Snapshot dimension

Snapshot_pKey	Snapshot_Date	Snapshot_Desc
1	07-APR	7 APR
2	07-MAY	07 MAY
3	07-JUN	07 JUN
4	07-JUL	07 JUL
5	07-AUG	07 AUG
6	07-SEP	07 SEP
7	07-OCT	07 OCT
8	07-NOV	07 NOV
9	07-DEC	07 DEC
10	14-JAN	14 JAN
11	21-JAN	21 JAN
12	28-JAN	28 JAN
13	28-FEB	END FEB
14	31-MAR	END MAR

The grain of the STUDENT_APPLICATION_SNAPSHOT_FACT is per semester, per program, per snapshot key, along with the admission category dimension. The model consists of five dimensions: three of them are conformed and two of them are specific to this model, as explained below:

- SNAPSHOT DIMENSIONS: The full academic year regarding admissions was divided into 14 different snapshots as recorded in Table 5-2. The academic calendar in this regard starts from April and ends in March the following year. The calculations are based on cumulative totals and "END MAR" indicates the total applications received in the admission calendar.

- ADMISSION CATEGORY: Provides information on calendar type such as UG-Jan (January Intake), PG-SEP (September intake), etc.

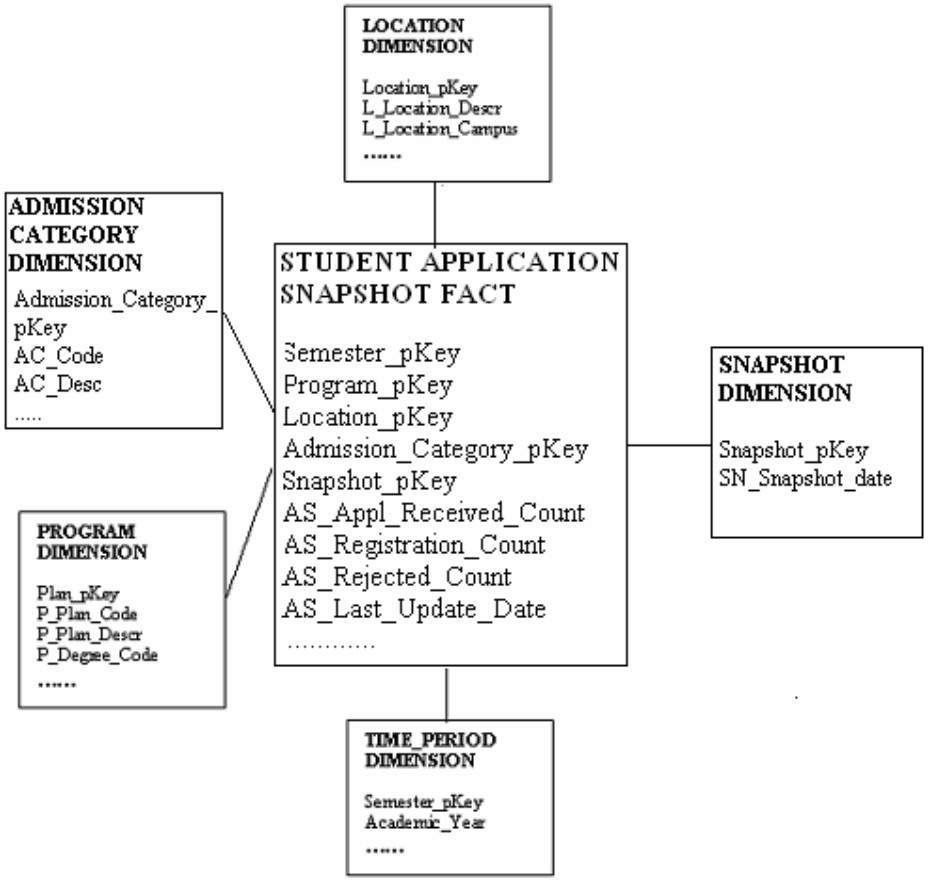

Figure 5-15: Admission applications snapshot star

The pivot table in Table 5-3, drawn from the application snapshot star, lists cumulative totals per snapshot for the 2004 and 2005 academic year applications. This information is very useful in identifying trends in order to compare counts of applications per snapshot.

Table 5-3: Pivot table for admission application snapshot facts

		Snapshot Description											
Academic Year	**Facts**	**7-Apr**	**7-May**	**7-Jun**	**7-Jul**	**7-Aug**	**7-Sep**	**7-Oct**	**7-Nov**	**7-Dec**	**28-Jan**	**FEB END**	**MAR END**
2004	Application Received	404	1,486	6,033	18,294	22,930	31,646	48,408	52,338	54,688	58,399	59,689	59,874
	Rejected	212	786	3415	11881	14731	20572	32874	34657	35701	36847	36976	36989
2005	Application Received	447	1,700	6,752	17,487	23,688	42,383	47,159	50,360	52,774	56,801	58,075	58,329
	Rejected	208	833	3508	10441	13980	26486	29241	30680	31717	33071	33211	33239

5.9. Undergraduate longitudinal studies model

Another example of star models for undergraduate longitudinal studies of first time entering students is shown in Figure 5-16.

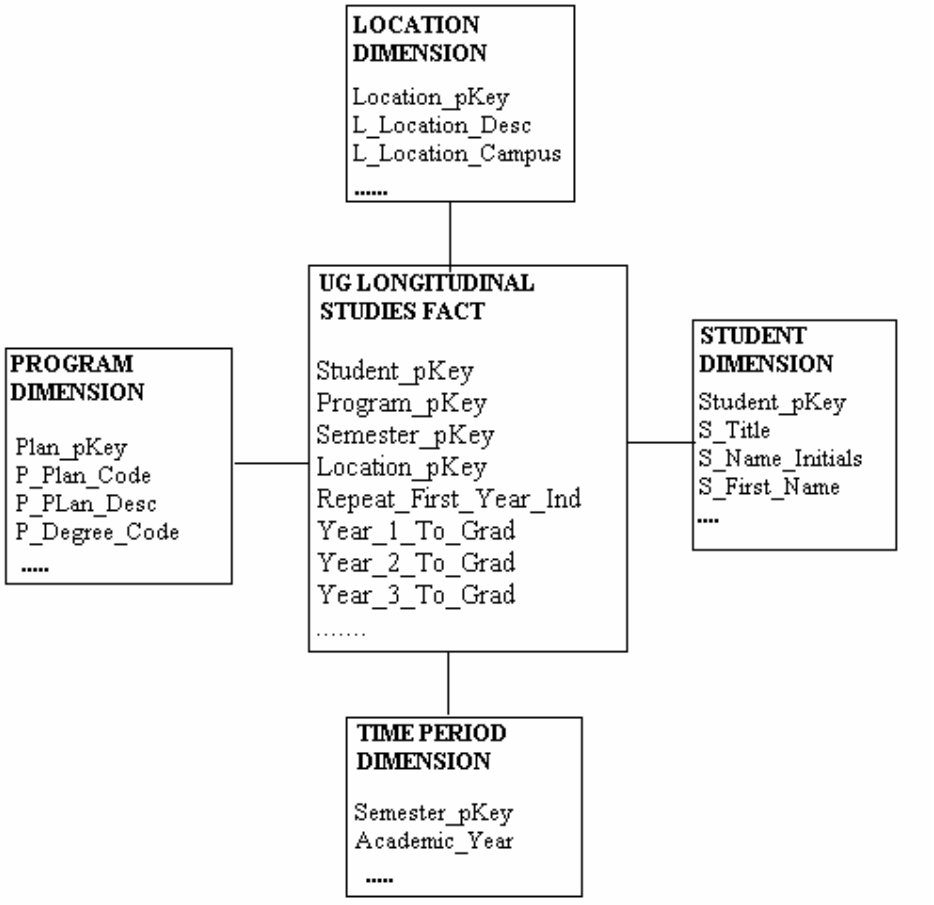

Figure 5-16: UG longitudinal studies star

This model is helpful in tracking student academic progress, which was very complex to extract from the OLTP system. The model is based on four conformed dimensions.

The grain of the fact table is per student, per program, per semester and per location. The fact table is very wide, containing 59 facts along with ten calculated facts.

The pivot table of Table 5-4, drawn from the UG longitudinal studies star, provides information on students' academic progress. In the table it is indicated that there were 4269 students registered in the year 2000, out of which 2249 graduated with an average number of 4.16 years registered, while 35 are still registered. 481 students had moved to other faculties or have registered for other programs. Similarly, other very useful information is available on dropouts, those excluded, etc.

Table 5-4: Pivot table for UG longitudinal students facts

	Academic Year					
	2000	**2001**	**2002**	**2003**	**2004**	**2005**
Total Regn	4269	5336	5585	6081	6257	4292
Total Grad	2249	2718	2738	2739	2219	963
Total Still Reg	35	90	187	482	1258	1906
Total Moved	481	597	549	514	386	376
Total Excl Aftr Passing	243	375	565	778	1336	525
Total Drop Aftr Passing	1261	1556	1546	1568	1058	522
Total Drop Aftr Reg Yrs	2137	2758	2650	2555	1894	832
Average Year to Graduate	4.16	4.01	3.86	3.66	3.45	2.73
% Graduate Current	52.68	50.94	49.02	45.04	35.46	22.44
% Dropout Only	29.54	29.16	27.68	25.79	16.91	12.16
% Excluded Only	5.69	7.03	10.12	12.79	21.35	12.23
% Moved Faculty	11.27	11.19	9.83	8.45	6.17	8.76
% Still Registered	0.82	1.69	3.35	7.93	20.11	44.41

5.10. Output subsidy model

This model provides HEMIS information on output subsidies and units generated by qualified students as shown in Figure 5-17. The model is based on two conformed dimensions and one teaching department dimension. The grain of the fact table is per student, per program, per semester, per location and per teaching department. The facts of the OUTPUT_SUBSIDY_FACTS store actual facts regarding units and output subsidies.

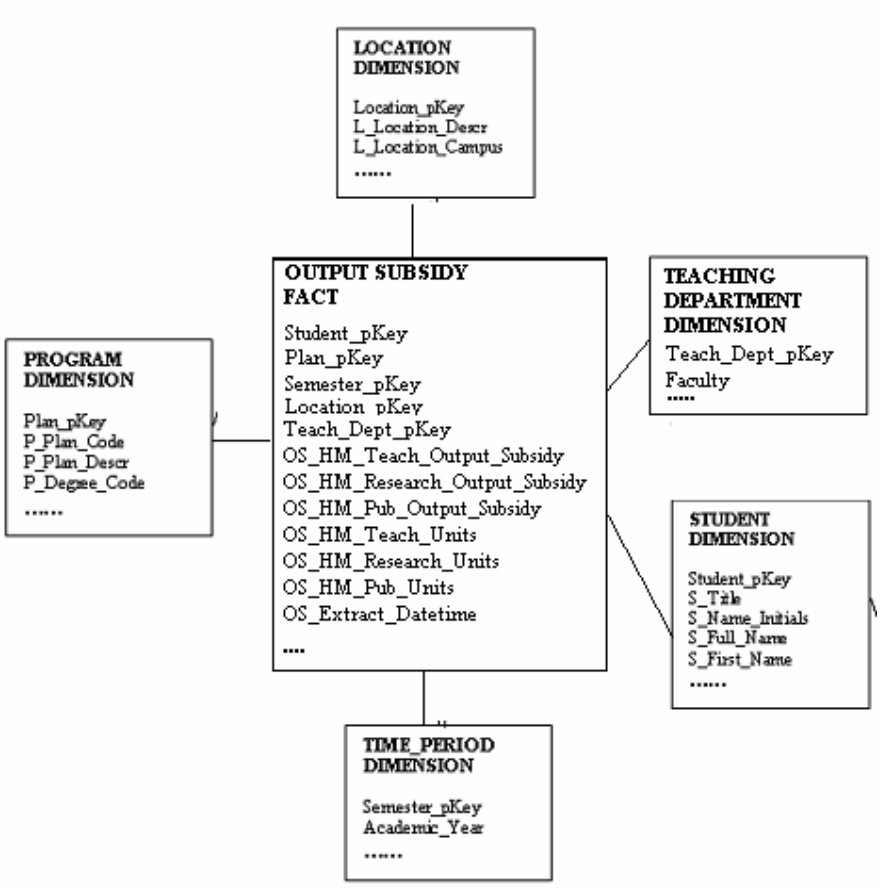

Figure 5-17: Output subsidy star

It is very easy to extract subsidy information from the SDM using the output subsidy model. The OUTPUT_SUBSIDY_FACT is loaded with all the calculations that are required to generate output subsidies and units information. It is simply a matter of dragging and dropping items on the pivot table to obtain the statistics shown in Table 5-5.

Table 5-5: Pivot table for output subsidy

	2002	2003	2004	2005	2006
Total Output Subsidy	124,090,245	127,569,441	133,338,017	153,568,388	181,705,089
Teach Output Subsidy	38,305,437	40,001,406	46,726,465	53,755,680	57,367,946
Research Output Subsidy	38,581,091	34,829,828	40,794,986	52,614,215	54,382,955
Publication Output Subsidy	47,203,717	52,738,206	45,816,567	47,198,492	69,954,188
Teach Input Unit	52,054	56,273	56,566	53,657	56,089
Teach Units	3,318	3,391	3,830	4,221	4,086
Research Units	542	465	562	619	615
Publication Units	663	704	631	555	791

5.11. Summary

In this chapter the TF for HEMIS and the query centric student data mart using dimensional modelling were presented. Information was provided regarding how to fit student academic records into star models. Discussions were also undertaken of the fact table grain, summarisation, naming conventions of tables and columns, and conformed and junk dimensions of the SDM. The next chapter is dedicated to the ETL process of the SDM. Several examples are cited where complex procedures, functions and mappings were written for data cleansing and loading into the SDM.

Chapter 6

Extraction transformation and loading issues in the student data mart

6.1. Introduction

In the previous chapter star models of the SDM were presented. Details were provided as regards to fitting institutional and HEMIS data on one platform using dimensional modelling. Differences in the grain and summarisation of the SDM as compared to other commercial dimensional designs were also discussed. The value of the SDM was highlighted by presenting models such as application snapshots, longitudinal studies and output subsidy that were not presented in the HEMIS system. Once the design of the DW system is finalised, the next major step is the ETL. In this chapter the ETL processes and issues while loading data into the SDM will be discussed. This is the fifth activity, "Implementation", in the AR cycle. Numerous examples of the efforts to carry out the data cleansing and transformation are furnished in this chapter. This must be done in order to convert the OLTP and HEMIS data into useful information.

6.2. The ETL process

The ETL process was broken down into two major phases of data staging and loading into the SDM. Microsoft data transformation services (DTS) were utilised owing to the tool's availability for writing DTS packages. In the DTS packages data cleansing, merging, and sorting were carried out before loading data into the SDM. To draw incremental loads from the OLTP systems, DTS packages were scheduled to take place daily, weekly and monthly.

6.2.1. Incremental loads

For incremental loads, a table STAGING_BATCH_ID is maintained by means of the Last_Update_Date, Last_Extract_Date and Batch_Id frequency as shown in the Table 6-1. The load is divided into three different frequencies so as to manage the load from the daily job executions queue. Daily loads are based on student data updated on the OLTP side on a daily basis. Typically such student data are Admissions,

Registrations, Fees, Financial Aid, etc. Weekly loads comprise the unit outcomes, research supervisor, etc., while the monthly loads are based on data that changes infrequently such as academic programmes, unit details, organisational structures, etc. The Batch_Id which is used to track the loads, has a numeric value and is assigned to each new or updated row in the SDM. For example, for daily loads, the Batch_Id 234 will be assigned to all rows extracted on 19-02-2007 from the source systems. At the end of the loading process, the Daily Batch_Id is incremented to the next number and the Last_Update_Date is updated with the current system date and the Last_Extract_Date is overwritten with the Last_Update_Date.

Table 6-1: STAGING_BATCH_ID table

Batch_Id	Last_Update_Date	Last_Extract_Date	Load_Frequency
234	20-02-2007	19-02-2007	Daily
134	18-02-2007	11-02-2007	Weekly
90	03-02-2007	07-01-2007	Monthly

6.2.2. Staging tables

Staging tables were used to bring the incremental loads into the SDM based on the Load_Frequency and Last_Extract_Date, as explained in the previous section. As clarified in chapter 3, OLTP systems contain data contaminated with numerous data errors and redundancy in the tables and columns that result in inconsistent data. In order to provide a meaningful load to the SDM database, views were written on the OLTP side by joining related tables and by enforcing business constraints for bringing in data that qualify for the SDM. Views were restricted to bringing in the incremental load by comparing the Last_Update_Stamp of the underlying tables that must be greater than equal to the Last_Extract_Date of the STAGING_BATCH_ID table.

6.2.3. Staging layer history tables

A history table is maintained against each SDM table in order to keep track of the changes made in the table. Newly inserted or updated rows in the SDM are recorded with the processed “P” flag while the rejected records that occur due to business rule violations are recorded with the “E” flag. The business rule violation can occur owing to the result of incoming “NULL” values for mandatory columns, multiple values for target tables, lookup failures and data not satisfying conditions and checks.

6.3. ETL for the student data mart

Once the data is brought into the staging table the next stage is loading the SDM layer. The DTS packages between the staging and SDM layers perform the data cleansing, sorting, and merging in order to separate the dimensional attributes and facts. A number of programs, written for data transformation, will be explained below. In this chapter the DTS packages for only a few of the models are discussed since it was not possible to provide information on the ETL process for each of the models presented in chapter 5.

6.4. The student dimension

The student dimension is one of the widest dimensions in this SDM with fifty nine attributes. Fifteen OLTP tables, as shown in Figure 5-5, were involved in extracting data for this dimension. The complexity involved in loading the student dimension from the DTS package is shown in Figure 6-1. A number of Execute SQL tasks, and Data Driven Tasks were written for data cleansing and integrating data into one dimension as explained in the following sub-sections.

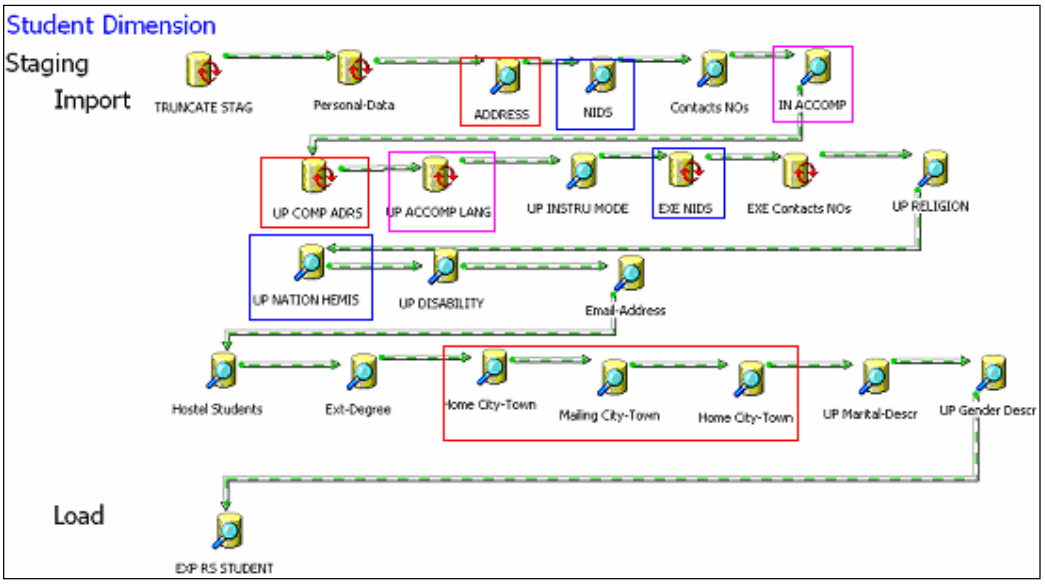

Figure 6-1: ETL for student dimension

6.4.1. Address

The student addresses at the OLTP side were problematic because data entry operators were storing address information erroneously in various columns (such as Address1, Address2, Address3, etc.) as shown in Table 6-2.

Table 6-2: Student Addresses table

Emplid	Effective_date	Address1	Address2	Address3	Address4	Address5
2004027864	1/20/2004		BFN	9501	19 Liesgang Street	Brandwag
2004027864	1/5/2005	312 Wynond Mouton	9501		Universitas	
2004027864	10/2/2006	18 Hoopwood Street			Bloemfontein	9500

Extracting the correct address information posed a major challenge. The stages involved in correcting student address are indicated with red boxes as shown in Figure 6-1. In stage 1 address information is brought into the staging table. A number of Microsoft SQL Server fuzzy logic and fuzzy grouping were used to clean this data. In stage 2 a procedure, P_Student_Complete_Address (see Appendix M), was written to clean such corrupted data in the staging tables before loading it into the SDM. This was done by splitting address information into the corresponding columns of street, city, province, and postal code. In cases where city and province were not captured or captured with the wrong spelling, a lookup table is maintained to extract city and province data information according to the postal code captured with the student address information as shown in stage 3.

6.4.2. Primary nationality and native language

The OLTP table, PS_PERS_NID table, holds student nationality information and a flag, "Y", should be set to indicate the primary nationality. In a number of cases this flag was set for all of the nationalities related to a student as shown in Table 6-3, thus it becomes difficult to establish the primary nationality in the absence of the Last_Update_Date in this table, that is, it is difficult to track the latest record.

Table 6-3: OLTP nationality table

Student Number	Nationality	Primary
2004027864	PAK	Y
2004027864	SAF	Y

DoE has divided the possible nationalities into the SADC and non-SADC groups of countries. This classification does not exist at the OLTP side. Similarly, the OLTP nationality codes do not match the DoE nationality code definitions. The stages used in cleaning student nationality data are indicated with blue boxes as shown in Figure 6-1. In stage 1 nationality data is brought into the staging table. In order to rectify the student nationality data issues a program, P_Student_Nationality (see Appendix N), was written and executes at stage 2. A mapping file was created to map the OLTP nationalities onto the DoE nationalities by dividing these into the SADC and non-

SADC groups of countries as indicated in stage 3. This program rectifies all the issues relating to the nationality data while loading data into the STUDENT dimension table.

Similar cases where the primary flag was not set accordingly were found regarding the student's native language as indicated in Table 6-4. Two stages as marked with pink boxes were used to rectify student language data. In stage 1 data is brought into the staging table.

Table 6-4: Student native language

Student Number	Language	Native Language
2004027864	URDU	Y
2004027864	URDU	Y
2004027864	ENGLISH	N

To correct these data issues with regards to the native language, a procedure, P_Student_Languages, was written (see Appendix O) that executes in stage 2 to update the defected record. Lookups on student nationality and address information were used to identify the native language. A few scenarios in the history data occurred where it was impossible to accurately identify the native language record. In such cases, the native language was selected randomly.

6.4.3. Typographical errors

Most of the transactional data entry screens include a "drop down" list of values to restrict users to selecting a value from such a predefined list thus preventing users from typing different versions of the same value. Unfortunately, the users can avoid using the list. One of the best examples of such errors is the city names that were incorrectly typed as shown in Table 3-7. Such typographic errors commonly occur in the student and staff demographic data, thus making reporting and obtaining a unified representation of the data difficult. In the ETL process such problems were resolved by creating lookup tables containing the correct information.

6.5. Programme dimension

The data of the academic plans and academic programmes is another example where numerous irregularities were found in the OLTP system. The DTS package as shown

in Figure 6-2 was written with a number of transformations and PL/SQL programs for repairing the data pertaining to the academic plans and academic programmes.

6.5.1. Academic plans with more than one academic programme

One of the major issues encountered in the OLTP system was querying the PS_ACAD_PLAN_TBL. It was explained in chapter 3 that in the UFS a degree can be broken down into one or more academic plans. Each year academic plans are revised and the possibilities exist for an academic plan to be moved from one area or faculty to another. Such changes are captured in the OLTP system by adding a new row in the PS_ACAD_PLAN_TBL table with a new effective date as

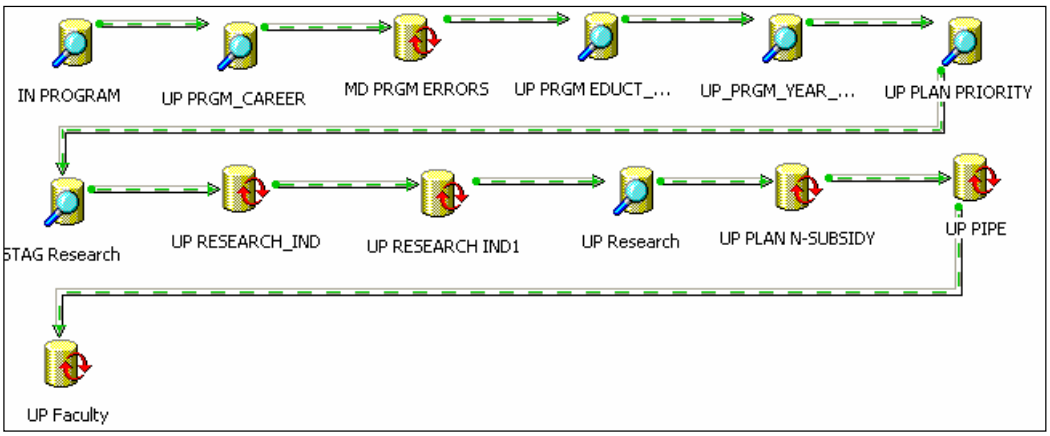

Figure 6-2: ETL for the program dimension

shown in Table 6-5 for academic plan 0081. The academic plan 0081 comprises three rows with three effective dates and academic programme combinations. This setup causes the querying of the PS_ACAD_PLAN_TBL to be complex. To find the latest academic programme to which the academic plan belongs, the query needs to use the maximum effective date as in the case of the academic plan 0081 which contains three effective dates. The logic works well in querying the latest data, but this mechanism becomes more complex when querying a previous year's data. The query needs to include certain strategies such as searching the academic plan and program using the effective date ranges with the maximum effective date.

This issue was resolved in the SDM by assigning a unique surrogate key to each combination of an academic plan and academic programme with the maximum effective date as shown in Table 6-8. Fact tables using the PROGRAM dimension

were loaded using a lookup on P_Plan_Code and P_Program_Code on the PROGRAM dimension for accessing the correct surrogate key.

Table 6-5: PS_ACAD_PLAN_TBL

Acad_ Plan	Effdt	Eff_Status	UFS_Dgr_Descriptn	Acad_Prog	Degree
0081	1901/01/01	A	Elective Student	M0000	0081
0081	2004/01/02	A	Elective Student	H8000	0081
0081	2005/01/02	A	Elective Student	H8000	0081
1506	1901/01/01	A	Baccalaureus Artium Honores (Corporate Communication)	M1030	1506
1506	1902/01/01	A	Baccalaureus Artium Honores (Corporate Communication)	M1031	1506

6.5.2. Academic career and education level of academic plans

In Table 6-6 it can be seen that the degree level and academic career were captured incorrectly and this might be due to typographic errors or staff negligence. For example, the degree code 8531 represents an Honours degree, and according to the DoE definitions as listed in Table 6-7, the degree level 6 represents post graduate degrees. Therefore the academic career should be “HSPG” instead of “HSUG”.

Table 6-6: Mismatch of academic careers with degree level

Degree Code	Description	Degree Level	Academic Organisation	Academic Career
2002	UNIVERSITY CERTIFICATE IN PHARMACOLOGY FOR PRIMARY TIALTH CARE	1	H8151	HSPG
8531	Honours Bachelor’s in Medical Sciences: Anatomical Pathology	6	H8070	HSUG
8532	Honours Bachelor’s in Medical Sciences: Bio-Engineering	6	H8070	HSUG

Such inconsistencies in the data were corrected by drawing the correct definition of the qualifications for the academic career, degree level, plan priority, research and non- research academic plans, subsidy and non-subsidy academic plans from the DoE VALPAC2 system which was loaded in the PROGRAM dimension.

Table 6-7: DoE definitions of education levels

Level Code	Description	Academic Career	Academic Career Code
11	Undergraduate Diploma or Certificate < 2 Years	Undergraduate	UG
01	Undergraduate Diploma or Certificate	Undergraduate	UG
02	General Academic First Bachelor's Degree	Undergraduate	UG
03	Professional First Bachelor's Degree	Undergraduate	UG
04	Post -graduate Diploma or Certificate	Postgraduate	PG
05	Post-graduate Bachelor's Degree	Postgraduate	PG
06	Honours Degree	Postgraduate	PG
07	Masters Degree	Postgraduate	PG
08	Masters with Research Work	Postgraduate	PG
09	Doctorate	Postgraduate	PG
ZZ	Program undertaken by Occasional Student	Occasional	Occ

Table 6-8: Program dimension

Program_Pkey	P_Plan_Code	P_Effective_Date	P_Effective_Status	P_Program_Code	P_Degree_Code	P_Degree_Desc
13	0081	1901/01/01	A	M0000	0081	Elective Student
14	0081	2004/01/02	A	H8000	0081	Elective Student
15	1506	1901/01/01	A	M1030	1506	Baccalaureus Artium Honores (Corporate Communication)
16	1506	1902/01/01	A	M1031	1506	Baccalaureus Artium Honores (Corporate Communication)

6.6. Student courses and end-term registrations

A number of efforts were invested for the purpose of repairing the data for enrolments and course registrations. Four different stages were developed to clean and process the data. Two staging tables, STAGING_TABLE_COURSES (see Appendix G) and STAGING_TABLE_ENROLMENT (see Appendix H), were created in the staging area to process student registrations. STAGING_TABLE_COURSES contains course registration details while STAGING_TABLE_ENROLMENT was designed to cater for term records drawn from the student course registration details. In the following sections enrolment stagings are discussed in detail.

6.6.1. Dropped academic plans

In the enrolment ETL package 1, as shown in Figure 6-3, incremental loads from course registrations are brought into the STAGING_TABLE_COURSES. It was explained in section 3.6.13 that "NULL" values were found in the Enrl_Actn_Rsn_Last column of the PS_STDNT_ENRL_TBL table where course registration details are captured.

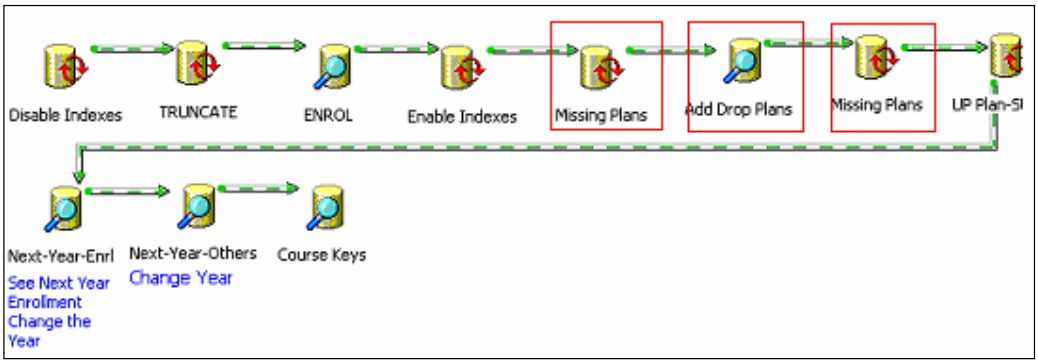

Figure 6-3: Enrolment ETL package 1

The "NULL" academic plans are updated at this stage by drawing the deleted records from the OLTP audit table for the PS_STDNT_ENRL_TBL_A. The SQL script written in the Data Driven Task "Add Drop Plans" updates the rows in the staging table by matching student records to the unique Class_Id that exists in both the PS_STDNT_ENRL_TBL and PS_STDNT_ENRL_TBL_A tables.

6.6.2. Next year's enrolment record

University departments are interested in tracking the activities of students throughout their academic life. For example, some of the questions raised are: When did the student first register? How many graduated? How many changed their academic programme? How many left the University voluntarily? How many have no record for enrolments for the following year? Such information was processed in the enrolment ETL package 1. For example, on loading the 2004 student enrolments, the program searches the student graduation records, while the student "next year enrolment" records if his/her degree has not been completed and also establishes whether a gap exists in the academic records. The flags as shown in the last three SQL tasks in Figure 6-3 are also set.

6.6.3. Changed academic plans

Each year the DoE provides all institutions with a revised list of academic programmes that are valid for the following year. A problem arises when a student was originally enrolled in one academic plan and received a degree in another academic plan due to an alteration in the qualification as requested by the DoE. An example of such a change is provided in Table 6-9. Owing to such major changes, it is difficult for the TI administration to track down the original plan in which the student had enrolled. The only remaining option is to manually view the records of the student in order to link newly changed plans. For example, in the following table, the student was enrolled for the academic plan "0030" and graduated in the plan "0031".

Table 6-9: Student changed plan

Student_ Number	Enrolled_Plan	Graduation_Plan
2004027864	0030	0031

This issue was resolved in the enrolment ETL package 2 in a set of tasks, as shown in Figure 6-4, by considering the course registration details and the plan's minimum length of study. On the OLTP side this information, which was the original academic plan for which the student had been registered, can be extracted because the student has an enrolment record for three years and received pass marks in the courses for which s/he was registered for. The same student has a graduation record showing another plan, with the enrolment and graduation dates but with no entries of the academic plan in the enrolment tables. By comparing the enrolment dates in both tables the Data Query Driven Task "Add Drop Plans" in stage 2 as indicated in Figure 6-3 with red box draws the previous plan into the old academic plan column and replaces the old academic plan with the academic plan in which the student graduated. The Execute SQL task "Missing Plans" in stage 3 indicated with red box in Figure 6-3 also sets a flag for changed academic plans so as to furnish the statistics of the students with changed academic plans.

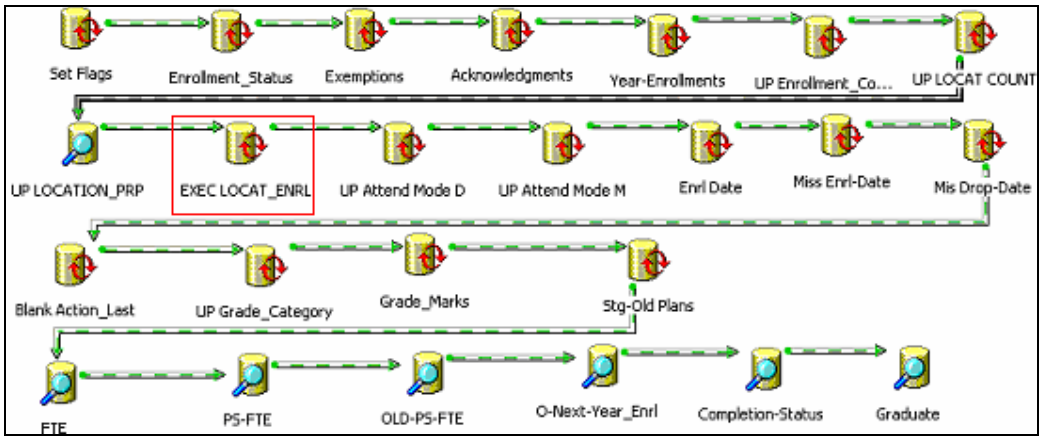

Figure 6-4: Enrolment ETL package 2

6.6.4. Academic programme primary location indicator

A student registered for an academic plan can study courses at other UFS campuses as shown in Table 6-10. For the student headcount record a student can only be counted once for that academic year based on his/her primary location.

Table 6-10: Student course location

Student Number	Course Code	Campus	Priority	Record Count
2004027864	FF1021	BFN	31	2
2004027864	FF1024	BFN	31	2
2004027864	FF1022	Qwaqwa	3	1
2004027864	FF1023	Welkom	6	1

The primary location of a student is calculated using a PL/SQL program, P_Primary_Location (see Appendix P) in the enrolment ETL package 2 (see Figure 6-4) indicated with red box in order to establish the student's primary location while loading data into the SDW table. The logic behind identifying the primary location is based on the number of courses taken and the priority level of the campuses as indicated in Table 6-11. In the first phase, the program assigns a priority level to each course according to the campus priority. In the second phase, the program calculates and assigns the record count of the number of courses taken per campus. From table 6-10 it can be seen that the student has taken two courses from BFN and one course from the QwaQwa and Welkom campuses. Based on the business logic, the number of courses taken from BFN is greater than the number of courses taken from QwaQwa and Welkom, therefore, BFN was calculated as the primary location of the student.

6.6.5. Missing enrolled dates

Enrolment dates play an important part in the student academic record. If no enrolment date exists, it is not possible to identify when the student was originally registered. For a student who qualify for a government subsidy it is important to find out whether the student was registered before or after the census date. A census date represents the last date after which the student registration and cancellation is no longer eligible to be considered by the DoE. The "Miss-Enrl-Date" Execute SQL task as shown in the enrolment ETL package 2 (see Figure 6-4) was written to update the missing qualification enrolment dates by extracting the minimum course enrolment date

Table 6-11: Campus location and priority

Sub Campus Short Name	Sub Campus Full Name	Campus	Type	Priority
ALIWAL	Aliwal North	Main	Contact	33
BFN	Bloemfontein	Main	Contact	31
CFP	Centre for Financial Planning	Distance	Distance	61
KZN	Kwazulu Natal Region	Main	Contact	23
LADYBRAND	Ladybrand	Main	Contact	22
NAMIBIABUS	Nambia Business School	Distance	Distance	62
QWAQWA	Qwaqwa	QwaQwa	Contact	3
RIEPQWA	Riep Qwaqwa	Main	Contact	16
RSA	Republic of South Africa	Distance	Distance	63
THABANCHU	ThabaNchu	Main	Contact	8
TSHIYA	Tshiya	QwaQwa	Contact	4
UPINGTON	Upington	Main	Contact	7
VISTA	Vista	Vista	Contact	5
WELKOM	Welkom	Main	Contact	6
Etc …..				

6.6.6. Student FTE

To calculate the student FTE, course credit values are extracted from the VALPAC2 system. The course credit values are added to the STAGING_AREA_COURSES against each course and the FTE is calculated as shown in Figure 6-4 in the FTE task. A view was created in the VALPAC2 Access database to draw the course credit values according to the academic calendar. FTE values are also calculated for students

who had not qualified for government subsidies due to late registrations. This information is helpful to estimate the total subsidy that a department can generate and to improve its business processes in order to decrease the number of students who, due to any reason, do not participate in generating subsidies for the department.

6.6.7. Year census flag

The DoE and University management require student enrolment status reports in order to ascertain whether the student enrolled before or after the census date. A census date is set by the tertiary institution's administration according to the dates provided by the DoE.

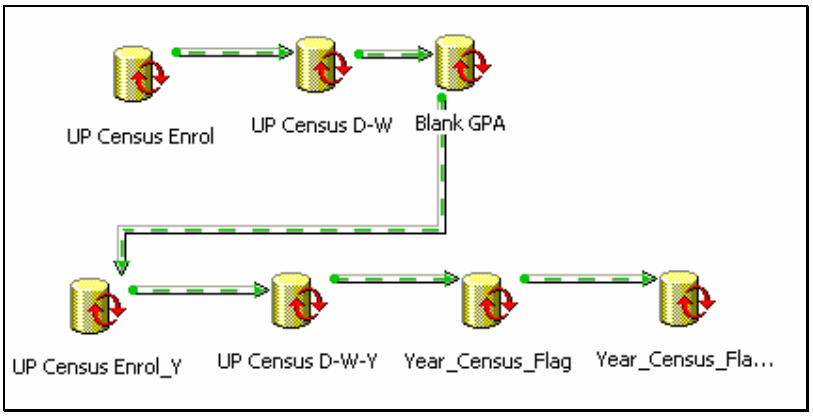

Figure 6-5: Enrolment ETL package 3

Enrolments after the census date are not considered for government subsidies. Similarly, if a student withdrew from a module after the census date, it is still considered for the DoE report. For example in Table 6-12 the enrolment status of a student is indicated as being before or after the census date. Five different Execute SQL tasks as shown in Figure 6-5 were written that derives the enrolment year status if searches for any enrolled status records during the year. In the following case, a student was registered for four modules during the year, while the census date for term 2041 was 18-01-2004 and for term 2042 it was 18-07-2004. According to the census date, the registration for module RIS615 falls within, that is, before, the census date. Due to this valid enrolment, the year status of the student was calculated as being enrolled.

Table 6-12: Enrolment status in staging area courses

Student Number	Term	Module	Enrolment Date	Term Status	Year Status
2004027864	2041	RIS614	02-03-2004	Dropped	Enrolled
2004027864	2041	RIS615	15-01-2004	Enrolled	Enrolled
2004027864	2041	RIS616	20-01-2004	Dropped	Enrolled
2004027864	2042	HUM644	06-08-2004	Enrolled	Enrolled

6.6.8. Staging table enrolment

A separate staging table, STAGING_TABLE_ENROLMENT, was created to extract the summary of the student's registration in order to populate the student enrolment star. In the enrolment ETL package 4, as indicated in Figure 6-6, a summary of the student term registration is extracted from STAGING_AREA_COURSES as shown in Table 6-13. The summary excludes student course registration details when data is imported into the STAGING_TABLE_ENROLMENT. Subsequent flags and indicators are calculated at summary level to populate the enrolment flag dimensions.

Table 6-13: Staging table enrolment

Student Number	Term	Enrolment Date	Year Status
2004027864	2041	02-03-2004	Enrolled
2004027864	2042	15-01-2004	Enrolled

6.6.9. Historical data

Data from the merged QwaQwa and Vista campuses before 2004 were also imported into the OLTP system so as to draw enrolments and other statistics based on their historical data. Due to the unreliability of the data, a flag was added with a VALID and INVALID status for the data in the enrolment ETL package 4 as shown in Figure 6-6. The MIS department, for most of the reports, had never drawn data relating to these campuses for reporting prior to 2004 due to the unreliability of that data.

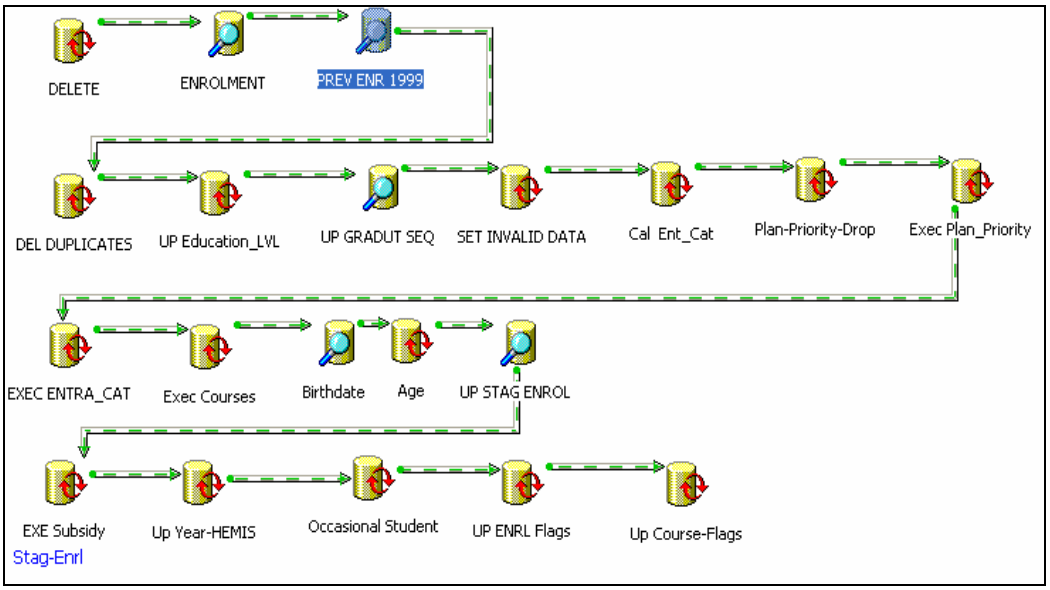

Figure 6-6: Enrolment ETL package 4

6.6.10.Primary academic plan

A student can register in more than one academic plan during an academic year as shown in Table 6-14. The DoE provides subsides to the TIs according to the student headcounts. Therefore, a student can obtain a subsidy for only one academic plan even if registered in more than one academic plan. For institutional internal reporting, however, a student must be counted for all of his / her registrations. For example, student 2004027864 was registered for two different academic programmes during the year (see Table 6-15). Before loading data into the SDM, the student's primary plan is calculated in the enrolment ETL package 4 using a PL/SQL program P_Plan_Priority (see Appendix Q). The PL/SQL program uses a lookup table for the academic career of the academic programme, as listed in Table 6-14, and assigns the priority to the student academic plan registration accordingly, as shown in Table 6-15.

Table 6-14: Academic career priority

Academic Career	Priority
PGRD	1
HSUG	2
UGRD	2

If in an academic year a student registered for more than one academic programme for the same academic career, then the program picks one of the academic plans randomly as the primary academic programme.

Table 6-15: Primary plan

Student Number	Academic Plan	Year	Plan Priority
2004027864	Masters in Computer Science	2006	1
2004027864	B Sc. Honours in Computer Science	2006	2

6.6.11.Entrance category

Important information that is very useful in developing institutional forecasting models is based on the entrance category of the student. An entrance category indicates to the DoE as well as the TI that the student entrance level to the University should be categorized according to the list in Table 6-16. This information should be captured at the time of registration. Data entry operators, however, sometimes omit this information or capture it incorrectly.

Table 6-16: Entrance category

Entrance Category	Description
F	First time entering student
T	Transferred student
E	Entering student
N	Non-entering student

Another factor that affects the reliability of this information is the fact that a number of transfers occur within the student academic plans at the beginning of the enrolment year and therefore the data stabilises only at the end of the year.

A complex PL/SQL program, P_Entrance_Category (see Appendix S), was written to derive the entrance category of the student. The PL/SQL program picks a student record, indicating the qualification type and the academic year in which s/he registered. The program checks the history record to ascertain whether the student has been previously registered in the same academic programme. If a match was found the student is identified as a non-entering student (“N”). If a student already possesses an undergraduate qualification and was registered for the same degree level again in some other academic plan, the student is identified as an entering student (“E”). In any case, if the history information was not found in the TI database against the student academic plan and qualification type, the student is marked as a first time entering student (“F”). The same logic is applied for students registered in post-

graduate degrees. The only impossibility was to decide whether the student had transferred ("T"), for which the only source available is the student her/himself. If in the registration from the previous institution is mentioned regarding the same degree for which s/he had previously registered, then s/he is considered as a transferred student ("T"). Another authentic source is the National Database of the DoE from which such a student can be identified, but this is not possible due to access rights to the National Database.

6.6.12.Subsidy student status

The DoE requires a report regarding student enrolments with the primary plan. This information is not available in the source data. Similarly a set of filters, as indicated in Figure 6-7, is used to ascertain student records that qualify for government subsidies based on:

- Enrolments before the census date.
- Barring students who had withdrawn their plans before the census date.
- Undergraduate students who fulfil their matriculation requirements.
- Barring students who were failed or have received re-assessments from the previous year.

```
/*      1. Student does not have matriculation record in Staging_Area_Extension file
        2. If student have his entries with U* he can be considered for Subsidy
        3. If student matriculation records found with Honours_Category value
                100 = No. Infor found in PS
        4. If the student have the following Grade_Categories he is also not considered
                for subsidies
                a.      00 - Not Applicable
                        00 - Not applicable appears if a student enrolled in the current year
                        and have not yet pass the module. IF a module have still 00
                        and student enrolled last year and and he is not
                        a research student. He willnot consider for subsidies.
                b.      05 - Incomplete
                c.      09 - Incomplete
                d.      20 - Discontinued
        5. If a plan is not for s subsidies the plan will not be submitted for subsidies.
        6. If student taking courses and are not taking classes they are from the
                grade category 21 and 68 and will not be considered for subsidies
        7.
        --Other Codes used for updateion
        A1 = Meet matriculation prerequisites
        A2 = Plan is not subsidised
        A3 = Discountinued/Incomplete/Not Applicable
        A4 = Sucessfully completed the module
        A5 = Enrollment After census date
        A6 = Dropped/Withdrawn Before census date
        A7 = Exemptions
        A8 = Unknown
        A9 = Missing grade category
        AA = Zero-FTE Value
        AB = Non-Primary plan
        AC = Unknown Plan
```

Figure 6-7: Subsidy student definition

In the STAGING_AREA_COURSES the student Subsidy_Status is set to values 0 or 1. This status is extracted from the DoE VALPAC2 system. All the students in the reports submitted to the DoE and present in the VAPAC2 systems are subsidy students. In enrolment ETL package 4 the data regarding student number, qualification and academic year is extracted from the VALPAC2 system and the Subsidy_Status in the staging table is set accordingly.

6.6.13.Enrolment flags dimension

A number of flags such as the primary location flag, primary plan, entrance category flag, census date, etc., were extracted in the staging area as shown in the previous sections. All of the flags and indicators are concatenated to formulate a unique combination as shown in Figure 6-8 in the Enrolment_Flags_Code column.

Enrolment_Flags_pKey	Enrolment_Flags_Code	SEF_Plan_Year_Priority	SEF_Plan_Year_Priority_Desc	SEF_Attendance_Mode	SEF_Attendance_Mode_Descr
1	1COM0	1	Primary Plan	C	Contact
2	2COM0	2	Plan Priority 2	C	Contact

Figure 6-8: Enrolment flags dimensions

When loading the enrolment flags dimension, a surrogate key is generated for each unique combination. The ETL stage for loading enrolment flags as shown in Figure 6-9 splits all these flags and indicators into separate columns for querying.

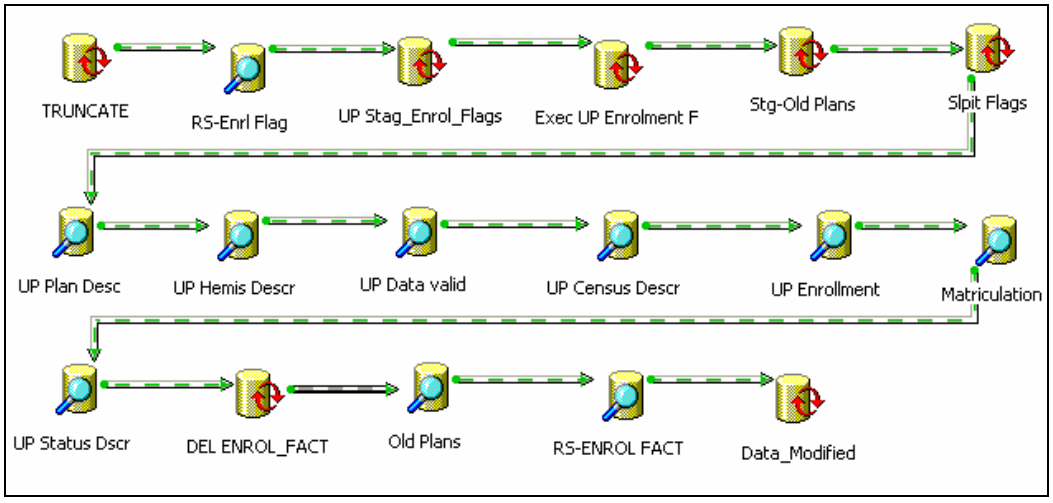

Figure 6-9: Loading of Enrolment Flags dimension

6.7. Loading of enrolment facts

After processing the four enrolment stages, the enrolment flag dimensions along with other participating dimensions of the student enrolment star (see Figure 5-6) are loaded. Finally, data is loaded into the STUDENT_ENROLMENT_FACT from the STAGING_AREA_ENROLMENT that provides a summary of the term registrations. During data loading, history tables are updated with the value, either processed, "P", or error, "E", in the Processed_Flag column. In the following sections some of the common errors that occur during the loading of the fact tables, are listed.

6.7.1. Nationality lookup failure

Nationality lookup failure errors occur due to typographical mistakes. At the front-end a drop down list from which a user can select a nationality code while entering data is available. The user can also insert the code directly so as to avoid scrolling through the list. While typing, users can easily type an incorrect entry while the system does not alert the data capturer of these errors. The ETL process can identify this error while querying the STUDENT dimension on the nationality code. If the nationality code does not exist, the whole record is inserted in the history tables with an error flag "E" and a message is then sent via email regarding the lookup failures at the end of the ETL process.

6.7.2. Duplicate race or gender

On the OLTP side, more than one race or gender can be entered against a student record or data operators can even skip this important entry. In the SDM the ETL process does not allow this type of contaminated data in the warehouse and entries are placed in the history tables with unique key violations.

6.7.3. Matric rating

Each student must have a "STAT" rating that determines the academic index. A matriculation rating is required for analysis of the student matriculation marks and grades. During the basic data entry for the student record, this information must be captured because it is a basic admission requirement. The ETL process has identified several students with regards to the 2006 registrations where this information has not been captured in the source OLTP system.

6.8. Data auditing package

In order to further verify the data loaded into the SDM, one needs to establish the confidence level of the number of rows on the OLTP side versus that on the SDM side. The data auditing package runs at the end of the ETL process and comes up with (+) or (-) counts. If the count is plus, this means that the SDM holds more records compared to the OLTP system and if the count is minus then it suggests that a few of the records have not been brought into the SDM due to certain reasons that can be checked from the history tables. This exercise has really created confidence amongst the end-users regarding the reliability of data residing in the SDM.

6.9. Summary

This chapter was dedicated to the ETL portion of the SDM. The ETL Process followed in the SDM project was explained. The setup used in bringing incremental loads from the OLTP system was discussed. Information was also furnished with regards to the staging history tables used to track data errors that were recorded for processed and error records. Several examples were provided in order to provide a clear picture of the quality of the OLTP data. The effort required to convert the HEMIS and OLTP data into useful information was also presented via the DTS packages and PL/SQL programs. Errors that mostly occur during the ETL due to the poor quality of data were also considered. In the next chapter, the comparisons between the HEMIS and the SDM regarding the similarities and differences between the two systems will be discussed.

Chapter 7

Comparisons between HEMIS and SDM systems

7.1. Introduction

In the previous chapter the ETL for the SDM was discussed in detail. Examples were provided regarding the efforts to cleanse the data and populate the star schemas. It was demonstrated that OLTP and HEMIS information can be displayed and made available from one single platform with a query centric design. It is not worth the resources required to maintain a separate third party HEMIS system that works on the subset of institutional data, especially in terms of the complexity involved in extracting the data. This chapter comprises a summary of the key similarities and differences between the HEMIS and SDM systems and represents the sixth activity, "Monitor in terms of research interests", in the AR cycle.

7.2. Similarities between the two systems

Although both systems are dissimilar in a number of aspects, some similarities were found between the two systems.

7.2.1. Separate hardware and software

The vendors of both the HEMIS and DW systems are in agreement with regards to utilising separate hardware and software to implement their solutions. The main reason for considering separate hardware is to relieve the overload of the data loading and querying burden from the OLTP systems. Because the loading and data cleansing portion consumes 80% of the resources of the server, this will definitely seriously hinder the server's performance for normal day-by-day transactions. Similarly, the query boost time cannot be enhanced for the OLTP server owing to the database configuration that is for efficient transaction processing.

7.2.2. Ongoing maintenance

Both the HEMIS and DW systems need either an in-house or permanent vendor support team to ensure the smooth running of the system. There is always a need for

assistance in adding new business constraints at the ETL layer and incorporating new fields in the reporting layer in order to satisfy incoming requests.

7.2.3. Historical data

Historical data is very useful for trend analysis and both systems are capable of retaining as much historical data as the TI would like. There is no restriction on the size of the database as in the case of the VALPAC2 system which is not capable of keeping more than two years' worth of data.

7.2.4. Data correction at source side

Both the HEMIS and SDM systems data validation process assists for correcting data on the source side. The HEMIS validation process allows one to set up an email sending mechanism for error records. For example, alerts regarding errors can be sent to the user for captured student demographic information pertaining to nationality, gender, race, etc. In the case of the SDM history tables, a mechanism was established to deal with records that fail to satisfy the business rules during the ETL process. The history tables are updated by setting the processed flag to "E" for error records. The system generates an email to the data owners for data correction by summarising the details of the errors from all of the history tables and MIS department communities.

7.2.5. Web interface and security

Both systems front-end reporting tools support web interfacing. This enables end-user access to underlying databases to be much easier and simpler while also reducing the burden of the MIS team in terms of managing security and user created reports.

7.2.6. Costing

Cost always poses a major challenge for TI decision makers in building and running new software systems. Both the HEMIS and DW systems incur significant costs regarding hardware, software and ongoing maintenance.

7.3. Differences

In the above sections, similarities between the HEMIS and SDM systems were discussed. The following sections will examine the core differences between the two

systems, which really need the attention of the TI decision makers when selecting a reporting solution for their organisation.

7.3.1. Data coverage

As previously stated, the HEMIS system supports only a subset of institutional data based on statistics of staff and students who qualified for government subsides, whereas data required for a number of other business areas in terms of reporting are missing from the system. The business areas not catered for regarding students, are Admissions, Financial Aid, Fees, Residence information, the Meal System, Library data, Research publications, and Prizes and Awards. With regards to staff, Payroll data, Human Resources data, Projects, etc. are missing. In the SDM, the entire body of student information, whether or not reported to the DoE, is provided for by utilising various star schema models, a few of which, such as financial aid, fees, and research publications, are not presented in this dissertation.

7.3.2. Data load frequency

It is not always possible to offer a real time decision support system, but when the latest information availability gap is decreased, more end-users are satisfied. For example, faculties, schools, and departments often desperately require the latest enrolment statistics at the beginning of the academic year so that they can plan and offer admissions to additional students before the enrolment cut-off date. Similarly, other statistics regarding student demographics, course credit values, subsidy information, etc. are required immediately. The HEMIS system data can only be made available to the University audience once the submission is finalised according to the deadlines as indicated in Figure 4-1. This delay in information availability also delays user queries whilst information is urgently required. Due to this factor, university users must rely on OLTP data which contains numerous data errors. On the other hand, the SDM data is loaded into the star models on a daily, weekly or monthly basis and information is made available to the University audience immediately once the load is completed.

7.3.3. Extraction, transformation and loading

The primary source of the HEMIS system is text files (six for students and two for staff) which a TI generates for DoE submissions. In order to facilitate the loading

process an interface is designed for importing these files. Before importing a submission into HEMIS, it is very important to make sure that the same submission has not been previously imported, otherwise the data validation process will never create an alarm regarding the duplications, and thus the data submission is duplicated. The data validation process was designed to ensure that all mandatory columns possess values. For example, if the race or gender of a student is missing, the system will reject such records and these error records will be forwarded to the data owners via email for data correction. In order to import data other than DoE text files, separate DTS packages using Microsoft SQL Server need to be written. This loading process constitutes a simple one-to-one map between OLTP and HEMIS tables, which, without any data cleansing, results in mushrooming tables with inconsistencies in the table and column names.

The SDM follows a standard ETL process and the source can be any OLTP or file system. The proper ETL process was designed for each data import, enforcing custom business rules and lookups. The process of loading HEMIS data from VALPAC2 Microsoft Access tables was automated by creating views that only draw the latest submission. The mappings written for ETL runs use the UPDATE/INSERT mode according to the unique key of the table that never duplicates the record. A proper error record keeping system was designed to track errors in the source systems. While loading the rejected records that result from the violation of referential integrity or business rules, they are stored in separate history tables with error flags and a time stamp. This process facilitates the continuation of the load process by rejecting error records instead of stopping the whole process, which would require a manual intervention to start the next step, if the load process failed.

7.3.4. Incremental loads

As stated in section 7.2.4, both the HEMIS and SDM systems can identify errors in the source OLTP system. The emailing procedure of both the systems communicates with the data owners to rectify the errors that resulted while capturing the data. The difference between the two systems is the availability of the data for querying and reporting that was corrected at the source OLTP side. In case of HEMIS, the fixed data will only be available the next time the submission is reloaded once the latest text files are extracted from the OLTP system. This whole process could sometimes take

weeks and even months, whereas the SDM employs a STAGING_BATCH_ID table (see Table 6-1) for incremental loads. Based on the last extract date the daily load process is made available for all the newly added and updated data at the OLTP side overnight in the SDM.

7.3.5. Data auditing

In HEMIS there is no mechanism to audit record counts of the HEMIS rows compared with the total number of rows on the OLTP side. For example, the data load process audits data according to the DoE definitions of the number of rows that were imported into HEMIS from text files. However, no statistics are available regarding the number of students registered for different courses and the number imported into HEMIS. In the SDM, audit packages run after the ETL process in order to compare the record counts in the SDM with the OLTP system. The record count with a "-" sign indicates that there were more records at OLTP side while a "+" sign indicates a greater number of records in the SDM. These audit packages help to provide confidence amongst the end-users with regards to the SDM data.

7.3.6. Database structure

The structure of the database plays an important role in extracting data from the base tables. The process of extracting data can only become efficient if join paths between tables are simple and contain numeric attributes. The major difference between HEMIS and the SDM is the database's structure. HEMIS follows the OLTP database structure with complex joins on textual attributes as depicted in Figure 7-1, whereas the SDM follows dimensional modelling star schemas with pre-calculated measures as illustrated in Figure 5-9.

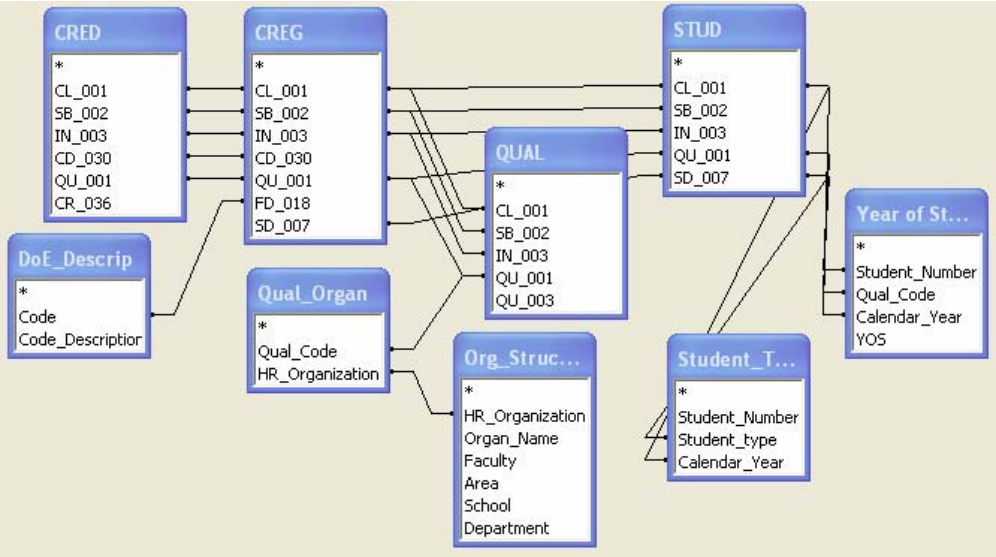

Figure 7-1: OLTP: database structure for course credit values

7.3.7. Course credit value and subsidy example

The course credit value example will validate the argument of simplicity in writing reports from the SDM as compared to the HEMIS system. The pivot table presented in Figure 5-11 shows facts regarding the teach-enrol-FTE, teach-input-units, and teach-input-subsidy columns that can be generated from both the HEMIS and the SDM systems. The differences are in the efforts required to generate these statistics from the HEMIS system. The dissimilarities between HEMIS and the SDM with regards to presenting course registration statistics follow below:

- **Pre-calculated facts**

In order for HEMIS to extract credit values, a query with complex joins as shown in Figure 7-2 is required. Similarly, complex queries are required for teach-input-units, and teach-input-subsidy. Queries of this nature extract the required facts on the fly by placing heavy loads on the reporting server, resulting in an extremely slow performance. On the other hand, by using the SDM course registration star presented in section 5.6, such information can be extracted by simply dragging and dropping columns on the pivot table.

- **Dimensions availability for analysis**

End-users continually need course credit values in terms of a breakdown per faculty, school, and department. This information is never available from HEMIS tables unless additional tables are created in the HEMIS database to incorporate the organisational hierarchy dimension. Similarly, some other important dimensions such as YEAR-OF-STUDY, STUDENT-TYPE, etc. are also not present in HEMIS. In the SDM all the required dimensions required in querying course registration statistics are incorporated in the design.

```
SELECT Org_Structure.Faculty, QUAL.QU_003, DoE_Descrip.Code_Description,
Sum(CRED.CR_036) AS SumOfCR_036, Student_Type.Student_type,
[Year of Study].YOS
FROM (((((((CRED INNER JOIN CREG ON (CRED.QU_001 = CREG.QU_001)
AND (CRED.CD_030 = CREG.CD_030) AND (CRED.IN_003 = CREG.IN_003)
AND (CRED.SB_002 = CREG.SB_002) AND (CRED.CL_001 = CREG.CL_001))
INNER JOIN QUAL ON (CREG.QU_001 = QUAL.QU_001)
AND (CREG.IN_003 = QUAL.IN_003)
AND (CREG.SB_002 = QUAL.SB_002)
AND (CREG.CL_001 = QUAL.CL_001))
INNER JOIN DoE_Descrip ON CREG.FD_018 = DoE_Descrip.Code)
INNER JOIN Qual_Organ ON QUAL.QU_001 = Qual_Organ.Qual_Code)
INNER JOIN Org_Structure ON Qual_Organ.HR_Organization =
Org_Structure.HR_Organization) INNER JOIN STUD ON (CREG.QU_001 = STUD.QU_001)
AND (CREG.IN_003 = STUD.IN_003) AND (CREG.SB_002 = STUD.SB_002)
AND (CREG.CL_001 = STUD.CL_001) AND (CREG.SD_007 = STUD.SD_007))
INNER JOIN [Year of Study] ON (STUD.CL_001 = [Year of Study].Calendar_Year)
AND (STUD.QU_001 = [Year of Study].Qual_Code) AND (STUD.SD_007 =
[Year of Study].Student_Number)) INNER JOIN Student_Type
ON (STUD.CL_001 = Student_Type.Calendar_Year)
AND (STUD.SD_007 = Student_Type.Student_type)
WHERE (((CREG.CL_001)='2006') AND ((CREG.SB_002)='3')
AND ((CREG.IN_003)='UFS'))
GROUP BY Org_Structure.Faculty, QUAL.QU_003,
DoE_Descrip.Code_Description,
Student_Type.Student_type, [Year of Study].YOS;
```

Figure 7-2: SQL for generating course credit value

- **Query performance**

Another factor that contributes to the success of query and reporting relies on the way in which efficient database returns result when an end-user submits a query. For better performance, a DSS system needs to be configured in respect of database structure as well as database configurations. The HEMIS system comes with a standard OLTP database configuration, basically to run day-to-day transactions so as to optimize database design for record inserts and updates. This database configuration never contributes to the query performance. Another factor that degrades performance is that of multiple join paths between tables on textual columns. The SDM was configured for the settings that were required for DW systems. Caching technology was used for better query performance with more space for buffering, a shared global

area and a number of other parameters as specified by database vendors for query optimization. In Table 7-1 differences in the response time of course registration statistics from both the HEMIS and SDM systems are presented.

Table 7-1: Query response time

	HEMIS	SDM
Load on the server	40%	5%
Response time	3 to 6 minutes	10 to 20 seconds

In the case of HEMIS, the load on the server was high. The database server consumed 40% of the CPU power during data aggregation, joining, etc. while generating course credit values, teach-input-units and subsidy. On the other hand, the SDM consumed only 5% of the CPU resources because data is stored in ready to use format by pre-calculating the required facts while loading data, and thus the query response time is much faster as compared to the HEMIS system.

7.3.8. Slowly changing dimensions (SCD)

Another major difference between HEMIS and the SDM is the management of SCD in preserving history. In HEMIS the SCD concept is totally absent and tables are managed by overwriting the contents of the previous columns with new data on each data load. One of the major candidates for preserving history is the QUAL (Qualification) table. In HEMIS the QUAL table (see Appendix D) was never designed to cater for home and teaching organisations/departments links with qualifications. In order to achieve this task a table, ORG-STRUCTURE, as shown in Figure 7-1 was created in the HEMIS database. A mapping table QUAL-ORGAN was also created in order to map organisations with qualifications. These settings work adequately for the current academic session when viewing the teaching or home organisation, that is, the faculty, school, and department of the qualifications. If a qualification is, however, moved from one such organisation to another organisation, there is no way to track the historical organisational structure of the qualification, thus resulting in inaccurate information. In the SDM, data regarding the qualification with a home or teaching organisation was provided from the PROGRAM dimension. To cater for the changes in the organisational structure, the SCD with type 2 change was implemented. Each time a change in the organisational structure (i.e., the qualification is moved from one organisation to another organisation) occurs, the SCD program

inserts a new row with a new surrogate key in the PROGRAM dimension along with other attributes. In Table 7-2 it is shown that the academic plan 7513 was moved from the academic programme Q7D3 to M7131. The data in the Start_Date column provides the cut-off date between the two academic programmes. The Plan_pKey was 1005 before 1/1/2004 and 8951 afterwards. For the latest academic plan the fact table rows are partitioned and preserve the history so as to display accurate information.

Table 7-2: SCD type change 2 in program dimension

Plan_pKey	P_Academic_Plan	P_Academic_Progam	Degree	Start_Date	End_Date	Status
1005	7513	Q7D3	7513	1/1/1901	31/12/2003	HISTORY
8951	7513	M7131	7513	1/1/2004		PRESENT

7.3.9. Report writing skills

The effectiveness of a new system can be checked from its users' group confidence level in generating statistics themselves. The HEMIS system users only receive training on how to open a report from the web interface already created by the HEMIS vendors or MIS team. To write a report from HEMIS, a user not only requires the skills to write reports using the reporting tools, but the major requirement is a knowledge of a database programming language / structured query language (PL/SQL) in order to establish join paths and database views for drawing data. It is very difficult to find such users with these capabilities other than those in a MIS team. Therefore, HEMIS report writing is limited to the MIS team and places users on hold for their ad-hoc requests, whereas the SDM star schema has simplified this process for which there is no need to create database views for drawing data at the reporting layer. Data is already stored and segregated into dimensions and facts that really make report writing much simpler and easier.

7.3.10.Front-end reporting tools

It is always a good practice to use only a few reporting tools because this simplifies tool management and the training processes. However, a single tool cannot provide all the functionality that a TI requires for collecting and generating various statistics. Most of the current reporting tools such as Business Objects, SAS, Cognos, Microstrategy, Oracle, etc. are based on the design of star schemas and in order to use the full functionality of these tools in designing a dashboards, predictive models, etc.,

facts and dimensions are the basic requirements. Owing to the relational structure of HEMIS, very few options remain in writing reports and in most cases a traditional

Table 7-3: Comparison between HEMIS and SDM

	Description	HEMIS	SDM
Similarities	Separate hardware and software	Required	Required
	Ongoing maintenance	Required	Required
	History data	Holds	Holds
	Data fixing at source side	Provided	Provided
	Web interface	Provided	Provided
	Costing	Expensive	Expensive
Differences	Data coverage	VALPAC2 files, subset of institutional data based on students qualified for subsidies	Institutional data that covers all business areas
	Data load frequency	Once submission is finalised in month(s) times	Daily loads
	Extraction, transformation and loading	Based on six VALPAC2 files using DoE data validation rules	Any source system. Uses DoE and institutional business rules
	Incremental loads	Each time submission is reloaded	Records are updated in the SDM from daily loads based on last extract date
	Data auditing	Not supported	Audit packages to check OLTP and SDM row counts
	Data structure	OLTP (ER Modelling)	Star schemas using dimensional modelling
	Pre-calculated measures	Not supported	Fully supported
	Dimension availability for analysis	Fewer dimensions available	Fully supported
	Slowly changing dimensions	Does not support	Fully supported, implemented using type 2 change
	Query performance	Very low due to transactional database optimization	Very high due to analytical database optimization
	Report writing skills	End-user needs reporting tool knowledge + database PL/SQL knowledge to format the data at backend level	End-user only needs reporting tool knowledge.
	Front-end reporting tools	Transactional reporting tools	Both transactional and analytical reporting tools

reporting layout is the only option in which to display data. On the other hand, the SDM dimensional modelling allows the writing of reports in a number of formats.

7.4. Comparison summary

A summary of the similarities and differences between the HEMIS and the SDM is presented in Table 7-3.

7.5. Summary

In this chapter comparisons between HEMIS and the SDM systems were summarised. It was explained that the similarities between the two systems contribute very little to the major differences which make the two systems totally disparate. It was also demonstrated that the HEMIS system's database structure, data coverage, ETL, incremental loads and the limited reporting toolset decreases its use for analytical and predictive modelling. On the other hand, the SDM query centric star schemas, standard ETL processes, management of the SCD and the richness in the data quality and reporting toolset renders it the real candidate for the solution for which the TIs are seeking help. The next chapter forms the heart of this study where the researcher presents his findings from his experience and surveys conducted during the current research.

Chapter 8

Survey results and researcher's experiences

8.1. Introduction

In the previous chapter comparisons between the HEMIS and SDM systems were provided. It was indicated that the SDM system is superior to the HEMIS system in a number of aspects. Major incompetencies of the HEMIS system exist in the database structure, the subset of the institutional data, the limited ETL functionality, and the efforts required extracting data for reporting and analysis. Therefore, the SDM is the perfect solution to be able to integrate and query institutional data. This chapter is purely focused on the research findings of the study and constitutes the seventh activity, "Evaluate effect of intervention in terms of research questions", in the AR cycle. The chapter begins by furnishing details of the email survey conducted in order to collect information with regards to the systems that were in use, or under development, by other TIs. The usability testing results where end-users have displayed confidence in the DW technology is also included. The researcher concludes the chapter with his experiences that draw attention to the factors which influence the use of DW systems.

8.2. Email survey

An email survey (see Appendix I) was conducted in 2005 in order to ascertain whether other TIs were in the process of establishing a DW for the institution. The major interest in conducting the email survey was to find out the level of DW awareness among the TIs management. It was also in the interests of the researcher to find out which back-end and front-end tools were planned or used by other TIs in their DW projects. Out of twenty two, only twelve institutions replied. The management of three TIs indicated that they had no knowledge of DW. It can be assumed that the rest of the TIs, where management never replied, may be lacking knowledge on DW. The survey results of the email survey are discussed in the following sections.

8.2.1. Data warehouse efforts

The intention of this question was to establish the efforts towards DW by other TIs. The figures for the different institutions are reflected in Figure 8-1. Out of the 12 TIs,

33.33 percent responded that a DW is in place and in production, 16.66 percent replied that DW efforts were in development, 25 percent were in the planning phase and the remaining 25 percent reported no future plans. The 25 percent who responded that no plans were in place, indicated that they were deterred by the cost of DW development and maintenance.

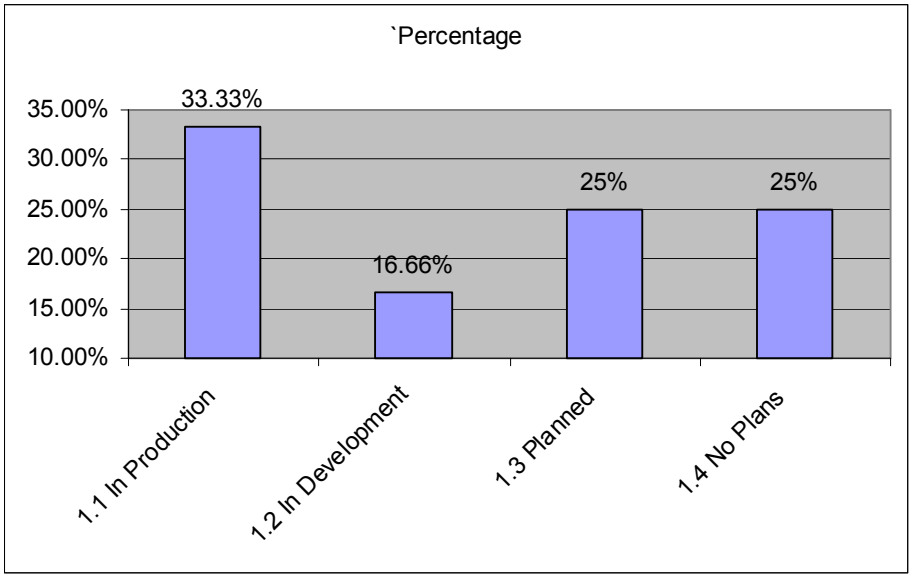

Figure 8-1: Data warehouse effort

8.2.2. Data warehouse data

Data in a TI is captured for different business areas such as Human Resources, payroll, etc., however, the main difficulty lies in the DW for student data. Faculties, schools, departments and top management have expressed concerns regarding subsidies and the quality of students because these are the major sources in generating subsides for the institution. From Figure 8-2 it is shown that out of the 4 TIs where DW is in production, all 4 institutions invested in student data, 4 of them also invested in financials, 3 of them in Human Resources, and 2 of them having invested in Payroll.

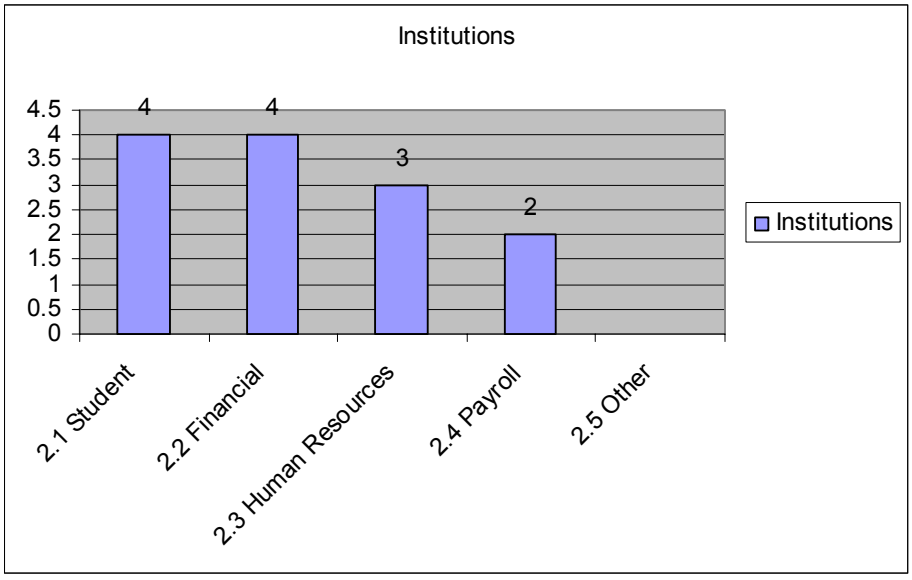

Figure 8-2: Data warehouse data

8.2.3. Executive sponsorship

The success of DW lies in executive sponsorship i.e. that the executives are involved in the DW development and maintenance. From Figure 8-3 it is indicated that of the 4 TIs whose DW efforts are in production, 75 percent of the DW projects were sponsored by the Chief Executive. The main reason behind this sponsorship is the large investment on DW systems that needs the recommendations and sponsorship from the Chief Executive that is the Vice-Chancellor of the Institution.

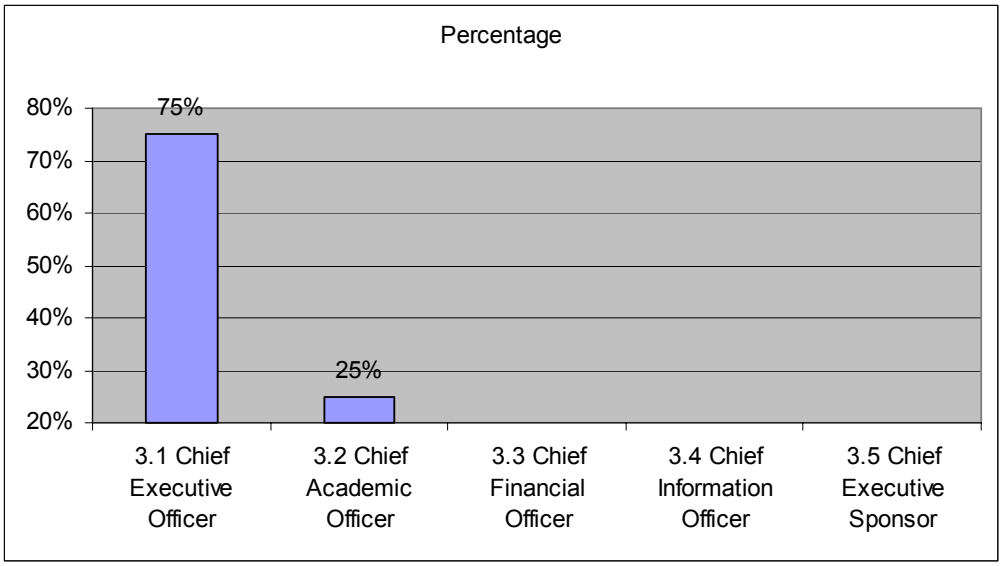

Figure 8-3: Executive sponsorship

8.2.4. Data dictionary software

One of the greatest challenges involved in any IT project is the selection of the software. Currently, in South African TIs the trend is towards the purchasing of ERP systems. The two main vendors for ERP solutions are Oracle and Peoplesoft. Both ERP applications use the Oracle database at the back-end. From Figure 8-4 it is evident that 50.00 percent of TIs that have their DW in production invested in Oracle, while 33.33 percent utilise Microsoft SQL server whereas 16.66 percent are busy with other databases.

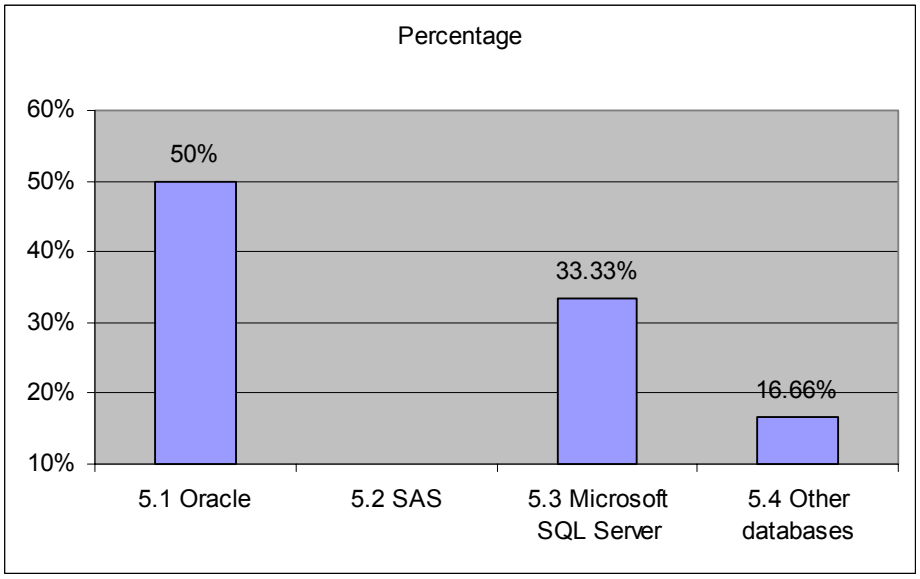

Figure 8-4: Data dictionary software

8.2.5. HEMIS system

The third party HEMIS system market is quite young and still maturing with regards to the improvement of providing an interactive interface for the DoE VALPAC2 system without considering the ease at the back-end database structure. As seen in Figure 8-5, out of 12 TIs only 16.67 percent of institutions use a third party HEMIS system such as HEDA, whereas 50 percent of institutions are bound to use VALPAC2 systems. The TIs who responded with "Other" most probably are using DW systems. There is still a strong possibility that other TIs still seeking assistance and ready to invest in an optimal solution could be motivated towards DW technology.

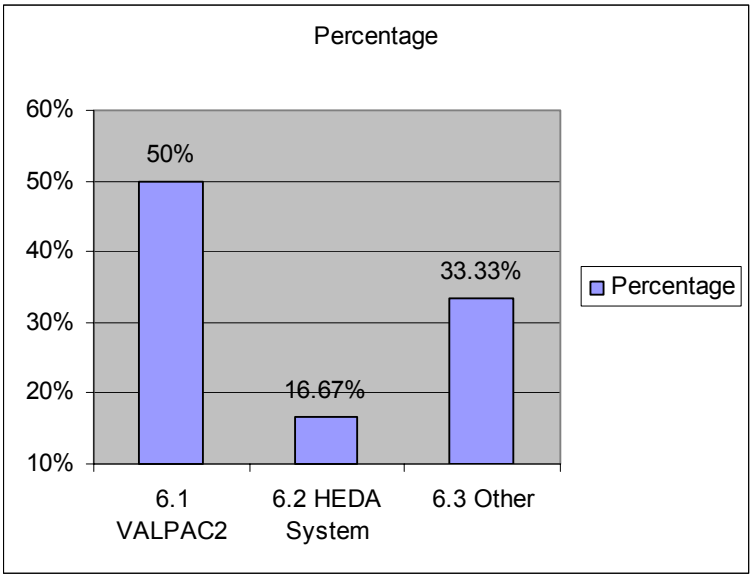

Figure 8-5: HEMIS system

8.3. Usability testing

Usability is the combination of fitness for purpose, ease of use, and ease of learning, all of which render a product effective. In this section information gathered with regards to usability testing will be reported. The results of the questionnaires that were completed while providing training to the end-users are available in Appendix L.

It was not possible to obtain feedback from end-users regarding the comparisons between the HEMIS and DW database designs from the back-end, because they have no interest in querying data directly from the database structure. Thus, the only method available for feedback as to the end-users' likes or dislikes while querying data from relational and multidimensional database structures is to afford them access to the reporting tools. Training and usability testing were carried out at four different levels of management.

8.3.1. Academic information student unit (AISU group)

This group is responsible for providing reports to faculties, schools, departments and top management regarding students. The unit is responsible for writing details and summary level reports from Student Registrations, Admissions, Fees, Financial Aid, Residence Occupation, Research Supervisors, Longitudinal Studies, etc. The group consisted of 6 users who has been exposed to relational as well as multidimensional reporting tools.

8.3.2. BI user group (BI group)

This group consists of business managers who would like to write their own reports or extract information for the requirements specific to their area. This group of 12 users has been exposed to both relational and multidimensional tools for reporting.

8.3.3. Head of schools

This group is interested in viewing summarised school reports with an OLAP capability to drill down to the lower details at the student level. This group is based on 35 users. Most of the users of this group are not interested in writing their own reports. This group has been exposed to multidimensional reporting tools because of the ease of extracting data.

8.3.4. Portfolio executives

This group of top managers is only interested in high level summary data at faculty level in order to ascertain the direction in which the organisation is moving and to determine whether it will be among the top TIs list in the next few years. The group consisted of 10 users who is excellent candidates for OLAP which offers fast responses while "slicing and dicing" different dimensions of the model.

8.4. Usability testing results

The following section covers usability testing statistics drawn from the end–user's experiences in terms of a few critical questions about the OLTP and DW systems. The percentages shown in the graph in each column represents the percentage responses within the group. The "Univ" group is representing the average responses of all the participating groups.

- **Are you happy with the report writing capabilities of the VALPAC system?**

The responses to this question are contained in Figure 8-6. As depicted in the graph, 42.22 percent of the University users responded that they are "Sometimes" happy with the reporting capabilities of the tool. Of the respondents, fewer members of the AISU group than those of the "heads of the school" group were satisfied, although the heads of the school group had not been exposed to report writing owing to the complexity involved in extracting and interpreting data in the underlying tables. The reason for this response by the heads of the school group can be attributed to their being assisted

by the AISU to successfully write the reports they require utilising the OLTP database.

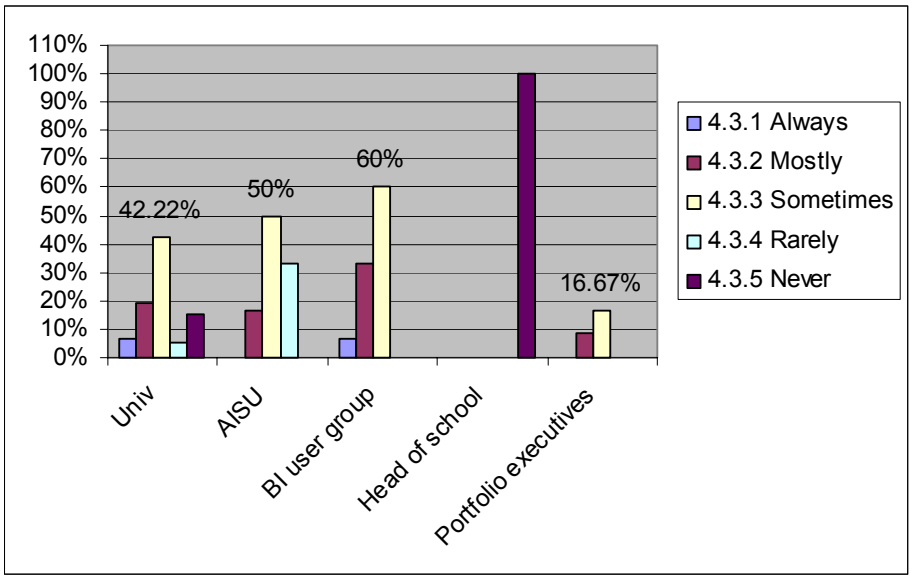

Figure 8-6: Report writing capabilities

- **Are you happy with the underlying database structure of the VALPAC system?**

A very small percentage of end-users were satisfied with the database structure. In Figure 8-7 it is evident that 81.5 percent of the audience responded that they have "Never" been satisfied.

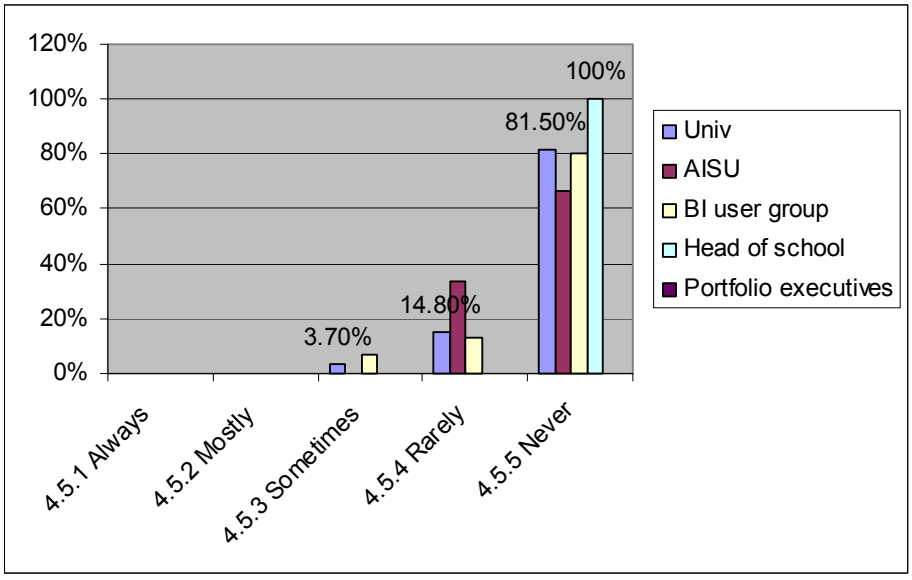

Figure 8-7: Underlying database structure

The basic reason for their dislike of the system is the joining of the tables on multiple columns, anomalies in the data, and the sub-query requirements for extracting the latest definitions.

- **How often do you ask the information system department staff to write reports for you?**

The responses to this question are diverse. 38.5 percent of the University users always depend on the MIS staff to write their reports. Similarly, 28.2 percent of the University users mostly asked the information systems team for assistance (see Figure 8-8). The AISU group which is mostly involved in assisting University users in providing data and report assistance are not willing to open access to the underlying tables to end-users. The main argument for this is the high probability that the end-users could easily extract incorrect data by selecting incorrect joins and replicate data in multiple tables.

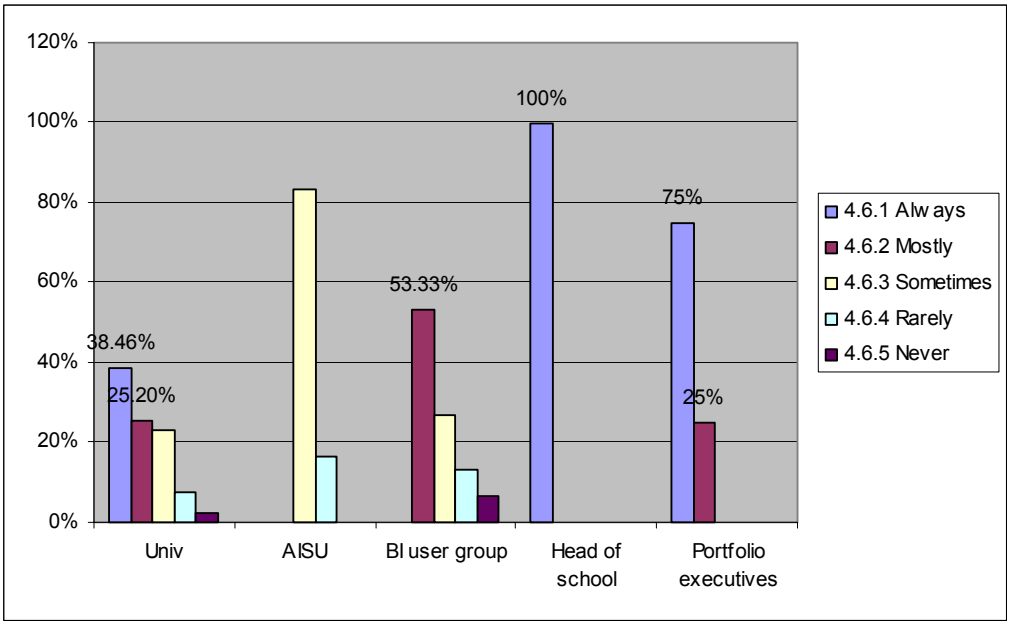

Figure 8-8: Report writing assistance

- **Do you agree that the current available systems can answer all reporting requirements for student and staff statistics?**

In Figure 8-9 it is shown that 43.6 percent of the University users responded that the current reporting tools and environment only sometimes fulfil their reporting needs, whereas 38.5% users responded that they rarely do so. The argument for this dislike is that the system's capability to generate transaction level details requires extra measures to generate FTE's, input and output subsidies by adding credit values and manipulating teaching-splits in the spreadsheets.

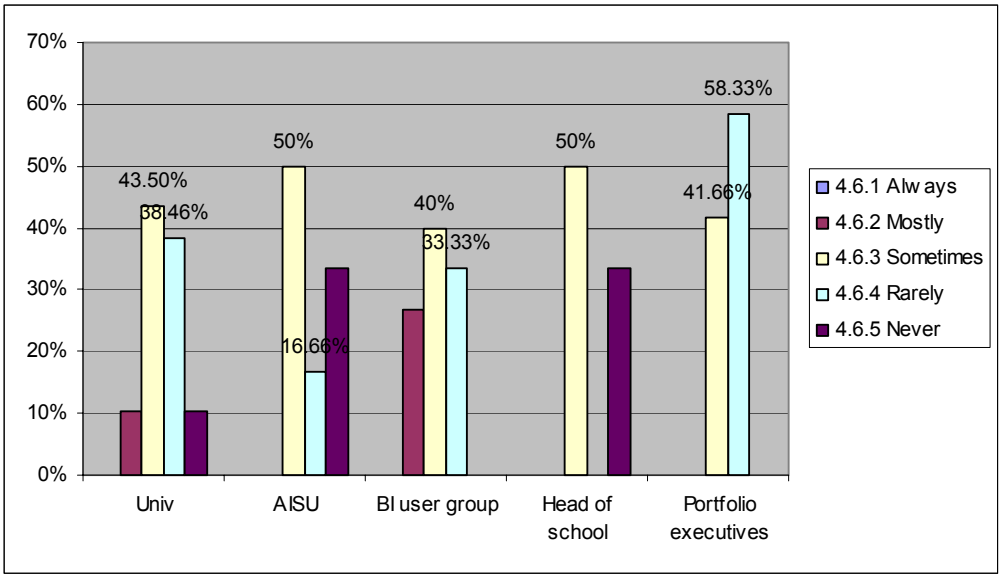

Figure 8-9: Reporting requirements

- **Please give your report-writing experiences regarding the SDM.**

Positive feedback is given with regards to the reporting environment setup of the SDM as illustrated in Figure 8-10. 76.9 percent of University users were satisfied with the structure of the data in the SDM due to its simplicity for the user in querying and writing reports. Similarly, the percentages in the usefulness of the SDM in other areas were also high as indicated in the above figure.

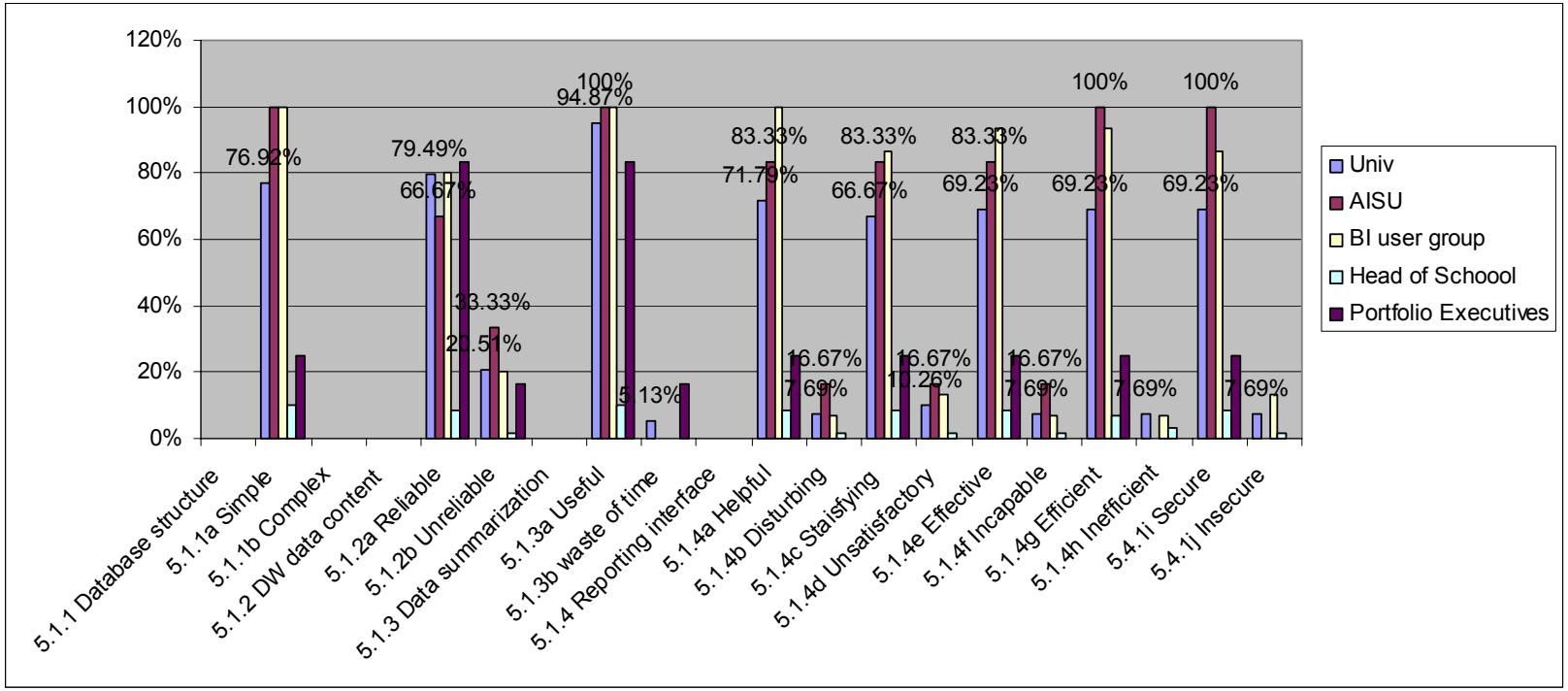

Figure 8-10: DW reporting

- **Do you think that the SDM reporting environment has made you self-sufficient in querying and writing your own reports?**

The response rate to this question indicated an increase in the DW suitability for the end-user audience of TIs as shown in Figure 8-11. Of the portfolio executives and heads of school group respondents, where it is not likely that they would ever show an interest in writing reports themselves, 83 percent responded with the option, "Always". The basic reasons for favouring the SDM are that the star schemas present data in terms of facts and dimensions, thus rendering the data extraction more understandable.

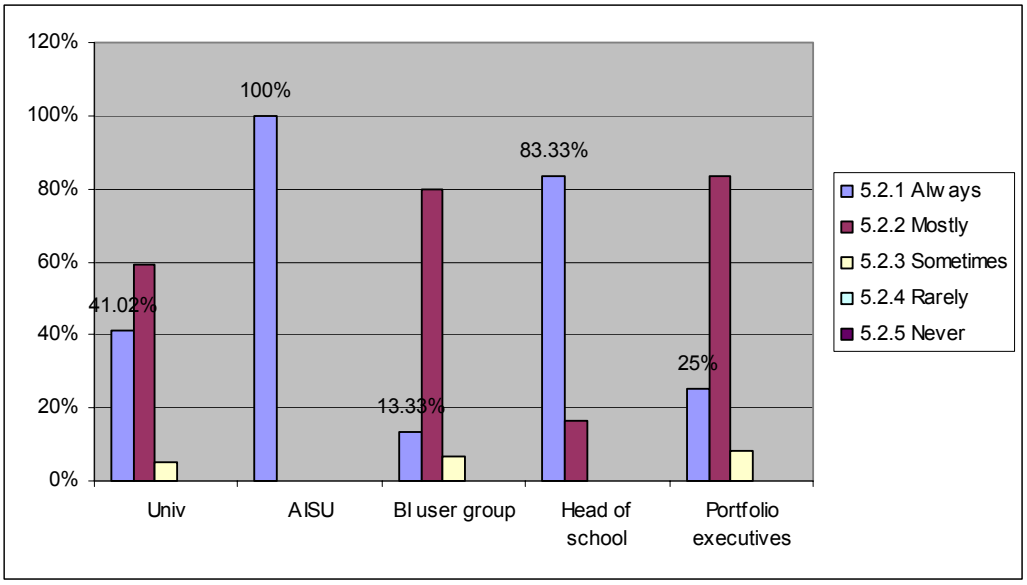

Figure 8-11: DW reporting environment

8.5. Researcher's experiences and research findings

The author of this study wishes to highlight factors that are forcing TIs to enter the DW arena in order to resolve problems that a TI currently faces while extracting data from the OLTP systems. In the following sections the researcher will describe his findings and research experiences pertaining to this study due to his active involvement in the development of the SDM.

8.5.1. Tertiary institutions' survival

Most of the surviving TIs have endured periods of tough rationalisation and mergers. In order to survive and remain competitive, they have grown and developed into large businesses in and of themselves. Strategic decision making has now become of utmost

importance to the TIs in order to meet the rapid changes in the government funding structure. TIs must develop advance enrolment plans in order to register quality students for qualifications that generate more subsidies for the institutions. To develop advance enrolment plans, quality and integrated data are a pre-requisite. This quality data would be helpful to provide an understanding of their students, services that institutions are providing, and areas that need greater attention. A DW can be helpful in fulfilling the management reporting requirements by readily providing data ready to use for them.

8.5.2. The OLTP system is not suitable for strategic information

Data integrity is the integral component of the OLTP system, but unfortunately OLTP systems were implemented without enforcing referential integrity. Another major factor is the presence of thousands of tables in the OLTP system. In the case of the UFS the database contains 16,000 tables that make data extraction extremely complex. The non-standardisation in table and column names makes the process of joining tables even more difficult. This lack of referential integrity, inconsistency of data, thousands of tables and numerous data errors, in any case are not suitable for strategic information where a stable set of rows is an imperative requirement.

8.5.3. Third party software is not suitable for strategic information

One of the main reasons for purchasing the third party HEMIS reporting systems is the need to clean and validate the data according to the formats required by the DoE. In reality these systems do not come with any data cleansing features and even the database structure is based on the OLTP system, which is complex for querying and reporting. For strategic information, query performance is the basic requirement and owing to the OLTP database structure, optimal performance is not achievable.

8.5.4. DW simplifies database structures

An OLTP system, such as a Student Information System, is designed to optimise transactional processing. The features that increase the efficiency of these systems are generally those that also make it difficult to extract data simply and without any major impact on the production databases (Stevenson, 1997). In order to optimise update operations in a transactional system, data redundancy is minimised. This makes extracting data complex because it is necessary to access and link a number of tables

in order to retrieve the required data. This linking also creates a severe load on the databases.

DW uses a star model based on dimensional modelling such as the various models presented in chapter 5. The dimensional model is simple and it is easy to extract data without writing complex queries. It is also optimised so as to expedite queries. The number of tables can be reduced to 15 by using the star model.

8.5.5. Problems in the OLTP database can be solved in the staging area

Another important critical factor regarding the support of DW is data cleansing which is difficult or in most cases impossible at the OLTP side. In a DW staging area this data can be filtered, standardised and enriched with correct data, both for internal management purposes and DoE purposes. The challenge for the DW is to cleanse the data so that it can be correlated and analysed appropriately. Data can be easily cleansed by utilising commercial cleansing tools to create transformed and standardised data according to business processes. The ETL process prevents data from reaching the DW layer unless it satisfies the business constraints and logic. This process assists data owners to repair the incorrectly captured data at the source.

8.5.6. Limitations of MOLAP storage and data retrieval

MOLAP technologies offer faster data retrieval at the cost of space storage and hardware requirements. An institutional DW cannot possibly exceed the size of that required for the data residing in banking, insurance, airlines, or other data warehouses. In the current installation, 250 GB of space was used for creating the different MOLAP cubes from the SDM which takes 7 to 8 hours for cube maintenance. In models, such as the one for course registrations where users drill down to the course level, the use of MOLAP technology results in frustration due to response times. Another challenge emerges when the model has more than nine dimensions. For example, the PROGRAM_ENROLMENT_FACT table as shown in Figure 5-6 possesses thirteen dimensions thus rendering it impossible to develop one cube using all these dimensions owing to the limitations of the technology. When using Oracle MOLAP, if one creates a cube with more than nine dimensions with full aggregation, it takes approximately two hours to calculate the totals of hierarchies and furthermore, this consumes a great deal of space as shown in Figure 8-12.

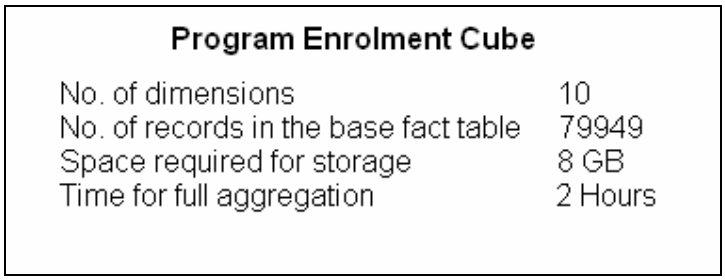

Figure 8-12: MOLAP Cube maintenance

On the other hand, ROLAP tools work much better than the MOLAP tools due to the limited size of institutional data. Queries run slightly slower than MOLAP queries but the user enjoys greater flexibility in writing a variety of reports from the star schema. Users can easily shift between different hierarchies and format the output as required. Another major advantage is that the MIS staff gets released from the cube maintenance process.

8.5.7. Usability testing results

From the feedback collected regarding the usability testing it was clear that the OLTP system is not helpful in management reporting as a result of the complexity of generating reports owing to the complex database structure and numerous errors in the database. On the other hand, users feel more comfortable and confident when compiling reports from the DW star schema. Heads of schools and portfolio executives were not likely to write their own reports, however, it was evident that users prefer to extract data themselves.

8.5.8. The tendency for other TIs to adopt data warehousing

From the email survey results presented in section 8.2 it was indicated that out of 12 TIs only 33.33 percent are using DW and in most cases management of these institutions are not aware of this technology. There is a strong possibility that other TIs who are still seeking assistance can be convinced that DW is possibly the only solution in this regard.

8.6. Recommendations

In the following sections the author of this study would like to present his recommendations that are more feasible in implementing DW solutions.

8.6.1. Start small

In the Introduction it was indicated that universities work with a limited budget. After several poor and costly decisions they are wary of any new IT expenditure. Ways and means had to be found to develop a DW at a minimum cost. Wierschem et al. (2003) pointed out that the development cost of a DW can be minimised by reducing the overall scope of the project which can also be broken down into smaller components and developed over a longer period of time. For these reasons it was decided to develop only the SDM as a start.

The selection of developing a SDM was based on the research of Thomas (1997) who found that institutions of higher education adopt a bottom-up approach in the development of a DW and most often begin with SDM. Financial data was the second most reported area, with Human Resources data a close third.

8.6.2. Use what is available

To further reduce costs no special hardware or software needed to be purchased, but existing hardware and software could be utilised. Preference should also be given to using existing in-house staff instead of seeking outside help (Lamont, 1999). This advice was followed and a single person trained in DW was placed in the Planning Unit (MIS Department) of the University in order to develop the SDM. Detailed requirements were established before and during the development process and therefore there was no chance for misunderstanding in the requirement gathering phase.

The UFS is currently using Oracle, MySql and Microsoft SQL Server for maintaining its production data. MySql does not include BI tools and to use Oracle BI tools, a separate software licence is required. Microsoft SQL Server comes with built-in BI tools. Instead of purchasing the expensive Oracle Warehouse Builder and other BI tools, preference was given to Microsoft SQL Server 2005 for OLAP, DM and ETL.

8.6.3. Make use of summarised granularity

In order to store student enrolments with plans and courses, two different fact tables were designed in the current SDM: STUDENT_ENROLMENT_FACT (see Figure 5.6), with granularity "student per semester per plan", and

STUDENT_COURSE_FACT (see Figure 5.9), with granularity "student per semester per plan per course". The main reason for designing two separate fact tables was the grain and summarisation of the fact tables. The STUDENT_ENROLMENT_FACT store summarises data and it, therefore, contains only 13 percent of the number of rows of the STUDENT_COURSE_FACT table. During the requirement gathering phase it was identified that most of the reports used for reporting and planning purposes were based on the head counts of students in a semester or a year. The STUDENT_ENROLLMENT_FACT table can answer these queries more rapidly than the STUDENT_COURSE_FACT table.

8.6.4. Derived and calculated fields and their effectiveness

Information is stored as raw data in the OLTP and HEMIS system and users are supposed to have intensive knowledge in extracting such data. For example, to calculate student FTE values a set of complex procedures are required to extract data. The Students' courses details reside in a separate Course Registration File (CREG) (see Appendix B) and the credit values per course reside in the Credit File (CRED) (see Appendix C). A user must download data into a spreadsheet (even in a reporting layer, data needs to be transformed and requires calculation) in order to achieve reasonable presentations. In a DW the presence of derived or calculated fields as facts in the fact tables makes this job much simpler and easier by affording access to information already calculated in the desired format. The presence of such values gives real value to DW users.

8.6.5. Make use of junk dimensions

In this project it was concluded that junk dimensions are very useful in the SDM in order to enrich data with regards to providing certain statistics which are not available in the source OLTP system (see the STUDENT_ENROLMENT_FLAGS in Figure 5.6). For example, the DoE needs student enrolments with the primary plan and this information is not available in the source data. A set of filters were used to find students' records that qualify for government subsidies. Some of these filters included: enrolments before or after the census date, students who had dropped out from their plans before or after the census date, approved qualifications, undergraduate students who fulfill their matriculation requirements and students who

failed or received re-assessments. This information is processed in the staging area and corresponding flags and indicators are added to the extracted data.

From Figure 5-6 one can see the benefits of using the STUDENT_ENROLMENT_FLAGS dimension. Prior to implementing the SDM, it was very difficult to determine the students' primary plans and students who qualified for government subsidies. There is no longer any need to write complex queries in order to find these statistics, because the STUDENT_ENROLMENT_FLAGS dimension renders this task very easy. One can merely drag and drop data elements onto the pivot table and obtain the required results.

8.6.6. Get management involved

For the DW to be successful it is imperative to cause management to be involved as soon as possible. It would have been ideal to have them involved from the beginning of the project. Unfortunately people in management and even in the MIS departments are not familiar with DW and its potential benefits and possibilities. They are only aware of the high costs involved and the many reported failures owing to wrong IT decisions in the past. Thus, usually with a limited budget, they are exceptionally wary of supporting new unfamiliar developments.

In this case it was found that the best way to gain the support of management was to gradually make them knowledgeable of what a DW can do for them. This was achieved by developing the SDM using in-house expertise and existing resources and demonstrating to them the ease and accuracy with which information can be extracted. During the demonstrations they were given the opportunity to request any student statistics they wished. By providing the answer immediately, through merely dragging, dropping and drilling down into the pivot tables, they became aware of the new possibilities. In addition, they could see that the DW provides the same as well as more information than the third party software systems for which they pay annual licence fees. Thus, when a fully fledged DW needs to be developed the chances are good that their support will be forthcoming.

8.6.7. Major subjects or topics as tables in the warehouse

There is no need to include everything in the DW. During the requirements analysis, the data or subject areas of interest to the users can be identified. The data is organised

in star schemas as shown in chapter 5 by splitting source complex data into dimensions and fact tables. It is very easy for end-users to draw their reports from star schemas as compared to the OLTP reporting tools where users are usually afraid of writing reports using complex joins which result in incorrect output.

The inclusion of a subject area does not mean a replication of a few of the tables from the OLTP database in the DW in the interest of the users. Actually, there is sometimes a need to create tables that do not exist at the source. For example, the student head count fact table is derived from the Student File (see Appendix A). In the Student File there is one entry for each qualification for which a student is registered whereas the DoE is only interested in approved qualifications and those for which there is at least one course registration for that academic year. Similarly, materialised views can be used to further summarise data from a fact table so as to improve performance.

8.6.8. Get end-user support

The return on investment of the data warehouse depends on its use. If users cannot understand the value of data in the DW it is possible that they prefer to access the live data because they can correct errors and view the changes online. On the other hand, in the DW environment, they have to wait until the next data is reloaded and this may cause frustration. Such users can be easily convinced of the value of DW by furnishing them with examples of the data cleansing process that does not exist in the OLTP systems and results in incorrect statistics. This can present a major challenge if users are not convinced of the value of DW data and continue to seek reporting tools on top of the OLTP database structure for efficient and reliable reporting.

8.7. Summary

In this chapter, survey results along with experiences of the researcher and his findings were described. The results of the email survey to ascertain the position of other TIs in terms of providing quality data regarding querying institutional data were presented. These results indicate that only a small percentage of TIs possess knowledge of DW technology and that there is a strong possibility that they would consider this as a possible solution for the problems they currently face. The usability test results clearly indicated that if an organisation wants all levels of management to become self sufficient in extracting and writing their own reports they should consider

DW dimensional modelling. Valuable lessons learned during this study were presented by the author in- order to convince management of other TIs, who have never utilised and explored the DW solution, of its value. In the next chapter the conclusion of the study is furnished. The researcher also needs to justify the research objectives and hypothesis of this study.

Chapter 9
Conclusion

9.1. Introduction

In the previous chapter, the results of the survey and the experiences of the researcher indicated that they collectively support the implementation of a DW. Furthermore, the usability testing also indicated that neither the OLTP nor HEMIS systems can provide value to the data with regards to querying and analysis as compared to the query centric DW system. The researcher also furnished recommendations that would render the implementation of a DW feasible according to the TI's resources and budget constraints. The present chapter contains the conclusion of the study, which constitutes the eighth activity in the AR cycle, "Exit, if questions are satisfactorily resolved".

9.2. Motivations for this study

This study focused on the data and reporting issues that a TI currently faces. There are number of factors that have opened research possibilities for the researcher in TIs. The backbone factor that is attracting research attention is the frequent changes in the government funding policies. Since the primary source of income of a TI is based on government subsidies, its income is dramatically affected by the changes in the government funding policies that require more and more student and staff statistics.

9.3. Factors enforcing TIs for seeking help

In order to survive in terms of the new funding formula a TI must increase the throughput of their students in order to generate an increasing amount of subsidies for the institution. These changes are forcing a TI to behave and react like any other commercial business. Therefore, the modern TI must develop advanced enrolment plans for quality students to register for qualifications that can generate more subsidies. It is also very important to monitor the progress of the students timeously in order to achieve the outputs (i.e. a greater number of students must qualify within the designated qualification time period followed by the DoE). The primary hurdle in establishing a proper MIS system for strategic and analytical reporting is the institutional data that is sitting with the following problems:

- **No enterprise view of institutional data**

In case of the UFS there are a number of database systems like the Student System, Library System, Research Publications System, Human Resources, etc. that are in use for maintaining the transactional records. Each system was built or purchased separately and resulted with different database structures, data definitions and ended with disintegration between the systems. It is not possible to create a heterogeneous environment for linking all these systems to create an enterprise view of data.

- **OLTP systems with numerous data errors**

To address the consistency issues in the database structure and definitions, the management of the UFS has purchased an OLTP system from an international company at huge cost. Within one year after the installation of the new OLTP system, the University faced a number of new challenges that are explained in details in chapter 3. The new system proved inadequate of generating the required statistics owing to numerous data errors occurring during the data migration and poor database design.

- **DoE HEMIS system**

At present, in order to maintain the DoE National Database, TIs use the VALPAC2 system for data submissions, which assists in data validation according to the DoE business rules. It was indicated in chapter 4 that the VALPAC2 system has no value for institutional internal reporting or for building forecasting and analytical models. The basic deficiencies of the VALPAC2 system are inherent in the subset of institutional data, the limitations of the system that cannot hold more than the data for two years, its complex database structure, a fixed set of reports, etc.

- **Third party HEMIS system**

The third party HEMIS has overcome the VALPAC2 limitations by providing interactive reporting interfaces with the web version and mechanisms for correcting data at the OLTP side. However, the primary limitations of the system such as the complex database structure, the data of students and staff being limited to the DoE files requiring complex PL/SQL programs to generate input / output subsidies of the

student FTEs, confines its usability to technical staff only. Even after implementing the third party HEMIS system, a TI must still contend with the same set of problems and data reporting issues that were faced after the implementation of the OLTP and ERP solutions.

9.4. Why the UFS was chosen for this study?

The study was conducted at the UFS because an ideal environment was available for this research. The UFS has implemented all these OLTP systems, the ERP solutions and third party HEMIS system in order to establish a proper reporting environment. But, unfortunately none of the systems were successful in establishing proper management information for the University.

9.5. Research design

To answer all the shortfalls that were addressed in the previous systems a DW infrastructure was selected in order to establish a MIS for the UFS due to its capabilities in integration, data cleansing and presentation of data to the end-users by means of dimensional modelling. The research methodology was based on AR due the active involvement of the present researcher in this development. The DW solution was implemented using a bottom-up approach by developing a SDM first. Various star schemas were designed, as described in chapter 5, for data querying. It was illustrated in the star schema diagrams that the HEMIS and other institutional data can be presented from one platform. Facts such as the FTEs, input/output subsidies, etc. requiring complex coding and PL/SQL programs, can be loaded during the ETL process, thus increasing the usability of the SDM. The experiences of the researcher while exploring the OLTP, HEMIS and DW systems explain why no system can better provide data cleansing mechanisms and a reporting environment than DW. The details regarding the HEMIS and DW comparisons were provided in chapter 7.

9.6. Feasibility of the DW solution

The usability testing that was carried out in order to gather feedback from the end-user with regards to their experiences with the OLTP, HEMIS and DW systems proved that a DW is the only solution they seek. The experience of the current researcher further contributes to the evaluation of the DW solution for a TI while the

survey results prove that a DW solution can be successfully implemented to provide a MIS for a TI despite the difficult challenges involved in the development thereof. In the following section details are provided on the empirical evidences that a DW is the most optimal and feasible solution to address all of the information issues that a TI are facing today.

- **Low budgets**

It was stated previously that universities are running with low budgets and that development of a DW requires heavy investment in implementing and maintaining the solution. It was shown in this study that the development and maintenance cost can be minimised by using existing hardware and by reducing the overall scope of the project, which can also be broken down into smaller components and developed over a longer period of time. It was also shown the cost of the third-party HEMIS system can be saved and used for the development of a DW.

- **Scarce BI resources**

In the country like SA with scarce BI resources the DW implementation seems to be expensive and not affordable by organisations like TIs that are already running on low budgets. The implementation of DW in such cases can be achievable by opening the door for new leaderships from Engineering and Computer Science departments. With proper training a successful DW can be developed using in-house experience.

- **DW dimensional modelling**

Another major supporting factor for the feasibility of a DW is its database design. None of the HEMIS system, DoE VALPAC2 and third-party solution supports this database design. It was shown in the usability testing results that it is very easy for end-users in querying data using facts and dimensions because data is organised in a way that is required for outputs. It was stated in the introduction that very few guidelines are available for fitting student and other institutional data into star schemas. The star models presented in chapter 5 shows that student data can be adjusted into dimensional modelling and the use of junk dimensions provides suitable places for flags and indicators that are used frequently in the institutional data.

- **Data problems at OLTP data can be fixed from the ETL**

The numerous errors reported in chapter 3 can be removed and minimised by developing a proper ETL before moving data into DW. The efforts invested in data cleansing are shown in chapter 6 and it was also shown that the ETL auditing process is helpful in identifying and correcting errors at the OLTP side.

- **ROLAP model in querying and analysis**

Decision makers always prefer systems having capabilities in slicing and dicing data in rapid fashion. Commercial businesses are investing in MOLAP technologies for providing better querying performance to their end-users. These businesses are able to bear hardware and maintenance cost that is required to populate MOLAP resources. Due to the size of institutional data, MOLAP resources are not required and ROLAP can easily satisfy its client base and this ultimately removes overheads from the MIS team in maintaining a separate repository for MOLAP.

The above evidences presented in this section prove that an effective DW solution can successfully be implemented to provide management information for a TI despite difficult challenges involved in its development.

9.7. Proposal for future studies

The focus of the proposed future study is intended to fall on the effect of the user's cultural background, experiences, and skill levels on the success of the DW. "In early implementations, the corporate DW was intended for managers, executives, business analysts, and a few other high-level employees as a tool for analysis and decision making. But today's DW is no longer confined to a select group of internal users. Under present conditions, corporations need to increase the productivity of all members in the corporation's value chain" (Ponniah, 2001, p.59).

When an organisation would like to open its DW to the entire community of users in the value chain, there is a need to identify certain issues in the selection of BI tools in terms of the user's cultural background, experience, skill levels and suitable training methods, which will emerge due to the diversity of users. According to Carmel et al. (2001), the increasing diversity of an information systems development workforce is bringing cultural issues to the fore. Similarly, as stated by Waldegg, (n.d) that "one of

the most important challenges is avoiding usability problems caused by (the) diversity of the users' cultural backgrounds". These cultural issues are more relevant in a country such as South Africa with its diverse peoples and cultures. De Wet, Bilgnaut, and Burger (2002) indicated that this multiculturalism is bound to influence the usability of software in this country. It is hoped that this study will increase the usability of the DW in the South African organisations in which a rich diversity is present at different levels of management.

9.8. Summary

This chapter concludes the research findings by summarising the problems that a TI currently faces in establishing a proper MIS for the institution. The consequences of implementing the OLTP, ERP, and HEMIS solutions were presented in the study. It was evident that these systems are predominantly deficient in a number of areas, in particular, with regards to poor data quality, complex database structures, and complex reporting infrastructures. All of these issues were addressed in the development of a SDM where the ETL and dimensional modelling renders a DW an ideal environment for data querying and analysis. Valuable lessons were learned during this process. These lessons were shared in this study in the hope that it will be of value to other TIs which still need to take the DW step.

References

Allan, R. G. (2000). *The impact of the OLAP/OLTP culture conflict on data warehousing*. Retrieved July 23, 2005, from http://www.georgetown.edu/users/allanr/Impact.pdf

Allan, R. G. (2001). *Data models for registrar's data mart*. Retrieved July 23, 2005, from http://www.tdwi.org/research/display.aspx?ID=5118

Baskerville, R. L., & Wood-Harper, A.T. (1996). A critical perspective on action research as a method for information systems research. *Journal of Information Technology*, Vol. 11, pp. 235-46.

Baskerville, R. L., & Wood-Harper, A.T. (1998). Diversity in information systems action research methods. *European Journal of Information Systems,* Vol. 7, pp. 90-107.

Baskerville, R. L. (1999). *Investigating information systems with action research.* Retrieved November, 20, from http://www.cis.gsu.edu/~rbaskerv/CAIS_2_19/CAIS_2_19.html

Battle L. & Degler, D. (2001). *Around the world in 80 clicks*. Retrieved August 8, 2005, from http://www.ipgems.com/writing/80clicks.htm

Bevan, N. (1995*). Human-computer interaction standards:* Proceedings of the 6th International Conference on Human Computer Interaction, Yokohama. Anzai & Ogawa (eds), Elsevier.

Bunting, I. (2002). *'Funding' in transformation in higher education: Global pressures and local realities in South Africa*, Centre for gigher education transformation in South Africa (CHET), Pretoria.

Bourges-Waldegg, P. & Scrivener, A.R. (1996). "Designing interfaces for culturally-diverse users". *Proceedings of the sixth Australian conference in CHI*. New Zealand, IEEE Computer Society Press, pp. 316-317

De Wet, L., Blignaut, P. & Burger, A. (2002). *Comprehension and usability variances among multicultural web-users in South Africa.* Development Consortium, CHI-SA/CHI 2002, Minneapolis, Minnesota, USA.

Del Galdo, E. (1996). *Culture and design: International user-interfaces*. New York: John Wiley.

Desruisseaux, P. (2000, May). Universities venture into venture capitalism. *The Chronicle of Higher Education,* A44. May 26.

Dick, B. (1999). Sources of rigour in action research: addressing the issues of trustworthiness and credibility. A paper presented at the Association for Qualitative

Research Conference "Issues of rigour in qualitative research" at the Duxton Hotel, Melbourne, Victoria, 6-10 July. Available on line at http://www.scu.edu.au/schools/gcm/ar/arp/rigour3.html

Donhardt, G. L. & Keel, D.M. (2001). The analytical data warehouse: Empowering institutional decision makers. *Educause Quarterly*, Number 4. pp. 56-58.

Dorsey, P. (2000). Data warehouses, ad hoc query tools and other ways to destroy your company. *Business intelligence / data warehousing while papers, Dulcian*. Retrieved August 18, 2005, from http://whitepapers.silicon.com/0,39024759,60006795p,00.htm

Eckerson, W. (1999, April). Data warehouse killers: Analyst Insight. *DM Review Magazine*. Retrieved August 9, 2005, from http://www.dmreview.com/article_sub.cfm?articleId=76

Eckerson, W. (2003a, November). Understanding business intelligence. *What works*. Vol 16. Retrieved March 3, 2005 from the TDWI online database: http://www.tdwi.org/research/display.aspx?ID=6838

Exkerson, W. (2003b). *Evaluating ETL and data integration platforms*, TDWI Spring conference. San Francisco.

Erdmann, M. (1997). The data warehouse as a means to support knowledge management. In (Ed.) *Proceedings of the 21st Annual German Conference on AI.* Freiburg, Germany.

Evans, C.V. (n.d). *Supporting decision making and reporting with a data warehouse / decision support initiative*. Illinois colleges and universities. Retrieved May 22, 2005, from http://www.uaps.uillinois.edu/campus_matters/Performance%20Report%202005/Effective%20Practices/UA%20Effective%20Prac/Supporting%20Decision%20Making%20and%20Reporting%20wData%20Warehouse..pdf

Furlow, G. (2001, July/Aug). *The case for building a data warehouse.* IT professional, Vol 3, pp. 31-34.

Gatziu, S, & Vavouras, A. (1999). Data warehousing: Concepts and mechanisms. *Informatik (Zeitschrift der Schweizerischem Informatikorganisationen.* Retrieved July 25, 2005 from http://www.ifi.unizh.ch/cgi-bin/db2web/Library2/article.d2w/report?lkey=GV-99

Glassey, K. (1998). Seducing the end user. *Common ACM*, 41(9), 62-69.

Gorla, A. (2003). Features to consider in a data warehousing system. *Communications of the ACM,* Volume 46, issue 11. pp. 111-115.

Gray, P., & Israel, C. (1999). The data warehouse: Industry. *Center for research on information and organizations*, University of California, Irvine.

Griffiths, S. (1995). Data warehousing – What, where, why and how. *Data Warehousing Conference*, 12-13 June 1995, Johannesburg, South Africa.

Han J., & Kamber, M. (2001). *Data mining concepts and techniques*. London: Morgan Kaufmann.

Inmon, B. (1995, February). The operational data store. *InfoDB*. Retrieved September 22, 2007, from http://www.evaltech.com/wpapers/ODS2.pdf

Inmon, B. (1999, November). Data mart does not equal data warehouse. *DM Direct*. Retrieved September 20, 2007 from http://www.dmreview.com/article_sub.cfm?articleId=1675

Kimball, R. (1996). *The data warehouse toolkit: Practical techniques for building dimensional data warehouses*. New York: John Wiley.

Kimball, R. & Ross, M. (2002). *The data warehouse toolkit. The complete guide to dimensional modeling*, 2nd edition. New York: John Wiley.

Kirpekar, H. (2005). Data warehousing: A perspective. Retrieved May 20, 2007, from http://www.objectdatalabs.com/WhitePapers/Dw-HK-WhitePaper.doc

Koch, V. J., & Fisher, L. J., (1998). Higher education and total quality management. *Total Quality Management*, Vol. 9, Number 8, pp. 659-668.

Koch, N.F., McQueen, R.J. & Scott, J.L. (1998). Can action research be made more rigorous in a positivist sense? The contribution of an interactive approach. Retrieved September, 2006 http://www.cis.temple.edu/~kock/public/jsit97/is-arw6.htm.

Lamont, J. (1999). 10 rules for successful data warehousing. *KMworld*. Retrieved 6 September, 2005 from http://www.kmworld.com/articles/printarticle.aspx?articleid=9081

Lederach, J.P. (1995). *Preparing for peace: Conflict transformation across cultures*, NY: Syracuse University Press.

Lazerson, M., Wagener, U. & Moneta, L. (2000). Like the cities they increasingly resemble, colleges must train and retain competent managers. *The Chronicle of Higher Education,* A72 July 28.

Liautaud, B. & Hammond, M. (2001). *E-business intelligence: Turning information into knowledge into profit*. *The ACM digital library*. McGraw-Hill Professional. Retrieved 10 August, 2006 from http://portal.acm.org/citation.cfm?id=557435&dl=ACM&coll=portal

List, B., Bruckner, R. M., Machaczek, K, & Schiefer, J. (2002). *A comparison of data warehouse development methodologies. Case study of the process warehouse*. Institute of Software Technology and Interactive Systems, Vienna University of Technology, 1040 Wien, Austria.

Lokken, B. (2001). Business intelligence: An intelligent move or not? *Proclarity: Enterprise Analytic Solutions.* Retrieved July 21, 2005, from http://businessintelligence.ittoolbox.com/pub/AO031202.pdf

Marcus, A. & Baumgartner, V.J. (2003). A practical set of cultural dimensions for global user-interface analysis and design. *Proceedings APCHI 2004, 6th Asia-Pacific Conference on Computer-Human Interaction*. Rotorua, New Zealand.

Marcus, A. & Gould, E. (2000). Cultural dimensions and global web user-interface design: What? So what? Now what? Retrieved August 9, 2005, From http://www.tri.sbc.com/hfweb/marcus/hfweb00_marcus.html

Marshall, P., & McKay, J. (n.d). The dual imperatives of action research. *ITP, 14, 1.* MCB University press. pp. 46-59.

McDonald, T. & Blignaut, P. (n.d). *The effect of cultural differences on the efficiency of searches on a university website*. Bloemfontein, South Africa.

Nazir, A. & McDonald, T. (2006a). Challenges in developing a cost-effective data warehouse for a tertiary institution in a developing country. Proceedings of the *7th International Conference on Data, Text and Web Mining: Business Applications and Management Information Engineering*, Prague, Czech Republic. pp. 389-397.

Nazir, A. & McDonald, T. (2006b). Lessons learned in developing a data warehouse for a tertiary institution in South Africa. Paper presented at the *2nd Foundation of Tertiary Institutions of the Northern Metropolis (FOTIM) International Quality Assurance Conference*, Pretoria, South Africa.

Nocera, J.A. (2001). Situating the interface in context: User diversity and cultural interactions. *International Workshop in Internationalization of Products and Systems*. The Open University. Milton Keynes, England.

Pendse, N. (2004). How not to buy an OLAP product. *The OLAP Report*. Retrieved August 9, 2005, from http://www.olapreport.com/How_not_to_buy.htm

Ponniah, P. (2001). *Data warehousing fundamentals*. New York: John Wiley.

Porter, J. D. & Rome, J. J. (1995). Lessons from a successful data warehouse implementation. CAUSE/EFFECT Winter. pp. 43-50

Power, D. J. (2004): A brief history of decision support systems: Retrieved July 18, 2005, from http://dssresources.com/history/dsshistory.html

Ross, M. & Kimball, R. (2005, March). Slowly changing dimensions are not always as easy as 1, 2, 3. *Intelligent Enterprise*. Retrieved April 18, 2005, from http://www.intelligententerprise.com/print_article.jhtml?articleID=59301280

SAS, BI Vendor (2005). Retrieved August 12, 2005, from http://www.sas.com/success/tshwane.html

Sekaran, U. (2000). *Research methods for business.* New York: John Wiley & Sons, Inc.

Stevenson, D. (1997). Data warehouses and executive information systems – Ignoring the hype. *European cooperation in higher education information systems (EUNIS)*, Grenoble, France. Ref: 022802

Subotzky, G. (2003). *African higher education: An international reference handbook*. Indiana University Press.

Swartz, D. & Orgill, K. (2001). Higher education ERP: Lessons learned. *Educause Quarterly,* 2.

Thomas, C. R. (1997). Information architecture: The data warehouse foundation. *Educause.* Retrieved May 10, 2005 from World Wide Web: http://www.educause.edu/ir/library/html/cem/cem97/cem9726.html

Thomas C.R., (2004). Data warehouse efforts and metadata foundations. *Educause.* Retrieved May 10, 2005 from http://www.educause.edu/ir/library/pdf/MWR0423.pdf

Trepte, K. (1997, November). Business intelligence tools. *DM Review Magazine*. Retrieved May 10, 2005, from http://www.dmreview.com/editorial/dmreview/print_action.cfm?articleId=964

Trompenaars, F. (1993). *Riding the waves of culture*. London: Nicholas Brealy.

Viljoen, M. (2006). *HEMIS systems*. HEMIS Institute, University of the Free State 8-10 May. Bloemfontein, South Africa.

Wagner, C., Cheung, K., Lee, F., & Ipiscw, R. (2003). Enhancing e-government in developing countries: Managing knowledge through virtual communities. Retrieved 01 March, 2006 from http://unpan1.un.org/intradoc/groups/public/documents/APCITY/UNPAN020328.pdf

Wally, B. (2003). *The digital age story teller*. Retrieved July 25, 2005, from http://www.bockinfo.com/docs/bifaq.htm

Wang, X. (n.d): Building data warehouses in higher education enterprises: Retrieved July 18, 2005, from http://www3.baylor.edu/~Xin_Wang/pdf/datawarehouse.pdf

Watson, H. & Haley, B. (1998). Managerial considerations. In *Communications of the ACM*, 41,(9).

Weber, P. & Weber, E. (n.d). The use and value of data warehousing in higher education. Retrieved July 18, 2005, from http://www.mountainplains.org/articles/mpa15.html

Westerman, P. (2001). *Data Warehousing. Using the Wal-Mart Model*. London: Morgan Kaufmann.

Wierschem, D., McMillen, J., & McBroom, R. (2003). What Academia can Gain from Bbuilding a Data Warehouse. Retrieved from World Wide Web 10 Jan, 2006: http://www.educause.edu/ir/library/pdf/EQM0316.pdf

Williams P. A. H., (2006). Making research real: Is action research a suitable methodology for medical information security investigation? Retrieved January 3, 2007 from http://scissec.scis.ecu.edu.au/conference_proceedings/2006/aism/Williams%20-%20Making%20Research%20Real%20-%20Is%20Action%20Research%20a%20Suitable%20Methodology%20for%20Medical%20Information%20Security%20Investigations.pdf

Winsberg, P. (n.d). Modeling the data warehouse and data mart. Retrieved September 18, 2005 from http://www.evaltech.com/wpapers/dwmodel.pdf

Zeichick, A. (2005). Developing for business intelligence. *Destination.Net.* Retrieved July 23, 2005, from World Wide Web: http://www.devx.com/SummitDays/Article/26785

Appendixes

Appendix A Student file

Element Number	Element Name	Field Name	Width	Start Column	End Column	Data type
007	Student number	STUDNUM	15	1	15	Test
008	South African identity number	SAIDBUM	15	16	30	Text
001	Qualification code	QUALCODE	15	31	45	Text
009	Qualification commencement date	COMDATE	8	46	53	Numeric
010	Entrance category	ENTRACT	1	54	54	Text
011	Date of birth	DOB	8	55	62	Numeric
012	Gender	GENDER	1	63	63	Text
013	Race	RACE	1	64	64	Text
014	Nationality	NATIONAL	3	65	67	Text
049	Citizen-resident status	CITIRES	2	68	69	Text
052	Home language	HOMELANG	4	70	73	Text
019	NSFAS status	NSFAS	2	74	75	Text
020	Institutional housing	HOUSING	4	76	79	Numeric
021	Previous year's activity	PREVACT	2	80	81	Text
022	Secondary education completion	SECED	2	82	83	Text
023	Matriculation	MATRIC	4	84	87	Numeric

	aggregate					
023	Attendance aggregate	MATRIC	4	84	87	Numeric
024	Attendance mode	ATTMODE	1	88	88	Text
025	Qualification requirement status	REQUIRE	1	89	89	Text
026	CESM category for first area of specialization	CESMSPE1	4	90	93	Text
027	CESM category for second area of specialisation	CESMSPE2	4	94	97	Text
028	CESM category for second area of specialisation	CESMSPE3	4	98	101	Text
029	CESM category for second area of specialisation	CESMSPE4	4	102	105	Text
015	Home postcode	POSTCODE	4	106	109	Text
055	Home address line 1	ADDRES1	38	110	147	Text
056	Home address line 2	ADDRES2	38	148	185	Text
016	Disability status	DISABIL	10	224	233	Text
017	Socio-economic status	SOCSTAT	2	234	235	Text
066	Student last name	LASTNAME	26	236	261	Text
067	Student first	FIRSTNAME	26	262	287	Text

	name					
068	Student middle name	MIDDNAME	26	288	313	Text
069	Postal address line 1	PSTADDR1	50	314	363	Text
070	Postal address line 2	PSTADDR2	50	364	413	Text
071	Postal address line 3	PSTADDR3	50	414	463	Text
072	Postal address postcode	PSTPCODE	4	464	467	Text
073	% research for masters qualification	RestTime	5	468	472	Numeric
	Comments	COMMENTS	30	468	497	Text

Appendix B Course registration file

Element Number	Element Name	Field Name	Width	Start Column	End Column	Data type
007	Student number	STUNUM	15	1	15	Text
001	Qualification code	QUALCODE	9	31	39	Text
030	Course code	CRSECODE	9	31	39	Text
064	Course repeat code	CRSEREPT	1	40	40	Text
018	Funding status	FUNDING	2	41	42	Text
032	Course completion status	COMPSTAT	1	43	43	Text
051	Examination-only indicator	EXAMONLY	1	44	44	Text
	Comments	COMMENTS	30	45	74	Text

Appendix C Credit file

Element Number	Element Name	Field Name	Width	Start Column	End Column	Data type
001	Qualification code	QUALCODE	15	1	15	Text

030	Course code	CRSECODE	9	16	24	Text
065	Filler1	FILLER1	1	25	25	Text
036	Course credit value	CRSECRED	8	26	33	Numeric
050	Completed research course credit value	CRSECRER	8	34	41	Numeric
	Comments	COMMENTS	30	42	71	Text

Appendix D Qualification file

Element Number	Element Name	Field Name	Width	Start Column	End Column	Data type
001	Qualification code	QUALCODE	15	1	15	Text
002	Previous years qualification code	QUALPREV	15	16	30	Text
003	Qualification name	QUALNAME	65	31	95	Text
004	Approval status	APSTATUS	1	96	96	Text
005	Qualification type	QUALTYPE	2	97	98	Text
053	Minimum time – total	MINTIMET	4	99	102	Numeric
054	Minimum time – experiential	MINTIMEX	4	103	106	Numeric
	Comments	COMMENTS	30	107	136	Text

Appendix E Qualification CESM file

Element Number	Element Name	Field Name	Width	Start Column	End Column	Data type
001	Qualification code	QUALCODE	15	1	15	Text
006	Major field CESM	CESMMAJ	4	16	19	Text
	Comments	COMMENTS	30	20	49	Text

Appendix F Course file

Element Number	Element Name	Field Name	Width	Start Column	End Column	Data type
030	Course code	CRSECODE	9	1	9	Text
065	Filler1	FILLER1	1	10	10	Text
031	Course	CRSEAPPR	1	11	11	Text

	approval status					
033	Course CESM	CESMCRSE	4	12	15	Text
034	Course level code	CLEVCODE	2	16	17	Text
059	Contact-only availability	CONMODE	1	19	19	Text
060	Distance-only availability	DISTMODE	1	19	19	Text
061	Mixed mode availability	MIXMODE	1	20	20	Text
062	Experiential training indicator	EXPTRAIN	1	21	21	Text
058	Course name	CRSENAME	65	22	86	Text
	Comments	COMMENTS	30	87	116	Text

Appendix G Staging_Table_Courses

Column Name	Properties
[Student_pKey]	[varchar] (11
[Plan_Pkey]	[varchar] (10)
[Semester_pKey]	[char] (4)
[Course_Oper_Key]	[varchar] (19)
[Catalog_Number]	[varchar] (10)
[Class_NBR]	[varchar] (8)
[Acad_Career]	[varchar] (4)
[Enrol_Status]	[char] (1)
[Enrol_Status_Reson]	[varchar] (4)
[Enrl_action_Last]	[char] (1)
[Enrol_Action_Prc_Last]	[char] (1)
[Status_Date]	[datetime]
[Enrl_Add_DT]	[datetime]
[Enrl_Drop_DT]	[datetime]
[Grading_Basis_Date]	[datetime]
[Crse_Grade_Off]	[varchar] (3)
[Crse_Grade_Input]	[varchar] (3)
[Grade_Date]	[datetime]
[STDNT_Position]	[varchar] (4)
[Include_In_Gpa]	[char] (1)
[Units_Attempted]	[char] (1)
[Manadatory_Grd_Bas]	[char] (1)
[Grade_Category]	[varchar] (4) DEFAULT ('ZZ'),
[Enrl_Req_Source]	[varchar] (2)
[Last_Upd_Enreq_Src]	[varchar] (2)
[Location]	[varchar] (10)
[Acad_Year]	[char] (4)
[Last_UPD_DT_Stmp]	[datetime]
[Last_Enrl_DT_Stmp]	[datetime]
[Session_Code]	[char] (3)
[Session_Oper_Key]	[varchar] (12)
[Department_pKey]	[varchar] (8)
[Entrance_Category]	[varchar] (4)
[Academic_Career_Sequence]	[decimal](18, 0)
[Education_Level]	[varchar] (22)
[Location_Count]	[decimal](18, 0)
[Location_Priority_Level]	[numeric](18, 0)
[Location_Calculated]	[varchar] (10)
[Attendance_Mode]	[char] (1) DEFAULT ('C'),
[Hemis_Student]	[char] (1) DEFAULT ('Y'),
[Hemis_Student_Reason]	[varchar] (60) DEFAULT (6),
[Hemis_Student_Reason_Overall]	[varchar] (60) DEFAULT (6),
[Matriculation_Status]	[char] (1) DEFAULT ('Y'),
[Plan_Year_Priority]	[char] (1) DEFAULT (2),
[Data_Validity]	[char] (2) DEFAULT ('D1'),

[Course_pKey]	[numeric](18, 0)
[Course_Flags_Key]	[char] (20)
[Course_Comp_Res_FTE]	[numeric](4, 3)
[Course_FTE]	[numeric](4, 3) DEFAULT (0),
[Course_Time_Completed_Days]	[numeric](18, 0) DEFAULT (0),
[Course_Time_Completed_Months]	[numeric](18, 0) DEFAULT (0),
[Course_Time_Completed_Descr]	[varchar] (20)
[Enrollment_DW_Count]	[numeric](18, 0) DEFAULT (0),
[Census_Date_Flag]	[char] (1) DEFAULT ('B'),
[Enrollment_Status]	[char] (1) DEFAULT ('D'),
[Course_Offer_pKey]	[numeric](18, 0)
[Year_Course]	[char] (1) DEFAULT ('N'),
[Completed_Research_Value]	[float] DEFAULT (0.0),
[Enrollment_Flags_pKey]	[numeric](18, 0)
[Year_Hemis_Student]	[char] (1) DEFAULT ('N'),
[Enrollment_Flags_Key]	[char] (8)
[Year_Census_Flag]	[char] (1) DEFAULT ('A'),
[Course_Flags_pKey]	[numeric](18, 0)
[Plan_Subsidy]	[char] (1) DEFAULT ('Y'),
[SC_Course_Marks]	[int] DEFAULT (0),
[Age_pKey]	[numeric](18, 0) DEFAULT (0),
[Year_Enrollment_Status]	[char] (1) DEFAULT ('A'),
[Old_Enrl_Date]	[datetime]
[Old_Enrollment_Plan]	[varchar] (10) DEFAULT ('Z'),
[Completion_Status]	[char] (1) DEFAULT ('U')
HM_Teach_Enrol_FTE	[numeric](18, 0)
HM_Teach_Credit_FTE	[numeric](18, 0)
HM_Teach_Weighted_FTE	[numeric](18, 0)
HM_Teach_Input_Unit	[numeric](18, 0)
HM_Teach_Input_Subsidy	[numeric](18, 0)

Appendix H Staging_Table_Enrolment

Column Name	Properties
[Student_pKey]	[varchar] (11)
[Plan_Pkey]	[varchar] (10)
[Semester_pKey]	[char] (4)
[Acad_Year]	[char] (4)
[First_Enrollment_Date]	[datetime]
[Grading_Basis_Date]	[datetime]
[Plan_Priority_Level]	[char] (2) ,
[Location]	[varchar] (10)

[Course_FTE]	[real]
[Enrollment_Status]	[char] (1) ,
[Year_Census_Flag]	[char] (1)
[Entrance_Category]	[varchar] (5) DEFAULT ('ZZZZ'),
[Academic_Career_Sequence]	[decimal](18, 0)
[Acad_Career]	[varchar] (4)
[Education_Level]	[varchar] (2)
[Plan_Year_Priority]	[char] (1) DEFAULT (2),
[Data_Validity]	[char] (2) DEFAULT ('D1'),
[Unique_Value]	**[decimal](18, 0) IDENTITY (1, 1)**
[Enrollment_Flags_pKey]	[numeric](18, 0)
[Student_Plan_Oper_Key]	[varchar] (21)
[Courses_Enrolled]	[numeric](18, 0)
[Last_Update_Stamp]	[datetime]
[Data_Mod—Ified]	[char] (1) DEFAULT ('N'),
[UFS_First_Enrollment]	[datetime]
[Age_pKey]	[numeric](18, 0)
[Birthdate]	[datetime]
[Cal_Ent]	[char] (1) DEFAULT ('Y'),
[Changed_With_Plan]	[varchar] (10) DEFAULT ('Z'),
[Last_Year_Plan]	[varchar] (10)
HM_Qualified_Research_Count	[numeric](18, 0)
HM_Qualified_No_Research_Count	[numeric](18, 0)
HM_Enrolled_FTE	[numeric](18, 0)
HM_Success_FTE	[numeric](18, 0)

Appendix I Data warehouse & metadata Email

Email to: Information Officers
From: Amer Nazir
Subject: Data Warehouse & Metadata

We are updating our paper on Data Warehouse efforts in Higher Education for a 2006 7th International Conference on Data Mining and Information Engineering, Prague and would appreciate you or someone on your campus answering six questions by return email. We will provide a summary of the results to all respondents.

Do you have a Data Warehouse effort?
[]in production []in development []planned []no plans

What data is included in your Data Warehouse?
[]Student []Financial []Human Resources []Payroll
[]Other:_____

Who is, or will be, the executive sponsor of your Data Warehouse effort?
[]Chief Executive Officer []Chief Academic Officer
[]Chief Financial Officer []Chief Information Officer
[]other:_____

Do you have a formal institution-wide Data Dictionary of Metadata?
[]yes, maintained by:
[]IT []Planning []Institutional Research []Other
[]no.

What software do you use, or plan to use, to maintain your Data Dictionary?
[]Oracle []SAS []Microsoft SQL Server
[]Other database system

What software you are using for HEMIS reporting?
[]VALPAC2 []HEDA []Other

Appendix J Questionnaire output

	All Institutions	
	All	**Percent**
Count	22	
Responded	12	
Q1: Data Warehouse Effort		
1.1 In Production	4	33.33
1.2 In Development	2	16.67
1.3 Planned	3	25
1.4 No Plans	3	25
Q2: Data Warehouse Data		
2.1 Student	4	
2.2 Financial	4	
2.3 Human Resources	3	
2.4 Payroll	2	
2.5 Other		
Q3 Executive Sponsor		
3.1 Chief Executive Officer	5	83.33
3.2 Chief Academic Officer	1	8.33
3.3 Chief Financial Officer		
3.4 Chief Information Officer		
3.5 Chief Executive Sponsor		
Q4 Data Dictionary		
4.1 Yes	6	100
4.1a Maintained by IT	6	100
4.1b Maintained by Planning		
4.1c Maintained by Institutional Research		
4.1d Maintained by other office		
4.2 In development and planned		
4.3 No data dictionary		
Q5: Data Dictionary Software		
5.1 Oracle	3	50
5.2 SAS		
5.3 Microsoft SQL Server	2	33.33
5.4 Other databases	1	16.66
Q6: HEMIS System		
6.1 VALPAC2	6	50
6.2 HEDA system	2	16.67
6.2 Other	4	33.33

Appendix K Usability testing questionnaire

Welcome to Data Warehouse Usability Study (QUIS) Questionnaire for User Interaction Satisfaction

Business intelligence (BI) depends on BI applications having access to properly prepared data. A transactional database is not well suited for BI. For BI applications, database programmers need to create data marts or DW, which are a properly formatted amalgamation of all the key enterprise data. Today's DW is no longer confined to a select group of internal users. Under present conditions, corporations need to increase the productivity of all members in the corporation's value chain. This questionnaire is designed to get personal information and views about the data warehouse (DW) usage, attempting to determine, how easy or efficient its use is?

Please choose "√" the appropriate selection.

1. Personal information

Optional

Name: _____
Contact-No: _____
e-mail: _____
Address: _____
Any other Information:_____
_____

1.1 Age ☐ 20-35 ☐ 36-45 ☐ 46-60

1.2 Gender ☐ Male ☐ Female

1.3. Designation _____
in the University

2. Computer utilization

2.1 Computer interaction:

☐ Direct (Data Entry) ☐ Indirect (Manager Reporting) ☐ Administrative (Decision Making) ☐ Other _____

2.2 Use of computer for:

☐ Reporting ☐ Word processing ☐ Web site Exploration ☐ Other _____

2.3 Which system(s) you prefer for collecting staff and students stats? (choose all that apply)

☐ ERP Applications ☐ VALPAC2 of DoE ☐ HEMIS system ☐ Support services

☐ Other _____ ☐ None of the above

Comments/remarks _____

3. Reporting experience

3.1 With the system identified in Q-2.3, how would you grade the system?

☐ Excellent ☐ Good ☐ Average ☐ Unsatisfactory ☐ Standard

3.2 Time spent on this system.

☐ <= 1 year
☐ Between 2 to 3 years
☐ > 3 years
☐ Other _____

3.3 Time that it took you to understand the system.

☐ <= 1 week
☐ Between 2 to 3 weeks
☐ > 3 weeks
☐ Other _____

3.4 How would you grade your report writing experiences?

	Very much	Some what	Neither	Some what	Very much	
Interesting						Boring
Simple						Complex
Useful						Waste of time
Helpful						Disturbing
Satisfying						Unsatisfactory
Effective						Incapable
Efficient						Inefficient
Safe						Insecure
Easy to remember						Hard to remember

Comments/remarks _____

4. Report writing

4.1 Would you prefer to be in a position of writing reports by your own?

Yes ☐ No ☐

If the answer to the question 4.1 is No, do you think the reason was?

☐ Complexity of the reporting tool
☐ Complex database structure with multiple joins
Data reliability issues
☐ Summarized data is not available
☐ Integrated information not available
☐ Other _____

4.2 Are you happy with the report writing capabilities of the system identified in Q-2.3?

☐ Always ☐ Mostly ☐ Sometimes ☐ Rarely ☐ Never

4.3 How often you would like to use PL/SQL tools for writing your own scripts?

☐ Always ☐ Mostly ☐ Sometimes ☐ Rarely ☐ Never

4.4 Are you happy from the underlying database structure of the system identified in Q-2.3?

☐ Always ☐ Mostly ☐ Sometimes ☐ Rarely ☐ Never

4.5 How often you ask information system department staff to write reports for you?

☐ Always ☐ Mostly ☐ Sometimes ☐ Rarely ☐ Never

4.6 Are you agreed that the current available systems can answer all reporting requirements for student and staff stats?

☐ Always ☐ Mostly ☐ Sometimes ☐ Rarely ☐ Never
Comments/remarks _____

5. Data warehouse (DW) reporting experience

5.1 Please give your experiences while report writing from DW.

		Very much	Some what	Neither	Some what	Very much	
Database structure							
	Simple						Complex
DW data content	Reliable						Unreliable
Data summarisation	Useful						Waste of time
Reporting interface	Helpful						Disturbing
	Satisfying						Unsatisfactory
	Effective						Incapable
	Efficient						Inefficient
	Secure						Insecure

5.2 Do you think that DW reporting environment has made you self sufficient in querying and writing your own reports?

☐ Always ☐ Mostly ☐ Sometimes ☐ Rarely ☐ Never

5.3 How much support does technical manuals and online help provides?

☐ Full support ☐ Some what ☐ Satisfying ☐ Confusing ☐ Not provided

Comments/remarks _____

Please use back side of the page to provide comments/remarks on any other information that in your opinion would be useful in the DW evaluation.
Thanks for your participation

Appendix L Usability testing results

	Univ	%	AISU	%	BI user group		Head of schools	%	Portfolio executives	%
Count	**39**		**6**		**15**		**6**		**12**	
Q 2.1: Computer Interaction										
2.1.1 Direct (data entry)										
2.1.2 Indirect (Manager reporting)	16	41.0	6	100.0	10	66.7				
2.1.3 Administrative (Decision making)	23	59.0			5	33.3	6	100.0	12	100.0
2.1.4 Other	0									
	0									
Q 2.2 Use of computer for (choose all that apply)	0									
2.2.1 Reporting	33	84.6	6	100.0	15	100.0	6	100.0	6	50.0
2.2.2 Word processing	28	71.8	6	100.0	10	66.7	6	100.0	6	50.0
2.2.3 Web site exploration	7	17.9			5	33.3	2	33.3		
2.2.4 Other	6		2	33.3					4	33.3
	0									
Q 2.3 Which system(s) you prefer for collecting staff and students stats?	0									
2.3.1 ERP applications	26	66.7	6	100.0	15	100.0			5	
2.3.2 VALPAC2 of DoE	2	5.1			2	13.3				
2.3.3 HEMIS System	2	5.1			2	13.3				
2.3.4 Support services	24	61.5	3	50.0	3	20.0	6	100.0	12	100.0
2.3.5 Other	0									
2.3.6 None of	0	0.0								

above										
	0									
Q 3.1 With the system identified in Q 2.3, how would you grade the system?	0									
3.1.1 Excellent	0									
3.1.2 Good	13	33.3	2	33.3	6	40.0	2	33.3	3	25.0
3.1.3 Average	16	41.0	4	66.7	6	40.0	4	66.7	2	16.7
3.1.4 Unsatisfactory	2	5.1			2	13.3		0.0		0.0
3.1.4 Standard	0									
	0									
Q.3.2 Time spent on this system	0									
3.2.1 <= 1 year	16	41.0	2	33.3	9	60.0			5	41.7
3.2.2 Between 2 to 3 years	8	20.5	4	66.7	4	26.7				0.0
3.2.3 > 3 years	2	5.1			2	13.3				
3.2.4 Other	0									
	0									
Q 3.3 Time that it took you to understand the system	0									
3.1.1 <= 1 week	8	20.5	4	66.7	2	13.3			2	
3.1.2 Between 2 to 3 years	14	35.9	2	33.3	10	66.7			2	16.7
3.1.3 > 3 weeks	4	10.3			3	20.0			1	
3.1.4 Other	0									
	0									
Q 3.4 Please give your opinion on working with the system	0									
3.4.1a Interesting	11	28.2	2	33.3	2	13.3	4	66.7	3	25.0
3.4.1b Boring	18	46.2	4	66.7	10	66.7	2	33.3	2	16.7
3.4.2a Simple	10	25.6	2	33.3	3	20.0	3	50.0	2	16.7

3.4.2b Complex	22	56.4	4	66.7	12	80.0	3	50.0	3	25.0
3.4.3a Useful	14	35.9	5	83.3	4	26.7	3	50.0	2	16.7
3.4.3b Waste of time	13	33.3	1	16.7	6	40.0	3	50.0	3	25.0
3.4.4a Helpful	10	25.6	3	50.0	3	20.0	2	33.3	2	16.7
3.4.4b Disturbing	17	43.6	3	50.0	7	46.7	4	66.7	3	25.0
3.4.5a Satisfying	12	30.8	4	66.7	4	26.7	2	33.3	2	16.7
3.4.5b Unsatisfactory	20	51.3	2	33.3	11	73.3	4	66.7	3	25.0
3.4.6a Effective	12	30.8	3	50.0	5	33.3	2	33.3	2	16.7
3.4.6b Incapable	19	48.7	3	50.0	9	60.0	4	66.7	3	25.0
3.4.7a Efficient	11	28.2	3	50.0	3	20.0	3	50.0	2	16.7
3.4.7b Inefficient	19	48.7	3	50.0	10	66.7	3	50.0	3	25.0
3.4.8a Safe	27	69.2	6	100.0	12	80.0	5	83.3	4	33.3
3.4.8b Insecure	5	12.8			3	20.0	1	16.7	1	8.3
3.4.9a Easy to remember	13	33.3	3	50.0	5	33.3	3	50.0	2	16.7
3.4.9b Hard to remember	19	48.7	3	50.0	10	66.7	3	50.0	3	25.0
	0									
Q 4.1 Would you prefer to be in a position of writing reports by your own?	0									
4.1.1 Yes	26	66.7	6	100.0	15	100.0	2	33.3	3	25.0
4.1.1 No	13	33.3					4	66.7	9	75.0
	0									
Q 4.2 If the answer to the question 4.1 is No, do you think the reason was (choose all that apply)	0									
4.2.1 Complexity of the reporting	3	7.7					2	33.3	1	8.3

tool										
4.2.2 Complex database structure with multiple joins	5	12.8					3	50.0	2	16.7
4.2.3 Data reliability issues	3	7.7					3	50.0		
4.2.4 Summarized data is not available	2	5.1					2	33.3		
4.2.5 Integrated information not available	10	25.6					4	66.7	6	50.0
4.2.6 Other	0									
	0									
Q 4.3 Are you happy with the report writing capabilities of the VALPAC system?	0									
4.3.1 Always	1	2.6			1	6.7				
4.3.2 Mostly	7	17.9	1	16.7	5	33.3			1	8.3
4.3.3 Sometimes	14	35.9	3	50.0	9	60.0			2	16.7
4.3.4 Rarely	2	5.1	2	33.3						
4.3.5 Never	6	15.4					6	100.0		
Q 4.4 How often you would like to use PL/SQL tools for writing your own scripts?	0									
4.4.1 Always	0									
4.4.2 Mostly	0									
4.4.3 Sometimes	2	5.1			2	13.3				
4.4.4 Rarely	3	7.7	2	33.3	1	6.7				
4.4.5 Never	21	53.8			12	80.0	6	100.0	3	25.0
	4		4	66.7						
Q 4.5 Are you happy with the	27									

underlying database structure of the VALPAC system?										
4.5.1 Always	0									
4.5.2 Mostly	0									
4.5.3 Sometimes	1	3.7			1	6.7				
4.5.4 Rarely	4	14.8	2	33.3	2	13.3				
4.5.5 Never	22	81.5	4	66.7	12	80.0	6	100.0		
	0									
Q 4.6 How often you ask information system department staff to write reports for you?	0									
4.6.1 Always	15	38.5					6	100.0	9	75.0
4.6.2 Mostly	11	28.2			8	53.3			3	25.0
4.6.3 Sometimes	9	23.1	5	83.3	4	26.7				
4.6.4 Rarely	3	7.7	1	16.7	2	13.3				
4.6.5 Never	1	2.6			1	6.7				
	0	0.0				0.0				
Q 4.7 Do you agree that the current available systems can answer all reporting requirements for student and staff statistics?	0									
4.6.1 Always	0									
4.6.2 Mostly	4	10.3			4	26.7		0.0		0.0
4.6.3 Sometimes	17	43.6	3	50.0	6	40.0	3	50.0	5	41.7
4.6.4 Rarely	15	38.5	1	16.7	5	33.3	2		7	58.3
4.6.5 Never	4	10.3	2	33.3			2	33.3		
	0									
Q 5.1 Please give your report-writing	0									

experiences regarding the SDM.										
5.1.1 Database structure	0									
5.1.1a Simple	30	76.9	6	100.0	15	100.0	6	100.0	3	25.0
5.1.1b Complex	0									
5.1.2 DW data content	0									
5.1.2a Reliable	31	79.5	4	66.7	12	80.0	5	83.3	10	83.3
5.1.2b Unreliable	8	20.5	2	33.3	3	20.0	1	16.7	2	16.7
5.1.3 Data summarisation	0									
5.1.3a Useful	37	94.9	6	100.0	15	100.0	6	100.0	10	83.3
5.1.3b waste of time	2	5.1				0.0		0.0	2	16.7
5.1.4 Reporting interface	0									
5.1.4a Helpful	28	71.8	5	83.3	15	100.0	5	83.3	3	25.0
5.1.4b Disturbing	3	7.7	1	16.7	1	6.7	1	16.7		0.0
5.1.4c Satisfying	26	66.7	5	83.3	13	86.7	5	83.3	3	25.0
5.1.4d Unsatisfactory	4	10.3	1	16.7	2	13.3	1	16.7		0.0
5.1.4e Effective	27	69.2	5	83.3	14	93.3	5	83.3	3	25.0
5.1.4f Incapable	3	7.7	1	16.7	1	6.7	1	16.7		0.0
5.1.4g Efficient	27	69.2	6	100.0	14	93.3	4	66.7	3	25.0
5.1.4h Inefficient	3	7.7			1	6.7	2	33.3		0.0
5.4.1i Secure	27	69.2	6	100.0	13	86.7	5	83.3	3	25.0
5.4.1j Insecure	3	7.7			2	13.3	1	16.7		0.0
Q 5.2 Do you think that DW reporting environment has made you self sufficient in querying and writing										

your own reports?										
5.2.1 Always	16	41.0	6	100.0	2	13.3	5	83.3	3	25.0
5.2.2 Mostly	23	59.0			12	80.0	1	16.7	10	83.3
5.2.3 Sometimes	2	5.1			1	6.7		0.0	1	8.3
5.2.4 Rarely										
5.2.5 Never										
Q 5.3 How much support does technical manuals and online help provides?										
5.3.1 Full support	7	17.9	2	33.3	3	20.0	1	16.7	1	8.3
5.3.2 Some what	16	41.0	4	66.7	5	33.3	5	83.3	2	16.7

Appendix M Program student address

This procedure was written to clean student address information that was wrongly captured at ERP side.

```
CREATE proc [dbo].[P_Student_Complete_Address]
        @vchStudentCode varchar(11)
as

        declare cursorStudent CURSOR for
        SELECT Address_Type,Address1,Address2,
        ltrim(Address3),ltrim(Address4),ltrim(City),Postal,Country
        FROM STAGING_AREA_ADDRESS
        WHERE STUDENT_PKEY =  @vchStudentCode

        ----- OPen cursor to get the recordset
        open cursorStudent

while 1 = 1

   begin
        declare @vchAddress_Type varchar(4)
        declare @vchAddress1 varchar(55)
        declare @vchAddress2 varchar(55)
        declare @vchAddress3 varchar(55)
        declare @vchAddress4 varchar(55)
        declare @vchCity     varchar(55)
        declare @vchCity_found     varchar(55)
        declare @vchCountry  varchar(30)
        declare @vchPostal   varchar(12)
        declare @vchCompleteAddress varchar(2000)



      fetch next from cursorStudent into @vchAddress_Type,@vchAddress1,
        @vchAddress2,@vchAddress3,@vchAddress4,@vchCity,@vchPostal,@vchC
ountry

      if @@fetch_status <> 0
         break
        --------- Concatinate Address fields in One field
        SELECT @vchCompleteAddress =  @vchAddress1

        IF @vchAddress1 is NULL
                        SELECT @vchCompleteAddress = @vchAddress2
        ELSE IF @vchAddress2 is NOT NULL
                SELECT @vchCompleteAddress = @vchCompleteAddress + " " +
@vchAddress2
```

```
	IF @vchAddress2 is NULL
		SELECT @vchCompleteAddress = @vchAddress3
	ELSE IF @vchAddress3 is NOT NULL
		SELECT @vchCompleteAddress = @vchCompleteAddress + " " +
@vchAddress3


	IF @vchAddress3 is NULL
		SELECT @vchCompleteAddress = @vchAddress4
	ELSE IF @vchAddress4 is NOT NULL
		SELECT @vchCompleteAddress = @vchCompleteAddress + " " +
@vchAddress4

	IF @vchCity is NOT NULL
		SELECT @vchCompleteAddress = @vchCompleteAddress + " " +
@vchCity

	IF @vchPostal is NOT NULL
		SELECT @vchCompleteAddress = @vchCompleteAddress + " " +
LTRIM(@vchPostal)

	IF @vchCountry is NOT NULL
		SELECT @vchCompleteAddress = @vchCompleteAddress + " " +
@vchCountry

	----- Update the Student Dimension Table Location Field
	IF @VCHADDRESS_TYPE = "CAMP"
		UPDATE UFS_WH.dbo.STUDENT
		SET S_Campus_Address_Complete = @vchCompleteAddress
		where Student_Pkey = @vchStudentCode
	ELSE IF @VCHADDRESS_TYPE = "HOME"
		begin
			----- Home address individual entries
			If Len(@vchAddress4) = 0
				BEGIN
					IF Len(@vchAddress3) = 0
						SELECT @vchCity_Found = @vchCity
					ELSE
						SELECT @vchCity_Found =
@vchAddress3
				END
			ELSE IF Len(@vchCity) = 0
				BEGIN
					IF Len(@vchAddress3) = 0
						SELECT @vchCity_Found =
@vchAddress4
					ELSE
						SELECT @vchCity_Found =
@vchAddress3
				END
```

```
                        ELSE IF Len(@vchAddress3) = 0
                                BEGIN
                                        IF Len(@vchAddress4) = 0
                                                SELECT @vchCity_Found = @vchCity
                                        ELSE
                                                SELECT @vchCity_Found =
@vchAddress4
                                        END

                        -------- UPDATE INDIVIDUALS MAILING COLUMNS
                        UPDATE UFS_WH.dbo.STUDENT
                        SET     S_Home_Street_Address = @vchAddress1,
                                S_Home_Suburb = @vchAddress2,
                                S_Home_City = @vchCity_Found,
                                S_Home_Country = @vchCountry,
                                S_Home_Post_Code = @vchPostal,
                                S_Home_Address_Complete = @vchCompleteAddress
                        where Student_Pkey = @vchStudentCode
                end
        ELSE IF @VCHADDRESS_TYPE = "MAIL"
      BEGIN

                        ----- Find out the city in different columns
                        ----- because operators are entering in Address 3 Address 4 or
City
                        ----- Mailing address
                        If Len(@vchAddress4) = 0
                                BEGIN
                                        IF Len(@vchAddress3) = 0
                                                SELECT @vchCity_Found = @vchCity
                                        ELSE
                                                SELECT @vchCity_Found =
@vchAddress3
                                END
                        ELSE IF Len(@vchCity) = 0
                                BEGIN
                                        IF Len(@vchAddress3) = 0
                                                SELECT @vchCity_Found =
@vchAddress4
                                        ELSE
                                                SELECT @vchCity_Found =
@vchAddress3
                                END

                        ELSE IF Len(@vchAddress3) = 0
                                BEGIN
                                        IF Len(@vchAddress4) = 0
                                                SELECT @vchCity_Found = @vchCity
                                        ELSE
```

```
                                        SELECT @vchCity_Found =
@vchAddress4
                                    END

                        -------- UPDATE INDIVIDUALS MAILING COLUMNS
                        UPDATE UFS_WH.dbo.STUDENT
                        SET     S_Mailing_Street_Address = @vchAddress1,
                                S_Mailing_Suburb = @vchAddress2,
                                S_Mailing_City = @vchCity_Found,
                                S_Mailing_Country = @vchCountry,
                                S_Mailing_Post_Code = @vchPostal,
                                S_Mailing_Address_Complete =
@vchCompleteAddress
                        where Student_Pkey = @vchStudentCode
                END
        ELSE IF @VCHADDRESS_TYPE = "WORK"
                UPDATE UFS_WH.dbo.STUDENT
                SET S_Work_Address_Complete = @vchCompleteAddress
                where Student_Pkey = @vchStudentCode



END
------- Close the Inner cursor
close cursorStudent
deallocate cursorStudent
```

Appendix N Program student nationality

This procedure is written to map ERP nationality with the nationalities provided by DoE. Program also figures out the primary Nationality of the student.

```
CREATE proc [dbo].[P_Student_Nationality]
        @vchStudentCode varchar(11)
as

        declare @vchCountry_UP varchar(3)
        declare @vchCountryName_UP varchar(30)
        declare @vchNationalIDType_UP varchar(6)
        declare @vchNationalID_UP varchar(20)
        declare @vchPrimaryNID_UP varchar(1)
        declare @vchCountryShort_UP varchar(2)
        declare @vchEUMemberState_UP varchar(1)
        declare @vchResidentStatus varchar(2)
        declare @vchPassportNO varchar(20)
        declare @vchOtherID varchar(20)
        -------- Derived Variables
        declare @vchNationalityHEMISCode Char(3)
        declare @vchNationalityHEMISCode_UP char(3)
        declare @vchSADCCountry Varchar(9)

        declare @nCountID integer
        Select @nCountID = 0
        declare cursorStudent CURSOR for
        SELECT Country,Country_Name,National_ID_Type,
        National_ID,Primary_NID,Country_Short,EU_Member_State,National_Hemi
s
        FROM STAGING_AREA_STUDENT_NID
        WHERE STUDENT_PKEY =  @vchStudentCode

        ----- OPen cursor to get the recordset
        open cursorStudent

while 1 = 1

   begin
        declare @vchCountry varchar(3)
        declare @vchCountryName varchar(30)
        declare @vchNationalIDType varchar(6)
        declare @vchNationalID varchar(20)
        declare @vchPrimaryNID varchar(1)
        declare @vchCountryShort varchar(2)
        declare @vchEUMemberState varchar(1)

        Select @nCountId = @nCountID + 1

      fetch next from cursorStudent into @vchCountry,@vchCountryName,
```

```
        @vchNationalIDType,@vchNationalID,@vchPrimaryNID,@vchCountryShor
t,
        @vchEUMemberState,@vchNationalityHEMISCode_UP

      if @@fetch_status <> 0
         break

        SELECT @vchResidentStatus = "ZZ" --- No Information

        ----- Update the Student Dimension Table Location Field
        IF @vchPrimaryNID = "Y" and @nCountID = 1
                BEGIN
                        SELECT @vchCountry_UP = @vchCountry
                        SELECT @vchCountryName_UP = @vchCountryName
                        SELECT @vchNationalIDType_UP = @vchNationalIDType
                        SELECT @vchNationalID_UP = @vchNationalID
                        SELECT @vchPrimaryNID_UP = @vchPrimaryNID
                        SELECT @vchCountryShort_UP = @vchCountryShort
                        SELECT @vchEUMemberState_UP = @vchEUMemberState
                        ---- Check either the current student is SA National or
                        ---- just residing in SA
                        IF @vchNationalID is NOT NULL
                                If Len(@vchNationalID) > 2
                                        SELECT @vchResidentStatus = "PR"
                                ELSE
                                        SELECT @vchResidentStatus = "SA"
                        ELSE
                                SELECT @vchResidentStatus = "SA"
                END
        ELSE
                BEGIN
                        ----- Check student other IDs
                        IF @vchNationalIDType = "PASSP"
                                SELECT @vchPassportNO = @vchNationalID

                        ELSE
                                SELECT @vchOtherID = @vchNationalID

                END

END
------- Close the Inner cursor
close cursorStudent
deallocate cursorStudent

----- Get HEMIS Nationality Code
SELECT @vchSADCCountry = "SADC"     --- South African Democratic Countries
Alliance
IF @vchCountry = "ZAF"
        SELECT @vchNationalityHEMISCode = "SAF"
```

```
ELSE IF @vchCountry = "NAM"
	SELECT @vchNationalityHEMISCode = "NAM"
ELSE IF @vchCountry = "BNA"
	SELECT @vchNationalityHEMISCode = "BOT"
ELSE IF @vchCountry = "ZWE"
	SELECT @vchNationalityHEMISCode = "ZIM"
ELSE IF @vchCountry = "AGO"
	SELECT @vchNationalityHEMISCode = "ANG"
ELSE IF @vchCountry = "MOZ"
	SELECT @vchNationalityHEMISCode = "MOZ"
ELSE IF @vchCountry = "LSO"
	SELECT @vchNationalityHEMISCode = "LES"
ELSE IF @vchCountry = "SWZ"
	SELECT @vchNationalityHEMISCode = "SWA"
ELSE IF @vchCountry = "MWI"
	SELECT @vchNationalityHEMISCode = "MAL"
ELSE IF @vchCountry = "ZMB"
	SELECT @vchNationalityHEMISCode = "ZAM"
ELSE IF @vchCountry = "MUS"
	SELECT @vchNationalityHEMISCode = "MAU"
ELSE IF @vchCountry = "TZA"
	SELECT @vchNationalityHEMISCode = "TAN"
ELSE IF @vchCountry = "COD"
	SELECT @vchNationalityHEMISCode = "DEM"
ELSE IF @vchCountry = "SYC"
	SELECT @vchNationalityHEMISCode = "SEY"
ELSE
	BEGIN
		SELECT @vchNationalityHEMISCode = "ZZZ"
		SELECT @vchSADCCountry = "NOT SADC"
	END

----- Forward the query tp update information in the STUDENT table
UPDATE UFS_WH.dbo.STUDENT
SET	S_Nationality_HEMIS_Code = @vchNationalityHEMISCode_UP,
	S_Nationality_UFS_Country_Code = @vchCountry,
	S_Nationality_2Char_Code = @vchCountryShort,
	S_Nationality_Country_Descr = @vchCountryName,
	S_Home_Country_EU_Member = @vchEUMemberState,
	S_National_ID = @vchNationalID,
	S_Passport_NO = @vchPassportNO,
	S_Other_ID = @vchOtherID,
	S_Citizen_Resident_Status = @vchResidentStatus,
	S_SADC_Member_Country = @vchSADCCountry
where Student_Pkey = @vchStudentCode

END
```

Appendix O Program native language

```
---- This procedure will fetch all the unique Student Keys
---- and then call the procedure P_Student_Languages
CREATE proc [dbo].[P_Student_Languages_Main]
as

   declare cursorStudentMain CURSOR for
         SELECT  STUDENT_PKEY
         FROM STUDENT
--       WHERE STUDENT_PKEY = "2004027864"
--       WHERE ACAD_YEAR BETWEEN 1999
--       AND 2005
   open cursorStudentMain

while 1 = 1
   begin
         declare @vchStudentCode varchar(11)

      fetch next from cursorStudentMain into @vchStudentCode
      if @@fetch_status <> 0
          break

         EXEC dbo.P_Student_Languages     @vchStudentCode

   end
Close cursorStudentMain
Deallocate cursorStudentMain

CREATE proc [dbo].[P_Student_Languages]
         @vchStudentCode varchar(11)
as
         Declare @nOtherCount integer

         declare cursorStudent CURSOR for
         SELECT Accomplishment_Descr,Native_Language
         FROM UVS_WH_STAGING.DBO.STAGING_AREA_ACCOMP
         WHERE STUDENT_PKEY =  @vchStudentCode
         Order BY Native_Language desc

         ----- OPen cursor to get the recordset
         open cursorStudent

         SELECT @nOtherCount = 0

while 1 = 1

   begin

         SELECT @nOtherCount = @nOtherCount + 1
```

```
        declare @vchAccomplishmentDescr varchar(20)
        declare @vchNativeLanguage char(1)

      fetch next from cursorStudent into
@vchAccomplishmentDescr,@vchNativeLanguage

      if @@fetch_status <> 0
         break

        ------ Declaration of second cursor to get count of locations in a semester

                IF @vchNativeLanguage = "Y" and @notherCount = 1
                        ----- Update the Staging_Area_Enrolment Table Location Field
                        UPDATE STUDENT
                        SET S_Native_Language = @vchAccomplishmentDescr
                        where Student_Pkey = @vchStudentCode
                ELSE IF @notherCount = 2
                        ----- Update Other Language 1
                        UPDATE STUDENT
                        SET S_Other_Language1 = @vchAccomplishmentDescr
                        where Student_Pkey = @vchStudentCode
                ELSE IF @notherCount = 3
                        ----- Update Other Language 2
                        UPDATE STUDENT
                        SET S_Other_Language1 = @vchAccomplishmentDescr
                        where Student_Pkey = @vchStudentCode


    END
------- Close the Inner cursor
close cursorStudent
deallocate cursorStudent



'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM dbo.sysobjects WHERE id =
OBJECT_ID(N'[dbo].[P_Student_Languages_Main]') AND
OBJECTPROPERTY(id,N'IsProcedure') = 1)
BEGIN
EXEC dbo.sp_executesql @statement = N'
```

Appendix P Program primary location

The purpose of this procedure is to figure out student primary location if a student is taking his courses from UFS from different campuses.

```
CREATE proc [dbo].[P_Primary_Location]
        @vchStudentCode varchar(11)
as

        declare cursorStudent CURSOR for
        SELECT Plan_PKey,Acad_Year
        FROM STAGING_AREA_ENROL
        WHERE STUDENT_PKEY =  @vchStudentCode
        Group BY Student_pKey,Plan_Pkey,Acad_Year

        ----- OPen cursor to get the recordset
        open cursorStudent

while 1 = 1

   begin
        declare @vchAcademicYear char(4)
        declare @vchAcademicPlan varchar(10)
        declare @vchSemester char(4)
        declare @vchLocation char(10)
        declare @vchLocationPriority char(2)
        declare @vchEnrollmentStatus varchar(4)
        declare @vchEnrolStatus varchar(10)
        declare @iCount int
   fetch next from cursorStudent into @vchAcademicPlan,@vchAcademicYear

                if @@fetch_status <> 0
         break

        ------ Declaration of second cursor to get count of locations in a semester
                declare cursorStudentLocation CURSOR for
                SELECT DISTINCT LOCATION,Location_Priority_Level
                FROM STAGING_Area_Enrol
                where Student_Pkey = @vchStudentCode
                and Plan_Pkey = @vchAcademicPlan
                and Acad_Year = @vchAcademicYear
                AND LOCATION_COUNT = (SELECT MAX(DISTINCT
LOCATION_COUNT)
                                FROM STAGING_AREA_ENROL
                                where Student_Pkey = @vchStudentCode
                                and Plan_Pkey = @vchAcademicPlan
                                and Acad_Year = @vchAcademicYear)
                Order By Location_Priority_Level

                ----- OPen cursor to get the recordset
                open cursorStudentLocation
```

```
                fetch next from cursorStudentLocation into
@vchLocation,@vchLocationPriority


                ----- Declaration of the third cursor to get Enrollment Status of
        ----- the student registration. Either student has drop/withdrawn all the courses
for
                ----- he/she was registered
                declare cursorStudentEnrollment CURSOR for
                SELECT DISTINCT
Enrol_Status_Reson,ENROLLMENT_DW_COUNT
                FROM STAGING_AREA_ENROL
                WHERE STUDENT_PKEY =  @vchStudentCode
                and Plan_Pkey = @vchAcademicPlan
                and Acad_Year = @vchAcademicYear
                and YEAR_ENROLLMENT_STATUS = "A"
                and Enrollment_DW_Count =
        (Select Max(B.Enrollment_DW_Count)
         from STAGING_AREA_ENROL B
                 WHERE STUDENT_PKEY =  @vchStudentCode
                 and Plan_Pkey = @vchAcademicPlan
                 and Acad_Year = @vchAcademicYear
                 and YEAR_ENROLLMENT_STATUS = "A")
                ORDER BY ENROLLMENT_DW_COUNT DESC
                ----- OPen cursor to get the recordset
                open cursorStudentEnrollment

                fetch next from cursorStudentEnrollment into @vchEnrollmentStatus,
                @iCount
                  if @@fetch_status <> 0
                                ----- Update the Staging_Area_Enrol Table Location
Field
                                UPDATE STAGING_AREA_ENROL
                                SET LOCATION_CALCULATED = @vchLocation
                                where Student_Pkey = @vchStudentCode
                                and Plan_Pkey = @vchAcademicPlan
                                and Acad_Year = @vchAcademicYear

                        ELSE
                                ----- Update the Staging_Area_Enrol Table Location
Field
                                begin
                                        if @vchEnrollmentStatus = "DROP"
                                                SELECT @vchEnrolStatus = "D"
                                        Else if @vchEnrollmentStatus = "WDRW"
                                                SELECT @vchEnrolStatus = "W"
                                        ELSE IF @vchEnrollmentStatus = "ACKG"
```

```
                                        SELECT @vchEnrolStatus = "K"

                                    ELSE
                                        SELECT @vchEnrolStatus = "A"

                                    UPDATE STAGING_AREA_ENROL
                                    SET LOCATION_Calculated = @vchLocation,
                                        YEAR_Enrollment_Status =
@vchEnrolStatus
                                    where Student_Pkey = @vchStudentCode
                                    and Plan_Pkey = @vchAcademicPlan
                                    and Acad_Year = @vchAcademicYear
                                end


                ------ Close the Location Cursor
                close cursorStudentLocation
                deallocate cursorStudentLocation

                close cursorStudentEnrollment
                deallocate cursorStudentEnrollment
        END
------- Close the Inner cursor
close cursorStudent
deallocate cursorStudent
```

Appendix Q Program plan priority level

```
CREATE proc [dbo].[P_Plan_Priority]
        @vchStudentCode varchar(11)
as

        declare @vchAcademicYearPrev char(4)
        declare @vchSemesterPrev char(4)

        declare cursorStudent CURSOR for
        SELECT DISTINCT Acad_Year
        FROM STAGING_AREA_ENROLMENT
        WHERE STUDENT_PKEY =  @vchStudentCode
        Order BY Acad_Year


        ----- OPen cursor to get the recordset
        open cursorStudent


while 1 = 1

   begin
        declare @vchAcademicYear char(4)
        declare @vchAcademicPlan varchar(10)
        declare @vchAcademicPlanPrev varchar(10)
        declare @vchSemester char(4)
        declare @vchPlanPriority char(2)
        declare @nYearPlanCount integer

        SELECT @nYearPlanCount = 0

      fetch next from cursorStudent into @vchAcademicYear

      if @@fetch_status <> 0
          break

        declare cursorStudentYearPlan CURSOR for
        SELECT DISTINCT Plan_Pkey,Plan_Priority_Level,SEMESTER_PKEY
        FROM STAGING_AREA_ENROLMENT
        WHERE STUDENT_PKEY =  @vchStudentCode
        AND Acad_Year = @vchAcademicYear
        Order By SEMESTER_PKEY ASC,Plan_Priority_Level DESC

        ----- OPen cursor to get the recordset
        open cursorStudentYearPlan

        WHILE @@Fetch_Status = 0
        BEGIN
                SELECT @nYearPlanCount = @nYearPlanCount + 1
```

```
            fetch next from cursorStudentYearPlan into
@vchAcademicPlan,@vchPlanPriority,@vchSemester
                IF @@Fetch_Status = 0
                        UPDATE STAGING_AREA_ENROLMENT
                        SET     Plan_Year_Priority = @nYearPlanCount
                        where Student_Pkey = @vchStudentCode
                        and Acad_Year = @vchAcademicYear
                        and Plan_Pkey = @vchAcademicPlan
        END
        close cursorStudentYearPlan
        Deallocate cursorStudentYearPlan


END
------- Close the Inner cursor
close cursorStudent
deallocate cursorStudent


'
END
```

Appendix R Program HEMIS subsidy status

```
----- This procedure set the Flag field of student
----- HEMIS status to N. If student have the following
----- Mising information
/*      1. Student does not have matriculation record in Staging_Area_Extension file
        2. If student have his entries with U* he can be considered for Subsidy
        3. If student matriculation records found with Honours_Category value
                100 = No. Infor found in PS
        4. If the student have the following Grade_Categories he is also not considered
                for subsidies
                a.      00 - Not Applicable
                        00 - Not applicable appears if a student enrolled in the current
year
                        and have not yet pass the module. IF a module have still 00
                        and student enrolled last year and and he is not
                        a research student. He willnot consider for subsidies.
                b.      05 - Incomplete
                c.      09 - Incomplete
                d.      20 - Discontinued
        5. If a plan is not for s subsidies the plan will not be submitted for subsidies.
        6. If student taking courses and are not taking classes they are from the
                grade category 21 and 68 and will not be considered for subsidies
        7.
        --Other Codes used for updateion
        A1 = Meet matriculation prerequisites
        A2 = Plan is not subsidised
        A3 = Discountinued/Incomplete/Not Applicable
        A4 = Sucessfully completed the module
        A5 = Enrollment After census date
        A6 = Dropped/Withdrawn Before census date
        A7 = Exemptions
        A8 = Unknown
        A9 = Missing grade category
        AA = Zero-FTE Value
        AB = Non-Primary plan
        AC = Unknown Plan
*/

CREATE proc [dbo].[P_Student_Subsidy_Status]
        @vchStudentCode varchar(11),
        @vchAcademicYearCurrent char(4)
as

        declare @cEntranceCategory Char(2)
        declare @iRowCount integer
        SELECT @iRowCount = 0

        declare cursorStudent CURSOR for
        SELECT  A.Course_Pkey, A.Semester_pKey, A.Grade_Category,
           A.Plan_Subsidy,A.Acad_Year,
```

```
                Substring(A.Entrance_Category,2,1) as FirstTime,
                A.Enrol_Status_Reson,A.Census_Date_Flag,Enrol_Status,
                A.Course_FTE,A.Plan_Year_Priority,A.PLAN_PKEY
        FROM dbo.STAGING_AREA_ENROL A
        WHERE       A.Student_Pkey = @vchStudentCode
        AND   A.Acad_Year = @vchAcademicYearCurrent

--      AND   A.Grade_Category NOT IN ("21", "68")
--      AND   A.Enrol_Status = "E"
--      AND A.Enrol_Status_Reson <> "DROP"

--- Use Enrollment_Status to see enrolled students

        ----- OPen cursor to get the recordset
        open cursorStudent
        ------- Itetrate for all courses
        WHILE 1 = 1

                BEGIN
                        declare @vchCourse varchar(15)
                        declare @vchSemester char(4)
                        declare @vchGradeCategory varchar(4)
                        declare @vchAcademicYear char(4)
                        declare @vchEnrlStatusReson varchar(4)
                        declare @cCensusDateFlag char(1)
                        declare @cSubsidyStatus char(1)
                        declare @cHemisStudentReason char(2)
                        declare @cHemisStudentReasonOverall char(2)
                        declare @cHemisStudent char(1)
                        declare @cEnrolStatus char(1)
                        declare @rCourseFTE real
                        declare @iPrimaryPlan CHAR(1)
                        declare @vchPlanKey varchar(10)
                        ---- Default Values
                        --SELECT  @cHemisStudentReason = "Z"
                        --SELECT  @cHemisStudentReasonOverall = "Z"

                    fetch next from cursorStudent into
@vchCourse,@vchSemester,@vchGradeCategory,
                        @cSubsidyStatus,@vchAcademicYear,@cEntranceCategory,

        @vchEnrlStatusReson,@cCensusDateFlag,@cEnrolStatus,@rCourseFTE,
                        @iPrimaryPlan,@vchPlanKey

                        If @@Fetch_Status <> 0
                                BREAK

                        IF @@FETCH_STATUS = 0
                                BEGIN
                                        declare @cMatriculationStatus Char(1)
```

```
                                        SELECT @cMatriculationStatus = "Y"

                                        IF @cEntranceCategory = "U" OR
                                                @cEntranceCategory = "Z"
                                                ----- If the students are first time entering
student
                                                ----- check their matriculation. Either it is
fullfiled or not
                                                BEGIN
                                                        declare @vchStudentCodeTemp
VARCHAR(11)

                                                        declare
cursorStudentMatriculation CURSOR for
                                                  SELECT DISTINCT
STUDENT_PKEY
                                                  FROM
STAGING_AREA_EXTENSION
                                                  WHERE     STUDENT_PKEY =
@vchStudentCode

                                                        Open cursorStudentMatriculation
                                                        FETCH
cursorStudentMatriculation INTO @vchStudentCodeTemp
                                                        IF @@FETCH_STATUS = 0
                                                                SELECT
@cMatriculationStatus = "Y" ---- Matriculation record exists
                                                        ELSE
                                                                SELECT
@cMatriculationStatus = "N"

                                                        Close cursorStudentMatriculation
                                                        Deallocate
cursorStudentMatriculation
                                                END
                                        ELSE
                                                ------ Student is non entering, entering, or
transfer student
                                                SELECT @cMatriculationStatus = "Y"
        ---- Matriculation record exists

                                        ---
))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
                                        ----SELECT @iRowCount = @iRowCount + 1
        ----- Increment the row count variable
                                        --SELECT @vchAcademicYear +
Year(getdate())

                                        ---- Check Grade catgeory of the module
                                        IF      (@vchGradeCategory = "21" OR
```

```
                                        @vchGradeCategory = "70")
                                        begin
                                                SELECT @cHemisStudent = "N"
                                                SELECT
@cHemisStudentReason = @vchGradeCategory
                                                SELECT
@cHemisStudentReasonOverall = @vchGradeCategory
                                        end
                                ELSE
                                        begin
                                                SELECT @cHemisStudent = "Y"
                                                SELECT
@cHemisStudentReason = @vchGradeCategory
                                                SELECT
@cHemisStudentReasonOverall = @vchGradeCategory
                                        end



                                ---- Check either student has enrolled after
census date
                                ---- or dropped/withdrawn before census date
                                ---- Enrl
                                IF @vchEnrlStatusReson = "ENRL" and
                                        @cCensusDateFlag = "A"
                                                begin
                                                        SELECT
@cHemisStudent = "N"
                                                        SELECT
@cHemisStudentReason = "A5"
                                                        SELECT
@cHemisStudentReasonOverall =  "A5"
                                                end
                                ELSE IF @vchEnrlStatusReson = "DROP" and
                                        @cCensusDateFlag = "B"
                                        begin
                                                SELECT @cHemisStudent = "N"
                                                SELECT
@cHemisStudentReason = "A6"
                                                SELECT
@cHemisStudentReasonOverall = "A6"
                                        end

/*                              ELSE IF @vchEnrlStatusReson = "WDRW" and
                                        @cCensusDateFlag = "B"
                                                begin
                                                        SELECT
@cHemisStudent = "N"
                                                        SELECT
@cHemisStudentReason = "A6"
```

```
                                        SELECT
@cHemisStudentReasonOverall = "A6"
                                    end*/

                        --- If Course-FTE is zero than the student is not
a subsidy student
                        --- Use this chack after 2002 records
                        IF @rCourseFTE = 0
                            begin
                                --select @rCourseFTE
                                SELECT @cHemisStudent = "N"
                                SELECT
@cHemisStudentReason = "AA"
                                SELECT
@cHemisStudentReasonOverall = "AA"
                            end

                        --- If a plan is dropped and no entry found in
PS_ENRL_REQ_DETAIL
                        --- ZZZZ was entered. Such plan is not for
subsidy
                        IF @vchPlanKey = "ZZZZ"
                            begin
                                --select @rCourseFTE
                                SELECT @cHemisStudent = "N"
                                SELECT
@cHemisStudentReason = "AC"
                                SELECT
@cHemisStudentReasonOverall = "AC"
                            end

                        ---- If module dows not belog to a primary plan
                        ---- than module does not belong to subsidy
                        ---- All plans for subsidies if an FTE value exists
                        /*IF @iPrimaryPlan <> "1"
                            begin
                                --select @rCourseFTE
                                SELECT @cHemisStudent = "N"
                                SELECT
@cHemisStudentReason = "AB"
                                SELECT
@cHemisStudentReasonOverall = "AB"
                            end*/

                    --      SELECT @VCHCOURSE

                        ---- Update the table contents
                        --select @cHemisStudent +
@cHemisStudentReason + @cHemisStudentReasonOverall
```

```
                                        --select @vchStudentCode + " " + @vchCourse
+ " " + @vchSemester + " " + @vchGradeCategory
                                        UPDATE STAGING_AREA_ENROL
                                        SET Hemis_Student = @cHemisStudent,
                                        Hemis_Student_Reason =
@cHemisStudentReason,
                                        Hemis_Student_Reason_Overall =
@cHemisStudentReasonOverall,
                                        Matriculation_Status = @cMatriculationStatus
                                        WHERE         Student_pKey =
@vchStudentCode
                                        AND   Course_pKey = @vchCourse
                                        AND Semester_Pkey = @vchSemester
                                        AND Grade_Category = @vchGradeCategory
                                END

        END
------- Close the Inner cursor
close cursorStudent
deallocate cursorStudent
```

Appendix S Program entrance category

This procedure calculates student entrance category status in the institutions.

```
CREATE proc [dbo].[P_Entrance_Category]
        @vchStudentCode varchar(11)
as

   declare cursorStudent CURSOR for
        SELECT
Plan_Pkey,Academic_Career_Sequence,Education_Level,Acad_Year
        FROM STAGING_AREA_ENROLLMENT
        WHERE STUDENT_PKEY =  @vchStudentCode
        AND DATA_MODIFIED = "N"
        AND CAL_ENT = "Y"          -- Left plans having ZZZZ and see only valid
plans
      Order by Acad_Year,Academic_career_Sequence

   open cursorStudent

while 1 = 1
   begin
        declare @vchAcademicPlanCode1 varchar(10)
        declare @vchAcademicCareerPostFix1 varchar(3)
        declare @vchAcademicCareerSequence1 char(1)
        declare @vchDegreeLevel char(1)
        declare @vchEducationLevel char(2)
        declare @vchAcadYear char(4)

      fetch next from cursorStudent into @vchAcademicPlanCode1,
        @vchAcademicCareerSequence1,@vchEducationLevel,@vchAcadYear

      if @@fetch_status <> 0
         break

        ---- Get Academic career to put postfix U and P to the
        ---- entrance category
        IF @vchEducationLevel = "01"
                SELECT @vchAcademicCareerPostFix1 = "U"
        Else If @vchEducationLevel = "02"
                SELECT @vchAcademicCareerPostFix1 = "U"
        Else If @vchEducationLevel = "03"
                SELECT @vchAcademicCareerPostFix1 = "U"
        Else If @vchEducationLevel = "11"
                SELECT @vchAcademicCareerPostFix1 = "U"
        Else If @vchEducationLevel = "04"
                SELECT @vchAcademicCareerPostFix1 = "P"
        Else If @vchEducationLevel = "05"
                SELECT @vchAcademicCareerPostFix1 = "P"
        Else If @vchEducationLevel = "06"
                SELECT @vchAcademicCareerPostFix1 = "P"
```

```
        Else If @vchEducationLevel = "07"
                SELECT @vchAcademicCareerPostFix1 = "P"
        Else If @vchEducationLevel = "08"
                SELECT @vchAcademicCareerPostFix1 = "P"

        IF @vchEducationLevel = "ZZ"
        ----- For Ocassional Students execute the following procedure
        ----- To get First, Non-Entering, and Entering students
                EXEC P_Entrance_Category_ZZ
@vchStudentCode,@vchAcademicPlanCode1,
                        @vchAcadYear,@vchEducationLevel

        ELSE
        ----- For rest of Education Level execute the following procedure
                BEGIN
                        ----- Concatenate education level to get Qualification HEMIS
code
                        SELECT @vchAcademicCareerPostFix1 =
@vchAcademicCareerPostFix1 + @vchEducationLevel

                        declare @vchAcademicPlanCode2 varchar(10)
                        declare @vchAcademicCareerSequence2 char(1)

                        ------- If student is an undergraduate student
                        ------- than check if any previous enrolment exists
                        ------- to decide either a student is not a first time entering
student
                        If @vchAcademicCareerSequence1 = 0
                                BEGIN
                                        ---- Declaring second cursor to fetch current
student statistics
                                        declare cursorStudentUGRD CURSOR for
                                        select DISTINCT Plan_pKey
                                    from STAGING_AREA_ENROLLMENT
                                where Student_Pkey = @vchStudentCode
                                        and Acad_Year < @vchAcadYear
                                        AND CAL_ENT = "Y"
                                        ----- Open the record set
                                        open cursorStudentUGRD

                                        fetch next from cursorStudentUGRD into
@vchAcademicPlanCode2

                                        If @@FETCH_STATUS = 0 ---If a record
exists
                                                BEGIN
                                                        --- Check either a student is
entering or non entering student
                                                        declare
@vchAcademicPlanCode3 varchar(10)
```

```
                                                declare
@vchAcademicCareerSequence3 char(1)

                                                declare
cursorStudentPreviousPlan CURSOR for
                                                select DISTINCT AA.Plan_pKey
                                              from
STAGING_AREA_ENROLLMENT AA
                                            where Student_Pkey =
@vchStudentCode
                                                and Academic_Career_Sequence
= @vchAcademicCareerSequence1
                                                and Acad_Year <
@vchAcadYear
                                                AND CAL_ENT = "Y"

                                                OPEN
cursorStudentPreviousPlan
                                                FETCH
cursorStudentPreviousPlan INTO @vchAcademicPlanCode3

                                                IF @@FETCH_STATUS = 0---
If the current plan exists
                                                    BEGIN
                                                        declare
cursorStudentPreviousUGRD CURSOR for
                                                        select DISTINCT
AA.Plan_pKey
                                                      from
STAGING_AREA_ENROLLMENT AA
                                                    where Student_Pkey =
@vchStudentCode
                                                        AND Plan_pKey
= @vchAcademicPlanCode1
                                                        and
Academic_Career_Sequence = @vchAcademicCareerSequence1
                                                        AND Acad_Year
< @vchAcadYear
                                                        AND CAL_ENT
= "Y"
                                                        ----- Open the
record set
                                                        open
cursorStudentPreviousUGRD
                                                        FETCH
cursorStudentPreviousUGRD INTO @vchAcademicPlanCode3
                                                        IF
@@FETCH_STATUS = 0     -- If Record exists
                                                            UPDATE
STAGING_AREA_ENROLLMENT
```

```
                                        Set
Entrance_Category = "N" + @vchAcademicCareerPostFix1
                                        Where
Student_pKey = @vchStudentCode
                                        AND
Plan_pKey = @vchAcademicPlanCode1
                                    AND
Acad_Year = @vchAcadYear
                                ELSE
                                    UPDATE
STAGING_AREA_ENROLLMENT
                                        Set
Entrance_Category = "E" + @vchAcademicCareerPostFix1
                                        Where
Student_pKey = @vchStudentCode
                                        AND
Plan_pKey = @vchAcademicPlanCode1

                                        AND
Acad_Year = @vchAcadYear
                                    close
cursorStudentPreviousUGRD
                                    deallocate
cursorStudentPreviousUGRD
                                END
                            ELSE  ----- Student is never
registered in any undergraduate program
                                ----- Student is a first
entry student
                                ----- Check either he is a
transfer student or first time entry student
                                BEGIN
                                    declare
@vchExtOrg varchar(11)
                                    declare
@vchDegreeYear Char(4)
                                    declare
cursorStudentPreviousDegree CURSOR for
                                    select
Ext_Org_Id,Year(Degree_Dt) as Date_of_Passing
                                from
STAGING_AREA_EXTENSION
                                where Student_Pkey
= @vchStudentCode
                                    and Ext_Org_Id
like "U%"       ---- Is a previous university degree student

                                    open
cursorStudentPreviousDegree
```

```
                                                                fetch next from
cursorStudentPreviousDegree into @vchExtOrg,@vchDegreeYear
                                                                IF
@@FETCH_STATUS = 0
                                                                    BEGIN
                                                                        IF
@vchDegreeYear <= @vchAcadYear
        UPDATE STAGING_AREA_ENROLLMENT
        Set Entrance_Category = "T" + @vchAcademicCareerPostFix1
        Where Student_pKey = @vchStudentCode
        AND   Plan_pKey = @vchAcademicPlanCode1
AND Acad_Year = @vchAcadYear
        ELSE
        UPDATE STAGING_AREA_ENROLLMENT
        Set Entrance_Category = "F" + @vchAcademicCareerPostFix1
        Where Student_pKey = @vchStudentCode
        AND   Plan_pKey = @vchAcademicPlanCode1
AND Acad_Year = @vchAcadYear
                                                                    END
                                                                ELSE
                                                                    UPDATE
STAGING_AREA_ENROLLMENT
                                                                    Set
Entrance_Category = "F" + @vchAcademicCareerPostFix1
                                                                    Where
Student_pKey = @vchStudentCode
                                                                    AND
Plan_pKey = @vchAcademicPlanCode1
                                                                AND
Acad_Year = @vchAcadYear

                                                                Close
cursorStudentPreviousDegree
                                                                Deallocate
cursorStudentPreviousDegree
                                                            END
                                                    Close cursorStudentPreviousPlan
                                                    Deallocate cursorStudentPreviousPlan
                                                    END
```

```
                                    ELSE  ---- Check the first time entry status of
the student
                                        BEGIN
                                            declare @vchExtOrg1
varchar(11)
                                            declare @vchDegreeYear1
Char(4)
                                            declare
cursorStudentPreviousDegree CURSOR for
                                            select
Ext_Org_Id,Year(Degree_Dt) as Date_of_Passing
                                          from
STAGING_AREA_EXTENSION
                                          where Student_Pkey =
@vchStudentCode
                                            and Ext_Org_Id like "U%"     ----
Is a previous university degree student

                                            open
cursorStudentPreviousDegree

                                            fetch next from
cursorStudentPreviousDegree into @vchExtOrg1,@vchDegreeYear1
                                            IF @@FETCH_STATUS = 0
                                                BEGIN
                                                    IF
@vchDegreeYear1 <= @vchAcadYear
                                                        UPDATE
STAGING_AREA_ENROLLMENT
                                                        Set
Entrance_Category = "T" + @vchAcademicCareerPostFix1
                                                        Where
Student_pKey = @vchStudentCode
                                                        AND
Plan_pKey = @vchAcademicPlanCode1
                                                    AND
Acad_Year = @vchAcadYear
                                                ELSE
                                                    UPDATE
STAGING_AREA_ENROLLMENT
                                                    Set
Entrance_Category = "F" + @vchAcademicCareerPostFix1
                                                    Where
Student_pKey = @vchStudentCode
                                                    AND
Plan_pKey = @vchAcademicPlanCode1
                                                AND
Acad_Year = @vchAcadYear
                                            END
```

```
                                        ELSE
                                            UPDATE
STAGING_AREA_ENROLLMENT
                                            Set Entrance_Category =
"F" + @vchAcademicCareerPostFix1
                                            Where Student_pKey =
@vchStudentCode
                                            AND   Plan_pKey =
@vchAcademicPlanCode1
                                          AND Acad_Year =
@vchAcadYear

                                        Close
cursorStudentPreviousDegree
                                        Deallocate
cursorStudentPreviousDegree
                                    END
                        Close cursorStudentUGRD
                        Deallocate cursorStudentUGRD

                        END
                ------))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
                ------))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
                ELSE  ------ Student is a Postgraduate student
                        BEGIN
                            ---- Declaring second cursor to fetch current
student statistics
                            declare cursorStudentPreviousPGRD CURSOR
for
                            select DISTINCT AA.Plan_pKey
                          from STAGING_AREA_ENROLLMENT AA
                        where Student_Pkey = @vchStudentCode
                            and Academic_Career_Sequence =
@vchAcademicCareerSequence1
                            AND Acad_Year < @vchAcadYear
                            AND CAL_ENT = "Y"
                            ----- Open the record set
                            open cursorStudentPreviousPGRD

                            fetch next from cursorStudentPreviousPGRD
into @vchAcademicPlanCode2

                            If @@FETCH_STATUS = 0 ---If a record
exists
                                    BEGIN
                                        declare
cursorStudentPreviousPlan CURSOR for
                                        select DISTINCT AA.Plan_pKey
                                      from
STAGING_AREA_ENROLLMENT AA
```

```
                                        where Student_Pkey =
@vchStudentCode
                                            AND Plan_pKey =
@vchAcademicPlanCode1
                                            and Academic_Career_Sequence
= @vchAcademicCareerSequence1
                                            AND Acad_Year <
@vchAcadYear
                                            AND CAL_ENT = "Y"
                                            ----- Open the record set
                                            open cursorStudentPreviousPlan

                                            fetch next from
cursorStudentPreviousPlan into @vchAcademicPlanCode2
                                            IF @@FETCH_STATUS = 0----
Previous plan record exists
                                                UPDATE
STAGING_AREA_ENROLLMENT
                                                Set Entrance_Category =
"N" + @vchAcademicCareerPostFix1
                                                Where Student_pKey =
@vchStudentCode
                                                AND   Plan_pKey =
@vchAcademicPlanCode1
                                              AND Acad_Year =
@vchAcadYear
                                            ELSE

                                                UPDATE
STAGING_AREA_ENROLLMENT
                                                Set Entrance_Category =
"E" + @vchAcademicCareerPostFix1
                                                Where Student_pKey =
@vchStudentCode
                                                AND   Plan_pKey =
@vchAcademicPlanCode1
                                              AND Acad_Year =
@vchAcadYear


                                        Close cursorStudentPreviousPlan
                                        Deallocate cursorStudentPreviousPlan

                                        END

                                    ELSE  ---- Check the first time entry status of
the Postgraduate student
                                        BEGIN

                                            declare
@vchAcademicCareerPlanPrevious varchar(10)
```

```
                                                declare
cursorStudentPreviousGRD CURSOR for
                                                select DISTINCT Plan_pKey
                                            from
STAGING_AREA_ENROLLMENT
                                        where Student_Pkey =
@vchStudentCode
                                                and Acad_Year <
@vchAcadYear
                                                AND CAL_ENT = "Y"

                                                open cursorStudentPreviousGRD

                                                fetch next from
cursorStudentPreviousGRD into @vchAcademicCareerPlanPrevious
                                                IF @@FETCH_STATUS = 0 --
If previous entries found
                                                        BEGIN
                                                                ---- If the post
graduate student is a Matser or Doctoral student
                                                                ---- than he is a
transfer student else First time entry student
                                                                IF
@vchEducationLevel = "07" OR
        @vchEducationLevel = "08"
                                                                        UPDATE
STAGING_AREA_ENROLLMENT
                                                                        Set
Entrance_Category = "T" + @vchAcademicCareerPostFix1
                                                                        Where
Student_pKey = @vchStudentCode
                                                                        AND
Plan_pKey = @vchAcademicPlanCode1
                                                                    AND
Acad_Year = @vchAcadYear
                                                                ELSE
                                                                        UPDATE
STAGING_AREA_ENROLLMENT
                                                                        Set
Entrance_Category = "F" + @vchAcademicCareerPostFix1
                                                                        Where
Student_pKey = @vchStudentCode
                                                                        AND
Plan_pKey = @vchAcademicPlanCode1
                                                                    AND
Acad_Year = @vchAcadYear
                                                        END
                                                ELSE
```

```
                                                    ----- Check student matriculation
entries
                                                        BEGIN
                                                            declare
@vchExtOrgPGRD varchar(11)
                                                            declare
@vchDegreeYearPGRD Char(4)
                                                            declare
cursorStudentPreviousDegreePGRD CURSOR for
                                                            select
Ext_Org_Id,Year(Degree_Dt) as Date_of_Passing
                                                        from
STAGING_AREA_EXTENSION
                                                            where Student_Pkey
= @vchStudentCode
                                                            and Ext_Org_Id
like "U%"       ---- Is a previous university degree student


                                                            open
cursorStudentPreviousDegreePGRD

                                                            fetch next from
cursorStudentPreviousDegreePGRD into
@vchExtOrgPGRD,@vchDegreeYearPGRD
                                                            IF
@@FETCH_STATUS = 0
                                                                BEGIN
                                                                    IF
@vchDegreeYearPGRD <= @vchAcadYear
        UPDATE STAGING_AREA_ENROLLMENT
        Set Entrance_Category = "T" + @vchAcademicCareerPostFix1
        Where Student_pKey = @vchStudentCode
        AND   Plan_pKey = @vchAcademicPlanCode1
AND Acad_Year = @vchAcadYear
        ELSE
        BEGIN
            ---- If the post graduate student is a Matser or Doctoral student
            ---- than he is a transfer student else First time entry student
            IF @vchEducationLevel = "07" OR
```

```
                @vchEducationLevel = "08"
                UPDATE STAGING_AREA_ENROLLMENT
                Set Entrance_Category = "T" + @vchAcademicCareerPostFix1
                Where Student_pKey = @vchStudentCode
                AND   Plan_pKey = @vchAcademicPlanCode1
              AND Acad_Year = @vchAcadYear
          ELSE
                UPDATE STAGING_AREA_ENROLLMENT
                Set Entrance_Category = "F" + @vchAcademicCareerPostFix1
                Where Student_pKey = @vchStudentCode
                AND   Plan_pKey = @vchAcademicPlanCode1
              AND Acad_Year = @vchAcadYear
    END
                                                    END
                                                ELSE
                                                    BEGIN
                                                        ----
If the post graduate student is a Matser or Doctoral student
                                                        ----
than he is a transfer student else First time entry student
                                                        IF
@vchEducationLevel = "07" OR
    @vchEducationLevel = "08"
    UPDATE STAGING_AREA_ENROLLMENT
    Set Entrance_Category = "T" + @vchAcademicCareerPostFix1
    Where Student_pKey = @vchStudentCode
    AND   Plan_pKey = @vchAcademicPlanCode1
AND Acad_Year = @vchAcadYear
    ELSE
```

```
        UPDATE STAGING_AREA_ENROLLMENT

        Set Entrance_Category = "F" + @vchAcademicCareerPostFix1

        Where Student_pKey = @vchStudentCode

        AND   Plan_pKey = @vchAcademicPlanCode1

AND Acad_Year = @vchAcadYear
                                                                END
                                                            Close
cursorStudentPreviousDegreePGRD
                                                            Deallocate
cursorStudentPreviousDegreePGRD


                                                        END

                                                    Close cursorStudentPreviousGRD
                                                    Deallocate cursorStudentPreviousGRD

                                                    END

                                                Close cursorStudentPreviousPGRD
                                                Deallocate cursorStudentPreviousPGRD
                                                END


                                        END

        END
        close cursorStudent
        deallocate cursorStudent
```

Appendix T Published papers

Two papers based on the research of this study have been accepted for publications. These are as follows:

- **Appendix T-1**

Nazir, A. & McDonald, T. (2006a). Challenges in developing a cost-effective data warehouse for a tertiary institution in a developing country. Proceedings of the *7th International Conference on Data, Text and Web Mining: Business Applications and Management Information Engineering*, Prague, Czech Republic. pp. 389-397.

- **Appendix T-2**

Nazir, A. & McDonald, T. (2006b). Lessons learned in developing a data warehouse for a tertiary institution in South Africa. Paper presented at the 2nd Foundation of Tertiary Institutions of the Northern Metropolis (FOTIM) International Quality Assurance Conference, Pretoria, South Africa.

Challenges in developing a cost-effective data warehouse for a tertiary institution in a developing country

A. Nazir & T. Mc Donald
Department of Computer Science and Informatics,
University of the Free State, South Africa

Abstract

Higher Education institutions have grown through the years and have developed into large businesses in and of themselves. Even though industry has experienced a dramatic increase in the use of data warehousing techniques, tertiary institutions have been slow to follow suit. Reasons for this can be attributed mainly to the many reported failures and the costs involved there in. Several factors are forcing these institutions to embrace data warehousing technologies. This paper will report on the challenges involved in taking this step and will show that a successful data warehouse can be developed, regardless of the obstacles involved.

Keywords: higher education, data warehousing, student data mart, star schema, developing countries.

1. Introduction

Industry in developed countries is experiencing a dramatic increase in the use of Data Warehousing (DW) techniques. This can, however, in most cases only be achieved at huge cost. According to the Western Michigan University [1] the primary consideration in the development of a DW is cost. Wierschem et al. [2] indicates that a DW requires millions of dollars to develop, plus significant hardware and personnel investment. Hammond [3] quotes a Meta Group survey that the average cost for an enterprise warehouse is $3 million. This can be a major obstacle in developing countries.

Wagner et al. [4] stated that the budgets of developing countries are not even sufficient to pay for the Knowledge Management (KM) enabling IT architecture. Less developed countries like the Philippines and Pakistan spend a smaller percentage of their budgets on IT, as compared to developed countries like the United States and the UK. For example, the Philippines spend 0.8 percent of its budget on IT, compared to the United States which spends 13 percent. From the above statistics there is a great need to find ways for entering DW technology in developing countries with their limited budgets for IT.

In South Africa (SA) a limited number of organizations are using DW and this technology is still emerging. Pioneering organizations like Electricity Companies and Banks set up very large databases for their management and executive information systems well before the warehouse concept was established (Griffiths [5]). DW technology entered SA in 2001 and a number of organizations like telecommunications companies and banks are currently using DW technologies successfully.

Higher Education (HE) has been slow to follow suit [6]. Before the advent of democracy in 1994, the SA government's tertiary education funding policies mirrored apartheid's divisions and the different governance models which it imposed on the HE system (Bunting [7]). For the new government that came into power in 1994, the focus was to address the imbalances of the past, especially health, housing and primary education. The result was that the subsidies allocated to universities (primary source of income) have drastically been cut. Most of universities that still survive today had to go through a period of tough rationalizations. The bad part of all this is that universities in SA now run on limited budgets. The good part is that they have grown through the years and educational institutions have developed into large businesses in and of themselves (Desruisseaux [8]). This change has resulted in a more business-like management of these institutions as well (Lazerson et al. [9]).
It is clear from the above that a number of factors are forcing universities in the direction of DW. Wierschem et al. [2] states that the environmental factors that encourage academic institutions to investigate DW options are decreases in governmental financial support, faculty supplies, and research funding, and increases in student tuitions, competition, faculty salaries, faculty support and the expectations from students, parents and employers. Add to the above list the fact that universities must nowadays compete as businesses in order to survive.

To enter this new world of DW, tertiary institutions must face a number of challenges. Cost is one of the challenges. Another one is that many of the current OLTP systems lack data integrity and errors abound. Still another challenge is that top management is unaware of what DW is and the advantages it brings. The DW must also be able to supply the required statistics to government and in-house information for strategic planning and decision making.

This paper reports on how one institution in South Africa, The University of the Free State, tackled and overcame the challenges. First some background will be provided on the history of the current MIS systems. That will be followed by a detailed discussion of the challenges involved to get a DW up and running and how the challenges were overcame. The paper then concludes with the lessons learned which can be applied by other universities in a similar position.

2. Background

2.1 Old IBM system

In 1986 an in-house system was developed by using the IBM platform to fulfil the requirements for data storage and retrieval. With the passage of time the system became inadequate to accomplish its tasks. Some of the major reasons for system failure were programming languages that became obsolete, developers having left the University and the system which was designed in patches and phases ended with lack of data integrity and inconsistency in system interface designing.

2.2 OLTP system from an international company

After the bad experience with the IBM in-house development, the University was ardent to buy a system from some international company, because more and more universities were opting to implement integrated software packages from this company. In 2003 the University purchased a new OLTP system at huge cost by considering the following factors: a complete package with design consistency, analytical and strategic reporting capability, new technology and full technical and maintenance support.

During the course of data transfer from the old to the new system (with a different database design) numerous data conversion errors generated anomalies and a lack of integrity in the database. The new system also proved inadequate to provide the necessary statistics.

Within one year after the installation of the new OLTP system, the University faced a number of new challenges that they have never considered when purchasing this commercial product. The main problem was the lack of customization of the product and they are now not in a position to afford the customization costs.

2.3 HEMIS analyzer

The operational system proved inadequate to provide the necessary statistics to the Department of Education (DOE), therefore, the Planning Unit of the University purchased a new system, HEMIS analyzer, from a local company at the end of 2005. The HEMIS system is basically designed according to the format specified by the DOE. Data is uploaded into the system from ASCII files which the University generates to provide unit record statistics of students and personnel to the DOE (which uses it to allocate the subsidies to the universities). This system provided a workable solution, but with changes in requirements, new reports must be developed and it was worthless for institutional planning and forecasting purposes.

3. Challenges

The following sections will provide details on the challenges that were faced during the development of a Student Data Mart (SDM).

3.1 Dirty data

The University of the Free State is among one of the oldest universities in SA. It has computerized student records from 1946 up to today. The University was interested to preserve this history data because historical data are necessary for business trend analysis [10]. A decision was made to migrate from the old IBM data to the new OLTP system. During this migration numerous data conversion errors generated anomalies and a lack of data integrity

.

3.1.1 Missing academic program and plans

There are numerous combinations of academic programs and academic plans for which students were enrolled in the past. These combinations are no longer valid or do not exist anymore. Therefore one can not trace the degree for which the student was enrolled.

3.1.2 No uniqueness

In the Academic Plan table several academic plans were entered several times with different effective dates. One can even enter a new row with the same date and with the same entries, because no primary key constraint was enforced on any table.

3.1.3 Inconsistency in data

Student demographic information is very important for certain types of analysis like drilling down to the country, state/province and to city level. In the OLTP system there is no way for standardizing cities or other such values. For example the city name Bloemfontein was entered 16 times with different spelling.

3.1.4 Spaces in mandatory columns

It was explained in previous sections that the system did not enforce integrity constraints, but Not Null constraints can be found in most of the tables. While populating history data from the old IBM system, spaces were added in such mandatory columns where no corresponding value was found.

3.1.5 Missing links

When students successfully completed their degrees, single entries were made per plan in the Academic Degree Plan and Academic Degree tables for

graduation records. In several cases no corresponding entry was found in the Student Enroll table with the plan on which students received their graduations.

The reason for these missing links was the fact that students enrolled in one academic plan, for example a four year bachelors degree, but after completing certain modules the student wanted to scale down to a three year bachelors degree. The department and administration accepted these requests and awarded a three year degree with a different plan. In the OLTP system there is no way to link these changes where the student was previously enrolled in plan A and now received his degree in plan B. In the same way there are students whose record exists in the Student Enroll table, but no related record exists in the student demographic table.

3.1.6 Wrong entries

The entries that the systems itself identify by picking different academic programs, plans and modules from the front-end were wrong. For example, the academic career for the same module was entered with different academic careers in different tables.

3.1.7 Year and semester modules conflicts

In the University there is a difference between the module start, end or census dates for semester and year modules. The OLTP database structure allows entry for semester modules only. For example, 2061 and 2062 represents the first and the second semester of the year 2006 respectively. This scheme works for entering semester modules' start, end, and census dates, but fails when entering year modules.

Administrative staff endeavours to fix or resolve the above data errors but fail to do so, because it is too difficult to fix the errors in the OLTP system now, unless by truncating data from all of the tables and enforcing integrity constraints for future data. This method is still not cheap, because the University has to get consultancy from the international company again and a huge budget is required to do so.

3.2 Limited budget

In the Introduction it was emphasised that the University works with a limited budget. After several wrong and costly decisions they are wary of any new IT expenditures. Ways and means had to be found to develop a DW with minimum cost.
Wierschem et al. [2] pointed out that the development cost of a DW can be reduced by reducing the overall scope of the project and the project can also be broken down into smaller components and developed over a longer period of time. According to a survey by Wagner et al. [4] it was concluded that enterprise

solutions are not suitable for developing countries. For these reasons it was decided to develop only the Student Data Mart for a start. Up till now not much work has been done on fitting student record data to the dimensional model star schema [11].

To further cut down on costs no special hardware or software was purchased, but existing hardware and software were utilized. Preference should be given to in-house existing staff instead of seeking outside help [12]. This advice was followed and a single person trained in DW was placed in the Planning Unit (MIS Department) of the University in order to develop the SDM. Detail requirements were gathered before and during the development process and therefore there was no chance for misunderstanding in the requirement gathering phase.

3.3 Prototypes of student data mart

In this section the star model for the SDM that is more suitable for extracting data for DOE reports and institutional strategic reporting, forecasting, and predictive modelling will be discussed. Details are also provided for the extraction, transformation and loading process into the DW. A number of procedures were written to fix the errors and enrich the data with information that is required by the DOE and institutional internal needs, but which is not available in the OLTP system.

3.3.1 Grain and summarization of the SDM

According to a study, [11], the grain of a student record star schema with dimensions of time (academic year, term), student demographics (one record per student) term (one record per student per term) and student matriculation (one record per student per course of study undertaken) would be “student per term per course of study.”

To store student enrollments with plans and courses two different fact tables were designed in the current SDM: Student_Enrollment_Fact (see fig. 1) with granularity “student per semester per plan” and Student_Course_Fact (not shown) with granularity “student per semester per plan per course”. The main reason for designing two separate fact tables is the grain and summarization of the fact tables. The Student_Enrollment_Fact store summarized data and it, therefore, contains only 13 percent of the number of rows of the Student_Course_Fact. During the requirement gathering phase it was identified that most of the reports used for reporting and planning purposes were based on the head counts of students in a semester or a year. Student_Enrollment_Fact can answer these queries more rapidly than the Student_Course_Fact.

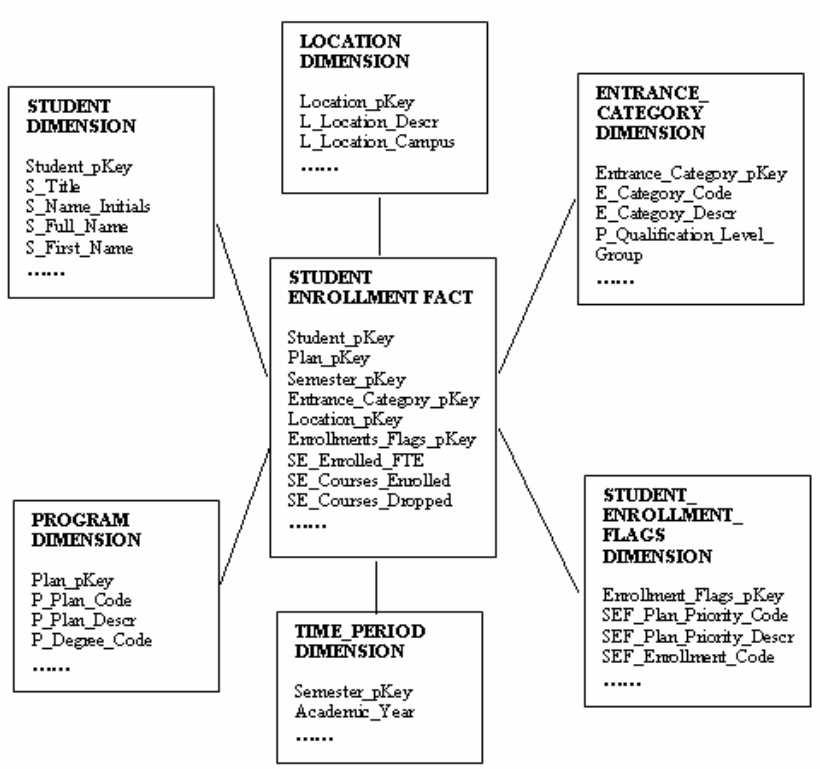

Figure 1: Student enrollment star

3.3.2 Student dimension

The student dimension is the biggest dimension in this SDM holding 59 columns. There are fifteen normalized OLTP tables which hold student records. To extract data from these fifteen tables a complex query was required with a number of outer joins and the need to find the last effective date. With this dimension one can drill down and roll up among different hierarchies. A set of procedures were written to calculate the student age group by fixing invalid student birth date, native language, nationality, ethnicity and resident city.

3.3.3 Junk dimensions

A junk dimension is a convenient grouping of flags and indicators. It is helpful, but not absolutely required [13]. In this project it was concluded that junk dimensions are very useful in the SDM to enrich data for providing certain statistics which are not available in the source OLTP system (see the Student_Enrollment_Flags in fig. 1). The DOE needs student's enrollments together with the primary plan and this information is not available in the source

data. A set of filters were used to find students' records which qualify for government subsidies. Some of these filters were: enrolments before the census date, students who had dropped out or withdrawn their plans before the census date, undergraduate students who fulfil their matriculations and students who failed or received re-assessments. This information is processed in the staging area and corresponding flags and indicators are added with the extracted data. A surrogate key is generated in the Student_Enrollment_Flags by extracting unique combinations of flags and indicators. The size of this junk dimension is 0.035 percent of the size of the Student_Enrollment_Fact rows.

From fig. 2 one can see the benefits of using the Student_Enrollment_Flags dimension. Before the SDM it was very difficult to determine the students' primary plans and students who qualified for government subsidies. Now there is no need to write complex queries for finding these statistics, because the Student_Enrollment_Flags dimension makes this task very easy. One can just drag and drop data elements onto the pivot table and get the required results.

Count of					Acader			
Plan	Enroll	HEMIS	Locatic	Enti	2003	2004	2005	2006
Primary Plan	Enrolled	Subsidy student	Distance	FU	91	257	210	88
			Distance Total		91	257	210	88
			Main	FU	2,840	3,155	3,310	3,319
			Main Total		2,840	3,155	3,310	3,319
			QwaQwa	FU	267	327	454	474
			QwaQwa Total		267	327	454	474
			Vista	FU	220	59	14	13
			Vista Total		220	59	14	13
		Subsidy student Total			3,418	3,798	3,988	3,894
	Enrolled Total				3,418	3,798	3,988	3,894
Primary Plan Total					3,418	3,798	3,988	3,894

Figure 2: Pivot table from student enrollment star

4. Conclusion

Current situations in South Africa are forcing HE institutions to enter the DW arena. They face several difficult challenges like low budgets, dirty OLTP data and the requirement to provide statistics to the government and at the same time have information available for strategic planning and decision making. It has been shown in this paper that the challenges can be overcome by starting small, using in-house knowledge and starting with existing hardware and software. By designing the data marts properly and loading them with clean data, all the required information is available in an easy-to-use manner.

5 References

[1] WMU Strategic Plan for Information Technology: Data warehouse proposal, http://www.wmich.edu/sis/pddv1.0.pdf

[2] Wierschem, D., McMillen, J., & McBroom R., What Academia Can Gain From Building A Data Warehouse. EDUCAUSE QUARTERLY, Number 1, 2003.
[3] Hammond, M., Research finds data warehousing market grew 34% in 97 PC Week Online, 1998.
[4] Wegner, C., et al, Enhancing E-Government in Developing Countries: Managing Knowledge through Virtual Communities. The Electronic Journal on Information Systems in Developing Countries, EJISDC, 14,4, pp. 1-20, 2003
[5] Griffiths, S., Data warehousing – what, where, why and how. Data warehousing conference, Johannesburg, South Africa, 12-13 June 1995,
[6] The Use And Value Of Data Warehousing In Higher Education, http://www.mountainplains.org/articles/mpa15.html
[7] Bunting, I., "Funding" in transformation in higher education: Global pressures and local realities in South Africa. Centre for higher education transformation in South Africa (CHET), Pretoria, 2002.
[8] Desruisseaux, P., Universities Venture into Venture Capitalism. The Chronicle of Higher Education, A44, 2000.
[9] Lazerson, M., Wagener, U., & Moneta, L., Like The Cities They Increasingly Resemble, Colleges Must Train and Retain Competent Managers. The Chronicle of Higher Education, A72, 2000.
[10] Data warehousing: Concepts and mechanisms, http://www.svifsi.ch/revue/pages/issues/n991/a991Gatziu.pdf
[11] Data Models For A Registrar's Data Mart, http://www.georgetown.edu/users/allanr/bridge.pdf
[12] 10 rules for successful data warehousing, http://www.kmworld.com/articles/printarticle.aspx?articleid=9081
[13] Kimball Tip#48: De-Clutter with junk (dimensions), http://www.kimballgroup.com/html/designtipsPDF/DesignTips2003/KimballDT48DeClutter.pdf

Lessons learned in developing a data warehouse for a tertiary institution in South Africa

Amer Nazir
Department of Computer Science and Informatics,
University of the Free State, South Africa
Cell#: +27-(0)73 732 8159
E-Mail: amernazirabn@yahoo.com

Theo McDonald
Department of Computer Science and Informatics,
University of the Free State, South Africa
Phone: +27-(0)51-4012297
E-Mail: Theo.SCI@mail.uovs.ac.za

Abstract

Even though industry in South Africa has utilized data warehousing technologies successfully for a number of years, institutions of higher education have lagged behind. This can in part be attributed to the high costs involved, many failures in the past and the fact that the decision makers of these institutions are unaware of what data warehousing is and the advantages it can bring. Several factors, however, are forcing institutions of higher education in the direction of data warehousing. They need all the help they can get to make this process as easy and as cost-effective as possible.

This paper will report on the lessons learned by one institution that, in spite of numerous problems, successfully developed a cost-effective data warehouse in-house that fulfilled all information requirements needed both by government and the institution itself.

Keywords: tertiary institution, data warehousing, student data mart, star schema, lessons learned.

1 Introduction

Industry in first world countries is experiencing a dramatic increase in the use of Data Warehousing (DW) tools and technologies. In South Africa (SA) pioneering organizations like Electricity Companies and Banks set up very large databases for their management and executive information systems well before the warehouse concept was established [Griffiths, 1995]. DW technology entered South Africa in 2001 and a number of organizations like telecommunications companies and banks are currently using DW technologies successfully.

Tertiary institutions have been slow to follow suit [Webner et al., 2006]. Before the advent of democracy in 1994, the SA government's tertiary education funding policies mirrored apartheid's divisions and the different governance models which it imposed on the tertiary system [Bunting, 2002]. For the new government that came into power in 1994, the focus was to address the imbalances of the past, especially health, housing and primary education. The result was that the subsidies allocated to universities (their primary source of income) have drastically been cut. Most universities still surviving today had to go through a period of tough rationalizations and mergers. Even though the universities in SA now run on limited budgets, they have grown through the though years and tertiary institutions have developed into large businesses in and of themselves [Desruisseaux, 2000]. This change has resulted in a more business-like management of these institutions [Lazerson et al., 2000].

It is clear from the above that a number of factors are forcing universities in the direction of DW. Wierschem et al. (2003) states that the environmental factors that encourage academic institutions to investigate DW options are decreases in governmental financial support, faculty supplies, and research funding, and increases in student tuitions, competition, faculty salaries, faculty support and the expectations from students, parents and employers. Add to

the above list the fact that universities as businesses need strategic information in order to survive.

This paper reports on how one institution in South Africa, The University of the Free State (UFS), tackled and overcame the challenges to develop a DW and the subsequent lessons they learned from it. First some background will be provided on DW technologies and tools to familiarize the reader with this possible new environment. That will be followed by a detailed discussion of the lessons learned in developing a DW for a tertiary institution. The paper then concludes with the contribution of this paper to the establishment of DW in tertiary institutions.

2 Background

Because most of the readers of this paper may not be familiar with basic business intelligence concepts and where DW fits in, a short introduction will be provided. It is also done in order to make the next section more understandable.

2.1 Business intelligence

Business intelligence (BI) is the process of getting enough of the right information in a timely manner and usable form and analyzing it so that it can have a positive impact on business strategy, tactics or operations [Wally, 2003].

2.2 Business intelligence tools

BI tools are back-end, infrastructure tools that deal with extracting data, cleaning it, transforming it, re-organizing it, and optimizing it for use in decision making. These back-end tools include extraction, transformation and loading (ETL), data warehousing and OLAP server tools.

BI front-end tools are designed to extract knowledge and insight from the data once it has been prepared. These include reporting, querying, on-line analysis and exploration, visualization, decision modeling and planning, and data mining tools. Portals, dashboards, and scorecards are also pieces of the puzzle that help further organize information for easy consumption [Lokken, 2001].

2.3 Data warehousing is the basis for business intelligence

A DW is the basis of BI and a DW itself does not create value; value comes from the use of the data in the warehouse [List, 2002]. According to Donhardt et al. the DW empowers institutional decision makers by placing inquiry and analysis tools at their fingertips. This has the following benefits:

- Users can produce customized reports anytime, anywhere.

- Easy access and quick information delivery support administrative decisions at all levels and improve the way the organization does business.
- Giving users the ability to generate their own reports greatly reduces the effort once spent by the MIS department developing ad-hoc programs and answering questions.

2.4 Data warehousing basic concepts

Online transaction processing (OLTP) systems are the systems that are used to run the day-to-day business of the company, the so called bred-and-butter systems [Ponniah, 2001, p.10]. In contrast a DW is a copy of transaction data specifically structured for querying and analysis [Kimball, 1996, p.310]. According to Ponniah (2001, p.13) the DW is an informational environment that:

- Provides an integrated and total view of the enterprise
- Makes the enterprise's current and historical information easily available for decision making
- Makes decision-support transactions possible without hindering OLTP systems
- Renders the organization's information consistent
- Presents a flexible and interactive source of strategic information.

Table 1 provides a better understanding of the differences between OLTP and DW systems [Ponniah, 2001, p.11].

Table 1: Difference between OLTP and DW systems

Feature	OLTP System	Data Warehousing
Data Content	Current values	Archived, derived, summarized
Data Structure	Optimized for transaction	Optimized for complex queries
Access Frequency	High	Medium to low
Access Type	Read, update, delete	Read
Usage	Predictable, repetitive	Adhoc, random, heuristic
Response Time	Sub-seconds	Several seconds to minutes
Users	Large number	Relatively small number

2.5 DW architecture components

The three major components of a DW are data acquisition, data storage and information delivery as shown in Figure 1. As can be seen in the figure a data mart is a subset of a data warehouse for use by a single department or function.

2.5.1 Data acquisition

Data from different sources are extracted and moved to the staging area and prepared for loading into the DW storage. The source can be any legacy system, flat file or OLTP system. In the staging area each file is extracted by performing various transformations like sort, merge, resolving inconsistencies, and cleansing of the data.

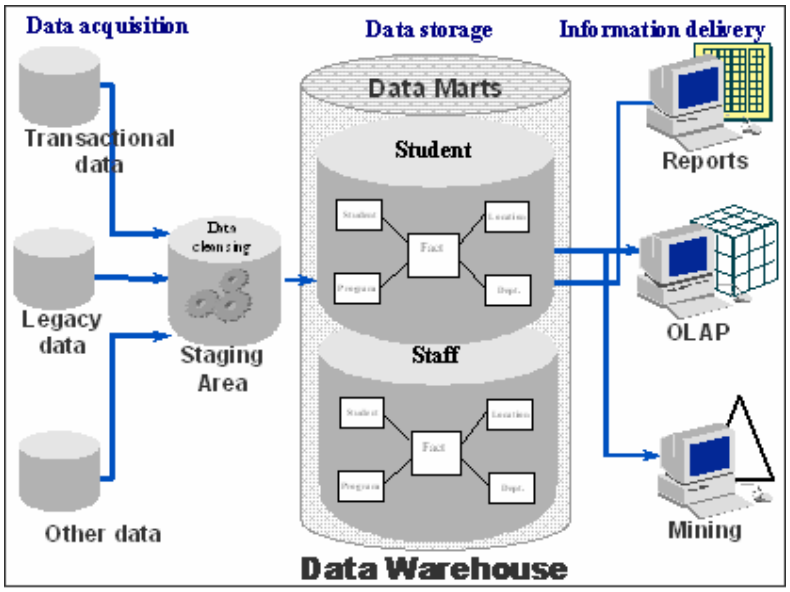

Figure 1: Data warehouse architecture

2.5.2 Data storage

The cleansed data from the staging area is stored in the DW using a star schema or also called a dimensional model. This is the fundamental data design technique for the DW. The star schema is based on one large central table called a fact table and a set of smaller attendant tables displayed in a radial pattern around the central table called dimensional tables. The dimension tables represent the business dimensions along which the metrics are analyzed. The attributes in a dimension table are of textual format. A fact table stores the measures of the business. The data grain is an important characteristic of the fact table and is the level of detail for the measurements or metrics. Some fact tables may just contain summary data and are called aggregate fact tables.

2.5.3 Information delivery

The information delivery component makes it easy for the users to access the information either directly from the DW or from the dependent data marts. The DW can satisfy users of all levels of management by providing data in the form of reports, Excel pivot tables, online analytical processing (OLAP) and data mining (DM) technologies.
OLAP technology organizes data in multidimensional tables (also called cubes) and provides access to the DW through an interactive graphical user interface [Gorla, 2003]. OLAP tools are for users who require intensive data

analysis capabilities. The business problem that OLAP tools solve is the need for users to “drill” seamlessly into information when additional details are required. These tools insulate users from the details surrounding the retrieval of information from the DW [Trepte, 1997].

Data mining refers to extracting or “mining” knowledge from large amounts of data [Han, 2001, p.5]. The benefit of DM is the ability to gain a deeper understanding of the patters previously unseen when using current available reporting capabilities.

3 Lessons learned in developing a DW for a tertiary institution

Many universities in South Africa are facing the same problems of obtaining and providing information as the University of the Free State. Data warehousing is seen as a possible solution for these problems. In this section valuable lessons learned during the process of designing a DW for the university will be discussed.

3.1 The OLTP system is not suitable for strategic information

In 1986 an in-house system was developed by using the IBM platform to fulfill the requirements for data storage and retrieval. With the passage of time the system became inadequate to accomplish its tasks. Some of the major reasons for system failure were programming languages that became obsolete, developers having left the University and the system which was designed in patches and phases ended with a lack of data integrity and inconsistency in system interface design.

After the bad experience with the IBM in-house development, the University was ardent to buy a system from some international company, because more and more universities were opting to implement integrated software packages from this company. In 2003 the University purchased a new OLTP system at huge cost by considering the following factors: a complete package with design consistency, analytical and strategic reporting capability, new technology and full technical and maintenance support. However, within one year after the installation of the new OLTP system, the University faced a number of new challenges that they have never considered when purchasing this commercial product. The main problem was the lack of customization of the product and they are currently not in a position to afford the customization costs.

During the data transfer from the old to the new system (with a different database design) numerous data conversion errors generated anomalies and a lack of integrity in the database. Because of all the errors, the new system also proved inadequate to provide the necessary statistics required by the Department of Education (DoE), as well as information required by top management.

3.2 Third party software is not suitable for strategic information

The DoE needs unit record statistics of students and staff quarterly or yearly from all tertiary institutions for planning and subsidy purposes. In August 1999, the Department provided to all institutions the technical details about the collections, specifications for file scopes and file structures. The department maintains a PC-based software package, VALPAC, to import and validate data in ASCII files which the institutions must submit to them. Although a set of validation rules check the accuracy of the data, it is not at acceptable levels and it resulted in incorrect and incomplete information. The users of the system also find the interface very user-unfriendly and even the error reports generated by the system are difficult to interpret and understand by non-technical users.

The DoE is of the opinion that instead of just using the system for data submission and validation, the system can also be helpful for institutional internal reporting, planning and forecasting. The system, however, allows reporting on only two years' data and as such has minimal value for institutional forecasting and planning.

The responsibility for ensuring the accuracy and completeness of the data in the returns provided to the Department rests with the institutions and the institutions must be confident about the reasonableness and accuracy of the data summarized prior to sending it to the Department. To provide clean and correct data to the DoE has become a major challenge for the tertiary institutions.

To assist in this regard, several institutions are also using the HEMIS reporting systems developed in-house, or in most of the cases purchased from some third parties. The HEMIS reporting systems are basically designed according to the format specified by the DoE. Data is uploaded into the system from ASCII files which the University generates. These systems also provide some sort of basic data validation by printing error reports. It is also possible to generate reports for in-house usage.

The VALPAC and HEMIS systems provide a workable solution, but with several drawbacks. To generate the ASCII files and reports complex stored procedures must be written. Changes in requirements require new procedures. The requirements are changing all of the time, for instance the DoE recently provided details to the institutions for new fields that are added to the existing database files, along with four new files for providing campus building information. Management as a rule also regularly requires new reports. These reports must all be hard-coded. It is a waste of resources to spend time on formatting and publishing these reports which have a very short life time.
One of the main reasons for purchasing the third party HEMIS reporting systems is the need to clean and validate the data according to the formats required by the DoE. In reality these systems do not come with any data cleansing features and even the database structure is based on the OLTP system which is complex for querying and reporting.

3.3 Problems in OLTP database can be solved in the staging area

The UFS is among one of the oldest universities in SA. It has computerized student records from 1946 up to today. The University wants to preserve this history data, because historical data are necessary for business trend analysis. Unfortunately numerous data conversion errors were generated during the migration from the old IBM system to the new OLTP system. This led to several anomalies and a lack of data integrity in the database as shown in the following sections.

3.3.1 Missing academic program and plans

There are numerous combinations of academic programs and academic plans for which students were enrolled in the past. These combinations are no longer valid or do not exist anymore. Therefore one can not trace the degree for which the student was enrolled.

3.3.2 No uniqueness

In the Academic Plan table several academic plans were entered several times with different effective dates. One can even enter a new row with the same date and with the same entries, because no primary key constraint was enforced on any table.

3.3.3 Inconsistency in data

Student demographic information is very important for certain types of analysis like drilling down to the country, state/province and to city level. In the OLTP system there is no way for standardizing cities or other such values. For example the city name Bloemfontein was entered with 16 different spellings.

3.3.4 Spaces in mandatory columns

It was explained in previous sections that the system did not enforce integrity constraints, but Not Null constraints can be found in most of the tables. While populating history data from the old IBM system, spaces were added in such mandatory columns where no corresponding value was found.

3.3.5 Missing links

When students successfully completed their degrees, single entries were made per plan in the Academic Degree Plan and Academic Degree tables for graduation records. In several cases no corresponding entry was found in the Student Enroll table with the plan on which students received their degrees.

3.3.6 Wrong entries

The entries which the system itself identifies by picking different academic programs, plans and modules from the front-end, were wrong. For example,

the academic career for the same module was entered with different academic careers in different tables.

3.3.7 Year and semester module conflicts

The University differentiates between the module start, end or census dates for semester and year modules. The OLTP database structure allows entry for semester modules only. For example, 2061 and 2062 represents the first and the second semester of the year 2006 respectively. This scheme works for entering semester modules' start, end, and census dates, but fails when entering year modules.

Administrative staff endeavors to fix or resolve the above data errors but fail to do so, because it is too difficult to fix the errors in the OLTP system now, unless by truncating data from all of the tables and enforcing integrity constraints for future data. This method is still not cheap, because the University has to get consultancy from the international company again and a huge budget is required to do so.

In a DW staging area this data can be filtered, standardized and enriched with correct data both for internal management purposes and DoE purposes. The challenge for the DW is to cleanse the data so that it can be correlated and analyzed appropriately. There are a number of DW vendors (like Oracle, Microsoft SQL Server, Teradata, etc.) that provide data cleansing, dimensional modelling wizards and a number of other tools that are easy to use for creating a single central data repository for all the organizational data.

3.4 DW simplifies database structures

An OLTP system, such as a Student Information System, is designed to optimise transactional processing. The features which increase the efficiency of these systems are generally those which also make it difficult to extract data simply and without major impact on the transactional databases [Stevenson, 1997]. OLTP systems are optimised for fast response time of predefined transactions, with a focus on update transactions [Gatziu, 1999]. In order to optimise update operations in a transactional system, data redundancy is minimised as shown in Figure 2. This makes extracting data complex, because it is necessary to access and link a number of tables to retrieve the required data. This linking also creates a severe load on the databases.

In the case of the UFS the database contains 16,000 tables with complex relationships with non standardization in the field names which are used in joining. Out of the 16,000 tables only 200 tables are in use. The rest of the tables are empty or have only a number of entries which are not up to date. The basic reason for this huge size is due to the commercial design of the product.

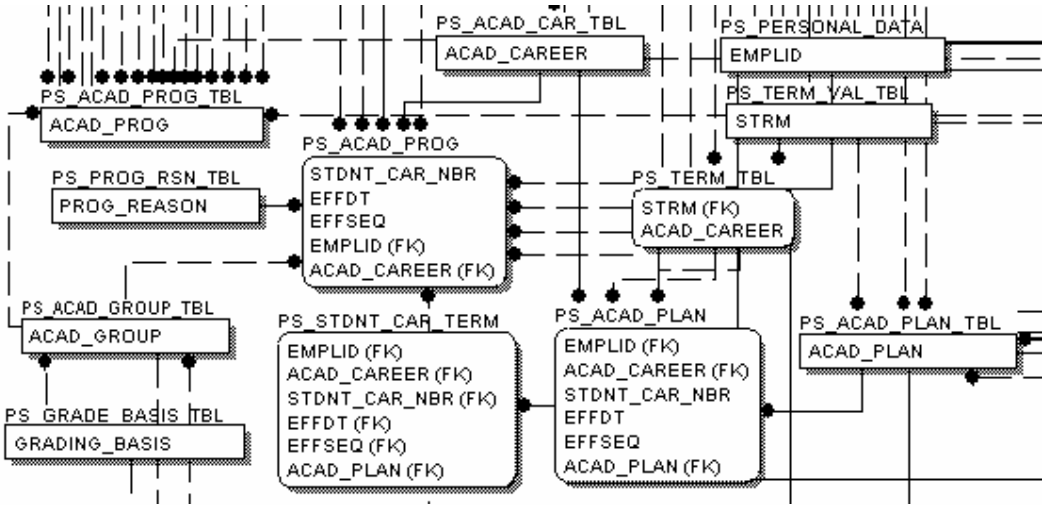

Figure 2: Entity relationship diagram of an OLTP database

A DW uses a star model based on dimensional modeling as can be seen in Figure 3. The dimensional model is simple and it is easy to extract data without writing complex queries. It is also optimized to speed up queries. Using the star model the number of tables can be reduced to 19.

3.5 Start small

In the Introduction it was indicated that the universities work with a limited budget. After several wrong and costly decisions they are wary of any new IT expenditures. Ways and means had to be found to develop a DW with minimum cost. Wierschem et al. (2003) pointed out that the development cost of a DW can be reduced by reducing the overall scope of the project. The project can also be broken down into smaller components and be developed over a longer period of time. For these reasons it was decided to develop only the Student Data Mart (SDM) for a start.

The decision of developing a SDM was based on the research of Thomas (1997) who found that institutions of higher education adopt a bottom-up approach in the development of a DW and start with a Student data most of the time. Financial data was the second most reported area, with Human Resources data a close third.

3.6 Use what is available

To cut down on costs no special hardware or software need to be purchased, but existing hardware and software can be utilized. Preference should also be given to using in-house existing staff, instead of seeking outside help [Lamont, 1999]. This advice was followed and a single person trained in DW was placed in the Planning Unit (MIS Department) of the University in order to develop the SDM. Detail requirements were gathered before and during the development process and therefore there was no chance for misunderstanding in the requirement gathering phase.

The UFS is currently using Oracle, MySql and Microsoft SQL Server for maintaining its transactional data. MySql does not come with BI tools and to use Oracle BI tools, a separate software licence is required. Microsoft SQL Server comes with built-in BI tools. Instead of purchasing the expensive Oracle Warehouse Builder and other BI tools, preference was given to Microsoft SQL Server 2005 for OLAP, DM and ETL.

3.7 Make use of summarised granularity

According to a study by Allan (2004) the grain of a student record star schema with dimensions of time (academic year, term), student demographics (one record per student), term (one record per student per term) and student matriculation (one record per student per course of study undertaken) would be "student per term per course of study."

To store student enrollments with plans and courses, two different fact tables were designed in the current SDM: Student_Enrollment_Fact (see Figure 3) with granularity "student per semester per plan" and Student_Course_Fact (not shown) with granularity "student per semester per plan per course". The main reason for designing two separate fact tables is the grain and summarization of the fact tables. The Student_Enrollment_Fact store summarized data and it, therefore, contains only 13 percent of the number of rows of the Student_Course_Fact. During the requirement gathering phase it was identified that most of the reports used for reporting and planning purposes were based on the head counts of students in a semester or a year. Student_Enrollment_Fact can answer these queries more rapidly than the Student_Course_Fact.

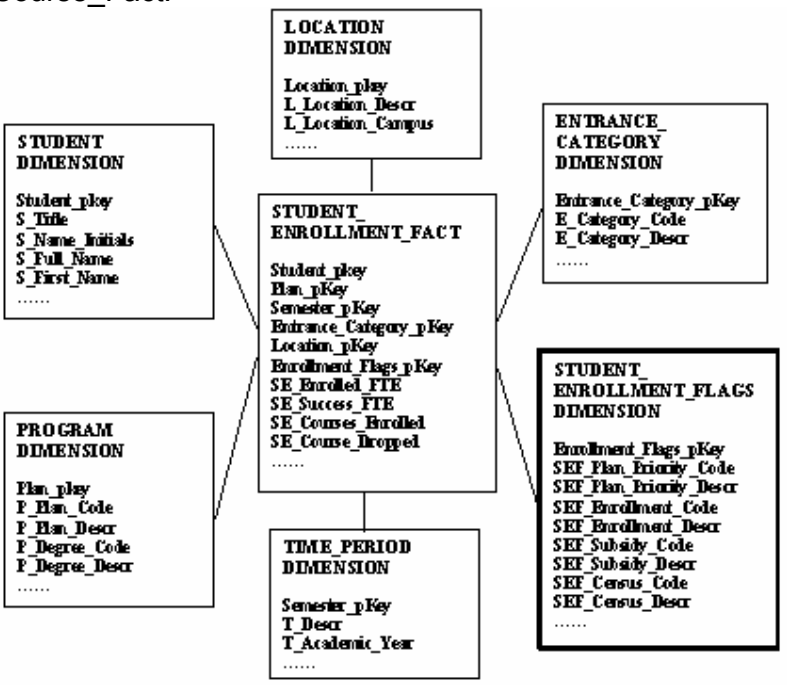

Figure 3: Student enrolment star

3.8 Make use of junk dimensions

A junk dimension is a convenient grouping of flags and indicators. It is helpful, but not absolutely required [Ross, 2003]. In this project it was concluded that junk dimensions are very useful in the SDM to enrich data for providing certain statistics, which are not available in the source OLTP system (see the Student_Enrollment_Flags in Figure 3). The DoE needs student's enrollments with the primary plan and this information is not available in the source data. A set of filters were used to find students' records which qualify for government subsidies. Some of these filters were: enrollments before or after the census date, students who had dropped out from their plans before or after the census date, approved qualifications, undergraduate students who fulfill their matriculations and students who failed or received re-assessments. This information is processed in the staging area and corresponding flags and indicators are added with the extracted data.

From Figure 4 one can see the benefits of using the Student_Enrollment_Flags dimension. Before the SDM it was very difficult to determine the students' primary plans and students who qualified for government subsidies. Now there is no need to write complex queries for finding these statistics, because the Student_Enrollment_Flags dimension makes this task very easy. One can just drag and drop data elements onto the pivot table and get the required results.

Data-Status	Current Data			
Plan-Priority	Primary Plan / Head Count			
Count of Student_Pkey			**Academic_ Year**	
Subsidy Status	**Census Status**	**Enrollment Status**	**2005**	**2006**
Non-Subsidy	After	Dropouts	1	1
		Enrollments	1	
	After Total		2	1
	Before	Dropouts	61	91
		Enrollments	536	408
	Before Total		597	499
Non-Subsidy Total			599	500
Subsidy	After	Dropouts	172	38
	After Total		172	38
	Before	Enrollments	24,700	23,431
		Transfer Enrol.	2	
	Before Total		24,702	23,431
Subsidy Total			24,874	23,469
Grand Total			**25,473**	**23,969**

Figure 4: Pivot table using student enrolment star

3.9 Get management involved

To have a successful DW it is imperative to get management involved as soon as possible. It would have been ideal to have them involved from the start. Unfortunately people in management and even in the MIS departments are not familiar with DW and what it can bring. Most probably all that they have heard is the high costs involved and the many reported failures. Because of wrong IT decisions in the past and a limited budget, they are exceptionally wary to support new unfamiliar developments.

In this case it was found that the best way to gain the support of management is to gradually make them knowledgeable of what a DW can do for them. This was achieved by developing the SDM using in-house expertise and existing resources and demonstrating to them the ease and accuracy with which information can be extracted. During the demonstrations they where given the opportunity to ask for any student statistics they would like to know. By providing the answer immediately by just dragging, dropping and drilling down in the pivot tables, their eyes were opened to new possibilities. In addition they can see that the DW provides the same and more information than the third party software systems for which they must pay annual licence fees. So when a full fledged DW needs to be developed, the chances are excellent for their forthcoming support.

4 Conclusion

Current situations in South Africa are forcing tertiary institutions to enter the DW arena. They face several difficult challenges like low budgets, dirty OLTP data and the requirement to provide statistics to the government, and at the same time have information available for in-house strategic planning and decision making. The UFS took the step to develop a Student Data Mart. Valuable lessons were learned during this process. These lessons were shared in this paper in the hope that it will be of value to other universities which still have to take the DW step.

References

Allan, R. G. (2004). Data Models for a Registrar's Data Mart, http://www.georgetown.edu/users/allanr/docs/reg.pdf

Bunting, I. (2002). "Funding" in transformation in higher education: Global pressures and local realities in South Africa. Centre for higher education transformation in South Africa (CHET): Pretoria.

Desruisseaux, P. (2000). Universities Venture into Venture Capitalism: The Chronicle of Higher Education, A44. May 26, 2000.

Donhardt, G.L., & Keel, D.M. (2001). The Analytical Data Warehouse: Empowering Institutional Decision Makers. EDUCAUSE QUARTERLY, Number 4.

Gatziu, S., & Varvouras, A. (1999). Data Warehousing: Concepts and Mechanisms, http://www.svifsi.ch/revue/pages/issues/n991/a991Gatziu.pdf

Gorla, A. (2003). Features to Consider in a Data Warehousing System: Communications of the ACM, Volume 46, No. 11.

Griffiths, S. (1995). Data warehousing – what, where, why and how. Data Warehousing Conference. Johannesburg, South Africa, 12-13 June.

Han, J., & Kamber, M. (2001). Data Mining Concepts and Techniques. USA: San Diego, Morgan Kaufmann Publishers.

Kimball, R. (1996). The Data Warehouse Toolkit: Practical Techniques for building Dimensional Data Warehouses. New York: John Wiley @ Sons, Inc.

Lamont, J. (1999).10 rules for successful data warehousing, http://www.kmworld.com/articles/printarticle.aspx?articleid=9081

Lazerson, M., Wagener, U., & Moneta, L. (2000). Like The Cities They Increasingly Resemble, Colleges Must Train and Retain Competent Managers: The Chronicle of Higher Education, A72. July 28, 2000.

List B., et al. (2002). A comparison of Data Warehouse Development Methodologies: Case Study of the Process Warehouse, http://www.ifs.tuwien.ac.at/~bruckner/pubs/dexa2002_dwh_development.pdf

Lokken, B. (2001). Business Intelligence: An Intelligent Move or Not?, http://businessintelligence.ittoolbox.com/pub/AO031202.pdf

Ponniah, P. (2001). Data Warehousing Fundamentals. New York: John Wiley @ Sons, Inc.

Ross, M. (2003). Kimball Tip#48: De-Clutter with junk (dimensions), http://www.kimballgroup.com/html/designtipsPDF/DesignTips2003/KimballDT48DeClutter.pdf

Stevenson, D. (1997). Data Warehouses and Executive Information Systems – Ignoring the Hype. http://www.eunis.org/html3/congres/EUNIS97/papers/022802.html

Thomas, C. R. (1997). Information Architecture: The Data Warehouse Foundation, http://www.educause.edu/ir/library/html/cem/cem97/cem9726.html

Trepte, K. (1997). Business Intelligence Tools, http://www.dmreview.com/editorial/dmreview/print_action.cfm?articleId=964

Wally, B. (2003). The Digital Age Story Teller, http://www.bockinfo.com/docs/bifaq.htm

Webner, R. P., & Webner, J. E. (2006).The Use And Value Of Data Warehousing In Higher Education. Retrieved May 2006 from http://www.mountainplains.org/articles/2000/general/mpa15.html

Wierschem, D., McMillen, J., & McBroom, R. (2003). What Academia Can Gain From Building A Data Warehouse: EDUCAUSE QUARTERLY, Number 1.