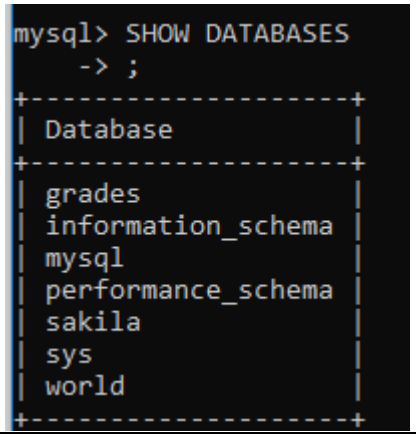
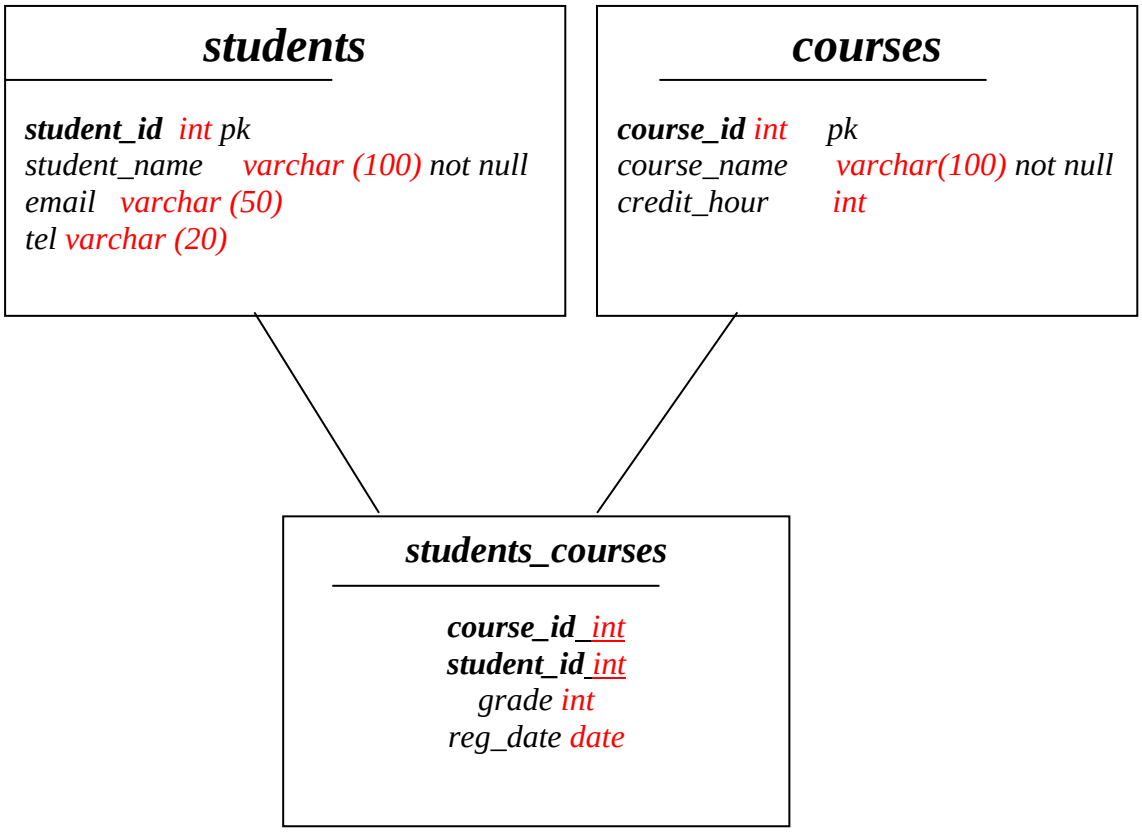
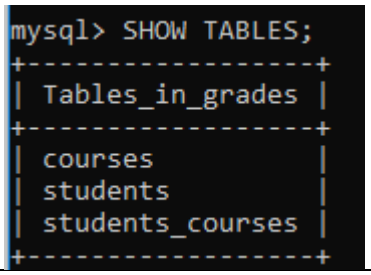
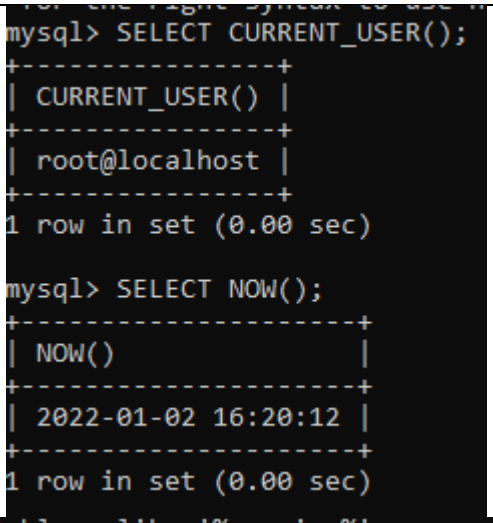
crMySQL Labs

MySQL (Day1):

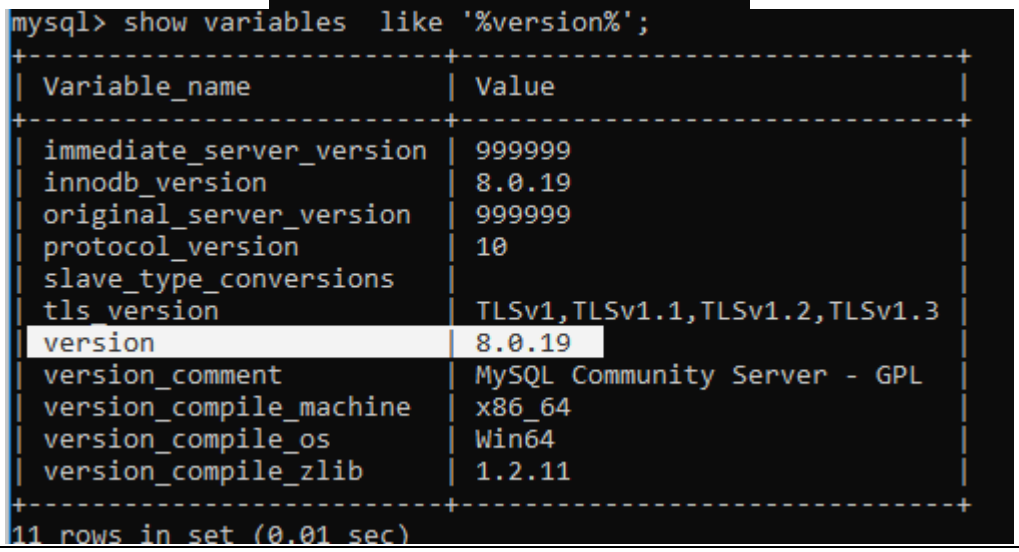
1.	**Create a database called grades**
	CREATE database grades CHARACTER SET utf8 COLLATE utf8_danish_ci; **USE grades;** 
2.	**Create the following tables in the grades database:** 

	CREATE TABLE IF NOT EXISTS students(**student_id INT(11) NOT NULL AUTO_INCREMENT,** **student_name VARCHAR(100) NOT NULL,** **email VARCHAR(50),** **tel VARCHAR(20),** **PRIMARY KEY (student_id))** **ENGINE = INNODB;** **CREATE TABLE IF NOT EXISTS *courses*(** **course_id INT(11) NOT NULL AUTO_INCREMENT,** **course_name VARCHAR(100) NOT NULL,** **credit_hour INT,** **PRIMARY KEY (course_id))** **ENGINE = INNODB;** **CREATE TABLE IF NOT EXISTS *students_courses*(** **student_id INT(11) NOT NULL,** **course_id INT(11) NOT NULL,** **grade INT(11),** **PRIMARY KEY (student_id, course_id),** **FOREIGN KEY (student_id) REFERENCES students (student_id),** **FOREIGN KEY (course_id) REFERENCES courses (course_id))** **ENGINE = INNODB;** 
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;**
4	**Add constraint to force unique email for each student**
	ALTER TABLE students **MODIFY email VARCHAR(50) UNIQUE;**
5	**Get Time, Date, Current user, MySQL Version using prompt?**
	SELECT NOW(); **SELECT CURRENT_USER();** **show variables like '%version%';**



```
mysql> SELECT CURRENT_USER();
+----------------+
| CURRENT_USER() |
+----------------+
| root@localhost |
+----------------+
1 row in set (0.00 sec)

mysql> SELECT NOW();
+---------------------+
| NOW()               |
+---------------------+
| 2022-01-02 16:20:12 |
+---------------------+
1 row in set (0.00 sec)
```



```
mysql> show variables  like '%version%';
+--------------------------+--------------------------------+
| Variable_name            | Value                          |
+--------------------------+--------------------------------+
| immediate_server_version | 999999                         |
| innodb_version           | 8.0.19                         |
| original_server_version  | 999999                         |
| protocol_version         | 10                             |
| slave_type_conversions   |                                |
| tls_version              | TLSv1,TLSv1.1,TLSv1.2,TLSv1.3  |
| version                  | 8.0.19                         |
| version_comment          | MySQL Community Server - GPL   |
| version_compile_machine  | x86_64                         |
| version_compile_os       | Win64                          |
| version_compile_zlib     | 1.2.11                         |
+--------------------------+--------------------------------+
11 rows in set (0.01 sec)
```

6	**Add gender column for the students table. It holds two value (male or female)**
	Alter table students add gender enum ('male', 'female');
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 COLUMN student_name; ALTER TABLE students ADD first_name VARCHAR(50) NOT NULL; ALTER TABLE students ADD last_name VARCHAR(50) NOT NULL;
9	**Insert your friend's data into the table students.**
	INSERT INTO students (student_id, first_name, last_name, tel, email, gender, birth_date) VALUES (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”);
	`mysql> select * from students;`
	`+------------+-------+------+--------+------------+------------+-----------+`
	`| student_id | email | tel | gender | birth_date | first_name | last_name |`
	`+------------+-------+------+--------+------------+------------+-----------+`
	`| 1 | NULL | NULL | male | 1991-10-01 | Ahmed | Aly |`
	`| 2 | NULL | NULL | male | 1991-09-01 | Ahmed | Ibrahim |`
	`| 3 | NULL | NULL | male | 1992-10-01 | Ahmed | Ossama |`
	`| 4 | NULL | NULL | female | 1991-09-01 | Hoda | Khaled |`
	`| 5 | NULL | NULL | female | 1992-10-01 | Mona | Khalil |`
	`+------------+-------+------+--------+------------+------------+-----------+`
	`5 rows in set (0.00 sec)`

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=”male”;
	`mysql> select * from male_students;`
	`+------------+-------+------+--------+------------+------------+-----------+`
	`| student_id | email | tel | gender | birth_date | first_name | last_name |`
	`+------------+-------+------+--------+------------+------------+-----------+`
	`| 1 | NULL | NULL | male | 1991-10-01 | Ahmed | Aly |`
	`| 2 | NULL | NULL | male | 1991-09-01 | Ahmed | Ibrahim |`
	`| 3 | NULL | NULL | male | 1992-10-01 | Ahmed | Ossama |`
	`+------------+-------+------+--------+------------+------------+-----------+`
	`3 rows in set (0.00 sec)`

Part II
Create another database “OS42”

Use OS42

Run Lab Script then answer the following

CREATE database OS42 CHARACTER SET utf8 COLLATE utf8_danish_ci;
USE OS42;

1	**Display all students’ information.**

	select * from students;
	`mysql> 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 |`
	`+------------+------------+-----------+------+-------+--------+------------+`

2	**Display male students only.**

	select * from students
	WHERE gender = “male”; // or WHERE gender LIKE “male”;
	`mysql> select * from students`
	`-> WHERE gender = "male";`
	`+------------+------------+-----------+------+-------+--------+------------+`
	`| 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 |`
	`+------------+------------+-----------+------+-------+--------+------------+`

3	**Display the number of female students.**

	select count(student_id) from students
	WHERE gender = “female”;
	`mysql> select count(student_id) from students`
	`-> WHERE gender = "female";`
	`+--------------------+`
	`| count(student_id) |`
	`+--------------------+`
	`| 2 |`
	`+--------------------+`
	`1 row in set (0.00 sec)`

4	**Display the students’ data for the students who are born before 1992-10-01.**
	select * from students **WHERE birth_date < “1992-10-01”;**

```
mysql> select * from students
    -> WHERE 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 |
|          4 | Hoda       | Khaled    | NULL | NULL  | female | 1991-09-01 |
+------------+------------+-----------+------+-------+--------+------------+
3 rows in set (0.01 sec)
```

5 **Display male students who are born before 1991-10-01.**

```
select * from students
WHERE birth_date < "1992-10-01"
AND gender LIKE "male";
```

```
mysql> select * from students
    -> WHERE birth_date < "1992-10-01"
    -> AND gender LIKE "male";
+------------+------------+-----------+------+-------+--------+------------+
| 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 |
+------------+------------+-----------+------+-------+--------+------------+
2 rows in set (0.00 sec)
```

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

```
select course_id, grade from students_courses
ORDER BY grade;
```

```
mysql> 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 |
+-----------+-------+
6 rows in set (0.01 sec)
```

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

```
select CONCAT(first_name, " ", last_name) from students
WHERE first_name LIKE "A%";
```

```
mysql> select CONCAT(first_name, " ", last_name) from students
    -> WHERE first_name LIKE "A%";
+-----------------------------------+
| CONCAT(first_name, " ", last_name) |
+-----------------------------------+
| Ahmed Aly                         |
| Ahmed Ibrahim                     |
| Ahmed  Ossama                     |
+-----------------------------------+
```

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

```
Select gender, count(student_id) from students
Group by gender;
```

```
mysql> Select gender, count(student_id) from students
    -> Group by gender;
+--------+-------------------+
| gender | count(student_id) |
+--------+-------------------+
| male   |                 3 |
| female |                 2 |
+--------+-------------------+
2 rows in set (0.01 sec)
```

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;**

```
mysql> Select first_name, count(student_id) from students
    -> Group by first_name
    -> Having count(student_id) > 2;
+------------+-------------------+
| first_name | count(student_id) |
+------------+-------------------+
| Ahmed      |                 3 |
+------------+-------------------+
1 row in set (0.00 sec)
```

10	**Display the subject with highest grade**
	Select c.course_name, max(e.grade) **From courses c, students_courses e** **Where c.course_id = e.course_id;**

```
mysql> Select c.course_name, max(e.grade)
    -> From courses c, students_courses e
    -> Where c.course_id = e.course_id;
+-------------+--------------+
| course_name | max(e.grade) |
+-------------+--------------+
| Database    |          100 |
+-------------+--------------+
1 row in set (0.00 sec)
```