**MySQL Labs**

**MySQL (Day1):**

|  |  |
| --- | --- |
|  | **Create a database called grades** |
|  | CREATE DATABASE IF NOT EXISTS 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)* |
|  | DROP DATABASE IF EXISTS grades;  CREATE DATABASE grades;  USE grades;  CREATE TABLE students  (  student\_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,  student\_id INT,  grade INT,  reg\_date DATE,  PRIMARY KEY(course\_id , student\_id),  KEY fk\_students\_courses\_courses\_idx (course\_id),  KEY fk\_students\_courses\_students\_idx (student\_id),  CONSTRAINT fk\_students\_courses\_courses FOREIGN KEY (course\_id) REFERENCES courses (course\_id),  CONSTRAINT fk\_students\_courses\_students FOREIGN KEY (student\_id) REFERENCES students (student\_id)  ); |
| **3** | **Modify the students table to allow for longer Student names (150 char)**  **Confirm your modification.** |
|  | ALTER TABLE students  MODIFY student\_name VARCHAR(150) NOT NULL; |
| **4** | **Add constraint to force unique email for each student** |
|  | ALTER TABLE students  MODIFY email VARCHAR(50) UNIQUE; |
| **5** | **Get Time, Date, Current user, MySQL Version using prompt?** |
|  | SELECT CURRENT\_TIME(),CURRENT\_DATE(),CURRENT\_USER(),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') |
| **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 student\_name,  ADD COLUMN first\_name VARCHAR(50) NOT NULL,  ADD COLUMN last\_name VARCHAR(50) NOT NULL; |
| **9** | **Insert your friend’s data into the table students.** |
|  | INSERT INTO students VALUES  (  DEFAULT,  'ranayoussef@gmail.com',  '01128554805',  'female',  '1997-07-07',  'Rana',  'Youssef'  ),  (  DEFAULT,  'karimali@gmail.com',  '01128554222',  'male',  '1996-10-20',  'Karim',  'Ali'  ) |
| **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 students.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(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 < '1992-10-01'; |
| **5** | **Display the students’ data for the male students who are born before 1991-10-01.** |
|  | SELECT \*  FROM students  WHERE birth\_date < '1991-10-01'  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 REGEXP '^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 MAX(grade) FROM students\_courses |