Lab 3 - More table operations

Alter

- All changes made to the table structure are done so using ALTER

Types of Alter Commands

- **ADD** – adds a column to the table
- **MODIFY** – changes the column data types and data constraints
- **DROP** – deletes the column

Task one: Adding a column

ALTER TABLE <TABLE_NAME> ADD COLUMN COLUMN_NAME DATA_TYPE

Eg. ALTER TABLE CARS ADD COLUMN NUMBER_PLATE VARCHAR(10)

1. Now your task is to add to the STUDENT_NEW table from lab 2. You will have to add the student's name to the table as a column (Note think of what the data type should be and its length). Then display your results

```
mysql> DESC STUDENT_NEW;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| STUDENT_NO   | int(10)     | YES  |     | NULL    |       |
| STUDENT_NAME | varchar(10) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

Task 2: Modify a column

ALTER TABLE <TABLE_NAME> MODIFY COLUMN COLUMN_NAME DATA_TYPE

Eg ALTER TABLE CARS MODIFY COLUMN OWNERS_NAME VARCHAR(30)

Let's assume that the table has an owner name column which was previously a varchar of length 10 which is obviously too short thus we have now corrected it to a length of 30

2. Your task is to modify the STUDENT_NEW table. Modify the column called student name to contain 50 characters. Display the display to visualize the changes

```
mysql> DESC STUDENT_NEW;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| STUDENT_NO   | int(10)     | YES  |     | NULL    |       |
| STUDENT_NAME | varchar(20) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

3. The university decides to change the student number system to now contain a mix of letters and numbers. The maximum length of the student number should be 10 units. MODIFY the datatype to accommodate these

```
mysql> DESC STUDENT_NEW;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| STUDENT_NO   | varchar(10) | YES  |     | NULL    |       |
| STUDENT_NAME | varchar(20) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

4. Now the head of the school says that we should never have a null value in the student number column. Modify the column once again to accommodate these changes.

    a. Eg. A car has to have a number plate
    ALTER TABLE CARS MODIFY COLUMN OWNERS_NAME VARCHAR(30) NOT NULL;

```
mysql> DESC STUDENT_NEW;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| STUDENT_NO   | varchar(10) | NO   |     | NULL    |       |
| STUDENT_NAME | varchar(20) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

5. The next task we are required to ensure that every students name is unique. Use the key word “UNIQUE” in the same manner as “NOT NULL” from the example above to incorporate these changes.

```
mysql> DESC STUDENT_NEW;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| STUDENT_NO   | varchar(10) | NO   |     | NULL    |       |
| STUDENT_NAME | varchar(20) | YES  | UNI | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

6. Since the requirements of the student number changed from task 3, we now need to change the name of the column to be called STUDENT_NEW_NO

   Eg ALTER TABLE CARS CHANGE NUMBER_PLATE NUMBER_PLATE_NEW VARCHAR(10)

   TABLENAME
   CURRENT COLUMN NAME
   NEW COLUMN NAME
   DATATYPE

```
mysql> DESC STUDENT_NEW;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| STUDENT_NEW_NO | varchar(10) | YES  |     | NULL    |       |
| STUDENT_NAME   | varchar(20) | YES  | UNI | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```

ALTER + DROP

7. Now we did all this hard work only to find out that we don’t even need the student name column after all. DROP THE STUDENT_NAME COLUMN

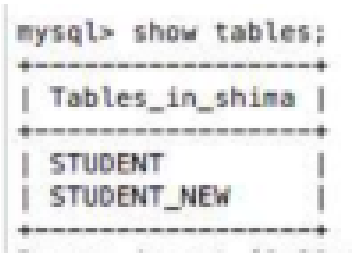
   ALTER TABLE <TABLE_NAME> DROP COLUMN_NAME;

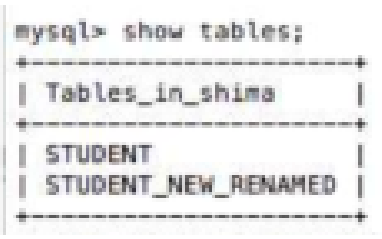
```
mysql> DESC STUDENT_NEW;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| STUDENT_ID | varchar(10) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

RENAME

8. Rename the table called STUDENT_NEW to STUDENT_NEW_RENAMED

RENAME TABLE <CURRENT_TABLE_NAME> TO <NEW_TABLE_NAME>





TRUNCATE

This is used to remove all the data from a table. However, it will retain the table structure

TRUNCATE <OLD_TABLE_NAME> TO <NEW_TABLE_NAME>

Follow these instructions for the truncation to apply it:

1. Go into the table called STUDENT_NEW_RENAMED
2. Insert a value into this table:
    a. INSERT INTO STUDENT_NEW_RENAMED VALUES('1922' , 'MIKHAIL');
3. Display the entries into the table using SELECT
    a. SELECT * FROM STUDENT_NEW_RENAMED
4. Apply the truncation:
    a. TRUNCATE STUDENT_NEW_RENAMED
5. Display the results to see its an empty table

a. SELECT * FROM STUDENT_NEW_RENAMED;

```
mysql> SELECT * FROM STUDENT_NEW_RENAMED;
Empty set (0.00 sec)
```

ADDITIONAL RESOURCES

READ THE CHAPTER ON SQL IN THE TEXTBOOK

MySQL REFERENCE ON ALTER TABLE

https://dev.mysql.com/doc/refman/8.0/en/alter-table.html