Lab2 Advanced Sql

Q1: Add gender column for the student table. It holds two value (male or female)

```
mysql> ALTER TABLE student add gender ENUM('male', 'female') NOT NULL;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Q2: Add birth date column for the student table.

```
mysql> ALTER TABLE student add birthDate date;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Q3: Delete the name column and replace it with two colums first name and last name.

```
mysql> alter table student drop name
    -> ;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> ALTER TABLE student add fName varchar(30);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE student add lName varchar(30);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> describe student;
+-----------+----------------------+------+-----+---------+----------------+
| Field     | Type                 | Null | Key | Default | Extra          |
+-----------+----------------------+------+-----+---------+----------------+
| Id        | int                  | NO   | PRI | NULL    | auto_increment |
| email     | varchar(30)          | YES  |     | NULL    |                |
| address   | varchar(50)          | YES  |     | NULL    |                |
| gender    | enum('male','female') | NO   |     | NULL    |                |
| birthDate | date                 | YES  |     | NULL    |                |
| fName     | varchar(30)          | YES  |     | NULL    |                |
| lName     | varchar(30)          | YES  |     | NULL    |                |
+-----------+----------------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

Q5: Add foreign key constrains in Your Tables with options on delete cascaded

```
mysql> ALTER TABLE phone  add CONSTRAINT fk_class_id FOREIGN KEY (stuId)REFERENCES student(id) ON DELETE CASCADE;
Query OK, 5 rows affected (0.07 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE exam add CONSTRAINT fk_id FOREIGN KEY (stuId) REFERENCES student(id) ON DELETE CASCADE;
Query OK, 5 rows affected (0.08 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> ALTER TABLE exam add CONSTRAINT fk_subject_id FOREIGN KEY (subId) REFERENCES subject(id) ON DELETE CASCADE;
Query OK, 5 rows affected (0.08 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> ALTER TABLE exam
    -> DROP FOREIGN KEY exam_ibfk_1;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> ALTER TABLE exam
    -> DROP FOREIGN KEY exam_ibfk_2;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Q6: Update your information by changing data

```
mysql> UPDATE student SET gender = 'male', birthDate = '2001-12-03', fName = 'John', lName = 'Doe' WHERE Id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE student SET gender = 'female', birthDate = '2002-05-15', fName = 'Jane', lName = 'Smith' WHERE Id = 2;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE student SET gender = 'female', birthDate = '2003-08-20', fName = 'Emily', lName = 'Davis' WHERE Id = 3;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE student SET gender = 'male', birthDate = '2000-11-30', fName = 'Michael', lName = 'Brown' WHERE Id = 4;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE student SET gender = 'female', birthDate = '1999-03-10', fName = 'Sarah', lName = 'Johnson' WHERE Id = 5;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Q7: Display all students' information.

```
mysql> select * from student
    -> ;
+----+----------------------+-----------------+--------+------------+---------+---------+
| Id | email                | address         | gender | birthDate  | fName   | lName   |
+----+----------------------+-----------------+--------+------------+---------+---------+
|  1 | johndoe@example.com  | 123 Maple Street | male   | 2001-12-03 | John    | Doe     |
|  2 | janesmith@example.com | 456 Oak Avenue  | female | 2002-05-15 | Jane    | Smith   |
|  3 | emilyd@example.com   | 789 Pine Road   | female | 2003-08-20 | Emily   | Davis   |
|  4 | michaelb@example.com | 101 Birch Lane  | male   | 2000-11-30 | Michael | Brown   |
|  5 | sarahj@example.com   | 202 Cedar Drive | female | 1999-03-10 | Sarah   | Johnson |
+----+----------------------+-----------------+--------+------------+---------+---------+
5 rows in set (0.00 sec)
```

Q8: Display male students only.

```
mysql> select * from student where gender="male";
+----+----------------------+-----------------+--------+------------+---------+-------+
| Id | email                | address         | gender | birthDate  | fName   | lName |
+----+----------------------+-----------------+--------+------------+---------+-------+
|  1 | johndoe@example.com  | 123 Maple Street | male   | 2001-12-03 | John    | Doe   |
|  4 | michaelb@example.com | 101 Birch Lane  | male   | 2000-11-30 | Michael | Brown |
+----+----------------------+-----------------+--------+------------+---------+-------+
2 rows in set (0.00 sec)
```

Q9: Display the number of female students.

```
mysql> select count(gender)as numberOfFemale from student where gender="female";
+----------------+
| numberOfFemale |
+----------------+
|              3 |
+----------------+
1 row in set (0.00 sec)
```

Q10: Display the students who are born before 1992-10-01.

```
mysql> select * from student where birthDate <'1992-10-01';
Empty set (0.00 sec)
```

Q11: Display male students who are born before 1991-10-01.

```
mysql> select * from student where birthDate <'1992-10-01' and gender="male";
Empty set (0.00 sec)
```

Q12: Display subjects and their max score sorted by max score.

```
mysql> select * from subject;
+-----+-------------+----------+------------------------+
| id  | name        | maxScore | description            |
+-----+-------------+----------+------------------------+
| 101 | Mathematics |      100 | Algebra and Geometry   |
| 102 | Physics     |       99 | Mechanics and Optics   |
| 103 | Chemistry   |      102 | Organic and Inorganic  |
| 104 | Biology     |      100 | Genetics and Anatomy   |
| 105 | English     |      100 | Grammar and Literature |
+-----+-------------+----------+------------------------+
5 rows in set (0.00 sec)

mysql> select name,maxScore from subject order by maxScore;
+-------------+----------+
| name        | maxScore |
+-------------+----------+
| Physics     |       99 |
| Mathematics |      100 |
| Biology     |      100 |
| English     |      100 |
| Chemistry   |      102 |
+-------------+----------+
5 rows in set (0.00 sec)
```

Q13: Display the subject with highest max score

```
mysql> select name,maxScore from subject order by maxScore Desc limit 1;
+-----------+----------+
| name      | maxScore |
+-----------+----------+
| Chemistry |      102 |
+-----------+----------+
1 row in set (0.00 sec)
```

Q14: Display students' names that begin with A.

```
mysql> select * from student where fName like 'a%';
Empty set (0.00 sec)
```

Q15: Display the number of students' their name is "Mohammed"

```
mysql> select count(fName) as number from student where fName="mohammed";
+--------+
| number |
+--------+
|      0 |
+--------+
1 row in set (0.00 sec)
```

Q16: Display the number of males and females.

```
mysql> select gender,count(gender) from student group by gender;
+--------+---------------+
| gender | count(gender) |
+--------+---------------+
| male   |             2 |
| female |             3 |
+--------+---------------+
2 rows in set (0.00 sec)
```

Q17: Display the repeated first names and their counts if higher than 2.

```
mysql> SELECT fName, COUNT(*) AS count
    -> FROM student
    -> GROUP BY fName
    -> HAVING COUNT(*) > 2;
Empty set (0.00 sec)
```

Q18: Display students' names, their score and subject name.

```
mysql> select concat(st.fName," ",st.lName) as fullName,s.name,stuScore from student st join exam e on st.id=e.stuId joi
n subject s on s.id=subId;
+--------------+------------+----------+
| fullName     | name       | stuScore |
+--------------+------------+----------+
| John Doe     | Mathematics |       85 |
| Jane Smith   | Physics    |       90 |
| Emily Davis  | Chemistry  |       78 |
| Michael Brown | Biology    |       88 |
| Sarah Johnson | English    |       92 |
+--------------+------------+----------+
5 rows in set (0.00 sec)
```

Q19: Delete students their score is lower than 50 in a particular subject exam.

```
mysql> DELETE st from student st join exam e on st.id=e.stuId join subject s on s.id=e.subId where e.stuScore<50 and s.name="English";
Query OK, 1 row affected (0.01 sec)
```