# Design of a Data Mart for The Faculty of Computer Science

By

Feras Naser Obeidat

M.Sc. in Information Technology, University Science Malaysia, 1998

A Report Submitted in Partial Fulfillment of
The Requirements for the Degree of

Master of Computer Science

In the Graduate Academic Unit of Computer Science

Supervisor: William Hyslop, PhD, Computer Science

Examining board: Rod Cooper, MMath, Computer Science, Chair
Lev Goldfarb, PhD, Computer Science
Dana Wasson, PhD, Computer Science

This report is accepted.

Gwendolyn Davies
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

May 2002
(c) Feras Naser Obeidat, 2002

*Your file Votre référence*
*ISBN: 0-612-82550-7*
*Our file Notre référence*
*ISBN: 0-612-82550-7*

University of New Brunswick

**HARRIET IRVING LIBRARY**

**This is to authorize the Dean of Graduate Studies to deposit two copies of my thesis/report in the University Library on the following conditions:**

**(<u>DELETE</u> one of the following conditions)**

**[✓ (a)] The author agrees that the deposited copies of this thesis/report may be made available to users at the discretion of the University of New Brunswick**

**<u>OR</u>**

**~~(b) The author agrees that the deposited copies of this thesis/report may be made available to users only with her/his written permission for the period ending~~**

______________________________

**~~<u>JUSTIFICATION:</u>~~** ______________________________

______________________________

**~~After that date, it is agreed that the thesis/report may be made available to users at the discretion of the University of New Brunswick*~~**

May, 8, 2002
**Date**

**Signature of Author**

**Signature of Supervisor**

Gwendolyn Davies
**Signature of the Dean of Graduate Studies**

* **Authors should consult the "Regulations and Guides for the Preparation and Submission of Graduate Theses and Reports" for information concerning the permissible period of restricted access and for the procedures to be followed in applying for this restriction. <u>The maximum period of restricted access of a thesis is four years.</u>**

***BORROWERS** must give proper credit for any use made of this thesis, and obtain the consent of the author if it is proposed to make extensive quotations, or to reproduce the thesis in whole or in part.*

# Abstract

Data warehousing is the foundation of Decision Support System (DSS) processing. The data warehouse solves the problem of getting information out of traditional operational systems quickly, accurately, and efficiently. It provides an easy and secure way to access data, thereby contributing very large advantages to the organization.

The main objective of this Master's project was to design a data mart for the Faculty of Computer Science at the University of New Brunswick. This project arose from the need of a better source of information on which to base faculty decision-making. In developing this data mart we used a life cycle development methodology proposed by Kimball [9]. Based on user requirements, we first extracted the required data from the DataTel source system. We then restructured the data from DataTel into a data mart with a new logical schema designed to make decision support easier. To show the utility of this new data mart, we also designed and developed some sample reporting applications using Oracle Discoverer.

# Acknowledgements

First and foremost, I would like to express my sincere appreciation to my supervisor, Dr William Hyslop, for his guidance, encouragement, assistance, patience, and motivation in completing this report.

I also wish to express deep sincere feelings to my family for their encouragement throughout this long endeavor of completing this report.

## Table of Contents

**Abstract** ii

**Acknowledgments** iii

**Table of contents** iv

**List of Figures** vi

**Chapter 1 Introduction** 1

1 Objectives.......................................................................... 2
2 Overview of Report........................................................3

**Chapter 2 Background** 4

2.1 Architecture of a Data Warehouse.............................................4
2.1.1 The Source Systems..................................................5
2.1.2 Data Staging Area......................................................5
2.1.3 Presentation Server......................................................5
2.1.4 End User Tools............................................................6

2.2 Data Warehouse Development Lifecycle ......................................7
2.2.1 Project Planning and Management.................................8
2.2.2 Business Requirements Definition.................................8
2.2.3 Data Design..................................................................9
2.2.3.1 Logical Design......................................................10
2.2.3.2 Physical Data Design...........................................12
2.2.3.3 Data Staging.........................................................13
2.2.4 End User Application, Specification and Development..........14
2.2.5 Technical Architecture Design and Product Selection............15
2.2.6 Deployment and Growth.................................................15

**Chapter 3 Project Planning and Requirements Definition** 17

3.1 Project-Planning Phase.....................................................18
3.2 User Requirements Phase.................................................18

**Chapter 4 Data Design** 21

4.1 Entity Relationship Model........................................................21
4.2 Dimensional Modeling.........................................................27
4.3 Physical Design....................................................................31
4.4 Data Staging & Loading......................................................31

**Chapter 5 Application Design** 34

5.1 Oracle Discoverer................................................................34
5.1.1 Administration Edition.....................................................35
5.1.2 User Edition.......................................................................35
5.2 Sample Reports.....................................................................40

**Chapter 6 Conclusions and Future Work** 48

**Bibliography** 50

**Appendix A: Physical Table Descriptions** 51

## List of Figures

2.1 Architecture of a data warehouse........................................................4

2.2 Data warehouse lifecycle........................................................................7

2.3 A sample dimensional model..................................................................11

3.1 Lifecycle stages for our project..............................................................17

4.1 A sample ER model.................................................................................22

4.2 ER diagram for our project......................................................................22

4.3 Student entity and its attributes...............................................................23

4.4 Course entity and its attributes.................................................................24

4.5 Section entity and its attributes................................................................25

4.6 High School entity and its attributes........................................................25

4.7 High School Mark entity and its attributes...............................................26

4.8 Transfer Credit entity and its attributes.....................................................26

4.9 Registration fact table...............................................................................28

4.10 Transfer Credit fact table........................................................................29

4.11 Admission fact table...............................................................................30

4.12 A sample control file...............................................................................32

4.13 A sample data file...................................................................................33

5.1 Assigning users.........................................................................................36

5.2 Choosing data items..................................................................................36

5.3 Choosing a report layout........................................................38

5.4 Choosing data items to view........................................................38

5.5 Specifying a condition........................................................39

5.6 Defining a calculation........................................................39

5.7 Course enrollment by term........................................................41

5.8 Course enrollment (different view)........................................................42

5.9 Course enrollment (graphical view)........................................................43

5.10 Enrollment trends for specific courses........................................................44

5.11 Enrollment by program and gender........................................................45

5.12 Enrollment by program and gender (graphical)........................................................46

5.13 Enrollment by program and fee status........................................................47

# Chapter 1

# Introduction

Over the last few decades information technology has made huge leaps as corporations have moved to computerize every possible business process. At the same time there has been an increased demand to use this stored information as a basis for corporate decision-making.

The need for decision support systems predates relational database systems. However, file, hierarchical, and to some extent network-based systems, were very difficult to query directly. For that reason new tools started to arise, such as relational databases and Structured Query Language (SQL). At first, relational databases seemed like an ideal solution. They provided a database store that appeared to satisfy both the operational and query users using a single copy of the data [1]. In time, it was discovered that relational databases in themselves could not provide the whole answer.

As a separate discipline, data warehousing first emerged in the period between 1984 and 1988 [9]. The first data warehouses supported products like FOCUS or RAMIS running on VM mainframes. They were loaded and administrated separately from the operational systems, much like warehouses of today. In the early days, data warehousing concentrated on providing strategic decision-making capability to knowledge workers. End users for the data warehouse were traditionally in areas such as strategic planning, marketing, and finance. Access to information increased the quality of the decision-making. In recent years, the evolution of data warehousing has reached a new advanced stage by deployment of decision support capability throughout an organization, and even beyond its traditional boundaries to partners and customers.

Data warehousing is the foundation of Decision Support System (DSS) processing. In the data warehouse environment there is a single integrated source of data that is easily accessible by end users. Therefore the job of the business analyst becomes easier. As well, for business executives, data warehousing adds significant competitive advantage by allowing current information to be used more effectively and accurately in management decisions. The data warehouse solves the problem of getting information out of traditional operational systems quickly, accurately, and efficiently. It provides an easy and secure way to access data, thereby contributing very large advantages to the organization. A 1996 study of 62 data warehousing projects undertaken in this period shows the average return on investment of 321% for these enterprise-wide implementations in an average payback period of 2.73 years. [4]

## 1.1 Objectives

The main objective of this Master's project is to design a data mart for the Faculty of Computer Science at the University of New Brunswick. This project arose from the need of a better source of information on which to base faculty decision-making. Currently the DataTel System is used to obtain the needed information. DataTel does not provide the data in a well-structured way. As a result, its support for management decision-making is limited. Also, the DataTel user interface doesn't directly provide decision support tools.

Our objective is to restructure the data from DataTel into a data mart, which will make decision support easier. We also plan to design some sample end user applications to be used specifically for decision support purposes.

The main users of our proposed system will be the Dean, Assistant Dean and support personnel. Some possible applications of our system may be ad-hoc reporting, satisfying requests for data from external sources, and helping in recruitment and retention activities.

### 1.2 Overview of the Report:

Chapter 2 of the report will provide background material on the data warehousing area. First, the architecture of the warehouse is discussed. Then the data warehouse development lifecycle is described.

In Chapter 3, we give an overview of the data warehousing project undertaken for the Faculty of Computer Science. The steps in the project life cycle will be discussed. Particularly the stages of project planning and requirements definition will be discussed in detail.

In Chapter 4, the data design phase for this project will be discussed. This includes Entity Relationship (ER) modeling, dimensional modeling, physical data modeling and data staging.

Chapter 5 discusses the application design. First, the tool used to implement this project (Oracle Discoverer) is discussed. Examples of the reports developed are then provided.

In Chapter 6 some conclusions are provided. Also some options for future work on the project will be explored.

# Chapter 2

# Background

In this Chapter some basic background information on the data warehousing area is provided. Lets begin with a definition. According to Brakett [3].

> " A data warehouse is simply a single, complete and consistent store of data obtained from a variety of sources and made available to end users in a way they can understand and use in a business context."

The components involved in data warehouse architecture are discussed in section 2.1, while the overall data warehouse development life cycle is described in section 2.2.

## 2.1 Architecture of a Data Warehouse.

Figure 2.1 shows the basic architecture of a data warehousing system [9]. The four major components of the system are described in the following sections.

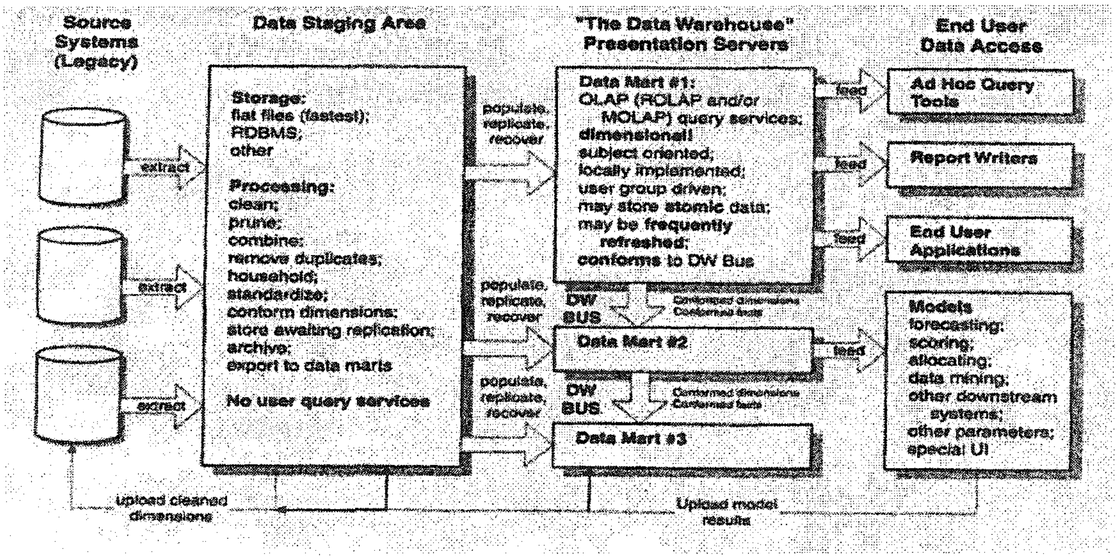


**Figure 2.1 Architecture of a data warehouse [9]**

### 2.1.1 The Source Systems

The source systems for the data warehouse are the transactional systems of an enterprise. The main goal and activity of the source systems are uptime and availability. Queries against source systems are narrow, usually part of the normal transaction flow, and severely restricted in their demands on the legacy system. Many systems are standard mainframe data storage facilities. IMS, IDMS, VSAM, flat files, and DB2 are all common in this environment.

### 2.1.2 Data Staging Area

The data staging area is the construction site for the data warehouse. The data staging area supplies the raw data needed to construct a data warehouse. At this stage the raw data is cleaned before it is loaded into the data warehouse. The data staging area also transforms, and prepares source data for use in the data warehouse. Staging the data in a relational database is common. Although it has many advantages, in some cases, the relational system imposes a lot of overhead to load, index, sort and otherwise manipulate the data.

### 2.1.3 Presentation Server

The presentation server is the machine that is accessed by the end users. On the presentation server, the data of the warehouse is well organized, clean, accurate and stored for direct querying by end users. Often the data stored on the presentation server is called a “data mart”. A data mart is a logical subset of a complete data warehouse. A data mart is the limitation of the data warehouse to a single business process or to a group of related business processes aimed toward a specific business group. It may be large or small depending on the subject area. A data mart can be viewed as a departmental warehouse; it comprises all the operational data required by a department to perform decision support.

### 2.1.4 End User Tools

Users need a collection of tools that query, analyze, and present information targeted to support a business need. User query tools provide the PC user with a user-friendly method of producing the desired queries and reports against the data from the presentation server. A minimal set of such tools would consist of: an end user data access tool, a spreadsheet, a graphics package, and a user interface facility for designing forms and facilitating the screen presentations to end users. In a relational data warehouse, such a user client maintains a session with the presentation server, sending a stream of separate SQL requests to the server. In response to these SQL requests the client receives a data set. The client then presents a screen of data, report, graph, or some other higher form of analysis to the user.

## 2.2 Data Warehouse Development Lifecycle

The data warehouse development lifecycle is shown in Figure 2.2 [2]. The lifecycle approach guarantees that the project stages are brought together in the right order and at the right time. The following figure describes the sequence of tasks required for effective data warehouse design, development, and operation. The following sections describe each task in detail.

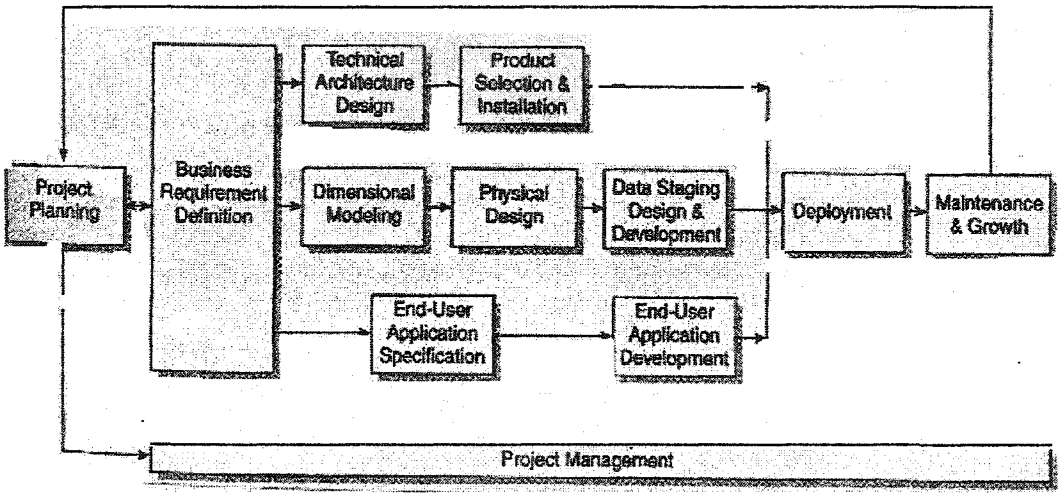

Project Planning
Business Requirement Definition
Technical Architecture Design
Product Selection & Installation
Dimensional Modeling
Physical Design
Data Staging Design & Development
End-User Application Specification
End-User Application Development
Deployment
Maintenance & Growth
Project Management

**Figure 2.2 Data warehouse lifecycle [2]**

### 2.2.1 Project Planning and Management

The Project planning activity is the most important aspect for a successful data-warehousing project. The project plan defines the scope of the data warehouse project by focusing on the resources and skills needed. Once we have a well-integrated plan we will be able to identify all tasks associated with the business dimensional lifecycle. Project management ensures that the business dimensional lifecycle activities remain on track. As illustrated in Figure 2.2, project management activities take place during the entire lifecycle. These activities focus on observing and controlling the project situation.

The project plan serves as the major component for the ongoing management of the data warehouse project. Project management includes the development of a comprehensive project communication plan that addresses both the business and information systems organizations. A good project plan provides a successful beginning and failure to do this task will result in misguided teams and missed requirements.

### 2.2.2 Business Requirements Definition

In this phase a plan is made to recognize the business requirements related to the use of the data warehouse. These requirements should support the business area. Business area rules change according to people demand, market supply and request conditions, and finally process changes. A relationship between business rules and data is very important for effective data warehouse design.

Business requirements play a significant role in the data warehouse world. Business requirements determine what kind of data must be available in the data warehouse and how it should be organized and updated. Business users and their requirements affect every decision made throughout the implementation of a data warehouse. Deployment of tasks, maintenance, and growth plans must be adapted according to user needs.

Business requirements collection starts by interviewing the key people who will be using the data. Interviews should be scheduled and organized. Interviews can be either with individuals or small groups. The advantages of the first approach are that we need to schedule a relatively limited amount of time on the interviewees' calendars. The interview also encourages a high degree of participation from interviewees, and ensures that every one has given their perception. The alternative approach is to schedule larger group sessions. This approach can be used to encourage creative ideas with a limited number of participants.

### 2.2.3 Data Design

Accurate data design is the foundation for the data warehouse. Users should play a significant role in this phase by reviewing the data model. In data design there are three integrated stages: logical modeling, physical design and data staging development. The logical and physical designs are the basis of the data warehouse. This design will transform the data resources into the final data warehouse structures. From these designs we will be able to set a strategy for data extraction and transformation steps. We will be able to approximate the overall size and administrative needs of the basic DBMS, and we will be able to plan and begin prototyping the final applications.

#### 2.2.3.1 Logical Design

The two main logical design approaches are Entity-Relationship (ER) modeling and Dimensional Modeling (DM). In this section we discuss both approaches, as well as the differences between them. In our project, we have used both the entity relationship and the dimensional modeling approaches.

Entity relationship modeling is a logical design technique that reduces redundant data from a system. An entity relationship model is normally composed of many logical entities. Each of these entities is normally converted into a physical table when the database is implemented. Entity relationship modeling has some disadvantages when used in a data warehousing design environment. In data warehousing we are dealing with enormous amounts of data and the use of entity-relationship modeling has some problems in processing and retrieving huge amount of data. Due to these limitations, most data warehouse designers prefer to use a simpler technique called "Dimensional Modeling" that achieves the aim of building a successful data warehouse.

Dimensional modeling is a logical design procedure that represents data in a standard supporting structure that allows for high performance access. A sample dimensional model is shown in Figure 2.3. Every dimensional model is composed of one table with multi part key, called the fact table, and a set of minor tables called dimension tables. Each dimension table has a primary key that is a foreign key that references one of the components of the multi part key in the fact table. These characteristics form a shape that looks like a star, and for that reason is called a star join. Dimension tables contain textual information that describes the table. The dimension attributes are the source of the interesting constraints in data warehouse queries and the source of row headers in the structured query language (SQL) answer set. The dimensional model identifies the fact table, related dimensions, and attributes. The logical database design is considered completed when suitable table structures and primary/foreign key relationships are identified.

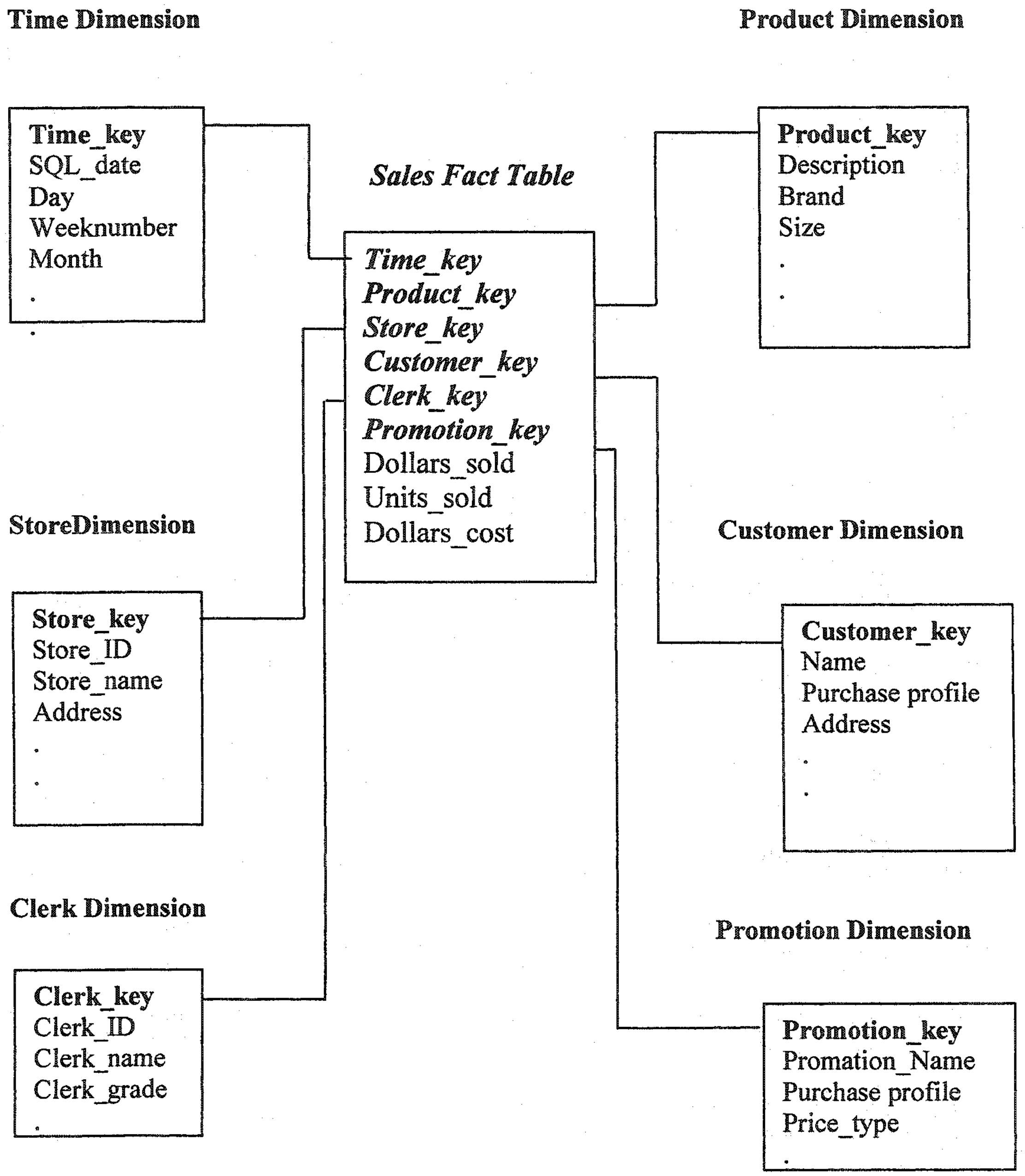


**Figure 2.3 A sample dimensional model.**

Dimensional modeling is ideal for data warehousing design, since it is able to process a huge amount of data. Data warehouse applications generally retrieve a huge amount of fact table records; sometimes it may reach a million rows. The dimensional model has many advantages for data warehousing compared with the entity-relationship model. First, the dimensional model is a predictable, standard framework. Reports writers, query tools, and user interfaces can all make better assumptions about the dimensional model because the dimensional model is more understandable by the end users.

A second advantage of the dimensional modeling is that of flexibility. The dimensional model has the capability to accept unexpected updates, such as inserting of new data elements and new design decisions. So the data should not need to be reloaded. Dimensional models can be updated and changed easily, while allowing old applications to continue to run without producing any unexpected results.

A final advantage of the dimensional model is the growing body of administrative utilities and software processes that manage and use aggregates. Aggregates are summary records that are used for better query performance.

#### 2.2.3.2 Physical Data Design.

The details of the physical data model are highly dependent on the individual factors of the project: the logical data model, the warehouse RDBMS, data volumes, and access tools. In the physical database design stage, the main goal is to identify the physical structures, which are necessary to provide for the logical database design. Providing the primary elements of this process include defining naming standards, setting up the database environments, and determining initial indexing strategies. Also it may involve physical details like partitioning, tablespaces, and disk file layout.

#### 2.2.3.3 Data Staging

The goal of data staging is to get the data ready to be loaded into a presentation server. The data staging process has three major steps: extraction, transformation, and load.

The extract process identifies data quality issues within the operational source systems. The extract step is the first step of getting data into the data warehouse environment. Extraction includes reading and understanding the source data, and then copying the parts that are needed to the data staging area for supplementary work.

The goal of the data transformation process is to improve the quality and usability of data in the data warehouse. Transformation processes include: cleaning the data, dealing with missing data elements, purging selected fields from legacy data that are not useful for data warehouse, and combining data sources. The most common and easiest form of data transformation is data type conversion. At the basic level, data that comes from a mainframe has to be converted from EBCDIC character set to ASCII. In addition the logical and physical models may specify that values be stored in a different data type than the source.

At the end of the transformation process, the data will in the form of a load record. Loading, in the data warehouse environment, usually takes the form of producing dimension tables and fact tables and then presenting these tables to the bulk loading services for each data mart. The bulk loader is a utility built to get data into the database as quickly as possible.

### 2.2.4 End User Application Specification and Development.

End user applications add a significant value to the data-warehousing environment by meeting the organization's range of data access. Successful warehouses contain a selection of tools that meet a variety of user requirements. Users at this stage should have a major role in selecting the necessary tools and in the development of applications, so the IT department should take into consideration the user contribution in selecting the proper end user tools.

Application specifications describe the report pattern and required calculations. These specifications ensure that the development team and business users have a common understanding of the applications to be delivered. There are four major steps in the specification process:

- Determining the preliminary template set by developing target list of reports templates.
- Determining template standards to help the users understand the type of reports, its contents, sources and timing.
- Developing detailed template specifications by collecting the thoughts and ideas produced during the requirements gathering process.
- Developing the browse strategy that will help the users to quickly find what they want.

In the end user application development stage the steps required are highly dependent upon each organization and data access tool to be used. The development of end user applications involves configuring the tool and constructing the specified reports. These applications are built using an advanced data access tool that provides significant productivity gains for the application development team. It also offers a powerful mechanism for business users to easily modify existing report outlines.

### 2.2.5 Technical Architecture Design and Product Selection

The technical architecture design establishes the overall architecture framework and vision. This phase involves a data acquisition component used for extracting data from source files and databases. It also includes a data manager component for creating, managing and accessing the data warehouse data. Finally, a data delivery component is required for distributing warehouse data to other external systems.

Next is the product selection and installation phase. Using the technical architecture design as a framework, specific architecture components such as the hardware platform, data base management system, or data access tool will need to be evaluated and selected. A standard technical evaluation process is defined along with specific evaluation factors for each structural design component. Once the products have been assessed and selected, they are then installed and tested to ensure an appropriate end-to-end combination within the data warehouse environment.

### 2.2.6 Deployment and Growth

Data warehouse acceptance and performance issues should be measured regularly. A data warehouse requires an approach that differs from traditional operational systems.

In traditional systems development, deployment signals a hand-off from the development team to the maintenance organization. Data in a data warehouse requires continuous care and cannot be simply turned over to a separate maintenance organization. We need to ensure that there will be sufficient financial support for data warehouse maintenance resources during the initial construction phase.

Deployment represents the convergence of data warehouse technology, data, and end user applications. Extensive planning is required to ensure that these pieces fit together properly. Business user aspects of the convergence must be developed and delivered. In addition, user support and communication, strategies should be established before any business users have access to the data warehouse.

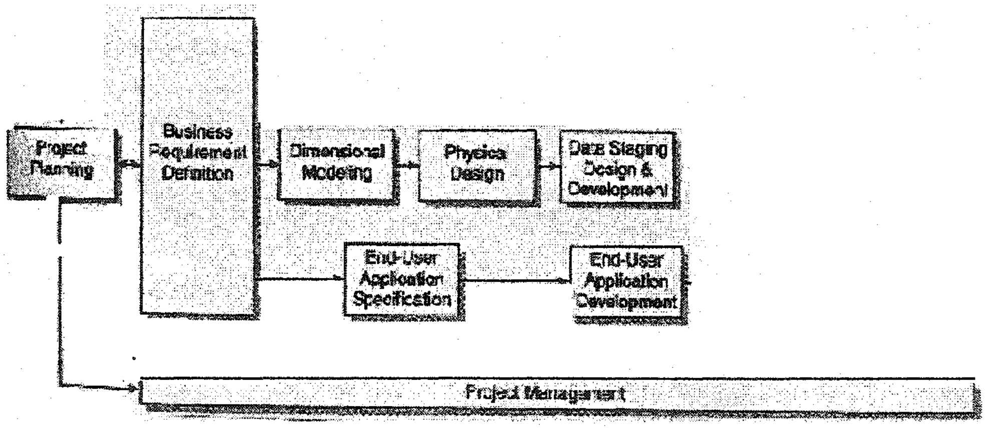
In this Chapter the data warehouse development lifecycle from a general perspective has been discussed. In the coming Chapters we will discuss in detail the lifecycle of this project. Chapter 3 discusses the project planning and requirements definitions phase. In Chapter 4 we discuss data design and finally in Chapter 5 application design will be discussed.

# Chapter 3

## Project Planning and Requirements Definition

The main objective of this project is to design a data mart for the Faculty of Computer Science. The goal is to restructure the current source data into a data mart, which will make decision support easier. We also plan to design some sample end user applications to be used specifically for decision support purposes.

Figure 3.1 represents the life cycle for the project. This project is mainly concerned with the following phases: project planning, business requirements definition, dimensional modeling, physical design, data staging and end user application design. There will be less focus on implementation issues, although a prototype system will be developed. In this Chapter we will discuss the planning and requirements analysis phases.



Project Planning
Business Requirement Definition
Dimensional Modeling
Physical Design
Data Staging Design & Development
End-User Application Specification
End-User Application Development
Project Management

Figure 3.1 Lifecycle stages for our project [2]

### 3.1 Project-Planning Phase.

Since the project plan determines the scope of the project, our main concern was to develop a solid project plan that would provide us with a successful beginning for our project. In this phase we first set the timetable for accomplishing the required tasks. Then we started looking at the necessary resources to accomplish the tasks.

First we defined the scope of the project to implement a data mart for the Faculty of Computer Science. It would only deal with undergraduate student information stored at the faculty level. The main purpose is to provide the users with the information for better decision-making. Operational information requirements were not included. The users of the system will be those people who are responsible for student data in the Faculty of Computer Science. These users include the Dean, Assistant Dean and support personnel. Some possible applications of our system would be ad-hoc reporting, satisfying requests for data from external sources, and helping in recruitment and retention activities.

The source system for our project consists of four files extracted from Datatel. The first file is concerned with student personal data. The second file is concerned with course registration information. The third file involves transfer credit information, and the fourth file concerns student admission data.

### 3.2 User Requirements Phase.

In order to accomplish a successful warehouse project we must take into account user requirements from the people who are responsible for student data. We must discover their application demands, and the information required to be used for better decision-making. For that purpose, we carried out several interviews to determine what kind of data must be available in our data mart and how this data should be organized and manipulated to achieve our goal.

In particular we conducted interviews with Dr. Rauch. He also provided us with documentation and sample reports from the current system. As Assistant Dean, he is the person primarily responsible for undergraduate student data. The current system was written by Dr. Rauch using the WatFile system. Basically, the raw data is downloaded from DataTel and a set of customized reporting functions are run against it. The primary use of the system is for degree audit, class lists, Dean's list, and other specific tasks. There is no ad-hoc query or management reporting capability.

Upon examining the current system, we found it suffers from the following problems:

- The data extraction process is manual and very time consuming
- There are issues of data quality; mostly from improper data entry in source systems
- There is no provision for historical data; each time the data is refreshed the previous data is overwritten.
- The current reporting functions require a large degree of manual intervention and are thus complex and time consuming.
- There is no ad-hoc query capability; therefore all new reports must be custom coded.
- The current system is not well documented and thus will be hard to maintain in the future especially after the retirement of Dr. Rauch.

Based on these limitations of the current system, and the needs expressed in interviews, we concluded that the new system would have the following requirements:

- The data extraction and refresh process must be automated as much as possible. Hopefully it can be designed to be run as a scheduled process.
- The existing raw data files are not in the proper format or structure to adequately provide for decision support functionality. A new data schema will have to be designed. This new schema must also account for storing historical data.
- Due to data quality issues, there must be substantial data cleaning performed before data is loaded into the new data schema.
- Existing reporting functions must be rebuilt using an ad-hoc query facility. This way they may be easily changed or customized in the future.
- New management reports must be designed to aid in decision support within the Faculty.
- The new reports must be well documented and easily maintained and extended.

The next two Chapters discuss a data mart constructed to satisfy these requirements. Chapter 4 discusses the design of a new data schema. In Chapter 5 we discuss the design of new reporting functions based on this schema.

# Chapter 4

# Data Design

In this Chapter the data design phase of the project will be discussed in detail. Entity Relationship modeling is discussed in section 4.1 and Dimensional Modeling (DM) is explained in section 4.2. The physical data design stage is discussed in section 4.3, while the data staging phase of the project is discussed in section 4.4.

We have used the two approachs for our logical design: Entity Relationship (ER) & Dimensional Modeling (DM) in order to explore the differences between the two approaches. Entity Relationship (ER) is a more flexible approach for ad-hoc reporting and operational applications. Also the mapping into physical tables using Entity Relationship (ER) is more flexible. Dimensional Modeling (DM) is more approprate for analytical data cubes in OLAP applications and for data mining.

## 4.1 Entity Relationship Model.

The Entity Relationship (ER) model is a conceptual data model used to define a conceptual schema. It is defined in terms of entities, relationships and attributes. An entity is an object with an independent existence. A relationship as an association between two or more entities. An attribute is a property of an entity. Each entity has a key attribute that uniquely identifies it. Relationship cardinality is the number of relationship instances an entity participate in (1:1, 1:N, N:N).

Figure 4.1 represents a sample of ER model. In this example we have two entities **Employee** and **Department**. The relationship **works for** associates the two entities by a many to one (N:1) relationship which indicates that the employee works for only one department, and one department may have many employees.

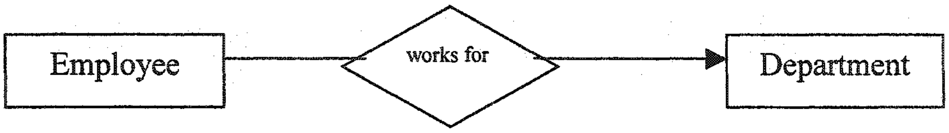


**Figure 4.1 A sample ER model**

Figure 4.2 represents the ER model for our project. The entities are: STUDENT, SECTION, COURSE, HIGH SCHOOL, HIGH SCHOOL MARK and TRANSFER CREDIT. In the following sections we will give more explanation for each entity.

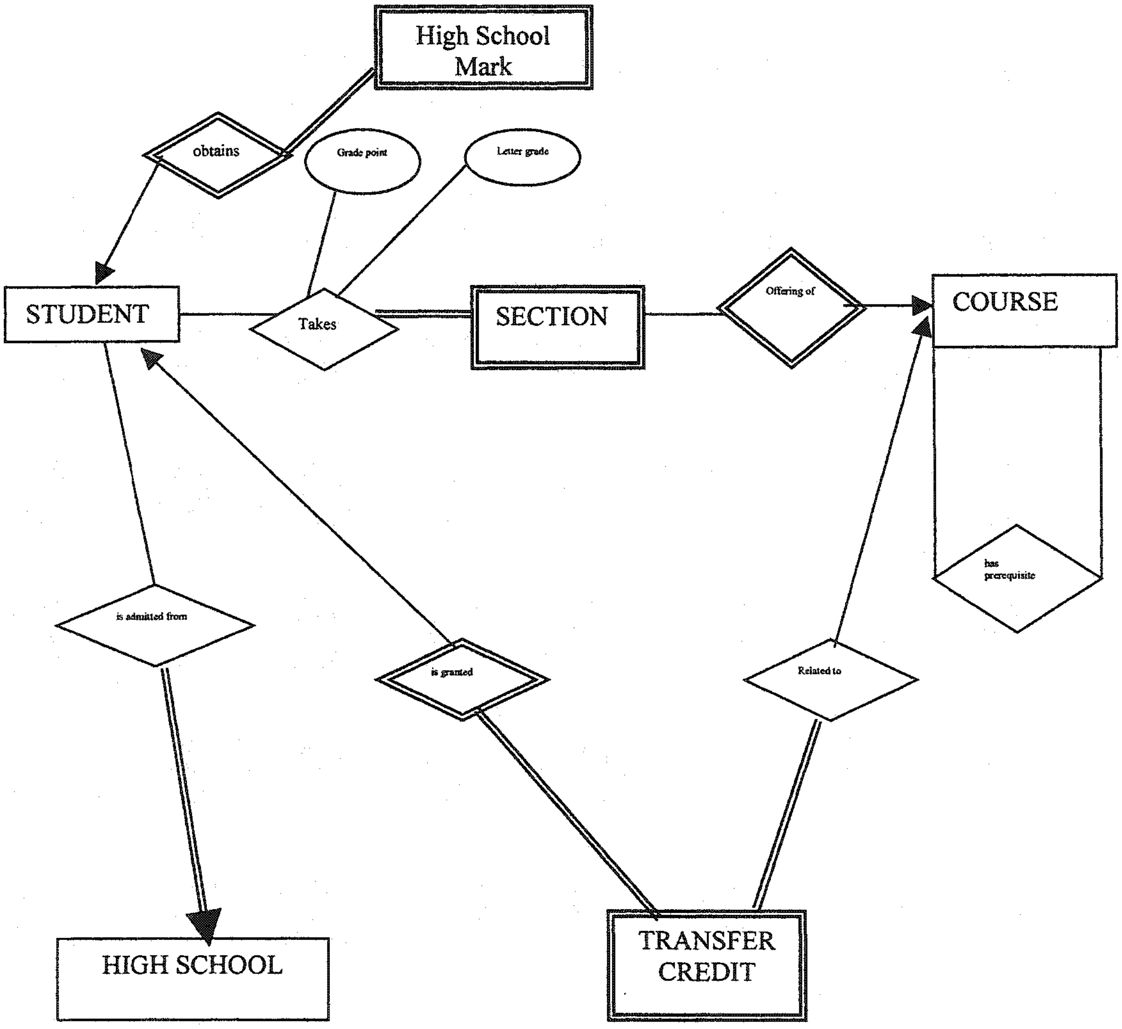


**Figure 4.2 ER diagram of our project**

Figure 4.3 represents the entity STUDENT with its all attributes. The key attribute for this entity is **Student_Id**. The STUDENT entity has many relationships with other entities. The relationship **Takes** associates the two entities STUDENT and SECTION. The purpose of this entity is to identify which sections a student has been enrolled in. The relationship attributes grade point and letter grade tell the mark assigned. STUDENT entity also has a relationship with one HIGH_SCHOOL entity. Each STUDENT entity may also be associated with one or more HIGH SCHOOL MARK entities. As well each STUDENT may be associated with one or more TRANSFER CREDIT entities.

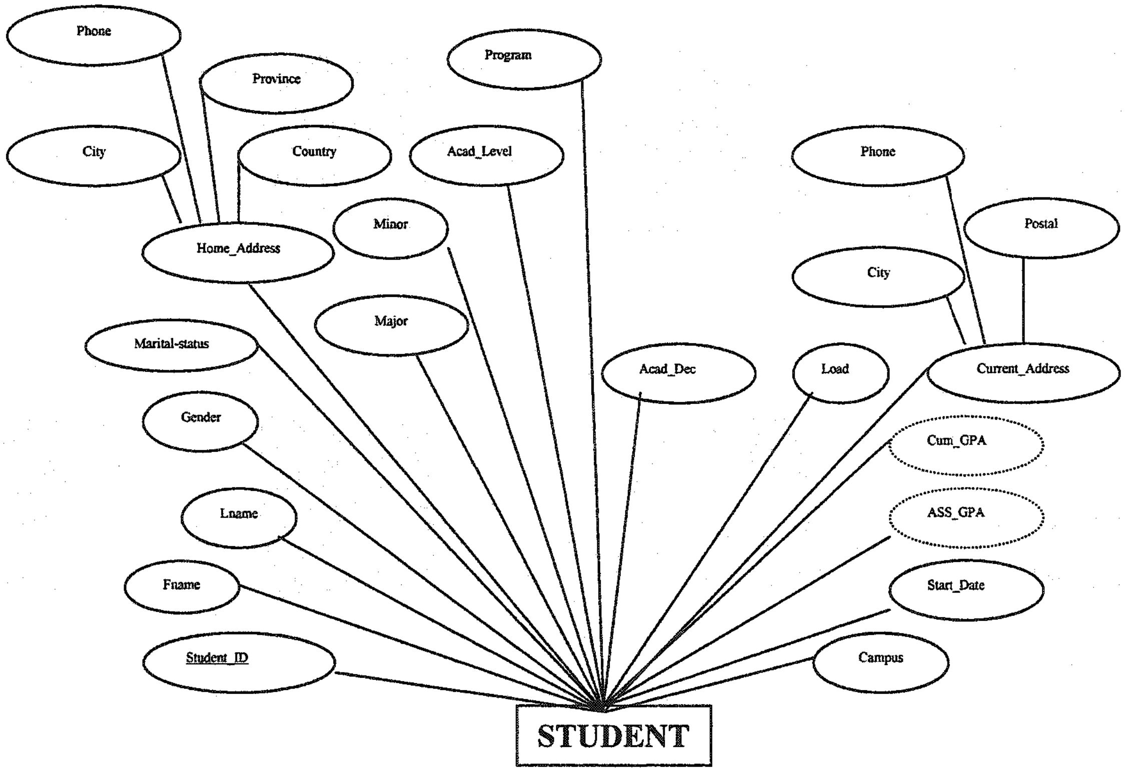


**Figure 4.3 Student entity and its attributes**

Figure 4.4 represents the COURSE entity with its all attributes. The key attribute for this entity is **Course_No.** The entity COURSE has a **Prerequiste** relationship which determines one or more prerequisite courses for a specific course. A given COURSE entity may have several offerings as represented by SECTION entities. A course may also be associated with several TRANSFER CREDIT entities.

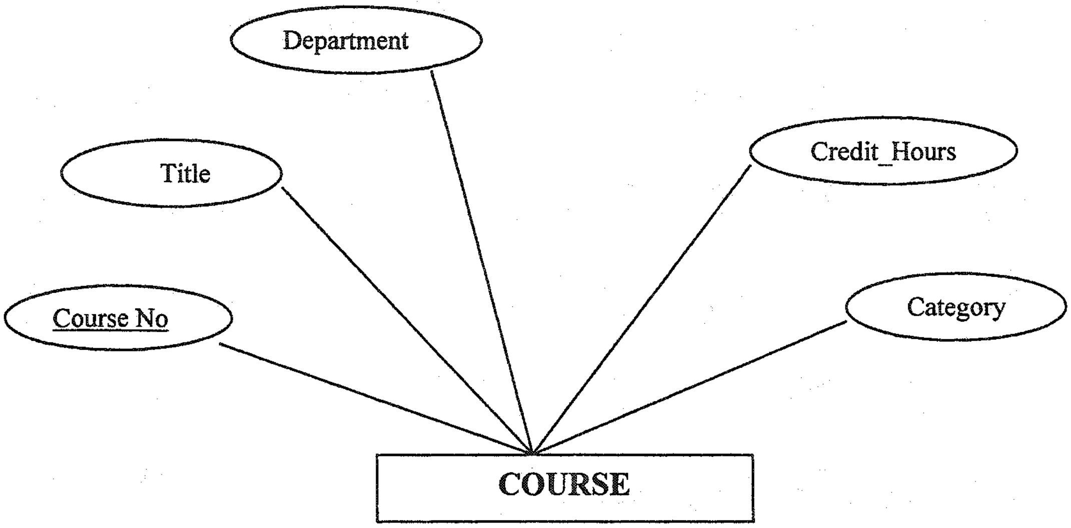


**Figure 4.4 Course entity and its attributes**

Figure 4.5 represents the SECTION entities with its all attributes. The SECTION entity is considered a weak entity since its existence depends on a given COURSE (owner) entity. In the figure a double rectangle identifies it as a weak entity. The double diamond relationship shows its owner entity. The partial key attributes for this enitiy are **(Section No, Term,Year)** which identify a section for a given COURSE entity.

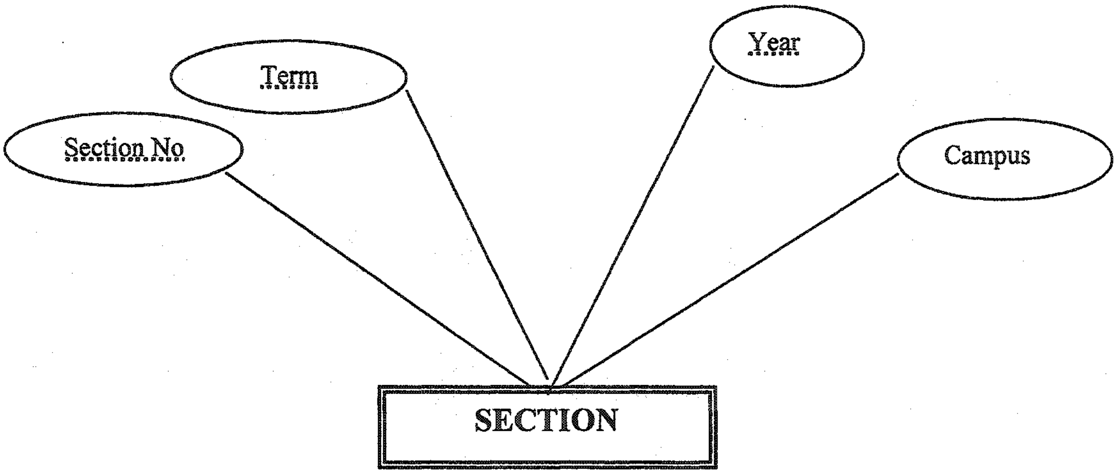


**Figure 4.5 Section entity and its attributes**

Figure 4.6 represents the HIGH SCHOOL entity with its all attributes. The key attribute for this entity is **High School Id**. Each student is associated with one HIGH SCHOOL entity.

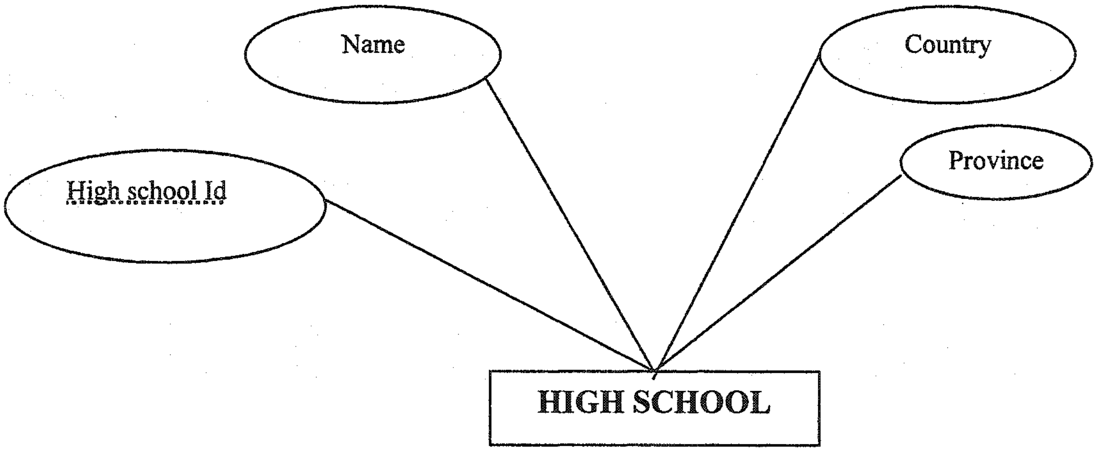


**Figure 4.6 High School entity and its attributes**

Figure 4.7 represents the weak entity **HIGH SCHOOL MARK** with its all attributes. The partial key attribute for this entity is the **Subject**. The attribute **Subject** determines the subject the student has taken and **Mark** attribute determines the mark achieved in the subject. Each STUDENT entity may be associated with a set of HIGH SCHOOL MARK entities.

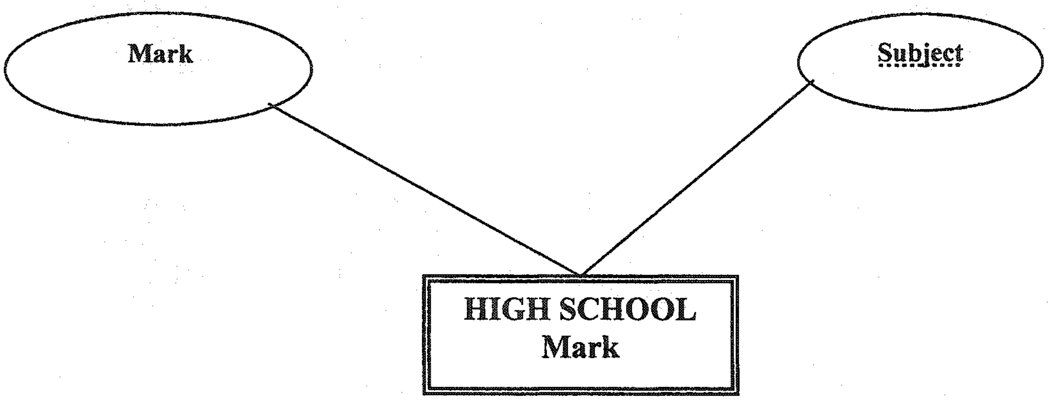


**Figure 4.7 High School Mark entity and its attributes**

Figure 4.8 represents the weak entity TRANSFER CREDIT with its attributes. The partial key attribute for this entity is **Institute name**. Each STUDENT entity may be associated with a set of TRANSFER CREDIT entities. Also each TRANSFER CREDIT entity is associated with a single COURSE entity.

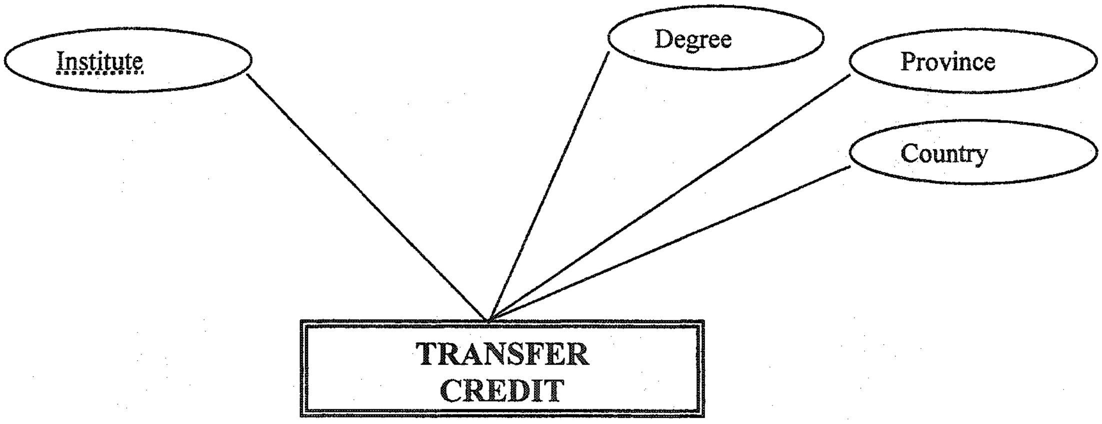


**Figure 4.8 Transfer Credit entity and its attributes**

## 4.2 Dimensional Modeling.

In the next phase we converted the entity relationship model into a dimensional model. The structure of our dimensional model is in the form of a star. The center of the star contains a *fact table*. Dimensions tables surround each fact table, and every dimensional table has it is own primary key.

In our project we use three fact tables: **Registration Fact Table, Transfer Fact Table**, and **Admission Fact Table.**

Figure 4.9 represents the first dimensional model for our project. The ***Registration Fact Table*** is surrounded by dimensions tables, **Student, Course, Prerequisite and Section**. Throughout this model we can get summarized reports concerning registration information such as: students transcript, number of students enrolled in each course, number of male and female in each major. etc.

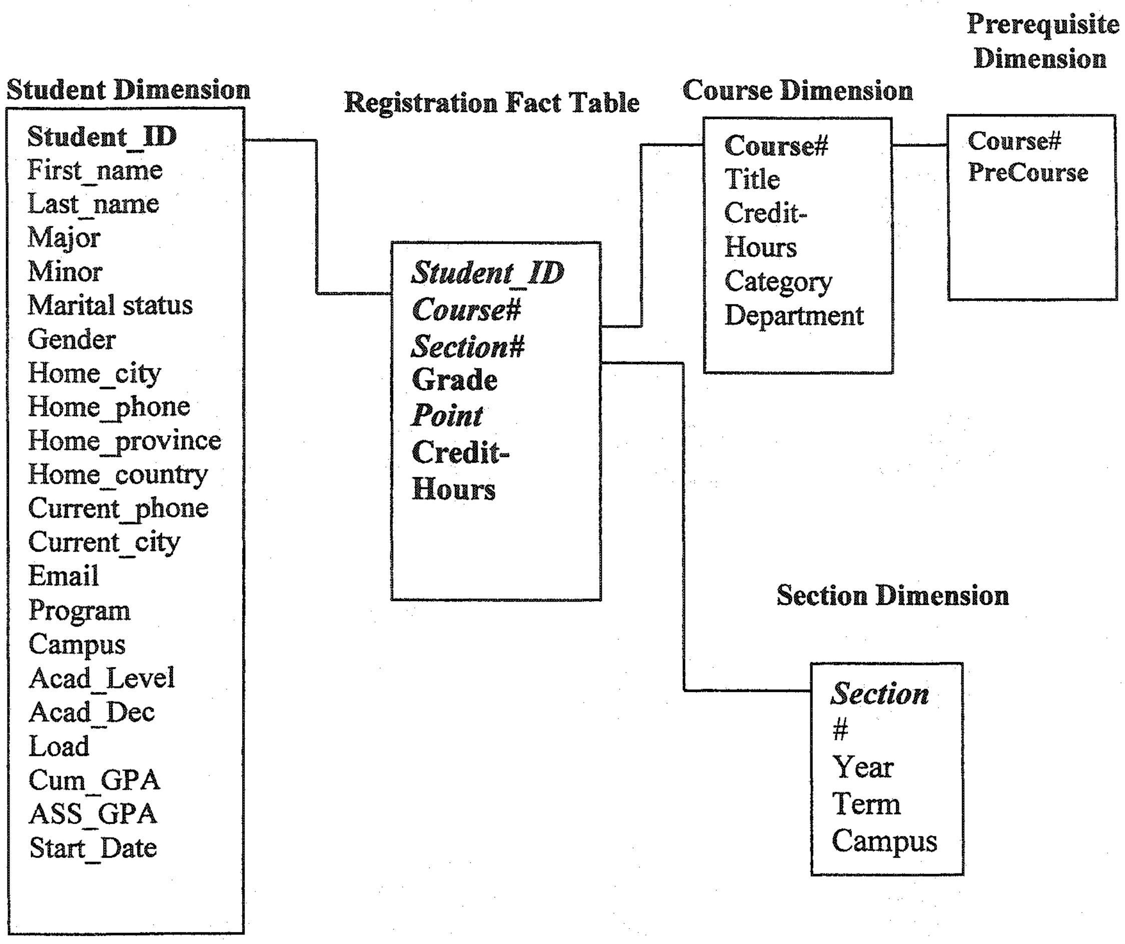


**Figure 4.9 Registration fact table**

Figure 4.10 represents the second dimensional model. The fact table in the following diagram is **Transfer Credit Fact Table,** which is surrounded by dimensions tables, **Student, Course,** and **Institution.** Through this model we can get summarized reports regarding transfer credit status. Information on how many credit hours are transferred for each student and the degree the student is enrolled in is easily obtained.

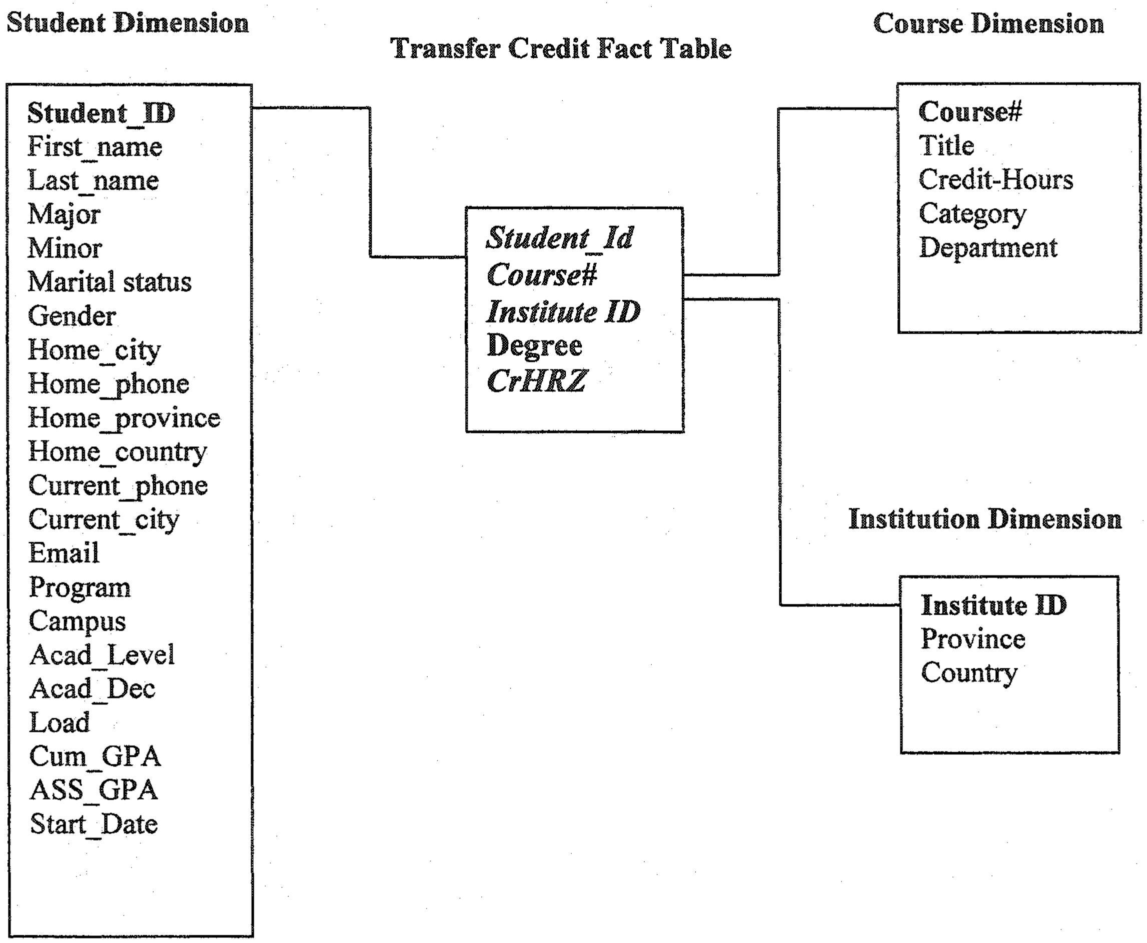


**Figure 4.10 Transfer Credit fact table**

Figure 4.11 represents the third dimensional model. The fact table is the **Admission Fact Table**. This fact table is surrounded by dimensions tables, **Student** and **High School.** From this model we can get summarized reports regarding admission status. We can get information about each student and the marks he has achieved in high school.

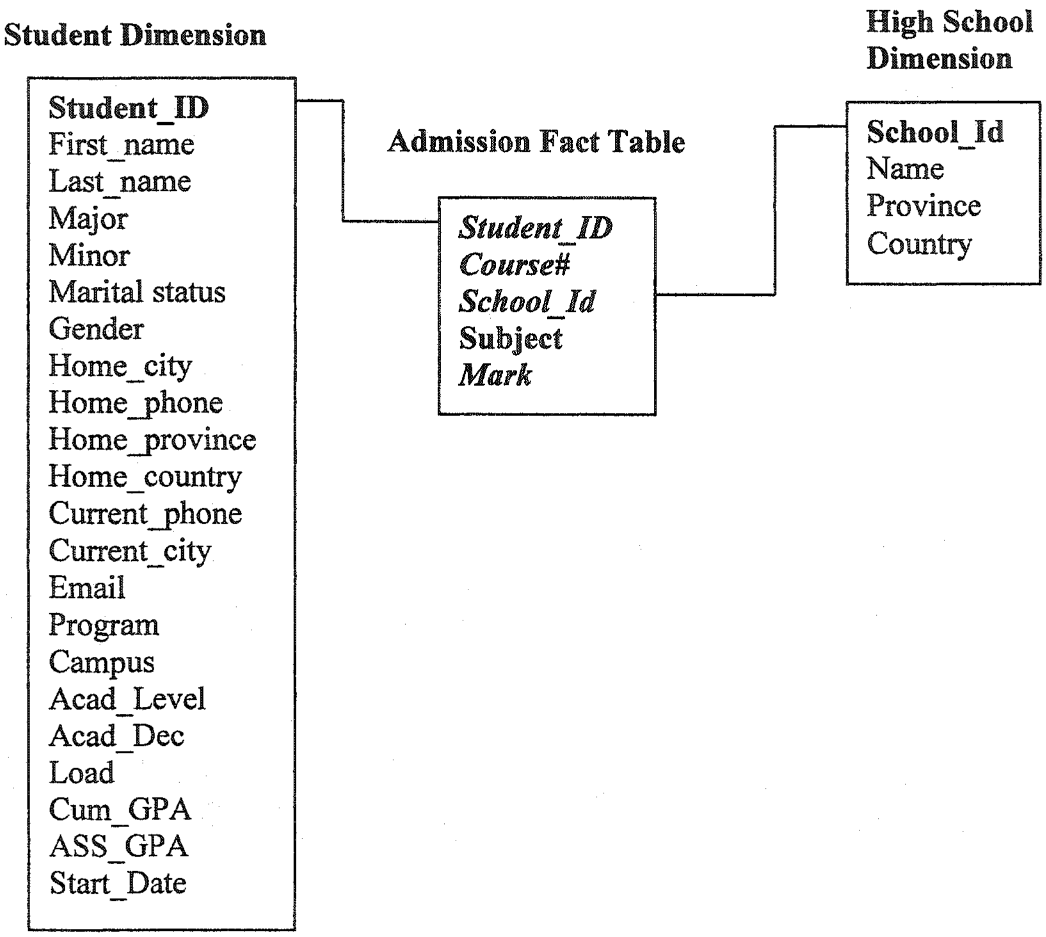


Figure 4.11 Admission fact table

## 4.3 Physical Design:

The next phase involves converting the logical design into a physical design. There are two approaches based on whether Entity Relationship (ER) or Dimensional Modeling (DM) modeling is used. For ER model there are defined mapping rules [10]. In the dimensional model approach the fact and dimensions tables are already mapped. In our project we have used the ER mapped tables for our physical design. This allows for more flexibility in doing ad-hoc reporting and future operational applications (such as degree audit) that may be built using these tables. A future project could choose to use the dimensional tables if OLAP or data mining were more of interest.

The physical design for our project has the following tables: STUDENT, COURSE, SECTION, HIGH_SCHOOL, HIGH_SCHOOL_MARK, TRANSFER_CREDIT, TAKES, and PREREQUISITE. Details of these tables are given in Appendix A.

## 4.4 Data Staging & Loading.

The data-staging phase is where raw data is loaded, cleaned, combined, and quickly ported to a presentation server. The presentation server in our project is an Oracle database. Once the data is well prepared, the load process is quite straightforward. All that we need to know is how to load the organized data into the target tables. For loading purposes we need to use bulk loader utilities. Bulk load utilities tend to be script driven and come with a range of parameters and transformation capabilities. In our project the source system is DataTel. We extracted the data from DataTel and loaded it to our target database using the Oracle SQL*Loader utility.

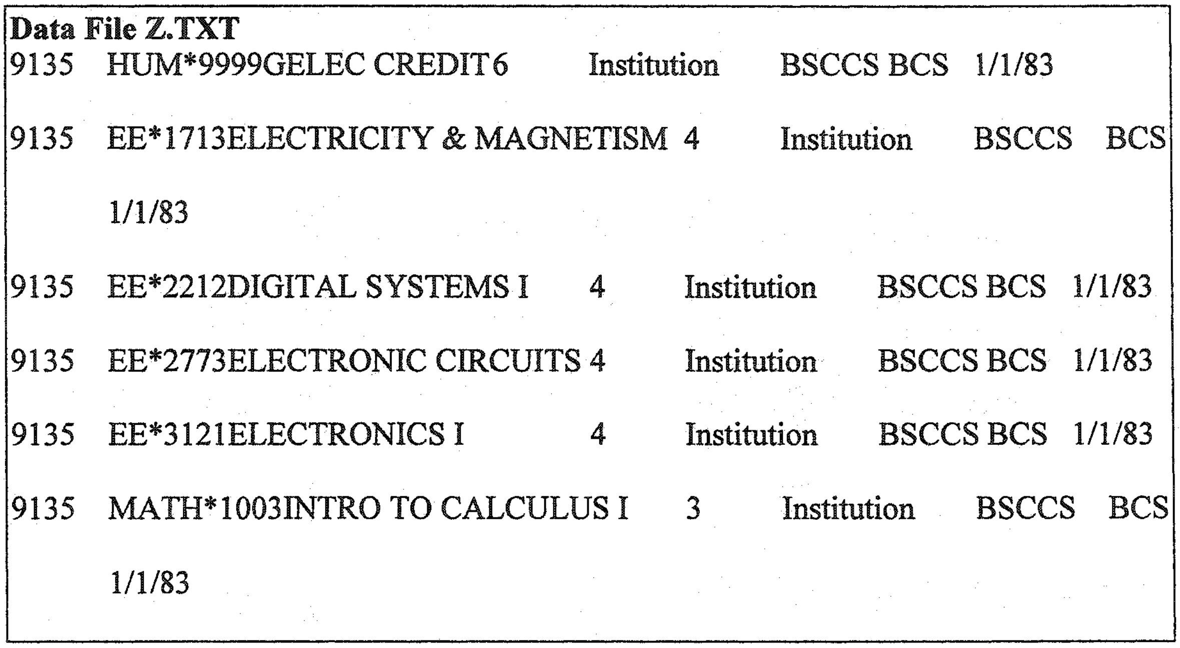
SQL*Loader is a bulk loader utility used for moving data from external files into the Oracle database. SQL*Loader supports various load formats, selective loading, and multi-table loads. To use the Oracle bulk loader, we need a control file, which specifies how data should be loaded into the database; and a data file, which specifies what data should be loaded. A sample control file is shown in Figure 4.12 and a sample data file is shown in Figure 4.13.

The command used to load the data is a follows:

```
sqlldr USERID=feras@fcs CONTROL=run2.txt LOG=run2.log
```

```
Control File RUN2.TXT

load data

infile 'z.txt'

into table rauch2

insert

FIELDS TERMINATED BY '          '

(Student_ID,CourseTitle,Credit_Hrs,Institution,Transfer_Degrees,Transfer_Date)
```

**Figure 4.12 A sample control file**



```
Data File Z.TXT
9135   HUM*9999GELEC CREDIT6        Institution     BSCCS BCS   1/1/83

9135   EE*1713ELECTRICITY & MAGNETISM  4      Institution     BSCCS    BCS

       1/1/83

9135   EE*2212DIGITAL SYSTEMS I     4      Institution     BSCCS BCS   1/1/83

9135   EE*2773ELECTRONIC CIRCUITS 4      Institution     BSCCS BCS   1/1/83

9135   EE*3121ELECTRONICS I         4      Institution     BSCCS BCS   1/1/83

9135   MATH*1003INTRO TO CALCULUS I    3      Institution     BSCCS    BCS

       1/1/83
```

**Figure 4.13 A sample data file**

In this Chapter we have discussed the two logical design approaches Entity Relationship (ER) & Dimensional Modeling (DM). We also discussed the physical design and data staging process for our project. In the coming Chapter we discuss application design and provide some sample reports.

# Chapter 5

# Application Design

In this Chapter we will discuss the design and development of some reporting applications used for decision support purposes. In section 5.1 we describe Oracle Discoverer the tool used to implement these applications. In section 5.2 we provide some sample reports.

## 5.1 Oracle Discoverer.

Oracle Discoverer is an easy to use ad hoc query and analysis tool for high-performance end user access to information. Discoverer enables business users to gain immediate access to information from relational data warehouses, data marts, or online transaction processing system. End users have access to information via a wizard-based interface. Extensive help, cue cards, and quick tours provide additional assistance. Oracle Discoverer lets users perform ad hoc queries, analyze and format query results, prepare the results for presentation, and manage data in a way that is meaningful to their business situations. It separates the more difficult database administration tasks from the simpler querying and reporting tasks. This allows analysts, managers, and other information workers to easily get their work done without having to know about databases or SQL. Oracle Discoverer is divided into two parts: the Administration Edition and the User Edition.

### 5.1.1 Administration Edition.

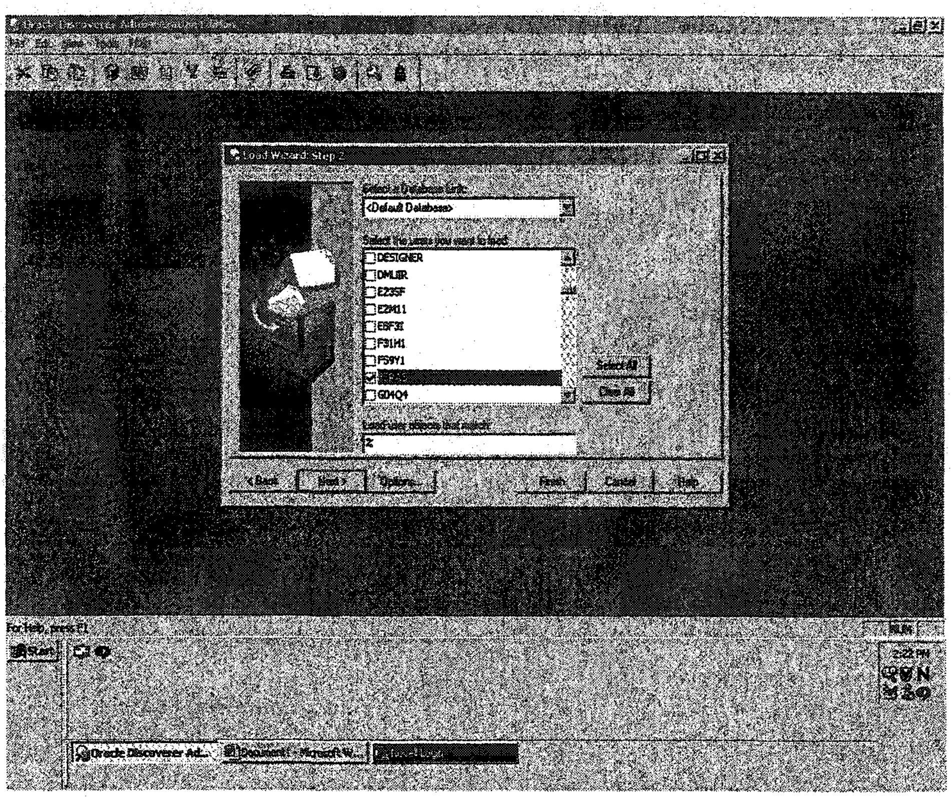
In the Administration Edition the administrator designs the business area that be granted to specific users. A business area is a user view of the data that is composed of one or more physical tables from the data warehouse. In essence, it is a data mart. The user can then access this business area from the User Edition.

The steps needed to create a business area are summarized as followings:

1. First a connection to the Oracle server must be established. The user name and password should have authorization to access the system.
2. Then a wizard is used to create a new business area. This business area will contain the tables to be used.
3. The next step is to assign the users who are going to use this business area. Figure 5.1 gives an example of this process.
4. Then we can choose the tables to be included in the business area as shown in Figure 5.2.
5. Finally, a name is assigned to business area. The users will use this name to access the business area from the User Edition.

### 5.1.2 User Edition.

In the User Edition the user creates queries and reports based on the business area created in the Administration Edition. In Discoverer terms, each report is called a *workbook.* We will take the first report represented in Section 5.2 (shown in Figure 5.7) as an example to explain the steps required in creating a report. This report gives detailed information about the number of students taking each course over a number of terms.



Load Wizard: Step 2

Select a Database Link:

<Default Database>

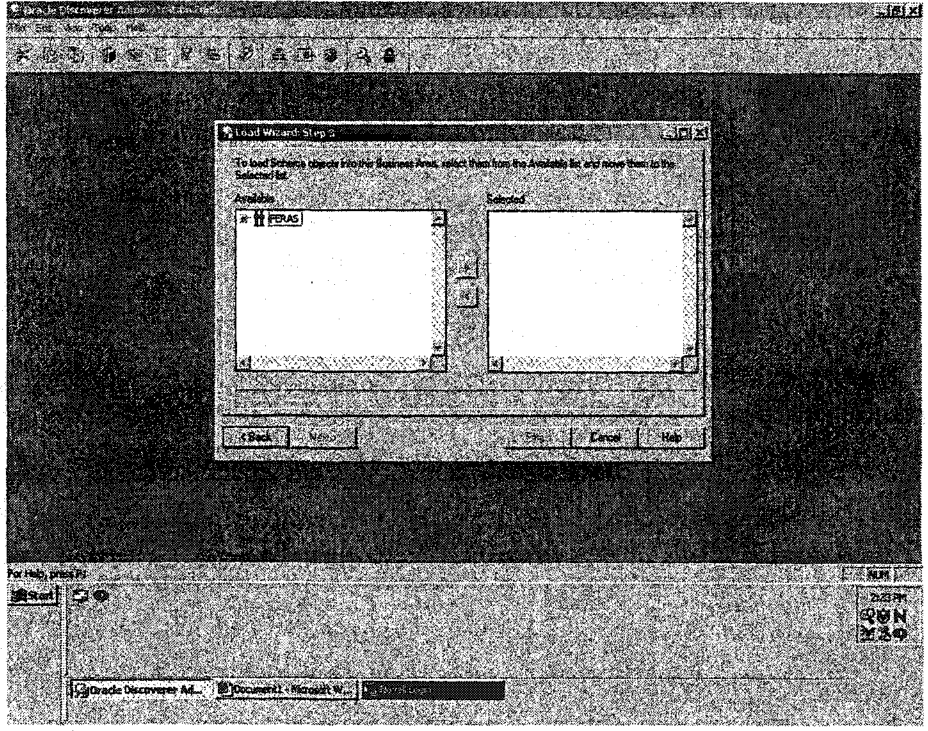
Select the users you want to load:

□ DESIGNER
□ DMLIIR
□ E23SF
□ E2M11
□ EBF3I
□ F31H1
□ F59Y1
☑ [illegible]
□ GD4Q4

Load user objects that match:

%

Figure 5.1 Assigning users



Load Wizard: Step 3

To load Schema objects into the Business Area, select them from the Available list and move them to the Selected list.

Available

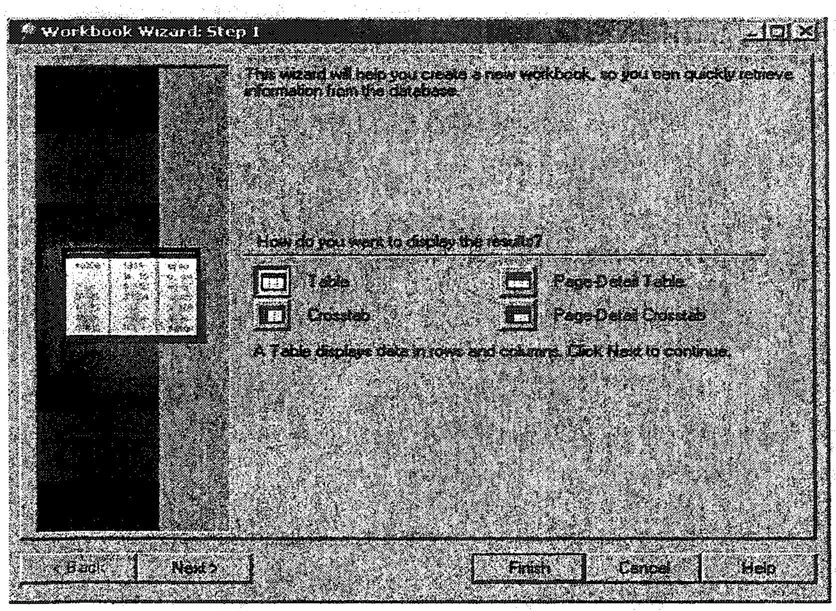
FERAS

Selected

Figure 5.2 Choosing data items

The process of creating this report is as follows:

1. Invoke a wizard to create a new workbook.
2. We then have to select the business area to use for data for the report. This business area must have been previously created in the Administration Edition.
3. Then we determine the way of displaying the results. The four options for layout are shown in Figure 5.3. In our example we choose the "Crosstab" option.
4. In the next step the business area is selected. We can then see the tables or items contained in this business area. We then choose which items we want to use and view for our report. Figure 5.4 shows an example of this process.
5. We may also need to apply a query against the data. This is done by specifying one or more conditions. To specify a condition, the Conditions Icon is clicked and then we can start designing our conditions as shown in Figure 5.5. In our example we are limiting the data to only include courses offered by the 'CS' faculty.
6. Usually some calculations are also required for a report. In our example we wanted to show the number of students who took each course. We choose Calculations from the menu and then specify the required calculation as shown in Figure 5.6. We have chosen COUNT from Functions, and then we selected Student_Id from Items.

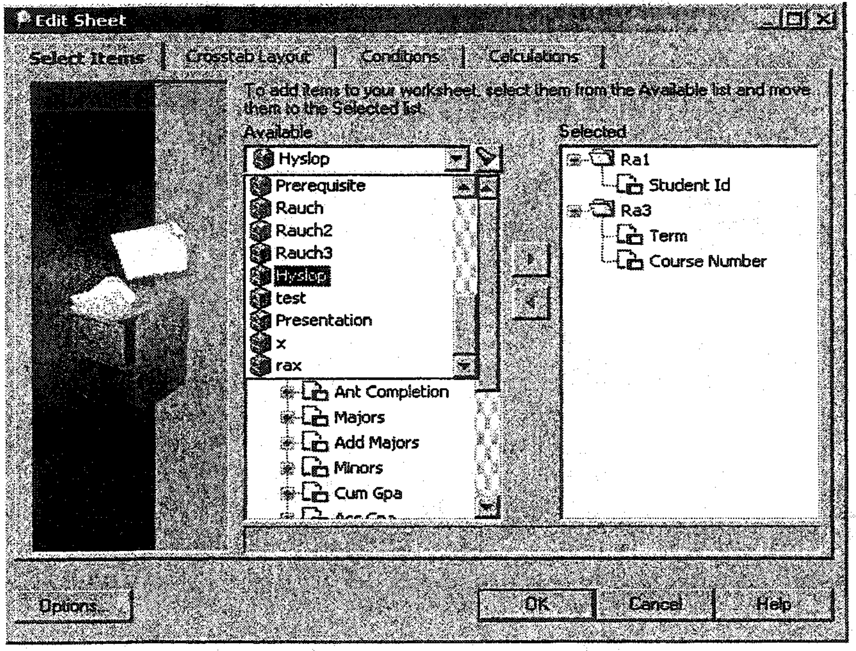


This wizard will help you create a new workbook, so you can quickly retrieve information from the database.

How do you want to display the results?

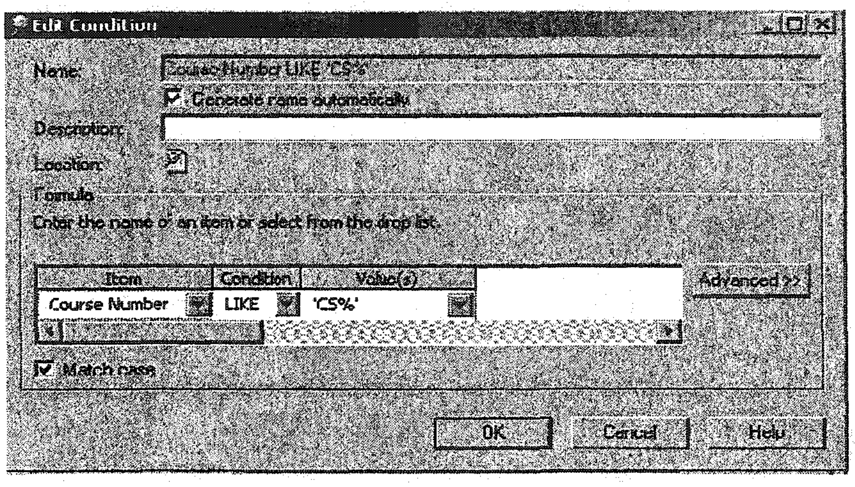
A Table displays data in rows and columns. Click Next to continue.

**Figure 5.3 Choosing a report layout**



To add items to your worksheet, select them from the Available list and move them to the Selected list.

**Figure 5.4 Choosing data items to view**



Name: Course Number LIKE 'CS%'

☑ Generate name automatically

Description:

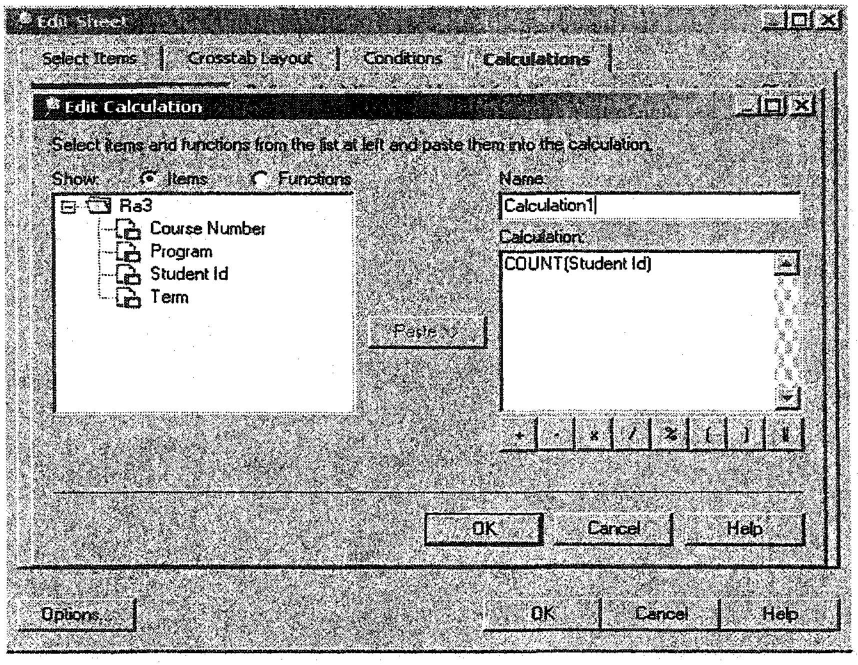
Formula

Enter the name of an item or select from the drop list.

| Item | Condition | Value(s) |
|---|---|---|
| Course Number | LIKE | 'CS%' |

☑ Match case

Figure 5.5 Specifying a condition



Select items and functions from the list at left and paste them into the calculation.

Show: ● Items ○ Functions

Name: Calculation1

Calculation: COUNT(Student Id)

Figure 5.6 Defining a calculation

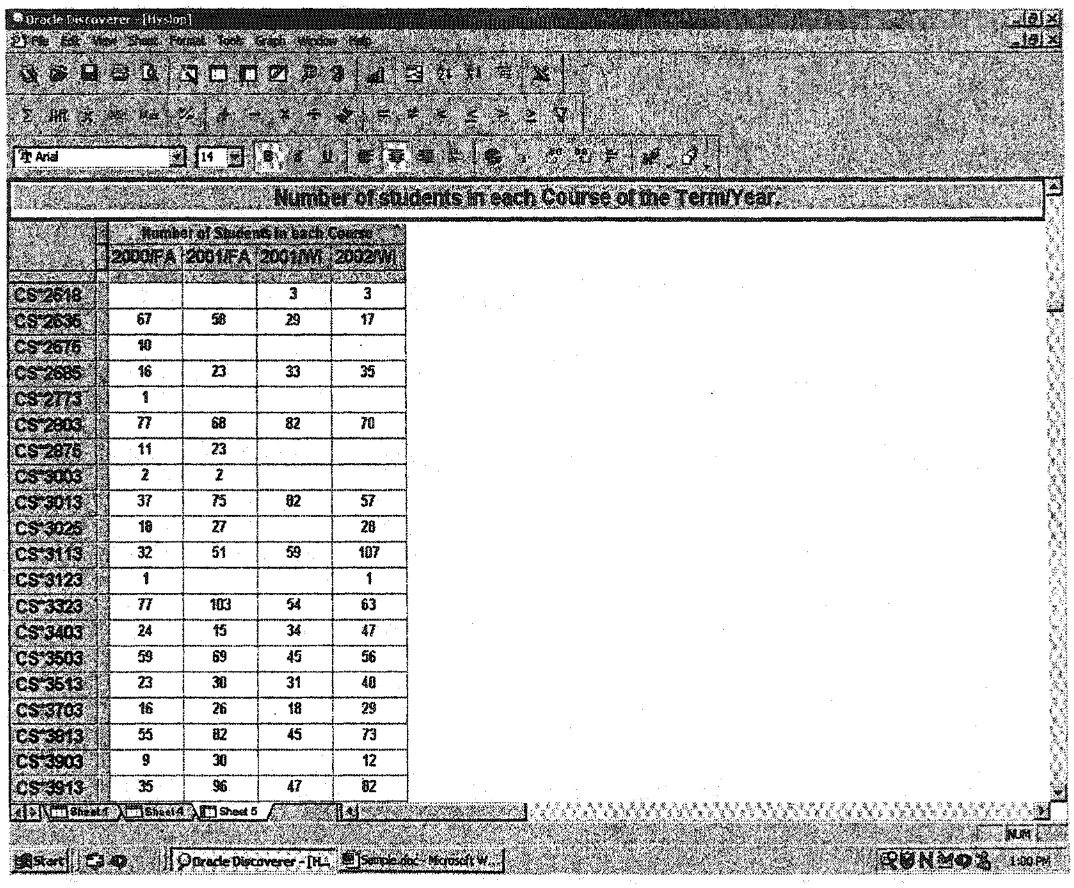
### 5.2 Sample Reports.

In this section some sample reports are presented that were produced using the data mart described in Chapter 4. The purpose of these reports was to show the utility of the data mart approach. Although these reports were based on user requirements, it was not the intent to design a full set of management reports. This set of sample reports can be used as a basis for generating a more complete set at a future date.

Figure 5.7 represents the first sample report. This report gives detailed information about the number of students taking each course over a number of terms. This report can be used by administrators to help in resource allocation to courses. According to this information they can decide how many sections need to be offered in a term, how many instructors are needed, and whether there is a necessity to offer the course in both terms or just one.

Figure 5.8 shows the same information as in Figure 5.7 but in different layout. In this report you can more clearly see the growth of demand on a given course. In this example we can see the history of CS1303. Figure 5.9 shows the same information in a graphical way. These three figures show the ease with which a user can get different views of the same data using Oracle Discoverer.

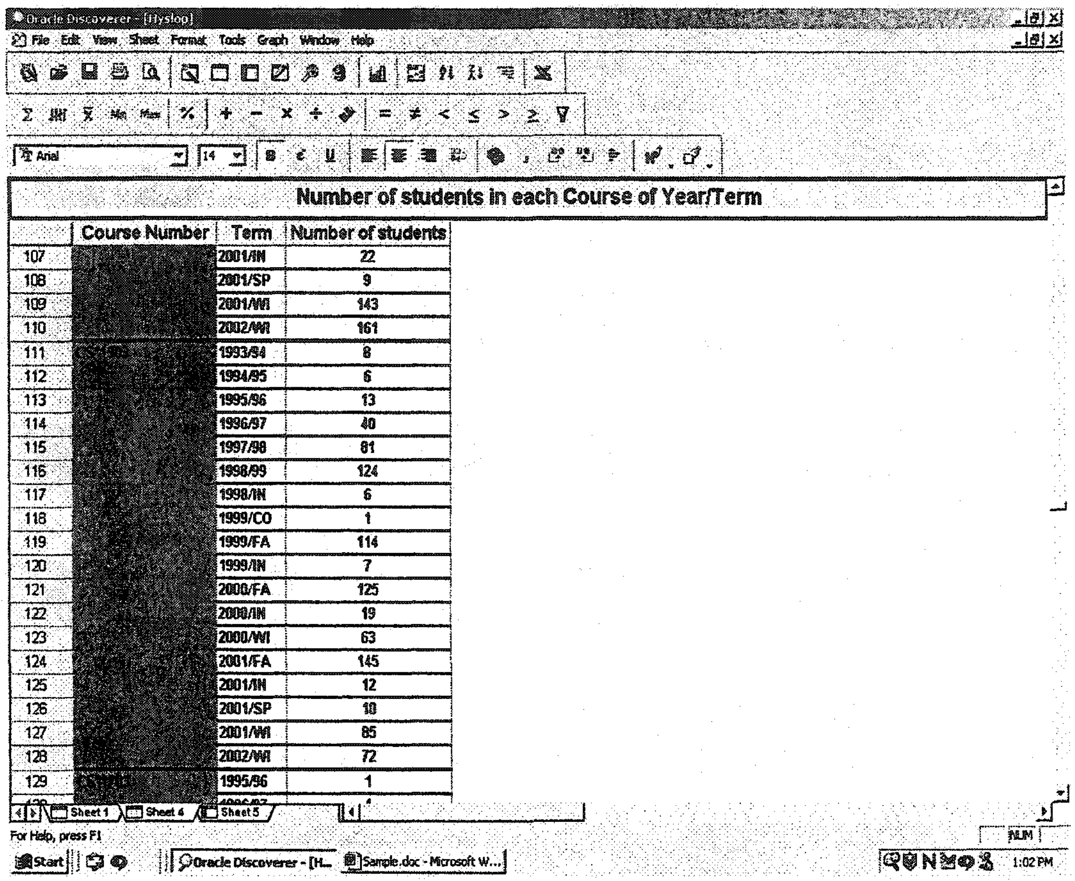
Based on user interviews it was determined that enrollment trends in certain courses were of most interest. Figure 5.10 shows a graph representing the enrollment trend for the courses: CS1073, CS1083, CS1303, CS 2803, and CS 3503 from 1995 to 2002.


Oracle Discoverer - [Hyslop]
File Edit View Sheet Format Tools Graph Window Help
Arial
14
Sheet 1
Sheet 4
Sheet 5
NUM
Start
Oracle Discoverer - [H...
Sample.doc - Microsoft W...
1:00 PM

Number of students in each Course of the Term/Year.

| | Number of Students in each Course | | | |
|---|---|---|---|---|
| | 2000/FA | 2001/FA | 2001/WI | 2002/WI |
| CS*2618 | | | 3 | 3 |
| CS*2636 | 67 | 58 | 29 | 17 |
| CS*2676 | 10 | | | |
| CS*2685 | 16 | 23 | 33 | 35 |
| CS*2773 | 1 | | | |
| CS*2803 | 77 | 68 | 82 | 70 |
| CS*2876 | 11 | 23 | | |
| CS*3003 | 2 | 2 | | |
| CS*3013 | 37 | 75 | 82 | 57 |
| CS*3025 | 18 | 27 | | 28 |
| CS*3113 | 32 | 51 | 59 | 107 |
| CS*3123 | 1 | | | 1 |
| CS*3323 | 77 | 103 | 54 | 63 |
| CS*3403 | 24 | 15 | 34 | 47 |
| CS*3503 | 59 | 69 | 45 | 56 |
| CS*3513 | 23 | 30 | 31 | 40 |
| CS*3703 | 16 | 26 | 18 | 29 |
| CS*3813 | 55 | 82 | 45 | 73 |
| CS*3903 | 9 | 30 | | 12 |
| CS*3913 | 35 | 96 | 47 | 82 |

**Figure 5.7 Course enrollment by term**

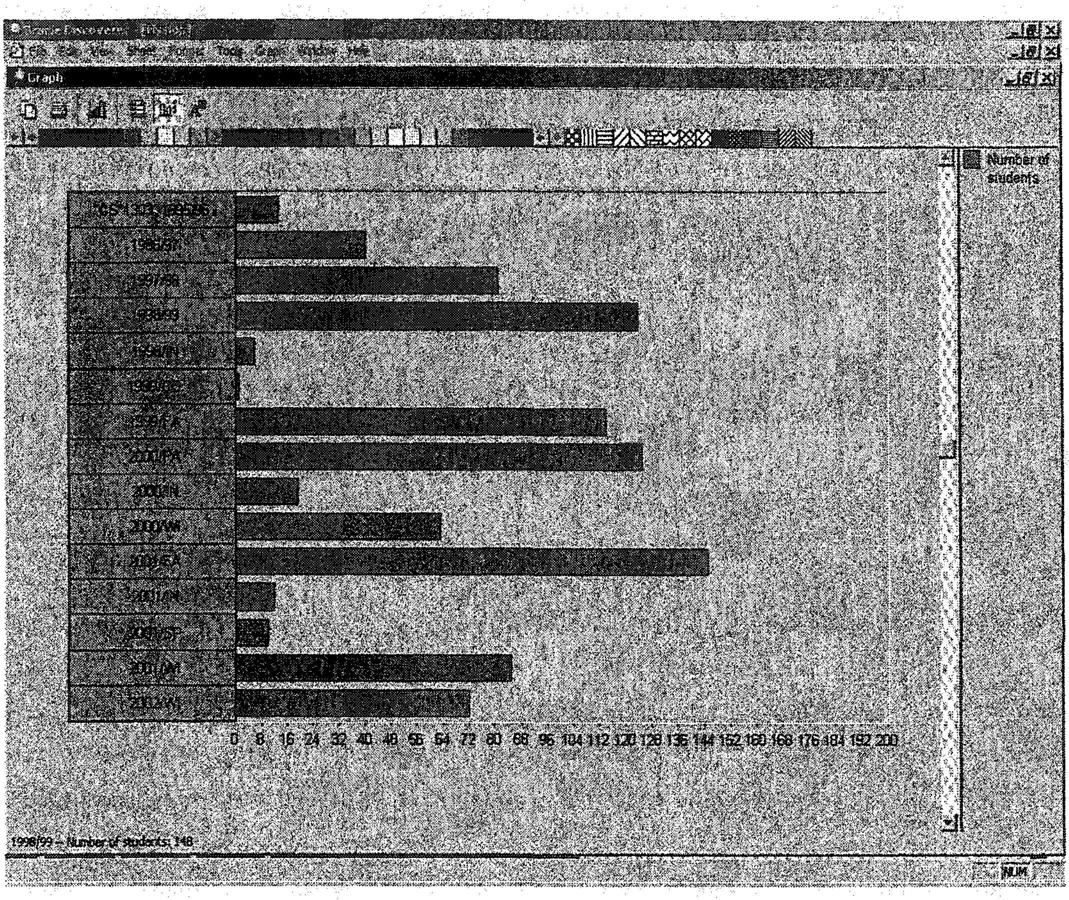


Oracle Discoverer - [Hyslop]
File Edit View Sheet Format Tools Graph Window Help
Arial
14
Sheet 1
Sheet 4
Sheet 5
For Help, press F1
NUM
Start
Oracle Discoverer - [H...
Sample.doc - Microsoft W...
1:02 PM

Number of students in each Course of Year/Term

|  | Course Number | Term | Number of students |
|---|---|---|---|
| 107 | [redacted] | 2001/IN | 22 |
| 108 | [redacted] | 2001/SP | 9 |
| 109 | [redacted] | 2001/WI | 143 |
| 110 | [redacted] | 2002/WI | 161 |
| 111 | [redacted] | 1993/94 | 8 |
| 112 | [redacted] | 1994/95 | 6 |
| 113 | [redacted] | 1995/96 | 13 |
| 114 | [redacted] | 1996/97 | 40 |
| 115 | [redacted] | 1997/98 | 81 |
| 116 | [redacted] | 1998/99 | 124 |
| 117 | [redacted] | 1998/IN | 6 |
| 118 | [redacted] | 1999/CO | 1 |
| 119 | [redacted] | 1999/FA | 114 |
| 120 | [redacted] | 1999/IN | 7 |
| 121 | [redacted] | 2000/FA | 125 |
| 122 | [redacted] | 2000/IN | 19 |
| 123 | [redacted] | 2000/WI | 63 |
| 124 | [redacted] | 2001/FA | 145 |
| 125 | [redacted] | 2001/IN | 12 |
| 126 | [redacted] | 2001/SP | 10 |
| 127 | [redacted] | 2001/WI | 85 |
| 128 | [redacted] | 2002/WI | 72 |
| 129 | [redacted] | 1995/96 | 1 |

**Figure 5.8 Course enrollment (different view)**



1998/99 – Number of students: 148

**Figure 5.9 Course enrollment (graphical view)**

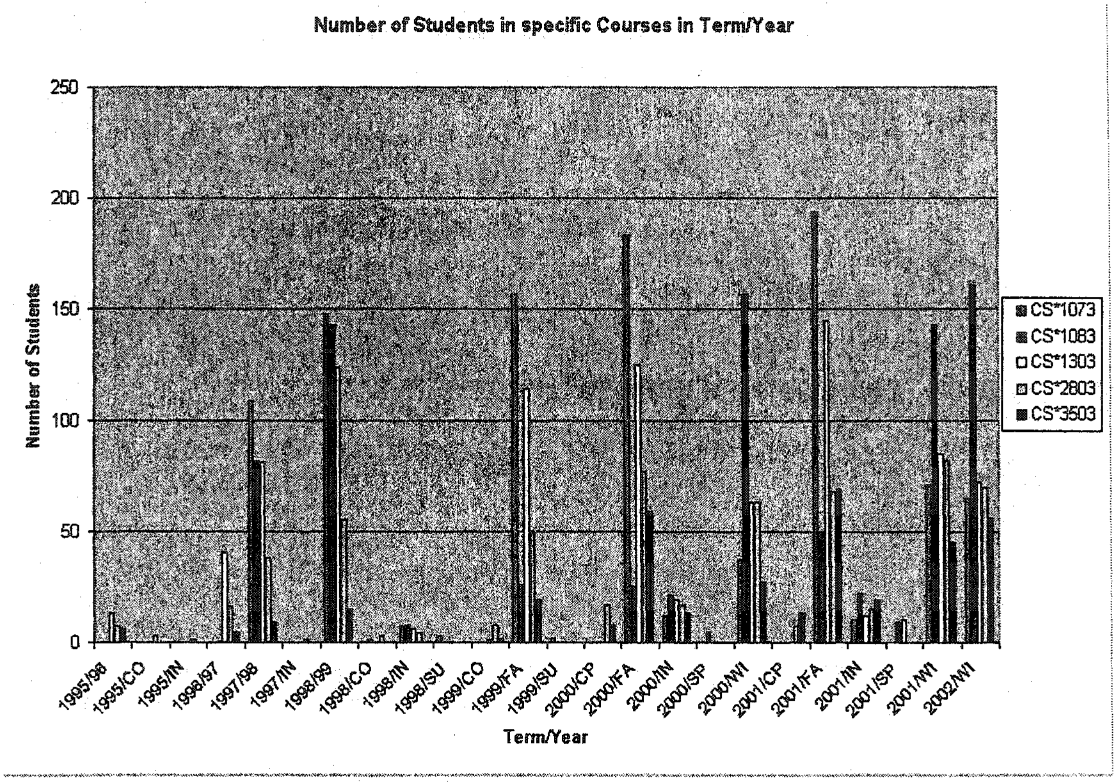
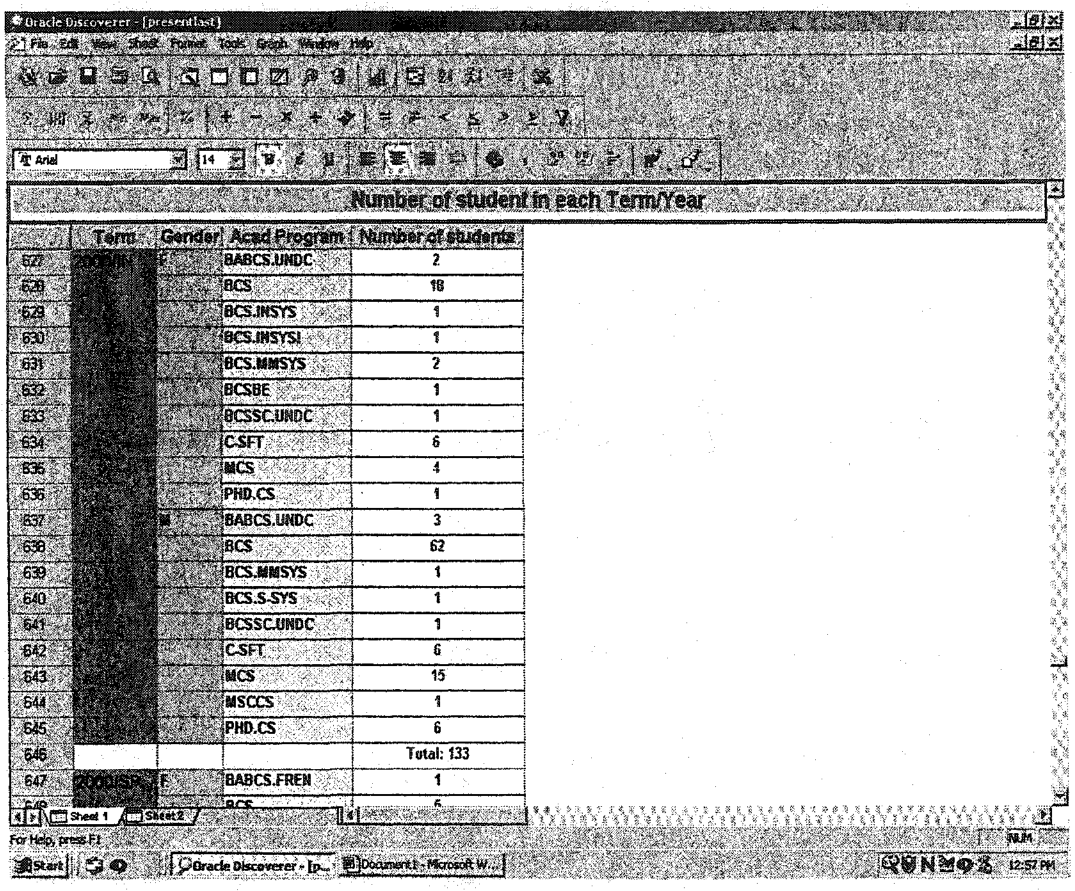


**Figure 5.10 Enrollment trends for specific courses**

Figure 5.11 shows the second sample report. This report provides us with the number of male and female students in each academic program for a specific term. This report is useful in showing demographic trends, which need to be reported to external agencies. Also it shows the trend of female enrollment which can be used to help in recruitment and retention activities. Figure 5.12 shows the same information in graphical form.


Oracle Discoverer - [presentlast]
File Edit View Sheet Format Tools Graph Window Help
Arial
14
Sheet 1
Sheet 2
For Help, press F1
NUM
Start
Oracle Discoverer - [p...
Document1 - Microsoft W...
12:57 PM

Number of student in each Term/Year

| | Term | Gender | Acad Program | Number of students |
|---|---|---|---|---|
| 627 | 2000WIN | F | BABCS.UNDC | 2 |
| 628 | | | BCS | 18 |
| 629 | | | BCS.INSYS | 1 |
| 630 | | | BCS.INSYSI | 1 |
| 631 | | | BCS.MMSYS | 2 |
| 632 | | | BCSBE | 1 |
| 633 | | | BCSSC.UNDC | 1 |
| 634 | | | C-SFT | 6 |
| 635 | | | MCS | 4 |
| 636 | | | PHD.CS | 1 |
| 637 | | M | BABCS.UNDC | 3 |
| 638 | | | BCS | 62 |
| 639 | | | BCS.MMSYS | 1 |
| 640 | | | BCS.S-SYS | 1 |
| 641 | | | BCSSC.UNDC | 1 |
| 642 | | | C-SFT | 6 |
| 643 | | | MCS | 15 |
| 644 | | | MSCCS | 1 |
| 645 | | | PHD.CS | 6 |
| 646 | | | | Total: 133 |
| 647 | 2000SP | F | BABCS.FREN | 1 |
| 648 | | | BCS | 5 |

**Figure 5.11 Enrollment by program and gender**

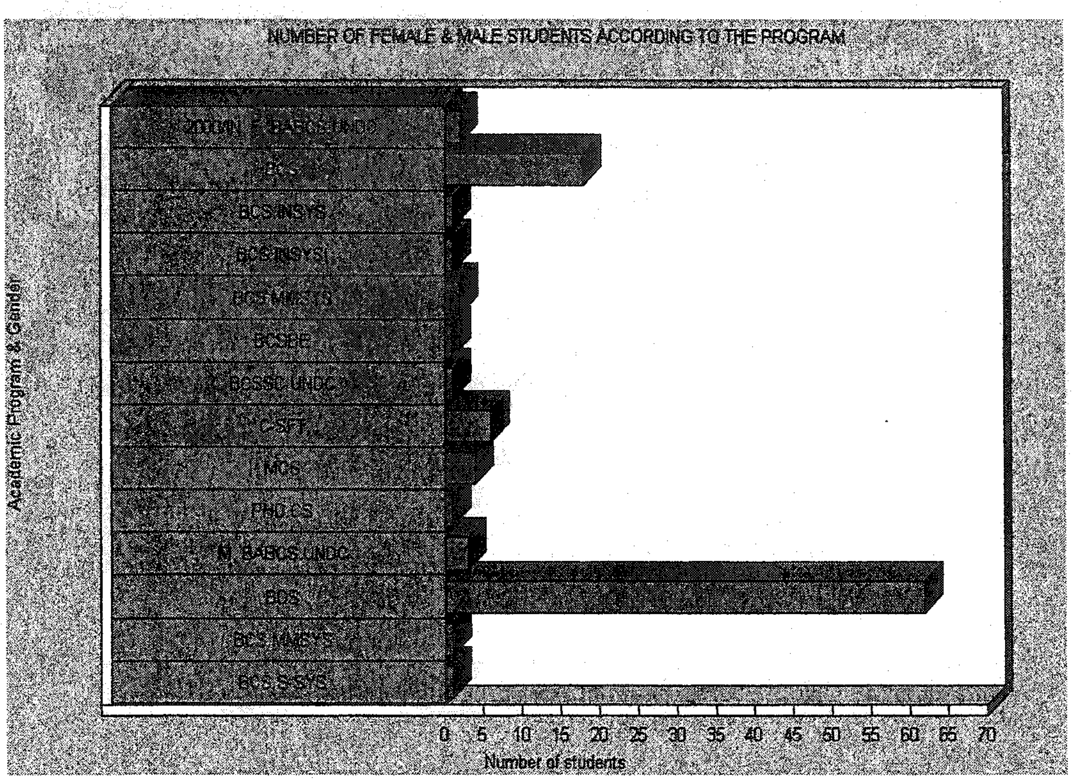
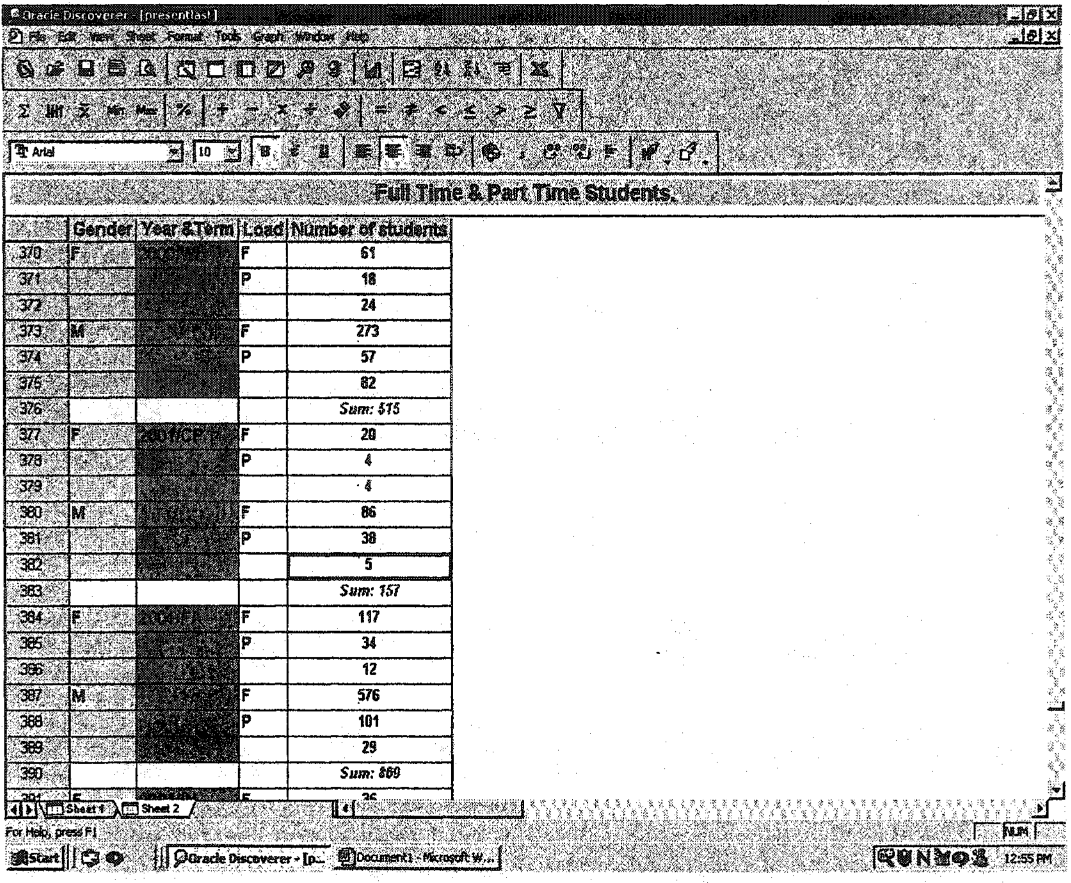


**Figure 5.12 Enrollment by program and gender (graphical)**

Figure 5.13 shows the third sample report. This report shows the number of full time and part time students in each academic program for a specific term. Through this report we will be able to identify the trend in full time and part time program. This report may suggest how our programs can be modified to better suit part-time students.


Oracle Discoverer - [presentlast]
File Edit View Sheet Format Tools Graph Window Help
Arial
10
Sheet 1
Sheet 2
For Help, press F1
NUM
Start
Oracle Discoverer - [p...
Document1 - Microsoft W...
12:55 PM

Full Time & Part Time Students.

| | Gender | Year &Term | Load | Number of students |
|---|---|---|---|---|
| 370 | F | [illegible] | F | 61 |
| 371 | | | P | 18 |
| 372 | | | | 24 |
| 373 | M | | F | 273 |
| 374 | | | P | 57 |
| 375 | | | | 82 |
| 376 | | | | *Sum: 515* |
| 377 | F | 2001/CP | F | 20 |
| 378 | | | P | 4 |
| 379 | | | | 4 |
| 380 | M | | F | 86 |
| 381 | | | P | 38 |
| 382 | | | | 5 |
| 383 | | | | *Sum: 157* |
| 384 | F | [illegible] | F | 117 |
| 385 | | | P | 34 |
| 386 | | | | 12 |
| 387 | M | | F | 576 |
| 388 | | | P | 101 |
| 389 | | | | 29 |
| 390 | | | | *Sum: 860* |
| 391 | F | | F | 36 |

**Figure 5.13 Enrollment by program and fee status**

The data used to generate these reports was based on an extraction from DataTel at a given point in time (January, 2002). Therefore the information appearing in the reports might not be 100% correct. This is because we have extracted the information before the final date for dropping courses. As well, we lack historical data, i.e. we only have currently active students and lack any students who have previously graduated. These problems can be solved by having a planned refresh interval for the data. This refresh interval would have to be determined based on university deadlines and the needs of the users of the data.

# Chapter 6

## Conclusions and Future Work

In this Chapter we will summarize the work accomplished and give some directions for future work. The main objective of this Master's project was to design a data mart for the Faculty of Computer Science at the University of New Brunswick. This objective has been accomplished. The faculty now has better source of information on which to base decision-making.

In developing this data mart a life cycle development methodology proposed by Kimball [9] has been used. Based on user requirements, we first extracted the required data from the DataTel source system. We then restructured the data from DataTel into a data mart with a new logical schema designed to make decision support easier. To show the utility of this new data mart, we also designed and developed some sample reporting applications using Oracle Discoverer.

Two logical data design approaches have been used in the project: Entity Relationship Modeling (ER) and Dimensional Modeling (DM). We have found that Entity Relationship Modeling is a more flexible approach for ad-hoc reporting and operational applications. Dimensional Modeling is more approprate for analytical data cubes in OLAP applications and for data mining. In this project, the ER mapped tables for the physical design has been used. This allows for more flexibility in performing ad-hoc reporting and for developing future operational applications (such as degree audit) that may be built using these tables. In the future, one could choose the dimensional modeling approach (and dimensional tables) if OLAP or data mining were more of interest.

The reports developed in this Masters project were primarily used to show the utility of the data mart approach. In the future, a wider set of reports should be designed. These reports may include reports for external agencies, as well as, reports to aid in recruitment and retention activities. As well, the data mart may serve as a basis for developing a new degree audit system.

For this project, the source data was limited in scope to the undergraduate student information contained in DataTel. In the future more student information can be added to the data mart. This may include information from the current co-op database, and the proposed graduate student information system. Also, the data mart could be expanded to include other sorts of information such as financial data useful for budgeting purposes.

Also, in the future the data mart can serve as the basis for performing multidimensional analysis using the Oracle OLAP tools. This would allow the examination of various trends and interdependencies in the data. For instance, it might allow the user to investigate how admission criteria affect the overall success of a student i.e. how high school marks determine GPA. As well, data mining studies could be done using the data mart to highlight any trends not obvious to the analyst.

## Bibliography

1. Barquin, Ramon. Building, Using, and Managing the Data Warehouse, Prentice Hall Publishing Company, New Jersey, 1997.
2. Bischoff, Joyce. Data Warehouse: Practical Advice from the Experts, Prentice Hall Publishing Company, New Jersey, 1997.
3. Brackett, Michael. The Data Warehouse Challenge, John Wiley & Sons, 1997.
4. Brobst, Steven. The Five Stages of an Active Data Warehouse Evolution, Teradata Magazine Online, http://www.teradatamagazine.com/articles_01/five_stages.htm.
5. Devlin, Barry. Data Warehouse from Architecture to Implementation, Addison-Wesley, 1994.
6. Dodge, Gary. Essential Oracle8i Data Warehousing, John Wiley & Sons, 2000.
7. Inmon, William. Using the Data Warehouse, John Wiley & Sons, 1994.
8. Inmon, William. Building the Data Warehouse Second Edition, John Wiley & Sons, New York, 1996.
9. Kimball, Ralph. The Data Warehouse Lifecycle Toolkit, John Wiley & Sons, New York, 1998.
10. Silberschatz, Avi. Database System Concepts 4th Edition, McGraw Hill, 2002.
11. Unknown, http://www.dwinfocenter.org/
12. Unknown, http://www.datawarehouseevolution.com/

# Appendix B.

## Physical Table Descriptions

```
create table student (
 Student_ID    NUMBER(9),
 FNAME         VARCHAR2(10),
 Lname         VARCHAR2(10),
 Middle_Name      VARCHAR2(10),
 Major         VARCHAR2(10),
 Minor         VARCHAR2(10),
 Birth_Date VARCHAR2(10),
 Marital_Status VARCHAR2(10),
 Gender         VARCHAR2(10),
 phone            NUMBER(9),
 Email            VARCHAR2(20),
 Country        VARCHAR2(10),
 C_city         VARCHAR2(15),
 C_phone         VARCHAR2(15),
 H_city         VARCHAR2(15),
 H_phone        VARCHAR2(15),
 Home_Prov  VARCHAR2(10),
 Home_Postal      VARCHAR2(10),
 Local_Address    VARCHAR2(10),
 Local_City       VARCHAR2(10),
 Local_Postal     VARCHAR2(10),
 Acad_Level VARCHAR2(10),
 Acad_Program     VARCHAR2(10),
 Campus           VARCHAR2(10),
 Start_Date       VARCHAR2(10),
 Ant_Completion   VARCHAR2(10),
 Cum_GPA    NUMBER(5,2),
 Ass_GPA    NUMBER(5,2),
 Cum_Credits      NUMBER(32),
 Load       VARCHAR2(3),
constraint student_pk primary key (Student_ID));

create table course (
 course_No    VARCHAR2(10),
 Title        VARCHAR2(15),
 CR_HOurse    NUMBER(4),
 Code   VARCHAR2(10),
 Dept         VARCHAR2(15),
constraint course_pk primary key (course_No));

create table section (
 section_No  NUMBER(2),
 c_No   VARCHAR2(10),
 Year     DATE,
 campus     VARCHAR2(10),
 constraint section_pk primary key (section_No,c_No),
 constraint section_fk1 foreign key (c_No) references
course(course_No));
```

```
create table highschool(
 course      VARCHAR2(10),
 mark        NUMBER(3),
 studentid   NUMBER(9),
 constraint highschool_pk primary key (studentid,course),
 constraint highschool_fk1 foreign key (studentid) references
student(Student_ID) );

create table transfer (
 Institute      VARCHAR2(15),
 c_No   VARCHAR2(10),
 Stt_Id      NUMBER(9),
 Degree           VARCHAR2(10),
 Province   VARCHAR2(15),
 Country    VARCHAR2(15),
constraint trans_pk primary key (c_No,Stt_Id),
constraint trans_fk1 foreign key (c_No) references course(course_No),
constraint trans_fk2 foreign key (Stt_Id) references
student(Student_ID) );

create table take (
 St_Id      NUMBER(9),
 courseNo    VARCHAR2(10),
 sec        NUMBER(2),
 grade   VARCHAR2(3),
 Year NUMBER(4),
 Term VARCHAR2(10),
constraint t_pk primary key (St_Id,sec,courseNo),
constraint t_fk1 foreign key (St_Id) references student(Student_ID),
constraint t_fk2 foreign key (courseNo) references course(course_No),
constraint t_fk3 foreign key (grade) references GPA(Grade));

create table prcourse (
 course      VARCHAR2(10),
 prereq      VARCHAR2(10),
 cdate       DATE,
constraint p_pk primary key (course,prereq),
constraint p_fk1 foreign key (course) references course(course_No),
constraint p_fk2 foreign key (prereq) references course(course_No) );

create table GPA (
 Grade      VARCHAR2(3),
 Point      NUMBER(5,2),
 constraint GPA_pk primary key (Grade));
```

# VITA

Candidate's full name:
Feras Naser Obeidat

Place and date of Birth
Tripoli, Libya, April 7, 1970

Educational Institutions Attended:

University Science Malaysia, Pinang, 1997-1998
Yarmouk University , Jordan. 1988-1992

Degrees Awarded:

M.Sc in Information Technology, University Science Malaysia, 1997-1998
B.Sc in Physics, Yarmouk University , Jordan. 1988-1992