Registration Database

Introduction

In this exercise you will practice

- basic relational data modeling,
- writing SQL create table statements,
- writing SQL schema and data insertion scripts, and
- writing basic SQL queries.

Problem Description

You work for the registrar of a major university and have been tasked with creating a database to hold courses and semester schedules. Create a **new word document and save it with your name and student number.** Now execute the following commands and write down the effect of each command in the document. You should also put screen shot of the result you get from a query (use the snipping tool software installed on most of the machines in lab). You should also **export your database. Zip the files and upload to Moodle.**

Part 1: Creating the database

- Created and using database registration.
  # Created a new database called registration and started using database registration

```
mysql> CREATE DATABASE registration;
Query OK, 1 row affected (0.00 sec)

mysql> USE registration;
Database changed
```

- student(student_id, name, gpa)
  # Created a table student providing datatypes (int, varchar and decimal) for each column. I have made student_id as the primary key which has to be NOT NULL and UNIQUE.

```
mysql> CREATE TABLE student(student_id INT NOT NULL UNIQUE,name VARCHAR(20),gpa DECIMAL(2,1),P
RIMARY KEY(student_id));
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DESCRIBE student;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| student_id | int(11)      | NO   | PRI | NULL    |       |
| name       | varchar(20)  | YES  |     | NULL    |       |
| gpa        | decimal(2,1) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

- dept(dept_id, name, dean, building, room)

  # Created a table dept providing datatypes (varchar,int) for each column. I have made dept_id as the primary key which has to be NOT NULL.

```
mysql> CREATE TABLE dept(dept_id VARCHAR(20) NOT NULL,name VARCHAR(30),dean VARCHAR(20),buildi
ng VARCHAR(20),room INT,PRIMARY KEY(dept_id));
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DESCRIBE dept;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| dept_id  | varchar(20) | NO   | PRI | NULL    |       |
| name     | varchar(30) | YES  |     | NULL    |       |
| dean     | varchar(20) | YES  |     | NULL    |       |
| building | varchar(20) | YES  |     | NULL    |       |
| room     | int(11)     | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

- course(dept_id, course_id, name, hours), where dept_id is a foreign key referencing the dept(dept_id)

  # Created a table course providing datatypes (varchar,int) for each column. I have made dept_id and course_id as a composite primary key which has to be NOT NULL. I have reference dept_id to the dept table.

```
mysql> CREATE TABLE course(dept_id VARCHAR(20) NOT NULL,course_id INT NOT NULL,name VARCHAR(20
),hours INT DEFAULT 3,PRIMARY KEY(dept_id,course_id),FOREIGN KEY(dept_id) REFERENCES dept(dept
_id));
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DESCRIBE course;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| dept_id   | varchar(20) | NO   | PRI | NULL    |       |
| course_id | int(11)     | NO   | PRI | NULL    |       |
| name      | varchar(20) | YES  |     | NULL    |       |
| hours     | int(11)     | YES  |     | 3       |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

- enrolled(dept_id, course_id, student_id), where (dept_id, course_id) is a foreign key referencing course(dept_id, course_id) and student_id is a foreign key referencing student(student_id)

  # Created a table enrolled providing datatypes (varchar,int) for each column. This table has all foreign keys. These foreign key are referencing the course and student. The foreign keys also updates and delete cascades to the other related tables.

```
mysql> CREATE TABLE enrolled(dept_id VARCHAR(20) NOT NULL,course_id INT NOT NULL,student_id I
NT NOT NULL,FOREIGN KEY(dept_id, course_id) REFERENCES course(dept_id, course_id) ON DELETE C
ASCADE ON UPDATE CASCADE,FOREIGN KEY(student_id) REFERENCES student(student_id) ON DELETE CAS
CADE ON UPDATE CASCADE);
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> DESCRIBE enrolled;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| dept_id    | varchar(20) | NO   | MUL | NULL    |       |
| course_id  | int(11)     | NO   |     | NULL    |       |
| student_id | int(11)     | NO   | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

- Show tables in database registration.

  # Showing the tables created in the registration database.

```
mysql> SHOW TABLES;
+------------------------+
| Tables_in_registration |
+------------------------+
| course                 |
| dept                   |
| enrolled               |
| student                |
+------------------------+
4 rows in set (0.00 sec)
```

Part 2: Populating the database

Write a SQL script that populates the data for Student, Enrolled, Department and Course:

- Student

  # Inserting values into students and showing student tables.

```
mysql> INSERT INTO student VALUES(11,'Bush',3.0),(12,'Cruz',3.2),(13,'Clinton',3.9),(22,'Sande
rs',3.0),(33,'Trump',3.8);
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> Select * from student;
+------------+---------+------+
| student_id | name    | gpa  |
+------------+---------+------+
|         11 | Bush    |  3.0 |
|         12 | Cruz    |  3.2 |
|         13 | Clinton |  3.9 |
|         22 | Sanders |  3.0 |
|         33 | Trump   |  3.8 |
+------------+---------+------+
5 rows in set (0.00 sec)
```

- Enrolled

  # Inserting values into enrolled and showing enrolled tables. All foreign key values must equal to its reference table.

```
mysql> INSERT INTO enrolled VALUES('CS',101,11),('Math',101,11),('CS',101,12),('CS',201,22),('
Math',201,33),('EE',102,33);
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM enrolled;
+---------+-----------+------------+
| dept_id | course_id | student_id |
+---------+-----------+------------+
| CS      |       101 |         11 |
| Math    |       101 |         11 |
| CS      |       101 |         12 |
| CS      |       201 |         22 |
| Math    |       201 |         33 |
| EE      |       102 |         33 |
+---------+-----------+------------+
6 rows in set (0.00 sec)
```

- Department

Inserting values into department and showing department tables.

```
mysql> INSERT INTO dept VALUES('CS','Computer Science','Rubio','Ajax',100),('Math','Mathmagics
','Carson','Acme',300),('EE','Electrical Engineering','Kasich','Ajax',200),('IE','Industrial E
ngineering','Cruz',' ',200),('Music','Musicology','Costello','North',100);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM dept;
+---------+-----------------------+----------+----------+------+
| dept_id | name                  | dean     | building | room |
+---------+-----------------------+----------+----------+------+
| CS      | Computer Science      | Rubio    | Ajax     |  100 |
| EE      | Electrical Engineering | Kasich   | Ajax     |  200 |
| IE      | Industrial Engineering | Cruz     |          |  200 |
| Math    | Mathmagics            | Carson   | Acme     |  300 |
| Music   | Musicology            | Costello | North    |  100 |
+---------+-----------------------+----------+----------+------+
5 rows in set (0.00 sec)
```

- Course

Inserting values into course and showing course tables.

```
mysql> INSERT INTO course VALUES('CS',101,'Programming',4),('CS',201,'Algorithms',3),('CS',202
,'Systems',3),('Math',101,'Algebra',3),('Math',201,'Calculus',4),('Math',301,'Analysis',4),('M
usic',104,'Jazz',3),('EE',102,'Circuits',3),('IE',101,'Proabability',3),('IE',102,'Statistics'
,3);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM course;
+---------+-----------+-------------+-------+
| dept_id | course_id | name        | hours |
+---------+-----------+-------------+-------+
| CS      |       101 | Programming |     4 |
| CS      |       201 | Algorithms  |     3 |
| CS      |       202 | Systems     |     3 |
| EE      |       102 | Circuits    |     3 |
| IE      |       101 | Proabability |     3 |
| IE      |       102 | Statistics  |     3 |
| Math    |       101 | Algebra     |     3 |
| Math    |       201 | Calculus    |     4 |
| Math    |       301 | Analysis    |     4 |
| Music   |       104 | Jazz        |     3 |
+---------+-----------+-------------+-------+
10 rows in set (0.00 sec)
```

Part 3: Updating the database

Once the database has been populated, make the following updates:

- Change the name of the Math dept to 'Mathematics'
  # Using the update command to change the name value Mathmagic into Mathematics.

```
mysql> UPDATE dept SET name='Mathematics' WHERE dept_id = 'Math';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM dept
    -> ;
+---------+------------------------+---------+----------+------+
| dept_id | name                   | dean    | building | room |
+---------+------------------------+---------+----------+------+
| CS      | Computer Science       | Rubio   | Ajax     |  100 |
| EE      | Electrical Engineering | Kasich  | Ajax     |  200 |
| IE      | Industrial Engineering | Cruz    |          |  200 |
| Math    | Mathematics            | Carson  | Acme     |  300 |
| Music   | Musicology             | Costello | North   |  100 |
+---------+------------------------+---------+----------+------+
5 rows in set (0.00 sec)
```

- Change the name of the Music dept to 'Rock n Roll'
  # Using the update command to change the name value Jazz into Rock n Roll.

```
mysql> UPDATE dept SET name='Rock n Roll' WHERE dept_id = 'Music';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM dept;
+---------+------------------------+---------+----------+------+
| dept_id | name                   | dean    | building | room |
+---------+------------------------+---------+----------+------+
| CS      | Computer Science       | Rubio   | Ajax     |  100 |
| EE      | Electrical Engineering | Kasich  | Ajax     |  200 |
| IE      | Industrial Engineering | Cruz    |          |  200 |
| Math    | Mathematics            | Carson  | Acme     |  300 |
| Music   | Rock n Roll            | Costello | North   |  100 |
+---------+------------------------+---------+----------+------+
5 rows in set (0.00 sec)
```

- Add an IE course, IE 202 Simulation, 3 hours

  # Using the insert command to add another row for course IE

```
mysql> INSERT INTO course VALUES('IE',202,'Simulation',3);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM course;
+---------+-----------+--------------+-------+
| dept_id | course_id | name         | hours |
+---------+-----------+--------------+-------+
| CS      |       101 | Programming  |     4 |
| CS      |       201 | Algorithms   |     3 |
| CS      |       202 | Systems      |     3 |
| EE      |       102 | Circuits     |     3 |
| IE      |       101 | Proabability |     3 |
| IE      |       102 | Statistics   |     3 |
| IE      |       202 | Simulation   |     3 |
| Math    |       101 | Algebra      |     3 |
| Math    |       201 | Calculus     |     4 |
| Math    |       301 | Analysis     |     4 |
| Music   |       104 | Jazz         |     3 |
+---------+-----------+--------------+-------+
11 rows in set (0.00 sec)
```

Part 4: Querying the database

Write a sql script with queries that answer the following questions.

Easy Queries:

- What are all the departments?

  # Select columns dept_id and name from the dept table

```
mysql> SELECT dept_id, name FROM dept;
+---------+------------------------+
| dept_id | name                   |
+---------+------------------------+
| CS      | Computer Science       |
| EE      | Electrical Engineering |
| IE      | Industrial Engineering |
| Math    | Mathematics            |
| Music   | Rock n Roll            |
+---------+------------------------+
5 rows in set (0.00 sec)
```

- What are the names of all the department deans?

  # Select columns name and dean from the dept table

```
mysql> SELECT name, dean FROM dept;
+------------------------+----------+
| name                   | dean     |
+------------------------+----------+
| Computer Science       | Rubio    |
| Electrical Engineering | Kasich   |
| Industrial Engineering | Cruz     |
| Mathematics            | Carson   |
| Rock n Roll            | Costello |
+------------------------+----------+
5 rows in set (0.00 sec)
```

- What is the name of the dean of the CS dept?

  # Select columns dept_id, name and dean from the dept table.I have also used the WHERE clause to set a condition dept_id = 'CS'.

```
mysql> SELECT dept_id, name, dean FROM dept WHERE dept_id = 'CS';
+---------+------------------+-------+
| dept_id | name             | dean  |
+---------+------------------+-------+
| CS      | Computer Science | Rubio |
+---------+------------------+-------+
1 row in set (0.00 sec)
```

- What are all the course numbers, e.g., 'CS 2316'?

  # Select columns dept_id and course_id from the course table to show department ID and course number.

```
mysql> SELECT dept_id, course_id FROM course;
+---------+-----------+
| dept_id | course_id |
+---------+-----------+
| CS      |       101 |
| CS      |       201 |
| CS      |       202 |
| EE      |       102 |
| IE      |       101 |
| IE      |       102 |
| IE      |       202 |
| Math    |       101 |
| Math    |       201 |
| Math    |       301 |
| Music   |       104 |
+---------+-----------+
11 rows in set (0.00 sec)
```

- What are the course numbers of all the first-year courses?

  # Select columns course_id and name from the course table. I used the between operator to select numbers from 100 to 199.This will indicate first year courses.

```
mysql> SELECT course_id,name FROM course WHERE course_id BETWEEN 100 AND 199;
+-----------+--------------+
| course_id | name         |
+-----------+--------------+
|       101 | Programming  |
|       102 | Circuits     |
|       101 | Proabability |
|       102 | Statistics   |
|       101 | Algebra      |
|       104 | Jazz         |
+-----------+--------------+
6 rows in set (0.00 sec)
```

- What are the course numbers of all the CS courses?

  # Select column course_id from the course table and placed it into order.

```
mysql> SELECT course_id FROM course ORDER BY course_id;
+-----------+
| course_id |
+-----------+
|       101 |
|       101 |
|       101 |
|       102 |
|       102 |
|       104 |
|       201 |
|       201 |
|       202 |
|       202 |
|       301 |
+-----------+
11 rows in set (0.00 sec)
```

- What are all the CS and IE courses?

  # Select it all values from the course table and used a where clause with the condition dept_id =’CS’ or ‘IE’.

```
mysql> SELECT * FROM course WHERE dept_id = 'CS' OR dept_id = 'IE';
+---------+-----------+-------------+-------+
| dept_id | course_id | name        | hours |
+---------+-----------+-------------+-------+
| CS      |       101 | Programming |     4 |
| CS      |       201 | Algorithms  |     3 |
| CS      |       202 | Systems     |     3 |
| IE      |       101 | Proabability|     3 |
| IE      |       102 | Statistics  |     3 |
| IE      |       202 | Simulation  |     3 |
+---------+-----------+-------------+-------+
6 rows in set (0.01 sec)
```

Intermediate queries

- What are the names of the students enrolled in CS 101?

  # Joined tables student and enrolled with the columns student.name enrolled. dept_id and enrolled.course_id. Used the where clause with the conditions enrolled. dept_id = ‘CS’ and enrolled.course_id = 101 to find students enrolled in CS 101.

```
mysql> SELECT DISTINCT student.name, enrolled.dept_id, enrolled.course_id FROM student, enroll
ed WHERE enrolled.dept_id = 'CS' AND enrolled.course_id = 101;
+---------+---------+-----------+
| name    | dept_id | course_id |
+---------+---------+-----------+
| Bush    | CS      |       101 |
| Cruz    | CS      |       101 |
| Clinton | CS      |       101 |
| Sanders | CS      |       101 |
| Trump   | CS      |       101 |
+---------+---------+-----------+
5 rows in set (0.00 sec)
```

- What is the total enrolment of each department?

  # Used Left outer join for departments and enrolled. Providing information of dept_id in the department table and course_id in the enrolled table.I have grouped the information based on common value dept_id. I have used the count function to indicate the total number of enrolment of each department.

```
mysql> SELECT dept.dept_id,COUNT(course_id) FROM dept LEFT OUTER JOIN enrolled ON dept.dept_id
 = enrolled.dept_id GROUP BY dept.dept_id;
+---------+------------------+
| dept_id | COUNT(course_id) |
+---------+------------------+
| CS      |                3 |
| EE      |                1 |
| IE      |                0 |
| Math    |                2 |
| Music   |                0 |
+---------+------------------+
5 rows in set (0.01 sec)
```

- What are the names of the deans of departments that have zero enrolment?

  ## Used Left outer join for departments and enrolled. Providing information of dept_id and dean in the department table and course_id in the enrolled table.I have grouped the information based on common value dept_id. I have used the count function to indicate the total number of enrolment of each department. Used the having clause to indicate count (course_id) to equal zero.

```
mysql> SELECT dept.dept_id,dept.dean,COUNT(course_id) FROM dept LEFT OUTER JOIN enrolled ON de
pt.dept_id = enrolled.dept_id GROUP BY dept.dept_id HAVING COUNT(course_id) = 0;
+---------+----------+------------------+
| dept_id | dean     | COUNT(course_id) |
+---------+----------+------------------+
| IE      | Cruz     |                0 |
| Music   | Costello |                0 |
+---------+----------+------------------+
2 rows in set (0.00 sec)
```

- Which department(s) have the greatest enrolment?

  ## Used Left outer join for departments and enrolled. Providing information of dept_id in the department table and course_id in the enrolled table.I have grouped the information based on common value dept_id. I have used the count function to indicate the total number of enrolment of each department. I have used order by desc to show the greatest to the least enrolment.

```
mysql> SELECT dept.dept_id,COUNT(course_id) FROM dept LEFT OUTER JOIN enrolled ON dept.dept_id
 = enrolled.dept_id GROUP BY dept.dept_id ORDER BY count(course_id) DESC;
+---------+------------------+
| dept_id | COUNT(course_id) |
+---------+------------------+
| CS      |                3 |
| Math    |                2 |
| EE      |                1 |
| Music   |                0 |
| IE      |                0 |
+---------+------------------+
5 rows in set (0.00 sec)
```