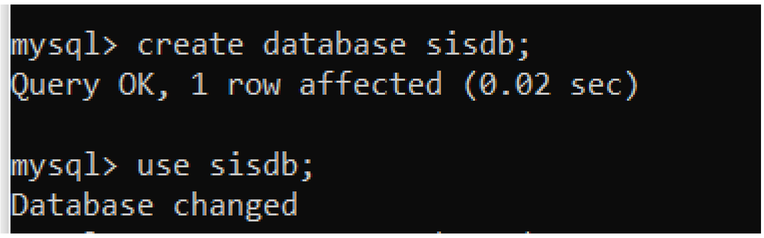
ASSIGNMENT 2

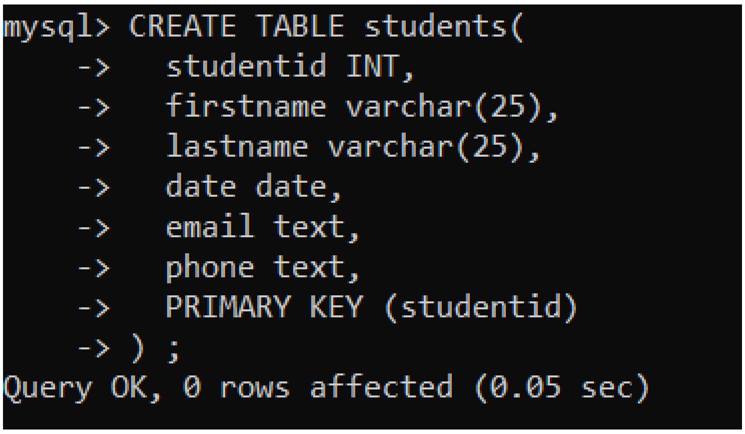
AKILESH K

1

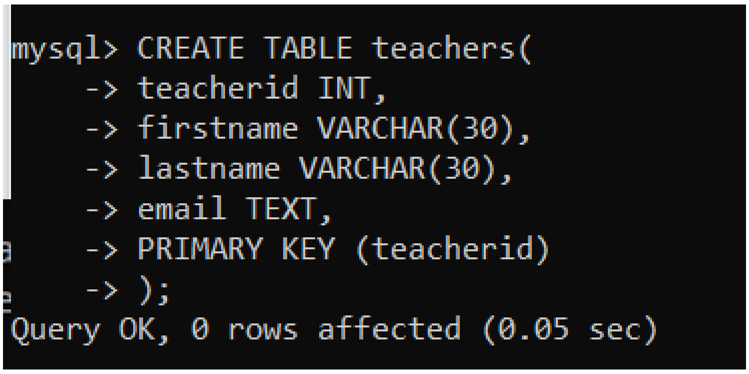


```
mysql> create database sisdb;
Query OK, 1 row affected (0.02 sec)

mysql> use sisdb;
Database changed
```

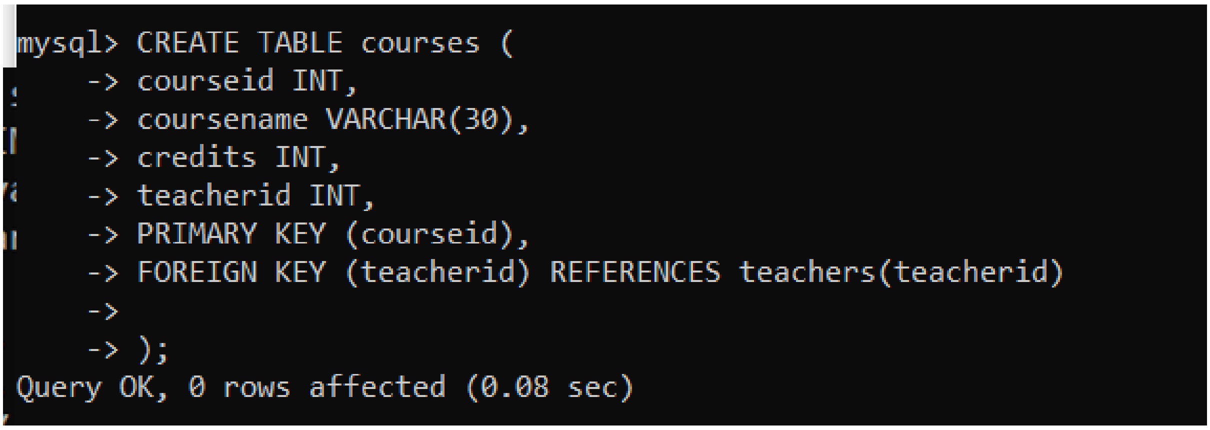
2



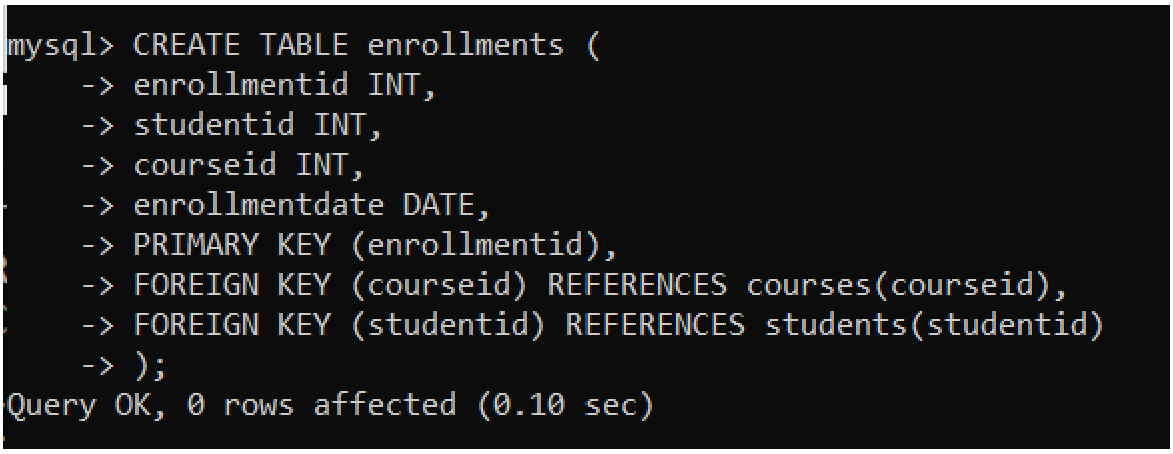
```
mysql> CREATE TABLE students(
    ->   studentid INT,
    ->   firstname varchar(25),
    ->   lastname varchar(25),
    ->   date date,
    ->   email text,
    ->   phone text,
    ->   PRIMARY KEY (studentid)
    -> ) ;
Query OK, 0 rows affected (0.05 sec)
```



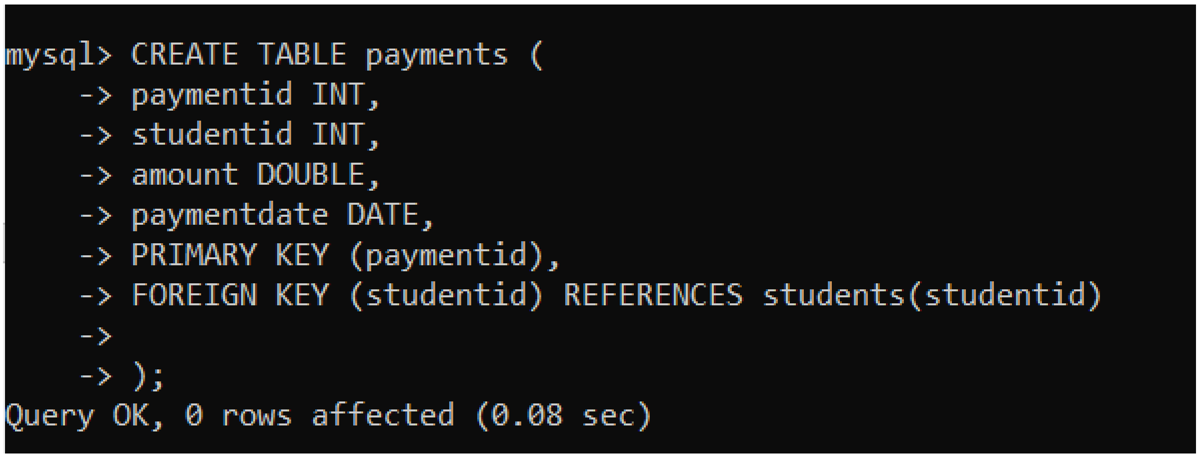
```
mysql> CREATE TABLE teachers(
    -> teacherid INT,
    -> firstname VARCHAR(30),
    -> lastname VARCHAR(30),
    -> email TEXT,
    -> PRIMARY KEY (teacherid)
    -> );
Query OK, 0 rows affected (0.05 sec)
```



```
mysql> CREATE TABLE courses (
    -> courseid INT,
    -> coursename VARCHAR(30),
    -> credits INT,
    -> teacherid INT,
    -> PRIMARY KEY (courseid),
    -> FOREIGN KEY (teacherid) REFERENCES teachers(teacherid)
    ->
    -> );
Query OK, 0 rows affected (0.08 sec)
```

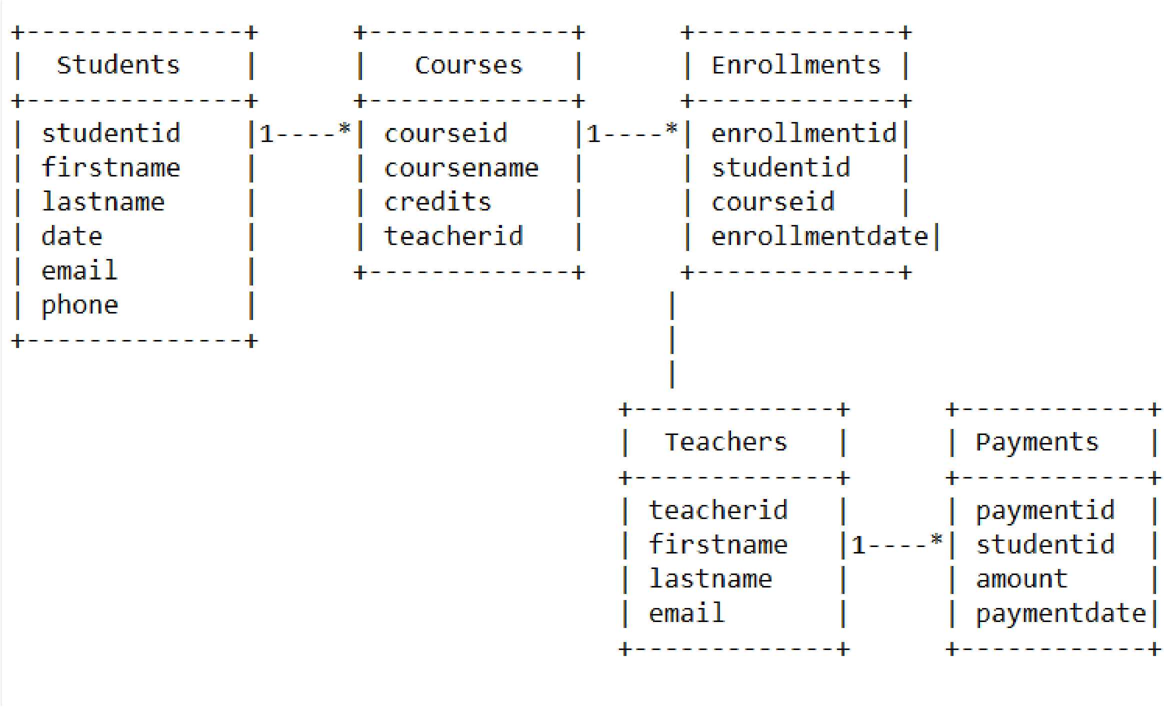


```
mysql> CREATE TABLE enrollments (
    -> enrollmentid INT,
    -> studentid INT,
    -> courseid INT,
    -> enrollmentdate DATE,
    -> PRIMARY KEY (enrollmentid),
    -> FOREIGN KEY (courseid) REFERENCES courses(courseid),
    -> FOREIGN KEY (studentid) REFERENCES students(studentid)
    -> );
Query OK, 0 rows affected (0.10 sec)
```

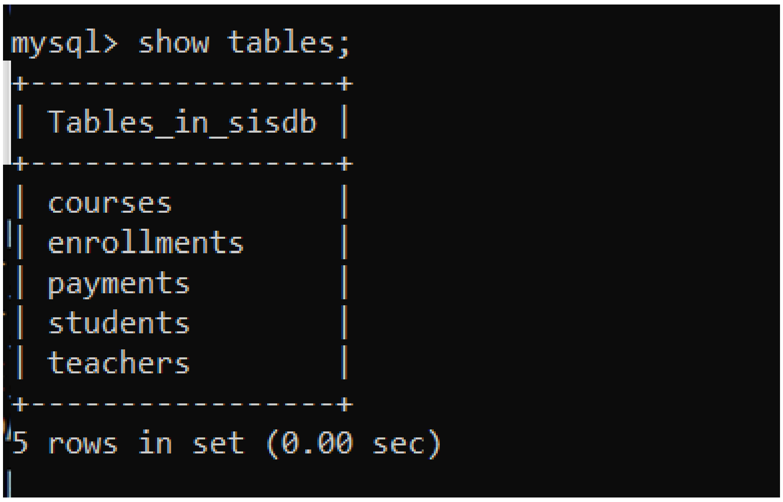


```
mysql> CREATE TABLE payments (
    -> paymentid INT,
    -> studentid INT,
    -> amount DOUBLE,
    -> paymentdate DATE,
    -> PRIMARY KEY (paymentid),
    -> FOREIGN KEY (studentid) REFERENCES students(studentid)
    ->
    -> );
Query OK, 0 rows affected (0.08 sec)
```

3)

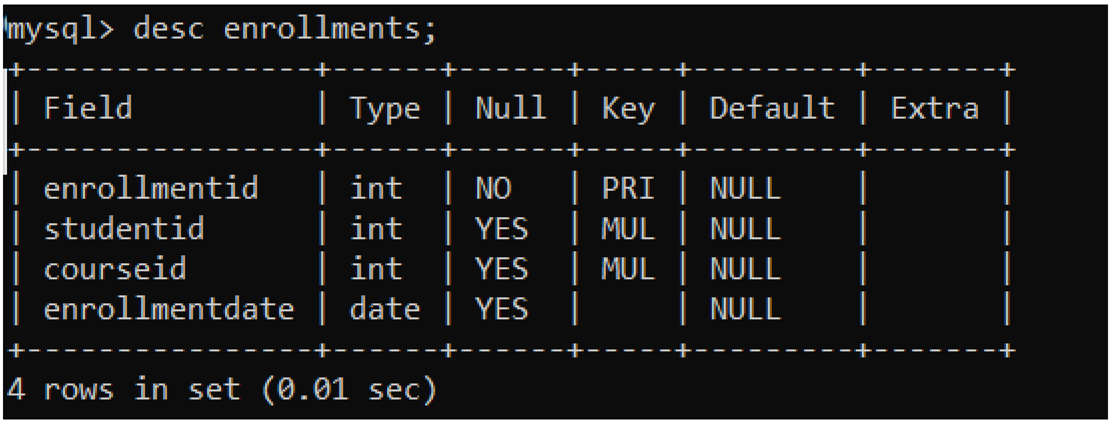


```
+-------------+      +-------------+      +-------------+
|  Students    |      |   Courses    |      | Enrollments |
+-------------+      +-------------+      +-------------+
| studentid    |1----*| courseid     |1----*| enrollmentid|
| firstname    |      | coursename   |      | studentid   |
| lastname     |      | credits      |      | courseid    |
| date         |      | teacherid    |      | enrollmentdate|
| email        |      +-------------+      +-------------+
| phone        |                           |
+-------------+                            |
                                           |
                                       +-------------+      +------------+
                                       |  Teachers    |      | Payments   |
                                       +-------------+      +------------+
                                       | teacherid    |      | paymentid  |
                                       | firstname    |1----*| studentid  |
                                       | lastname     |      | amount     |
                                       | email        |      | paymentdate|
                                       +-------------+      +------------+
```

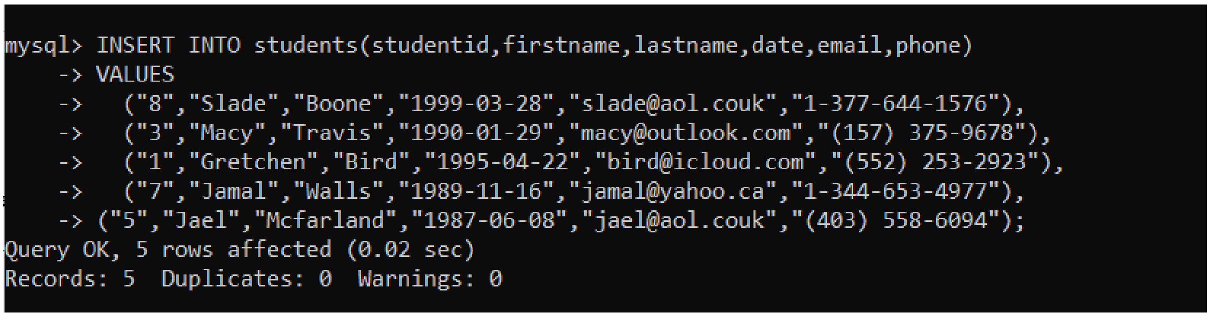


```
mysql> show tables;
+------------------+
| Tables_in_sisdb  |
+------------------+
| courses          |
| enrollments      |
| payments         |
| students         |
| teachers         |
+------------------+
5 rows in set (0.00 sec)
```

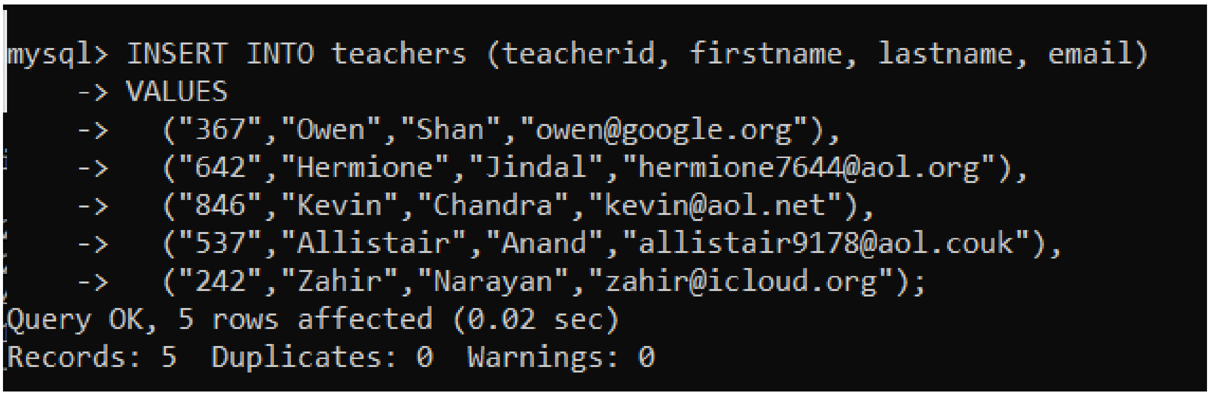
4)



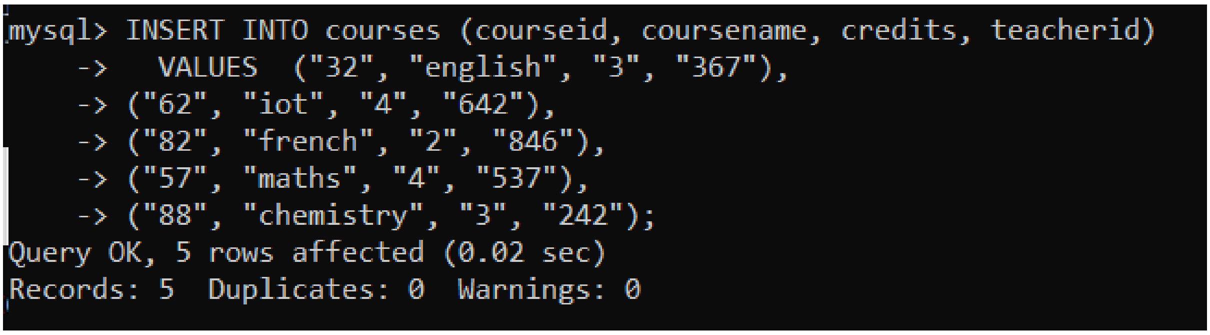
```
mysql> desc enrollments;
+----------------+------+------+-----+---------+-------+
| Field          | Type | Null | Key | Default | Extra |
+----------------+------+------+-----+---------+-------+
| enrollmentid   | int  | NO   | PRI | NULL    |       |
| studentid      | int  | YES  | MUL | NULL    |       |
| courseid       | int  | YES  | MUL | NULL    |       |
| enrollmentdate | date | YES  |     | NULL    |       |
+----------------+------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```



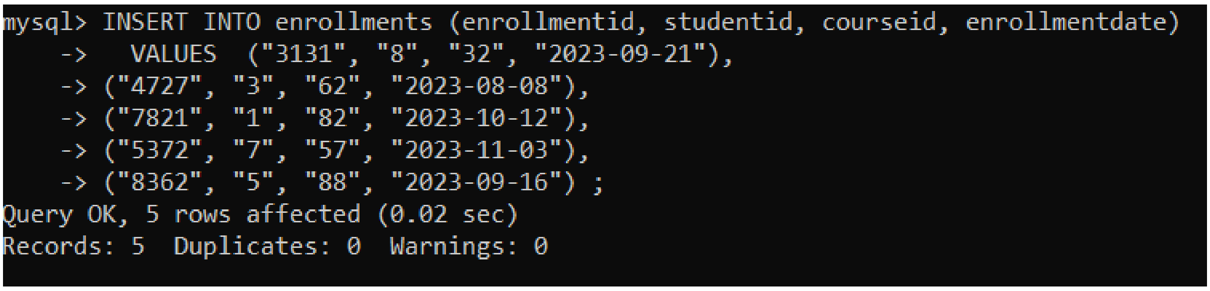
```
mysql> INSERT INTO students(studentid,firstname,lastname,date,email,phone)
    -> VALUES
    ->   ("8","Slade","Boone","1999-03-28","slade@aol.couk","1-377-644-1576"),
    ->   ("3","Macy","Travis","1990-01-29","macy@outlook.com","(157) 375-9678"),
    ->   ("1","Gretchen","Bird","1995-04-22","bird@icloud.com","(552) 253-2923"),
    ->   ("7","Jamal","Walls","1989-11-16","jamal@yahoo.ca","1-344-653-4977"),
    -> ("5","Jael","Mcfarland","1987-06-08","jael@aol.couk","(403) 558-6094");
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0
```



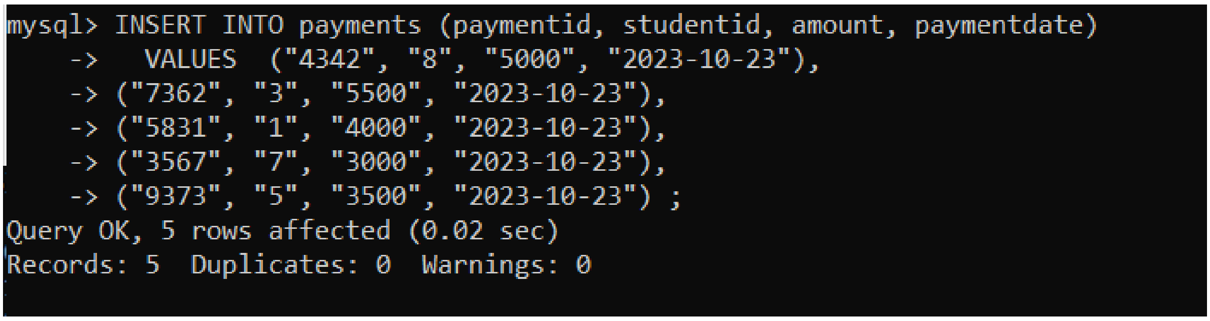
```
mysql> INSERT INTO teachers (teacherid, firstname, lastname, email)
    -> VALUES
    ->   ("367","Owen","Shan","owen@google.org"),
    ->   ("642","Hermione","Jindal","hermione7644@aol.org"),
    ->   ("846","Kevin","Chandra","kevin@aol.net"),
    ->   ("537","Allistair","Anand","allistair9178@aol.couk"),
    ->   ("242","Zahir","Narayan","zahir@icloud.org");
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0
```



```
mysql> INSERT INTO courses (courseid, coursename, credits, teacherid)
    ->   VALUES  ("32", "english", "3", "367"),
    -> ("62", "iot", "4", "642"),
    -> ("82", "french", "2", "846"),
    -> ("57", "maths", "4", "537"),
    -> ("88", "chemistry", "3", "242");
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0
```



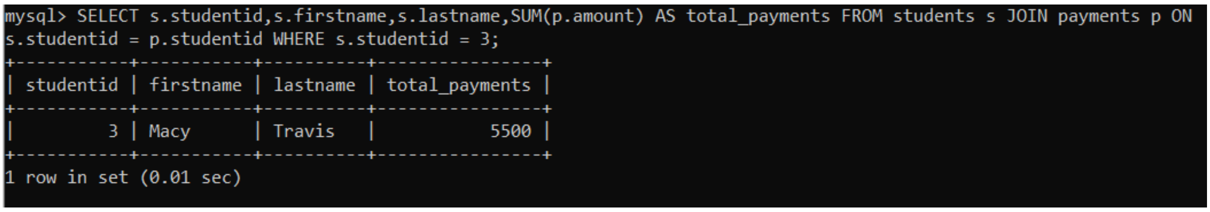
```
mysql> INSERT INTO enrollments (enrollmentid, studentid, courseid, enrollmentdate)
    ->   VALUES  ("3131", "8", "32", "2023-09-21"),
    -> ("4727", "3", "62", "2023-08-08"),
    -> ("7821", "1", "82", "2023-10-12"),
    -> ("5372", "7", "57", "2023-11-03"),
    -> ("8362", "5", "88", "2023-09-16") ;
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0
```



```
mysql> INSERT INTO payments (paymentid, studentid, amount, paymentdate)
    ->   VALUES  ("4342", "8", "5000", "2023-10-23"),
    -> ("7362", "3", "5500", "2023-10-23"),
    -> ("5831", "1", "4000", "2023-10-23"),
    -> ("3567", "7", "3000", "2023-10-23"),
    -> ("9373", "5", "3500", "2023-10-23") ;
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

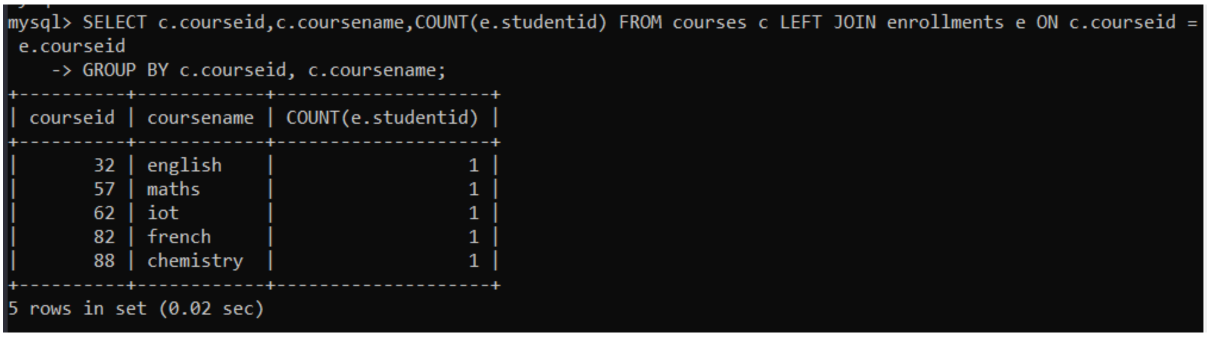
Task 2

1)



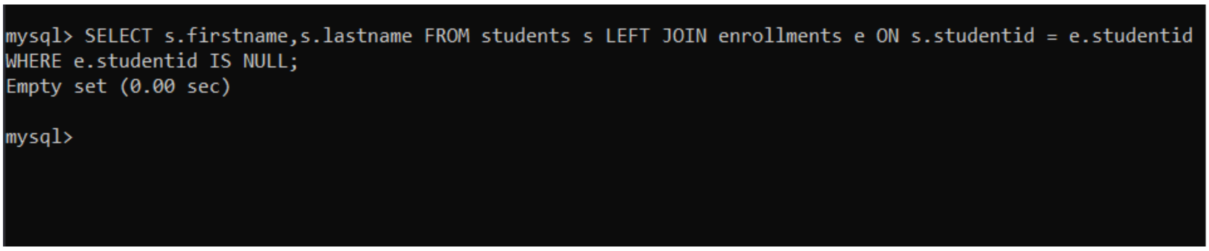
```
mysql> SELECT s.studentid,s.firstname,s.lastname,SUM(p.amount) AS total_payments FROM students s JOIN payments p ON
s.studentid = p.studentid WHERE s.studentid = 3;
+-----------+-----------+----------+----------------+
| studentid | firstname | lastname | total_payments |
+-----------+-----------+----------+----------------+
|         3 | Macy      | Travis   |           5500 |
+-----------+-----------+----------+----------------+
1 row in set (0.01 sec)
```

2)



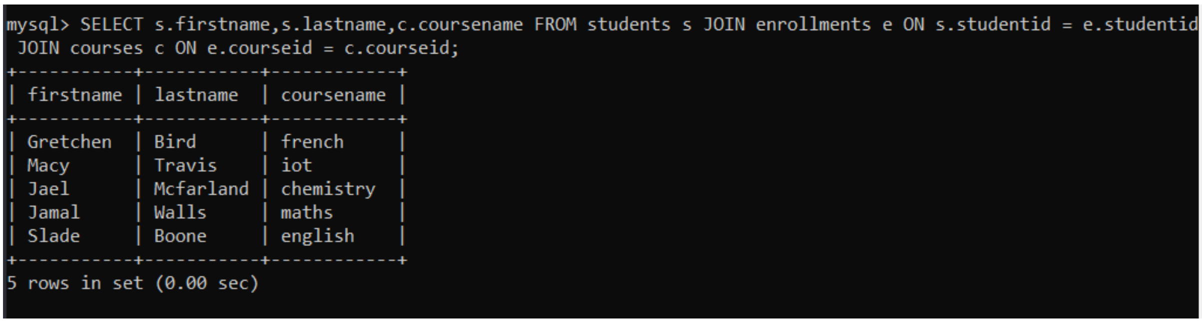
```
mysql> SELECT c.courseid,c.coursename,COUNT(e.studentid) FROM courses c LEFT JOIN enrollments e ON c.courseid =
 e.courseid
    -> GROUP BY c.courseid, c.coursename;
+----------+------------+--------------------+
| courseid | coursename | COUNT(e.studentid) |
+----------+------------+--------------------+
|       32 | english    |                  1 |
|       57 | maths      |                  1 |
|       62 | iot        |                  1 |
|       82 | french     |                  1 |
|       88 | chemistry  |                  1 |
+----------+------------+--------------------+
5 rows in set (0.02 sec)
```

3)



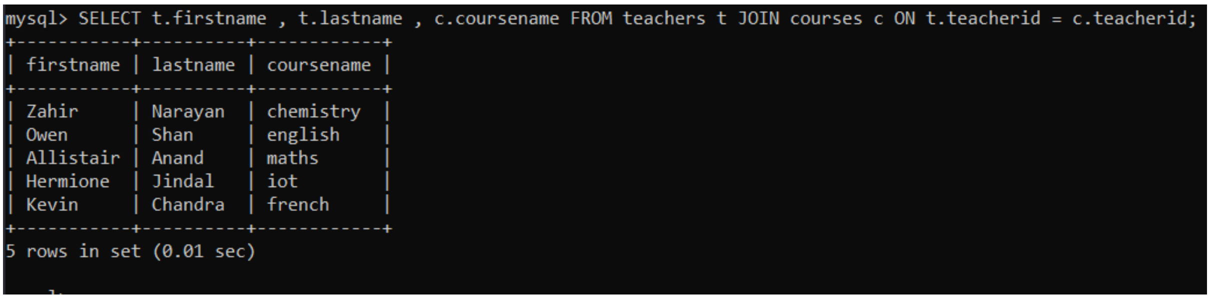
```
mysql> SELECT s.firstname,s.lastname FROM students s LEFT JOIN enrollments e ON s.studentid = e.studentid
WHERE e.studentid IS NULL;
Empty set (0.00 sec)

mysql>
```

4)



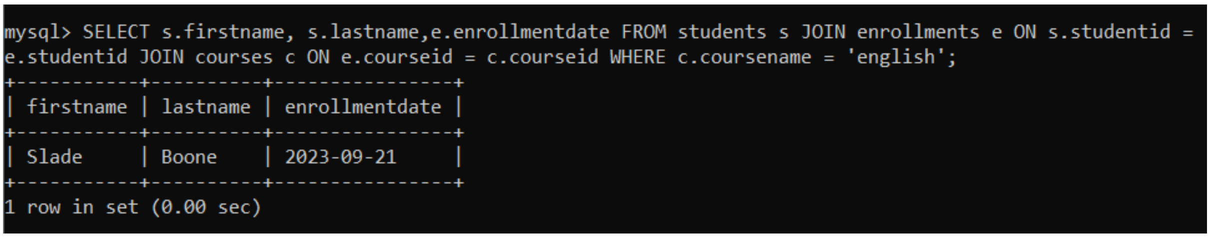
```
mysql> SELECT s.firstname,s.lastname,c.coursename FROM students s JOIN enrollments e ON s.studentid = e.studentid
 JOIN courses c ON e.courseid = c.courseid;
+-----------+-----------+------------+
| firstname | lastname  | coursename |
+-----------+-----------+------------+
| Gretchen  | Bird      | french     |
| Macy      | Travis    | iot        |
| Jael      | Mcfarland | chemistry  |
| Jamal     | Walls     | maths      |
| Slade     | Boone     | english    |
+-----------+-----------+------------+
5 rows in set (0.00 sec)
```

5)



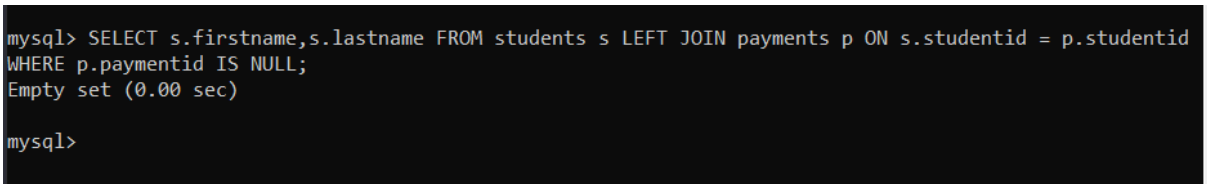
```
mysql> SELECT t.firstname , t.lastname , c.coursename FROM teachers t JOIN courses c ON t.teacherid = c.teacherid;
+-----------+----------+------------+
| firstname | lastname | coursename |
+-----------+----------+------------+
| Zahir     | Narayan  | chemistry  |
| Owen      | Shan     | english    |
| Allistair | Anand    | maths      |
| Hermione  | Jindal   | iot        |
| Kevin     | Chandra  | french     |
+-----------+----------+------------+
5 rows in set (0.01 sec)
```

6)



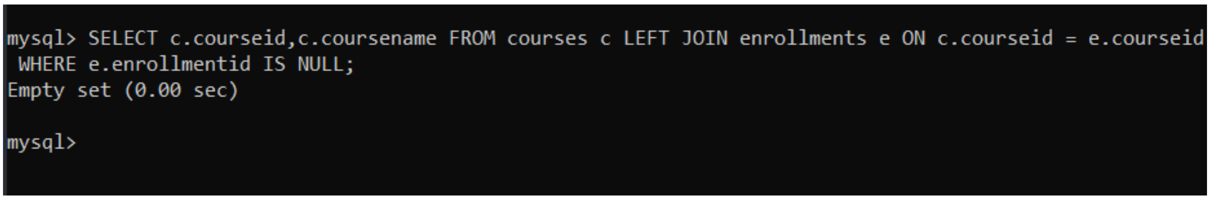
```
mysql> SELECT s.firstname, s.lastname,e.enrollmentdate FROM students s JOIN enrollments e ON s.studentid =
e.studentid JOIN courses c ON e.courseid = c.courseid WHERE c.coursename = 'english';
+-----------+----------+----------------+
| firstname | lastname | enrollmentdate |
+-----------+----------+----------------+
| Slade     | Boone    | 2023-09-21     |
+-----------+----------+----------------+
1 row in set (0.00 sec)
```

7)



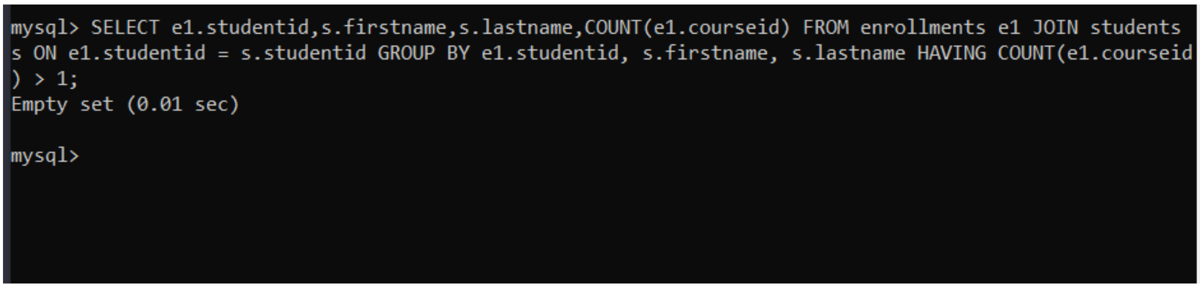
```
mysql> SELECT s.firstname,s.lastname FROM students s LEFT JOIN payments p ON s.studentid = p.studentid
WHERE p.paymentid IS NULL;
Empty set (0.00 sec)

mysql>
```

8)



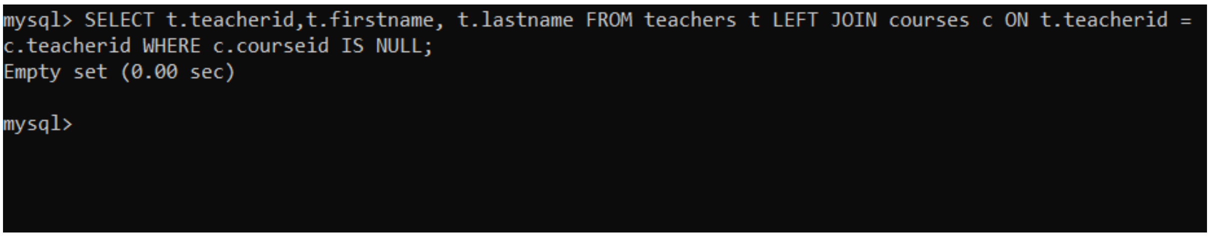
```
mysql> SELECT c.courseid,c.coursename FROM courses c LEFT JOIN enrollments e ON c.courseid = e.courseid
 WHERE e.enrollmentid IS NULL;
Empty set (0.00 sec)

mysql>
```

9)



```
mysql> SELECT e1.studentid,s.firstname,s.lastname,COUNT(e1.courseid) FROM enrollments e1 JOIN students
s ON e1.studentid = s.studentid GROUP BY e1.studentid, s.firstname, s.lastname HAVING COUNT(e1.courseid
) > 1;
Empty set (0.01 sec)

mysql>
```

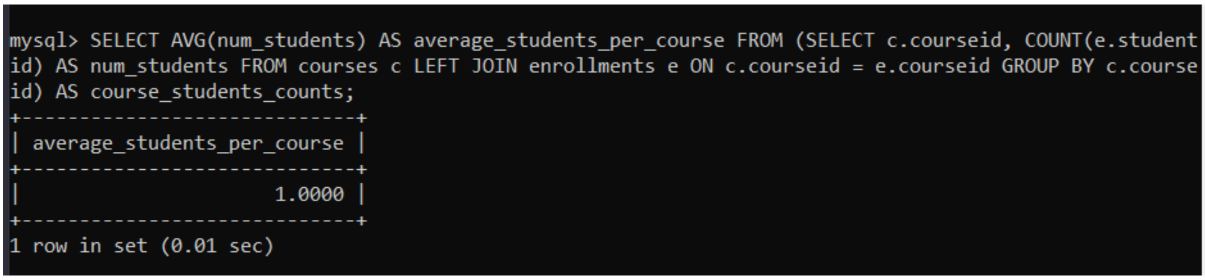
10)



```
mysql> SELECT t.teacherid,t.firstname, t.lastname FROM teachers t LEFT JOIN courses c ON t.teacherid =
c.teacherid WHERE c.courseid IS NULL;
Empty set (0.00 sec)

mysql>
```

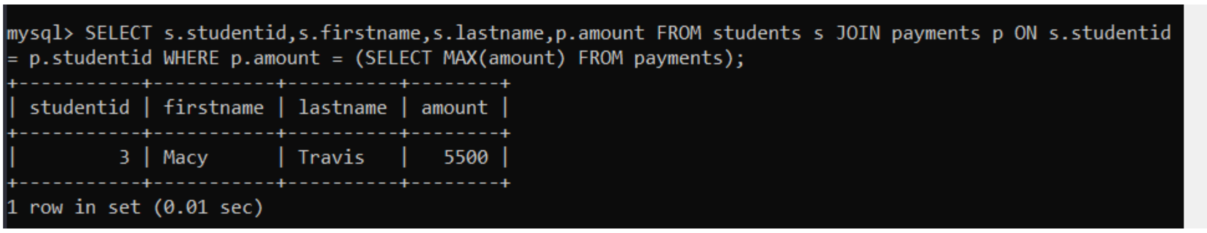
Task 3

1)



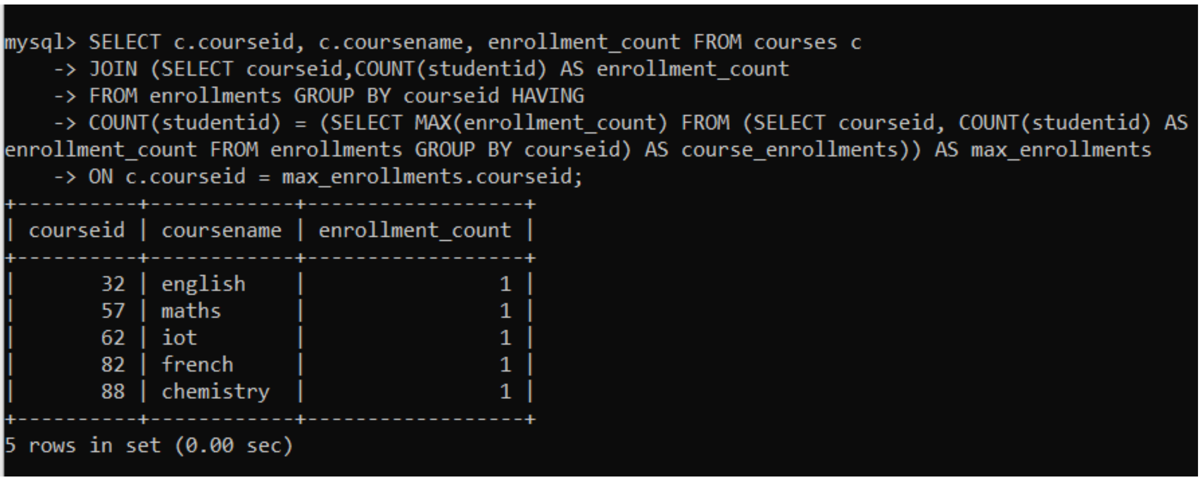
```
mysql> SELECT AVG(num_students) AS average_students_per_course FROM (SELECT c.courseid, COUNT(e.student
id) AS num_students FROM courses c LEFT JOIN enrollments e ON c.courseid = e.courseid GROUP BY c.course
id) AS course_students_counts;
+-----------------------------+
| average_students_per_course |
+-----------------------------+
|                      1.0000 |
+-----------------------------+
1 row in set (0.01 sec)
```

2)



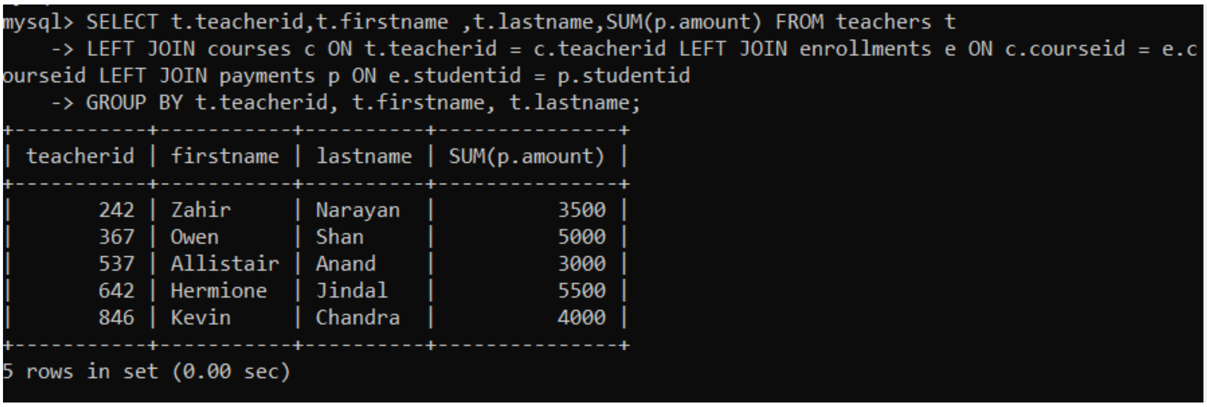
```
mysql> SELECT s.studentid,s.firstname,s.lastname,p.amount FROM students s JOIN payments p ON s.studentid
= p.studentid WHERE p.amount = (SELECT MAX(amount) FROM payments);
+-----------+-----------+----------+--------+
| studentid | firstname | lastname | amount |
+-----------+-----------+----------+--------+
|         3 | Macy      | Travis   |   5500 |
+-----------+-----------+----------+--------+
1 row in set (0.01 sec)
```

3)



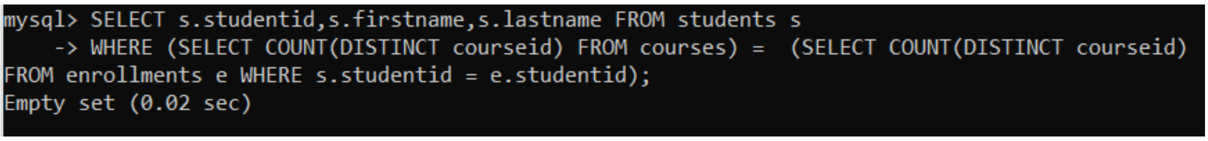
```
mysql> SELECT c.courseid, c.coursename, enrollment_count FROM courses c
    -> JOIN (SELECT courseid,COUNT(studentid) AS enrollment_count
    -> FROM enrollments GROUP BY courseid HAVING
    -> COUNT(studentid) = (SELECT MAX(enrollment_count) FROM (SELECT courseid, COUNT(studentid) AS
enrollment_count FROM enrollments GROUP BY courseid) AS course_enrollments)) AS max_enrollments
    -> ON c.courseid = max_enrollments.courseid;
+----------+------------+------------------+
| courseid | coursename | enrollment_count |
+----------+------------+------------------+
|       32 | english    |                1 |
|       57 | maths      |                1 |
|       62 | iot        |                1 |
|       82 | french     |                1 |
|       88 | chemistry  |                1 |
+----------+------------+------------------+
5 rows in set (0.00 sec)
```

4)



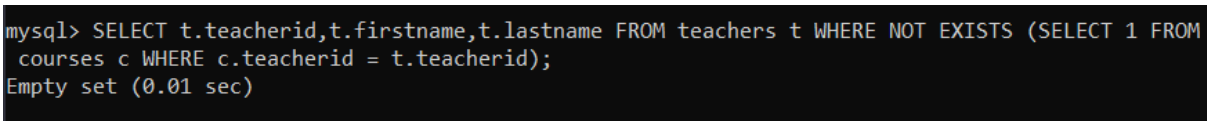
```
mysql> SELECT t.teacherid,t.firstname ,t.lastname,SUM(p.amount) FROM teachers t
    -> LEFT JOIN courses c ON t.teacherid = c.teacherid LEFT JOIN enrollments e ON c.courseid = e.c
ourseid LEFT JOIN payments p ON e.studentid = p.studentid
    -> GROUP BY t.teacherid, t.firstname, t.lastname;
+-----------+-----------+----------+---------------+
| teacherid | firstname | lastname | SUM(p.amount) |
+-----------+-----------+----------+---------------+
|       242 | Zahir     | Narayan  |          3500 |
|       367 | Owen      | Shan     |          5000 |
|       537 | Allistair | Anand    |          3000 |
|       642 | Hermione  | Jindal   |          5500 |
|       846 | Kevin     | Chandra  |          4000 |
+-----------+-----------+----------+---------------+
5 rows in set (0.00 sec)
```

5)



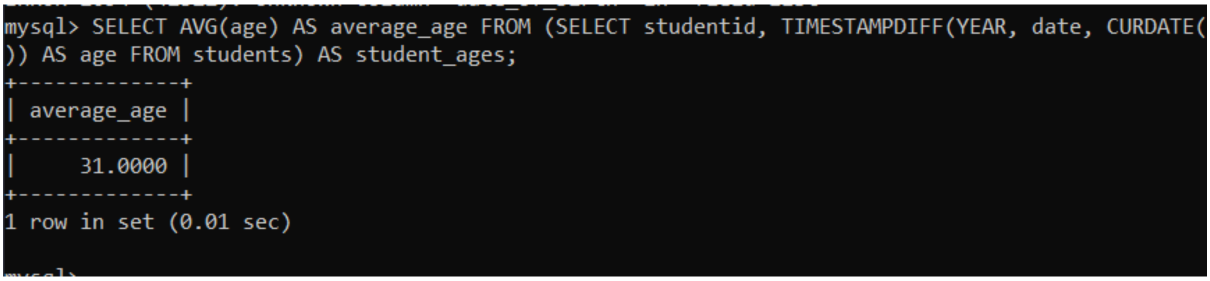
```
mysql> SELECT s.studentid,s.firstname,s.lastname FROM students s
    -> WHERE (SELECT COUNT(DISTINCT courseid) FROM courses) =  (SELECT COUNT(DISTINCT courseid)
FROM enrollments e WHERE s.studentid = e.studentid);
Empty set (0.02 sec)
```

6)



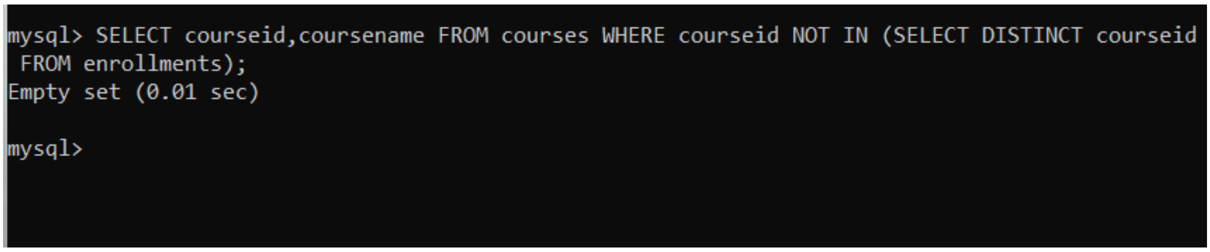
```
mysql> SELECT t.teacherid,t.firstname,t.lastname FROM teachers t WHERE NOT EXISTS (SELECT 1 FROM
 courses c WHERE c.teacherid = t.teacherid);
Empty set (0.01 sec)
```

7)



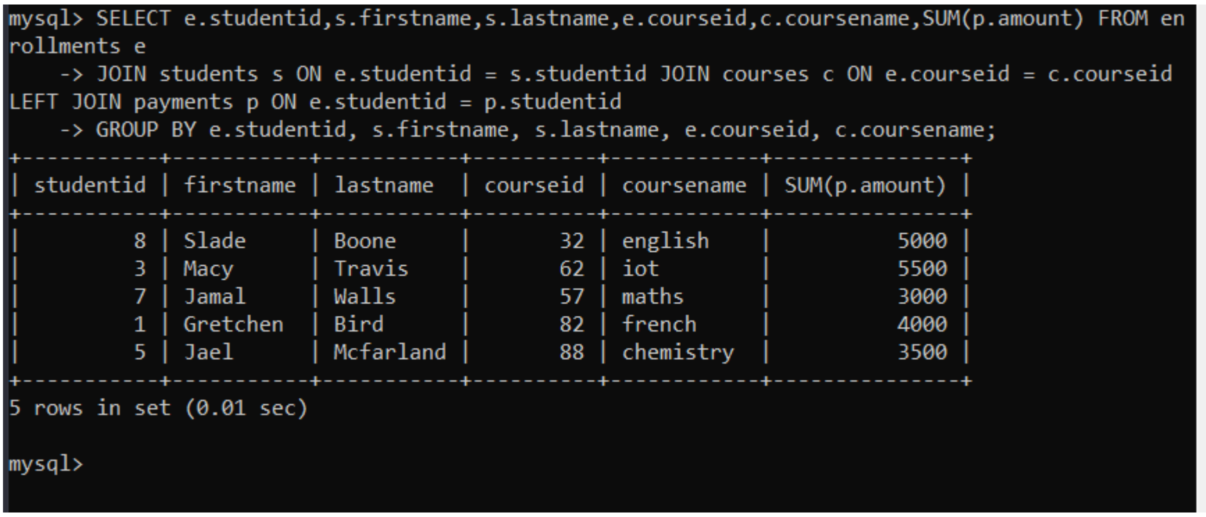
```
mysql> SELECT AVG(age) AS average_age FROM (SELECT studentid, TIMESTAMPDIFF(YEAR, date, CURDATE(
)) AS age FROM students) AS student_ages;
+-------------+
| average_age |
+-------------+
|     31.0000 |
+-------------+
1 row in set (0.01 sec)
```

8)



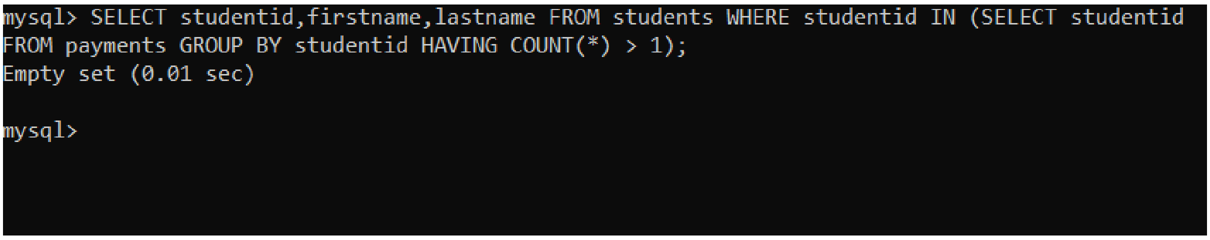
```
mysql> SELECT courseid,coursename FROM courses WHERE courseid NOT IN (SELECT DISTINCT courseid
 FROM enrollments);
Empty set (0.01 sec)

mysql>
```

9)



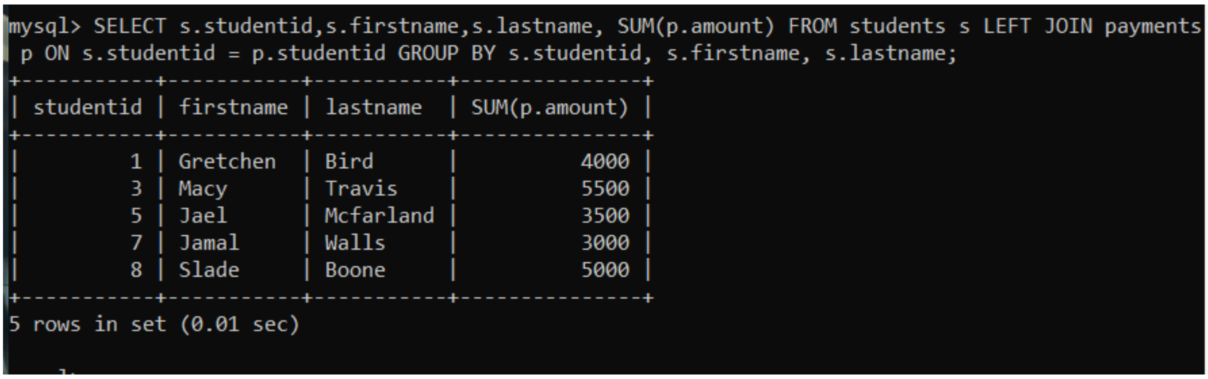
```
mysql> SELECT e.studentid,s.firstname,s.lastname,e.courseid,c.coursename,SUM(p.amount) FROM en
rollments e
    -> JOIN students s ON e.studentid = s.studentid JOIN courses c ON e.courseid = c.courseid
LEFT JOIN payments p ON e.studentid = p.studentid
    -> GROUP BY e.studentid, s.firstname, s.lastname, e.courseid, c.coursename;
+-----------+-----------+-----------+----------+------------+---------------+
| studentid | firstname | lastname  | courseid | coursename | SUM(p.amount) |
+-----------+-----------+-----------+----------+------------+---------------+
|         8 | Slade     | Boone     |       32 | english    |          5000 |
|         3 | Macy      | Travis    |       62 | iot        |          5500 |
|         7 | Jamal     | Walls     |       57 | maths      |          3000 |
|         1 | Gretchen  | Bird      |       82 | french     |          4000 |
|         5 | Jael      | Mcfarland |       88 | chemistry  |          3500 |
+-----------+-----------+-----------+----------+------------+---------------+
5 rows in set (0.01 sec)

mysql>
```

10)



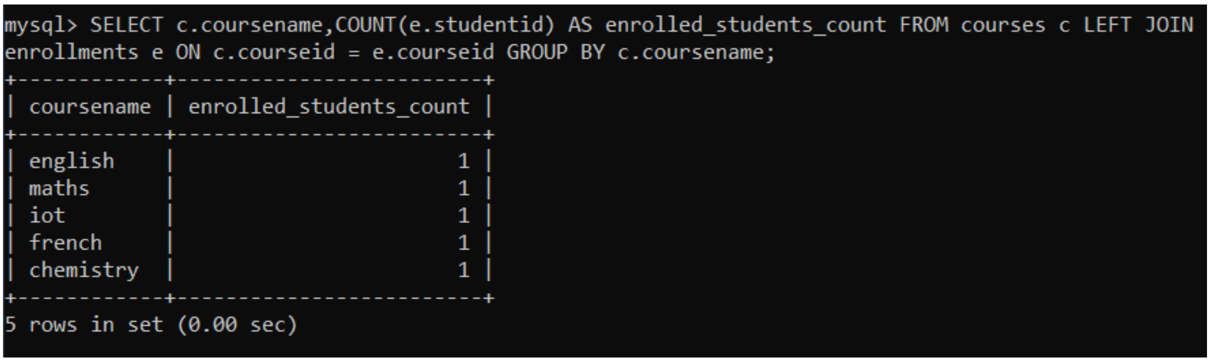
```
mysql> SELECT studentid,firstname,lastname FROM students WHERE studentid IN (SELECT studentid
FROM payments GROUP BY studentid HAVING COUNT(*) > 1);
Empty set (0.01 sec)

mysql>
```

11)



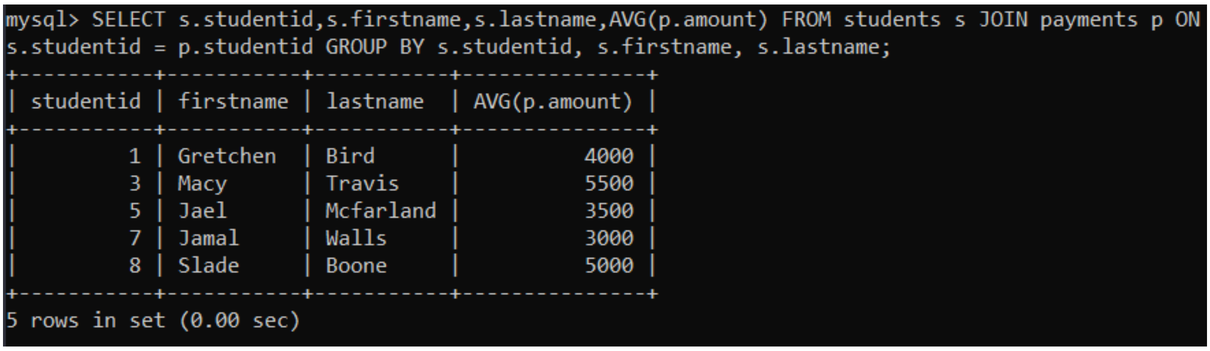
```
mysql> SELECT s.studentid,s.firstname,s.lastname, SUM(p.amount) FROM students s LEFT JOIN payments
 p ON s.studentid = p.studentid GROUP BY s.studentid, s.firstname, s.lastname;
+-----------+-----------+-----------+---------------+
| studentid | firstname | lastname  | SUM(p.amount) |
+-----------+-----------+-----------+---------------+
|         1 | Gretchen  | Bird      |          4000 |
|         3 | Macy      | Travis    |          5500 |
|         5 | Jael      | Mcfarland |          3500 |
|         7 | Jamal     | Walls     |          3000 |
|         8 | Slade     | Boone     |          5000 |
+-----------+-----------+-----------+---------------+
5 rows in set (0.01 sec)
```

12)



```
mysql> SELECT c.coursename,COUNT(e.studentid) AS enrolled_students_count FROM courses c LEFT JOIN
enrollments e ON c.courseid = e.courseid GROUP BY c.coursename;
+------------+-------------------------+
| coursename | enrolled_students_count |
+------------+-------------------------+
| english    |                       1 |
| maths      |                       1 |
| iot        |                       1 |
| french     |                       1 |
| chemistry  |                       1 |
+------------+-------------------------+
5 rows in set (0.00 sec)
```

13)



```
mysql> SELECT s.studentid,s.firstname,s.lastname,AVG(p.amount) FROM students s JOIN payments p ON
s.studentid = p.studentid GROUP BY s.studentid, s.firstname, s.lastname;
+-----------+-----------+-----------+---------------+
| studentid | firstname | lastname  | AVG(p.amount) |
+-----------+-----------+-----------+---------------+
|         1 | Gretchen  | Bird      |          4000 |
|         3 | Macy      | Travis    |          5500 |
|         5 | Jael      | Mcfarland |          3500 |
|         7 | Jamal     | Walls     |          3000 |
|         8 | Slade     | Boone     |          5000 |
+-----------+-----------+-----------+---------------+
5 rows in set (0.00 sec)
```