**MySQL Labs**

**MySQL (Day2):**

**C:\Users\Administrator\Desktop\xampp\mysql\bin**

|  |  |
| --- | --- |
|  | **Create a database called grades** |
|  | **Create database grades;**    **SHOW DATABASES;**    USE grades; |
|  | **Create the following tables in the grades database:**  SHOW FULL TABLES;  or  SHOW tables;    CREATE TABLE Students (Student\_ID int PRIMARY KEY,  Student\_Name varchar (100) not null, Email varchar (50), Telephone varchar (20));  SHOW tables;    SHOW COLUMNS FROM Students;    CREATE TABLE Courses (Course\_ID int PRIMARY KEY,  Course\_Name varchar (100) not null, Credit\_Hourvarchar (20));  SHOW tables;    SHOW COLUMNS FROM Courses;    CREATE TABLE Students\_Courses (Course\_ID int, Student\_ID int, Grade int, Reg\_Date date, PRIMARY KEY (Course\_ID, Student\_ID), FOREIGN KEY (Course\_ID) REFERENCES Courses (Course\_ID), FOREIGN KEY (Student\_ID) REFERENCES Students (Student\_ID));  SHOW COLUMNS FROM Courses;    SHOW COLUMNS FROM Students\_Courses;    ***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)* |
| **3** | **Modify the students table to allow for longer Student names (150 char)**  **Confirm your modification.** |
|  | From    ALTER TABLE Students MODIFY COLUMN Student\_Name varchar (150) not null;    SHOW COLUMNS FROM Students; |
| **4** | **Add constraint to force unique email for each student** |
|  | ALTER TABLE Students MODIFY COLUMN Email varchar (50) **unique**; |
| **5** | **Get Time, Date, Current user, MySQL Version using prompt?** |
|  | select now();    select current\_timestamp();    select current\_time,curtime(),current\_time();    select current\_date,curdate(),current\_date();    select current\_user(); |
| **6** | **Add gender column for the students table. It holds two value (male or female)** |
|  | ALTER TABLE Students ADD Gendervarchar (10) check (gender in ('Female','Male')); |
| **7** | **Add birth\_date column for the students table.** |
|  | ALTER TABLE Students ADD Birth\_Date date not null; |
| **8** | **Drop the student\_name column and replace it with first name and last name.** |
|  | ALTER TABLE Students DROP COLUMN Student\_Name;    ALTER TABLE Students ADD First\_Namevarchar (50) not null;    ALTER TABLE Students ADD Last\_Namevarchar (50) not null; |
| **9** | **Insert your friend’s data into the table students.** |
|  | INSERT INTO Students (Student\_ID, First\_Name, Last\_Name, Email, Telephone, Gender, Birth\_Date ) VALUES ('336699','Ahmed','Saber','Ahmed\_Saber@gmail.com','01077033966', 'Male','1997-3-20');    INSERT INTO Students (Student\_ID, First\_Name, Last\_Name, Email, Telephone, Gender, Birth\_Date ) VALUES ('669933','Ali','Saber',' Ali\_Saber@gmail.com','01067033966','Male','1999-1-20');    INSERT INTO Students (Student\_ID, First\_Name, Last\_Name, Email, Telephone, Gender, Birth\_Date) VALUES ('993366','Aya','Saber',' Aya\_Saber@gmail.com','01087033966','Female','2000-5-10'); |
| **10** | **Create a new table (male\_students) based on students table and fill it with the data of male students** |
|  | SELECT \* FROM Students    CREATE DATABASE Male\_Grades;    CREATE TABLE Male\_Grades.Male\_Students  AS  SELECT \* FROM Grades.Students  WHERE Gender = 'Male';  SHOW tables;  SHOW COLUMNS FROM Male\_Grades.Male\_Students;  SELECT \* FROM Male\_Grades.Male\_Students;  SELECT \* FROM Grades.Students; |

**Part II**

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

**Create database CMS\_Alex;**

**Use CMS\_Alex**

**Run Lab Script then answer the following.**

|  |  |
| --- | --- |
| **1** | **Display all students’ information.** |
|  | SHOW DATABASES; + USE cms\_alex; + show tables;    source C:\Users\Administrator\Desktop\lab1\_script.txt    show tables;    **The Required is: (Display all students’ information)**  **SELECT \* FROM cms\_alex.Students;** |
| **2** | **Display male students only.** |
|  | **SELECT \* FROM cms\_alex.Students**  **WHERE gender = 'Male';** |
| **3** | **Display the number of female students.** |
|  | **SELECT COUNT(gender) AS Number\_Of\_Female 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 students who are born after 1991-10-01.** |
|  | **SELECT \* FROM Students**  **WHERE birth\_date> '1991-10-01';** |
| **6** | **Display course\_id and their grades sorted by grades.** |
|  | **SELECT course\_id, grade FROM students\_courses ORDER BY grade;**    **ORDER BY grade ASC;**    **ORDER BY grade DESC;** |
| **7** | **Display students’ names that begin with A.** |
|  | SELECT \* FROM students WHERE first\_name LIKE 'a%';    SELECT first\_name FROM students WHERE first\_name LIKE 'a%'; |
| **8** | **Display the gender, number of males and females.** |
|  | SELECT gender FROM students;  Select  count(if(gender ='male',1,null)) as Total Number\_Of\_Males,  count(if(gender ='female',1,null)) as Number\_Of\_Females from students;    select distinct gender, count(gender) as Total\_Number  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;    SELECT first\_name, COUNT(first\_name) as Number\_of\_repeatition  FROM students  GROUP BY first\_name  HAVING COUNT(first\_name) > 2; |
| **10** | **Display the subject with highest grade** |
|  | SELECT \*  FROM students\_courses  WHERE grade = (SELECT MAX(grade) FROM students\_courses); |