FACULTY OF COMPUTING & INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY

FCIT
KAU

كلية الحاسبات وتقنية المعلومات
جامعة الملك عبدالعزيز

Database Administration (CPIT345)

College Database

Student name	Student ID
Moayad Aamri	2035765
Abdulaziz Alfozan	2036008
Fahad Alsifri	1743998
Mohammad Banjar	2037025

Instructor: Dr. Marwan Alsahafi

Table of contents

Introduction

We designed a special database for a college system considering the characteristics and stages to build an ideal database environment for the college.

The database contains information related to several entities such as students, departments, sections, and courses taught within these sections, instructors with courses taken, and other relationships between database tables.

The report will also discuss several topics such as database environment and what type of DBMS will be use, database design (er-diagram & relational-schema), metadata, performance, availability, security (privileges and roles will be given to certain users), backup and recovery plan, disaster plan, and the storage needed for the database.

Database environment

A relational database with high maintainability and services will be needed to establish a college database, which will be a sizable database.

We will use Oracle SQLplus 11g enterprise edition, which is supported by the windows 10 operating system used in the organization.

This database offers outstanding performance, the ability to scale the database in case the storage becomes almost full, and the capacity to accommodate the enormous amount of data required for a college.

Given that Oracle is one of the most well-known DBMSs and that SQLplus 11g includes a large range of tools for creating and administering databases, there are many experts who are proficient in using and managing Oracle DBMSs.

Data and storage and management

The storage space required for our database is calculated as follow: assuming that page size is 2048 bytes for all tables and page header is 48 bytes for all tables.

- **Department Table**

DName = 30 Bytes, DCode = 6 Bytes, DOffice = 6 Bytes, DPhone = 15 Bytes

Bytes Row size = 30 + 6 + 6 + 15 = 57

Bytes Rows Per Page = 2048 – 48 = 2000 / 57 = 35

Table size = (300 / 35) * 2048 = 17554Bytes

- **Instructor Table**

Id= 15 Bytes, IPhone = 15 Bytes, Fname = 12 Bytes, Lname = 12 Bytes, IOffice = 6 Bytes,

Rank = 20 Bytes

Bytes Row size = 15 + 15 + 12 + 12 + 6 + 20 = 80

Bytes Rows Per Page = 2048 – 48 = 2000 / 80 = 25

Table size = (250 / 25) * 2048 = 20480 Bytes

- **Student Table**

S_ID = 15 Bytes, Fname = 12 Bytes, Lname = 12 Bytes, Phone= 15 Bytes, Address= 15 Bytes,

Dob = 10 Bytes

Bytes Row size = 15 + 12 + 12 + 15 + 15 + 10 = 79

Bytes Rows Per Page = 2048 – 48 = 2000 / 79 = 25

Table size = (400 / 25) * 2048 = 32768 Bytes

- **Course Table**

CoName = 50 Bytes, CCode = 7 Bytes, CLevel = 15 Bytes

Bytes Row size = 50 + 7 + 15 = 72

Bytes Rows Per Page = 2048 – 48 = 2000 / 72 = 28

Table size = (1000 / 28) * 2048 = 73143 Bytes

- **Section Table**

SecID = 15 Bytes, SecNo = 3 Bytes, CNumber = 15 Bytes, DaysTime = 3 Bytes

Bytes Row size = 15 + 3 + 15 + 3 = 36

Bytes Rows Per Page = 2048 – 48 = 2000 / 36 = 56

Table size = (400000 / 56) * 2048 = 14628571 Bytes

- **Takes Table**

S_ID = 15 Bytes, Sec_ID = 15 Bytes, Grade = 15 Bytes

Bytes Row size = 15 + 15 + 15 = 45

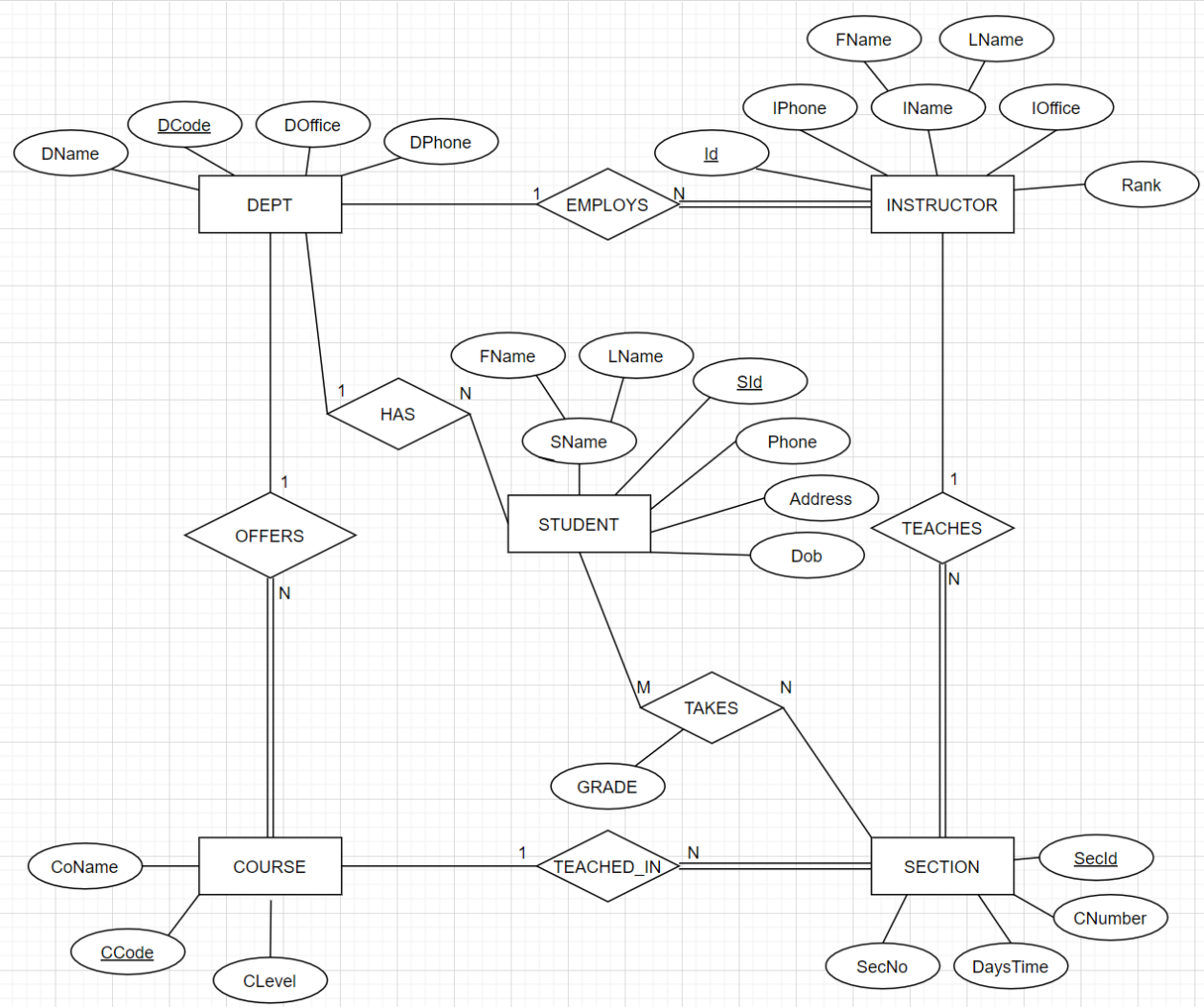
Bytes Rows Per Page = 2048 – 48 = 2000 / 45 = 44

Table size = (250000 / 44) * 2048 = 11636364Bytes

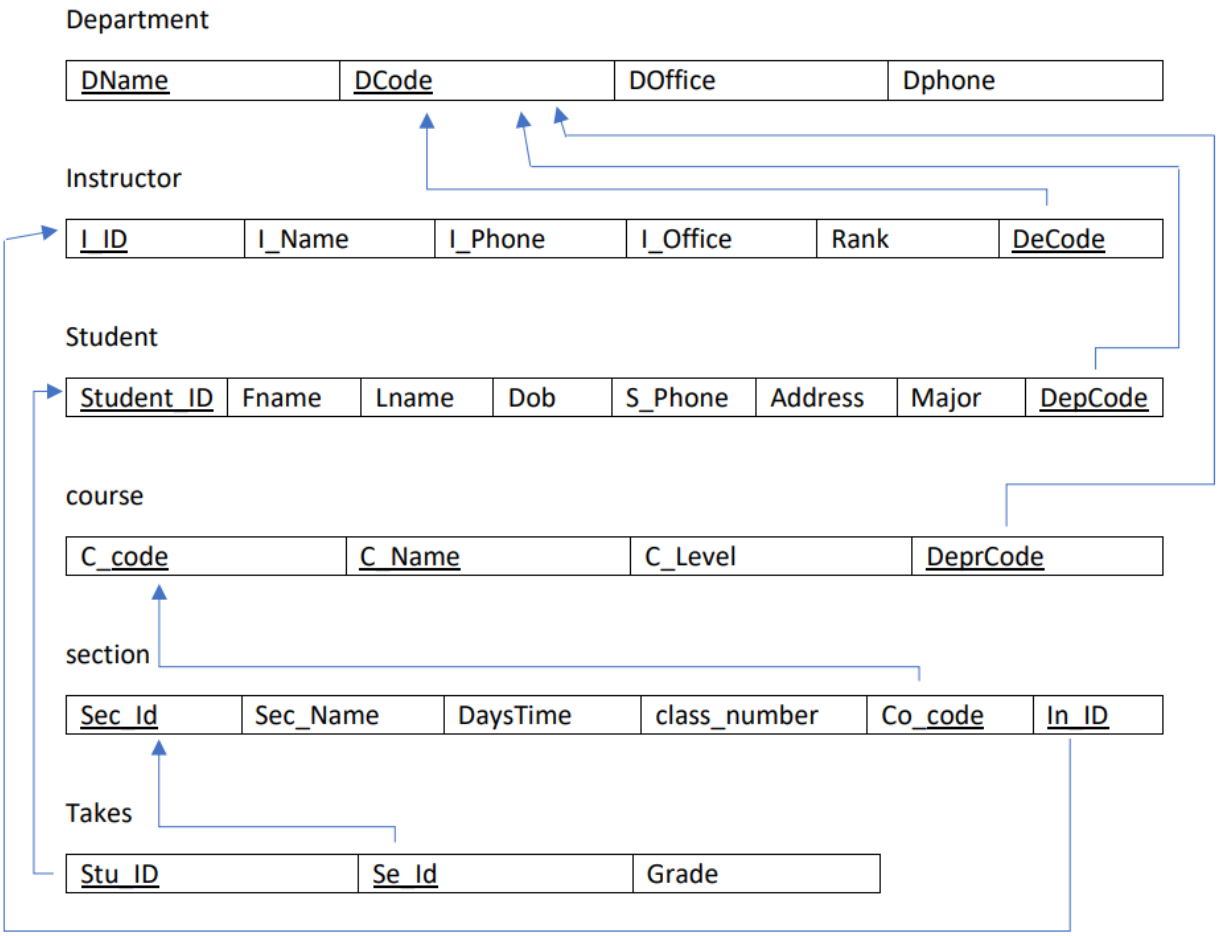
All the table sizes accumulate to 26408880 Bytes which is equal to 26 MegaBytes which is the estimated size of only one copy of the database.

Database Design

ER-Diagram



Relational Schema



Description

Department: code (DCode) is a primary key, name (DName), department office (DOffice), phone number (DPhone).

Instructor: (I_Id) is a primary key, instructor name (I_FName, I_LName), phone number (IPhone), and Office (IOffice), Instructor Rank (Rank), Department code (DeCode) is a foreign key from Department.

Student: (Student_ID) is a primary key, name (FName, LName), birth date (Dob), phone number (Phone), address (Address), Major (Major), Department code (DepCode) is a foreign key from Department.

Course: Course code (CCode) is a primary key, Course name (CName), level (CLevel), Department code (DeprCode) is a foreign key from Department.

Section: Section ID (Sec_ID) is a primary key, name (Sec_Name), daytime (daysTime), class number (Class_Number), Course code (Co_code) is a foreign key from Course, Instructor ID (In_ID) Is a foreign key from Instructor.

Takes: Student Id (Stu_ID) is a foreign key from Student, Section Id (Sec_ID) is a foreign key from Section, Grade (Grade).

Relationships

Offers: a relationship between department and course, the department can offer many courses, and the course can only be in one department.

Employs: a relationship between department and instructor, a department can have many instructors, and an instructor can only have one department.

Has: a relationship between department and students, a department can have many students, and the student only has one department.

Teaches: a relationship between instructor and section, instructors can teach many sections, and a section can only have one instructor.

Takes: a relationship between student and section with one attribute: Grade, a student can have many sections, and a section can contain many students.

Teached_In: a relationship between course and section, courses can have many sections, and every section belong to one course only.

Metadata

Technology metadata Business metadata

Department

Field Name	Description	Type
DCode(PK)	Department code	Varchar2(30)
DName	Department name	Varchar2(6)
DOffice	Department office	Varchar2(6)
DPhone	Phone number	Integer

Instructor

Field Name	Description	Type
I_ID	Instructor ID	Integer
I_FName	Instructor first name	Varchar2(12)
I_LName	Instructor last name	Varchar2(12)
I_Phone	Instructor phone number	Integer
IOffice	Instructor office	Varchar2(6)
Rank	Instructor rank	Varchar2(20)
DeCode(FK)	Department code in which the instructor works in	Varchar2(6)

Student

Field Name	Description	Type
Student_ID(PK)	Student ID	Integer
Fname	Student first name	Varchar2(12)
Lname	Student last name	Varchar2(12)
Dob	Student date of birth	Date
Phone	Student phone number	Integer
Address	Student address	Varchar2(15)
DepCode(FK)	Department code in which the student studies	Varchar2(6)

Course

Field Name	Description	Type
CCode(PK)	Course code	Varchar2(7)
CName	Course name	Varchar2(50)
CLevel	Course level	Integer
DeprCode(FK)	Department code in which the course belongs	Varchar2(6)

Section

Field Name	Description	Type
Sec_ID(PK)	Section ID	Integer
Sec_Name	Section name	Varchar2(3)
DaysTime	Section lecture days	Varchar2(3)
Class_Number	Class number	Integer
Co_Code(FK)	Course code for the section	Varchar2(7)
In_ID(FK)	Instructor id who teaches the section	Integer

Takes

Field Name	Description	Type
Stu_ID(FK)	Student id who is studying in this section	Integer
Sec_ID(FK)	Section number	Integer
Grade	Student grade in this section	Integer

Database connectivity

The database will be stored in a distributed client/server architecture, specifically multi-level architecture.
In this architecture, separate two or more servers will house both the database and the application. the client will handle all the presentation logic. Therefore, it is regarded as a thin client because the client side handles the presentation while the server side handles the logic. Because the data is not stored on a single server, the distributed architecture will be able to fulfill the database's requirements for high availability and scalability while also minimizing data loss in the event of hardware failure.
To make it easier for clients to access data on the database server, We had to create a website to allow customers to use the functions and access the data.

Database security

There will be 4 users in the database with different roles:

DBA: the DBA will have full access to create, insert, update, or delete in the database.

Department Manager: the department manager will have the access to retrieve, insert, update, and delete from instructors, students, courses, sections, and grades but only if it belongs to the department.

Instructor: the instructor will have the access to retrieve, insert, update, and delete from the grades and only retrieve from instructor, student, course, and section.

Student: the student will have the ability to retrieve and insert to takes (his courses) and only retrieve from section.

Backup and Recovery plan

We will maintain backups for 25 years to ensure that data is secure and recoverable. The plan is to execute a full back up on both the database objects and the log file on Fridays and incremental backups once per day from Sunday to Thursday.

Two types of recovery will be used in our database:
- Recover to Current
- Off-site disaster Recovery

In the event of a hardware malfunction, the **recover to current** will be used. In the event of a natural disaster (earthquake, fire, etc...) or other accident, where the primary location is inaccessible, the **off-site disaster recovery** will be used.

1 copy = 26MB, having 1 full back up a week means that a 104MB is needed a month, which means 1248MB a year, which equals to 31200MB = 31.2GB for 25 years. Having 6 incremental backups a week, where each backup will equal approximately 5KB, weekly it will equal 30KB, monthly it equals 120KB, and yearly it equals 1440KB, which equals to 36000 KB = 36MB, 31.2GB + 36MB = 32GB approximately.
A single log file for our database is approximately 78MB, which equals 312MB a month, 3744MB a year, and 93600MB = 0.0936TB needed for 25 years' worth of log file, which makes the total amount of storage needed for our database = 0.0936TB + 32GB = 0.1256TB.

Disaster recovery planning

Disasters and crises can occur at any time, and they frequently happen without notice. So, taking in advantage the team set the stage and direction for the plan to determining the amount of preparedness in the recovery team to reduce data loss, speed up service recovery, limits of disruption and when an incident becomes an emergency. The Disaster recovery team has tested the recovery plan as a crucial part, which they do twice a year in the beginning of February and August. Its an important to update the Disaster recovery plan which should be done before each testing. The strategy that used for disaster recovery backup is a storage management software. Such an approach greatly simplifies disaster recovery preparation and execution, but this strategy may require a significant system outage to accomplish correctly.

Performance management

The rise in workload, which is a collection of online transactions and commands directed via the system at any moment, is one of many elements that affects our database performance.

We employed the subsequent techniques to combat it:

- Reducing system downtime by doing routine maintenance while they are still active.
- Offer database servers with the quickest network connections for users.

Database performance was also affected by optimization, which is the creating efficient data access routes through the process of evaluating database requests.

Data movement and distribution

Nothing can be more terrifying than losing important data because your system has suddenly crashed. So our team tends to use replication method of data movement as it allows you to continue working by switching to a replica of your data to prevent critical data loss. Replication in short is copying entire tables to multiple locations. We chose this method because of The advantages of data replication are accessibility to several hosts or data centers and simplification of data sharing between systems on a large scale by dividing the network load between heterogeneous systems.

Our college system can expect to experience the following advantages from implementing data replication services:

- Data Reliability and Availability
- Disaster Recovery
- Server Performance
- Better Network Performance
- Data Analytics Support
- Enhanced Test System Performance

Due to these advantages, we found that it is an suitable method for the strategy and the plan of our system, as we can afford the cost of this strategy to prevent data loss or system crashes.