**MySQL Labs**

**MySQL (Day1):**

|  |  |
| --- | --- |
|  | **Create a database called grades** |
|  | **CREATE DATABASE Grades;** |
|  | **Create the following tables in the grades database:**  ***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 TABLE `students` (**  **`student\_id` INT NOT NULL,**  **`student\_name` VARCHAR(100) NOT NULL,**  **`email` VARCHAR(50) NULL,**  **`tel` VARCHAR(20) NULL,**  **PRIMARY KEY (`student\_id`));**  **CREATE TABLE `courses` (**  **`course\_id` INT NOT NULL,**  **`course\_name` VARCHAR(100) NOT NULL,**  **`credit\_hour` INT NULL,**  **PRIMARY KEY (`course\_id`));**  **CREATE TABLE `students\_courses` (**  **`course\_id` INT NOT NULL,**  **`student\_id` INT NOT NULL,**  **`grade` INT NULL,**  **`reg\_date` DATE NULL,**  **PRIMARY KEY (`student\_id`, `course\_id`),**  **FOREIGN KEY (`student\_id`) REFERENCES `students`(`student\_id`),**  **FOREIGN KEY (`course\_id`) REFERENCES `courses` (`course\_id`));** |
| **3** | **Modify the students table to allow for longer Student names (150 char)**  **Confirm your modification.** |
|  | **ALTER TABLE students**  **MODIFY COLUMN 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 NOW(), CURDATE(), CURRENT\_USER(), VERSION();** |
| **6** | **Add gender column for the students table. It holds two value (male or female)** |
|  | **ALTER table students add gender enum ('male', 'female');** |
| **7** | **Add birth\_date column for the students table.** |
|  | **ALTER table students add birth\_date date;** |
| **8** | **Drop the student\_name column and replace it with first name and last name.** |
|  | **ALTER TABLE students ADD first\_name VARCHAR(50) NOT NULL;**  **ALTER TABLE students ADD last\_name VARCHAR(50) NOT NULL;** |
| **9** | **Insert your friend’s data into the table students.** |
|  | **INSERT INTO students VALUES(10, 'HS@gmail.com', '01000', 'male', 'Hossam', '1995-8-23', 'Adel');** |
| **10** | **Create a new table (male\_students) based on students table and fill it with the data of male students** |
|  | **CREATE TABLE male\_students AS SELECT \* FROM students WHERE gender = "male";** |  |

**Part II**

**Create another database “OS42”**

**Use OS42**

**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(gender)**  **from students**  **where gender = 'female';** |
| **4** | **Display the students’ data for the students who are born before 1992-10-01.** |
|  | **select \* from students where birth\_date <'1992-10-1';** |
| **5** | **Display male students who are born before 1991-10-01.** |
|  | **SELECT \* FROM students WHERE (birth\_date<'1992-10-1' AND gender = 'male');** |
| **6** | **Display course\_id and their grades sorted by grades.** |
|  | **SELECT course\_id, grade FROM students\_courses**  **ORDER BY grade;** |
| **7** | **Display students’ names that begin with A.** |
|  | **SELECT first\_name FROM students WHERE first\_name LIKE 'A%';** |
| **8** | **Display the gender, number of males and females.** |
|  | **SELECT gender, COUNT(gender) FROM students GROUP BY(gender);** |
| **9** | **Display the repeated first names and their counts if higher than 2.** |
|  | **SELECT first\_Name, COUNT(first\_Name) FROM students**  **GROUP BY(first\_Name) HAVING COUNT(first\_Name)>2;** |
| **10** | **Display the subject with highest grade** |
|  | **SELECT course\_id, max(grade) from students\_courses;** |