|  |  |
| --- | --- |
| **Create a database called grades**  *CREATE DATABASE grades;* |  |
| **Create the following tables in the grades database:**  *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, primary key (course\_id,student\_id),*  *Grade int,*  *reg\_date date*  *ADD CONSTRAINT fk\_student\_id*  *FOREIGN KEY (student\_id)*  *REFERENCES students(student\_id)*  *ADD CONSTRAINT fk\_course\_id*  *FOREIGN KEY (course\_id)*  *REFERENCES courses (course\_id));* |  |
| ***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)* | **:** |
|  |  |
| **ALTER TABLE students modify student\_name varchar(150);** | **Modify the students table to allow for longer Student names (150 char)**  **Confirm your modification.** |
|  |  |
| **4**  **ALTER TABLE students ADD CONSTRAINT unique\_email UNIQUE (email);** | **Add constraint to force unique email for each student** |
|  |  |
| **5 select curtime();**  **SELECT CURRENT\_DATE();**  **SELECT USER();**  **SELECT VERSION();** | **Get Time, Date, Current user, MySQL Version using prompt?** |
|  |  |
| **6**  **ALTER TABLE students ADD gender ENUM('male', 'female');** | **Add gender column for the students table. It holds two value (male or female)** |
|  |  |
| **7**  **ALTER TABLE students ADD birth\_date DATE;** | **Add birth\_date column for the students table.** |
|  |  |
| **8 ALTER TABLE students**  **DROP COLUMN student\_name,**  **ADD COLUMN first\_name VARCHAR(50),**  **ADD COLUMN last\_name VARCHAR(50);** | **Drop the student\_name column and replace it with first name and last name.** |
|  |  |
| **9**  **INSERT INTO students (first\_name, last\_name, gender, birth\_date, email)**  **VALUES ('reham', 'ismail', 'female', '1991-10-12', 'rehamismail91@gmail.com');** | **Insert your friend’s data into the table students.** |
|  |  |
| **10**  **CREATE TABLE male\_students LIKE students;**  **INSERT INTO male\_students**  **SELECT \* FROM students WHERE gender = 'male';** | **Create a new table (male\_students) based on students table and fill it with the data of male students** |
|  |  |

**MySQL Labs**

**MySQL (Day1):**

**Part II**

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

**Use CMS\_Alex**

*CREATE DATABASE* **CMS\_Alex;**

**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 < '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 DESC;** |
| **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 gender, COUNT(\*) FROM students GROUP BY gender;** |
| **9** | **Display the repeated first names and their counts if higher than 2.** |
|  | **SELECT first\_name, COUNT(\*) as name\_count FROM students GROUP BY first\_name HAVING name\_count > 2;** |
| **10** | **Display the subject with highest grade** |
|  | **SELECT c.course\_name, MAX(sc.grade) as max\_grade**  **FROM students\_courses sc**  **JOIN courses c ON sc.course\_id = c.course\_id;** |