Write SQL statements for the following tasks using MySQL:
Create a database called lab2

```
CREATE DATABASE lab2
```

```
mysql> SHOW DATABASES
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| lab2               |
| mysql              |
| performance_schema |
| sys                |
| TEST_1             |
| TEST_2             |
+--------------------+
7 rows in set (0.01 sec)
```

Create three tables as following in the database lab2

- Students, three columns: first_name, last_name, student_id. The first two column should be text; the last one is an integer. Use student_id as the primary key.

```
mysql> CREATE TABLE Students
    -> (
    -> first_name TEXT(40) NOT NULL,
    -> last_name TEXT(40) NOT NULL,
    -> student_id INT NOT NULL,
    -> PRIMARY KEY (student_id)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> SHOW TABLES;
+----------------+
| Tables_in_lab2 |
+----------------+
| Students       |
+----------------+
1 row in set (0.01 sec)
```

- Courses, two columns: course_title, course_id. They are both text. course_id should be the primary key.

```
mysql> CREATE TABLE Courses
    -> (
    -> Course_Title TEXT(40) NOT NULL,
    -> Course_ID TEXT(10) NOT NULL,
    -> PRIMARY KEY (Course_ID(10))
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> SHOW TABLES;
+----------------+
| Tables_in_lab2 |
+----------------+
| Courses        |
| Students       |
+----------------+
2 rows in set (0.01 sec)
```

- Registration, three columns: student_id, a foreign key referencing student_id in the Students table; course_id, a foreign key referencing course_id in the Courses table; status, a text column that is meant to be 'registered', 'withdrawn', or 'pending permission', etc. You do not need to implement the constraint for the status column. The primary key should be the combination of the course_id and the student_id.

```
mysql> CREATE TABLE Registration
    -> (
    ->
    -> student_id INT NOT NULL,
    -> Course_ID TEXT(10) NOT NULL,
    -> Status TEXT(40) NOT NULL,
    -> PRIMARY KEY (student_id, Course_ID(10))
    -> );
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> SHOW TABLES;
+----------------+
| Tables_in_lab2 |
+----------------+
| Courses        |
| Registration   |
| Students       |
+----------------+
3 rows in set (0.01 sec)
```

```
mysql> ALTER TABLE Registration ADD FOREIGN KEY (student_id) REFERENCES Students(student_id);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> ALTER TABLE Registration ADD FOREIGN KEY (Course_ID(10)) REFERENCES
Courses(Course_ID);
```

(Note– Professor, kept getting the following error when I was putting the Foreign key in the CREATE statement–
ERROR 1089 (HY000): Incorrect prefix key; the used key part isn't a string, the used length is longer than the key part, or the storage engine doesn't support unique prefix keys.
It went away when I defined them in the ALTER statement, after a bit of research on the web)

- Add 5 students in the Students table. Choose any names as you like.

```
mysql> INSERT INTO
    -> Students (first_name, last_name, student_id)
    -> VALUES
    -> ('Mick', 'Jagger', '1000'),
    -> ('Keith', 'Richards', '1001'),
    -> ('Brian', 'Jones', '1002'),
    -> ('Ronnie', 'Wood', '1003'),
    -> ('Tony', 'Chapman', '1004');
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM STUDENTS;
+------------+-----------+------------+
| first_name | last_name | student_id |
+------------+-----------+------------+
| Mick       | Jagger    |       1000 |
| Keith      | Richards  |       1001 |
| Brian      | Jones     |       1002 |
| Ronnie     | Wood      |       1003 |
| Tony       | Chapman   |       1004 |
+------------+-----------+------------+
5 rows in set (0.00 sec)
```

- Add 2 courses in the Courses table. Choose any names as you like.

```
mysql> INSERT INTO
    -> Courses (course_title, course_id)
    -> VALUES
    -> ('SCRIPTING and SQL' , 'CISC5500'),
    -> ('PROGRAMMING WITH PYTHON', 'CISC5200');
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM Courses;
+-------------------------+-----------+
| Course_Title            | Course_ID |
+-------------------------+-----------+
| PROGRAMMING WITH PYTHON | CISC5200  |
| SCRIPTING and SQL       | CISC5500  |
+-------------------------+-----------+
2 rows in set (0.00 sec)
```

- Register some students with both courses; some with each of the two course; and some with no course.

```
mysql> INSERT INTO
    -> Registration (student_id, Course_ID, Status)
    -> VALUES
    -> ('1000', 'CISC5200', 'Registered'),
    -> ('1000', 'CISC5500', 'Registered'),
    -> ('1001', 'CISC5500', 'Withdrawn'),
    -> ('1001', 'CISC5200', 'Pending'),
    -> ('1002', 'CISC5500', 'Registered'),
    -> ('1002', 'CISC5200', 'Pending'),
    -> ('1003', 'CISC5500', 'Withdrawn'),
    -> ('1003', 'CISC5200', 'Withdrawn'),
    -> ('1004', 'CISC5500', 'Registered'),
    -> ('1004', 'CISC5200', 'Registered');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM Registration;
+------------+-----------+------------+
| student_id | Course_ID | Status     |
+------------+-----------+------------+
|       1000 | CISC5200  | Registered |
|       1000 | CISC5500  | Registered |
|       1001 | CISC5200  | Pending    |
|       1001 | CISC5500  | Withdrawn  |
|       1002 | CISC5200  | Pending    |
|       1002 | CISC5500  | Registered |
|       1003 | CISC5200  | Withdrawn  |
|       1003 | CISC5500  | Withdrawn  |
|       1004 | CISC5200  | Registered |
|       1004 | CISC5500  | Registered |
+------------+-----------+------------+
10 rows in set (0.00 sec)
```

- Find out the total number of students in each course. Be ready to do some other simple queries on this database.

```
mysql> SELECT Course_ID, COUNT(*) AS Number
    -> FROM Registration
    -> WHERE Status = 'Registered'
    -> GROUP BY COURSE_ID;
+-----------+--------+
| Course_ID | Number |
+-----------+--------+
| CISC5200  |      2 |
| CISC5500  |      3 |
+-----------+--------+
2 rows in set (0.00 sec)
```