**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 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(20) not null,  -> `credit-hour` int  -> );  create table `students-courses`(  -> `course-id` int REFERENCES courses(`course-id`),  -> `student-id` int REFERENCES students(`student-id`),  -> grade int,  -> `reg-date` date  ->);  ALTER TABLE `students-courses` ADD FOREIGN KEY(`student-id`) REFERENCES students(`student-id`),  -> ADD FOREIGN KEY(`course-id`) REFERENCES courses(`course-id`),  -> ADD PRIMARY KEY(`course-id`, `student-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) NOT NULL; |
| **4** | **Add constraint to force unique email for each student** |
|  | ALTER TABLE students ADD UNIQUE(email); |
| **5** | **Get Time, Date, Current user, MySQL Version using prompt?** |
|  | SELECT CURRENT\_DATE(), CURRENT\_TIME(),Version(), CURRENT\_USER(); |
| **6** | **Add gender column for the students table. It holds two value (male or female)** |
|  | ALTER TABLE students ADD gender ENUM ('Female','Male'); |
| **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  DROP `student-name`,  ADD `first-name` varchar(50) Not NULL,  ADD `last-name` varchar(50) NOT NULL; |
| **9** | **Insert your friend’s data into the table students.** |
|  | INSERT INTO `students` (`student-id`, `email`, `tel`, `gender`, `last-name`, `first-name`) VALUES ('1', 'hager@gmail.com', '0144444444', 'Female', 'khaled', 'hager');  INSERT INTO `students` VALUES ('2', 'marim@gmail.com', '014445544', '1999-02-2', 'Female', 'khaled', 'marim'), ('3', 'mahmoud@gmail.com', '0123', '2002-02-22', 'Male', 'mohamed', 'mahmoud'); |
| **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”**

create database php;

**Use php**

use php;

**Run Lab Script then answer the following**

source E:php.txt;

|  |  |
| --- | --- |
| **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 < '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, last\_name FROM students WHERE first\_name LIKE 'a%'; |
| **8** | **Display the gender, number of males and females.** |
|  | SELECT COUNT(\*), gender FROM students GROUP BY(gender); |
| **9** | **Display the repeated first names and their counts if higher than 2.** |
|  | SELECT COUNT(first\_name), first\_name FROM students GROUP BY(first\_name) HAVING COUNT(first\_name)>2; |
| **10** | **Display the subject with highest grade** |
|  | SELECT \* FROM students\_courses ORDER BY grade DESC LIMIT 1; |