Educational Resource Management System

INDEX

Project Description:

An organized and systematic office solution is essential for all universities and organizations. There are many departments of administration for the maintenance of college information and student databases in any institution. All the modules in college administration are interdependent. They are maintained manually. So they need to be automated and centralized as, Information from one module will be needed by other modules.

Educational Resource Management System deals with all kind of student details, academic related reports, professor details, course details, curriculum, batch details, internship details, E-learning material details, book store, summer project details and other resource related details too. It tracks all the details of a student from the day one to the end of his course which can be used for all reporting purpose, tracking of attendance, progress in the course, completed semesters years, coming semester year curriculum details, exam details, project, final exam result etc. Our design can facilitate us to explore all the activities happening in the college, even we can get to know which faculty is assigned to which course, the current status of a student, attendance percentage of a student and upcoming requirements of a student like books.

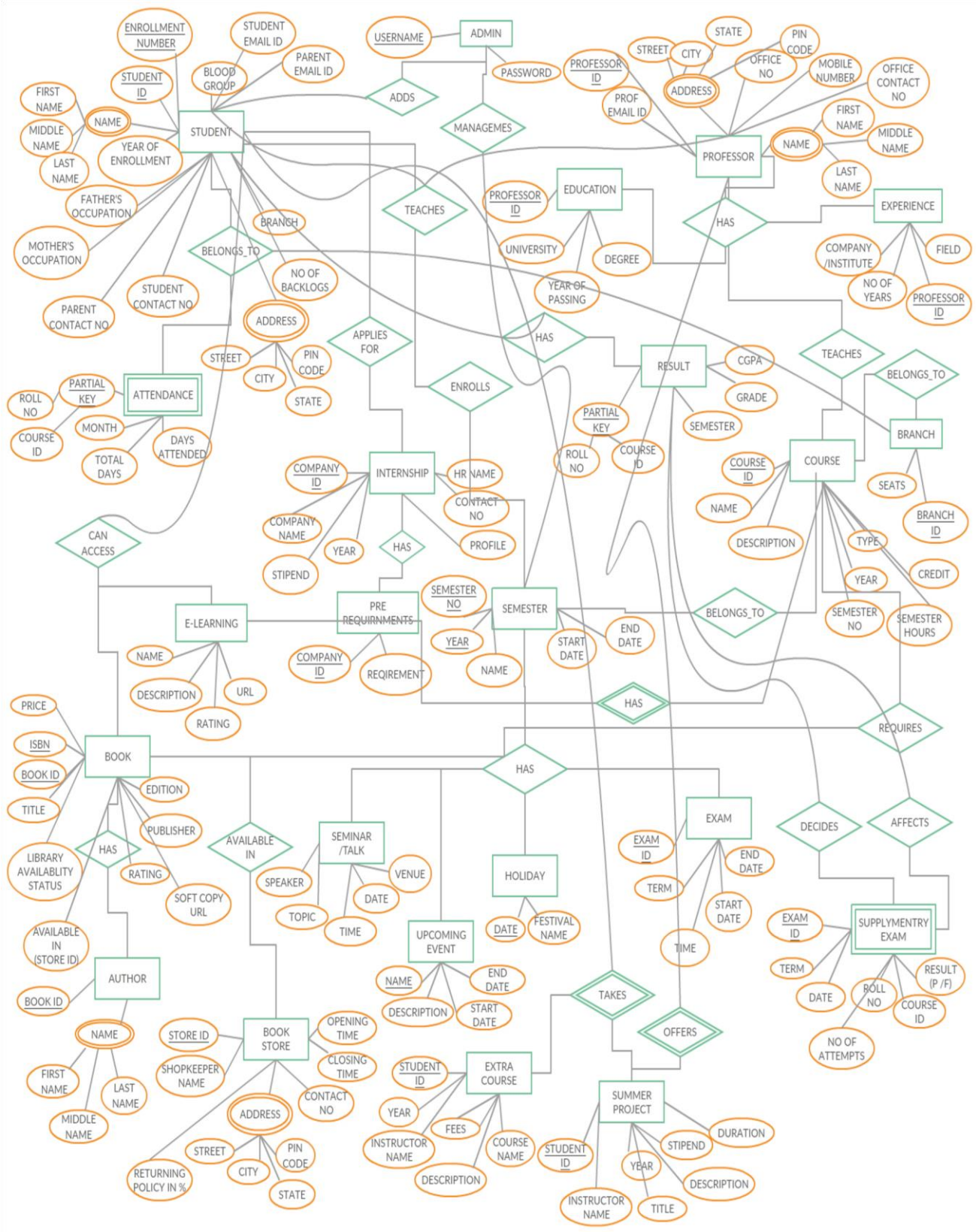
The Educational Resource Management System is an automated version of manual system. In case of manual system they need a lot of time, manpower etc. Here all work is computerized. So the accuracy is maintained. Maintaining backup is very easy which it can do with in a few minutes. Our system has two type of accessing modes, administrator and user. Student management system is managed by an administrator. It is the job of the administrator to insert update and monitor the whole process. While a student would only view details of different types. He/she can't perform any changes.

Main features of this Database System:

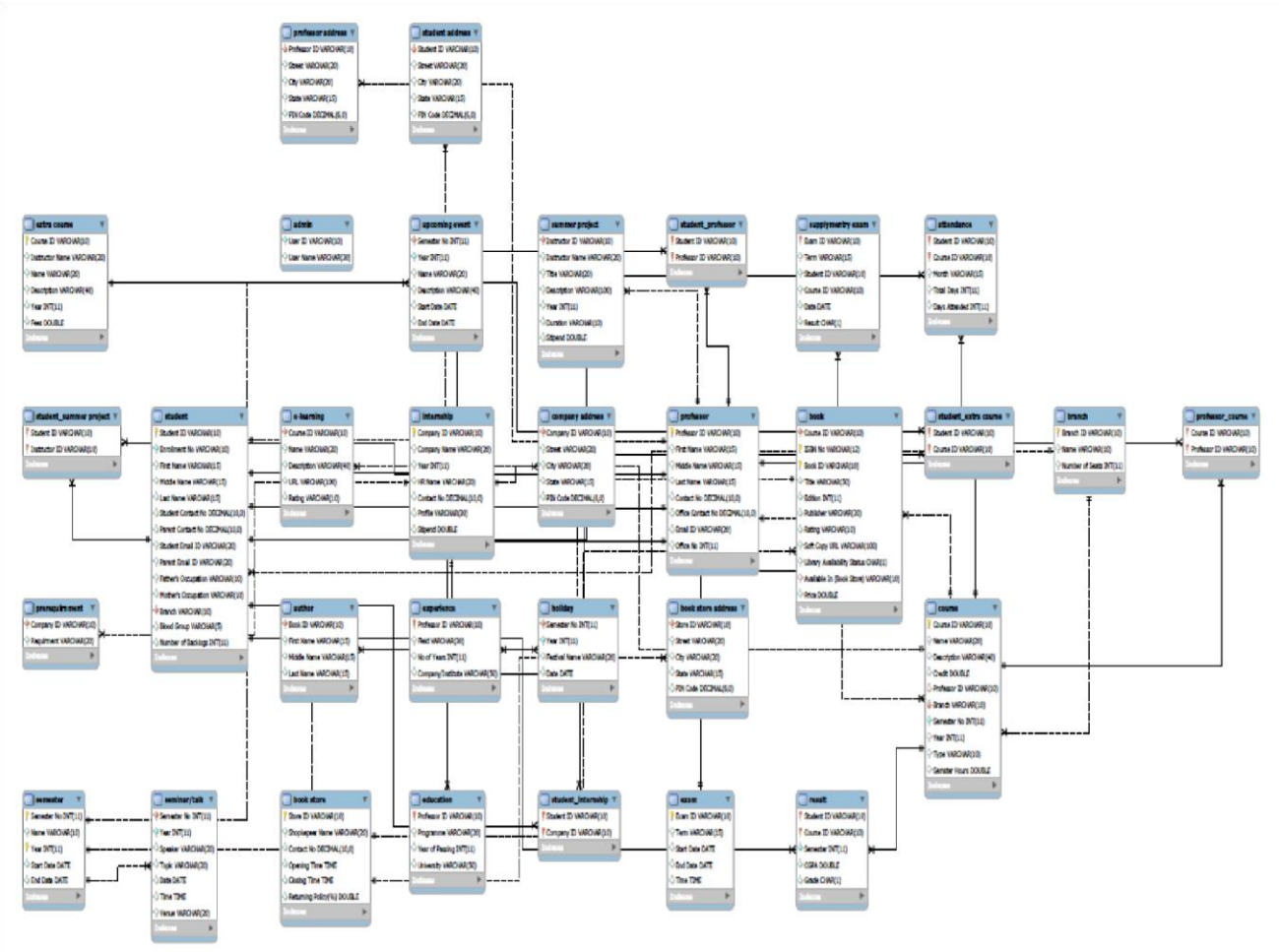
- Find internships
- Find summer projects & extra credit courses of university
- Know about upcoming events and seminars
- Know about exam schedule
- Explore professor details
- Checkout library availability status of any book
- Find book shops near you
- Checkout your attendance
- Know your GPA & CGPA

Thus, this can be a very useful data base to all the students of an institution because it includes almost all kind of details related to study whether it is a finding a soft copy of any book or checking out upcoming seminars at university or finding internship opportunity.

- Entity Relation Diagram



- Schema Diagram



- Attribute, Data type and Constraints of each Entity

Entity: Student

Attribute	Data type	Constraints
Student ID	Varchar(10)	Primary Key
Enrollment No	Varchar(10)	Not Null
First Name	Varchar(15)	Not Null
Middle Name	Varchar(15)	Not Null
Last Name	Varchar(15)	Not Null
Student Contact No	Decimal(10, 0)	Not Null
Parent Contact No	Decimal(10, 0)	Not Null
Student Email ID	Varchar(20)	Not Null
Parent Email ID	Varchar(20)	Not Null
Father's Occupation	Varchar(10)	Not Null
Mother's Occupation	Varchar(10)	Not Null
Branch	Varchar(10)	Foreign Key(Branch), Not Null
Blood Group	Varchar(5)	Not Null
Number of Backlogs	Int	

Entity: E-Learning

Attribute	Datatype	Constraints
Course ID	Varchar(10)	Foreign Key(Course), Not Null
Name	Varchar(20)	
Description	Varchar(40)	
URL	Varchar(100)	Not Null
Rating	Varchar(10)	

Entity: Internship

Constraint	Constraint	Constraint

Company ID	Varchar(10)	Primary Key, Not Null
Company Name	Varchar(20)	Not Null
Year	Int	Not Null
HR Name	Varchar(20)	Not Null
Contact No	Decimal(10, 0)	Not Null
Profile	Varchar(30)	
Stipend	Real	

Entity: Company Address

Attribute	Datatype	Constraints
Company ID	Varchar(10)	Foreign Key(Internship), Not Null
Street	Varchar(20)	Not Null
City	Varchar(20)	Not Null
State	Varchar(15)	Not Null
PIN Code	Decimal(6, 0)	Not Null

Entity: Professor

Attribute	Datatype	Constraints
Professor ID	Varchar(10)	Primary Key
First Name	Varchar(20)	Not Null
Middle Name	Varchar(20)	Not Null
Last Name	Varchar(20)	Not Null
Contact No	Decimal(10, 0)	Not Null
Office Contact No	Decimal(10, 0)	Not Null
Email ID	Varchar(20)	Not Null
Office No	Int	Not Null

Entity: Book

Attribute	Datatype	Constraints

Course ID	Varchar(10)	Foreign Key(Course), Not Null
ISBN No	Varchar(12)	Not Null
Book ID	Varchar(10)	Primary Key
Title	Varchar(50)	Not Null
Edition	Int	
Publisher	Varchar(30)	Not Null
Rating	Varchar(10)	
Soft Copy URL	Varchar(100)	
Library Availability Status	Char	Not Null
Available In (Book Store)	Varchar(10)	
Price	Real	Not Null

Entity: Student_extra course

Constraint	Constraint	Constraint
Student ID	Varchar(10)	Foreign Key(Student), Primary Key
Course ID	Varchar(20)	Foreign Key(Course), Primary Key

Entity: Branch

Attribute	Datatype	Constraints
Branch ID	Varchar(10)	Primary Key
Name	Varchar(10)	Not Null
Number of Seats	Int	Not Null

Entity: Professor_course

Attribute	Datatype	Constraints
Professor ID	Varchar(10)	Foreign Key(Professoe), Primary Key
Course ID	Varchar(20)	Foreign Key(Course), Primary Key

Entity: Pre requirement

Attribute	Datatype	Constraints
Company ID	Varchar(10)	Foreign Key(Internship)
Requirement	Varchar(20)	

Entity: Author

Attribute	Datatype	Constraints
Book ID	Varchar(10)	Foreign Key(Book)
First Name	Varchar(15)	Not Null
Middle Name	Varchar(15)	Not Null
Last Name	Varchar(15)	Not Null

Entity: Experience

Attribute	Datatype	Constraints
Professor ID	Varchar(10)	Foreign Key(Professor), Primary Key
Field	Varchar(30)	Not Null
Number of Years	Int	Not Null
Company/Institute	Varchar(50)	Not Null

Entity: Holiday

Attribute	Datatype	Constraints
Semester No	Int	Foreign Key(Semester)
Year	Int	Not Null
Festival Name	Varchar(20)	Not Null
Date	Date	Not Null

Entity: book store address

Attribute	Data type	Constrain
Store ID	VARCHAR(10)	Foreign key (Book Store), NOT NULL
Street	VARCHAR(20)	NOT NULL

City	VARCHAR(20)	NOT NULL
State	VARCHAR(15)	NOT NULL
PIN Code	DECIMAL(6,0)	NOT NULL

Entity: Course

Attribute	Data type	Constrain
Course ID	VARCHAR(10)	PRIMARY KEY, NOT NULL
Name	VARCHAR(20)	NOT NULL
Description	VARCHAR(40)	NOT NULL
Credit	REAL	Foreign key (Professor), NOT NULL
Professor ID	VARCHAR(10)	UNIQUE, NOT NULL
Branch	VARCHAR(10)	Foreign key (Branch), NOT NULL
Semester No	INT	NOT NULL
Year	INT	NOT NULL
Type (core, elective)	VARCHAR(10)	NOT NULL
Semester Hours	REAL	

Entity: Semester

Attribute	Data type	Constrain
Semester No	INT	PRIMARY KEY, NOT NULL
Year	INT	
Name (winter /monsoon)	VARCHAR(10)	NOT NULL
Start Date	DATE	NOT NULL
End Date	DATE	

Entity: Education

Attribute	Data type	Constrain
Professor ID	VARCHAR(10)	PRIMARY KEY, FOREIGN KEY(Professor), NOT NULL

Programme	VARCHAR(20)	NOT NULL
Year of Passing	INT	NOT NULL
University	VARCHAR(50)	NOT NULL

Entity:

Seminar/Talk

Attribute	Data type	Constraint
Semester No	INT	FOREIGN KEY(semester), NOT NULL
Year	INT	NOT NULL
Speaker	VARCHAR(10)	NOT NULL
Topic	DATE	
Date	DATE	NOT NULL
Time	TIME	NOT NULL
venue	VARCHAR	NOT NULL

Entity: Book store

Attribute	Data type	Constraint
Store ID	VARCHAR(10)	PRIMARY KEY, NOT NULL
Shopkeeper Name	VARCHAR(20)	NOT NULL
Contact No	DECIMAL(10, 0)	NOT NULL
Opening Time	DATE	NOT NULL
Closing Time	DATE	NOT NULL
Returning Policy (%)	REAL	NOT NULL

Entity: Student_Internship

Attribute	Data type	Constraint
Student ID	VARCHAR(10)	PRIMARY KEY, FOREIGN KEY(Student), NOT NULL
Company ID	VARCHAR(10)	PRIMARY KEY, FOREIGN KEY(Internship), NOT NULL

Entity: Exam

Attribute	Data type	Constraint
Exam ID	VARCHAR(10)	PRIMARY KEY, NOT NULL

Entity:

Term	VARCHAR(15)	NOT NULL
Start Date	DATE	NOT NULL
End Date	DATE	NOT NULL
Time	TIME	NOT NULL

Entity: Student_Summer Project

Attribute	Data Type	Constraint
Student ID	Varchar(10)	Not null
Instructor ID	Varchar(10)	Not null

Professor Address

Attribute	Data Type	Constraint
Professor ID	Varchar(10)	Not null
Street	Varchar(20)	Not null
City	Varchar(20)	Not null
State	Varchar(15)	Not null
PIN Code	Decimal(6,0)	Not null

Entity: Result

Attribute	Data type	Constraint
Student ID	VARCHAR(10)	PRIMARY KEY, FOREIGN KEY(Student),NOT NULL
Course ID	VARCHAR(10)	PRIMARY KEY, FOREIGN KEY(Course), NOT NULL
Semester	INT	NOT NULL
CGPA	REAL	NOT NULL
Grade	CHAR	NOT NULL

Entity: Student Address

Attribute	Data Type	Constraint
Student ID	Varchar(10)	Not null
Street	Varchar(20)	Not null

Entity:

City	Varchar(20)	Not null
State	Varchar(15)	Not null
PIN Code	Decimal(6,0)	Not null

Entity: Extra Course

Attribute	Data Type	Constraint
Course ID	Varchar(10)	Not null
Instructor Name	Varchar(20)	Not null
Name	Varchar(20)	Not null
Description	Varchar(40)	-
Year	Int	Not null
Fees	Double	-

Entity: Student_Professor

Attribute	Data Type	Constraint
Student ID	Varchar(10)	Not null
Professor ID	Varchar(10)	Not null

Admin

Attribute	Data Type	Constraint
User ID	Varchar(10)	Not null
User Name	Varchar(30)	Not null
Fname	Varchar(50)	Not null
Lname	Varchar(45)	Not null

Entity: Upcoming Event

Attribute	Data Type	Constraint
Semester No	Int	Not null
Year	Int	Not null
Name	Varchar(20)	Not null
Description	Varchar(20)	-
Start Date	Date	-

Entity:

End Date	Date	-

Entity: Summer Project

Attribute	Data Type	Constraint
Instructor ID	Varchar(10)	Not null
Instructor Name	Varchar(20)	Not null
Title	Varchar(20)	Not null
Description	Varchar(100)	-
Year	Int	Not null
Duration	Varchar(10)	Not null
Stipend	Double	-

Entity: Supplementary Exam

Attribute	Data Type	Constraint
Exam ID	Varchar(10)	Not null
Term	Varchar(15)	Not null
Student ID	Varchar(10)	Not null
Course ID	Varchar(10)	Not null
Date	Date	Not null
Result	Char	Not null

Entity: Attendance

Attribute	Data Type	Constraint
Student ID	Varchar(10)	Not null
Course ID	Varchar(10)	Not null
Month	Varchar(15)	Not null
Total Days	Int	Not null
Days Attended	Int	Not null

List of Queries:

1. List the attendance of student ‘X’ in course ‘Y’.
2. List the students who have attendance less the 80% in course ‘X’.
3. List the total attendance of student ‘X’ in month ‘Y’.
4. Retrieve all students of B.Tech.(ICT) batch ‘X’ having CGPA >= 3.0.
5. List Courses offered by Instructor ‘X’ in ‘Winter’ of 2017.
6. Display the CGPA of student ‘X’.
7. Display the Grade of student ‘X’ in course ‘Y’
8. List IDs of students of B.Tech.(ICT) batch ‘X’ having any F grade in subject ‘Y’.
9. List students who have got A grade in subject ‘X’.
10. List the names of students who have passed supplementary exam of course ‘X’ in 1 attempt.
11. List Company names offering internship in year ‘X’.
12. List the pre requirements of company ‘X’ for profile ‘Y’.
13. List company names who are offering stipend more than amount ‘X’.
14. List all the contact details of company ‘X’.
15. List all the offered summer projects in year ‘X’.
16. List all courses offered by Prof ‘X’.
17. List the students who has back logs more than 2.
18. List the email address of all the students of semester ‘X’.
19. List all the student of branch ‘X’.
20. Find number of students in batch ‘X’ and branch ‘Y’.
21. List all the students who are not from Ahmedabad.
22. List the students who got internship in company ‘X’.
23. List the names of all the professor who teaches DBMS in semester 4.
24. List the names of professors who has experience of more than 10 years.
25. List all the professors who has Ph.D. degree.
26. List professors having experience in field /company ‘X’.
27. Display all the contact details of professor ‘X’, like email id, office no, mobile no etc.
28. Display the list of holidays in year ‘X’.

29. Display the speaker and topic of talk on date ‘X’.
30. Display all the seminars of month ‘X’ in year ‘Y’.
31. Give the description of event ‘X’.
32. List the students who have taken extra course ‘X’ under the mentorship of instructor ‘Y’.
33. List the students who have taken extra course after every semester.
34. List the students who have taken summer project ‘X’ under the mentorship of instructor ‘Y’.
35. List out all the e learning websites of course ‘X’ in ascending order.
36. List out all the websites of competitive coding.
37. List out all the books of course ‘X’.
38. Find the library availability status of book ‘X’.
39. Display URL of soft copy of all the courses of semester ‘X’ of branch ‘Y’.
40. Display no of books which are currently available in library.
41. Display names of all the authors of book ‘X’.
42. List the books written by author ‘X’.
43. List all the book store of city ‘X’ having book ‘Y’.
44. Find the closing time of all the book store in street ‘X’.
45. List the library availability status of all the books of semester ‘X’ branch ‘Y’.
46. Show all the books of course ‘X’ in ascending order of their rating.
47. Show all the books of course ‘X’ in descending order of their price.
48. Show returning policy of all book stores of city ‘X’.
49. List the book having title ‘X’ of any edition which are currently available in the library.
50. Display the starting and ending date of mid semester and end semester exam of semester ‘X’.

- Normalised form of tables

- Admin ○ (User ID,User Name)
  Primary key -> {User ID}

- Branch ○ (Branch ID ,Name , Number of Seats)
  Primary key -> {Branch ID}
  Candidate key -> {Name}

- Student ○ (Student ID ,Enrollment No,First Name ,Middle Name,Last Name ,Student Contact
  No,Parent Contact No ,Student Email ID,Parent Email ID,Father's
  Occupation,Mother's Occupation,Branch, Blood Group,Number of Backlogs`)
  Primary key -> {Student ID}
  Candidate key -> {Enrollment no, Student Email ID }

- Professor ○ (Professor ID, First Name ,Middle Name ,Last Name ,Contact No, Office Contact No,Email ID,Office No)
  Primary key -> { Professor ID }
  Candidate key -> {Email ID , Office No}

- Education ○ (Professor ID, Programme ,Year of Passing, University)
  Primary key -> { Professor ID }

- Experience ○ (Professor ID, Field ,No of Years,Company/Institute)
  Primary key {Professor ID, Company/Institute`}

- Semester ○ (Semester No, Name, Year , Start Date , End Date)
  Primary key { Semester No, Year}

- Course
  - (Course ID , Name , Description ,Credit, Professor ID , Branch , Semester No, Year,Type Semster Hours)

  Primary key { Course ID}

  Candidate key {Name}

- Attendance ○ (Student ID ,Course ID, Month, Total Days,Days Attended)
- Primary key { Student ID, Course ID , month}
- `Summer Project`
  - (Instructor ID ,Instructor Name, Title ,Description , Year, Duration, Stipend)

- Extra Course ○ (Course ID ,Instructor Name ,Name ,Description ,Year, Fees) Primary key {Course ID}

- E-Learning ○ (Course ID, Name ,Description, URL ,Rating)

  Primary key {Course ID}

  Candidate key{URL}

- Book Store ○ (Store ID, Shopkepeer Name ,Contact No , Opening Time, Closing Time ,Returning Policy(%))

  Primary key {Store ID}

  Candidate key {Contact No}

- Book
  - (Course ID,ISBN No, Book ID ,Title , Edition , Publisher ,Soft Copy URL ,Library Availability , Available In (Book Store) , Price)

  Primary key { Book ID, ISBN No }

  Candidate key{ Soft Copy URL}

- Author ○ (Book ID, First Name, Middle Name, Last Name)

  Primary key { Book ID}

- Seminar/Talk
  - (Semester No ,Year ,Speaker, Topic, Date ,Time , Venue)

- Upcoming event
  - (Year, Name , Description , Start Date, End Date)

- Holiday
    - (Year , Festival Name , Date)
- Exam
    - (Exam ID, Term ,Start Date ,End Date, Time)

  Primary key { Exam ID}

- Supplymentry Exam ○ (Exam ID , Term , Student ID ,Course ID , Date, Result) Primary key { Exam ID}
- Result ○ (Student ID, Course ID ,Semester , CGPA , Grade) Primary key { Student ID ,Course ID}
- Internship ○ (Company ID , Company Name ,Year ,HR Name , Contact No , Profile , Stipend(per month))

  Primary key { Company ID}

- Prerequirnment ○ (Company ID , Requirment) Primary key { Company ID}

1NF - For the table to be in 1NF table cell should contain single value and each record needs to be unique.

Example from our tables- In table, Professor, experience and education are multivalued , they could have been included in one row but to make it in 1NF we have made different tables of education and experience.

Professor ID	First Name	Middle Name	Last Name	Contact No	Office Contact No	Email ID	Office No
SEASP001	Anurag	NULL	Lakhlani	9809265735	7922665454	anurag@gmail.com	124
SEASP009	Dhaval	NULL	Patel	9802135735	7912344454	dhaval@gmail.com	98
SEASP011	Anupam	NULL	Lakhlani	9879265735	7922065454	anupam@gmail.com	224
SEASP012	Rita	NULL	Sharma	9234265735	7921225542	rita@gmail.com	111
SEASP019	Bharat	NULL	Patel	9802995735	7912340054	bharat@gmail.com	298

Professor ID	Field	No of Years	Company/Institute
SEASP001	System Designing	2	ABC Corporation
SEASP001	System Designing	6	Designing Corporation
SEASP009	Probability	8	University of America
SEASP012	Economics	7	LD arts
SEASP032	Electronics	12	Stanford

Professor ID	Programme	Year of Passing	University
SEASP001	M.TECH	1981	University of Chicago
SEASP001	B.TECH	1979	L.D.
SEASP012	M.TECH	1983	University of San-fransisco
SEASP012	B.TECH	1981	University of Chicago
SEASP032	M.TECH	1969	University of London

For Professor ID SEASP001, we have two different values of company i.e. ABC Corporation and Designing Corporation which we have shown using two different rows.

2NF- A table is said to be in 2NF if the table is in 1NF (First normal form) and no non-prime attribute is dependent on the proper subset of any candidate key of table.

Our all tables are in 2NF. There's no table in our project which has partial dependency.

3NF- A table design is said to be in 3NF if the table is in 2NF and if the Transitive functional dependency of non-prime attribute on any super key is removed.

Student ID	Enrollment No	First Name	Middle Name	Last Name	Student Contact No	Parent Contact No	Student Email ID
201301015	E201501015	Dhruti	Siddharth	Chandarana	9409265735	9658774345	dhruti@gmail.com
201301115	E201501115	Dhruti	Siddharth	Shah	7409265735	9658770345	dhruti@yahoo.com
201401021	E201501021	Charmi	Hiren	Chokshi	7359265567	9658774123	charmi@gmail.com
201401121	E201501121	Chandani	NULL	Chokshi	6659265567	9655774123	cc1510@yahoo.com
201501008	E201501008	Ashna	NULL	Jain	7365676787	9645674020	ashna1jain@gmail.com

As we can derive one's phone number from his/her name but if in case two person have same name then we can only get one's phone number from the student id.

BCNF- It is an advance version of 3NF that's why it is also referred as 3.5NF. BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every functional dependency X>Y, X should be the super key of the table.

Student ID	Enrollment No	First Name	Middle Name	Last Name	Student Contact No	Parent Contact No	Student Email ID
201301015	E201501015	Dhruti	Siddharth	Chandarana	9409265735	9658774345	dhruti@gmail.com
201301115	E201501115	Dhruti	Siddharth	Shah	7409265735	9658770345	dhruti@yahoo.com
201401021	E201501021	Charmi	Hiren	Chokshi	7359265567	9658774123	charmi@gmail.com
201401121	E201501121	Chandani	NULL	Chokshi	6659265567	9655774123	cc1510@yahoo.com
201501008	E201501008	Ashna	NULL	Jain	7365676787	9645674020	ashna1jain@gmail.com

Here, studentID is the primary key while enrollmentNo is a non prime key. In general enrollmentNo should be derived from studentID but in this table studentID can be derived from enrollmentNo. For that reason another table is made with two columns as enrollmentNo and studentID.

Thus, our database system is in BCNF (Boyce–Codd normal form).

MySQL Code to create database and insert data

```
CREATE SCHEMA `educational resource management system`;

USE `educational resource management system`;


CREATE TABLE Admin(
    `User ID` INT(4) NOT NULL PRIMARY KEY AUTO_INCREMENT ,
    `User Name` VARCHAR(30)
);

insert into Admin values(null,'Neha_Shukla'); insert into
Admin(`User Name`) values('Tanu_Jain'); insert into
Admin(`User Name`)  values('Manu_Dubey'); insert into
Admin(`User Name`)  values('Rahul_Chokshi'); insert into
Admin(`User Name`)  values('Dhruv_Chandarana'); insert into
Admin(`User Name`)  values('Nehal_Shukla'); insert into
Admin(`User Name`)  values('Tanmana_Jain'); insert into
Admin(`User Name`)  values('Manu_Shah'); insert into
Admin(`User Name`)  values('Rita_Chokshi');
insert into Admin(`User Name`)  values('Dhruvi_Chandarana');
```

```
CREATE TABLE Branch(
    `Branch ID` VARCHAR(10) NOT NULL,
   `Name` VARCHAR(10),
   `Number of Seats` INT,

   PRIMARY KEY(`Branch ID`)
);

insert into Branch values('SEAS01', 'ICT',120); insert
into Branch values('SEAS02', 'Mechanical',60); insert
into Branch values('SEAS03', 'Chemical',60); insert
into Branch values('SEAS04', 'Data Sci',60);
insert into Branch values('SEAS05', 'Electrical',30);

CREATE TABLE Student(
    `Student ID` VARCHAR(10) NOT NULL,
    `Enrollment No` VARCHAR(10) NOT NULL,
    `First Name` VARCHAR(15),
    `Middle Name` VARCHAR(15),
    `Last Name` VARCHAR(15),
    `Student Contact No` DECIMAL(10, 0),
    `Parent Contact No` DECIMAL(10, 0),
    `Student Email ID` VARCHAR(20),
    `Parent Email ID` VARCHAR(20),
    `Father's Occupation` VARCHAR(20),
    `Mother's Occupation` VARCHAR(50),
    `Branch` VARCHAR(10) NOT NULL,
    `Blood Group` VARCHAR(5),
    `Number of Backlogs` INT,

   PRIMARY KEY(`Student ID`),

   FOREIGN KEY(`Branch`) REFERENCES Branch(`Branch ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

insert into Student
values(201501051,'E201501051','Manasi',null,'Dubey',7359265735,9658774020,'manasitanu@gmail.com','jk
@gmail.com','IAF Officer','House wife','SEAS01','O+',0); insert
into Student
values(201501008,'E201501008','Ashna',null,'Jain',7365676787,9645674020,'ashna1jain@gmail.com','aj@gm
ail.com','Business man','House wife','SEAS01','A+',0); insert into Student
values(201401021,'E201501021','Charmi','Hiren','Chokshi',7359265567,9658774123,'charmi@gmail.com','h
c@gmail.com','Business man','House wife','SEAS03','A+',3); insert into Student
```

```
values(201301015,'E201501015 ','Dhruti','Siddharth','Chandarana',9409265735,9658774345,'dhruti@gmail.com','sc@gmail.com','Teacher','House wife','SEAS02','B+',0);

insert into Student
values(201501151,'E201501151','Manasi','Rakesh','Dubey',6359265735,9655774020,'manasitanu@yahoo.com','jk@gmail.com','IAF Officer','House wife','SEAS05','O',3); insert into Student
values(201501108,'E201501108','Ashna','Rahul
bhai','Jain',6365676787,9545674020,'ashna1jain@yahoo.com','aj@gmail.com','Business man','House wife','SEAS05','A',3); insert into Student
values(201401121,'E201501121','Chandani',null,'Chokshi',6659265567,9655774123,'cc1510@yahoo.com','hc@gmail.com','Business man','House wife','SEAS04','A',1); insert into Student
values(201301115,'E201501115','Dhruti','Siddharth','Shah',7409265735,9658770345,'dhruti@yahoo.com','sc@gmail.com','Teacher','House wife','SEAS04','B-',0);

CREATE TABLE Professor(
    `Professor ID` VARCHAR(10) NOT NULL,
   `First Name` VARCHAR(15),
    `Middle Name` VARCHAR(15),
    `Last Name` VARCHAR(15),
    `Contact No` DECIMAL(10, 0),
   `Office Contact No` DECIMAL(10, 0),
   `Email ID` VARCHAR(20),
   `Office No` INT,

   PRIMARY KEY(`Professor ID`)
);

insert into Professor
values('SEASP001','Anurag',null,'Lakhlani',9809265735,'07922665454','anurag@gmail.com',124);
insert into Professor
values('SEASP012','Rita',null,'Sharma',9234265735,'07921225542','rita@gmail.com',111); insert
into Professor
values('SEASP032','Ashok',null,'Ranade',769265735,'07922663454','ashok@gmail.com',432); insert
into Professor
values('SEASP051','Daxesh',null,'Shah',9409265735,'07912365454','daxesh@gmail.com',876); insert
into Professor
values('SEASP009','Dhaval',null,'Patel',9802135735,'07912344454','dhaval@gmail.com',098);

insert into Professor
values('SEASP011','Anupam',null,'Lakhlani',9879265735,'07922065454','anupam@gmail.com',224); insert
into Professor
values('SEASP112','Ritesh',null,'Sharma',9234765735,'07921220542','ritesh@gmail.com',112); insert
into Professor
values('SEASP132','Amar',null,'Ranade',769267735,'07922663450','amar@gmail.com',422); insert
into Professor
```

```
values('SEASP151','Daxa',null,'Shah',9409265775,'07912365454','daxa@gmail.com',872); insert
into Professor
values('SEASP019','Bharat',null,'Patel',9802995735,'07912340054','bharat@gmail.com',298);

CREATE TABLE Education(
    `Professor ID` VARCHAR(10) NOT NULL,
   `Programme` VARCHAR(20),
    `Year of Passing` INT,
   `University` VARCHAR(50),


   FOREIGN KEY (`Professor ID`) references Professor(`Professor ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

insert into Education values('SEASP001','M.TECH',1981,'University of Chicago'); insert
into Education values('SEASP001','B.TECH',1979,'L.D.');
insert into Education values('SEASP012','M.TECH',1983,'University of San-fransisco');
insert into Education values('SEASP012','B.TECH',1981,'University of Chicago'); insert
into Education values('SEASP032','M.TECH',1969,'University of London'); insert into
Education values('SEASP032','B.TECH',1967,'University of London'); insert into
Education values('SEASP051','M.TECH',1985,'IIT Bombay'); insert into Education
values('SEASP051','B.TECH',1983,'LDRP'); insert into Education
values('SEASP009','B.TECH',1989,'Stanford');
insert into Education values('SEASP009','PHD',1991,'Punjab Technical University');

CREATE TABLE Experience(
    `Professor ID` VARCHAR(10) NOT NULL,
   `Field` VARCHAR(30),
   `No of Years` INT,
   `Company/Institute` VARCHAR(50),

   PRIMARY KEY(`Professor ID`,`Company/Institute`),

   FOREIGN KEY (`Professor ID`) references Professor(`Professor ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);


insert into Experience values('SEASP001','System Designing',6,'Designing Corporation');
insert into Experience values('SEASP001','System Designing',2,'ABC Corporation');
insert into Experience values('SEASP012','Economics',7,'LD arts'); insert into
Experience values('SEASP032','Electronics',10,'XYZ Company'); insert into Experience
values('SEASP032','Electronics',12,'Stanford'); insert into Experience
values('SEASP051','DBMS',12,'Ganpat University'); insert into Experience
values('SEASP009','Probability',8,'University of America');
```

```
CREATE TABLE Semester(
    `Semester No` INT NOT NULL,
   `Name` VARCHAR(10),
   `Year` INT NOT NULL,
   `Start Date` DATE,
   `End Date` DATE,

   PRIMARY KEY(`Semester No`, `Year`)
);


insert into Semester values(1,'Monsoon',2015,'2015/01/08','2015/12/15');
insert into Semester values(1,'Monsoon',2016,'2015/08/03','2015/12/15');
insert into Semester values(2,'Winter',2015,'2015/08/01','2015/12/20'); insert
into Semester values(2,'Winter',2016,'2016/08/01','2016/05/15'); insert into
Semester values(3,'Monsoon',2016,'2016/08/01','2016/12/15'); insert into
Semester values(4,'Winter',2017,'2017/01/01','2017/05/17'); insert into
Semester values(5,'Monsoon',2017,'2017/08/01','2017/12/15'); insert into
Semester values(6,'Winter',2018,'2018/01/01','2018/05/16'); insert into
Semester values(7,'Monsoon',2018,'2018/08/01','2018/12/17'); insert into
Semester values(8,'Winter',2019,'2019/01/01','2019/05/15');

CREATE TABLE Course(
    `Course ID` VARCHAR(10) NOT NULL,
   `Name` VARCHAR(20),
   `Description` VARCHAR(300),
`Credit` REAL,
   `Professor ID` VARCHAR(10),
   `Branch` VARCHAR(10) NOT NULL,
   `Semester No` INT NOT NULL,
   `Year` INT,
   `Type` VARCHAR(20),
   `Semster Hours` REAL,

   PRIMARY KEY(`Course ID`),

   FOREIGN KEY(`Professor ID`) REFERENCES Professor(`Professor ID`)
            ON DELETE CASCADE ON UPDATE CASCADE,

   FOREIGN KEY(`Branch`) REFERENCES Branch(`Branch ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

insert into Course values('SEASC001','ESD','Learn about embedded system and
designing',3,'SEASP001','SEAS01',4,2017,'Core Course',40);
```

```
insert into Course values('SEASC002','Economics',null,3,'SEASP012','SEAS02',3,2016,'Elective',30); insert into Course values('SEASC003','ADC','One can learn about analog and digital communication through this course',4,'SEASP032','SEAS01',4,2017,'Core Course',40);
insert into Course values('SEASC004','DBMS','One can learn about database management system through this course',3,'SEASP051','SEAS01',4,2017,'Core Course',40);
insert into Course values('SEASC005','PRP','One can learn about concepts of probability through this course',3,'SEASP009','SEAS01',4,2017,'Core Course',30);

insert into Course values('SEASC006','ICP','Learn about introduction to computing programming',3,'SEASP011','SEAS01',4,2017,'Core Course',40);
insert into Course values('SEASC007','Economics-II',null,3,'SEASP012','SEAS02',3,2016,'Elective',30); insert into Course values('SEASC008','OOP','One can learn about analog and digital communication through this course',4,'SEASP032','SEAS01',4,2017,'Core Course',40);

CREATE TABLE Attendance(
    `Student ID` VARCHAR(10),
   `Course ID` VARCHAR(10),
   `Month` VARCHAR(15),
   `Total Days` INT,
   `Days Attended` INT,

   PRIMARY KEY(`Student ID`, `Course ID`,`month`),

   FOREIGN KEY (`Student ID`) references Student(`Student ID`)
            ON DELETE CASCADE ON UPDATE CASCADE,

    FOREIGN KEY (`Course ID`) references Course(`Course ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);


insert into Attendance values(201501008,'SEASC001','January',25,25);
insert into Attendance values(201501008,'SEASC001','February',23,22);
insert into Attendance values(201501008,'SEASC001','March',26,20); insert into Attendance values(201501008,'SEASC001','April',27,26);

insert into Attendance values(201501051,'SEASC001','January',25,25);
insert into Attendance values(201501051,'SEASC001','February',23,15);
insert into Attendance values(201501051,'SEASC001','March',26,25); insert into Attendance values(201501051,'SEASC001','April',27,20);

CREATE TABLE `Summer Project`(
    `Instructor ID` VARCHAR(10) NOT NULL,
   `Instructor Name` VARCHAR(20),
   `Title` VARCHAR(20),
   `Description` VARCHAR(100),
```

```
    `Year` INT,
    `Duration` VARCHAR(10),
    `Stipend` REAL,

    FOREIGN KEY (`Instructor ID`) references Professor(`Professor ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);


insert into `Summer Project` values('SEASP009','Dhaval Patel','WiVi','See through wall using wifi
signals',2017,40,null);
insert into `Summer Project` values('SEASP009','Dhaval Patel','V-to-V','Vehicular communicaion',2017,40,2000);
insert into `Summer Project` values('SEASP012','Rita Sharma','Women_empowerment','Research on woment
empowerment in India',2017,30,null);
insert into `Summer Project` values('SEASP012','Rita Sharma','Chinese_degrowth','Research on
chinese',2017,30,null);

CREATE TABLE `Extra Course`(
     `Course ID` VARCHAR(10) NOT NULL,
     `Instructor Name` VARCHAR(20),
    `Name` VARCHAR(70),
    `Description` VARCHAR(200),
    `Year` INT,
    `Fees` REAL,

    PRIMARY KEY(`Course ID`)
);

insert into `Extra Course` values('SEASEC001','Dhaval Patel','Bhagwat Geeta',null,2017,null); insert into
`Extra Course` values('SEASEC002','Dhaval Patel','Reader, Writer and Text','the workshop will enhance
your reading and writing skills',2017,3000);
insert into `Extra Course` values('SEASEC003','Ashok Ranade','Inside Music',null,2017,1000); insert
into `Extra Course` values('SEASEC004','Rita Sharma','Accounts',null,2017,null);


CREATE TABLE `E-Learning`(
     `Course ID` VARCHAR(10) NOT NULL,
    `Name` VARCHAR(20),
    `Description` VARCHAR(60),
    `URL` VARCHAR(100),
    `Rating` VARCHAR(10),

     FOREIGN KEY (`Course ID`) references Course(`Course ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
insert into `E-Learning` values('SEASC004','Code Cheff','competitive coding','https://www.codechef.com',4.5);
insert into `E-Learning` values('SEASC004','HackerEarth','competitive coding','https://www.hackerearth.com',3.5);
insert into `E-Learning` values('SEASC004','HackerRank','competitive
coding','https://www.hackerrank.com',3); insert into `E-Learning`
values('SEASC004','Tutorial
Points',null,'https://www.tutorialspoint.com/dbms/',4.5);
insert into `E-Learning` values('SEASC001','Radio Electronics',null,'http://www.radio-electronics.com',3.5); insert
into `E-Learning` values('SEASC004','MIT PRP Lactures','this is a very good video lactures by the instructor of
MIT','https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/',4);


CREATE TABLE `Book Store`(
    `Store ID` VARCHAR(10) NOT NULL,
   `Shopkepeer Name` VARCHAR(20),
   `Contact No` DECIMAL(10, 0),
   `Opening Time` varchar(15),
    `Closing Time` varchar(15),
    `Returning Policy(%)` varchar(15),

   PRIMARY KEY(`Store ID`)
);

INSERT INTO `Book Store` VALUES ('B_S_011','Ramesh Patel',9874563211,'10:30 AM','7:00 PM','45%');
INSERT INTO `Book Store` VALUES ('B_S_051','Ramesh Shah',9877763211,'9:30 AM','7:30 PM','65%');
INSERT INTO `Book Store` VALUES ('B_S_001','Raghav Patel',9974563210,'9:30 AM','8:00 PM','45%');
INSERT INTO `Book Store` VALUES ('B_S_055','Amar Shukla',9874763200,'8:00 AM','5:00 PM','50%');
INSERT INTO `Book Store` VALUES ('B_S_007','Chandan Patel',8784563211,'10:00 AM','7:00 PM','60%');


CREATE TABLE Book(
    `Course ID` VARCHAR(10) NOT NULL,
    `ISBN No` VARCHAR(12),
   `Book ID` VARCHAR(10) NOT NULL,
   `Title` VARCHAR(50),
   `Edition` INT,
   `Publisher` VARCHAR(30),
   `Rating` VARCHAR(10),
   `Soft Copy URL` VARCHAR(100),
   `Library Availability Status` CHAR,
   `Available In (Book Store)` VARCHAR(10),
   `Price` REAL,

   PRIMARY KEY(`Book ID`, `ISBN No`),
```

```
   FOREIGN KEY (`Course ID`) references Course(`Course ID`)
            ON DELETE CASCADE ON UPDATE CASCADE,

   FOREIGN KEY (`Available In (Book Store)`) references `Book Store`(`Store ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

INSERT INTO `Book` VALUES ('SEASC004','ISBNB001','B001','DBMS in Easy Way!',4,'Mextell
Publisher',3.5,'https://drive.google.com/open?id=0BxbDnO3BJ0BddHMQzA','Y','B_S_007',450);
INSERT INTO `Book` VALUES ('SEASC004','ISBNB001','B002','DBMS in Easy Way!',4,'Mextell
Publisher',3.5,'https://drive.google.com/open?id=0BxbDnO3BJ0BddHMQzA','Y','B_S_007',450);
INSERT INTO `Book` VALUES ('SEASC004','ISBNB004','B004','DBMS in Easy Way!',3,'Mextell
Publisher',3.0,'https://drive.google.com/open?id=0BxbdcsDnO3BJ0BddHMQzA','Y',null,400);
INSERT INTO `Book` VALUES ('SEASC004','ISBNB077','B088','Learn DBMS',1,'Loyal+
Publisher',4.5,'https://drive.google.com/open?id=MQzA','N','B_S_007',700);



CREATE TABLE Author(
    `Book ID` VARCHAR(10) NOT NULL,
   `First Name` VARCHAR(15),
   `Middle Name` VARCHAR(15),
   `Last Name` VARCHAR(15),

    FOREIGN KEY (`Book ID`) references Book(`Book ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

insert into Author values('B001','Shrinivasan',null,'Iyer');
insert into Author values('B001','Smith','William','Loy'); insert
into Author values('B001','Mex',null,'Frost');

insert into Author values('B002','R.D.',null,'Sharma'); insert
into Author values('B004','Robert','Bill','Frost'); insert into
Author values('B088','Amartya',null,'Sen');

CREATE TABLE `Seminar/Talk`(
    `Semester No` INT NOT NULL,
   `Year` INT NOT NULL,
   `Speaker` VARCHAR(30),
   `Topic` VARCHAR(70),
   `Date` DATE,
   `Time` varchar(15),
```

```
    `Venue` VARCHAR(20),

    FOREIGN KEY (`Semester No`) references Semester(`Semester No`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

insert into `Seminar/Talk` values('4',2017,'Mr. H. K. Rao','Data Science','2017/04/14','2:00 pm','Auditoriam');
insert into `Seminar/Talk` values('6',2017,'Dr. Mukesh Mishra','Designing','2017/02/22','3:00 pm','Room no: 116');
insert into `Seminar/Talk` values('2',2017,'Dr. Mukesh Paramar','User Centered Designing','2017/02/22','3:00 pm','Room no: 117');


CREATE TABLE `Upcoming event`(

    `Year` INT NOT NULL,
    `Name` VARCHAR(50),
    `Description` VARCHAR(80),
    `Start Date` DATE,
    `End Date` DATE
);

insert into `Upcoming event` values(2017,'Agaaz-17','The cultural festival of SEAS!','2017/02/14','2017/02/18');
insert into `Upcoming event` values(2016,'Ghummar-16','Garaba night at SEAS','2016/10/22','2016/10/22'); insert into `Upcoming event` values(2017,'Music Concert','first ever concert of SEAS by Astitva the bend','2017/04/22','2017/04/22');

CREATE TABLE Holiday(

    `Year` INT NOT NULL,
     `Festival Name` VARCHAR(20),
    `Date` varchar(20)
);

insert into Holiday values(2017,'Republic day','January 26');
insert into Holiday values(2017,'Sankranti','January 14'); insert
into Holiday values(2017,'Maha Shivaratri','February 13'); insert
into Holiday values(2017,'Holi','March 02');
insert into Holiday values(2017,'Independence day','Auguest 15');


CREATE TABLE Exam(
     `Exam ID` VARCHAR(10) NOT NULL,
    `Term` VARCHAR(15),
    `Start Date` DATE NOT NULL,
```

```
    `End Date` DATE NOT NULL,
    `Time` varchar(30),

    PRIMARY KEY(`Exam ID`)
);

insert into Exam values('SEASE001','mid','2017/02/10','2017/02/15','10:00 am - 1:00 pm'); insert
into Exam values('SEASE002','end','2017/05/10','2017/05/22','10:00 am - 1:00 pm'); insert into
Exam values('SEASE022','mid','2016/05/10','2016/05/22','10:30 am - 1:30 pm'); insert into
Exam values('SEASE023','end','2016/05/10','2016/05/22','10:30 am - 1:30 pm');

CREATE TABLE `Supplymentry Exam`(
     `Exam ID` VARCHAR(10) NOT NULL,
    `Term` VARCHAR(15),
    `Student ID` VARCHAR(10),
    `Course ID` VARCHAR(10),
    `Date` DATE,
    `Result` CHAR,

    PRIMARY KEY(`Exam ID`),

    FOREIGN KEY (`Exam ID`) references Exam(`Exam ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

insert into `Supplymentry Exam` values('SEASE001','mid','201401021','SEASC001','2017/06/15','F');
insert into `Supplymentry Exam` values('SEASE002','end','201401021','SEASC002','2017/06/20','F');
insert into `Supplymentry Exam` values('SEASE022','mid','201401021','SEASC004','2017/06/22','F');

CREATE TABLE Result(
     `Student ID` VARCHAR(10) NOT NULL,
    `Course ID` VARCHAR(10) NOT NULL,
    `Semester` INT NOT NULL,
    `CGPA` REAL,
    `Grade` VARCHAR(2),

    PRIMARY KEY(`Student ID`, `Course ID`),

    FOREIGN KEY (`Student ID`) references Student(`Student ID`)
            ON DELETE CASCADE ON UPDATE CASCADE,

     FOREIGN KEY (`Course ID`) references Course(`Course ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
insert into Result values('201401021','SEASC001',4,2.02,'B'); insert
into Result values('201401021','SEASC002',4,2.02,'A'); insert into
Result values('201401021','SEASC003',4,2.02,'B+'); insert into
Result values('201401021','SEASC004',4,2.02,'B+'); insert into
Result values('201401021','SEASC005',4,2.02,'B-');

insert into Result values('201501008','SEASC001',4,3.02,'A+');
insert into Result values('201501008','SEASC002',4,3.02,'A'); insert
into Result values('201501008','SEASC003',4,3.02,'B+'); insert into
Result values('201501008','SEASC004',4,3.02,'B+'); insert into
Result values('201501008','SEASC005',4,3.02,'A-');

insert into Result values('201501051','SEASC001',4,2.22,'A'); insert
into Result values('201501051','SEASC002',4,2.22,'A-'); insert into
Result values('201501051','SEASC003',4,2.22,'B+'); insert into
Result values('201501051','SEASC004',4,2.22,'B+'); insert into
Result values('201501051','SEASC005',4,2.22,'B-');

insert into Result values('201301015','SEASC001',4,3.22,'A'); insert
into Result values('201301015','SEASC002',4,3.22,'A-'); insert into
Result values('201301015','SEASC003',4,3.22,'B+'); insert into
Result values('201301015','SEASC004',4,3.22,'A'); insert into
Result values('201301015','SEASC005',4,3.22,'A-');

CREATE TABLE Internship(
    `Company ID` VARCHAR(10) NOT NULL,
   `Company Name` VARCHAR(20),
   `Year` INT,
   `HR Name` VARCHAR(20),
   `Contact No` DECIMAL(10, 0),
   `Profile` VARCHAR(30),
   `Stipend(per month)` REAL,

  PRIMARY KEY(`Company ID`)
);

INSERT INTO `Internship` VALUES ('IS2017001','Google Inc.',2017,'Ujjvala Britto',9874564560,'STEP
Intern',30000);
INSERT INTO `Internship` VALUES ('IS2017002','Google Inc.',2017,'Sharmala
Huyei',9993840091,'Software Intern',35000);
INSERT INTO `Internship` VALUES ('IS2016001','Amazon',2016,'Punam Shah',9877850012,'SDE
Intern',20000);
INSERT INTO `Internship` VALUES ('IS2016041','Tatva Soft',2016,'Rakesh Shah',7417412301,'Web
Developer',null);
INSERT INTO `Internship` VALUES ('IS2015004','Tatva Soft',2015,'Gavrav Patel',7456321522,'Web
Developer',null);
```

```
CREATE TABLE Prerequirnment(
    `Company ID` VARCHAR(10) NOT NULL,
   `Requirment` VARCHAR(40),

   FOREIGN KEY (`Company ID`) references Internship(`Company ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

INSERT INTO `Prerequirnment` VALUES('IS2017001','Data Structure and Algorithm');
INSERT INTO `Prerequirnment` VALUES('IS2017001','DBMS');
INSERT INTO `Prerequirnment` VALUES('IS2017001','C');
INSERT INTO `Prerequirnment` VALUES('IS2017001','C++');
INSERT INTO `Prerequirnment` VALUES('IS2015004','Data Structure and Algorithm');
INSERT INTO `Prerequirnment` VALUES('IS2015004','PHP');
INSERT INTO `Prerequirnment` VALUES('IS2015004','JAVA Script');
INSERT INTO `Prerequirnment` VALUES('IS2016001','Data Structure');
INSERT INTO `Prerequirnment` VALUES('IS2016001','DBMS');
INSERT INTO `Prerequirnment` VALUES('IS2016001','PHP');


CREATE TABLE `Student Address`(
    `Student ID` VARCHAR(10) NOT NULL,
   `Street` VARCHAR(20),
   `City` VARCHAR(20),
   `State` VARCHAR(15),
   `PIN Code` DECIMAL(6, 0),

   FOREIGN KEY (`Student ID`) references Student(`Student ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

INSERT INTO `Student Address` VALUES ('201401021','Navarngpura','Ahmedabad','Gujarat',380009);
INSERT INTO `Student Address` VALUES ('201501008','Manasi Circle','Navasari','Gujarat',340078);
INSERT INTO `Student Address` VALUES ('201501051','Ashram Road','Ahmedabad','Gujarat',380012);
INSERT INTO `Student Address` VALUES ('201301015','Sadhu Vasvani
Road','Rajakot','Gujarat',384501);


CREATE TABLE `Professor Address`(
    `Professor ID` VARCHAR(10) NOT NULL,
   `Street` VARCHAR(20),
   `City` VARCHAR(20),
   `State` VARCHAR(15),
```

```
    `PIN Code` DECIMAL(6, 0),

    FOREIGN KEY (`Professor ID`) references Professor(`Professor ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

INSERT INTO `Professor Address` VALUES
('SEASP001','Navarngpura','Ahmedabad','Gujarat',380009);
INSERT INTO `Professor Address` VALUES ('SEASP012','Navarngpura','Navasari','Gujarat',340009);
INSERT INTO `Professor Address` VALUES ('SEASP032','Ashram
Road','Ahmedabad','Gujarat',380012);
INSERT INTO `Professor Address` VALUES ('SEASP051','Sadhu Vasvani
Road','Rajakot','Gujarat',384501);
INSERT INTO `Professor Address` VALUES ('SEASP009','University
Road','Delvada','Rajasthan',774009);


CREATE TABLE `Book Store Address`(
    `Store ID` VARCHAR(10) NOT NULL,
    `Street` VARCHAR(20),
    `City` VARCHAR(20),
    `State` VARCHAR(15),
    `PIN Code` DECIMAL(6, 0),

    FOREIGN KEY (`Store ID`) references `Book Store`(`Store ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

INSERT INTO `Book Store Address` VALUES
('B_S_011','Navarngpura','Ahmedabad','Gujarat',380009);
INSERT INTO `Book Store Address` VALUES
('B_S_051','Navarngpura','Ahmedabad','Gujarat',380009);
INSERT INTO `Book Store Address` VALUES ('B_S_001','Ashram
Road','Ahmedabad','Gujarat',380012);
INSERT INTO `Book Store Address` VALUES ('B_S_055','Sadhu Vasvani
Road','Rajakot','Gujarat',384501);
INSERT INTO `Book Store Address` VALUES ('B_S_007','Indira Circle','Rajakot','Gujarat',384009);


CREATE TABLE `Company Address`(
    `Company ID` VARCHAR(10) NOT NULL,
    `Street` VARCHAR(20),
    `City` VARCHAR(20),
    `State` VARCHAR(15),
```

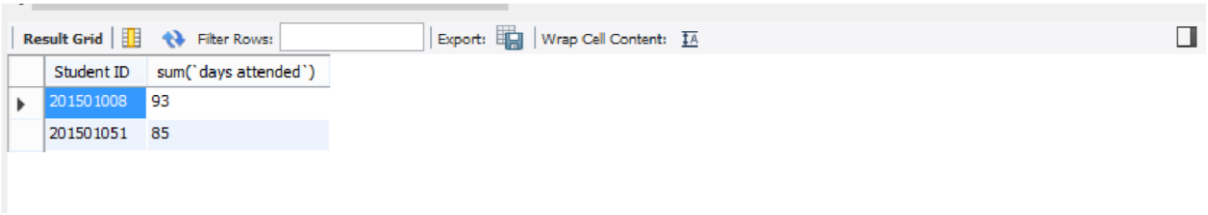
```
    `PIN Code` DECIMAL(6, 0),

   FOREIGN KEY (`Company ID`) references Internship(`Company ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

INSERT INTO `Company Address` VALUES ('IS2017001','Indira Circle','Rajakot','Gujarat',384009);
INSERT INTO `Company Address` VALUES ('IS2017002','Indira Circle','Rajakot','Gujarat',384009);
INSERT INTO `Company Address` VALUES ('IS2016001','Ramji road','Hinjewadi','Puna',384009);
INSERT INTO `Company Address` VALUES ('IS2016041','Mahatma
Circle','Hyderabad','Telangana',384009);
INSERT INTO `Company Address` VALUES ('IS2015004','Indira road','Ahmedabad','Gujarat',384009);

CREATE TABLE `Student_Professor`(
    `Student ID` VARCHAR(10) NOT NULL,
   `Professor ID` VARCHAR(10) NOT NULL,

    PRIMARY KEY(`Student ID`, `Professor ID`),

   FOREIGN KEY (`Student ID`) references Student(`Student ID`)
            ON DELETE CASCADE ON UPDATE CASCADE,

    FOREIGN KEY (`Professor ID`) references Professor(`Professor ID`)
            ON DELETE CASCADE ON UPDATE CASCADE
);

INSERT INTO `Student_Professor` VALUES (201501051,'SEASP001');
INSERT INTO `Student_Professor` VALUES (201501051,'SEASP009');
INSERT INTO `Student_Professor` VALUES (201501051,'SEASP012');
INSERT INTO `Student_Professor` VALUES (201501051,'SEASP032'); INSERT
INTO `Student_Professor` VALUES (201501051,'SEASP051');

INSERT INTO `Student_Professor` VALUES (201401021,'SEASP001');
INSERT INTO `Student_Professor` VALUES (201401021,'SEASP009');
INSERT INTO `Student_Professor` VALUES (201401021,'SEASP012');
INSERT INTO `Student_Professor` VALUES (201401021,'SEASP032'); INSERT
INTO `Student_Professor` VALUES (201401021,'SEASP051');

INSERT INTO `Student_Professor` VALUES (201501008,'SEASP001');
INSERT INTO `Student_Professor` VALUES (201501008,'SEASP009');
INSERT INTO `Student_Professor` VALUES (201501008,'SEASP012');
INSERT INTO `Student_Professor` VALUES (201501008,'SEASP032');
INSERT INTO `Student_Professor` VALUES (201501008,'SEASP051');
```

```
CREATE TABLE `Professor_Course`(
    `Course ID` VARCHAR(10) NOT NULL,
   `Professor ID` VARCHAR(10) NOT NULL,

    PRIMARY KEY(`Course ID`, `Professor ID`),

   FOREIGN KEY (`Course ID`) references Course(`Course ID`)
            ON DELETE CASCADE ON UPDATE CASCADE,

    FOREIGN KEY (`Professor ID`) references Professor(`Professor ID`)
            ON DELETE CASCADE ON UPDATE CASCADE

);

insert into `Professor_Course` values('SEASC001','SEASP001'); insert
into `Professor_Course` values('SEASC002','SEASP012'); insert into
`Professor_Course` values('SEASC003','SEASP032'); insert into
`Professor_Course` values('SEASC004','SEASP051'); insert into
`Professor_Course` values('SEASC005','SEASP009');

CREATE TABLE `Student_Internship`(
    `Student ID` VARCHAR(10) NOT NULL,
   `Company ID` VARCHAR(10) NOT NULL,

    PRIMARY KEY(`Student ID`, `Company ID`),

   FOREIGN KEY (`Student ID`) references Student(`Student ID`)
            ON DELETE CASCADE ON UPDATE CASCADE,

 FOREIGN KEY (`Company ID`) references Internship(`Company ID`)
            ON DELETE CASCADE ON UPDATE CASCADE

);

insert into `Student_Internship` values('201401021','IS2017001'); insert
into `Student_Internship` values('201401021','IS2016001'); insert into
`Student_Internship` values('201501008','IS2017001'); insert into
`Student_Internship` values('201501051','IS2016041');

CREATE TABLE `Student_Summer Project`(
    `Student ID` VARCHAR(10) NOT NULL,
   `Instructor ID` VARCHAR(10) NOT NULL,

    PRIMARY KEY(`Student ID`, `Instructor ID`),
```

```
    FOREIGN KEY (`Student ID`) references Student(`Student ID`)
            ON DELETE CASCADE ON UPDATE CASCADE,

    FOREIGN KEY (`Instructor ID`) references Professor(`Professor ID`)
            ON DELETE CASCADE ON UPDATE CASCADE

);

insert into `Student_Summer Project` values(201401021,'SEASP009'); insert
into `Student_Summer Project` values(201401021,'SEASP012'); insert into
`Student_Summer Project` values(201501008,'SEASP009'); insert into
`Student_Summer Project` values(201501008,'SEASP012'); insert into
`Student_Summer Project` values(201301015,'SEASP009');


                CREATE TABLE `Student_Extra Course`(
                    `Student ID` VARCHAR(10) NOT NULL,
                  `Course ID` VARCHAR(10) NOT NULL,

                    PRIMARY KEY(`Student ID`, `Course ID`),

                  FOREIGN KEY (`Student ID`) references Student(`Student ID`)
                            ON DELETE CASCADE ON UPDATE CASCADE,

                    FOREIGN KEY (`Course ID`) references `Extra Course`(`Course ID`)
                            ON DELETE CASCADE ON UPDATE CASCADE

                );

              insert into `Student_Extra Course` values(201501051,'SEASEC001'); insert
               into `Student_Extra Course` values(201501051,'SEASEC002'); insert into
                 `Student_Extra Course` values(201301015,'SEASEC003'); insert into
                 `Student_Extra Course` values(201301015,'SEASEC001'); insert into
                 `Student_Extra Course` values(201401021,'SEASEC002'); insert into
                       `Student_Extra Course` values(201401021,'SEASEC004');
```

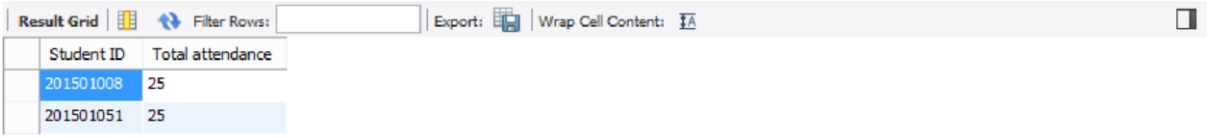
Queries and Output

Query: 1- Total Attendance of students for course 'X'. SELECT
`Attendance`.`Student ID`, sum(`days attended`) from
Attendance
where `Attendance`.`Course ID` = 'SEASC001' group
by (`Attendance`.`Student ID`) ;



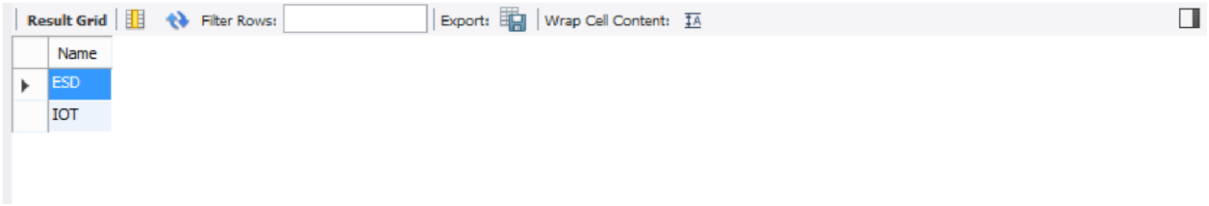
Student ID	sum(`days attended`)
201501008	93
201501051	85

Query: 2- Total attendance of students for course 'X' and month 'Y'.
SELECT `Attendance`.`Student ID`,sum(`days attended`) as `Total attendance` from
Attendance
where Attendance.`Course ID` = 'SEASC001' AND Attendance.`Month` = 'January' group
by(Attendance.`Student ID`);



Student ID	Total attendance
201501008	25
201501051	25

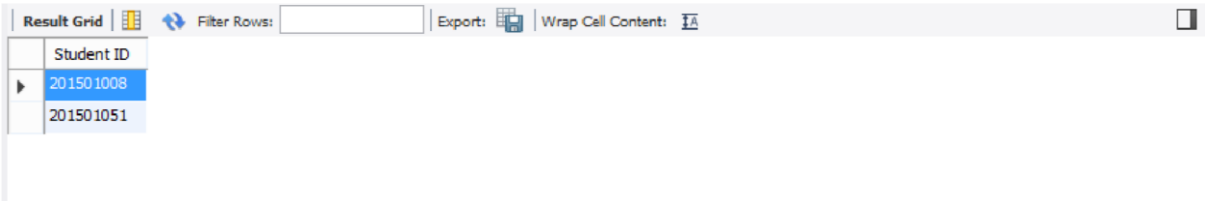
Query: 3- Name of courses taken by professor 'X' in winter. SELECT
`Course`.`Name`
from Course
where Course.`Professor ID` = 'SEASP001' AND (Course.`Semester No` = 2 OR
Course.`Semester No` = 4 OR Course.`Semester No` = 6 OR Course.`Semester No` = 8) AND
Course.`Year` = 2017;



Name
ESD
IOT

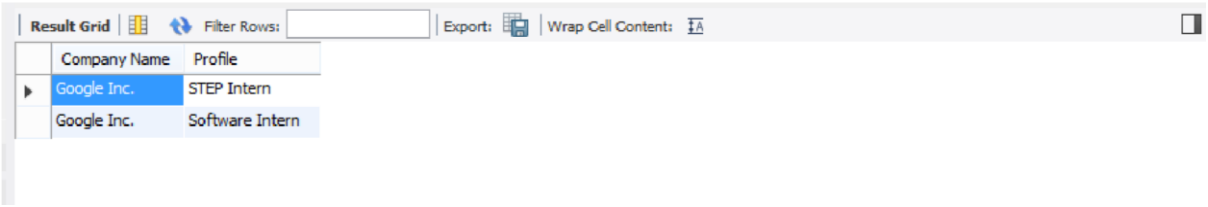
Query: 4- List of students who failed in course ‘X’ and are in branch ‘Y’.

```
SELECT `Student`.`Student ID`
FROM Student,Result
WHERE `Result`.`Grade` = 'F' AND `Result`.`Course ID` = 'SEASC005' AND
`Student`.`Branch` = 'SEAS01' AND (`Student`.`Student ID` = `Result`.`Student ID`);
```



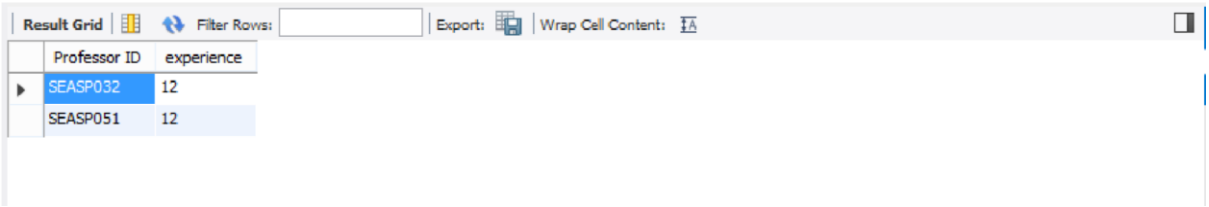
Query: 5- List details of companies (name, profile) who provide stipend more than 3000, located in city ‘X’ and visited in year ‘Y’.

```
SELECT `Internship`.`Company Name`, `Internship`.`Profile`
FROM Internship, `Company Address`
WHERE `Internship`.`stipend(per month)` > 3000 AND `Internship`.`Year` = 2017 AND
`Company Address`.`City` = 'Rajakot' AND `Internship`.`company id` = `Company
Address`.`company id`;
```



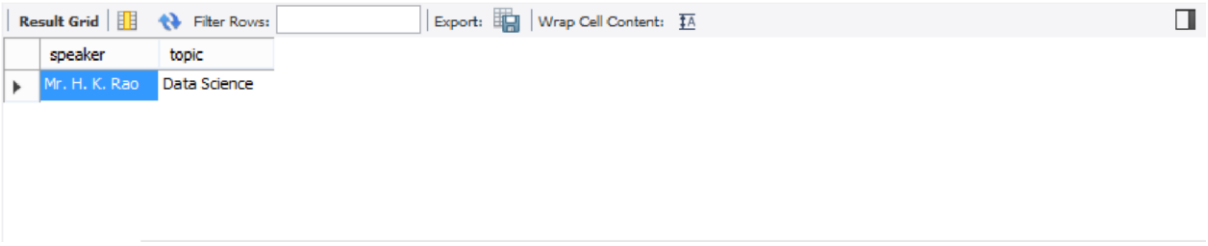
Query: 6- List of professors who have experience more than 10 years.

```
SELECT `Experience`.`Professor ID`, sum(`no of years`) as experience  from
Experience,Professor
WHERE `Experience`.`Professor ID` = `Professor`.`Professor ID` And `No of years` > 10 group
by (`Experience`.`Professor ID`);
```

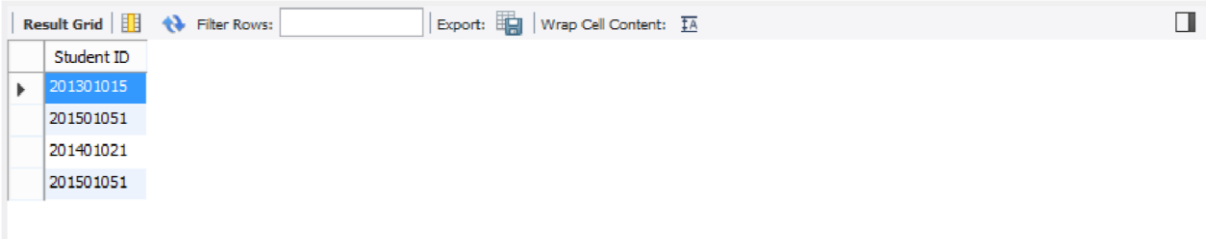


Query: 7- Seminar/ Talk details on date ‘X’.

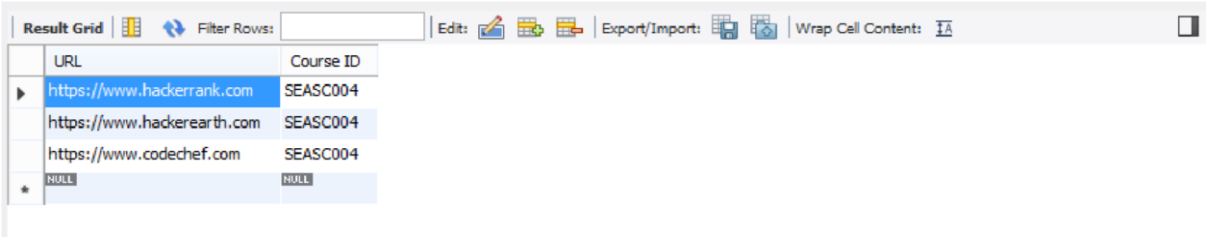
```
SELECT `Seminar/Talk`.`speaker`,`Seminar/Talk`.`topic` from
`Seminar/Talk`
where `Seminar/Talk`.`Date` = '2017/04/14';
```



Query: 8- List of students who have taken extra course under professor 'X'.
SELECT `Student_Extra Course`.`Student ID` from `Extra Course`,`Student_Extra Course`
WHERE `Extra Course`.`Course ID` = `Student_Extra Course`.`Course ID` AND `Extra Course`.`Instructor Name` = 'Dhaval Patel';



Query: 9- List of websites for 'Competitive Coding' in descending order of their rating.
SELECT `E-learning`.`URL`, `E-learning`.`Course ID` from `E-learning`
WHERE `E-learning`.`Description` = 'competitive coding'
ORDER BY (`E-learning`.`rating`) asc;



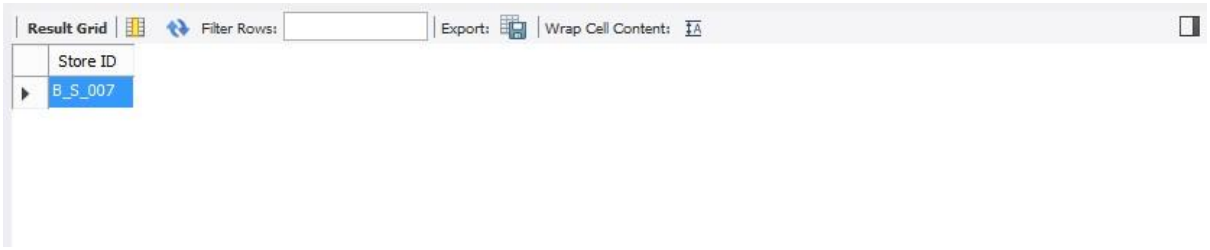
Query: 10- Softcopy URL of a book suggested for course of branch 'X'. SELECT `Book`.`Soft Copy URL`
from `Book`, `Course`
WHERE `Course`.`Branch` = 'SEAS01'
AND `Book`.`Course ID` = `Course`.`Course ID`;

Query: 11- Total number of books available in library.

```
SELECT count(distinct `Book ID`) as 'NO OF BOOKS' from
book
WHERE `Book`.`Library Availability Status` = 'Y';
```

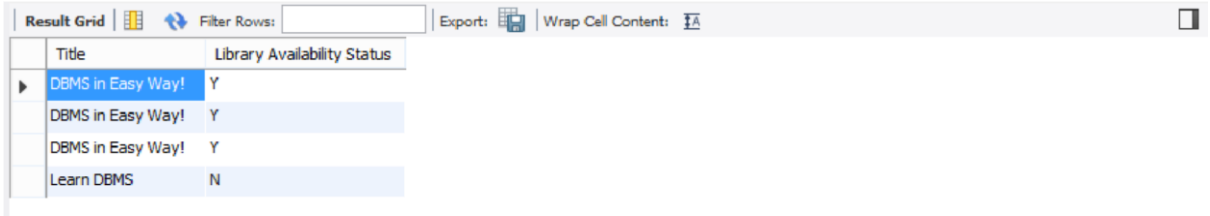
Query: 12- List of stores which are located in city 'X' and books of course 'Y' are available there.

```
SELECT distinct `Book store`.`Store ID`
from `Book Store Address`,`Book`,`Book store`
WHERE `Book store`.`Store ID` = `Book`.`Available In (Book Store)` AND `Book Store
Address`.`City` = 'Ahmedabad' AND `Book`.`Course ID` = 'SEASC004';
```



Query: 13- Book Details (title, library status) of semester 'X' and branch 'Y'.

```
SELECT `Book`.`Title`,`Book`.`Library Availability Status` from Book
, Course
WHERE `Course`.`Semester No` = '4' AND `Course`.`Branch` = 'SEAS01' AND
`Book`.`Course ID` = `Course`.`Course ID` ;
```

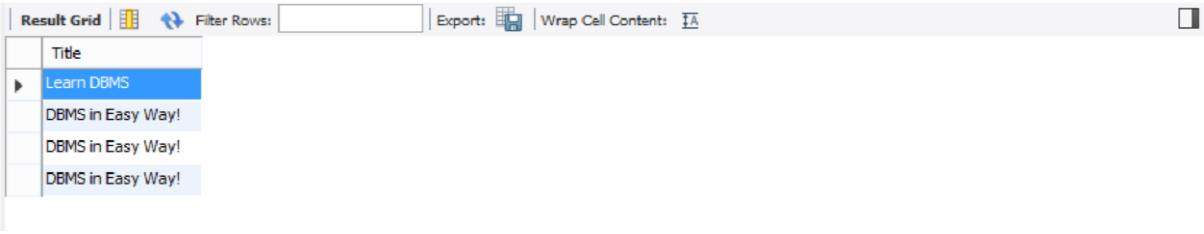


Query: 14- Top 3 Students of course 'X'.

```
SELECT `Student`.`First Name`, `Student`.`Middle Name`, `Student`.`Last Name` from
Student , Result
WHERE `Student`.`Student ID` = `Result`.`Student ID` AND `Result`.`Course ID` = 'SEASC001'
ORDER BY (`Result`.`Grade`) DESC LIMIT 3;
```

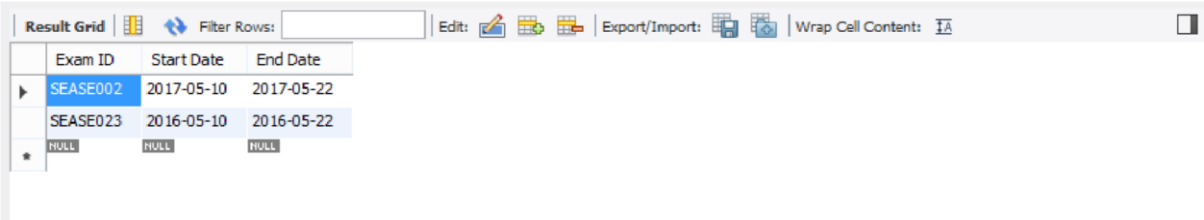
First Name	Middle Name	Last Name
Charmi	Hiren	Chokshi
Ashna	NULL	Jain
Dhruti	Siddharth	Chandarana

Query: 15- List of books of course 'X' in descending order of their price. SELECT
`Book`.`Title`
from Book
where `Book`.`Course ID` = 'SEASC004' ORDER
BY (`Book`.`Price`) DESC ;



Title
Learn DBMS
DBMS in Easy Way!
DBMS in Easy Way!
DBMS in Easy Way!

Query: 16- Details of all end semester exams.
SELECT `Exam`.`Exam ID`, `Exam`.`Start Date`, `Exam`.`End Date` FROM EXAM WHERE
`Exam`.`Term` = 'End';



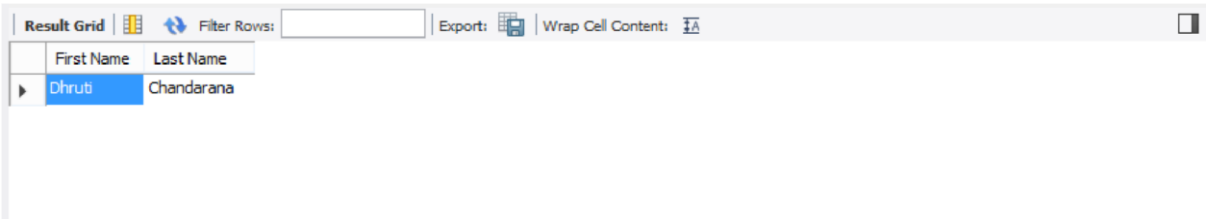
Exam ID	Start Date	End Date
SEASE002	2017-05-10	2017-05-22
SEASE023	2016-05-10	2016-05-22
NULL	NULL	NULL

Query: 17- List of students who got grade 'X' in course 'Y'.
SELECT `Result`.`Student ID`, `Result`.`Grade` FROM Result
WHERE `Result`.`Course ID` = 'SEASC004' AND `Result`.`Grade` = 'A';

Student ID	Grade
201301015	A

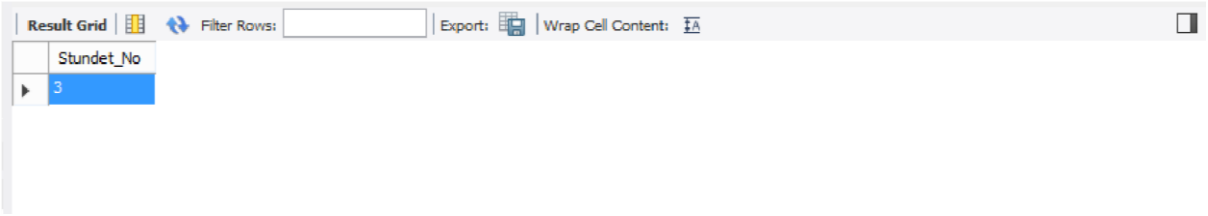
Query: 17- List of students who are living in the same city where company 'X' is located.
SELECT `Student`.`First Name`, `Student`.`Last Name` FROM Student
WHERE `Student`.`Student ID` = (
SELECT `Student Address`.`Student ID` FROM `Student Address`
WHERE `Student Address`.city =

(SELECT `Company Address`.`city` FROM `Company Address`
WHERE `Company Address`.`Company ID` = 'IS2017001'));



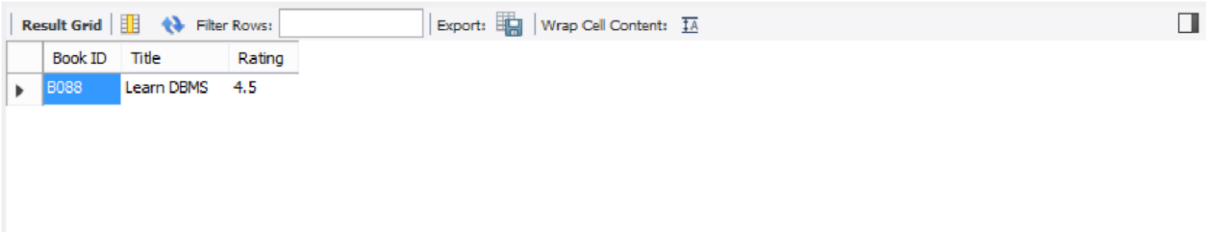
Query: 18- Number of students who have taken summer project under professor 'X'.

SELECT count(`Student_Summer Project`.`Student ID`) AS Stundet_No FROM
`Student_summer project`
WHERE `Student_Summer Project`.`Instructor ID` = 'SEASP009';



Query: 19- List of books which are not available in library of course 'X' in descending order of their rating.

SELECT `Book`.`Book ID`, `Book`.`Title`, `Book`.`Rating` FROM Book
WHERE `Book`.`Course ID` = 'SEASC004' AND `Book`.`Library Availability Status` = 'N'
ORDER BY (`Book`.`Rating`) DESC;



Query: 20- List of seminar/ talk details for in ascending order of semester.

SELECT distinct Semester.`Semester No`,`Seminar/Talk`.`Date`,`Seminar/Talk`.`Speaker` ,
`Upcoming event`.`Name` as 'Upcoming event',`Upcoming event`.`Description`
from Semester,`Seminar/Talk`,Holiday,Exam,`Upcoming event` Where
Semester.`Semester No` = `Seminar/Talk`.`Semester No` order
by(Semester.`Semester No`) ASC;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Semester No	Date	Speaker	Upcoming event	Description
2	2017-02-22	Dr. Mukesh Paramar	Ghummar-16	Garaba night at SEAS
2	2017-02-22	Dr. Mukesh Paramar	Agaaz-17	The cultural festival of SEAS!
2	2017-02-22	Dr. Mukesh Paramar	Music Concert	first ever concert of SEAS by Astitva the bend
4	2017-04-14	Mr. H. K. Rao	Music Concert	first ever concert of SEAS by Astitva the bend
4	2017-04-14	Mr. H. K. Rao	Ghummar-16	Garaba night at SEAS
4	2017-04-14	Mr. H. K. Rao	Agaaz-17	The cultural festival of SEAS!
6	2017-02-22	Dr. Mukesh Mishra	Ghummar-16	Garaba night at SEAS
6	2017-02-22	Dr. Mukesh Mishra	Agaaz-17	The cultural festival of SEAS!
6	2017-02-22	Dr. Mukesh Mishra	Music Concert	first ever concert of SEAS by Astitva the bend

Query: 21- List of companies which are located in city 'X' in ascending order of their visited year.
SELECT distinct Internship.`company ID`,`Company Name`,`Requirment`,Internship.`Year` from Internship,Prerequirnment,`Company address` where `Company address`.`city` = 'Ahmedabad' order by(Year) ASC;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

company ID	Company Name	Requirment	Year
IS2015004	Tatva Soft	JAVA Script	2015
IS2015004	Tatva Soft	Data Structure and Algorithm	2015
IS2015004	Tatva Soft	C++	2015
IS2015004	Tatva Soft	Data Structure	2015
IS2015004	Tatva Soft	DBMS	2015
IS2015004	Tatva Soft	PHP	2015
IS2015004	Tatva Soft	C	2015
IS2016001	Amazon	Data Structure and Algorithm	2016
IS2016041	Tatva Soft	C++	2016
IS2016001	Amazon	C++	2016
IS2016041	Tatva Soft	Data Structure	2016
IS2016041	Tatva Soft	DBMS	2016
IS2016001	Amazon	Data Structure	2016
IS2016001	Amazon	DBMS	2016
IS2016041	Tatva Soft	PHP	2016
IS2016001	Amazon	PHP	2016

Query: 22- Result of students in course 'X'.
SELECT Result.`Student ID` , Result.`Course ID` , Result.grade from Result , Student
where Result.`Student ID` = Student.`Student ID` and Result.`Course ID` = 'SEASC002';

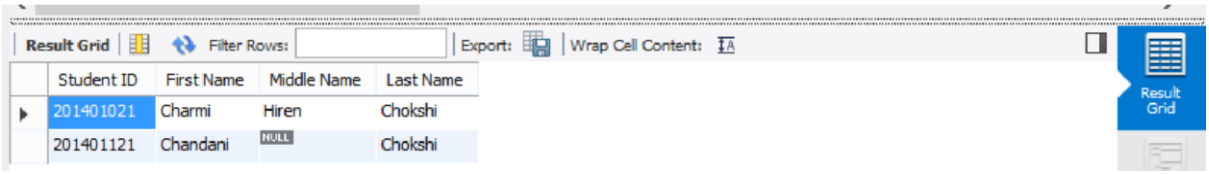
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Student ID	Course ID	grade
201301015	SEASC002	A-
201401021	SEASC002	A
201501008	SEASC002	A
201501051	SEASC002	A-

Query: 23- Display the details of all the students who have taken DBMS course in 2015.

SELECT Student.`Student ID`,Student.`First Name`,Student.`Middle Name`,Student.`Last Name`

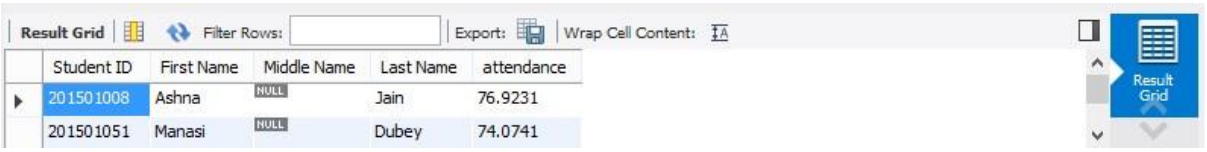
```
FROM Student,Course
WHERE  Course.`Year` - MID(Student.`Student ID`,1,4) > 2 AND Course.`Year` -
MID(Student.`Student ID`,1,4) < 4 AND Course.`Name` = 'DBMS';
```



Student ID	First Name	Middle Name	Last Name
201401021	Charmi	Hiren	Chokshi
201401121	Chandani	NULL	Chokshi

Query: 24- Display the details of all the students whose attendance is less than 80% in DBMS.

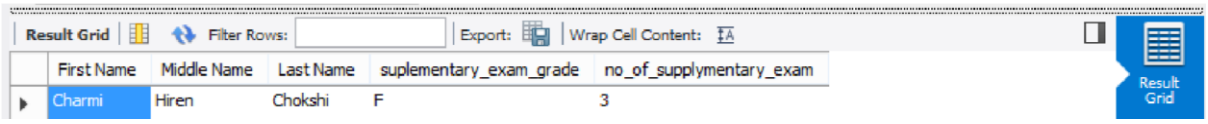
```
SELECT distinct Student.`Student ID`, Student.`First Name`,Student.`Middle
Name`,Student.`Last Name` ,(Attendance.`Days Attended`/ Attendance.`Total Days`)*100 as
attendance
FROM Student ,Attendance
WHERE Attendance.`Course ID` = 'SEASC001' AND ((Attendance.`Days Attended`/
Attendance.`Total Days`)*100) <80 and Student.`Student ID` = Attendance.`Student ID` order
by(Student.`Student ID`);
```



Student ID	First Name	Middle Name	Last Name	attendance
201501008	Ashna	NULL	Jain	76.9231
201501051	Manasi	NULL	Dubey	74.0741

Query: 25- Display the details of student who have failed in supplymentary exam.

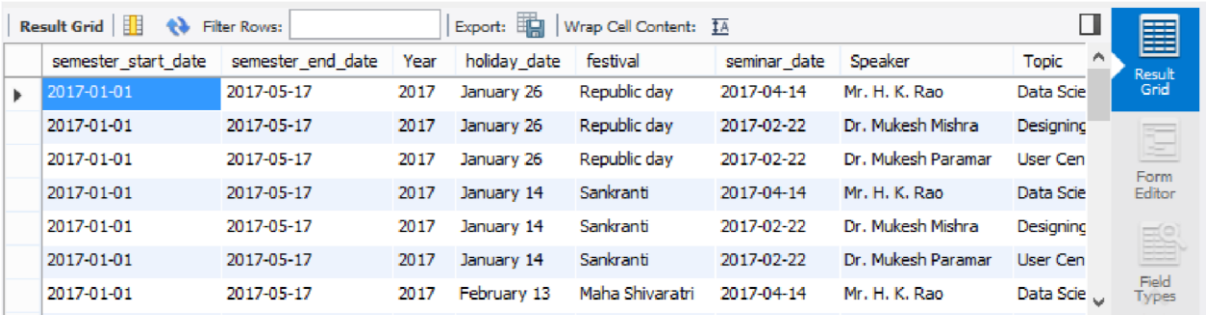
```
SELECT Student.`First Name`,Student.`Middle Name`,Student.`Last Name`, `Supplymentry
Exam`.`Result` suplementary_exam_grade,count(Student.`Student ID`) as
no_of_supplymentary_exam from Student,`Supplymentry Exam`
where Student.`Student ID` = `Supplymentry Exam`.`Student ID` and `Supplymentry
Exam`.`Result` = 'F'
group by(Student.`Student ID`);
```



First Name	Middle Name	Last Name	suplementary_exam_grade	no_of_supplymentary_exam
Charmi	Hiren	Chokshi	F	3

Query: 26- Display all the details of monsoon semester.

SELECT Semester.`Start Date` as semester_start_date,Semester.`End Date` as semester_end_date, Semester.`Year`,Holiday.`Date` as holiday_date,Holiday.`Festival Name` as festival,`Seminar/Talk`.`Date` as seminar_date, `Seminar/Talk`.`Speaker`, `Seminar/Talk`.`Topic` from
Semester,`Seminar/Talk`,Holiday
where Semester.`Year` = Holiday.`Year` and Semester.`Year` = `Seminar/Talk`.`Year`;



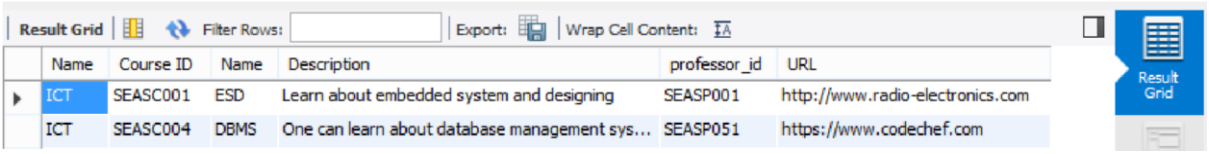
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

semester_start_date	semester_end_date	Year	holiday_date	festival	seminar_date	Speaker	Topic
2017-01-01	2017-05-17	2017	January 26	Republic day	2017-04-14	Mr. H. K. Rao	Data Scie
2017-01-01	2017-05-17	2017	January 26	Republic day	2017-02-22	Dr. Mukesh Mishra	Designing
2017-01-01	2017-05-17	2017	January 26	Republic day	2017-02-22	Dr. Mukesh Paramar	User Cen
2017-01-01	2017-05-17	2017	January 14	Sankranti	2017-04-14	Mr. H. K. Rao	Data Scie
2017-01-01	2017-05-17	2017	January 14	Sankranti	2017-02-22	Dr. Mukesh Mishra	Designing
2017-01-01	2017-05-17	2017	January 14	Sankranti	2017-02-22	Dr. Mukesh Paramar	User Cen
2017-01-01	2017-05-17	2017	February 13	Maha Shivaratri	2017-04-14	Mr. H. K. Rao	Data Scie

Result Grid | Form Editor | Field Types

Query: 27- Display all the course offered in ICT with their e-learning websites.

SELECT distinct Branch.`Name`,Course.`Course ID`,Course.`Name`,Course.`Description`,Course.`Professor ID` as professor_id,`ELearning`.`URL` as URL from Course,Branch,`E-learning`
where Course.`Course ID` = `E-learning`.`Course ID` AND `Branch`.`Name` = 'ICT' group by(Course.`Course ID`);



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Name	Course ID	Name	Description	professor_id	URL
ICT	SEASC001	ESD	Learn about embedded system and designing	SEASP001	http://www.radio-electronics.com
ICT	SEASC004	DBMS	One can learn about database management sys...	SEASP051	https://www.codechef.com

Result Grid

- Query using Sequence :

```
create SEQUENCE autoinc
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10;
```

1	Neha	Shukla	Neha_Shukla
2	Tanu	Jain	Tanu_Jain
3	Manu	Dubey	Manu_Dubey
4	Rahul	Chokshi	Rahul_Chokshi
5	Dhruv	Chandarana	Dhruv_Chandarana

- Queries using Stored Procedures:

Queri-1:
Result of a particular Course

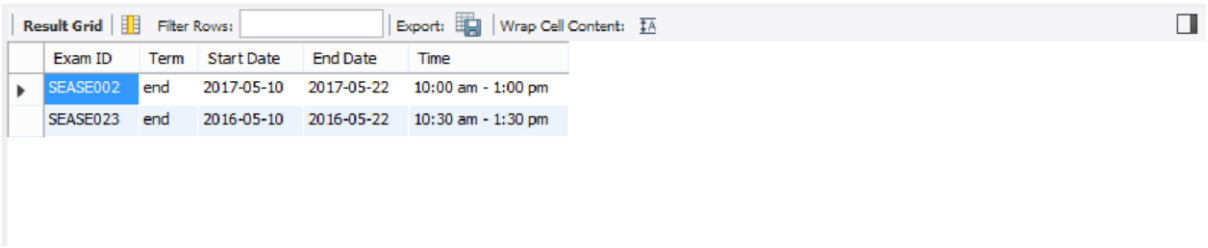
```
DELIMITER $$
USE `educational resource management system` $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `Course_Result`(C_ID
VARCHAR(10))
BEGIN
SELECT Result.`Student ID` , Result.`Course ID` ,  Result.Grade from
Result , Student
where Result.`Student ID` = Student.`Student ID` and  Result.`Course ID` = C_ID;
END$$
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Student ID	Course ID	Grade
201301015	SEASC004	A
201401021	SEASC004	B+
201501008	SEASC004	B+
201501051	SEASC004	B+

Queri-2:
Details of a particular exam (term)

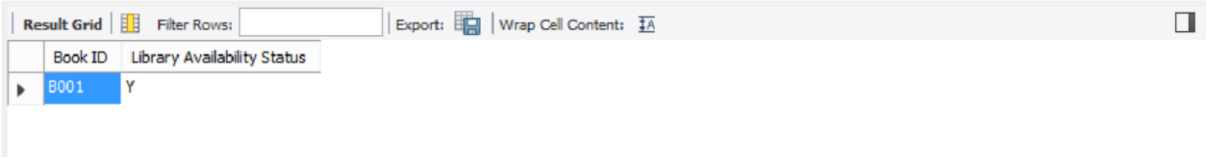
```
DELIMITER $$
USE `educational resource management system` $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `Exam_Details`(E_Term
VARCHAR(15))
BEGIN
SELECT * FROM EXAM
WHERE Term = E_Term;
END $$
```



Exam ID	Term	Start Date	End Date	Time
SEASE002	end	2017-05-10	2017-05-22	10:00 am - 1:00 pm
SEASE023	end	2016-05-10	2016-05-22	10:30 am - 1:30 pm

Queri-3:
Library status of a particular book.

```
DELIMITER $$
USE `educational resource management system` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `Library_Status`(B_ID
VARCHAR(10))
BEGIN
SELECT distinct `Book ID`,`Library Availability Status` from
Book
WHERE `Book ID` = B_ID;
END $$
```

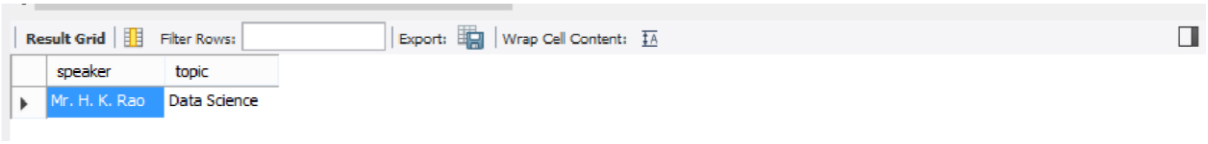


Book ID	Library Availability Status
B001	Y

Queri-4:
Details of a talk on a particular date.

```
DELIMITER $$
USE `educational resource management system` $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `Talk_Details`(D Date)
BEGIN
SELECT `speaker`,`topic` from
`Seminar/Talk`
```

where `Date` = D;
END $$



speaker	topic
Mr. H. K. Rao	Data Science

Queri-5:
Total attendance of a student in a particular course.

```
DELIMITER $$
USE `educational resource management system` $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `Total_Attendance`(`Course ID`
VARCHAR(10))
BEGIN
SELECT `Student ID`,sum(`days attended`) as total_attendance from
Attendance
where `Course ID` = `Course ID`
Group by(`Student ID`);
END $$
```

Student ID	total_attendance
201501008	93
201501051	85

Queries using Triggers:

Querie: 1- Auto generate username for admin

```
CREATE TABLE Admin_Info(
 `User ID` VARCHAR(10) NOT NULL,
   `fname` VARCHAR(45),
   `lname` VARCHAR(45),
 `User Name` VARCHAR(30)
);

CREATE TRIGGER ahead_insert
BEFORE INSERT
ON Admin_Info
FOR EACH ROW
SET NEW.`User Name` = CONCAT(
```

NEW.fname, '_', NEW.lname);

SHOW TRIGGERS

```
CREATE TABLE Admin_Info(
    `User ID` VARCHAR(10) NOT NULL,
    `fname` VARCHAR(45),
    `lname` VARCHAR(45),
    `User Name` VARCHAR(30)
);

CREATE TRIGGER ahead_insert
BEFORE INSERT
ON Admin_Info
FOR EACH ROW
SET NEW.`User Name` = CONCAT(
NEW.fname, '_', NEW.lname);

SHOW TRIGGERS
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Trigger	Event	Table	Statement	Timing	Created	sql_mode
ahead_insert	INSERT	admin_info	SET NEW.`User Name` = CONCAT(NEW.fnam...	BEFORE	2017-05-06 19:39:21.10	STRICT_TRANS_T

INSERT INTO Admin_Info(`User ID`, `fname`, `lname`) VALUES(1, 'Tanu', 'jain');

SELECT * FROM Admin_Info;

```
INSERT INTO Admin_Info(`User ID`, `fname`, `lname`) VALUES(1, 'Tanu', 'jain');

SELECT * FROM Admin_Info;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

User ID	fname	lname	User Name
1	Tanu	jain	Tanu_jain

Query: 2 –Audit of result table when updated

CREATE TABLE Result(
`Student ID` VARCHAR(10) NOT NULL,
`Course ID` VARCHAR(10) NOT NULL,
`Semester` INT NOT NULL,
`CGPA` REAL,
`Grade` VARCHAR(2),

PRIMARY KEY(`Student ID`, `Course ID`),

FOREIGN KEY (`Student ID`) references Student(`Student ID`)
ON DELETE CASCADE ON UPDATE CASCADE,

```
 FOREIGN KEY (`Course ID`) references Course(`Course ID`)
  ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE table `result_change`(
 `Student ID` VARCHAR(10) NOT NULL,
   `Course ID` VARCHAR(10) NOT NULL,
   `Semester` INT NOT NULL,
   `CGPA_From` REAL,
   `CGPA_To` REAL,
   `Grade_From` VARCHAR(2),
   `Grade_To` VARCHAR(2),
   `action` VARCHAR(50),
   `Date` DATE
);


DELIMITER $$
CREATE TRIGGER result_trigger
AFTER UPDATE ON `Result`
FOR EACH ROW
BEGIN
 INSERT INTO result_change
   SET action = "updated",
   `Course ID` = NEW.`Course ID`,
   `Semester` = old.`Semester`,
   `CGPA_From` = old.`CGPA`,
   `CGPA_To` = NEW.`CGPA`,
   `Grade_From` = old.`Grade`,
   `Grade_To` = NEW.`Grade`,
   `Student ID` = NEW.`Student ID`,
   `Date` = NOW();
END; $$
DELIMITER $$

SHOW TRIGGERS
```

Trigger	Event	Table	Statement	Timing	Created	sql_mode
ahead_insert	INSERT	admin_info	SET NEW.`User Name` = CONCAT(NEW.fnam...	BEFORE	2017-05-06 19:47:05.15	STRICT_TRANS
Attendance_Trriger	INSERT	attendance	SET NEW.`Attendance in %` = TRUNCATE((NE...	BEFORE	2017-05-07 08:04:28.58	STRICT_TRANS
result_trigger	UPDATE	result	BEGIN INSERT INTO result_change SET acti...	AFTER	2017-05-07 22:49:21.83	STRICT_TRANS
s_trigger	DELETE	supplymentry exam	BEGIN INSERT INTO s_e_change SET action...	BEFORE	2017-05-07 09:02:54.29	STRICT_TRANS

```
insert into Result values('201401021','SEASC001',4,2.02,'B');
insert into Result values('201401021','SEASC002',4,3.02,'A');
insert into Result values('201401021','SEASC003',4,2.22,'B+');
insert into Result values('201401021','SEASC004',4,2.22,'B+');
insert into Result values('201401021','SEASC005',4,2.00,'B-');
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Student ID	Course ID	Semester	CGPA	Grade
201401021	SEASC001	4	2.02	B
201401021	SEASC002	4	3.02	A
201401021	SEASC003	4	2.22	B+
201401021	SEASC004	4	2.22	B+
201401021	SEASC005	4	2	B-
NULL	NULL	NULL	NULL	NULL

```
UPDATE `Result`
SET `Grade` = 'A+' ,
 `CGPA` = 4.02
WHERE (`Course ID` = 'SEASC004' AND
 `Student ID` = 201401021);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Student ID	Course ID	Semester	CGPA	Grade
201401021	SEASC001	4	2.02	B
201401021	SEASC002	4	3.02	A
201401021	SEASC003	4	2.22	B+
201401021	SEASC004	4	4.02	A+
201401021	SEASC005	4	2	B-
NULL	NULL	NULL	NULL	NULL

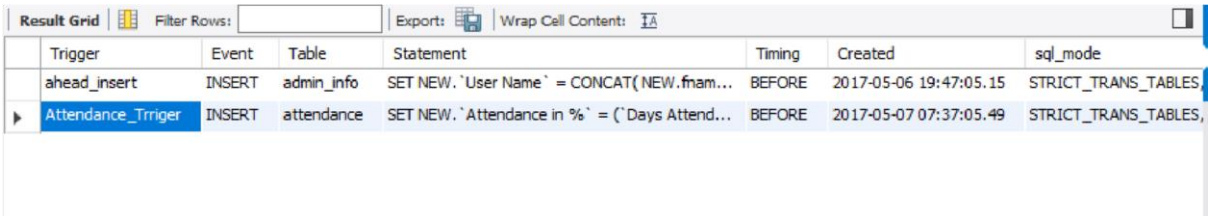
Updated table result_trigger

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Student ID	Course ID	Semester	CGPA_From	CGPA_To	Grade_From	Grade_To	action	Date
201401021	SEASC004	4	2.22	4.02	B+	A+	updated	2017-05-07

Querie: 3- Auto generate attendance in %

```
CREATE TABLE Attendance(
 `Student ID` VARCHAR(10),
   `Course ID` VARCHAR(10),
   `Month` VARCHAR(15),
   `Total Days` INT,
   `Attended Days` INT,
   `Attendance in %` REAL,

   PRIMARY KEY(`Student ID`, `Course ID`,`month`),

   FOREIGN KEY (`Student ID`) references Student(`Student ID`)
  ON DELETE CASCADE ON UPDATE CASCADE,

 FOREIGN KEY (`Course ID`) references Course(`Course ID`)
  ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TRIGGER Attendance_Trriger
BEFORE INSERT
ON Attendance
FOR EACH ROW
SET NEW.`Attendance in %` = TRUNCATE((NEW.`Attended Days` * 100 / NEW.`Total
Days`),2);

SHOW TRIGGERS
```

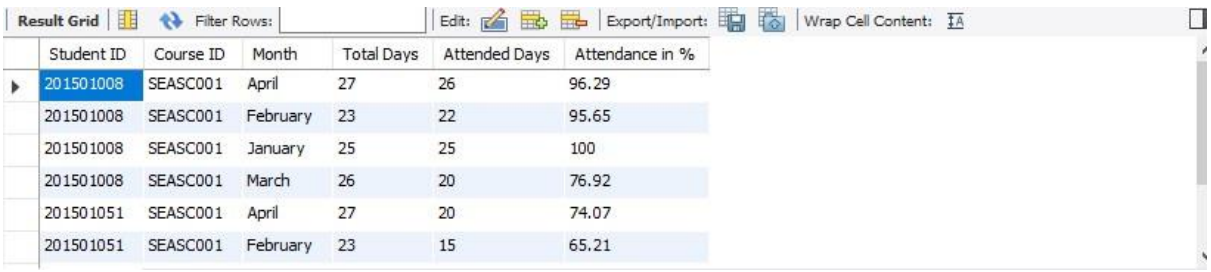


Trigger	Event	Table	Statement	Timing	Created	sql_mode
ahead_insert	INSERT	admin_info	SET NEW.`User Name` = CONCAT(NEW.fnam...	BEFORE	2017-05-06 19:47:05.15	STRICT_TRANS_TABLES,
Attendance_Trriger	INSERT	attendance	SET NEW.`Attendance in %` = (`Days Attend...	BEFORE	2017-05-07 07:37:05.49	STRICT_TRANS_TABLES,

```
INSERT INTO Attendance(`Student ID`,`Course ID`,`Month`,`Total Days`,`Attended
Days`) VALUES(201501008,'SEASC001','January',25,25);
INSERT INTO Attendance(`Student ID`,`Course ID`,`Month`,`Total Days`,`Attended
Days`) VALUES(201501008,'SEASC001','February',23,22);
INSERT INTO Attendance(`Student ID`,`Course ID`,`Month`,`Total Days`,`Attended
Days`) VALUES(201501008,'SEASC001','March',26,20);
INSERT INTO Attendance(`Student ID`,`Course ID`,`Month`,`Total Days`,`Attended
Days`) VALUES(201501008,'SEASC001','April',27,26);
INSERT INTO Attendance(`Student ID`,`Course ID`,`Month`,`Total Days`,`Attended
Days`) VALUES(201501051,'SEASC001','January',25,25);
```

INSERT INTO Attendance(`Student ID`,`Course ID`,`Month`,`Total Days`,`Attended Days`) VALUES(201501051,'SEASC001','February',23,15);
INSERT INTO Attendance(`Student ID`,`Course ID`,`Month`,`Total Days`,`Attended Days`) VALUES(201501051,'SEASC001','March',26,25);

INSERT INTO Attendance(`Student ID`,`Course ID`,`Month`,`Total Days`,`Attended Days`) VALUES(201501051,'SEASC001','April',27,20);

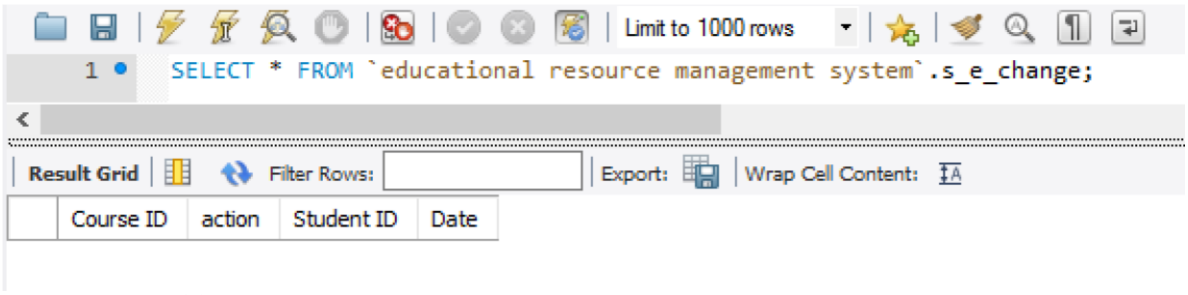


Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Student ID	Course ID	Month	Total Days	Attended Days	Attendance in %
201501008	SEASC001	April	27	26	96.29
201501008	SEASC001	February	23	22	95.65
201501008	SEASC001	January	25	25	100
201501008	SEASC001	March	26	20	76.92
201501051	SEASC001	April	27	20	74.07
201501051	SEASC001	February	23	15	65.21

Querie: 4- Storing details of deleted student from supplementary exam table using trigger.

```
CREATE table `s_e_change`(
   `Course ID` VARCHAR(10),
   `action` VARCHAR(50),
   `Student ID` VARCHAR(10),
   `Date` DATE
);
```




```
DELIMITER $$
CREATE TRIGGER s_trigger
BEFORE DELETE ON `supplymentry exam`
FOR EACH ROW
BEGIN
 INSERT INTO s_e_change
   SET action = "deleted",
   `Course ID` = OLD.`Course ID`,
   `Student ID` = OLD.`Student ID`,
```

```
    `Date` = NOW();
END; $$
DELIMITER $$
SHOW TRIGGERS
```

Trigger	Event	Table	Statement	Timing	Created	sql_mode
ahead_insert	INSERT	admin_info	SET NEW.`User Name` = CONCAT(NEW.fnam...	BEFORE	2017-05-06 19:47:05.15	STRICT_TRANS_
Attendance_Trriger	INSERT	attendance	SET NEW.`Attendance in %` = TRUNCATE((NE...	BEFORE	2017-05-07 08:04:28.58	STRICT_TRANS_
s_trigger	DELETE	supplymentry exam	BEGIN INSERT INTO s_e_change SET action...	BEFORE	2017-05-07 09:02:54.29	STRICT_TRANS_

```
insert into `Supplymentry Exam`
values('SEASE001','mid','201401021','SEASC001','2017/06/15','F');
insert into `Supplymentry Exam`
values('SEASE002','end','201401021','SEASC002','2017/06/20','F');
insert into `Supplymentry Exam`
values('SEASE022','mid','201401021','SEASC004','2017/06/22','F');
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Exam ID	Term	Student ID	Course ID	Date	Result
SEASE001	mid	201401021	SEASC001	2017-06-15	F
SEASE002	end	201401021	SEASC002	2017-06-20	F
SEASE022	mid	201401021	SEASC004	2017-06-22	F
NULL	NULL	NULL	NULL	NULL	NULL

```
DELETE FROM `Supplymentry Exam` WHERE (`Student ID` = 201401021 AND `EXAM ID` = 'SEASE022');
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Exam ID	Term	Student ID	Course ID	Date	Result
SEASE001	mid	201401021	SEASC001	2017-06-15	F
SEASE002	end	201401021	SEASC002	2017-06-20	F
NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT * FROM `educational resource management system`.s_e_change;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Course ID	action	Student ID	Date
SEASC004	deleted	201401021	2017-05-07

...Thank You...