

# MySQL Labs

## MySQL (Day1):

1.	Create a database called grades
	CREATE DATABASE IF NOT EXISTS grades;
2.	<div>Create the following tables in the grades database:<div><div><div>students</div><div>student_id int pk student_name varchar (100) not null email varchar (50) tel varchar (20)</div></div><div><div>courses</div><div>course_id int pk course_name varchar(100) not null credit_hour int</div></div><div><div>students_courses</div><div>course_id int student_id int grade int reg_date date</div></div></div></div>
	<div>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 (</div>

	<pre>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) );</pre>								
3	<b>Modify the students table to allow for longer Student names (150 char)</b> <b>Confirm your modification.</b>								
	<pre>ALTER TABLE students MODIFY student_name VARCHAR(150) NOT NULL;</pre>								
4	<b>Add constraint to force unique email for each student</b>								
	<pre>ALTER TABLE students MODIFY email VARCHAR(50) UNIQUE;</pre>								
5	<b>Get Time, Date, Current user, MySQL Version using prompt?</b>								
	<pre>SELECT CURRENT_TIME(),CURRENT_DATE(),CURRENT_USER(),VERSION();</pre> <table><tr><th>CURRENT_TIME()</th><th>CURRENT_DATE()</th><th>CURRENT_USER()</th><th>VERSION()</th></tr><tr><td>18:55:10</td><td>2023-01-13</td><td>root@localhost</td><td>10.4.27-MariaDB</td></tr></table>	CURRENT_TIME()	CURRENT_DATE()	CURRENT_USER()	VERSION()	18:55:10	2023-01-13	root@localhost	10.4.27-MariaDB
CURRENT_TIME()	CURRENT_DATE()	CURRENT_USER()	VERSION()						
18:55:10	2023-01-13	root@localhost	10.4.27-MariaDB						
6	<b>Add gender column for the students table. It holds two value (male or female)</b>								
	<pre>ALTER TABLE students ADD COLUMN gender ENUM('male','female')</pre>								
7	<b>Add birth_date column for the students table.</b>								
	<pre>ALTER TABLE students ADD COLUMN birth_date DATE</pre>								
8	<b>Drop the student_name column and replace it with first name and last name.</b>								
	<pre>ALTER TABLE students DROP COLUMN student_name, ADD COLUMN first_name VARCHAR(50) NOT NULL, ADD COLUMN last_name VARCHAR(50) NOT NULL;</pre>								
9	<b>Insert your friend's data into the table students.</b>								
	<pre>INSERT INTO students VALUES ( DEFAULT, 'ranayoussef@gmail.com', '01128554805', 'female', '1997-07-07', 'Rana', 'Youssef' ).</pre>								

	( DEFAULT, 'karimali@gmail.com', '01128554222', 'male', '1996-10-20', 'Karim', 'Ali' )
<b>10</b>	<b>Create a new table (male_students) based on students table and fill it with the data of male students</b>
	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	<b>Display all students' information.</b>
	SELECT * FROM students
2	<b>Display <u>male</u> students only.</b>
	SELECT * FROM students WHERE gender = 'male';
3	<b>Display the <u>number of female</u> students.</b>
	SELECT COUNT(gender) FROM students WHERE gender = 'female';
4	<b>Display the <u>students' data</u> for the students who are born before 1992-10-01.</b>
	SELECT * FROM students WHERE birth_date < '1992-10-01';
5	<b>Display the <u>students' data</u> for the male students who are born before 1991-10-01.</b>
	SELECT * FROM students WHERE birth_date < '1991-10-01' AND gender = 'male';
6	<b>Display <u>course id</u> and their grades sorted by grades.</b>
	SELECT course_id , grade FROM students_courses ORDER BY grade
7	<b>Display <u>students' names</u> that begin with A.</b>
	SELECT first_name FROM students WHERE first_name REGEXP '^A';
8	<b>Display the <u>gender, number of males and females</u>.</b>
	SELECT gender , COUNT(gender) FROM students GROUP BY gender;
9	<b>Display the <u>repeated first names</u> and <u>their counts</u> if higher than 2.</b>
	SELECT first_name , COUNT(first_name) FROM students GROUP BY first_name HAVING COUNT(first_name) > 2;
10	<b>Display the <u>subject with highest grade</u></b>
	SELECT MAX(grade) FROM students_courses