**MySQL Labs**

**MySQL (Day1):**

|  |  |
| --- | --- |
|  | **Create a database called grades** |
| **create database grades;** |  |
| ***courses***  ***course\_id*** *int pk*  *course\_name varchar(100) not null*  *credit\_hour int*  ***students\_courses***  ***course\_id*** *int*  ***student\_id*** *int*  *grade int*  *reg\_date date*  ***students***  ***student\_id*** *int pk*  *student\_name varchar (100) not null*  *email varchar (50)*  *tel varchar (20)* | **Create the following tables in the grades database:** |
| **-create table students(students\_id int primary key, student\_name varchar(100) not null,email varchar(50),tel varchar(20));**  **- create table courses(course\_id int primary key,course\_name varchar(100) not null,credit\_hour int);**  **-create table students\_courses(course\_id int references course(course\_id),student\_id int references student(student\_id), grade int,reg\_date date);** |  |
| **3** | **Modify the students table to allow for longer Student names (150 char)**  **Confirm your modification.** |
| **-ALTER TABLE students MODIFY student\_name VARCHAR(150);** |  |
| **4** | **Add constraint to force unique email for each student** |
| **ALTER TABLE students**  **-> ADD CONSTRAINT unique\_email**  **-> UNIQUE (email);** |  |
| **5** | **Get Time, Date, Current user, MySQL Version using prompt?** |
| **SELECT CURTIME();**  **SELECT CURDATE();**  **SELECT USER();**  **SELECT VERSION();** |  |
| **6** | **Add gender column for the students table. It holds two value (male or female)** |
| **ALTER TABLE students ADD COLUMN gender ENUM('male', 'female') NOT NULL DEFAULT 'male';** |  |
| **7** | **Add birth\_date column for the students table.** |
| **ALTER TABLE students ADD COLUMN birth\_date data;** |  |
| **8** | **Drop the student\_name column and replace it with first name and last name.** |
| **-ALTER TABLE students ADD COLUMN first\_name VARCHAR(50);**  **-ALTER TABLE students ADD COLUMN last\_name VARCHAR(50);**  **-UPDATE students SET first\_name = SUBSTRING\_INDEX(student\_name, ' ', 1), last\_name = SUBSTRING\_INDEX(student\_name, ' ', -1);**  **-ALTER TABLE students DROP COLUMN student\_name;** |  |
| **9** | **Insert your friend’s data into the table students.** |
| **INSERT INTO students (first\_name, last\_name, gender, email, phone\_number, birth\_date)**  **VALUES ('friend\_first\_name', 'friend\_last\_name', 'gender\_value', 'friend\_email', 'friend\_phone\_number', 'friend\_date\_of\_birth');** |  |
| **10** | **Create a new table (male\_students) based on students table and fill it with the data of male students** |
| **CREATE TABLE male\_students LIKE students;**  **INSERT INTO male\_students SELECT \* FROM students WHERE gender = 'male';** |  |

**Part II**

**Create another database “CMS\_Alex”**

**Use CMS\_Alex**

**Run Lab Script then answer the following**

|  |  |
| --- | --- |
| **1** | **Display all students’ information.** |
|  | **SELECT \* FROM students;** |
| **2** | **Display male students only.** |
|  | **SELECT \* FROM students WHERE gender = 'male';** |
| **3** | **Display the number of female students.** |
|  | **SELECT COUNT(\*) FROM students WHERE gender = 'female';** |
| **4** | **Display the students’ data for the students who are born before 1992-10-01.** |
|  | **SELECT \* FROM students WHERE birthdate < '1992-10-01';** |
| **5** | **Display the students’ data for the students who are born after 1991-10-01.** |
|  | **SELECT \* FROM students WHERE birthdate >= '1991-10-01';** |
| **6** | **Display course\_id and their grades sorted by grades.** |
|  | **SELECT course\_id, grade FROM students\_courses ORDER BY grade ASC;** |
| **7** | **Display students’ names that begin with A.** |
|  | **SELECT first\_name, last\_name FROM students WHERE first\_name LIKE 'A%';** |
| **8** | **Display the gender, number of males and females.** |
|  | **SELECT gender, COUNT(\*) AS count FROM students GROUP BY gender;** |
| **9** | **Display the repeated first names and their counts if higher than 2.** |
|  | **SELECT first\_name, COUNT(\*) AS count FROM students GROUP BY first\_name HAVING count > 2;** |
| **10** | **Display the subject with highest grade** |
|  | **SELECT course\_id FROM students\_course WHERE grade = (SELECT MAX(grade) FROM students\_course);** |