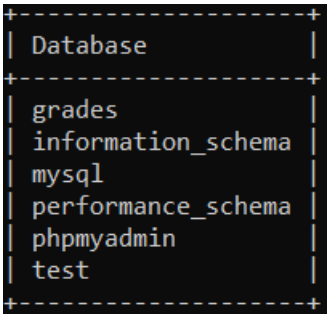
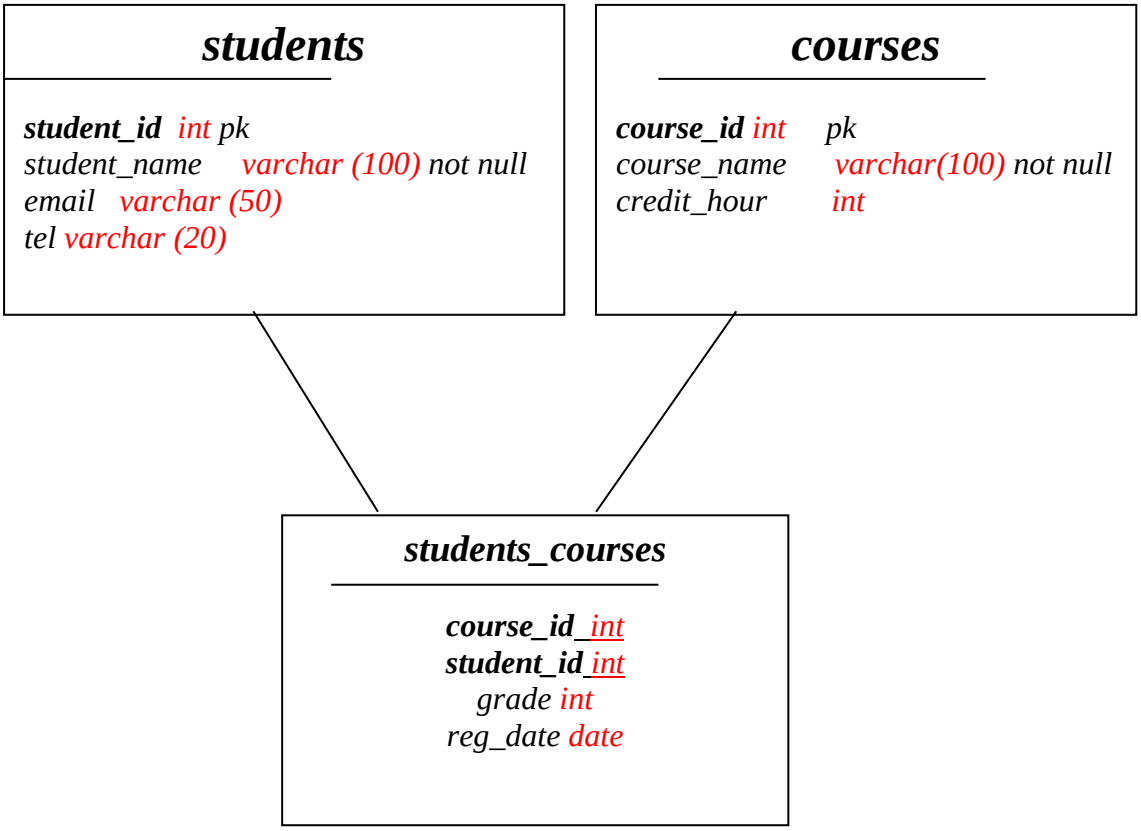
MySQL Labs

MySQL (Day1):

1.	**Create a database called grades**
	CREATE DATABASE grades CHARACTER SET utf8 COLLATE utf8_danish_ci; 
2.	**Create the following tables in the grades database:** 
	CREATE TABLE students(student_id INT(11) NOT NULL, student_name VARCHAR(100) NOT NULL, email VARCHAR(50),

tel VARCHAR(20),
PRIMARY KEY(student_id));

CREATE TABLE courses(
course_id INT(11) NOT NULL,
course_name VARCHAR(100) NOT NULL,
credit_hour INT(11),
PRIMARY KEY(course_id));

CREATE TABLE students_courses(
student_id INT(11),
course_id INT(11),
grade INT(11),
reg_date DATE,
FOREIGN KEY(student_id) REFERENCES students(student_id),
FOREIGN KEY(course_id) REFERENCES courses(course_id)
PRIMARY KEY(student_id,course_id));

```
+-------------------+
| Tables_in_grades  |
+-------------------+
| courses           |
| students          |
| students_courses  |
+-------------------+
```

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;

```
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| student_id   | int(11)      | NO   | PRI | NULL    |       |
| student_name | varchar(150) | NO   |     | NULL    |       |
| email        | varchar(50)  | YES  |     | NULL    |       |
| tel          | varchar(20)  | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
```

4 **Add constraint to force unique email for each student**

ALTER TABLE students
MODIFY email VARCHAR(50) UNIQUE;

```
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| student_id   | int(11)      | NO   | PRI | NULL    |       |
| student_name | varchar(150) | NO   |     | NULL    |       |
| email        | varchar(50)  | YES  | UNI | NULL    |       |
| tel          | varchar(20)  | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
```

5	**Get Time, Date, Current user, MySQL Version using prompt?**
	SELECT NOW(); SELECT CURRENT_USER(); SHOW VARIABLES LIKE '%version%'; +---------------------+ | NOW() | +---------------------+ | 2023-01-21 13:33:13 | +---------------------+ +----------------+ | CURRENT_USER() | +----------------+ | root@localhost | +----------------+ +----------------------------------+--+ | Variable_name | Value | +----------------------------------+--+ | in_predicate_conversion_threshold | 1000 | | innodb_version | 10.4.27 | | protocol_version | 10 | | slave_type_conversions | | | system_versioning_alter_history | ERROR | | system_versioning_asof | DEFAULT | | tls_version | TLSv1.1,TLSv1.2,TLSv1.3 | | version | 10.4.27-MariaDB | | version_comment | mariadb.org binary distribution | | version_compile_machine | x64 | | version_compile_os | Win64 | | version_malloc_library | system | | version_source_revision | 0946c99e7d6f7ac9dfcf3e60dae6ae85161d5ef2 | | version_ssl_library | WolfSSL 5.5.1 | +----------------------------------+--+
6	**Add gender column for the students table. It holds two value (male or female)**
	ALTER TABLE students ADD COLUMN gender ENUM('male','female') NOT NULL; +--------------+----------------------+------+-----+---------+-------+ | Field | Type | Null | Key | Default | Extra | +--------------+----------------------+------+-----+---------+-------+ | student_id | int(11) | NO | PRI | NULL | | | student_name | varchar(150) | NO | | NULL | | | email | varchar(50) | YES | UNI | NULL | | | tel | varchar(20) | YES | | NULL | | | gender | enum('male','female') | NO | | NULL | | +--------------+----------------------+------+-----+---------+-------+
7	**Add birth_date column for the students table.**
	ALTER TABLE students ADD COLUMN birth_date DATE; +--------------+----------------------+------+-----+---------+-------+ | Field | Type | Null | Key | Default | Extra | +--------------+----------------------+------+-----+---------+-------+ | student_id | int(11) | NO | PRI | NULL | | | student_name | varchar(150) | NO | | NULL | | | email | varchar(50) | YES | UNI | NULL | | | tel | varchar(20) | YES | | NULL | | | gender | enum('male','female') | NO | | NULL | | | birth_date | date | YES | | NULL | | +--------------+----------------------+------+-----+---------+-------+
8	**Drop the student_name column and replace it with first name and last name.**
	ALTER TABLE students DROP student_name;

ALTER TABLE students
ADD COLUMN first_name VARCHAR(50) NOT NULL;
ALTER TABLE students
ADD COLUMN last_name VARCHAR(50) NOT NULL;

Field	Type	Null	Key	Default	Extra
student_id	int(11)	NO	PRI	NULL	
email	varchar(50)	YES	UNI	NULL	
tel	varchar(20)	YES		NULL	
gender	enum('male','female')	NO		NULL	
birth_date	date	YES		NULL	
first_name	varchar(50)	NO		NULL	
last_name	varchar(50)	NO		NULL	

9 **Insert your friend's data into the table students.**

INSERT INTO students (student_id,first_name,last_name,gender,email,tel,birth_date)
VALUES
(1,"Hager","Abd El Galil","female","hager@gmail.com","01201422915","1999-12-24"),
(2,"Alaa","Abd El Galil","female","alaa@gmail.com","01022589735","2006-4-22"),
(3,"Nada","Saeed","female","nada@gmail.com","01126498733","1998-9-24"),
(4,"Ahmed","Mahmoud","male","ahmed@gmail.com","01111254334","2000-5-5"),
(5,"Amir","Ali","male","amir@gmail.com","01522322463","1997-7-5");

student_id	email	tel	gender	birth_date	first_name	last_name
1	hager@gmail.com	01201422915	female	1999-12-24	Hager	Abd El Galil
2	alaa@gmail.com	01022589735	female	2006-04-22	Alaa	Abd El Galil
3	nada@gmail.com	01126498733	female	1998-09-24	Nada	Saeed
4	ahmed@gmail.com	01111254334	male	2000-05-05	Ahmed	Mahmoud
5	amir@gmail.com	01522322463	male	1997-07-05	Amir	Ali

10 **Create a new table (male_students) based on students table and fill it with the data of male students**

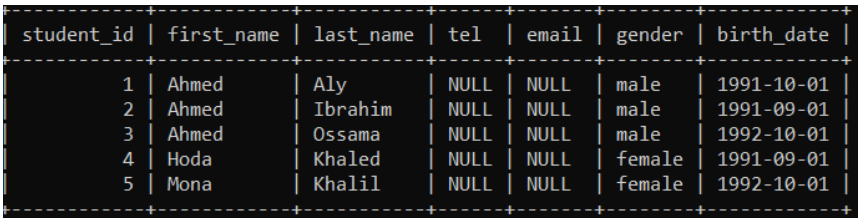
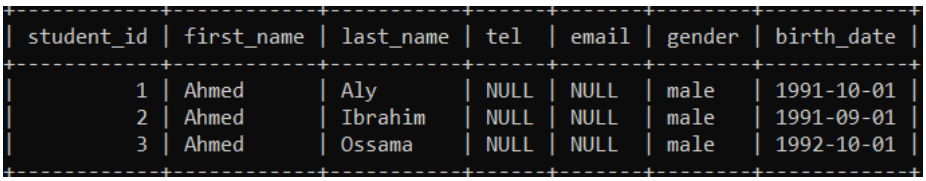
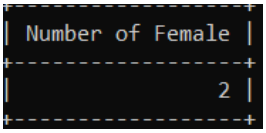
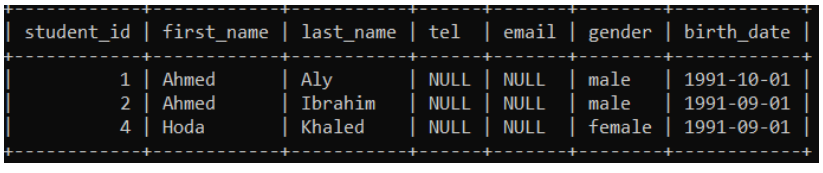
CREATE TABLE male_students
SELECT * FROM students
WHERE gender LIKE "male";

student_id	email	tel	gender	birth_date	first_name	last_name
4	ahmed@gmail.com	01111254334	male	2000-05-05	Ahmed	Mahmoud
5	amir@gmail.com	01522322463	male	1997-07-05	Amir	Ali

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 LIKE “male”; 
3	**Display the number of female students.**
	SELECT COUNT(*) AS “Number of Female” FROM students WHERE gender LIKE “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

student_id	first_name	last_name	tel	email	gender	birth_date
1	Ahmed	Aly	NULL	NULL	male	1991-10-01
2	Ahmed	Ibrahim	NULL	NULL	male	1991-09-01
3	Ahmed	Ossama	NULL	NULL	male	1992-10-01
4	Hoda	Khaled	NULL	NULL	female	1991-09-01
5	Mona	Khalil	NULL	NULL	female	1992-10-01

student_id	first_name	last_name	tel	email	gender	birth_date
1	Ahmed	Aly	NULL	NULL	male	1991-10-01
2	Ahmed	Ibrahim	NULL	NULL	male	1991-09-01
3	Ahmed	Ossama	NULL	NULL	male	1992-10-01

Number of Female
2

student_id	first_name	last_name	tel	email	gender	birth_date
1	Ahmed	Aly	NULL	NULL	male	1991-10-01
2	Ahmed	Ibrahim	NULL	NULL	male	1991-09-01
4	Hoda	Khaled	NULL	NULL	female	1991-09-01

WHERE (gender LIKE “male”) AND (birth_date < “1992-10-01”);

student_id	first_name	last_name	tel	email	gender	birth_date
1	Ahmed	Aly	NULL	NULL	male	1991-10-01
2	Ahmed	Ibrahim	NULL	NULL	male	1991-09-01

6 **Display course_id and their grades sorted by grades.**

SELECT course_id , grade FROM students_courses
ORDER BY grade;

course_id	grade
4	70
1	80
3	80
2	90
2	99
3	100

7 **Display students’ names that begin with A.**

SELECT CONCAT(first_name,” “,last_name) AS “Full Name” FROM students
WHERE first_name LIKE “A%”;

Full Name
Ahmed Aly
Ahmed Ibrahim
Ahmed Ossama

8 **Display the gender, number of males and females.**

SELECT gender,COUNT(*) FROM students
GROUP BY gender;

gender	COUNT(*)
male	3
female	2

9 **Display the repeated first names and their counts if higher than 2.**

SELECT first_name,COUNT(student_id) FROM students
GROUP BY first_name
HAVING COUNT(student_id) > 2;

first_name	COUNT(student_id)
Ahmed	3

10	**Display the <u>subject with highest grade</u>**
	SELECT c.course_id AS “COURSE ID” ,c.course_name AS “COURSE NAME” FROM courses c, students_courses sc WHERE (c.course_id = sc.course_id) AND (grade = (SELECT MAX(grade) FROM students_courses)); +-----------+-------------+ | COURSE ID | COURSE NAME | +-----------+-------------+ | 3 | Network | +-----------+-------------+ Or SELECT course_id FROM students_courses WHERE grade = (SELECT MAX(grade) FROM students_courses); +-----------+ | course_id | +-----------+ | 3 | +-----------+