Name: Divya Jha

Roll no:L021

Sap id: 40778230025

Practical No: Subquery-join operations on Relational Schema

1. Design ERD for the following schema and execute the following Queries on it:

Consider the schema for Movie Database:

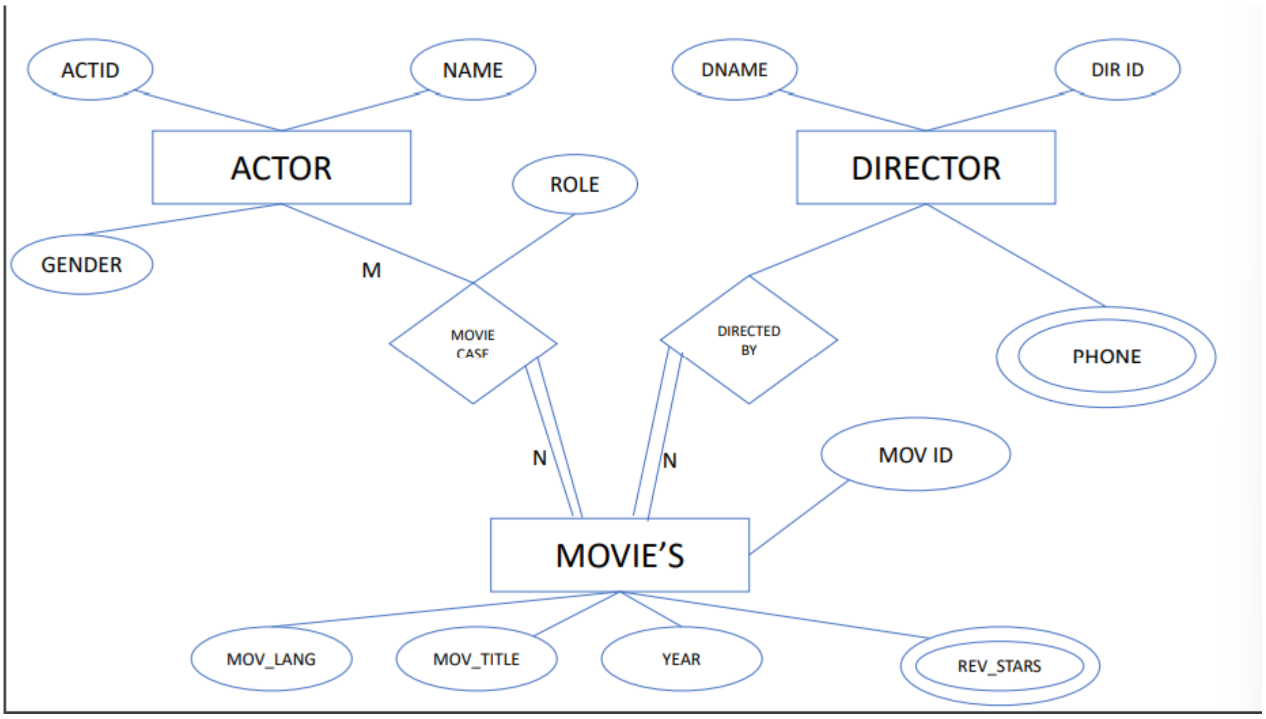
ACTOR (Act_id, Act_Name, Act_Gender)

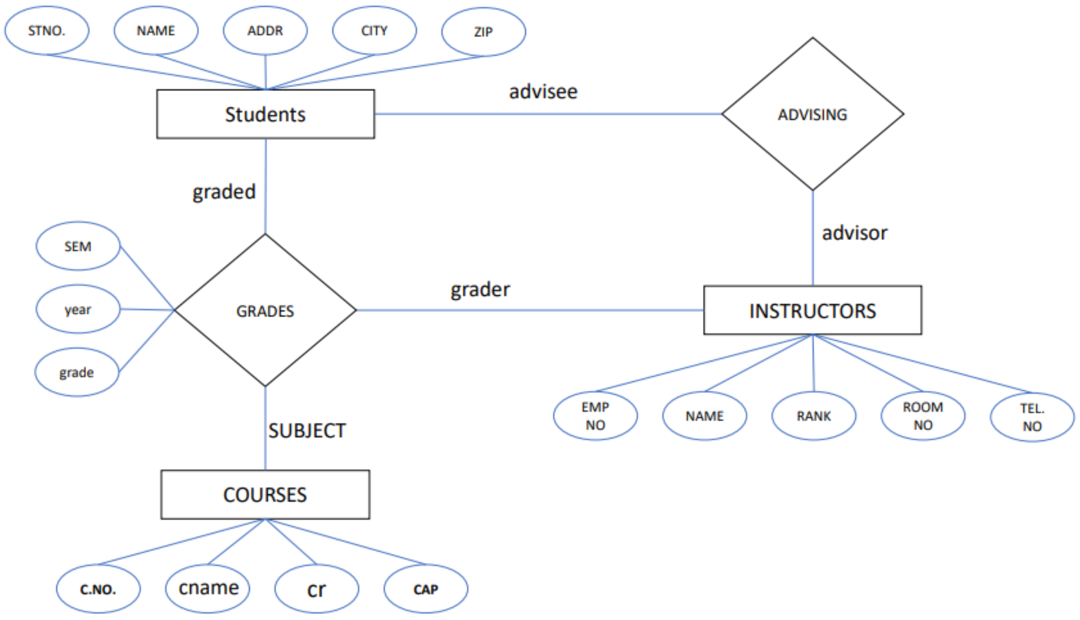
DIRECTOR (Dir_id, Dir_Name, Dir_Phone)

MOVIES (Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

MOVIE_CAST (Act_id, Mov_id, Role)

RATING (Mov_id, Rev_Stars)




STNO.
NAME
ADDR
CITY
ZIP
Students
advisee
ADVISING
graded
advisor
SEM
year
grade
GRADES
grader
INSTRUCTORS
EMP NO
NAME
RANK
ROOM NO
TEL. NO
SUBJECT
COURSES
C.NO.
cname
cr
CAP

#1. USING (practical 1)

```
mysql> use dbms;
mysql> select * from salesman;
+-------------+------------+----------+------------+
| salesman_id | name       | city     | commission |
+-------------+------------+----------+------------+
|        5001 | James Hooq | New York |       0.15 |
|        5002 | Nail Knite | Paris    |       0.13 |
|        5003 | Lauson Hen |          |       0.12 |
|        5005 | Pit Alex   | London   |       0.11 |
|        5006 | Mc Lyon    | Paris    |       0.14 |
|        5007 | Paul Adam  | Rome     |       0.13 |
+-------------+------------+----------+------------+
6 rows in set (0.06 sec)

mysql> select * from customer;
+-------------+---------------+------------+-------+-------------+
| customer_id | customer_name | city       | grade | salesman_id |
+-------------+---------------+------------+-------+-------------+
|        3001 | Brad Guzan    | London     |  NULL |        NULL |
|        3002 | Nick Rimando  | New York   |   100 |        5001 |
|        3003 | Jory Altidor  | Moncow     |   200 |        5007 |
|        3004 | Fabian Johns  | Paris      |   300 |        5006 |
|        3005 | Graham Zusi   | California |   200 |        5002 |
|        3007 | Brad Davis    | New York   |   200 |        5001 |
|        3008 | Julian Green  | London     |   300 |        5002 |
|        3009 | Geoff Camero  | Berlin     |   100 |        NULL |
+-------------+---------------+------------+-------+-------------+
8 rows in set (0.05 sec)

mysql> select * from orders;
+----------+-----------+------------+-------------+-------------+
| order_no | purch_amt | order_date | customer_id | salesman_id |
+----------+-----------+------------+-------------+-------------+
|    70001 |     150.5 | 2016-10-05 |        3005 |        5002 |
|    70009 |    270.65 | 2016-09-10 |        3001 |        NULL |
|    70002 |     65.26 | 2016-10-05 |        3002 |        5001 |
|    70004 |     110.5 | 2016-08-17 |        3009 |        NULL |
|    70007 |     948.5 | 2016-09-10 |        3005 |        5002 |
|    70005 |    2400.6 | 2016-07-27 |        3007 |        5001 |
|    70008 |      5760 | 2016-09-10 |        3002 |        5001 |
|    70010 |   1983.43 | 2016-10-10 |        3004 |        5006 |
|    70003 |    2480.4 | 2016-10-10 |        3009 |        NULL |
|    70012 |    250.45 | 2016-06-27 |        3008 |        5002 |
|    70011 |     75.29 | 2016-08-17 |        3003 |        5007 |
+----------+-----------+------------+-------------+-------------+
11 rows in set (0.06 sec)
```

1. Count the customers with grades above NewYork average

```
mysql> select count(*) from customer where grade > (select avg(grade) from customer where city="New York");
+----------+
| count(*) |
+----------+
|        5 |
+----------+
1 row in set (0.00 sec)
```

2.Find the name and numbers of all salesmen who had more than one customer
```
mysql> select salesman_id, name from salesman a where 1<(select count(*) from customer where salesman_id = a.s
+-------------+------------+
| salesman_id | name       |
+-------------+------------+
|        5001 | James Hooq |
|        5002 | Nail Knite |
+-------------+------------+
2 rows in set (0.01 sec)
```

3)Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted
```
mysql> delete from orders where salesman_id=5002;
Query OK, 3 rows affected (0.03 sec)

mysql> delete from customer where salesman_id=5002;
Query OK, 2 rows affected (0.01 sec)

mysql> delete from salesman where salesman_id=5002;
Query OK, 1 row affected (0.01 sec)

mysql> select * from salesman;
+-------------+------------+----------+------------+
| salesman_id | name       | city     | commission |
+-------------+------------+----------+------------+
|        5001 | James Hooq | New York |       0.15 |
|        5005 | Pit Alex   | London   |       0.11 |
|        5006 | Mc Lyon    | Paris    |       0.14 |
|        5007 | Paul Adam  | Rome     |       0.13 |
+-------------+------------+----------+------------+
4 rows in set (0.00 sec)

mysql> select * from orders;
+----------+-----------+------------+-------------+-------------+
| order_no | purch_amt | order_date | customer_id | salesman_id |
+----------+-----------+------------+-------------+-------------+
|    70009 |    270.65 | 2016-09-10 |        3001 |        NULL |
|    70002 |     65.26 | 2016-10-05 |        3002 |        5001 |
|    70004 |     110.5 | 2016-08-17 |        3009 |        NULL |
|    70005 |    2400.6 | 2016-07-27 |        3007 |        5001 |
|    70008 |      5760 | 2016-09-10 |        3002 |        5001 |
|    70010 |   1983.43 | 2016-10-10 |        3004 |        5006 |
|    70003 |    2480.4 | 2016-10-10 |        3009 |        NULL |
|    70011 |     75.29 | 2016-08-17 |        3003 |        5007 |
+----------+-----------+------------+-------------+-------------+
8 rows in set (0.00 sec)

mysql> select * from customer;
+-------------+---------------+----------+-------+-------------+
| customer_id | customer_name | city     | grade | salesman_id |
+-------------+---------------+----------+-------+-------------+
|        3001 | Brad Guzan    | London   |  NULL |        NULL |
|        3002 | Nick Rimando  | New York |   100 |        5001 |
|        3003 | Jory Altidor  | Moncow   |   200 |        5007 |
|        3004 | Fabian Johns  | Paris    |   300 |        5006 |
|        3007 | Brad Davis    | New York |   200 |        5001 |
|        3009 | Geoff Camero  | Berlin   |   100 |        NULL |
+-------------+---------------+----------+-------+-------------+
6 rows in set (0.00 sec)
```

Q2. Design ERD for the following schema and execute the following Queries on it:
Consider the schema for Movie Database:
ACTOR (Act_id, Act_Name, Act_Gender)
DIRECTOR (Dir_id, Dir_Name, Dir_Phone)
MOVIES (Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)
MOVIE_CAST (Act_id, Mov_id, Role)
RATING (Mov_id, Rev_Stars)

```
mysql> CREATE TABLE ACTOR (ACT_ID int(10) primary key,ACT_NAME  VARCHAR (20),ACT_GENDER CHAR (1));
Query OK, 0 rows affected (0.16 sec)

mysql> CREATE TABLE director(dir_id int(10) primary key,dir_name varchar(20),dir_phone int(20));
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE MOVIES (MOV_ID int (10) primary key,MOV_TITLE VARCHAR (25),MOV_YEAR int(4),MOV_LANG V
));

mysql> CREATE TABLE MOVIE_CAST (ACT_ID int,foreign key(act_id) references actor(act_id),MOV_ID int, foreign key(m

mysql> CREATE TABLE RATING (MOV_ID int, foreign key (mov_id) references movies(mov_id),rev_stars varchar(20));

mysql> insert into Actor values(301, 'anuska','f'),
 -> (302,'PRABHAS','M'),
 -> (303,'PUNITH','M'),
 -> (304,'jermy','M');
Query OK, 4 rows affected (0.03 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> insert into director values(60, 'rajamouli',8751611001),
 -> (61,'HITCHCOCK', 7766138911),
 -> (62,'FARAN', 9986776531),
 -> (63,'STEVEN SPIELBERG', 8989776530);
Query OK, 4 rows affected (0.00 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> insert into movies values(1001,'BAHUBALI-2', 2017, 'TELAGU', 60),
 -> (1002,'BAHUBALI-2', 2015, 'TELAGU', 60),
 -> (1003,'AKASH', 2008, 'KANNADA', 61),
 -> (1004,'WAR HORSE', 2011, 'ENGLISH', 63);
Query OK, 4 rows affected (0.00 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> INSERT INTO MOVIE_CAST VALUES (301, 1002, 'HEROINE'),
 -> (301, 1001, 'HEROINE'),
 -> (303, 1003, 'HERO'),
 -> (303, 1002, 'guest'),
 -> (304, 1004, 'hero');
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO RATING VALUES (1001, 4),
 -> (1002, 2),
 -> (1003, 5),
 -> (1004, 4);
Query OK, 4 rows affected (0.00 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

#Write SQL queries to

1. List the titles of all movies directed by ‘Hitchcock

```
mysql> select mov_title from movies where dir_id in(select dir_id from director
where dir_name='hitchcock');
+-----------+
| mov_title |
+-----------+
| AKASH |
+-----------+
1 row in set (0.00 sec)
```

2. Find the movie names where one or more actors acted in two or more movies.

```
mysql> select mov_title from movies m,movie_cast mv where m.mov_id=mv.mov_id and
act_id in(select act_id from movie_cast group by act_id having count(act_id)>1)
group by mov_title having count(*)>1;
+------------+
| mov_title |
+------------+
| BAHUBALI-2 |
+------------+
1 row in set (0.00 sec)
```

3. List all actors who acted in a movie before 2000 and also in a movie after
2015 (use JOIN operation).

```
mysql> select a.act_name,c.mov_title,c.mov_year from actor a,movie_cast b,movies c
where a.act_id=b.act_id and b.mov_id=c.mov_id and c.mov_year not between 2000 and
2015;
+----------+------------+----------+
| act_name | mov_title | mov_year |
+----------+------------+----------+
| anuska | BAHUBALI-2 | 2017 |
+----------+------------+----------+
1 row in set (0.00 sec)
```

4. Find the title of movies and number of stars for each movie that has at least one
rating and find the highest number of stars that movie received. Sort the result by
movie title

```
mysql> select mov_title,max(rev_stars) from movies inner join rating using(mov_id)
group by mov_title having max(rev_stars)>0 order by mov_tit
le;
+------------+----------------+
| mov_title | max(rev_stars) |
+------------+----------------+
| AKASH | 5 |
| BAHUBALI-2 | 4 |
| WAR HORSE | 4 |
+------------+----------------+
3 rows in set (0.00 sec)
```

5. Update rating of all movies directed by ‘Steven Spielberg’ to 5.

```
mysql> update rating set rev_stars=5 where mov_id in(select mov_id from movies where
dir_id in (select dir_id from director where dir_name='STEVEN SPIELBERG'));
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> select * from rating;
+--------+-----------+
| mov_id | rev_stars |
+--------+-----------+
| 1001 | 4 |
| 1002 | 2 |
| 1003 | 5 |
| 1004 | 5 |
+--------+-----------+
4 rows in set (0.00 sec)
```

3. Design ERD for the following schema and execute the following Queries on it:

```
mysql> CREATE TABLE students (
 -> stno INT PRIMARY KEY,
 -> name VARCHAR(50),
 -> addr VARCHAR(255),
 -> city VARCHAR(50),
 -> state VARCHAR(2),
 -> zip VARCHAR(10)
 -> );
Query OK, 0 rows affected (0.01 sec)
mysql> CREATE TABLE INSTRUCTORS (
 -> empno INT PRIMARY KEY,
 -> name VARCHAR(50),
 -> ranks VARCHAR(20),
 -> roomno VARCHAR(10),
 -> telno VARCHAR(15)
 -> );
Query OK, 0 rows affected (0.01 sec)
mysql> CREATE TABLE COURSES (
 -> cno text PRIMARY KEY,
 -> cname VARCHAR(50),
 -> cr INT,
 -> cap INT
 -> );
Query OK, 0 rows affected (0.01 sec)
mysql> CREATE TABLE GRADES (
 -> stno INT,
 -> empno INT,
 -> cno VARCHAR(50),
 -> sem VARCHAR(10),
 -> year INT,
 -> grade INT,
 -> FOREIGN KEY (stno) REFERENCES students(stno),
 -> FOREIGN KEY (empno) REFERENCES INSTRUCTORS(empno),
 -> FOREIGN KEY (cno) REFERENCES COURSES(cno)
 -> );
Query OK, 0 rows affected (0.02 sec)
mysql> CREATE TABLE ADVISING (
 -> stno INT,
 -> empno INT,
 -> PRIMARY KEY (stno, empno),
 -> FOREIGN KEY (stno) REFERENCES students(stno),
 -> FOREIGN KEY (empno) REFERENCES INSTRUCTORS(empno)
 -> );
Query OK, 0 rows affected (0.02 sec)
mysql> insert into students values
 ->(1011,'edwards p. david','10 red rd','newton','MA','02159')
 ->(2415, 'Grogan A. Mary', '8 Walnut St', 'Malden', 'MA', '02148'),
 -> (2661, 'Mixon Leatha', '100 School St', 'Brookline', 'MA', '02146'),
 -> (2890, 'McLane Sandy', '30 Case Rd', 'Boston', 'MA', '02122'),
 -> (3442, 'Novak Roland', '42 Beacon St', 'Nashua', 'NH', '03060'),
 -> (3566, 'Pierce Richard', '70 Park St', 'Brookline', 'MA', '02146'),
 -> (4022, 'Prior Lorraine', '8 Beacon St', 'Boston', 'MA', '02125'),
 -> (5544, 'Rawlings Jerry', '15 Pleasant Dr', 'Boston', 'MA', '02115'),
 -> (5571, 'Lewis Jerry', '1 Main Rd', 'Providence', 'RI', '02904');
mysql> select * from students;
+------+------------------+----------------+------------+-------+-------+
| stno | name | addr | city | state | zip |
+------+------------------+----------------+------------+-------+-------+
| 1011 | edwards p. david | 10 red rd | newton | MA | 02159 |
| 2415 | Grogan A. Mary | 8 Walnut St | Malden | MA | 02148 |
| 2661 | Mixon Leatha | 100 School St | Brookline | MA | 02146 |
| 2890 | McLane Sandy | 30 Case Rd | Boston | MA | 02122 |
| 3442 | Novak Roland | 42 Beacon St | Nashua | NH | 03060 |
| 3566 | Pierce Richard | 70 Park St | Brookline | MA | 02146 |
| 4022 | Prior Lorraine | 8 Beacon St | Boston | MA | 02125 |
| 5544 | Rawlings Jerry | 15 Pleasant Dr | Boston | MA | 02115 |
| 5571 | Lewis Jerry | 1 Main Rd | Providence | RI | 02904 |
+------+------------------+----------------+------------+-------+-------+
9 rows in set (0.00 sec)
mysql> INSERT INTO instructors VALUES
 -> (19, 'Evans Robert', 'Professor', '82', '7122'),
 -> (23, 'Exxon George', 'Professor', '90', '9101'),
 -> (56, 'Sawyer Kathy', 'Assoc Prof', '91', '5110'),
 -> (126, 'Davis William', 'Assoc Prof', '72', '5411'),
 -> (234, 'Will Samuel', 'Assist Prof', '90', '7024');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
mysql> select * from instructors;
+-------+---------------+-------------+--------+-------+
| empno | name | ranks | roomno | telno |
+-------+---------------+-------------+--------+-------+
| 19 | Evans Robert | Professor | 82 | 7122 |
| 23 | Exxon George | Professor | 90 | 9101 |
| 56 | Sawyer Kathy | Assoc Prof | 91 | 5110 |
| 126 | Davis William | Assoc Prof | 72 | 5411 |
| 234 | Will Samuel | Assist Prof | 90 | 7024 |
+-------+---------------+-------------+--------+-------+
5 rows in set (0.00 sec)
mysql> insert into courses values
 -> ('cs110', 'Introduction to Computing', 4, 120),
 -> ('cs210', 'Computer Programming', 4, 100),
 -> ('cs240', 'Computer Architecture', 3, 100),
 -> ('cs310', 'Data Structures', 3, 60),
 -> ('cs350', 'Higher Level Languages', 3, 50),
 -> ('cs410', 'Software Engineering', 3, 40),
 -> ('cs460', 'Graphics', 3, 30);
Query OK, 7 rows affected (0.00 sec)
Records: 7 Duplicates: 0 Warnings: 0
mysql> select * from courses;
+-------+---------------------------+------+------+
| cno | cname | cr | cap |
+-------+---------------------------+------+------+
| cs110 | Introduction to Computing | 4 | 120 |
| cs210 | Computer Programming | 4 | 100 |
| cs240 | Computer Architecture | 3 | 100 |
| cs310 | Data Structures | 3 | 60 |
| cs350 | Higher Level Languages | 3 | 50 |
| cs410 | Software Engineering | 3 | 40 |
| cs460 | Graphics | 3 | 30 |
+-------+---------------------------+------+------+
7 rows in set (0.00 sec)
mysql> insert into grades values
 -> (1011, 019, 'cs110', 'Fall', 2001, 40),
 -> (2661, 019, 'cs110', 'Fall', 2001, 80),
 -> (3566, 019, 'cs110', 'Fall', 2001, 95),
 -> (5544, 019, 'cs110', 'Fall', 2001, 100),
 -> (1011, 023, 'cs110', 'Spring', 2002, 75),
 -> (4022, 023, 'cs110', 'Spring', 2002, 60),
 -> (3566, 019, 'cs240', 'Spring', 2002, 100),
 -> (5571, 019, 'cs240', 'Spring', 2002, 50),
 -> (2415, 019, 'cs240', 'Spring', 2002, 100),
 -> (3442, 234,'cs410', 'Spring', 2002, 60),
 -> (5571, 234, 'cs410', 'Spring', 2002, 80),
 -> (1011, 019, 'cs210', 'Fall', 2002, 90),
 -> (2661, 019, 'cs210', 'Fall', 2002, 70),
 -> (3566, 019, 'cs210', 'Fall', 2002, 90),
 -> (5571, 019, 'cs210', 'Spring', 2003, 85),
 -> (4022, 019, 'cs210', 'Spring', 2003, 70),
 -> (5544, 56, 'cs240', 'Spring', 2003, 70),
 -> (1011, 56, 'cs240', 'Spring', 2003, 90),
 -> (4022, 56, 'cs240', 'Spring', 2003, 80),
 -> (2661, 234, 'cs310', 'Spring', 2003, 100),
 -> (4022, 234, 'cs310', 'Spring',2003, 75);
Query OK, 21 rows affected (0.00 sec)
Records: 21 Duplicates: 0 Warnings: 0
mysql> select * from grades;
+------+-------+-------+--------+------+-------+
| stno | empno | cno | sem | year | grade |
+------+-------+-------+--------+------+-------+
| 1011 | 19 | cs110 | Fall | 2001 | 40 |
| 2661 | 19 | cs110 | Fall | 2001 | 80 |
| 3566 | 19 | cs110 | Fall | 2001 | 95 |
| 5544 | 19 | cs110 | Fall | 2001 | 100 |
| 1011 | 23 | cs110 | Spring | 2002 | 75 |
| 4022 | 23 | cs110 | Spring | 2002 | 60 |
| 3566 | 19 | cs240 | Spring | 2002 | 100 |
| 5571 | 19 | cs240 | Spring | 2002 | 50 |
| 2415 | 19 | cs240 | Spring | 2002 | 100 |
| 3442 | 234 | cs410 | Spring | 2002 | 60 |
| 5571 | 234 | cs410 | Spring | 2002 | 80 |
| 1011 | 19 | cs210 | Fall | 2002 | 90 |
| 2661 | 19 | cs210 | Fall | 2002 | 70 |
| 3566 | 19 | cs210 | Fall | 2002 | 90 |
| 5571 | 19 | cs210 | Spring | 2003 | 85 |
| 4022 | 19 | cs210 | Spring | 2003 | 70 |
| 5544 | 56 | cs240 | Spring | 2003 | 70 |
| 1011 | 56 | cs240 | Spring | 2003 | 90 |
| 4022 | 56 | cs240 | Spring | 2003 | 80 |
| 2661 | 234 | cs310 | Spring | 2003 | 100 |
| 4022 | 234 | cs310 | Spring | 2003 | 75 |
+------+-------+-------+--------+------+-------+
21 rows in set (0.00 sec)
mysql> insert into advising values
 -> (1011,019);
 -> (2415,019),
 -> (2661,0023),
 -> (2890,023),
 -> (3442,0056),
 -> (3566,126),
 -> (4022,234),
 -> (5544,023),
 -> (5571,234);
Query OK, 8 rows affected (0.00 sec)
Records: 8 Duplicates: 0 Warnings: 0
mysql> select * from advising;
+------+-------+
| stno | empno |
+------+-------+
| 1011 | 19 |
| 2415 | 19 |
| 2661 | 23 |
| 2890 | 23 |
| 5544 | 23 |
| 3442 | 56 |
| 3566 | 126 |
| 4022 | 234 |
| 5571 | 234 |
+------+-------+
9 rows in set (0.00 sec)
```

#Queries

1. Find the names of students who took only four-credit courses.

```
mysql> SELECT DISTINCT s.name
 -> FROM students s
 -> JOIN grades g ON s.stno = g.stno
 -> JOIN courses c ON g.cno = c.cno
 -> WHERE c.cr = 4
 -> AND g.cno NOT IN (
 -> SELECT cno
 -> FROM courses
 -> WHERE cr != 4
 -> );
+------------------+
| name |
+------------------+
| edwards p. david |
| Mixon Leatha |
| Pierce Richard |
| Rawlings Jerry |
| Prior Lorraine |
| Lewis Jerry |
+------------------+
6 rows in set (0.00 sec)
```

2.Find the names of students who took no four-credit courses.

```
mysql> SELECT DISTINCT s.name
 -> FROM students s
 -> WHERE s.stno NOT IN (
 -> SELECT DISTINCT g.stno
 -> FROM grades g
 -> JOIN courses c ON g.cno = c.cno
 -> WHERE c.cr = 4
 -> );
+----------------+
| name |
+----------------+
| Grogan A. Mary |
| McLane Sandy |
| Novak Roland |
+----------------+
3 rows in set (0.00 sec)
```

3. Find the names of students who took cs210 or cs310

```
mysql> select name from students where stno in (select stno from grades where
cno='cs210' or cno='cs310');
+------------------+
| name |
+------------------+
| edwards p. david |
| Mixon Leatha |
| Pierce Richard |
| Prior Lorraine |
| Lewis Jerry |
+------------------+
5 rows in set (0.00 sec)
```

4. Find names of all students who have a cs210 grade higher than the highest grade given in cs310 and did not take any course with Prof. Evans.

```
mysql> SELECT s.name
 -> FROM students s
 -> WHERE s.stno IN (
 -> SELECT g1.stno
 -> FROM grades g1
 -> WHERE g1.cno = 'cs210'
 -> AND g1.grade > (
 -> SELECT MAX(g2.grade)
 -> FROM grades g2
 -> WHERE g2.cno = 'cs310'
 -> )
 -> )
 -> AND s.stno NOT IN (
 -> SELECT g3.stno
 -> FROM grades g3
 -> JOIN instructors i ON g3.empno = i.empno
 -> WHERE i.name = 'Evans Robert'
 -> );
Empty set (0.00 sec)
```

5.. Find course numbers for courses that enrol at least two students,solve the same query for courses that enroll at least three students

```
mysql> SELECT cno
 -> FROM grades
 -> GROUP BY cno
 -> HAVING COUNT(DISTINCT stno) >= 2;
+-------+
| cno |
+-------+
| cs110 |
| cs210 |
| cs240 |
| cs310 |
| cs410 |
+-------+
5 rows in set (0.00 sec)
mysql> SELECT cno
 -> FROM grades
 -> GROUP BY cno
 -> HAVING COUNT(DISTINCT stno) >= 3;
+-------+
| cno |
+-------+
| cs110 |
| cs210 |
| cs240 |
+-------+
3 rows in set (0.00 sec)
```

6. Find the names of students who obtained the highest grade in cs210.

```
mysql> SELECT s.name
 -> FROM students s
 -> JOIN grades g ON s.stno = g.stno
 -> WHERE g.cno = 'cs210' AND g.grade = (SELECT MAX(grade) FROM grades WHERE cno
= 'cs210');
+------------------+
| name |
+------------------+
| edwards p. david |
| Pierce Richard |
+------------------+
2 rows in set (0.00 sec)
```
7. Find course numbers for courses that enroll exactly two students;
```
mysql> SELECT cno
 -> FROM grades
 -> GROUP BY cno
 -> HAVING COUNT(DISTINCT stno) = 2;
+-------+
| cno |
+-------+
| cs310 |
| cs410 |
+-------+
2 rows in set (0.00 sec)
```
8. Find the names of all students for whom no other student lives in the same city.
```
mysql> SELECT DISTINCT s1.name
 -> FROM students s1
 -> WHERE NOT EXISTS (
 -> SELECT 1
 -> FROM students s2
 -> WHERE s2.city = s1.city AND s2.stno <> s1.stno
 -> );
+------------------+
| name |
+------------------+
| edwards p. david |
| Grogan A. Mary |
| Novak Roland |
| Lewis Jerry |
+------------------+
4 rows in set (0.00 sec)
```
9.Find the names of students whose advisor did not teach them any course
```
mysql> SELECT s.name
 -> FROM students s
 -> WHERE NOT EXISTS (
 -> SELECT 1
 -> FROM advising a
 -> WHERE a.stno = s.stno
 -> AND NOT EXISTS (
 -> SELECT 1
 -> FROM grades g
 -> WHERE g.stno = a.stno
 -> AND g.empno = a.empno
 -> )
 -> );
+------------------+
| name |
+------------------+
| edwards p. david |
| Grogan A. Mary |
| Prior Lorraine |
| Lewis Jerry |
+------------------+
4 rows in set (0.00 sec)
```
10.Find the highest grade of a student who never took cs110
```
mysql> SELECT MAX(grade) AS highest_grade
 -> FROM grades
 -> WHERE stno NOT IN (
 -> SELECT stno
 -> FROM grades
 -> WHERE cno = 'cs110'
 -> );
+---------------+
| highest_grade |
+---------------+
| 100 |
+---------------+
1 row in set (0.00 sec)
```