Lab 10 - HW

1.

```
CREATE DATABASE UniversityDB;
```

2.

```
CREATE TABLE IF NOT EXISTS students
(
    student_id int ,
    first_name varchar(15) not null,
    lastname varchar(15),
    date_of_birth date,
    email varchar(50) unique ,

    constraint PK_Stu primary key (student_id)
);
CREATE TABLE IF NOT EXISTS courses
(
    course_id   varchar(10),
    course_name    varchar(30) not null ,
    credits int,

    constraint PK_courses primary key (course_id)
);
CREATE TABLE IF NOT EXISTS grades
(
    grade_id   int primary key auto_increment,
    student_id    int,
    course_id    int,
    grade int,
    constraint FK_student foreign key(student_id) references students(student_id),
    constraint FK_course foreign key(course_id) references courses(    course_id),
    check ( grade between 0 and 100)
);

ALTER TABLE grades AUTO_INCREMENT = 10;
```

3. Bulk insert data into students table

```
INSERT INTO students (student_id, first_name, last_name, date_of_birth, email)
VALUES
  (202120580, 'Ali', 'Osama', '2003-03-18', 'ali.osama@gju.edu.jo'),
  (202110739, 'Aseel', 'Adam', '2003-09-27', 'aseel.adam@gju.edu.jo'),
  (202022453, 'Fatimah', 'Ahmad', '2002-04-02', 'fatimah.ahmad@gju.edu.jo'),
  (202219876, 'Ahmad', 'Zaid', '2004-08-07', 'ahmd.zaid@gju.edu.jo');
```

4.

```
INSERT INTO students
VALUES (20221501042, 'Ahmad', 'Mahmoud', '2004-01-28', 'a.mahmoud@gju.edu.jo');
```

5.

```
INSERT INTO courses (course_id, course_name, credits)
VALUES
  ('CS116', 'Computing Fundamentals', 3),
  ('CS1160', 'Computing Fundamentals LAB', 1),
  ('CS263', 'Database Management Systems', 3);
```

6.

```
INSERT INTO courses
VALUES
  ('CS222', 'Theory of Algorithms', 3);
```

7.

```
INSERT INTO grades (student_id, course_id, grade)
VALUES
  (202219876, 'CS116', 98),
  (202219876, 'CS1160', 97),
  (202110739, 'CS1160', 40),
  (202022453, 'CS263', 77);
```

8.

```
INSERT INTO grades(student_id, course_id, grade)
VALUES
  (20221501042, 'CS263', 50);
```

9.

```
CREATE TABLE gradeCopy as select CONCAT(s.first_name, ' ', s.last_name) AS
student_name, c.course_name, G.grade
FROM grades G join students s join courses c on G.student_id = s.student_id AND
g.course_id = c.course_id;

ALTER  table gradeCopy
add constraint gC_PK primary key (student_name, course_name);
```

10.

UniversityDB is a DB that tracks a student's info regarding his courses and their grades, it consists of 4 tables:
students, courses, grades, and gradesCopy which seems more of a backup solution for grades table or a grade table with names instead of IDs, gradeCopy seems more useful to use in a Transcript, whilst grade is used for saving Data.
the tables 'grades' and 'gradesCopy' have relationships with 'students' and 'courses' tables.

```
-   1. `course_id` of **grades** isn't the same `course_id` of **courses**, `int`
!=   `varchar()`.
-   2.  My ID isn't in the range of `int`, this requires the Datatype change to
`bigint`.
-   3.  When Editing the Data Types, It's required to check all columns in the
database that refer to that Column, and ensuring that they all have the same
Datatype.
```

No Challenges other than remembering SQL's weird and unneccesary syntax, also I'm in support of providing a cheatsheet of MYSQL's syntax during the final Exam, because in real life googling is allowed in work.

Correct Version with no Errors.

```
-- 1.
CREATE DATABASE UniversityDB;

-- 2.
CREATE TABLE IF NOT EXISTS students
(
    student_id bigint ,
    first_name varchar(15) not null,
    last_name varchar(15),
    date_of_birth date,
```

```sql
    email varchar(50) unique ,

    constraint PK_Stu primary key (student_id)
);

CREATE TABLE IF NOT EXISTS courses
(
    course_id   varchar(10),
    course_name    varchar(30) not null ,
    credits int,

    constraint PK_courses primary key (course_id)
);

CREATE TABLE IF NOT EXISTS grades
(
    grade_id   int primary key auto_increment,
    student_id    bigint,
    course_id    varchar(10),
    grade int,

    constraint FK_student foreign key(student_id) references students(student_id),
    constraint FK_course foreign key(course_id) references courses(    course_id),
    check ( grade between 0 and 100)
);

ALTER TABLE grades AUTO_INCREMENT = 10;

-- 3. Bulk insert data into students table
INSERT INTO students (student_id, first_name, last_name, date_of_birth, email)
VALUES
  (202120580, 'Ali', 'Osama', '2003-03-18', 'ali.osama@gju.edu.jo'),
  (202110739, 'Aseel', 'Adam', '2003-09-27', 'aseel.adam@gju.edu.jo'),
  (202022453, 'Fatimah', 'Ahmad', '2002-04-02', 'fatimah.ahmad@gju.edu.jo'),
  (202219876, 'Ahmad', 'Zaid', '2004-08-07', 'ahmd.zaid@gju.edu.jo');

-- 4.
INSERT INTO students
VALUES (20221501042, 'Ahmad', 'Mahmoud', '2004-01-28', 'a.mahmoud@gju.edu.jo');

-- 5.
INSERT INTO courses (course_id, course_name, credits)
VALUES
  ('CS116', 'Computing Fundamentals', 3),
  ('CS1160', 'Computing Fundamentals LAB', 1),
  ('CS263', 'Database Management Systems', 3);
```

```sql
-- 6.
INSERT INTO courses
VALUES
  ('CS222', 'Theory of Algorithms', 3);

-- 7.
INSERT INTO grades (student_id, course_id, grade)
VALUES
  (202219876, 'CS116', 98),
  (202219876, 'CS1160', 97),
  (202110739, 'CS1160', 40),
  (202022453, 'CS263', 77);

-- 8.
INSERT INTO grades(student_id, course_id, grade)
VALUES
  (20221501042, 'CS263', 50);

-- 9.
CREATE TABLE gradeCopy as select CONCAT(s.first_name, ' ', s.last_name) AS
student_name, c.course_name, G.grade
FROM grades G join students s join courses c on G.student_id = s.student_id AND
G.course_id = c.course_id;

ALTER  table gradeCopy
add constraint gC_PK primary key (student_name, course_name);
```