COLLEGE DATABASE MANAGEMENT SYSTEM

Report -Database Lab (DSE 2260)
Department of Data Science & Computer Applications

B. Tech Data Science
4th Semester – Batch: B1

Submitted By

Ishita Saha	200968082
Balaji Kartheek	200968080
Maley Bhuvan Chandra	200968096
NA	NA

Mentored By

Vinayak M
Assistant Professor-Senior
DSCA, MIT

Archana H/ Shameem
Assistant Professor-Senior
DSCA, MIT

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Date:

CERTIFICATE

This is to certify that the < Ishita Saha(200968082),Balaji Kartheek(200968080), Maley Bhuvan Chandra(200968096) have successfully executed a mini project titled “College Database Management System” rightly brining fore the competencies and skill sets they have gained during the course- Database Lab (DSE 2262 & DSE), thereby resulting in the culmination of this project.

Vinayak M	**Archana H / Shameem**
Assistant Professor-Senior	**Assistant Professor-Senior**
DSCA, MIT	**DSCA, MIT**

COLLEGE DATABASE MANAGEMENT SYSTEM

1. Introduction:

The system is an Internet-based application that aims at providing information to all the levels of the management within an organization. For a given student/staff (Technical /Non-technical) the Administrator creates a login id & password for each of them, By this, they can access the system to either upload or download some information from the database

2.1 Proposed System

Abstract:

In the current college administrative system, all sorts of activities are performed manually. They maintain all sorts of student and college information recorded manually by using paper. The existence of this type of system is highly involving a lot of paperwork and calculation and therefore may lose the data, Due to its manual nature, it is difficult to update, delete, add or view the data whenever required. Manual data management is expensive and time-consuming any human error leads to inaccurate information.

The College database management system makes the whole data in an organized manner that reduces the manual work of persons in the admin panel and the bundle of the registers that were searched when to find the information of a previous student. The database of the system will help us to check the particular details using his/her student id. Integrity constraints are important when inserting new data fields.

The database system will reduce the load on the college administrative staff. The machine performs all calculations, hence chances of error are nil. The student, staff, and department information can easily be retrieved and any additional constraint checking can be applied. Redundancy and inconsistency of the data are managed as this is useful in not storing various copies of the same data in the database.

This system helps educational institutions by automating regular administrative tasks. Awareness and the right information about any college are essential for both the development of the student as well as the faculty. So this serves the right purpose in achieving the desired requirements of both the communities.

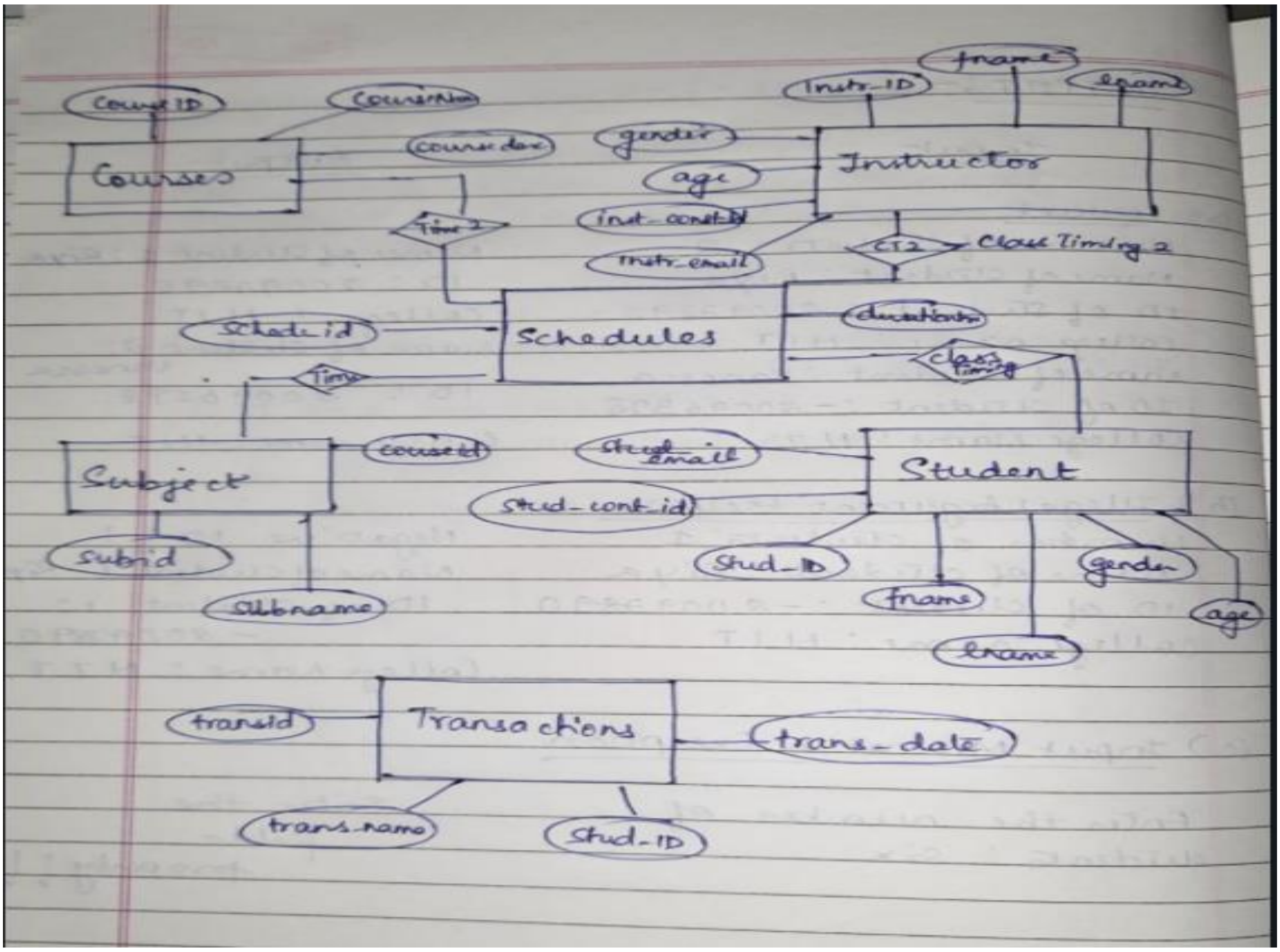
2.2 OBJECTIVES:

- **The main goal is to keep track of the details of the institution, staff, program, and assignments.**
- **The system provides for user-ID validation, hence unauthorized access is prevented.**
- **To manage Concurrent access by multiple users, it is needed for high performance in accessing the data.**
- **Compare and analyze the information in different departments…**
- **To retrieve the details of the student to whomever we required**
- **To remove the redundancy and inconsistency in the data.**

3. Functional Requirements

- The functions of a DBMS include concurrency, security, backup and recovery, and data descriptions.
- Different versions of the same data are handled by this system (Redundancy problem).
- This system allows for concurrent access by multiple users, each with their specific role may be some contribute to adding new data, while others design and manage the database these all things are handled at the same time.

4.2 ER Model Diagram:



4.2 Schema Diagram

Student(stud_ID, fname, lname, gender, age, stud_cont_id, stud_email)

Transacion->stud_ID refers to Student Relation

Transaction(trans_ID(PK), trans_name, stud_ID, trans_date)

Courses(course_ID, course_name, course_desc**)**

Students->course_ID refers to Courses

Subjects(sub_ID,sub_name,course_ID**)**

Instructor(Instr_ID, fname, lname, gender, age, Ins_cont_id,Instr_email**)**

Schedules->course_ID refers to Courses

Schedules->sub_ID refers to Subjects

Schedules->Instr_ID refers to Instructor

Schedules->stud_ID refers to Student

Schedules(sched_ID, course_ID, sub_ID, instr_ID, stud_ID, duration_hr**)**

4.3 Data Dictionary:

Student Table:

Column Name	Datatype	Constraint	Constraint Name
Stud_id	Number(10)	Primary Key	
fname	Varchar2(20)		
Lname	Varchar2(20)		
Gender	Char(1)		
Age	Number(3)		
Stud_cont_id	Number(10)		
Stud_email	Varchar(30)		

Transaction Table:

Column Name	Datatype	Constraint	Constraint Name
Trans_id	Number(10)	Primary Key	

Trans_name	Varchar2(20)		
Stud_id	Number(10)	Foreign Key referencing Student	
Trans_data	Varchar2(10)		

Courses Table:

Column Name	**Datatype**	**Constraint**	**Constraint Name**
Course_id	Number(10)	Primary Key	
Course_name	Varchar2(20)		
Course_desc	Varchar(30)		

Subjects Table:

Column Name	**Datatype**	**Constraint**	**Constraint Name**
Sub_id	Number(10)	Primary Key	
Sub_name	Varchar2(20)		
Course_id	Number(10)	Foreign Key referencing courses	

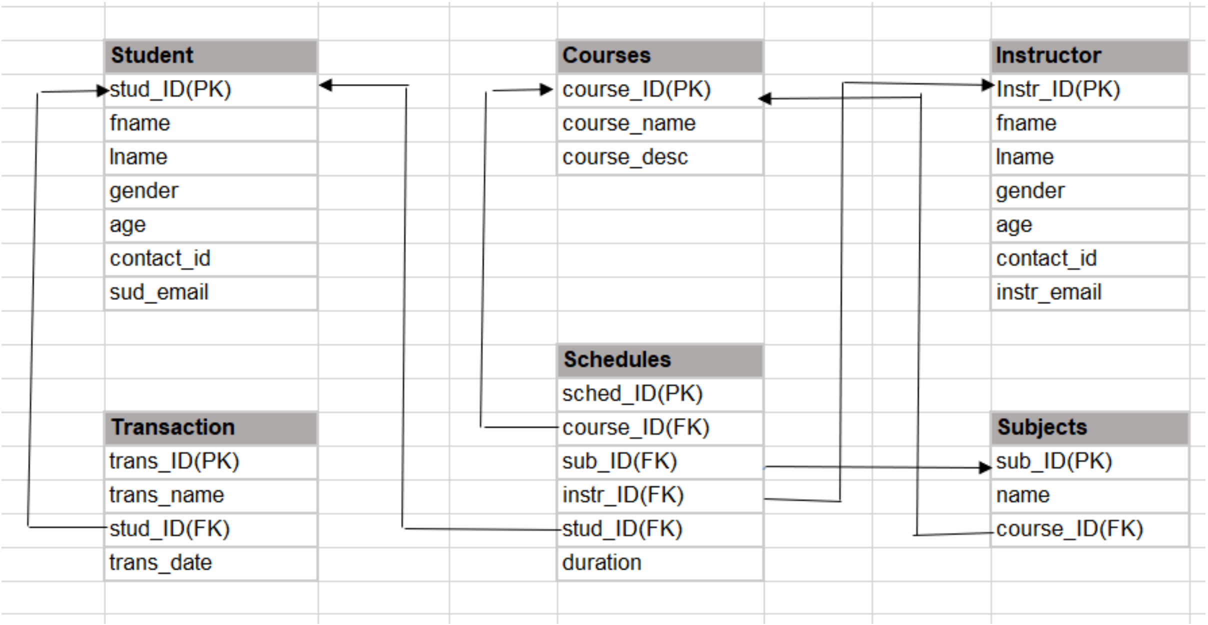
Instructor Table:

Column Name	**Datatype**	**Constraint**	**Constraint Name**
Instr_id	Number(10)	Primary Key	
Fname	Varchar2(20)		
Lname	Varchar2(20)		
Gender	Char(1)		
Age	Number(3)		
Ins_const_id	Number(10)		
Instr_email	Varchar2(20)		

Schedules Table:

Column Name	Datatype	Constraint	Constraint Name
Sched_ID	Number(10)	Primary Key	
Course_id	Number(10)	Foreign Key referencing Courses	
Sub_id	Number(10)	Foreign Key referencing subjects	
Instr_id	Number(10)	Foreign Key referencing Instructor	
Stud_id	Number(10)	Foreign key referencing students	
Duration_hr	Number(3)		

ER-MODEL:



Student
stud_ID(PK)
fname
lname
gender
age
contact_id
sud_email

Courses
course_ID(PK)
course_name
course_desc

Instructor
Instr_ID(PK)
fname
lname
gender
age
contact_id
instr_email

Transaction
trans_ID(PK)
trans_name
stud_ID(FK)
trans_date

Schedules
sched_ID(PK)
course_ID(FK)
sub_ID(FK)
instr_ID(FK)
stud_ID(FK)
duration

Subjects
sub_ID(PK)
name
course_ID(FK)

4.4 Relational Model Implementation

```
create table Student(
   stud_ID number(10) primary key,
   fname varchar2(10),
   lname varchar2(10),
   gender char(1),
   age number(3),
   stud_cont_id number(10),
   stud_email varchar(30));
create table Transaction(
   trans_id number(10),
   trans_name varchar2(20),
   trans_date varchar(10),
   stud_id number(10) references Student);



create table Courses(
   course_id number(10) primary key,
   course_name varchar2(20),
   course_desc varchar(30));


create table Subjects(
   sub_id number(10) primary key,
   sub_name varchar2(20),
   course_id number(10) references Courses);
```

```
create table Instructor(
   instr_id number(10) primary key,
   fname varchar2(20),
   lname varchar2(20),
   gender char(1),
   age number(3),
   instr_cont_id number(10),
   instr_email varchar2(20));
create table Schedules(
   sched_id number(10) primary key,
   course_id references Courses,
   sub_id references Subjects,
   instr_id references Instructor,
   stud_id references Student,
   duration_hr number(3));
```