

UNIVERSITY DATA WAREHOUSE AND DECISION
SUPPORT SYSTEM

A CASE STUDY OF A UNIVERSITY

GEORGE GATONGU GITHAE

MSC DATA COMMUNICATION

SUBMITTED IN FULLFILLMENT OF THE REQUIREMENT FOR THE
DEGREE OF MASTER OF DATA COMMUNICATION

FACULTY OF COMPUTING AND INFORMATION MANAGEMENT

KCA UNIVERSITY

NOVEMBER, 2014

SUBMITTED TO: DR. ALICE

Declaration

This thesis is a presentation of my original work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

The work was done under the guidance of Dr. Alice, at KCA University, Nairobi

Candidate's Name.....

Signature.....

Date.....

In my capacity as supervisor of the candidate's thesis, I certify that the above statements are true to the best of my knowledge.

Supervisor Name.....

Signature.....

Date.....

Abstract

The reporting and sharing of information has been synonymous with databases as long as there have been systems to host them. Systems users expect to share information in an efficient and secure manner. However, most enterprises have a large number of systems that work separately which makes it difficult to share information in an effective manner. Today's enterprises really need to integrate data in one location to effectively utilize it and improve decision making. Data warehousing involves integrating all enterprise data in one location. Universities have their own way of measuring success different from business oriented enterprises. Also the analysis done on university data is quite different from that of a business entity. The main objective of this research is to develop a University data Warehouse and business intelligence prototype to integrate university data for report writers to efficiently analyze, report and present information to University decision makers. The process of data warehousing starts with extracting data from the University operational systems which is copied to a server. At the server it is transformed, and added with data from other sources. The resulting data warehouse becomes the main source of information for report generation, analysis, and presentation through ad hoc reports, portals, and dashboards (Rapid decision, 2013).

Acknowledgment

I would like to thank Dr. Alice Njuguna who was my supervisor for the support and guidance she offered for the period I was doing this research. I also thank my family for their understanding and support for the two years I have been studying. Lastly the Technical University of Kenya staff for answering my questions. God bless

Table of Contents

Contents

Declaration.....	1
Abstract.....	2
Acknowledgment	3
Table of Contents.....	4
List of tables.....	7
List of Figures	8
List of Abbreviations	9
1.0 Introduction	10
1.1 Background to the Study.....	10
1.2 Causes of problems in research area.....	12
1.3 Definition of terms.....	14
1.4 Problem Statement.....	16
1.5 Aim of the study.....	17
1.6 Objectives of the Study.....	17
1.7 Significance of the study	17
2.0 Literature Review.....	21
2.1 Concepts of a Data Warehouse	21
2.2 Data Warehouse Components.....	21
2.2.1 Operational Source Systems	22
2.2.2 Data Staging Area.....	22
2.2.3 Data Presentation	22
2.2.4 Data access Tools	23
2.3 Data Warehouse Modelling	23
2.3.1 Star schema.....	23
2.3.2 Snowflakes schema.....	23
2.4 Data Warehouse Methodologies	23
2.5 Data Warehouses compared to Operational Systems.....	24

2.6 Data Marts	26
2.7 Models of a Data Warehouse	27
2.7.1 Centralized Data Warehouse model.....	27
2.7.2 Independent Data Mart model.....	28
2.7.2 Federated Architecture model.....	28
2.7.3 Hub and Spoke Architecture model.....	28
2.7.4 Data Mart Bus Architecture model.....	28
2.8 University Information Management	29
2.9 Summary and Implications	31
2.10 University Decision-Making	32
3.0 Conceptual model.....	34
4.0 Methodology.....	35
4.1 Study Area.....	35
4.2 Data warehouse Lifecycle Approach.....	36
4.2.1 Project Planning	36
4.2.2 Data Track: Dimensional Modeling.....	37
4.2.3 Data Track: Physical Design.....	37
4.2.4 Data Track: Data Staging Design and Development	37
4.2.5 Technology Track: Technical Architecture Design	37
4.2.6 Technology Track: Product Selection and Installation.....	38
4.2.7 Application Track: End User Application Specification	38
4.2.8 Application Track: End User Application Development.....	38
4.2.9 Deployment.....	38
4.2.10 Maintenance and Growth.....	39
5.0 Implementation	40
5.1 Project Planning	40
5.2 Data Track: Dimensional Modeling.....	40
5.2.1 Users requirements.....	40
5.2.2 Analysis of operational source systems.....	41
5.2.3 Logical and physical model	45
5.3 Data Track: Data Staging Design and Development	47
5.3.1 Creating Common Data Connections.....	48

5.3.2 Defining a Data Flow to Extract and Load.....	49
5.4 User Interface	58
5.5 Reports.....	59
5.6 Analysis on the Benefits of the Prototype	65
6.0 Conclusion.....	67
6.1 Limitations of the study	68
6.2 Future Study.....	68
Bibliography	69
Appendix A: Interview Guide	72
Appendix B: Physical dimension and Fact Table Descriptions.....	73
Appendix C: Interview Guide	86

List of tables

Table 1: Data warehouse VS OLTP systems	26
Table 2: Data Mart VS Data Warehouse	27

List of Figures

Figure 1: Decision making levels	32
Figure 2: conceptual model	34
Figure 3: Implementation model	36
Figure 4: HR database structure	42
Figure 5: Student admission database structure	43
Figure 6: Medical database structure	44
Figure 7: Exam database structure	44
Figure 8: HR star schema	46
Figure 9: student star schema.....	47
Figure 10: BIDs	48
Figure 11: ETL for HR schema	50
Figure 12: ETL for student schema	51
Figure 13: ETL for medical schema	52
Figure 14: Lookup transformation editor	54
Figure 15: Specification of join columns and use of reference columns	55
Figure 16: Loading HR fact table	56
Figure 17: Loading fact temporary employees	57
Figure 18: Loading student fact table	58
Figure 19: user interface	59
Figure 20: Report 1	61
Figure 21: Report 2	62
Figure 22: Report 3	63
Figure 23: Report 4	65

List of Abbreviations

SSIS - SQL Server Integration Services

ETL - Extracting, Transforming, Loading

BIDs - Business Intelligence Development Studio

OLAP - Online Analytical Processing

OLTP - Online Transaction Processing

E/R - entity-relationship

DW – Data Warehouse

DB - Database

1.0 Introduction

1.1 Background to the Study

In most enterprises there are applications systems that are important in running the businesses. (PONNIAH, 2001) Noted that “ without the application systems most business would collapse this is because to efficiently receive or make payments, process orders and keep record of your clients and to maintain inventory you must use them”. (PONNIAH, 2001) Also noted that “In 1960 is when enterprises started using these systems and currently they totally rely on them and as they expand more systems are created to support business processes”. These application systems perform their tasks effectively by gathering, storing, and processing data for the daily operations. Some of these systems are also available online and can produce reports from the available data to monitor and run the business (PONNIAH, 2001).

(PONNIAH, 2001) Noted that “Businesses grew more complex in 1990s and competition became stiffer which made the decision makers desperate for information to improve their businesses”. Daily operations were done with the operational computer systems and it was perfect but the decision makers needed ready information to make strategic decisions (PONNIAH, 2001). Operational systems were not able to provide strategic information thus most businesses were forced to find new ways to get strategic information (PONNIAH, 2001). (PONNIAH, 2001) Noted that it is “in the 1990s that organizations began to achieve competitive advantage by building data warehouse systems”.

Data warehouse is a system that periodically retrieves and brings together data from several source systems into a dimensional or normalized data store. It is queried for business intelligence or other analytical activities and usually keeps years of history. Usually it is updated in batches, not when a transaction occurs in the source system (Azwa Abdul Aziz, 2012). With the possibility of a student or employee existing on more than one system and all the University information spread across the organization, implementing University Data Warehouse is the best solution. Data warehouses are normally used to provide strategic information for executives in most competitive business entities with huge amounts of data.

Business Intelligence is one of the main components in DW architecture. Business Intelligence refers to skills, processes, technologies, applications and practices which are used to support decision making (Cognos, 2008). Huge investment in data and systems is required for business intelligence to provide strategic information that advances decision making for a business. Online Analytical Processing (OLAP) is one of the main functions offered by BI tools to support multi-dimensional model of Data Warehouse. Most of DW projects have been implemented in business sector whereby information is crucial in ensuring the companies' growth and in maximizing their profits (Azwa Abdul Aziz, 2012).

Like other enterprises the University also comprises of various departments and for each of these departments there is electronic or manual system to conduct business operations. This leads to load of data stored electronically or in files. Therefore, a report writer can only get reports from each of these departments. The basic idea of the university student

performance data Warehouse is to offer reports delivering a full view of a student and instructor's in the University. This provides all information about them that would give executives the knowledge to better their decision.

1.2 Causes of problems in research area

In a Universities large number of executives who are the decision makers find it difficult to get critical information that is already in existence in the University. This is despite the availability of more and more powerful computers on everyone's desk and communication networks. As an education institution, the University every day creates data about students, staff and academic programs, most part of this data is locked up in transactional databases found in various departments.

Due to the several locations which store information a report writer is required to traverse multiple databases which could lead to incorrect reporting. This also reduces efficiency in distribution of data as more time is required to consolidate information from various sources (Rapid decision, 2013). For example, there is a possibility of having student or instructor's data in two databases without a link. The issue here is how to combine this data or identify this one person who is appearing in both databases. (Matija Lah, 2012).

Decision makers in the University must wait for a report writer from the affected department to submit their report which causes delays. Further they might need to manipulate this data to fit there need and this could lead to errors. A business entity could make losses by making decision based on erroneous reports. Availability of correct

information for decision makers whenever it's required and in the shortest time possible is of essence to any business entity (Leonard, 2011).

Data warehouse is a single database for information security officer to secure including denying or granting rights to the users. But with the operational systems Information security officers have a difficult task for controlling and managing over several databases from all types of attack and user's access might be interrupted preventing them from fulfilling their business needs (Kimball & Ross, 2013). Keeping historical data on operational systems is impossible as the users are updating and deleting data regularly (Leonard, 2011).

Operational systems use normalized relational databases which consist of a number of tables without naming conventions. This makes it hard to tell where the data required for report generation is stored (Matija Lah, 2012). The report writer is required to learn or join data from these separate databases and this is a difficult task. The general database performance is affected while a user runs queries as the system could be in a process of update or delete which could also lead to incorrect reporting. (Leonard, 2011).

Finally, the old rule, "garbage in garbage out," applies to analyses as well. Parts of the data could be missing or other parts could be wrong. Also there could be different representations of the same data.

1.3 Definition of terms

Meta data – This is not the actual data but information about characteristics of data such as names, source of data, the way it is extracted, transformations performed on the data, and data elements definitions and the complete data transfer process to its destination (Wai & Aung, 2009).

Star schema - this provides a denormalized database structure for report creation thus it's the focal point of data warehouse. It is also known as a dimensional design of a database (Adamson, 2010).

Online Analytical Processing (OLAP) – This is a technology I software engineering with fast, interactive and consistent information access mainly used for planning, decision making, managing and creating business structure to allow users answer business questions (Developing a data warehouse for a university decision, 2007).

Online Transaction Processing (OLTP) – mainly used for operational systems like placing orders or making sale they are not appropriate for decision support (Arun Sen and Atish P. Sinha, March 2005).

Data Mart – Similar to Data warehousing but limited to a departmental or specific subject need. Can be defined as data relating to a department or a specific function (Begg, 2002).

Data warehouse - Data Warehousing is a technique of bringing collectively all of a company's data from different computer systems, including those connecting to customers, employees, suppliers, product, and financials. By connecting different database together the data warehouse offers executives and managers more inclusive data set for making decision (Ofori Boateng, 2011).

Fact tables – fact tables consist of measures of specific business processes. They are also numeric and can be aggregated. The values stored are of interest to business like, sales, orders and discount amount (Ralph Kimball. Laura Reeves, 2001).

Normalization – is whereby one database table represents one entity with a goal of having complete and non-redundant schema. Data integrity and consistency is enforced by storing each piece of information once. Normalized databases ends up with thousands of tables (Ralph Kimball. Laura Reeves, 2001).

Source System - An operational system of record whose function it is to capture the transactions of the business. A source system is often called a “legacy system” in a mainframe environment (Ralph Kimball. Laura Reeves, 2001).

Data Staging Area - A storage area and set of processes that clean, transform, combine, de-duplicate, household, archive, and prepare source data for use in the data warehouse. The data staging area is everything in between the source system and the presentation server (Ralph Kimball. Laura Reeves, 2001).

Presentation Server - The target physical machine on which the data warehouse data is organized and stored for direct querying by end users, report writers, and other applications (Ralph Kimball. Laura Reeves, 2001).

Dimensional Model – An alternative to entity relationship modeling but the data is packaged symmetrically. Its goal is query performance, user understandability and resilience to change (Ralph Kimball. Laura Reeves, 2001).

1.4 Problem Statement

Traditionally, there has been less focus on revenue and profit in higher education, but with ever-escalating costs and competition, Universities and Colleges cannot ignore these financial metrics. They want to attract and retain students who align with their academic and other institutional objectives. There's a strong interest in analyzing what students are "buying" in terms of courses each term and the associated academic outcomes. Colleges and universities want to understand many aspects of the student's experience, along with maintaining an ongoing relationship well beyond graduation (Kimball & Ross, 2013).

Thus this research intends to design and implement a University data Warehouse prototype as the main source of information for report generation, analysis, and presentation to facilitate decision making by the executives. This gives the University report writers common platform which is secure with consistent data for analysis and report generation to set their business appropriately.

1.5 Aim of the study

In challenging times good decision-making becomes critical. The best decisions are made when all the relevant data available is taken into consideration. The best possible source for that data is a well-designed data warehouse (Rapid decision, 2013). The aim of this thesis is to design and implement a student performance Data Warehouse prototype within the context of a University environment.

1.6 Objectives of the Study

The main objectives of this research are:

1. To identify aspects which are the key in the design and implementation of data warehouse,
2. To develop a University Enterprise Data Warehouse and business intelligence prototype
3. To evaluate how the decision tools would assist the decision maker in taking better decision about student performance
4. To test the design of data warehouse and business intelligence prototype using the case study.

1.7 Significance of the study

Informed decision-making is required for competitive success in the new global marketplace, which is fraught with uncertainty and rapid technology changes. Decision makers must adjust operational processes, corporate strategies, and business models at lightning speed and must be able to leverage business intelligence instantly and take

immediate action (Economics, 2011). Sound decisions are based on data that is analyzed according to well-defined criteria. Such data typically resides in a Database Warehouse for purposes of performing statistical and analytical processing efficiently (Dr. Kamal Kakish, 2012).

Data analysis, report generation, and presentation of data is the most significant inspiration of developing a data warehouse. By combining all University databases in a single environment it becomes a data warehouse where users obtain information from. With the operational system a report writer is required to learn or join data from these separate databases and this is a difficult task. Also an operational database general performance is affected while a user runs queries as the system could be in a process of update or delete which could also lead to incorrect reporting. (Leonard, 2011).

Organization's information is more accessible using a data warehouse. Data warehouse contents are clearer, easy to maneuver and can be accessed faster. The content are well labeled and it's easier to get what you want in the shortest time possible with zero wait time. There are no fixed limits or boundaries to the requirements (Matija Lah, 2012). Uptime and availability are the main priorities of the source systems. Source systems queries are account-based which form part of the normal transactions and their demands are restricted on the legacy system. Reporting from source systems burdens them and they keep little historical data. The data warehouse is queried in broad and unexpected ways unlike the source systems and it keeps historical data (Matija Lah, 2012)

A data warehouse makes organization's information more consistent. Information from one part of the organization can be matched with information from another part of the organization. If two measures of an organization have the same name, then they must mean the same thing. Conversely, if two measures don't mean the same thing, then they are labeled differently. Consistent information means high-quality information. It means that all of the information is accounted for and is complete. Anything else is a compromise and therefore something that we must improve (Matija Lah, 2012).

A data warehouse is an adaptive and resilient source of information. The data warehouse is designed for continuous change. When new questions are asked of the data warehouse, the existing data and the technologies are not changed or disrupted. When new data is added to the data warehouse, the existing data and the technologies are not changed or disrupted. The design of the separate data marts that make up the data warehouse must be distributed and incremental. Anything else is a compromise and therefore something that we must improve (Matija Lah, 2012).

A part from tables for reports auditing tables may also be included in a data warehouse. This would record all updates, when they were made and fact and dimension rows affected. You can also measure performance by calculating time taken for each load and take action. However this audit information is only useful when done regularly (Matija Lah, 2012).

Security is enhanced in data warehousing as there is only one database for which to secure and grant or deny users rights. Thus the Information security officers will also have an easy task for controlling and managing over enterprise data from all types of attack. Users therefore can have uninterrupted access to the system whenever required to. The small databases found in an enterprise also have the chance to have their data in the data warehouse (Leonard, 2011).

Significantly, this research will also benefit the academicians and the University managers and executives in general as it will add to the academic resource pool available on the development of a university data warehouse. Research findings will be published in a reputable journal for easier availability of the information when needed.

2.0 Literature Review

2.1 Concepts of a Data Warehouse

The main aim of data warehouse is to build the data in a befitting way for access and use in an efficiently and effectively (Pias M.M, 2008). As stated by (M, 2001), consistency of information is the main responsibility of a data warehouse. A combination of variety of tools like querying, reporting and analysis tools is the most effective way of linking information. Data warehouse main aim is to integrate data into a single repository (M, 2001). Presently, data warehousing plays a major role in most organizations. To obtain competitive advantage business entities need to integrate data from the different sources and convert it into valuable information (del Hoya & Lees, 2002). Data warehouse is a combination of technologies in decision support to enable decision makers to come up with the best strategies within a short time (Dayal & Chaudhuri, 1997) Intelligence, understanding, adaptability and profiting from experience are very important when developing a data warehouse (March & Henver, 2007). Also users of the data warehouse should be able to collect information by identifying various situations and the reasons why they occur. Finally, a data warehouse should allow users in an organization to apply the relevant knowledge and to measure the effect of decisions made after a period of time (March & Henver, 2007)

2.2 Data Warehouse Components

(Kimball & Rosi, 2002) Identifies that a data warehouse environment constitutes of few components with each component providing a specific function these are:

- Operational source system

- Data Staging Area
- Data Presentation Area
- Data Access Tools

2.2.1 Operational Source Systems

This is primarily concerned with processing performance and availability. The source systems only maintain little amount of historical data or none at all. The source systems queries are narrow and they operate as part of the normal transaction flow. These queries act according to the demands on the operational system (Kimball & Rosi, 2002).

2.2.2 Data Staging Area

This is a temporary storage of data (Kimball & Rosi, 2002). It is also referred to as the Extract Transformation Load (ETL). This is where data extraction, transformation and loading occurs. First step when moving data to the data warehouse is extraction. The second step is to cleanse the data (by fixing misspellings, fixing missing elements, or putting it into standard formats), combine various sources of data, de-duplicating data, and to allocation warehouse keys happens at this stage (Kimball & Rosi, 2002). Finally is to load data to the data warehouse presentation area.

2.2.3 Data Presentation

This is where users can have access to the available data which has been organized and stored in the data warehouse. From the data warehouse the decision makers using data access tools can see the data and gain access. (Kimball & Rosi, 2002), states that data presentation is also known as a combination of various data marts which presents a specific business process.

2.2.4 Data access Tools

This as the final component of a data warehouse gives decision makers capabilities to control the presentation area. This involves data mining or some simple queries (Kimball & Rosi, 2002).

2.3 Data Warehouse Modelling

This identifies various things including the data warehouse data mart, data warehouse view, data warehouse schema, data required for decision support and information required by users (Borysowich, 2007).

2.3.1 Star schema

Star schemas consists of facts and dimension tables. Dimension tables are directly linked to the fact table. In a data warehouse star schema represents the multidimensional data model (Dayal & Chaudhuri, 1997). The fact table consists of keys and measurements while the dimension contains the attributes related to each dimension (Mih, 2010).

2.3.2 Snowflake schemas

The same as the star schema but, snowflake schemas dimension tables are normalized and connected to other dimension tables (Dayal & Chaudhuri, 1997).

2.4 Data Warehouse Methodologies

The top - down method and the bottom - up method are the basic data warehouse design methodological approaches used (Golfarelli. m, 2009). Top - down method relies on user requirements to plan, analyze and develop the data warehouse. This method is associated with high cost, difficulty in analyzing and collecting all specific needs of all the business entity and added development time. In the bottom - up method the data warehouse is built

and then several data marts are created. Bottom – up method takes a fraction of the whole system and this leads to a system risks (Golfarelli. m, 2009). The bottom-up method is the widely accepted method for most users.

2.5 Data Warehouses compared to Operational Systems

The key difference between data and warehousing operational systems is that operational systems are designed to support transaction processing (OLTP) and data warehousing systems are designed to support online analytical processing (OLAP). Operational systems users deal with one record at a time and the operational tasks are done repeatedly. Data warehouses are used to handle huge capacities of data for analysis to help decision makers make timely decisions (Kimball & Rosi, 2002). The following table displays the common variations of online transaction processing system (OLTP) and a data warehouse (Kadlec, 2005).

	OLTP	DATA WAREHOUSE
DATA	Real time data from line of business applications such as OE, ERP, accounting, inventory, control, etc	Historical data, migrated from the OLTP systems, that is aggregated and scrubbed for analysis and reporting purposes to determine trends and patterns critical for the organizations
Purpose	Supports day to day business operations	Supports tactical and strategic decision making
Front end applications	Web based or desktop applications with data entry textboxes and searching functionality to following the business cycle	Analytical tools with charting or graphing interfaces with the ability to review aggregate data and drill into the details
Database size	Small to medium sized databases typically 1 to 100 gigabytes	Large databases, 100 gigabytes to multiple terabytes that are gathered from the OLTP systems staged on a separate server where the data is scrubbed and aggregated before the final load to data warehouse.
Users	Large number of users	Small number of users, traditionally but a recent trend of expanded user base with specialized data access

Concurrency	High number of simultaneous transactions	Low number of transactions, but growing with the user community
Transactions	Short milliseconds to seconds with a limited amount of data returned to clients	Long—minutes to hours with functions (SUM, AVG, STDEV etc.) executed that are exploratory by nature.
Design	Normalized relational model	Star or snow flake model with pre-aggregations
Data entry	Direct updates from users	Data migrated from OLTP systems

Table 1: Data warehouse VS OLTP systems

According to (H, 2005), the use of information technology is faced with various challenges, these includes credibility of data and how to transform data into information. Data credibility is caused by discrepancy in time, lack of common sources of data and issues with external data which makes departmental reports incompatible. Productivity becomes a when analyzing data from various departments as reports are extracted from all operational systems.

2.6 Data Marts

Data marts are the small units that make up a data warehouse (Bonifati, Catteneo, Ceri, Fuggetta, & Paraboschi, 2001). Data mart is a collection of subject areas organized for decision support based on the needs of a given department” (H, 2005). Each data mart is created depending on the departmental needs therefore all departments know how there data mart should look like.

Data Mart	Data Warehouse
Departmental	Corporate
High level of granularity	Low level of granularity
Star join structure	Star join/Snowflake structure
Modest amount of historical data	Robust amount of historical data
Technology optimal for access and analysis	The best technology for holding, and management of big volumes of data
The structure of each department is different	The structure is the best for corporate data

Table 2: Data Mart VS Data Warehouse

2.7 Models of a Data Warehouse

There are five different data warehouse architectural designs which provide guidance when creating a new design (PONNIAH, 2001). These are:

- Centralized data warehouse architecture model
- Independent data marts (IDM) model
- Federated architecture (FED) model
- Hub and spoke model
- Data marts bus architecture model

2.7.1 Centralized Data Warehouse model

This considers business information requirements where atomic level data is maintained in the third normal form although summarized data will be stored at times. Separate data marts cannot be developed with this architecture (PONNIAH, 2001).

2.7.2 Independent data Mart model

These are developed to meet individual unit in the organization (Ariyachandra & Watson, 2005). Every data mart is created independently from the operational systems autonomous teams who deploy tools, software, hardware, and processes. This architectural model has difficulties with consistent data and because of the use of dissimilar measures and dimensions analyses of business data across the data marts is almost impossible. Other problems identified by Marco were: redundant data and processing, scalability and lack of integration (Marco, 2002).

2.7.2 Federated Architecture model

The existing architectural decision support structures are left in place (e.g., operational systems) (Ariyachandra & Watson, 2010). Data integration in the warehouse is through logical or physical methods using share keys, global Meta data, and distributed queries among others. (Ariyachandra & Watson, 2010) Stated that this architecture is more favorable with business entities where complex decision support systems exists.

2.7.3 Hub and Spoke Architecture model

This is the same as to centralized architectural model because it contains atomic level data which are normalized into third normal form. Independent data marts are attached to this centralized data warehouse from where they acquire data. The centralized data warehouse act as a hub and the independent data marts act as spokes. These independent data marts are developed for various purposes of the business entities (PONNIAH, 2001).

2.7.4 Data Mart Bus Architecture model

This is designed depending on the business decision maker requirements. Initially the design consists of dimensions and measurements, measurement data is added later on. Data

charts are organized in star schemas which consist of atomic and summarized data (PONNIAH, 2001).

2.8 University Information Management

A study done at one of the leading universities in Kenya (Moi University) revealed that the university had embraced information technology as the main tool for information management and communication to support various business transactions in the institution (Agnes, 2012). But despite this the management of electronic records gathered from various departments was a neglected area of the institution record keeping system. Various factors highlighted as the main contributors to this were lack of policies and procedures to guide the management of electronic records and lack of knowledge and skills in records management and in particular electronic records management.

To a large measure a university depends on the availability of, and access to, university records by the university community and other stakeholders (Agnes, 2012). Therefore records are critical to University administration without which operational processes and functions cannot be executed. Without these records the decision making process of the university will be incapacitated.

The various strategies used to store electronic records at Moi University were: One for future reference all (100%) respondents printed and stored hardcopies of electronic records that they used. This was to make sure the electronic record was available when needed. Second (92.9%) of the respondents used storage devices. Thirdly (71.4%) of the respondents

created personal folders on computers to store electronic records. Some (47.6%) of the respondents did back up information they created and received electronically while (2.3%) respondent used microfilming as a means of storing electronic records. She also established that it was a personal initiative for (71.4%) of the respondents to store electronic records in personal folders with names only known to them. The study also noted that management of computer files was under no controls to prevent their misuse (Agnes, 2012).

Agnes Peninah Nasieku also noted that records creators could forget the names and the location of the files and folders they had stored on the computers so they could not retrieve them. According to the study most employees were not even aware of the requirements for naming and storing files for easier retrieval. The staff who backed up their records also did it as a personal initiative and there was no guidance on when to back up the data (Agnes, 2012).

The study also noted that on most CDs where information was stored were unreadable due to scratches caused by careless storage in desks, drawers and on window panes. Other group of staff did not label the storage media this made it difficult to know what information was contained in the devices unless one scrolled through the contents. This took a lot of time especially when the information was urgent. This shows how unreliable the information for the (71.4%) of the respondents is who reported that they used personal folders to store electronic records in their personal computers (Agnes, 2012).

2.9 Literature Summary

The goal of a data warehouse is to gather data from various operational systems and storing it in a single repository. Data warehousing consists of various elements which include: Operational source system, Data Staging Area, Data Presentation Area, Data Access Tools. As discussed in the earlier chapters there are different data warehouse modelling methods each with its pros and cons but the one that most developers use is the multi-dimensional or data cube approach. Modelling techniques used to represent multidimensional data are the star schema and snowflakes schema. Top- down approach and bottom-up are the most popular methodological approaches for developers.

Data warehouse and operational systems are totally different two types of systems. Operational systems are used for transaction processing while data warehouse systems are used for online analytical processing. Other variations are on how data is used the size of the database, the type of transactions done among others.

In a University there are different sources of data in big volumes. Thus a data warehouse is required to gather, organize and distribute University data in a data warehouse. Decision making in a university is a very difficult task this is due to lack of means to gather data from various sources. The benefits of integrating University data includes improved access, faster and clear decisions including quality services. Implementing a university data warehouse is among the paramount ways that the executives can use to get strategic information for informed decision making.

2.10 University Decision-Making

Organizational decision making levels are: the top level, tactical and operational level. Decision making level involves strategic decision making by the executives by focusing on the goals of the business (Johns, 2002). Most of these decisions involve courses to be offered, location, among others. Tactical decision making relates to the tactical departments of the organization which may include student admission and fees payment (Johns, 2002). The last level involves the daily operations of the organization among them providing security, ordering supplies, medications and processing bills.

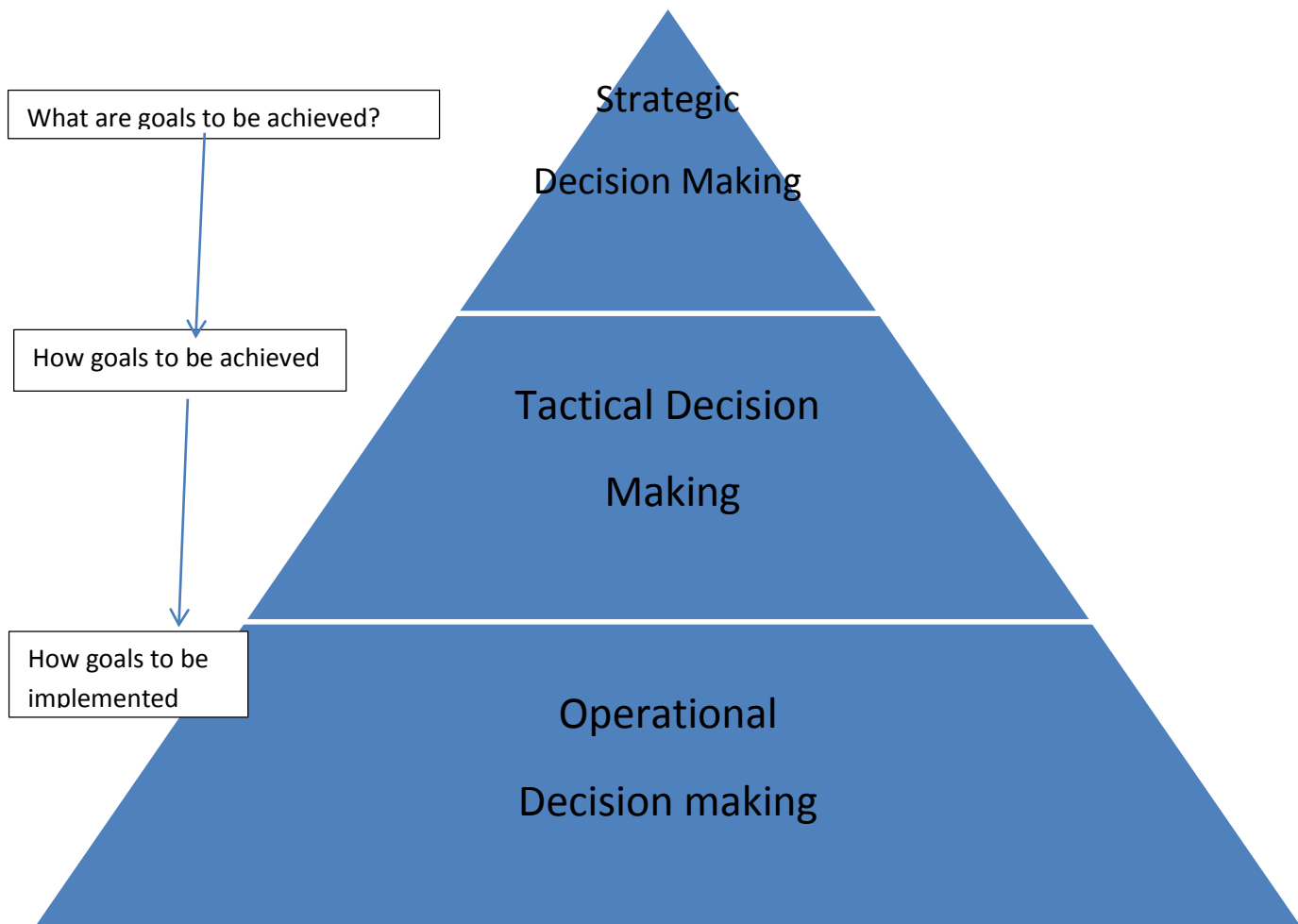


Figure 1: Decision making levels

3.0 Conceptual model

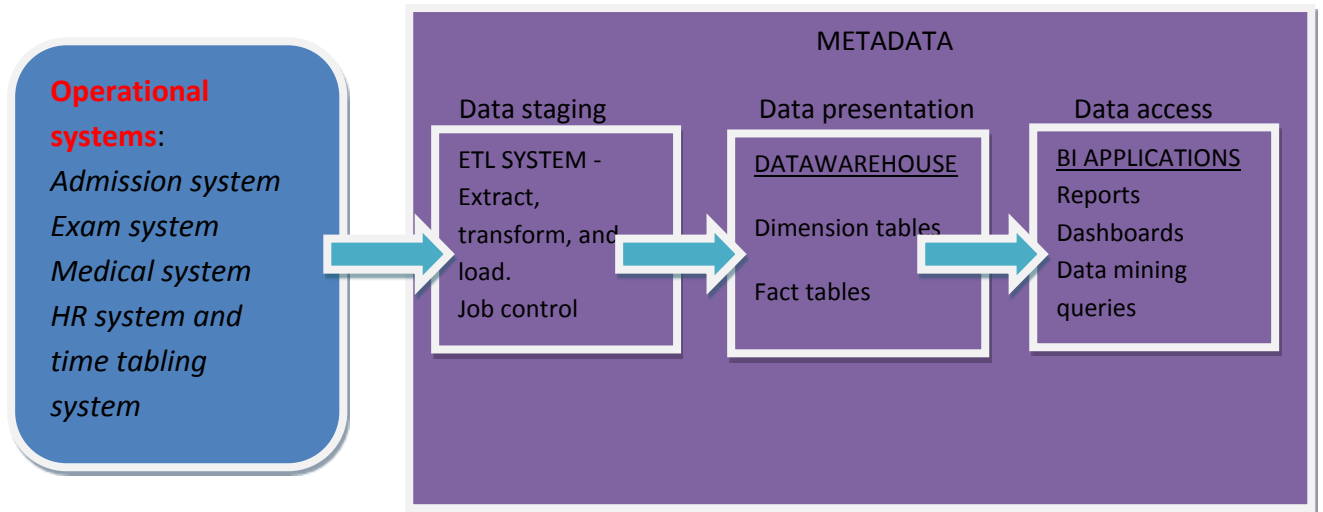


Figure 2: conceptual model

4.0 Methodology

4.1 Study Area

The case study will be done at the Technical University of Kenya one of the higher education institutions in Kenya. In the first phase, interviews are conducted to better understand the decision making process at the University. Two report writers are interviewed at the Technical university of Kenya. The guide to the interview is shown in the Appendix A.

Secondly, is to analyze data warehousing literature and mainly focused on: enterprise systems, decision support systems, design methodologies Data Warehouse development techniques, and use of Data Warehousing and business intelligence in the universities. This is to ensure the most recent literature is used to design the data warehouse.

Thirdly a second interview is carried out to get feedbacks from the report writers at the technical university of Kenya. Again the two staffs we interviewed at the initial stages will give their views of the prototype. The guide to this interview is in Appendix B.

The data warehouse is designed with an SQL Server 2008 database and Microsoft visual web developer 2010 express. The prototype data will be based on 30 numbers of students, 5 academic staff, and 2 courses.

4.2 Data warehouse Lifecycle Approach

The design of a data warehouse goes through a sequence of high-level tasks required for effective data warehouse design, development, and deployment according to the data warehouse life cycle tool kit by Ralph Kimball (Ralph Kimball, Laura Reeves, 2001). These main phases are discussed below.

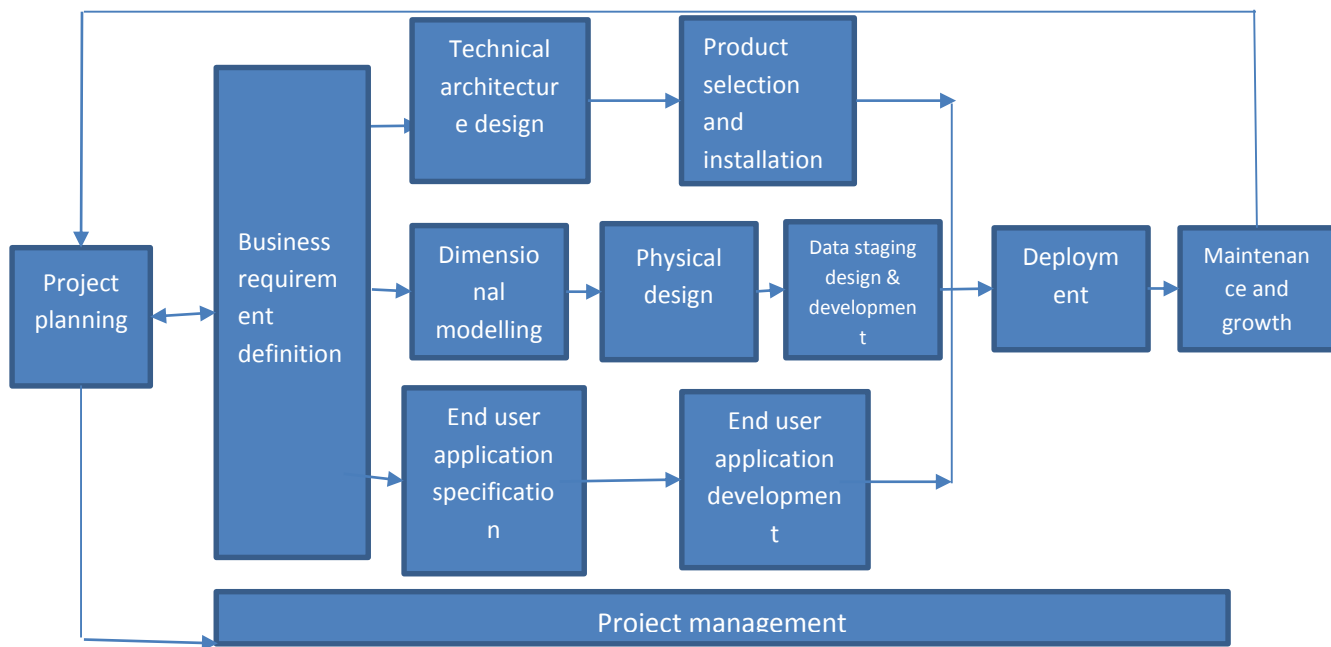


Figure 3: Implementation model

4.2.1 Project Planning

The Lifecycle begins with project planning which addresses the definition and scoping of the data warehouse project, including readiness assessment and business justification. Due to the high costs associated with most data warehouse projects this should be done before the project commences. Thorough understanding of business and users requirements is the key to success of a data warehouse. Designers must put into considerations the vital features driving the business to successfully determine business requirements. The basis of

technology, data, and end user applications of a data warehouse. Depend on business requirements

4.2.2 Data Track: Dimensional Modeling

A dimensional model is developed by pairing business requirements and the analysis of the source system. Dimensional modelling identifies fact tables, dimensions, attributes table structures and primary and foreign key relationship. Finally source to destination mapping is done.

4.2.3 Data Track: Physical Design

This is the physical design required to support the logical database design Here naming standards are defined and the database environment is set up including indexing and partitioning.

4.2.4 Data Track: Data Staging Design and Development

This involves data extraction, transformation, and loading. To enhance data quality issues data extraction process is the key.

4.2.5 Technology Track: Technical Architecture Design

In a data warehousing various technologies are integrated. Architectural framework and vision is established by technical architectural design. Various factors are considered for architectural design, business requirements, current technical environment, and planned strategic technical directions.

4.2.6 Technology Track: Product Selection and Installation

Architectural components and tools should be selected and evaluated these include: database management system, hardware platform, data staging and access tools. Once selection and evaluation is done they are installed and tested to guarantee good integration of a data warehouse environment

4.2.7 Application Track: End User Application Specification

A standard end user applications should be selected describing the report template, user driven parameters and required calculations. This is to ensure that the development team and system users agree on the system to be delivered.

4.2.8 Application Track: End User Application Development

The development of end user applications involves configuring the tool metadata and constructing the specified reports. Optimally, these applications are built using an advanced data access tool that provides significant productivity gains for the application development team. In addition, it offers a powerful mechanism for business users to easily modify existing report templates.

4.2.9 Deployment

Deployment signifies merging of technology, data and the end user application that user accesses from the computer. For these to fit together properly extensive planning is required. The system user should be educated on how these aspects are integrated and communication feedback established before users can access the data warehouse.

4.2.10 Maintenance and Growth

The need to provide users with support and education.

5.0 Implementation

5.1 Project Planning

The technical university student performance data warehouse prototype scope is limited to student performance management in the university. The goal is to track and get integrated reports of students and instructors from the existing operational systems for the period they have been on campus. The data warehouse will be developed on a windows 7 platform with 2.4 GHZ dual core processor, 2GB memory and 500 GB hard disk.

5.2 Data Track: Dimensional Modeling

5.2.1 Users requirements

In order to accomplish a successful data warehouse prototype, user requirements must be defined by report writers at the technical university of Kenya. Comprehensive data analysis of the available source systems must be performed. Interviews are carried out to determine how data should be integrated and manipulated to meet user requirements. At the technical University most of the staff have a computer at their desk. They also have skilled Information Communication and Technology staff but despite this most departmental functions are done manually. The university processes linked to student administration are categorized as follows:

- Admission system: This is automated system manages student admitted to the university
- Exam: This manages exam records for each unit mostly done with Microsoft excel

- Health center: This is done manually and manages medical records for staff and students
- Human resource system: This is an automated system and manages employees of the university
- Time tabling: this is done using Microsoft Excel and is used to allocate units to lecturer and the venue for those classes

Some of the issues gathered from the interview that required automation and integration to a data warehouse were:

- A report of the units taught by a specific lecturer, the number of students in that class and the mean grade achieved.
- A report of the total number of students and the number of instructors in a department, this would reveal student lecturer ratio and the ratio between boy and girls
- A report of all student who failed and those who had medical cases to determine the cause of the failure
- Fess status of student before sitting for their exams or having a checkup at the health center
- Show pass rate for both boys and girls in various courses or units.
- Graduation pass rate

5.2.2 Analysis of operational source systems

Since the University did not offer electronic data that is stored in their operational systems and most of it was not electronic, Microsoft access databases were created with sample data. The first Microsoft Access databases that is part of this prototype is in the HR

database which portrays university staff information and is very vital to any university. This database keeps track of all university staff including teaching and non-teaching staff there job history and departments and location where they work. The database is in a Microsoft access database hosted on a Windows operating system.

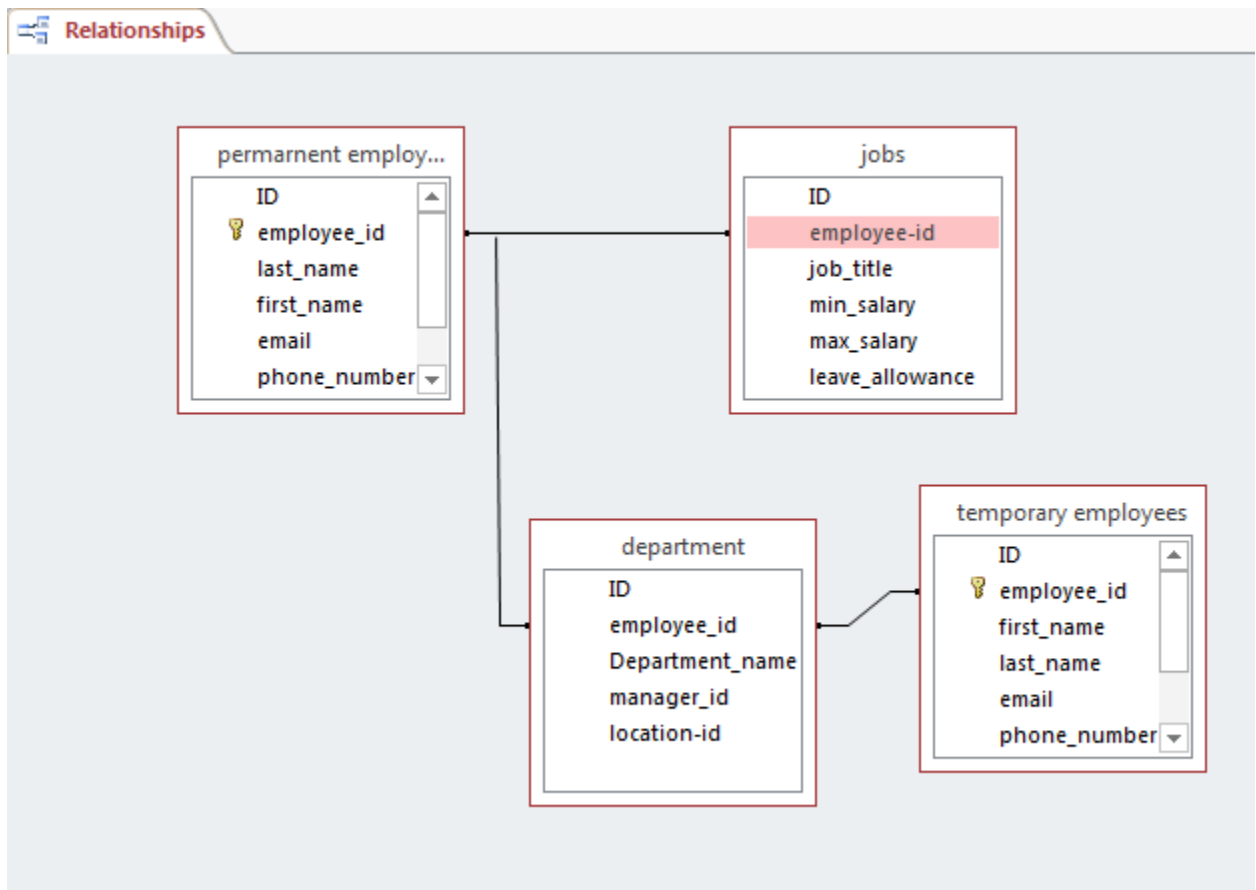


Figure 4: HR database structure

The second source system data that is part of this prototype is the admission database which portrays all students in the university. This database keeps track of all students in the university including, courses and departments. The database is in a Microsoft access database hosted on a Windows operating system.

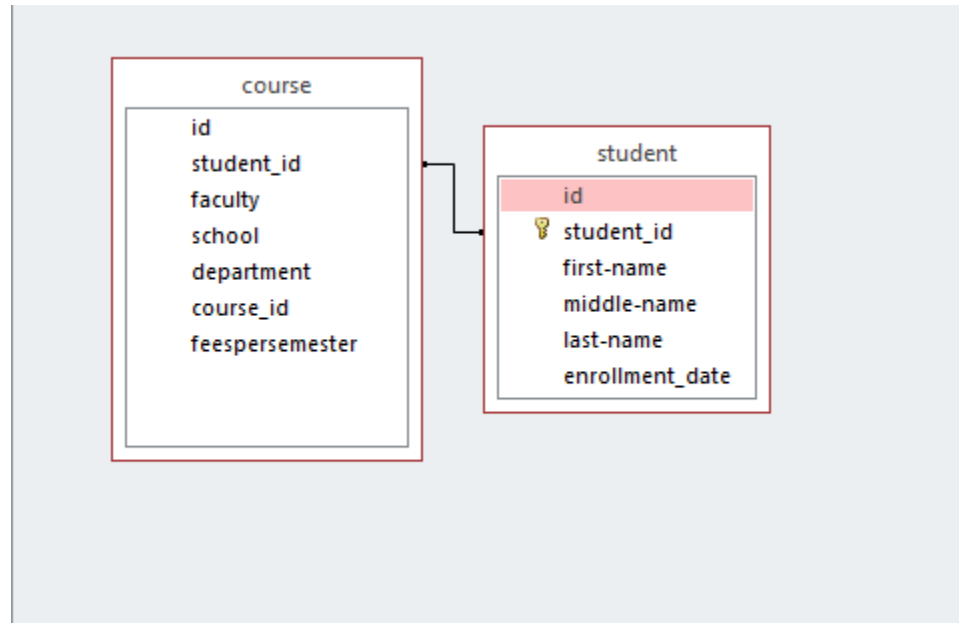


Figure 5: Student admission database structure

The third source system data that is part of this prototype is the medical database which records information about students who visit the clinics in the university. This database keeps track of all patients who visit the clinic including diagnosis and treatment. The database is in a Microsoft access database hosted on a Windows operating system.

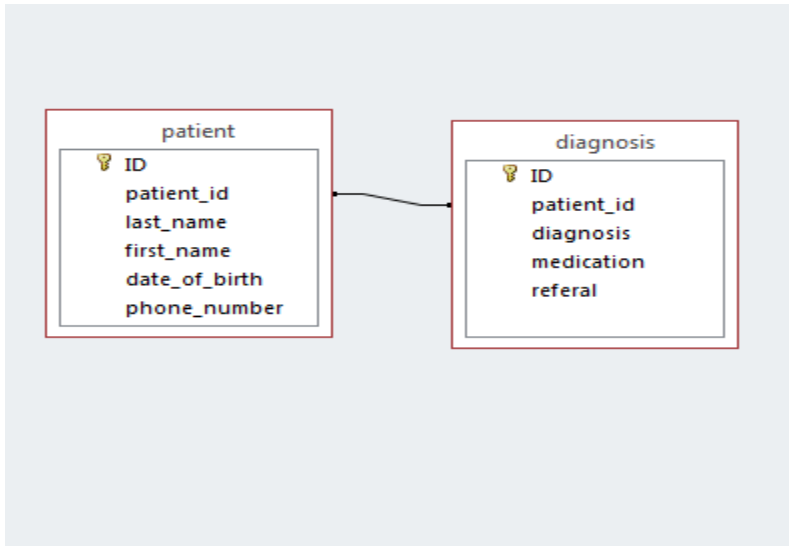


Figure 6: Medical database structure

The fourth source system data that is part of this prototype is the exam database which records information about students, marks and grades achieved in each unit. The database is in a Microsoft access database hosted on a Windows operating system.

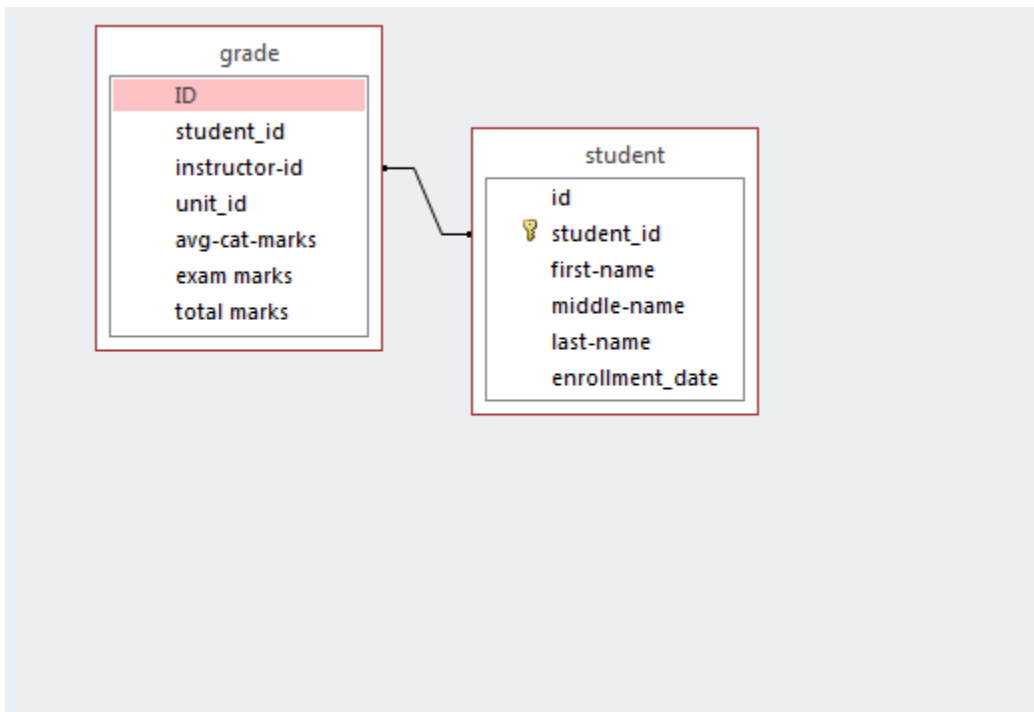


Figure 7: Exam database structure

5.2.3 Logical and physical model

With the understanding of the administration requirement a logical model development starts. The first step is to determine the measures for the solution. Measures are the numerical values that are required for analysis. Secondly determine which attributes to include with each dimension. Attributes are the individual elements in a dimension (corresponding to columns in a table) that you can use to group or filter data for analysis, such as employee-id or Year in the Date dimension.

There will be a star schema for each transactional database. The dimension tables are the same as source database tables, but with dimension ID. This is an identity-valued field which increments by one known as a surrogate key (except for the Date dimension). One benefit of using surrogate keys is that you can combine data from multiple systems without the risk of duplicate keys. For easier reporting Fact table consists of a unique Fact ID measures together with all the related Dimension IDs. Lookup tables will also be used to add data to the fact tables. For this prototype data is loaded to empty destination tables although the data warehouse loading process is incremental.

A Date dimension is also included because business analysis often compares changes in measures by date, week, month, quarter or year. Date dimension doesn't usually map to a source table. Instead, a script is used to generate the data to load records into the table. Now that logical modeling is complete, physical tables are created that the ETL processes

will load and a host database for these tables. The two star schemas are built in SQL 2008 server.

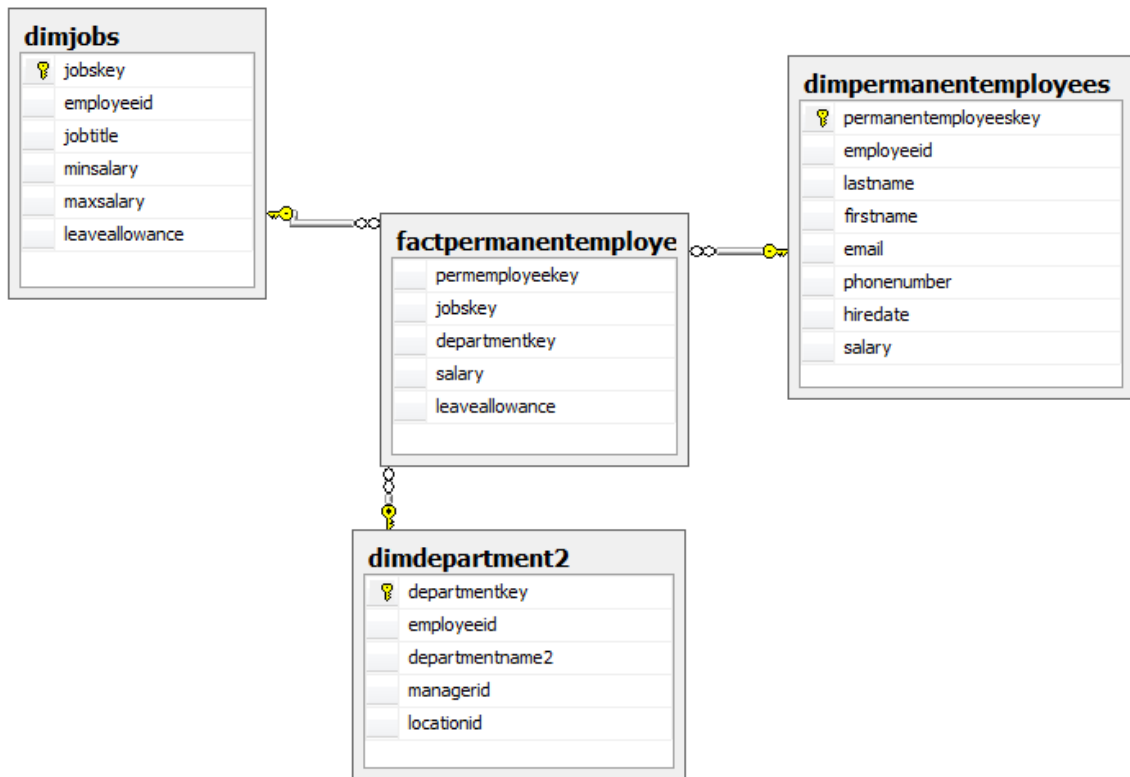


Figure 8: HR star schema

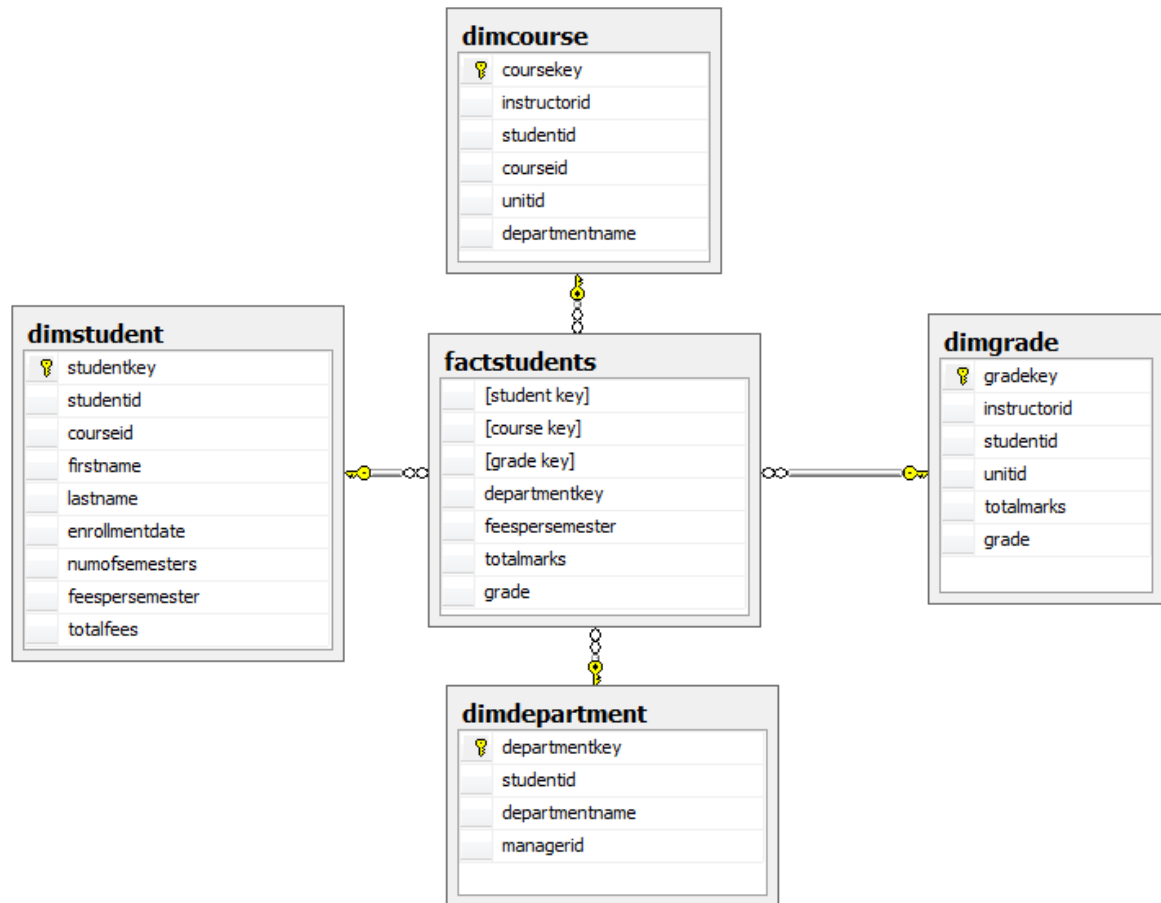


Figure 9: student star schema

5.3 Data Track: Data Staging Design and Development

Data staging process has three major steps: extraction, transformation, and load. ETL includes all the technological processes by which data is extracted from data sources, transformed, and then loaded into the data warehouse. The first step in developing an ETL process is to create a new project in Business Intelligence Development Studio (BIDS). To start BIDS, go to Start\Programs\Microsoft SQL Server 2008\Business Intelligence

Development Studio and select File\New Project. You should see the New Project template shown below.

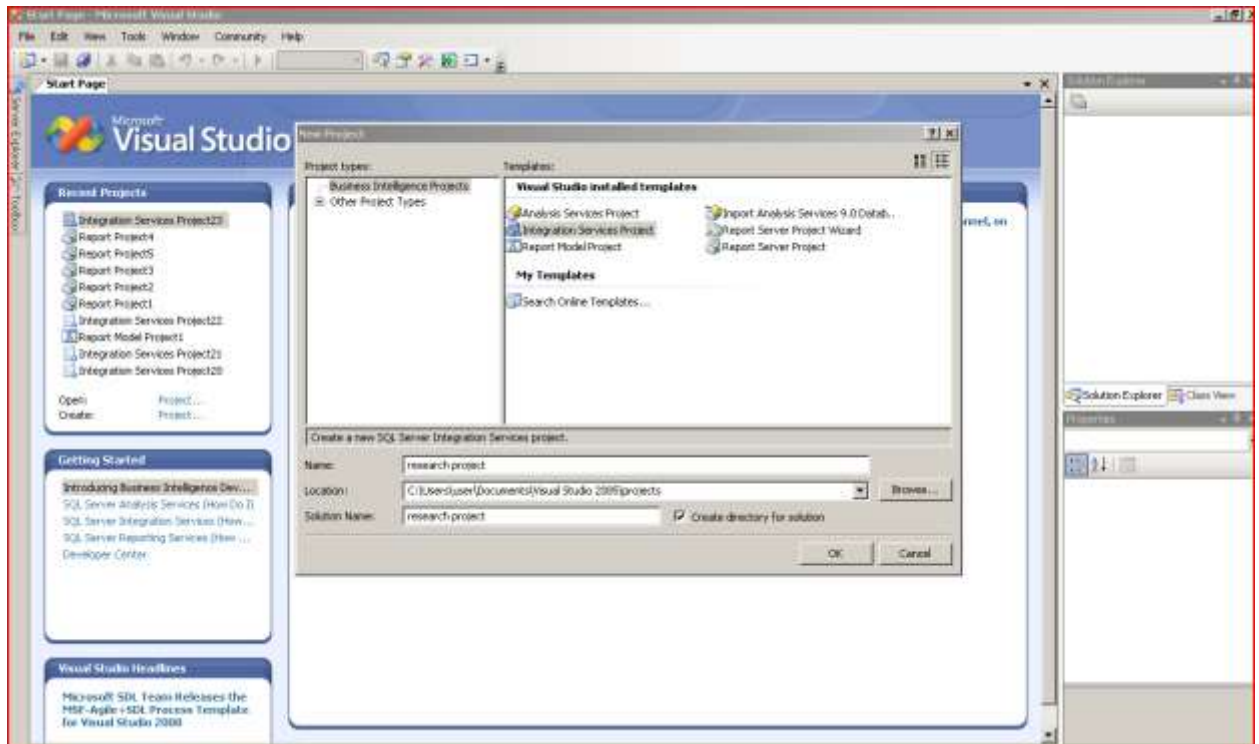


Figure 10: BIDs

5.3.1 Creating Common Data Connections

The second step is to create data sources connections. You can define a data source connection once and then reference it in one or more SSIS packages within the solution. It required that data is extracted from the transactional database table's data into the University data warehouse tables. Therefore source and destination data connections are created.

5.3.2 Defining a Data Flow to Extract and Load

SSIS includes a data flow task that encapsulates everything needed to implement the ETL. By dragging a data flow task from the Toolbox onto the Control Flow designer's surface. Then double-click the Data Flow task in the designer. BIDS now displays the Data Flow designer. The extract portion needs to query the transactional database tables. To set up this task, drag an OLE DB Source component from the Toolbox onto the Data Flow designer surface and then rename the OLE DB Source component the name of the table to extract from.

Next is to define the load portion package to load into the data warehouse. Drag a new instance of the OLE DB Destination component onto the Data Flow designer surface and rename with the data destination table in the data warehouse. Then click the OLE DB Source component and drag the green arrow that appears on the OLE DB Source to the load Destination component to connect the two components.

Next configure each component so that SSIS knows how to extract and load the data. Right-click the extract department source component and select edit. Make sure source database connection manager is selected. Then expand the Name of the table drop-down and select the department table. Finally, click the Mappings tab to confirm that the mappings are correct. Click OK to confirm the mappings.

Next right click data Destination component, and select Edit. With the OLE DB Destination Editor open, make sure the data warehouse is selected as the OLE DB connection manager. Then expand the Name of the table drop-down and select the data warehouse table to load the data. Finally, click the Mappings tab to confirm that the mappings are correct. Click OK to confirm the mappings. The ETL package is now complete. Repeat this process for all source and destination tables from the transactional Databases. To test the package, open the package and press F5. Below figures shows a successful ETL's.

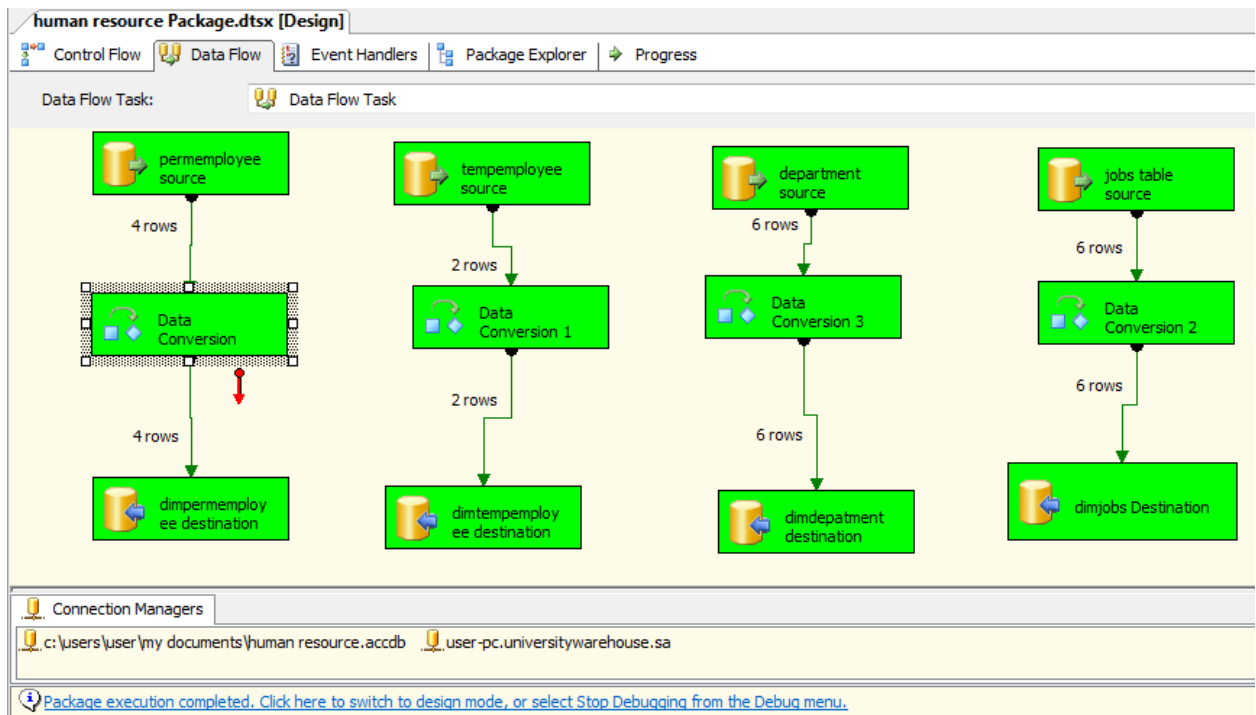


Figure 11: ETL for HR schema

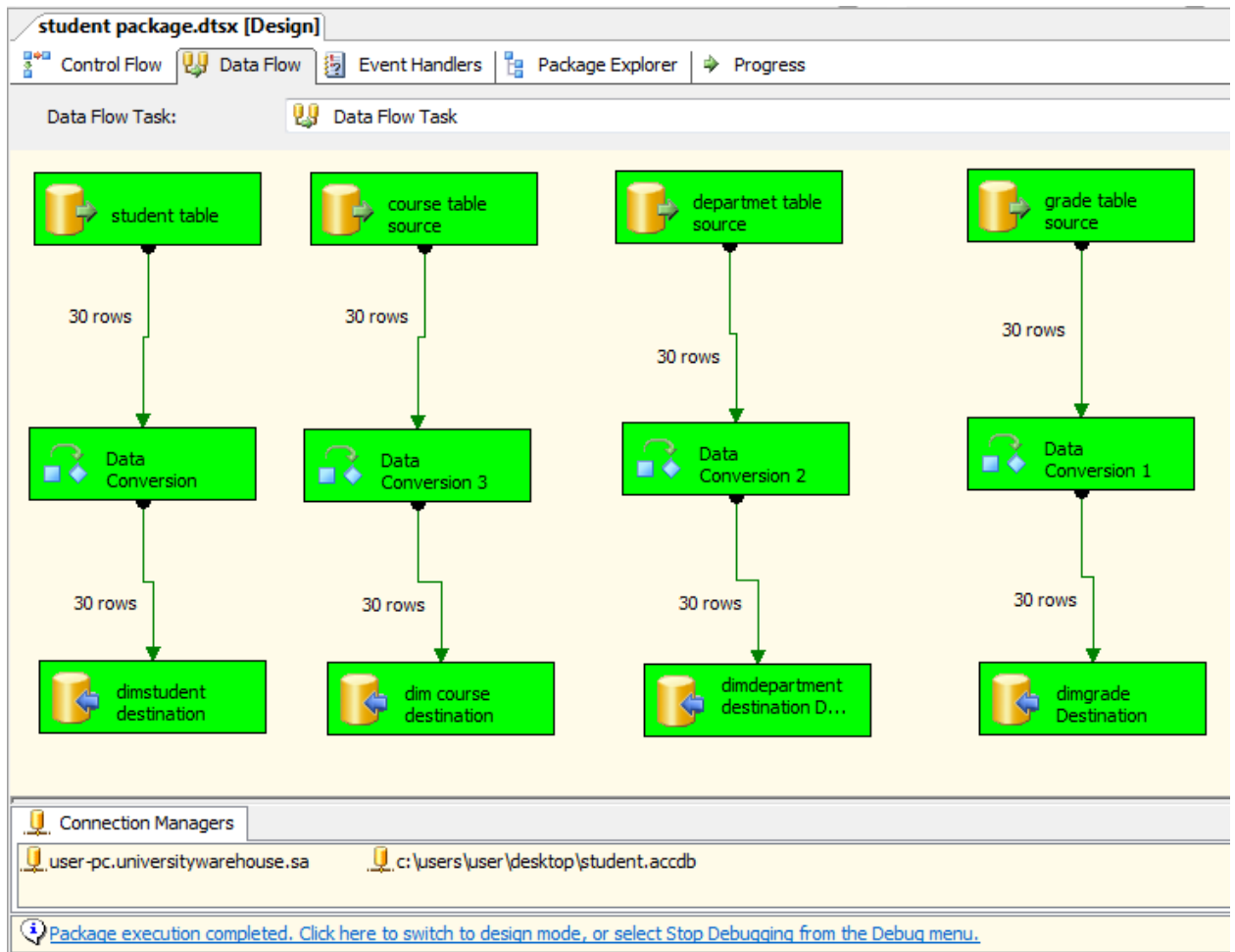


Figure 12: ETL for student schema

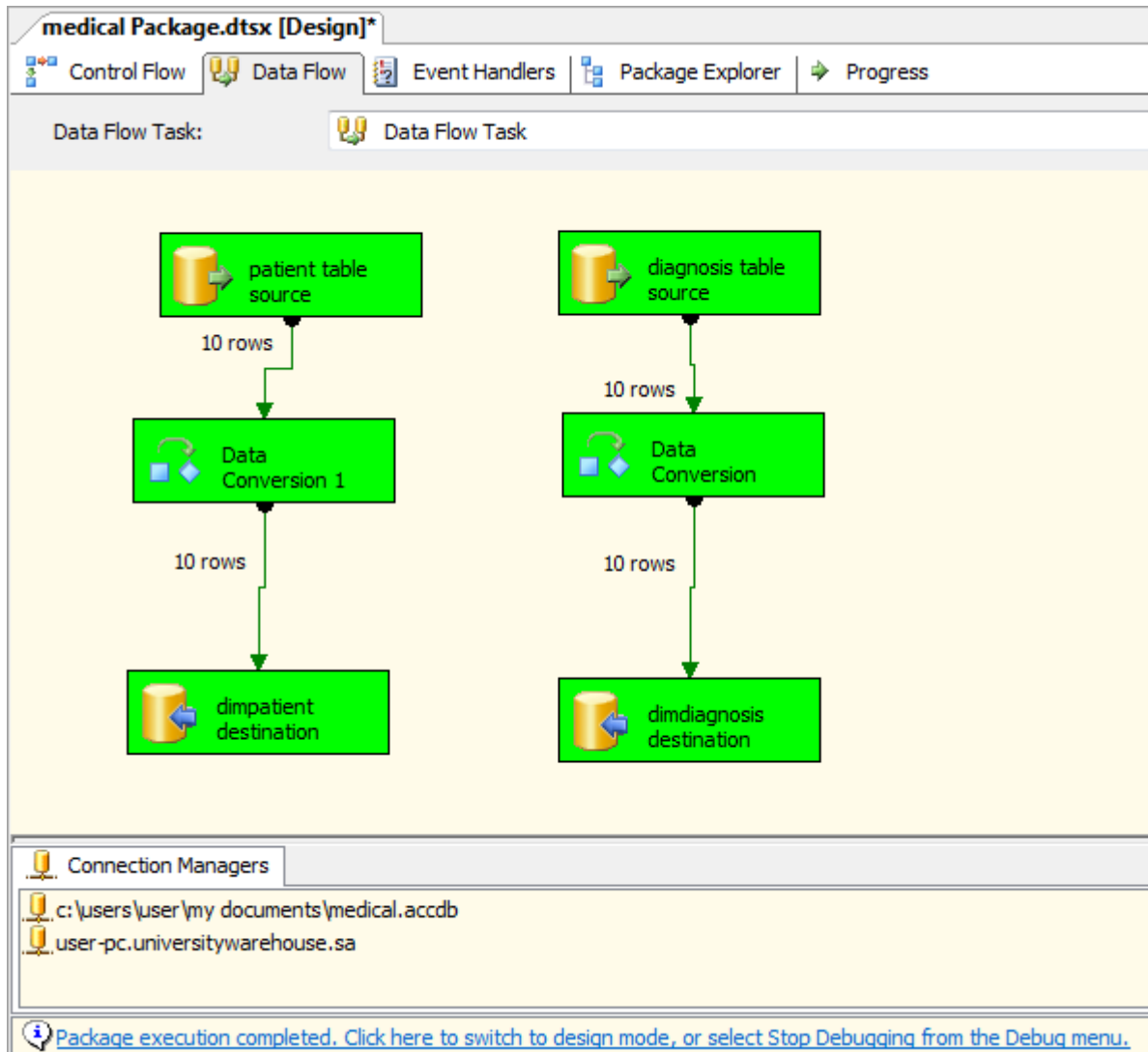


Figure 13: ETL for medical schema

Next is to configure each component so that SSIS knows how to extract and load the data to the fact tables. Drag a data flow task from the Toolbox onto the Control Flow designer's surface. Double-click the Data Flow task in the designer. BIDS now displays the Data Flow designer. The extract portion needs to query data warehouse database tables. To set up this task, drag an OLE DB Source component from the Toolbox onto the Data Flow designer surface and then rename the OLE DB Source component extract fact.

Next is to define the load portion package to load into the fact tables. I drag a new instance of the OLE DB Destination component onto the Data Flow designer surface and rename it load Factstudent. To load Facstudent link the available columns from the first extract, then join to the appropriate dimension table using the join of like fields.

Right-click the extract fact source component and select edit. Select data warehouse connection manager. Then expand the Name of the table drop-down and select the first table am extracting from. Finally, click the Mappings tab to confirm columns. Click OK to confirm the mappings. Next right click lookup component, and select Edit. With the lookup Editor open select data warehouse connection manager. Then expand the Name of the table drop-down and select the second table to extract from. Finally, click the columns tab to specify join columns and use of reference columns. Confirm that the mappings are correct. Click OK to confirm the mappings.

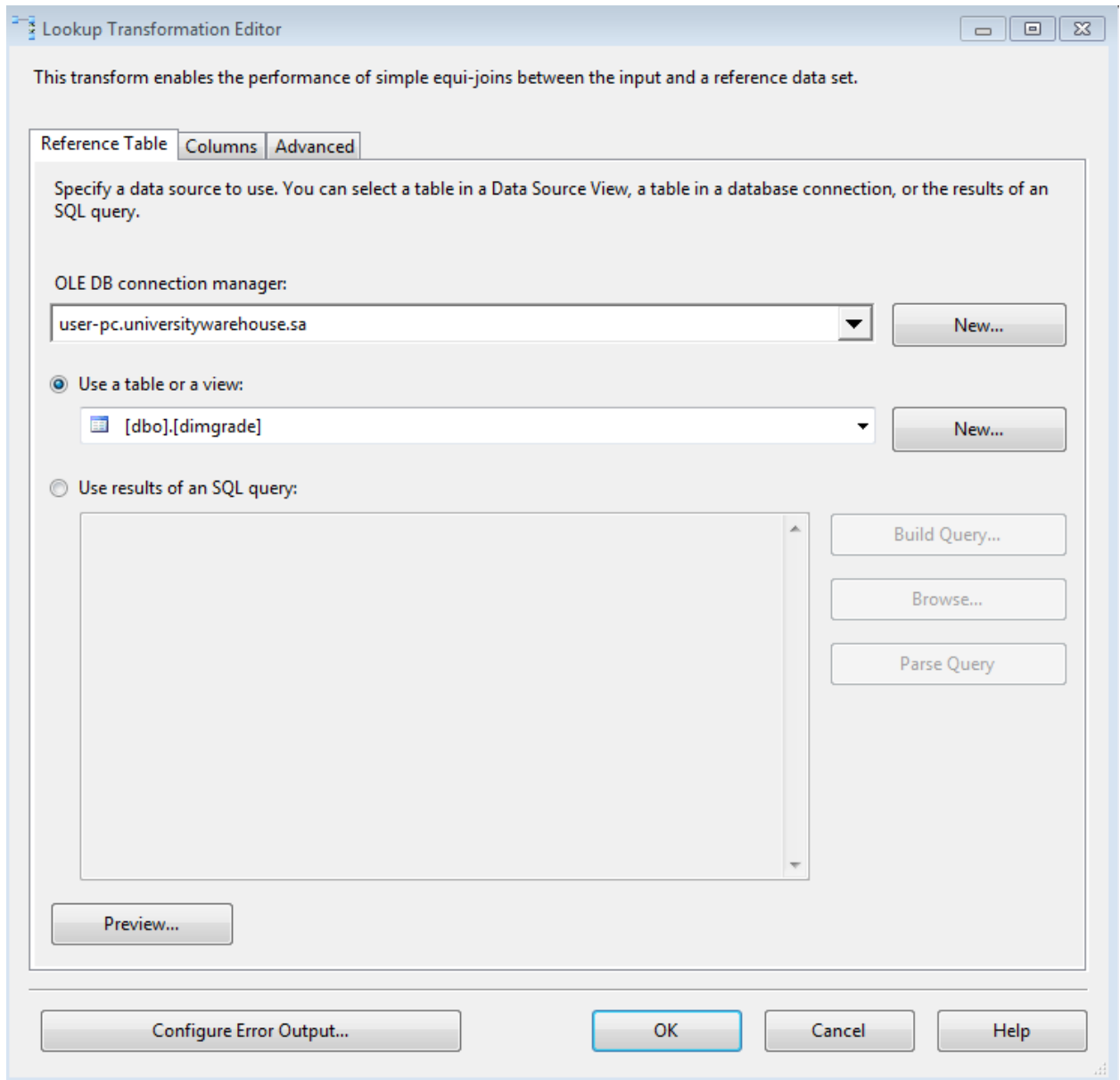


Figure 14: Lookup transformation editor

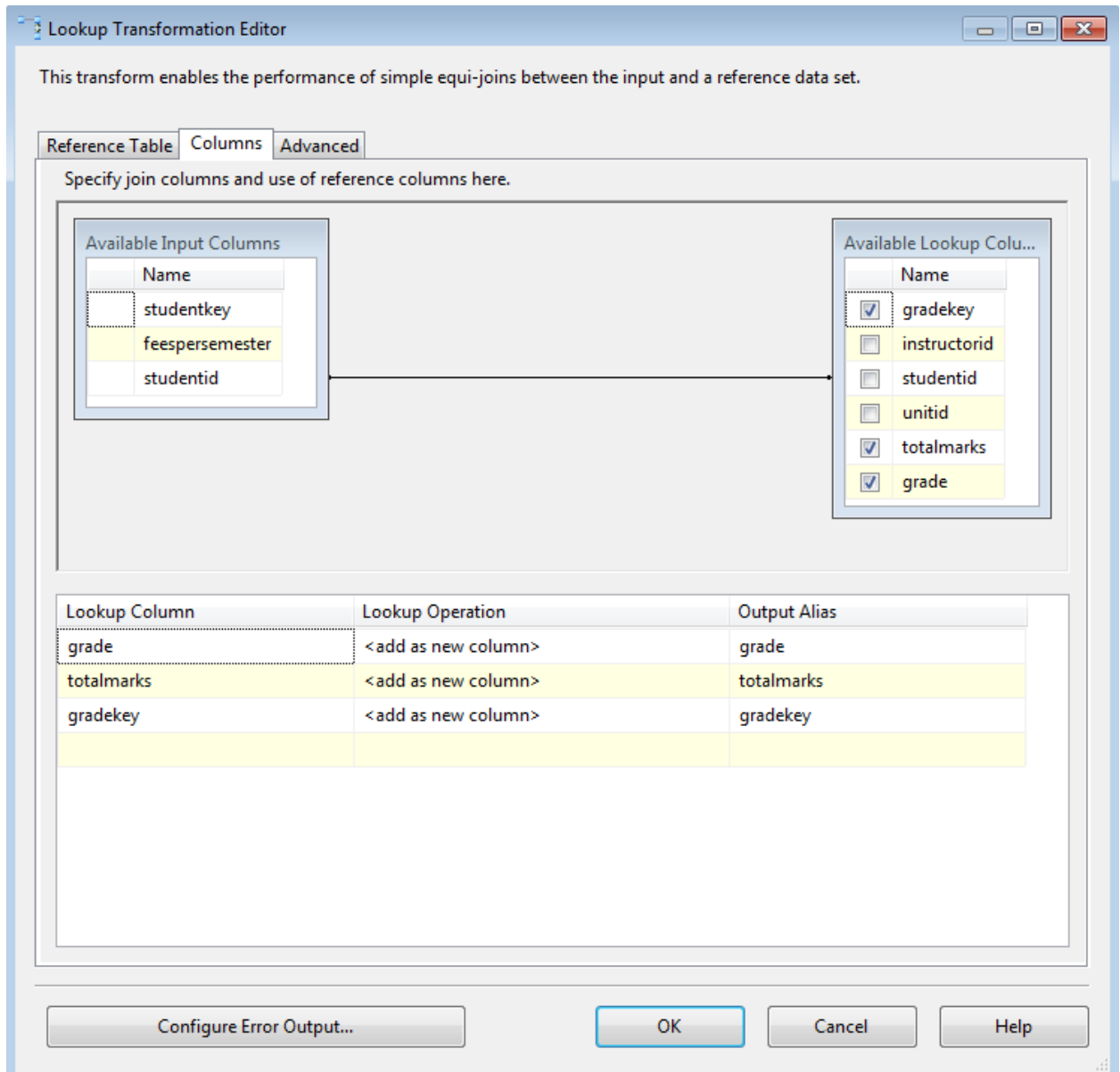


Figure 15: Specification of join columns and use of reference columns

Next right click data Destination component, and select Edit. With the OLE DB Destination Editor open, make sure the data warehouse is selected as the OLE DB connection manager. Then expand the Name of the table drop-down and select the Factstudent table. Finally, click the Mappings tab to confirm that the mappings are correct. Click OK to confirm the mappings. The factstudent ETL package is now complete. When this is done repeat the

same for other fact tables. To test the package, open the package and press F5. Below figures shows a successful ETL.

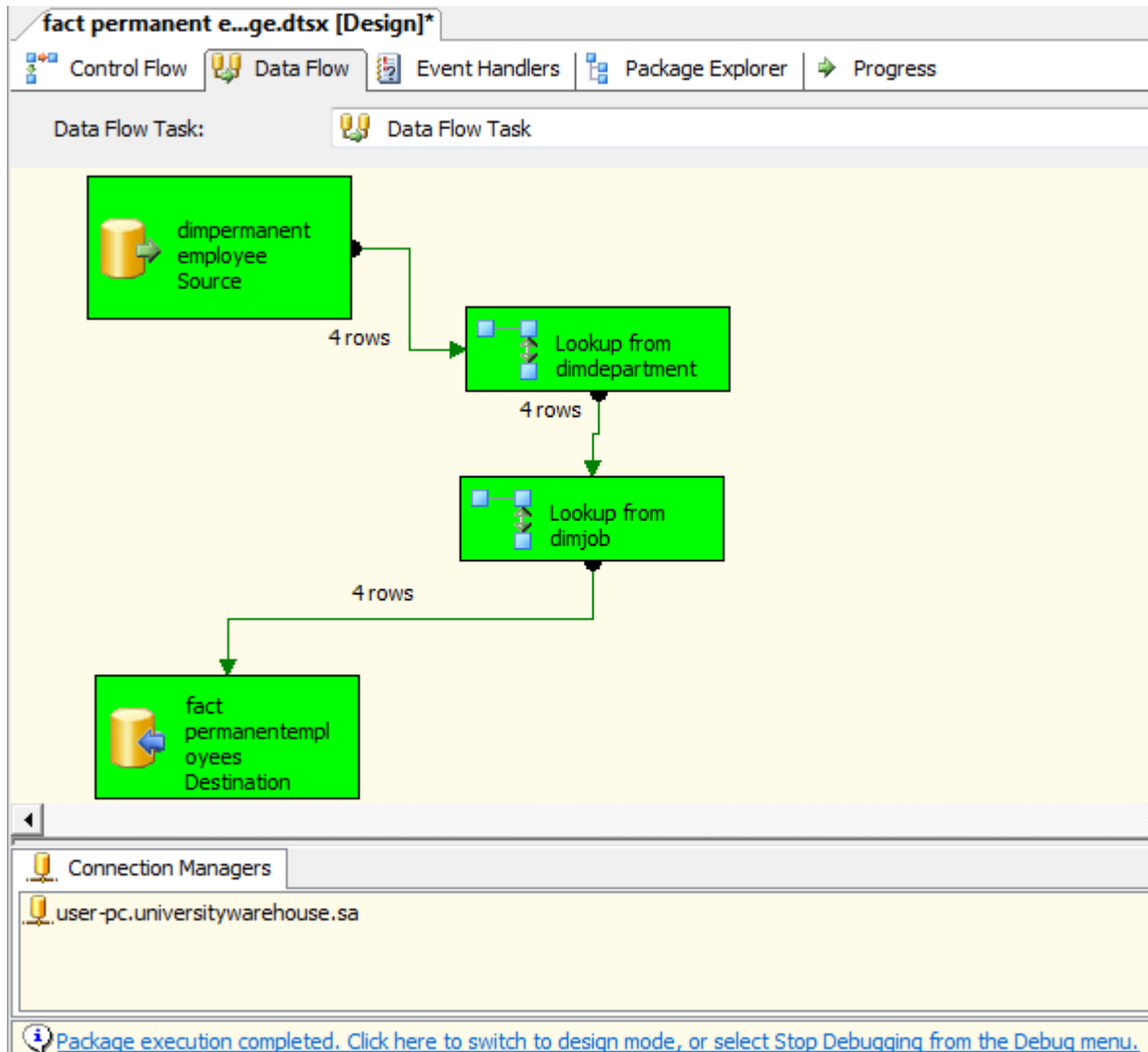


Figure 16: Loading HR fact table

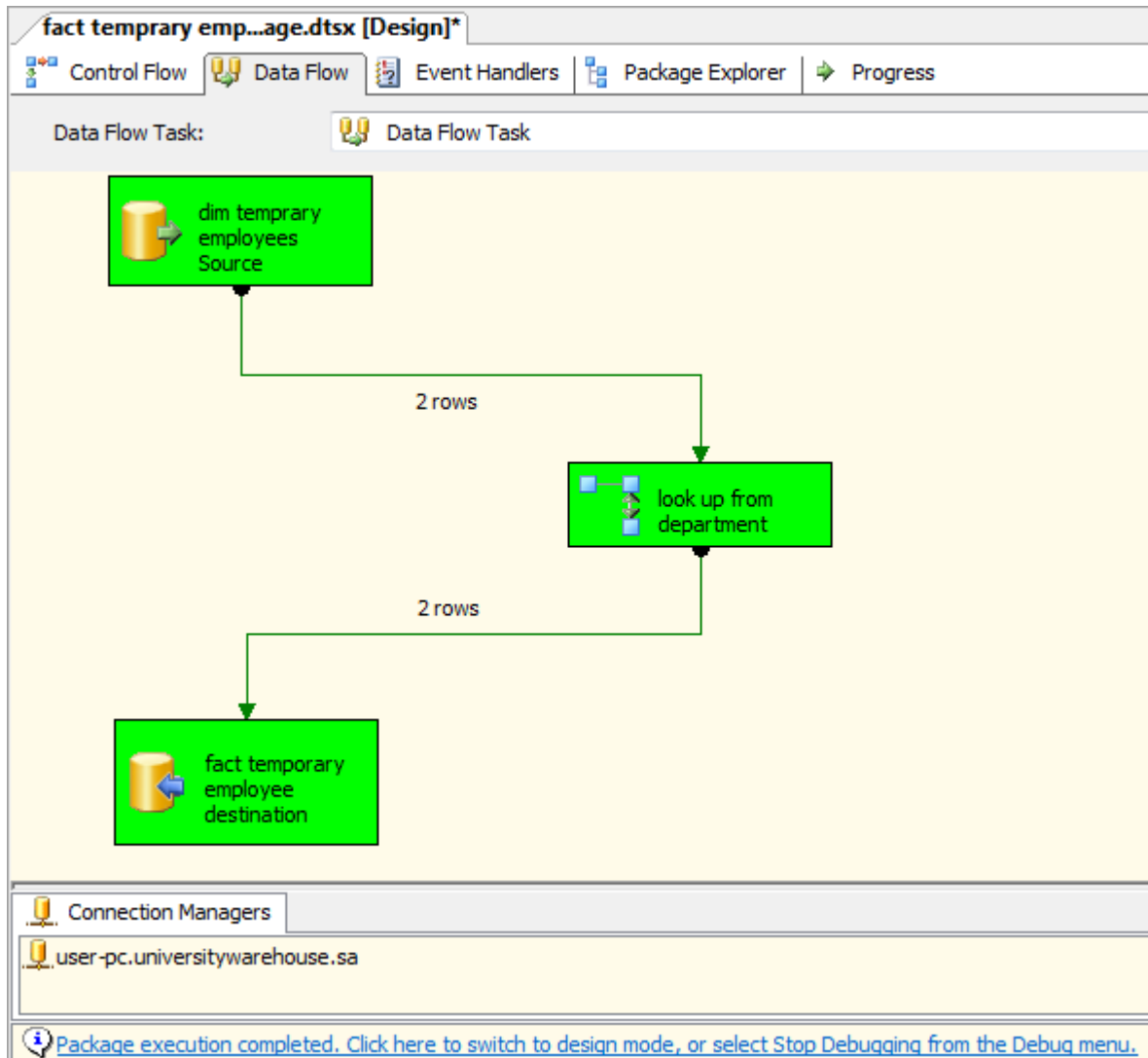


Figure 17: Loading fact temporary employees

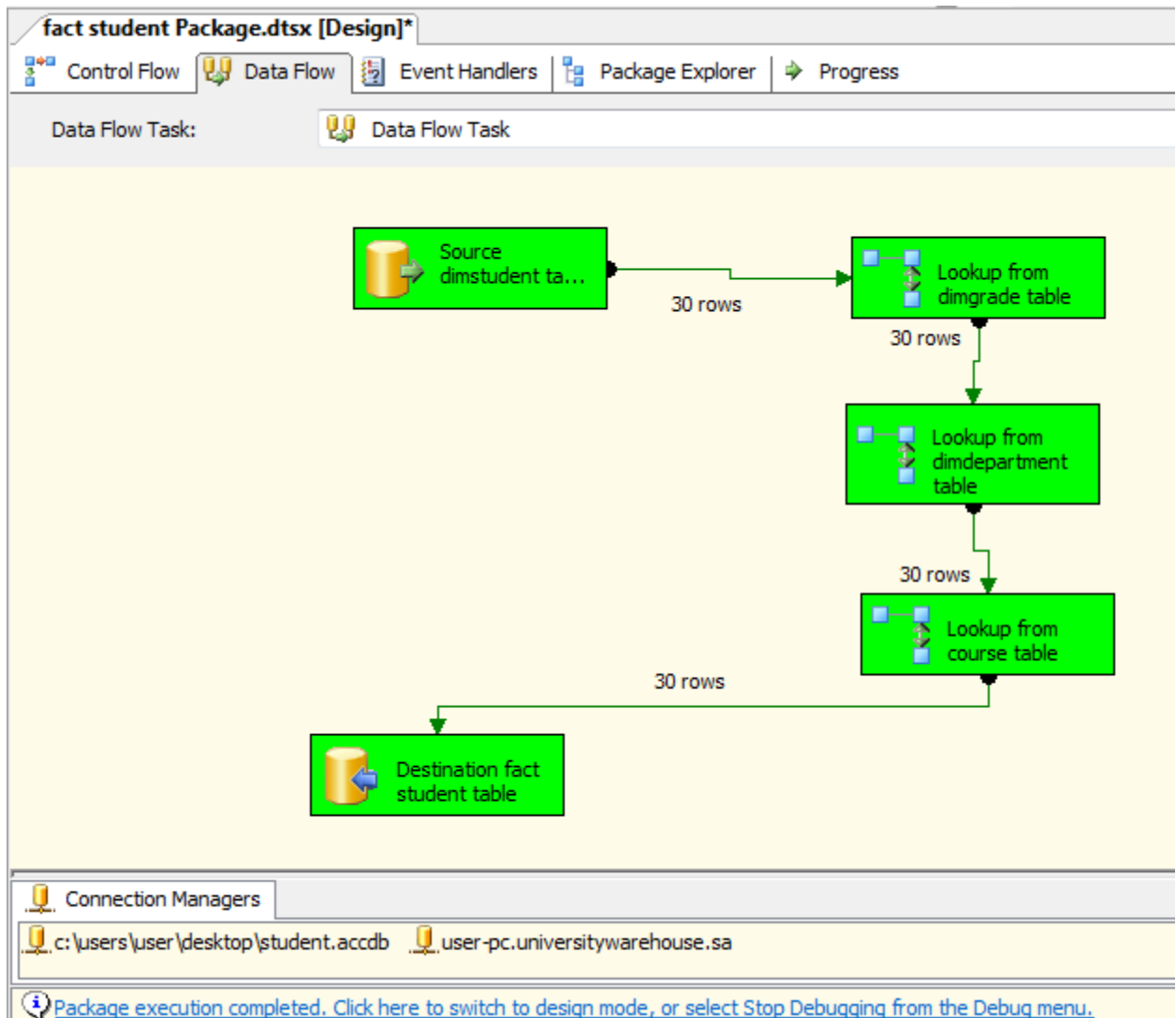


Figure 18: Loading student fact table

5.4 User Interface

The user interface has been developed with Microsoft visual web developer 2010 express. This is an internet based environment which users can access when connected to the internet. It is also compatible with most web browsers. The users have various hyperlinks to access the data warehouse and more can be added if required. For this prototype the reports can only be viewed in HTML though other methods like pdf's can be added. The

figure below shows a screen short of the University performance data warehouse prototype user interface.

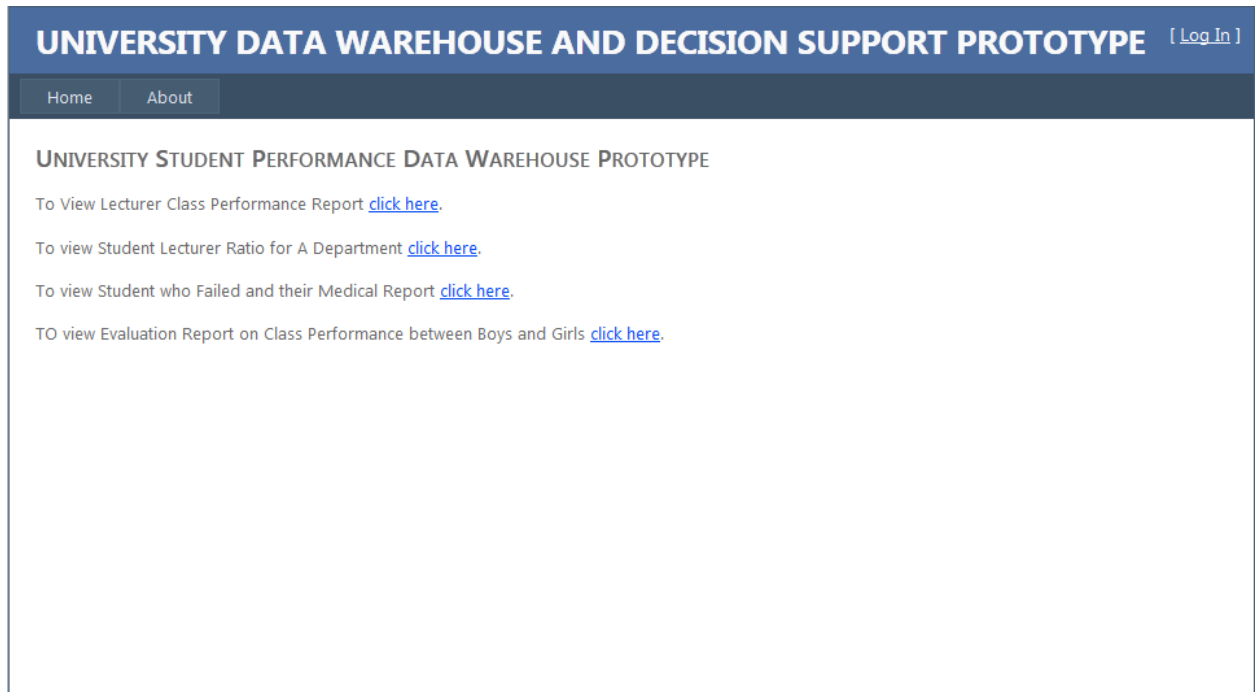


Figure 19: user interface

5.5 Reports

The example reports developed with this prototype shows how a report writer within the university can use the data warehouse. The first example – When a lecturer is hired a user in the human resource department will enter a new instructor into the *HR* database then assign an employee id. In the department the lecturer will be assigned a class and another user in the *time tabling system* database enters the instructor with the employee-id which the lecturer was assigned by the human resource officer and the units they are teaching and the number of students. At the end of the semester another user enters the student marks in

the exam database. The data is then loaded to the warehouse dimension and fact tables where a link is formed between the HR and *time tabling* and *exam* databases through the employee-id and student-id. This example shows employment details of lecturers from the *HR* database, and the units they taught, the number of student in that class including the average marks achieved. The code to build the report is:

```
select distinct dimgrade.unitid, count(unitid) as numofstudents,  
avg(dimgrade.totalmarks)as average, dimpermanentemployees.employeeid,  
dimpermanentemployees.firstname, dimpermanentemployees.lastname from dimgrade,  
dimpermanentemployees where unitid=unitid and  
dimpermanentemployees.employeeid=dimgrade.instructorid group by unitid,  
employeeid, firstname, lastname, salary
```

UNIVERSITY DATA WAREHOUSE AND DECISION SUPPORT PROTOTYPE [Log In]					
Home		About			
LECTURERS CLASS PERFORMANCE REPORT					
unitid	numofstudents	average	employeeid	firstname	lastname
CPA 402	5	71	CPA/6/2014	KENNETH	NGANGA
CPA 403	4	48	CPA/5/2014	COLLINS	MAYABI
ICT 202	5	54	ICT/10/2014	DANIEL	MACHOGU
ICT 203	6	68	ICT/11/2014	BENSON	THUKU

Figure 20: Report 1

The report shows data from three systems operating separately which is a real example of the importance of the University data warehouse as this could have taken a report writer too much time and energy to process considering that the university could be having hundreds of instructors. Using this report the school administrators have the time to make the right decision and improve the quality of education in the university.

The second example - When a lecturer is hired a user in the human resource department will enter a new instructor into the *HR* database then assign an employee id, designation and department. When a student is admitted a completely different user in the Admissions

database enters the new student with all the details including the department. The data is then loaded to the warehouse dimension and fact tables where a link is formed between the HR and admissions databases through the department. This report shows the total number of students from each department in comparison with lecturers from the same department.

The code to build the report is:

```
select distinct dimstudent.department, count(dimstudent.department) as numofstudents,
      (select count( dimdepartment2.departmentname2) from dimdepartment2 where
      departmentname2='ict') as numoflecturers from dimstudent where department='ict' group
      by department
```



Figure 21: Report 2

The third example - When a student is admitted a user will enter new student into the STUDENT database then assign a studentid. A student database keeps track keeps track of all student details including grades for each and every unit that has been pursued by the student. If a student gets ill and visits the health unit department another different users

(the nurse) records the patient referencing the studentid given by the admission officer in the medical system. The data is then loaded to the warehouse dimension and fact tables where a link is formed between STUDENT database and MEDICAL database through the studentid and patientid.

```

select dimstudent.studentid, dimstudent.firstname, dimstudent.lastname,
dimdepartment.departmentname, dimgrade.grade, dimdiagnosis.diagnosis from dimstudent
join dimdepartment on dimstudent.studentid=dimdepartment.studentid join dimgrade on
dimdepartment.studentid=dimgrade.studentid join dimdiagnosis on
dimgrade.studentid=dimdiagnosis.patientid

```

UNIVERSITY DATA WAREHOUSE AND DECISION SUPPORT PROTOTYPE [Log In]						
Home	About					
A REPORT SHOWING ALL STUDENT WHO CHECKED INTO THE HEALTH CENTER AND HOW THEY PERFORMED.						
studentid	firstname	lastname	gender	departmentname	grade	diagnosis
ICT/04640/2014	MWANGI	NGUGI	male	ICT	D	COMPUTER VISION SYNDROME
ICT/04641/2014	JAMES	MWANGI	male	ICT	E	COMPUTER VISION SYNDROME
ICT/04642/2014	GITHI	MUTHEE	male	ICT	E	COMPUTER VISION SYNDROME
ICT/04643/2014	DENNIS	GICHIMU	male	ICT	D	COMPUTER VISION SYNDROME
CPA/02641/2014	PETER	MACHARIA	male	ARCHITECTURE	E	MALARIA
CPA/02642/2014	DANIEL	MWANGI	male	ARCHITECTURE	E	MALARIA
ICT/04644/2014	WAIRIMU	KAGUNDA	female	ICT	E	MALARIA
ICT/04645/2014	JAMLICK	MUTHOMI	male	ICT	D	MALARIA
ICT/04646/2014	TARI	FATUMA	female	ICT	E	COMPUTER VISION SYNDROME
ICT/04647/2014	AKINYI	ONYANGO	female	ICT	D	COMPUTER VISION SYNDROME

Figure 22: Report 3

The report shows that most ICT department student had eyes problem. This could have been caused by computers they use on day to day basis. By this the school administrators can be able to make a quick decision on the cause of failures by these student and take the

necessary actions. This could be by hiring an optician to advice on how to prevent this disease and stop poor performances from this department.

The fourth example - The third example - When a student is admitted a user in the Admissions database enters the new student with all the details including the student-id and gender. On the examinations database another user records the student's marks using the student-id allocated to student on admission. The data is then loaded to the warehouse dimension and fact tables where a link is formed between the admission and exam databases through the student-id. The report compares performance between girls and boys in various departments. The code to build the report is:

```
select dimstudent.studentid, dimstudent.gender, dimgrade.department, dimgrade.course,
       dimgrade.grade, dimgrade.avgmarks from dimstudent, dimgrade where
       dimstudent.studentid=dimgrade.studentid and gender='female' group by
       dimstudent.studentid, dimgrade.department, dimgrade.course, dimstudent.gender,
       dimgrade.grade, dimgrade.avgmarks
```

UNIVERSITY DATA WAREHOUSE AND DECISION SUPPORT PROTOTYPE [Log In]					
Home	About				
A REPORT COMPARING PERFORMANCE BETWEEN GIRLS AND BOYS IN VARIOUS DEPARTMENTS.					
studentid	gender	department	course	grade	avgmarks
CPA/02643/2014	female	ARCHITECTURE	CPA	B	61
CPA/02654/2014	female	ARCHITECTURE	CPA	A	73
ICT/04644/2014	female	ICT	computer science	E	30
ICT/04647/2014	female	ICT	computer science	D	43
ICT/04649/201	female	ICT	computer science	A	77
ICT/04650/201	female	ICT	computer science	A	72
ICT/O4646/2014	female	ICT	computer science	E	34

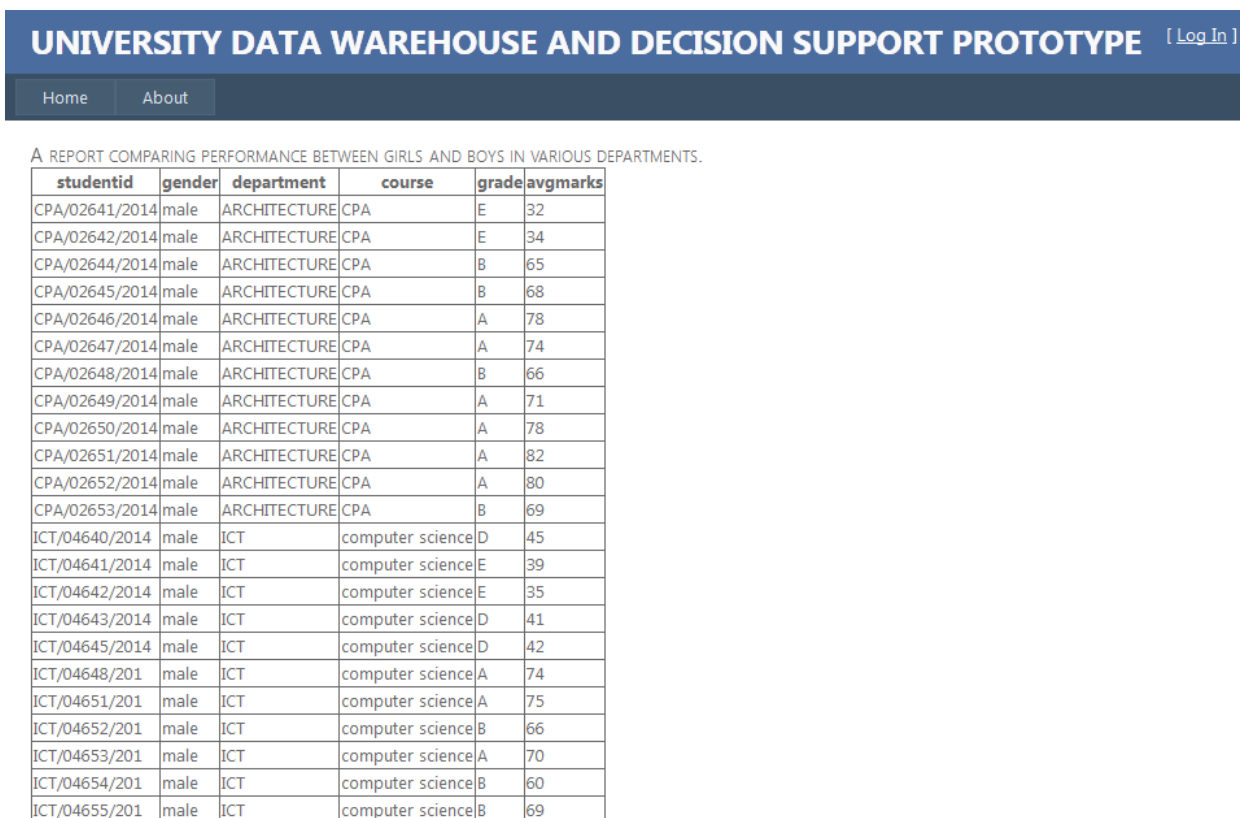


Figure 23: Report 4

5.6 Analysis on the Benefits of the Prototype

This section is for analyzing the user requirements gathered in the methodology step. Interviews are done with two people in the University in administrative and academic positions. The interviews are being done to gather their views mostly on this prototype and to know additional requirements the users might have for future study. Appendix C contains the interview questions.

On the question of how they thought the system would improve efficiency. They found the prototype very efficient for analyzing students' information by analyzing and building reports from the available data, without traversing the various operational systems which would mostly result to reports full of errors. They also found that the system would greatly

improve resource planning and faster decision making in the university as information was readily available in the data warehouse for University management to decide the way forward.

6.0 Conclusion

This research covers implementation of a University student performance Data Warehouse to bring together all the information created by various department and improve on analysis, reporting and presentation of information for better decision making. Managing the design, development, implementation, and operation of a data warehouse can be a difficult and time consuming task. Performing requirement analysis is critical to success of any project in order to identify specific problems that the system will solve. The problems should be clearly stated, specific and testable for the success of the warehouse. The team leader should then should then publicize these problems and get users and management feedback. Secondly the team leader needs to know the resources allocated to the system to help define the development path. In case the budget is low he may consider downscaling the effort or if time is short, consider using off-the-shelf products extensively. If personnel resources are inadequate he may consider outsourcing. This is because being realistic of the actual resources helps prevent overruns and project disappointment.

Data warehouse implementation requires management commitment and a strong believe in the benefits of the project. The management team should also be responsible for keeping the project team motivated. Data warehouse implementation should be keen on documented current techniques used worldwide for it to be a success. By implementing a data warehouse the executives are able to look at the university as a whole instead of the many departments that the university comprises. The other key benefit is that the a data warehouse is the ability to handle server tasks connected to queries which are not applicable to most transaction databases used by various departments in a university.

The model of a data warehouse is in such a way that it speeds up reporting. This is often done via star schemas which are not recommended for transaction processing system. This is because certain models could slow down transaction processing system. Most data warehouses can be set in a way that simple queries can be written by users who do not have a lot of technical skill. The prototype developed in this thesis shows the advantages of implementing a University Data Warehouse to offer strategic information for better management of resources and strategic decision making.

6.1 Limitations of the study

Limitations encountered in the course of this study include: Use of sample data because the University only described the type of data stored in their databases but could not provide real data due to security reasons. If the university data had been provided, the output of the reports would have been more realistic.

The sources of data used in this study are only data that relates to students administration. Other sources of data can be included to cover all University processes. Lastly free tools were used for this study because there was no financial support and the research period was limited.

6.2 Future Study

This study used sample data thus future research should focus on using real data. Also future researchers should add more data sources to cover the entire University environment including the library system, distance learning systems, alumni system and any other

system used in the university. Finally the future research can focus on design of a tool to generate more friendly queries.

Bibliography

(n.d.).

Adamson, C. (2010). *Star Schema: The Complete Reference*. New York:: McGraw Hill,.

Agnes, P. N. (2012). A FRAMEWORK FOR MANAGING ELECTRONIC RECORDS: THE CASE OF MOI UNIVERSITY, KENYA. *SCECSAL XXth KLA*. Nairobi: KLA.

Ariyachandra, T., & Watson, H. (2005). *Data warehouse architecture: factors in the selection, decision and success of the architecture*. [http: www.terry.uga.edu](http://www.terry.uga.edu).

Ariyachandra, T., & Watson, H. (2010). scopus database.

Arun Sen and Atish P. Sinha. (March 2005). A Comparison of Data Warehousing. *Communications of the ACM*, vol. 48. no.3, .

Azwa Abdul Aziz, W. M. (2012). Intelligent System for Personalizing Students' AcademicBehaviors- A Conceptual Framework. *International Journal on New Computer Architectures and Their Applications*, 138-153.

Begg, T. C. (2002). *Database Systems*. 3th Edition, Addison-Wesley, .

Bonifati, A., Catteneo, F., Ceri, S., Fuggetta, A., & Paraboschi, A. (2001). Designing Data marts for Data Warehouses. *ACM Transaction Software Engineering methodology* 10(4), 452-483.

Borysowich, C. (2007). *Better data Warehouse Modeling*.

Cognos. (2008). *The Right Architecture for Business Intelligence*. Burlington, USA: Cognos Inc.

Dayal, & Chaudhuri. (1997). An overview of data warehousing and OLAP technology. *SIGMOND* 26(1), 65-74.

del Hoya, B., & Lees, D. (2002). The Use of Data Warehouses in Health Care Sector. *Health informatics Journal* 8(1), 43-46.

(2007). *Developing a data warehouse for a university decision*. Atılım University.

- Dr. Kamal Kakish, D. T. (2012). *ETL Evolution for Real-Time Data Warehousing*. New Orleans Louisiana, USA: Education Special Interest Group of the AITP.
- Dr. Kamal Kakish, D. T. (2012). *ETL Evolution for Real-Time Data Warehousing*. New Orleans Louisiana, USA: Education Special Interest Group of the AITP.
- Economics, O. (2011). *Real-Time Business:Playing to Win in the New Global Marketplace, A white paper produced in collaboration with SAP*.
- Golfarelli, m, R. S. (2009). *Data warehouse design: Modern principles and methodologies*. New York: McGraw-Hill Companies.
- H, I. (2005). *Building the data warehouse*. Indianapolis: Wiley Publishing Inc.
- Hagerty,J, H. S. (2011). *Magic Quadrant for Business Intelligence Platforms*. Stamford, CT: Gartner Group.
- Henschen.D. (2012). *BI and Information Management Trends*,. Information Week.
- Kadlec, J. (2005). *SQL Server OLTP vs. Data warehouse performance tuning*.
- Kimball Group. (2010). *The Evolving Role of the Enterprise Data Warehouse in the Era of Big Data Analytics 1*. boulevard: Kimball Group.
- Kimball, R., & Rosi, M. (2002). *The data warehouse toolkit 2nd edition ed*. Toronto: John Woley and Sons Inc.
- Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit 3rd Edition: the Definitve Guide to Dimensional modelling*. Boulevard: John Wiley & Sons, Inc.
- Leonard, E. M. (2011). *Design and Implementation of an Enterprise Data Warehouse*. Marquette University.
- M, K. E. (2001). An Aproach for Integrating heterogeneous Information Sources in a Medical Data Warehouse. *Journal of medical systems* 23(3), 167-176.
- March, S., & Henvner, A. (2007). Integrated decision support systems: A data warehousing perspective. *Decision Support Systems* 43(3), 1031-1043.
- Marco, D. (2002). *Independent Data Marts-part 1*. www. tdan.com: The data administration newsletter.
- Matija Lah, G. J. (2012). *Implementing a Data Warehouse with Microsoft® SQL Server 2012*. California: Microsoft corporation.
- Mih, E. (2010). *Dimentional model schemas-star, snow-flake constellation*.

- Mustafa, M. S. (2012). Usability of Data Warehousing and Data Mining for Interactive Decision Making in Textile Sector. *Journal of Computer Science and Technology* , Volume 12 Issue 7 Version 1.0.
- Nguyen, P. V. (2011). *USING DATA WAREHOUSE TO SUPPORT BUILDING STRATEGY OR FORECAST BUSINESS TEND*. Ho Chi Minh City, Vietnam: Faculty of Information Technology, Hung Vuong Univesity,.
- Ofori Boateng, J. S. (2011). DATA WAREHOUSING. *Business Intelligence Journal*, 224-234.
- oracle. (2013). *Oracle database 12c for data warehousing and big data*. Oracle.
- Pias M.M, T. T. (2008). *Data Warehouse Architecture Through Viewpoint of Information Architecture*. IEEC Computer Society Database.
- PONNIAH, P. (2001). *Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals*. New York: A Wiley-Interscience.
- Ralph Kimball. Laura Reeves, M. R. (2001). *The data warehouse lifecycle toolkit: Expert methods for designing developing and deploying data warehouses*. Kimball group.
- Rapid decision. (2013). *Why you need a data warehouse* . 700 West Johnson Avenue: Rapid decision.
- Roddick, K. M. (2007). *Advances and research Directions in Data Warehousing Technology*.
- Tan, R. (2006). *Online analytical processing systems*. www. irma-international.org.
- Veda C. Storey, C. H. (December 2012). BUSINESS INTELLIGENCE AND ANALYTICS:FROM BIG DATA TO BIG IMPACT. *MIS Quarterly Vol. 36 No. 4,,* 1165-1188.
- Wai, T., & Aung, S. (2009). Metadata Based Student Data Extraction from Universities Data Warehouse,. *International Conference on Signal Processing Systems*, 670-673.

Appendix A: Interview Guide

1. What is the type of student performance analysis is performed?
2. Where is the data for analysis stored and how do you get it?
3. Does the system in place provide solutions to your needs?
4. What problems do you encounter with the current system?
5. What is the procedure of doing this analysis?
6. Do you require any historical information?
7. What capabilities would you like included in the system?

Appendix B: Physical dimension and Fact Table Descriptions

//creating dimension tables

create table dimstudent

(studentkey int not null identity(1,1)primary key,

studentid varchar(20) not null,

courseid varchar(20) not null,

lastname varchar(20) not null,

enrollmentdate datetime,

numofsemesters int not null,

feespersemester int not null,

totalfees int not null,

)

create table dimcourse

(coursekey int not null identity(1,1)primary key,

instructorid varchar(20) not null,

studentid varchar(20) not null,

courseid varchar(20) not null,

unitid varchar(20) not null,

departmentname varchar(20) not null,

)

create table dimgrade

(gradekey int not null identity(1,1)primary key,

instructorid varchar(20) not null,

studentid `varchar(20)` not null,

unitid `varchar(20)` not null,

totalmarks `int` not null,

grade `varchar(20)` not null,

)

`create table` dimdepartment

(departmentkey `int` not null `identity(1,1)`primary key,

studentid `varchar(20)` not null,

departmentname `varchar(20)` not null,

managerid `varchar(20)` not null,

)

`create table` dimpermanentemployees

(permanentemployeeskey `int` not null `identity(1,1)`primary key,

employeeid `varchar(20)` not null,

lastname `varchar(20)` not null,

firstname `varchar(20)` not null,

email `varchar(20)` not null,

phonenummer `int` not null,

hiredate `datetime` not null,

salary `int` not null,

)

`create table` dimtemporaryemployees

(permanentemployeeskey `int` not null `identity(1,1)`primary key,

```
employeeid varchar(20) not null,  
lastname varchar(20) not null,  
firstname varchar(20) not null,  
email varchar(20) not null,  
phonenumbers int not null,  
numofunits int not null,  
costperunit int not null,  
totalcost int not null,  
)
```

```
create table dimjobs
```

```
(jobskey int not null identity(1,1)primary key,  
employeeid varchar(20) not null,  
jobtitle varchar(20) not null,  
minsalary int not null,  
maxsalary int not null,  
leaveallowance int not null,  
)
```

```
create table dimdepartment2
```

```
(department2key int not null identity(1,1)primary key,  
departmentname varchar(20) not null,  
managerid varchar(20) not null,  
locationid varchar(20) not null,  
)
```

```
create table dimdepatient
(patientkey int not null identity(1,1)primary key,
patientid varchar(20) not null,
lastname varchar(20) not null,
firstname varchar(20) not null,
dateofbirth datetime not null,
phonenummer int not null,
)
```

Go

```
// creating fact tables
```

```
Create Table factstudent
```

```
(
studentKey int,
courseKey int,
gradeKey int,
departmentKey int,
feespersemester int not null,
totalmarks int not null,
grade varchar not null,
)
```

-- Add relation between fact table foreign keys to Primary keys of Dimensions

ALTER TABLE Factstudent ADD CONSTRAINT

FK_studentkey FOREIGN KEY (studentkey)REFERENCES Dimstudent(studentkey);

ALTER TABLE Factstudent ADD CONSTRAINT

FK_coursekey FOREIGN KEY (coursekey)REFERENCES Dimcourse(coursekey);

ALTER TABLE Factstudent ADD CONSTRAINT

FK_gradekey FOREIGN KEY (gradekey)REFERENCES Dimgrade(gradekey);

ALTER TABLE Factstudent ADD CONSTRAINT

FK_departmentkey FOREIGN KEY (departmentkey)REFERENCES

Dimdepartment(departmentkey);

Go

Create Table factpermanentemployees

(

permemployeeKey int,

jobsKey int,

departmentKey int,

salary int not null,

leaveallowance int not null,

grade varchar not null,

)

-- Add relation between fact table foreign keys to Primary keys of Dimensions

ALTER TABLE Factpermanentemployees ADD CONSTRAINT

FK_permemployeekey FOREIGN KEY (permemployeekey)REFERENCES

Dimpermanentemployeee(permanentemployeeskey);

ALTER TABLE Factpermanentemployees ADD CONSTRAINT

FK_jobskey FOREIGN KEY (jobskey)REFERENCES dimjob(jobskey);

ALTER TABLE Factpermanentemployees ADD CONSTRAINT

FK_departmentkey FOREIGN KEY (departmentkey)REFERENCES

Dimdepartment2(departmentkey);

Go

Create Table facttemporaryemployees

(

tempemployeeKey int,

departmentKey int,

numofunits int not null,

costperunit int not null,

totalcost varchar not null,

)

-- Add relation between fact table foreign keys to Primary keys of Dimensions

ALTER TABLE Facttemporaryemployees ADD CONSTRAINT

FK_tempemployeekey FOREIGN KEY (tempemployeekey)REFERENCES

Dimtemporaryemployeee(temporaryemployeeskey);

ALTER TABLE Facttemporaryemployees ADD CONSTRAINT

```
FK_departmentkey FOREIGN KEY (departmentkey)REFERENCES
dimdepartment(departmentkey);

go

//Create Time Dimension Table
//Create Table [dbo].[DimTime]
SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[DimTime](
[TimeKey] [int] NOT NULL,
[TimeAltKey] [int] NOT NULL,
[Time30] [varchar](8) NOT NULL,
[Hour30] [tinyint] NOT NULL,
[MinuteNumber] [tinyint] NOT NULL,
[SecondNumber] [tinyint] NOT NULL,
[TimeInSeconds] [int] NOT NULL,
[HourlyBucket] varchar(15)not null,
[DayTimeBucketGroupKey] int not null,
[DayTimeBucket] varchar(100) not null
CONSTRAINT [PK_DimTime] PRIMARY KEY CLUSTERED
```

```
(
[TimeKey] ASC
)
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, _
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
)
ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/****Time Dimension with Values****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[FillDimTime]
as
BEGIN
--Specify Total Number of Hours You need to fill in Time Dimension
DECLARE @Size INTEGER
--iF @Size=32 THEN This will Fill values Upto 32:59 hr in Time Dimension
Set @Size=23
```

```
DECLARE @hour INTEGER
DECLARE @minute INTEGER
DECLARE @second INTEGER
DECLARE @k INTEGER
DECLARE @TimeAltKey INTEGER
DECLARE @TimeInSeconds INTEGER
DECLARE @Time30 varchar(25)
DECLARE @Hour30 varchar(4)
DECLARE @Minute30 varchar(4)
DECLARE @Second30 varchar(4)
DECLARE @HourBucket varchar(15)
DECLARE @HourBucketGroupKey int
DECLARE @DayTimeBucket varchar(100)
DECLARE @DayTimeBucketGroupKey int
SET @hour = 0
SET @minute = 0
SET @second = 0
SET @k = 0
SET @TimeAltKey = 0
WHILE(@hour<= @Size )
BEGIN
if (@hour <10 )
begin
```

```
set @Hour30 = '0' + cast( @hour as varchar(10))
end
else
begin
set @Hour30 = @hour
end
--Create Hour Bucket Value
set @HourBucket= @Hour30+':00' + '-' +@Hour30+':59'
WHILE(@minute <= 59)
BEGIN
WHILE(@second <= 59)
BEGIN
set @TimeAltKey = @hour *10000 +@minute*100 +@second
set @TimeInSeconds =@hour * 3600 + @minute *60 +@second
If @minute <10
begin
set @Minute30 = '0' + cast ( @minute as varchar(10) )
end
else
begin
set @Minute30 = @minute
end
if @second <10
```

```

begin
set @Second30 = '0' + cast ( @second as varchar(10) )
end
else
begin
set @Second30 = @second
end
--Concatenate values for Time30
set @Time30 = @Hour30 + ':' + @Minute30 + ':' + @Second30
--DayTimeBucketGroupKey can be used in Sorting of DayTime Bucket In proper Order
SELECT @DayTimeBucketGroupKey =
CASE
WHEN (@TimeAltKey >= 00000 AND @TimeAltKey <= 25959) THEN 0
WHEN (@TimeAltKey >= 30000 AND @TimeAltKey <= 65959) THEN 1
WHEN (@TimeAltKey >= 70000 AND @TimeAltKey <= 85959) THEN 2
WHEN (@TimeAltKey >= 90000 AND @TimeAltKey <= 115959) THEN 3
WHEN (@TimeAltKey >= 120000 AND @TimeAltKey <= 135959) THEN 4
WHEN (@TimeAltKey >= 140000 AND @TimeAltKey <= 155959) THEN 5
WHEN (@TimeAltKey >= 50000 AND @TimeAltKey <= 175959) THEN 6
WHEN (@TimeAltKey >= 180000 AND @TimeAltKey <= 235959) THEN 7
WHEN (@TimeAltKey >= 240000) THEN 8
END
--print @DayTimeBucketGroupKey

```

```

-- DayTimeBucket Time Divided in Specific Time Zone
-- So Data can Be Grouped as per Bucket for Analyzing as per time of day
SELECT @DayTimeBucket =
CASE
WHEN (@TimeAltKey >= 00000 AND @TimeAltKey <= 25959) _
THEN 'Late Night (00:00 AM To 02:59 AM)'
WHEN (@TimeAltKey >= 30000 AND @TimeAltKey <= 65959) _
THEN 'Early Morning(03:00 AM To 6:59 AM)'
WHEN (@TimeAltKey >= 70000 AND @TimeAltKey <= 85959) _
THEN 'AM Peak (7:00 AM To 8:59 AM)'
WHEN (@TimeAltKey >= 90000 AND @TimeAltKey <= 115959) _
THEN 'Mid Morning (9:00 AM To 11:59 AM)'
WHEN (@TimeAltKey >= 120000 AND @TimeAltKey <= 135959) _
THEN 'Lunch (12:00 PM To 13:59 PM)'
WHEN (@TimeAltKey >= 140000 AND @TimeAltKey <= 155959)_
THEN 'Mid Afternoon (14:00 PM To 15:59 PM)'
WHEN (@TimeAltKey >= 50000 AND @TimeAltKey <= 175959)_
THEN 'PM Peak (16:00 PM To 17:59 PM)'
WHEN (@TimeAltKey >= 180000 AND @TimeAltKey <= 235959)_
THEN 'Evening (18:00 PM To 23:59 PM)'
WHEN (@TimeAltKey >= 240000) THEN 'Previous Day Late Night _
(24:00 PM to '+cast( @Size as varchar(10)) +':00 PM )'
END

```

```
-- print @DayTimeBucket

INSERT into DimTime (TimeKey,TimeAltKey,[Time30],[Hour30],_
[MinuteNumber],[SecondNumber],[TimeInSeconds],[HourlyBucket],_
DayTimeBucketGroupKey,DayTimeBucket)
VALUES (@k, @TimeAltKey ,@Time30 ,@hour ,@minute,@Second , _
@TimeInSeconds, @Hour Bucket, @DayTimeBucketGroupKey, @Daytime Bucket)

SET @second = @second + 1

SET @k = @k + 1

END

SET @minute = @minute + 1

SET @second = 0

END

SET @hour = @hour + 1

SET @minute =0

END

END

Go

Exec [FillDimTime]

go

select * from DimTime//
```

Appendix C: Interview Guide

1. What is your view about the prototype?
2. If implemented can it make your work easy?
3. Do you think something should be added to make it better?
4. Can this system improve decision making and resource planning in the University?