كلية الهندسة

جامعة عين شمس

Database Project

Title : University Database

Submitted to : Dr.Hoda Korashy

Submitted by :

Mohamed Elsayed Mohamed 1601129

Mohamed Ahmed Khalil 1601104

Mohamed Ahmed Hassan 1601100

Mohamed Osama El-Shafie 1501147

Mohamed Ashraf Mohamed 1501156

Introduction:

-each college in the university must has many areas of study (Department), each college must put course schedule, and course schedule must schedule many courses from course catalogue.

-course schedule must contain course daily schedule and display the class room for the lectures which exist in one of the buildings.

- People in college may classified into students or employees, each Student must study in one of this area of studies, each student must make course enrollment to enroll of one of courses in course catalog and each student achieves grades in each course.

- regarding employee, each employee owns his job information.

Important data and reports:

College: CollegeID(PK),CollegeName,CourseScheduleID

Area Of Study: AreaOfStudyID(PK),CollegeID,StudyTittle

CourseCatalog: CourseCode(PK),CourseNo(PK),CourseName

CourseSchedule: CourseScheduleID(PK),Location,CourseCode,CourseNo

CourseDailySchedule: DailyID(PK),StartTime,EndTime,CourseCode,CourseNo

ClassRoom: ClassRoomID(PK),RoomNumber,BuildingNo

Building: BuildingID(PK),BuildingName

People:PersonID(PK),FirstName,LastName,Address,DOB

Student:AreaOfStudyID,PersonID

Employee:PersonID,Salary,JobID

Grade:GradeID(PK),SubjectName(PK),PersonID,Grade

CourseEnrollment:EnrollmentID

JobInformation:JobID(PK),JobDesc,Salary

Important data and reports:

Reports:

1-Select Students which has AreaofStudyID = 1

2- Select first name, last name of employee which has salary over 123.

3- for each employee which has address='aseedf1', select number of employees which live in this address.

4- select first name of students whose study tittle is ‘B’.

5-Select all course name which start with ‘cs’.

6- Select students which has lectures in hall_A.

Assumptions:

1-College must put many course schedules and must has many areas of study.

2-Area of study must be studied by many students, area of study must be contained in one college.

3-courseschedule must be put by the college, must schedule many course catalogues, must contain many courses daily schedule and must display many classrooms.

4-course daily schedule must be contained in many course schedules

5-class room must exist in one building

6-building must have many class rooms

7- people may be a student or employee

8- student must studies in one of area of study, student must achieve many grades and student must make many course enrollment

9-employee must own one job information only

10- job information must be owned by one employee

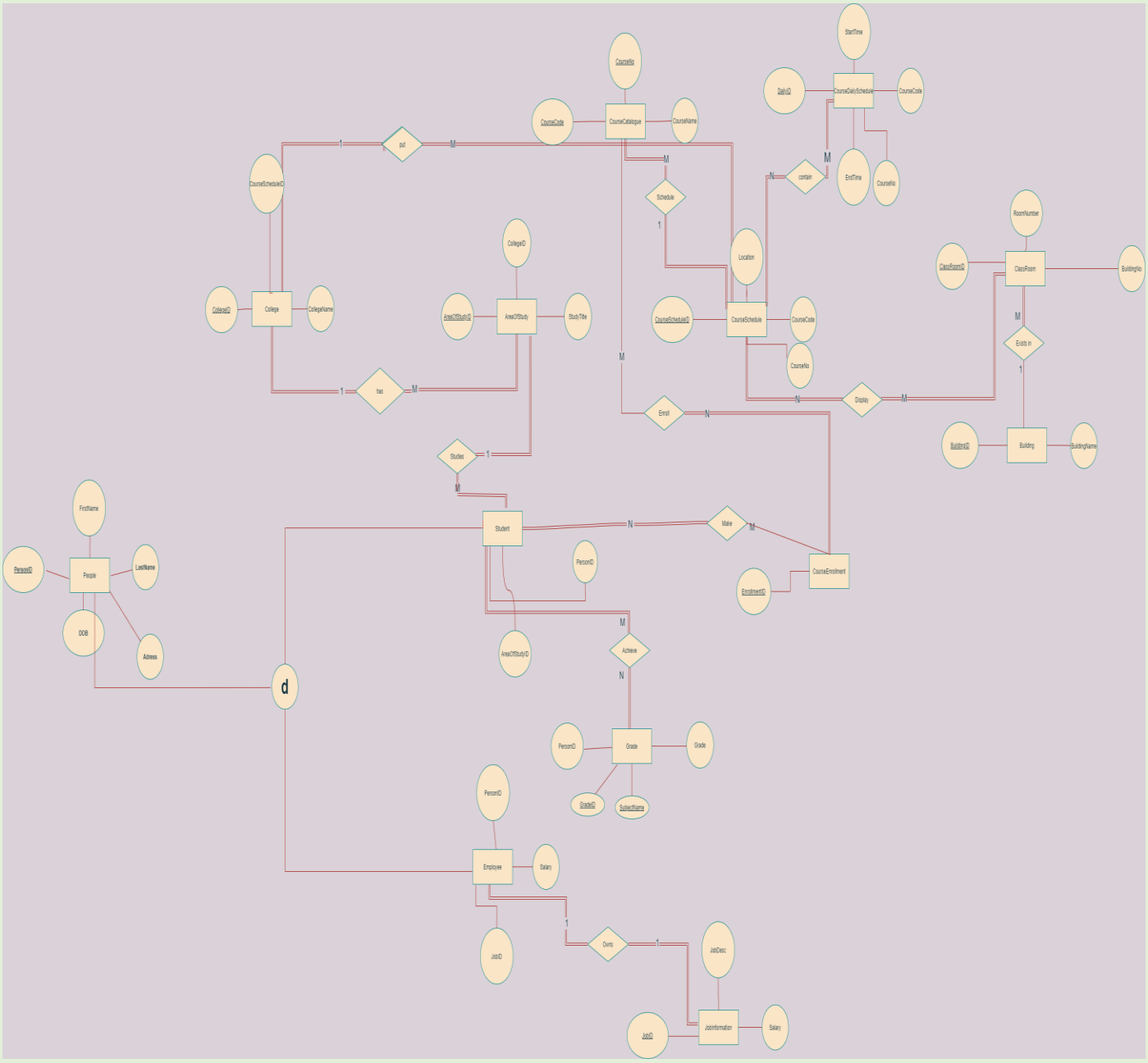
11- course enrollment must enroll many course catalogues and may made by many students

12- course catalogue may be enrolled by many course enrollments and must be scheduled by one course schedule

13- grade must be achieved by many students.

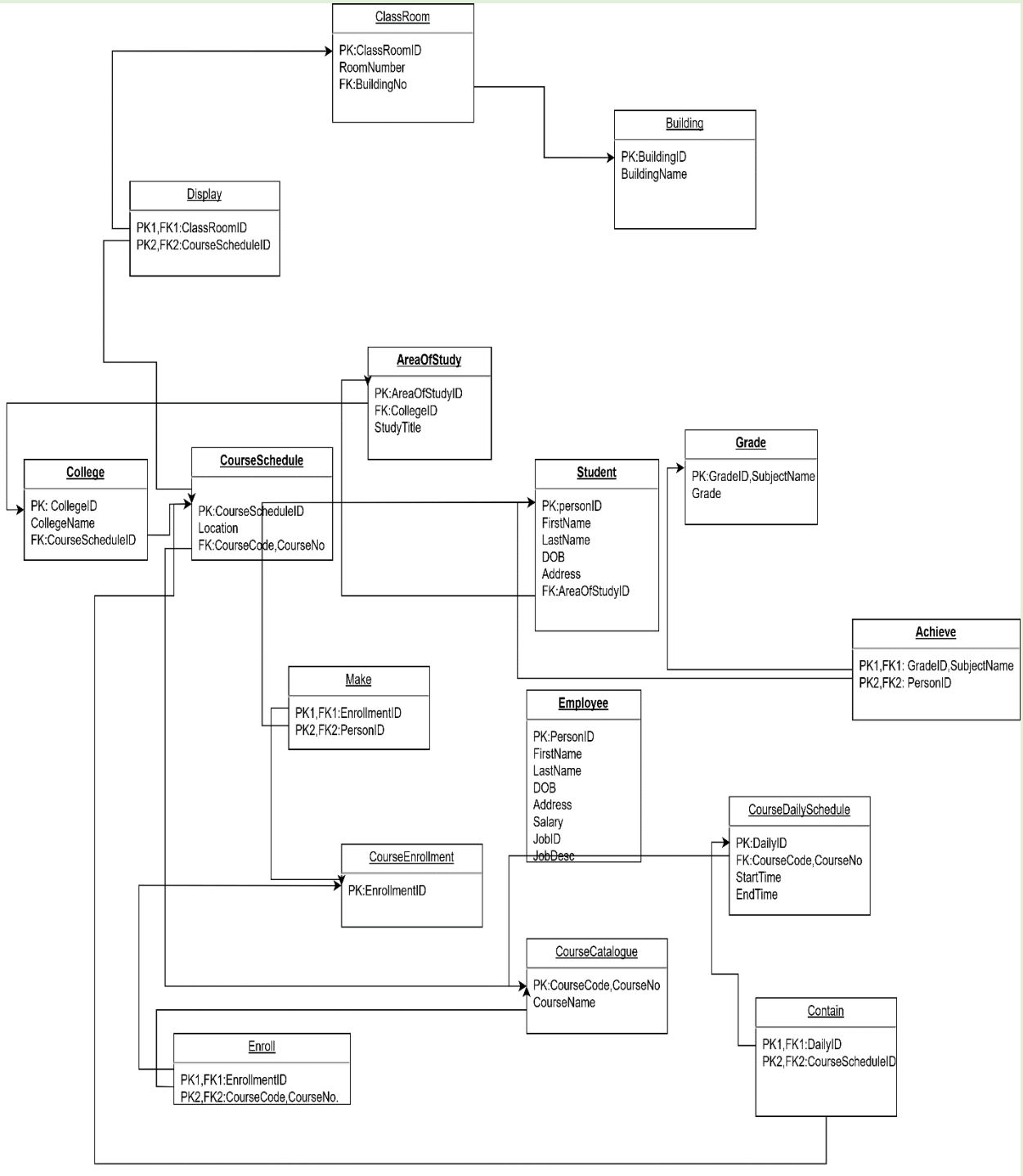
EER diagram:

Note: this image has high resolution so u can zoom in and it won't get blurry



Database Schema:

Note: this image has high resolution so u can zoom in and it won't get blurry



6- Sample of SQL

Create Tables And insert values :

```
CREATE TABLE CourseCatalogue(
CourseCode INT NOT NULL  ,
CourseNo INT NOT NULL  ,
PRIMARY KEY (CourseCode,CourseNo) ,
CourseName VARCHAR(255) NOT NULL );


CREATE TABLE CourseEnrollment(
EnrollmentID INT NOT NULL PRIMARY KEY  );


CREATE TABLE Enroll(
EnrollmentID INT NOT NULL,
CourseCode INT NOT NULL  ,
CourseNo INT NOT NULL ,
```

```
PRIMARY KEY (CourseCode,CourseNo,EnrollmentID) ,

FOREIGN KEY (CourseCode,CourseNo) REFERENCES
coursecatalogue(CourseCode,CourseNo),

FOREIGN KEY (EnrollmentID) REFERENCES
courseenrollment(EnrollmentID),

)ENGINE=INNODB;


CREATE TABLE CourseSchedule(

CourseScheduleID INT NOT NULL,

Location VARCHAR(255) NOT NULL,

CourseCode INT NOT NULL  ,

CourseNo INT NOT NULL ,

PRIMARY KEY ( CourseScheduleID)

FOREIGN KEY (CourseCode,CourseNo) REFERENCES
coursecatalogue(CourseCode,CourseNo));


CREATE TABLE College(

CollegeID  INT NOT NULL,
```

```
CollegeName  VARCHAR(50) NOT NULL,

CourseScheduleID  INT NOT NULL,

PRIMARY KEY (CollegeID )  ,

FOREIGN KEY  ( CourseScheduleID)  REFERENCES
courseschedule(CourseScheduleID));


CREATE TABLE  AreaOfStudy(

AreaOfStudyID  INT NOT NULL,

StudyTitle  VARCHAR(255) NOT NULL,

CollegeID  INT NOT NULL,

PRIMARY KEY( AreaOfStudyID),

FOREIGN KEY ( collegeID) REFERENCES college(CollegeID));


CREATE TABLE  Student(

PersonID  INT NOT NULL,

FirstName  VARCHAR(50) NOT NULL,

LasttName  VARCHAR(50) NOT NULL,

Address  VARCHAR(50) NOT NULL,

DOB  DATE NOT NULL,
```

```
AreaOfStudyID  INT NOT NULL,

PRIMARY KEY(PersonID),

FOREIGN KEY (AreaOfStudyID) REFERENCES
areaofstudy(AreaOfStudyID));



CREATE TABLE Employee(

PersonID  INT NOT NULL,

FirstName  VARCHAR(50) NOT NULL,

LasttName  VARCHAR(50) NOT NULL,

Address  VARCHAR(50) NOT NULL,

DOB  DATE NOT NULL,

Salary  INT NOT NULL,

JobID  INT NOT NULL,

JobDesc  VARCHAR(255) NOT NULL,

PRIMARY KEY(PersonID));

CREATE TABLE Make(

PersonID INT NOT NULL,
```

```
EnrollmentID INT NOT NULL,

PRIMARY KEY(PersonID,EnrollmentID),

(FOREIGN KEY  (PersonID) REFERENCES student( PersonID),

FOREIGN KEY (EnrollmentID) REFERENCES
courseenrollment(EnrollmentID));



CREATE TABLE Grade(

GradeID INT NOT NULL,

SubjectName VARCHAR(50) NOT NULL,

Grade VARCHAR(50) NOT NULL,

PRIMARY KEY(GradeID,SubjectName));


CREATE TABLE Achieve(

GradeID INT NOT NULL,

SubjectName VARCHAR(50) NOT NULL,

PersonID INT NOT NULL,

PRIMARY KEY(GradeID,SubjectName,PersonID),
```

```
FOREIGN KEY (GradeID,SubjectName) REFERENCES
grade(GradeID,SubjectName),

FOREIGN KEY (PersonID) REFERENCES student(PersonID));


CREATE  TABLE CourseDailySchedule(

DailyID INT NOT NULL,

CourseCode INT NOT NULL,

CourseNo INT NOT NULL,

StartTime VARCHAR(50) NOT NULL,

EndTime VARCHAR(50) NOT NULL,

PRIMARY KEY (DailyID),

FOREIGN KEY (CourseCode,CourseNo) REFERENCES
coursecatalogue(CourseCode,CourseNo));


CREATE TABLE Contain(

DailyID INT NOT NULL,

CourseScheduleID INT NOT NULL,

PRIMARY KEY(DailyID ,CourseScheduleID),
```

```
FOREIGN KEY (DailyID) REFERENCES
coursedailyschedule(DailyId),

FOREIGN KEY (CourseScheduleID) REFERENCES
courseschedule(CourseScheduleID));



CREATE Table Building(

BuildingID INT NOT NULL,

BuildingName VARCHAR(50) NOT NULL,

PRIMARY KEY (BuildingID));


CREATE TABLE  ClassRoom(

ClassRoomID INT NOT NULL,

RoomNumber INT NOT NULL,

BuildingNo INT NOT NULL,

PRIMARY KEY (CLassRoomID),

FOREIGN KEY (BuildingNo) REFERENCES building(BuildingID));


CREATE TABLE  Display(
```

```
ClassRoomID INT NOT NULL,

CourseScheduleID INT NOT NULL,

PRIMARY KEY(ClassRoomID ,CourseScheduleID),

FOREIGN KEY (CourseScheduleID) REFERENCES
courseschedule(CourseScheduleID),

FOREIGN KEY (ClassRoomID) REFERENCES
ClassRoom(ClassRoomID));



INSERT INTO
coursecatalogue(CourseCode,CourseNo,CourseName)

VALUES(111,0,'csa');


INSERT INTO
coursecatalogue(CourseCode,CourseNo,CourseName)

VALUES(112,1,'csb');


INSERT INTO
coursecatalogue(CourseCode,CourseNo,CourseName)

VALUES(113,2,'csc');
```

```
INSERT INTO courseenrollment(EnrollmentID)
VALUES(0);
INSERT INTO courseenrollment(EnrollmentID)
VALUES(1);

INSERT INTO courseenrollment(EnrollmentID)
VALUES(2);

INSERT INTO enroll(EnrollmentID,CourseCode,CourseNo)
VALUES(0,111,0);
INSERT INTO enroll(EnrollmentID,CourseCode,CourseNo)
VALUES(1,112,1);
INSERT INTO enroll(EnrollmentID,CourseCode,CourseNo)
VALUES(2,113,2);
```

```
INSERT INTO
courseschedule(CourseScheduleID,Location,CourseCode,Course
No)

VALUES(0,'hall_A','111',0);

INSERT INTO
courseschedule(CourseScheduleID,Location,CourseCode,Course
No)

VALUES(1,'hall_B','112',1);

INSERT INTO
courseschedule(CourseScheduleID,Location,CourseCode,Course
No)

VALUES(2,'hall_C','113',2);


INSERT INTO
college(CollegeID,CollegeName,CourseScheduleID)

Values(0,'Engineering',0);

INSERT INTO
college(CollegeID,CollegeName,CourseScheduleID)

Values(1,'Art',1);

INSERT INTO
college(CollegeID,CollegeName,CourseScheduleID)
```

```
Values(2,'Mass_Communication',2);

INSERT INTO AreaOfStudy(AreaOfStudyID,StudyTitle,CollegeID)
VALUES(0,'A',0);
INSERT INTO AreaOfStudy(AreaOfStudyID,StudyTitle,CollegeID)
VALUES(1,'B',1);

INSERT INTO AreaOfStudy(AreaOfStudyID,StudyTitle,CollegeID)
VALUES(2,'C',2);


INSERT INTO
employee(PersonID,FirstName,LastName,DOB,Address,
Salary,JobID,JobDesc)
VALUES(0,'ali','jo','1970-01-10','asdf1',123,0,'sw');
INSERT INTO
employee(PersonID,FirstName,LastName,DOB,Address,
Salary,JobID,JobDesc)
VALUES(1,'alirn','joeen','1970-01-11','aseedf1',133,1,'sws');
```

```
INSERT INTO
employee(PersonID,FirstName,LastName,DOB,Address,

Salary,JobID,JobDesc)

VALUES(2,'alien','jor','1970-01-12','aseedf1',143,2,'swj');


INSERT INTO grade(GradeID,SubjectName,Grade)

VALUES(0,'math','A+');

INSERT INTO grade(GradeID,SubjectName,Grade)

VALUES(1,'English','B');

INSERT INTO grade(GradeID,SubjectName,Grade)

VALUES(2,'Physicis','A');


INSERT INTO achieve(GradeID,SubjectName,PersonID)

VALUES(0,'math',0);

INSERT INTO achieve(GradeID,SubjectName,PersonID)

VALUES(1,'English',1);

INSERT INTO achieve(GradeID,SubjectName,PersonID)

VALUES(2,'Physicis',2);
```

```
INSERT INTO
coursedailyschedule(DailyID,CourseCode,CourseNo,StartTime,E
ndTime)

VALUES(0,'111',0,'1','2');

INSERT INTO
coursedailyschedule(DailyID,CourseCode,CourseNo,StartTime,E
ndTime)

VALUES(1,'112',1,'3','4');

INSERT INTO
coursedailyschedule(DailyID,CourseCode,CourseNo,StartTime,E
ndTime)

VALUES(2,'113',2,'5','6');


INSERT INTO contain(DailyID,CourseScheduleID)

VALUES(0,0);

INSERT INTO contain(DailyID,CourseScheduleID)

VALUES(1,1);

INSERT INTO contain(DailyID,CourseScheduleID)

VALUES(2,2);
```

```
INSERT INTO building(BuildingID,BuildingName)

VALUES(0,'hall_A');

INSERT INTO building(BuildingID,BuildingName)

VALUES(1,'hall_B');

INSERT INTO building(BuildingID,BuildingName)

VALUES(2,'hall_C');


INSERT INTO
ClassRoom(ClassRoomID,RoomNumber,BuildingNo)

VALUES(0,0,0);

INSERT INTO
ClassRoom(ClassRoomID,RoomNumber,BuildingNo)

VALUES(1,1,1);

INSERT INTO
ClassRoom(ClassRoomID,RoomNumber,BuildingNo)

VALUES(2,2,2);


INSERT INTO Display(ClassRoomID,CourseScheduleID)
```

```
VALUES(0,0);
INSERT INTO Display(ClassRoomID,CourseScheduleID)
VALUES(1,1);
INSERT INTO Display(ClassRoomID,CourseScheduleID)
VALUES(2,2);

INSERT INTO make(EnrollmentID,PersonID)
VALUES(0,0);
INSERT INTO make(EnrollmentID,PersonID)
VALUES(1,1);
INSERT INTO make(EnrollmentID,PersonID)
VALUES(2,2);
```

Updates And DELETE :

```
UPDATE coursedailyschedule
SET StartTime=6
WHERE DailyID=2;
```

```
UPDATE coursedailyschedule
SET EndTime=8
WHERE DailyID=2;

INSERT INTO college(CollegeID,CollegeName,CourseScheduleID)
Values(3,'Politiics',2);

DELETE from college
WHERE CollegeID=3;
```

We implement 6 Reports:

1- Select Students which has AreaofStudyID = 1

```
SELECT * FROM student

WHERE AreaOfStudyID=1;
```

2- Select firstname , last name of employee which has salary over 123

```
SELECT FirstName,LastName

From employee

WHERE Salary>123;
```

3- for each employee which has address='aseedf1' , select number of employess which live in this address

```
SELECT COUNT(*)AS Count1,Address

from employee

WHERE Address='aseedf1'

GROUP BY Address;
```

4- select first name of students which study tittle is B

```
Select S.Firstname

From Student As S inner join AreaOfStudy as a

on s.AreaOfStudyID =  a.AreaOfStudyID
```

where StudyTitle='B';

5- Select all course name which start with cs

select CourseName

from coursecatalogue

where CourseName like 'CS%';

6- Select students which has lectures in hall_A

Select S.PersonID,S.FirstName,C.Location

From Student As s , AreaOfStudy As A , College As co , CourseSchedule As c

where S.Areaofstudyid=A.Areaofstudyid AND A.collegeID = Co.CollegeID AND Co.coursescheduleID = C.coursescheduleID AND Location="hall_A";

7-Implemenation:

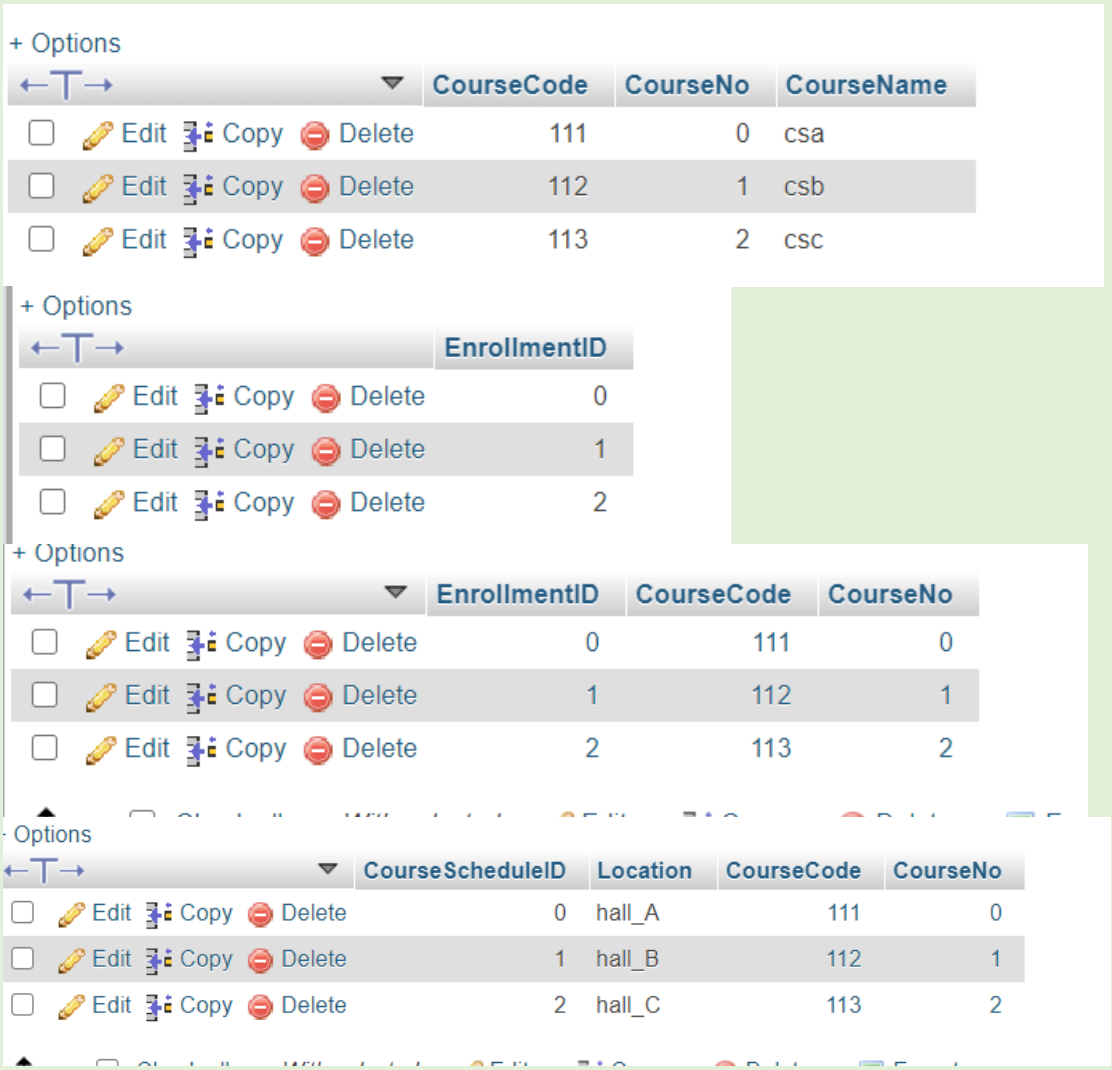
1- Create

a-Create CourseCatalogue , CourseEnrollment,Enroll,

CourseSchedule And insert data into them

```
CREATE TABLE CourseCatalogue(
    CourseCode INT NOT NULL ,
    CourseNo INT NOT NULL,
    PRIMARY KEY (CourseCode,CourseNo),
    CourseName VARCHAR(255) NOT NULL
    );

CREATE TABLE CourseEnrollment(
     EnrollmentID INT NOT NULL PRIMARY KEY
     );

 CREATE TABLE Enroll(
     EnrollmentID INT NOT NULL,
     CourseCode INT NOT NULL ,
    CourseNo INT NOT NULL,
    PRIMARY KEY (CourseCode,CourseNo,EnrollmentID),
    FOREIGN KEY (CourseCode,CourseNo) REFERENCES coursecatalogue(CourseCode,CourseNo),
      FOREIGN KEY (EnrollmentID) REFERENCES courseenrollment(EnrollmentID),
     )ENGINE=INNODB;

CREATE TABLE CourseSchedule(
    CourseScheduleID INT NOT NULL ,
    Location VARCHAR(255) NOT NULL ,
     CourseCode INT NOT NULL ,
    CourseNo INT NOT NULL,
    PRIMARY KEY ( CourseScheduleID),
    FOREIGN KEY (CourseCode,CourseNo) REFERENCES coursecatalogue(CourseCode,CourseNo)
    );
```

```
INSERT INTO coursecatalogue(CourseCode,CourseNo,CourseName)
VALUES(111,0,'csa');

NSERT INTO coursecatalogue(CourseCode,CourseNo,CourseName)
VALUES(112,1,'csb');

INSERT INTO coursecatalogue(CourseCode,CourseNo,CourseName)
VALUES(113,2,'csc');


INSERT INTO courseenrollment(EnrollmentID)
VALUES(0);
INSERT INTO courseenrollment(EnrollmentID)
VALUES(1);

INSERT INTO courseenrollment(EnrollmentID)
VALUES(2);

INSERT INTO enroll(EnrollmentID,CourseCode,CourseNo)
VALUES(0,111,0);
INSERT INTO enroll(EnrollmentID,CourseCode,CourseNo)
VALUES(1,112,1);
INSERT INTO enroll(EnrollmentID,CourseCode,CourseNo)
VALUES(2,113,2);

INSERT INTO courseschedule(CourseScheduleID,Location,CourseCode,CourseNo)
VALUES(0,'hall_A','111',0);
INSERT INTO courseschedule(CourseScheduleID,Location,CourseCode,CourseNo)
VALUES(1,'hall_B','112',1);
INSERT INTO courseschedule(CourseScheduleID,Location,CourseCode,CourseNo)
VALUES(2,'hall_C','113',2);
```



b-Update And Delete

Update values of StartTime,Endtime in CourseDailySchedule

```
UPDATE coursedailyschedule
SET StartTime=6
WHERE DailyID=2;

 UPDATE coursedailyschedule
SET EndTime=8
WHERE DailyID=2;
```

Options

	DailyID	CourseCode	CourseNo	StartTime	EndTime
Edit Copy Delete	0	111	0	1	2
Edit Copy Delete	1	112	1	3	4
Edit Copy Delete	2	113	2	6	8

b- Delete from College Table row of CollegeID=3

```
INSERT INTO college(CollegeID,CollegeName,CourseScheduleID)
 Values(3,'Politiics',2);


DELETE from college
WHERE CollegeID=3;
```

+ Options

	CollegeID	CollegeName	CourseScheduleID
Edit Copy Delete	0	Engineering	0
Edit Copy Delete	1	Art	1
Edit Copy Delete	2	Mass_Communication	2

C- 6 Reports Implementation

1-

```
261 SELECT * FROM student
262 WHERE AreaOfStudyID=1;
```

Options

	PersonID	FirstName	LastName	Address	DOB	AreaOfStudyID
Edit Copy Delete	1	jooyhn	jackw	1dd	1998-06-03	1

2-

```
SELECT FirstName,LastName
From employee
WHERE Salary>123;
```

ear Format Get auto-saved qu

+ Options

	FirstName	LastName
Edit Copy Delete	alirn	joeen
Edit Copy Delete	alien	jor

3-

```
SELECT COUNT(*)AS Count1,Address
from employee
WHERE Address='aseedf1'
GROUP BY Address;
```

+ Options

Count1	Address
2	aseedf1

Query results operations

4-

```
Select S.Firstname
From Student As S inner join AreaOfStudy as a
on s.AreaOfStudyID =  a.AreaOfStudyID
where StudyTitle='B';
```

+ Options

Firstname
jooyhn

5-

```
select CourseName
from coursecatalogue
where CourseName like 'CS%';
```

+ Options

	CourseName
Edit Copy Delete	csa
Edit Copy Delete	csb
Edit Copy Delete	csc

6-

```
Select S.PersonID,S.FirstName,C.Location
From Student As s , AreaOfStudy As A , College As co , CourseSchedule As c
where S.Areaofstudyid=A.Areaofstudyid AND A.collegeID = Co.CollegeID AND Co.courseScheduleID = C.courseScheduleID AND Location="hall_A";
```

+ Options

PersonID	FirstName	Location
0	john	hall_A

Show all | Number of rows: 25 Filte