**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 (**  **-> students\_id int(30) NOT NULL,**  **-> students\_name CHAR(100) NOT NULL,**  **-> email CHAR(50),**  **-> telephone CHAR(20),**  **-> PRIMARY KEY(students\_id)**  **-> );**  **CREATE TABLE course (**  **-> course\_id int(30) NOT NULL,**  **-> course\_name CHAR(100) NOT NULL,**  **-> credit\_hour int(30),**  **-> PRIMARY KEY(course\_id)**  **-> );**  **CREATE TABLE students\_courses (**  **-> course\_id int(30) REFERENCES course(course\_id),**  **-> students\_id CHAR(100) REFERENCES students(students\_id) ,**  **-> grade int(30),**  **-> reg\_date DATE**  **-> );** |
| **3** | **Modify the students table to allow for longer Student names (150 char)**  **Confirm your modification.** |
|  | **ALTER TABLE students MODIFY students\_name CHAR(150);**  **describe students;** |
| **4** | **Add constraint to force unique email for each student** |
|  | **ALTER TABLE students**  **-> ADD CONSTRAINT email\_unique UNIQUE(email);** |
| **5** | **Get Time, Date, Current user, MySQL Version using prompt?** |
|  | **select now();**  **SELECT CURRENT\_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 ;** |
| **7** | **Add birth\_date column for the students table.** |
|  | **alter table students**  **-> add column birth\_date Date;** |
| **8** | **Drop the student\_name column and replace it with first name and last name.** |
|  | **alter table students**  **-> DROP COLUMN students\_name ,**  **-> ADD COLUMN first\_name CHAR(20) NOT NULL,**  **-> ADD COLUMN last\_name CHAR(20) NOT NULL;** |
| **9** | **Insert your friend’s data into the table students.** |
|  | **insert into students (students\_id ,email,telephone,gender,first\_name ,last\_name )**  **-> values(1,"shahd\_tamer@gmail.com","01280789865","FEMALE","shahd","tamer");** |
| **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 “php”**

**Use php**

**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 birth\_date<cast('1992-10-01' as date);** |
| **5** | **Display the students’ data for the male students who are born before 1991-10-01.** |
|  | **select \* from students where birth\_date<cast('1991-10-01' as date) 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 CONCAT(first\_name," ", last\_name) from students where first\_name LIKE 'A%';** |
| **8** | **Display the gender, number of males and females.** |
|  | **select gender , COUNT(student\_id) from students GROUP BY gender;** |
| **9** | **Display the repeated first names and their counts if higher than 2.** |
|  | **SELECT first\_name , count(\*) from students GROUP BY first\_name HAVING COUNT(\*)>2;** |
| **10** | **Display the subject with highest grade** |
|  | **SELECT course\_id from students\_courses**  **Where grade = (SELECT max(grade) From students\_courses);** |