1:

DBMS: LAB 3

DDL (Data Definition Language) Commands

University Fest Management System

OBJECTIVE: To learn and understand DDL statements while executing queries.

DDL (Data Definition Language) queries in MySQL are used to define and manage the structure and characteristics of your database and its objects.

Some of the important DDL commands are:

- **CREATE:** This DDL query establishes a new database object, such as a table or an index, with defined attributes and structure.
- **ALTER:** The ALTER query modifies the structure of an existing database object, enabling tasks like adding or modifying columns.
- **TRUNCATE:** This DDL query removes all the data from a table while retaining the table structure, resulting in improved performance compared to the DROP TABLE query.
- **DROP:** This DDL query deletes a database object, such as a table or an index, along with its associated data irreversibly.
- **RENAME:** The RENAME query changes the name of an existing database object, like altering the name of a table.

INSTRUCTIONS:

- As a part of LAB 3, there are 2 tasks that are to be completed wrt to the case study shared earlier with ER diagram and Relational Schema of University Fest Management.
  - o **TASK 1:** As per the given Description, ER diagram and Relational Schema – create the tables using DDL commands and add the required constraints.
  - o **TASK 2:** There are certain questions that have been given. These are to be executed on the DDL statements created in TASK 1.
- As a part of the submission process, the following are to be submitted:
  - o A **PDF** document, containing all the Screenshots for both tasks as suggested
     Name of the file: `<your SRN>_University_Fest_DB_Lab3.pdf o The **“.sql”** file for the same, shall contain all the commands that have been executed in the lab
     Name of the file: `<your SRN>_University_Fest_DB_Lab3.sql

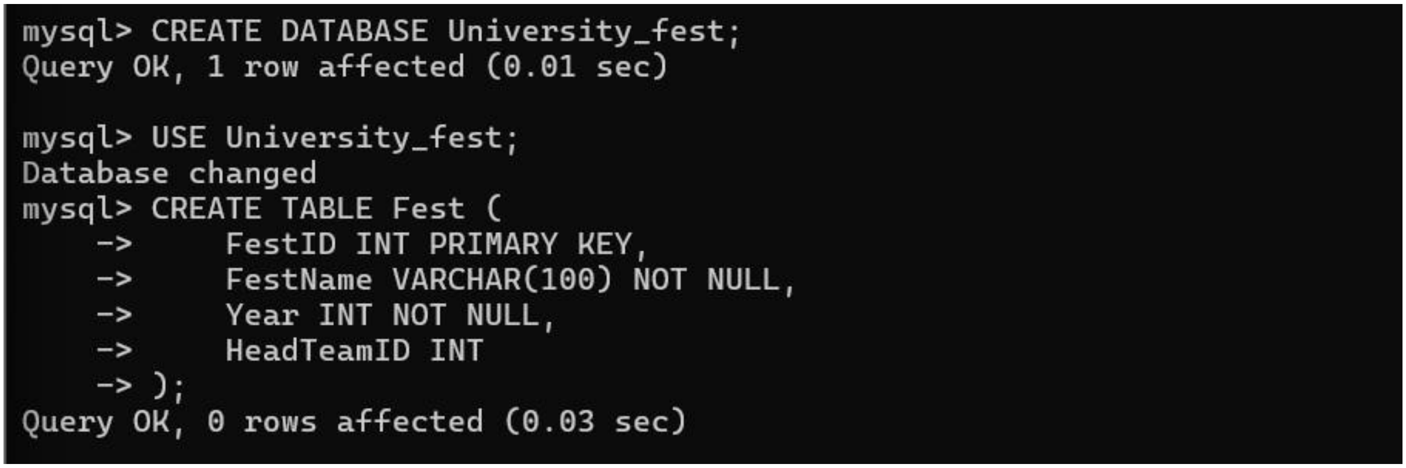
Example:

Refer to the sample submissions given below. This will give you an idea about the details that must be included in your submissions

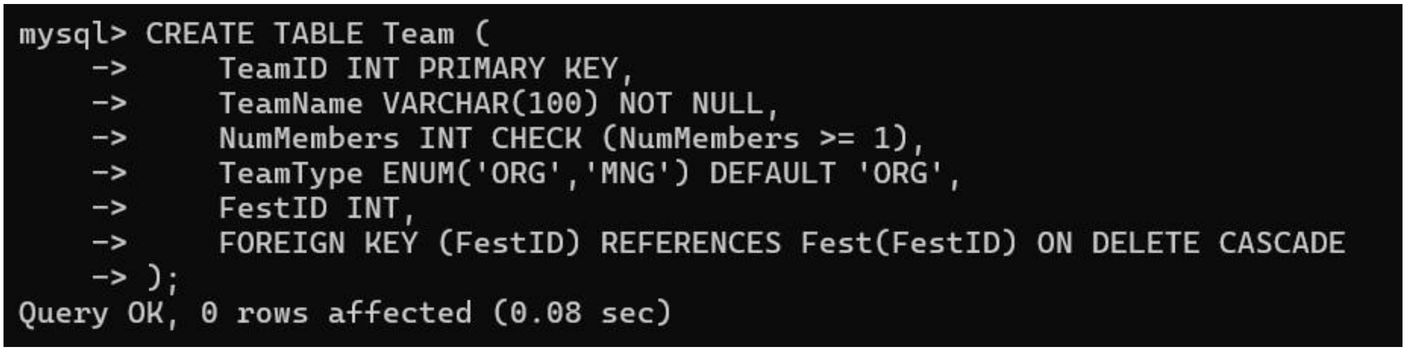
NOTE: Screenshots can be taken either from “**MySQL workbench**” or “**Command Line**” **Task**

1:

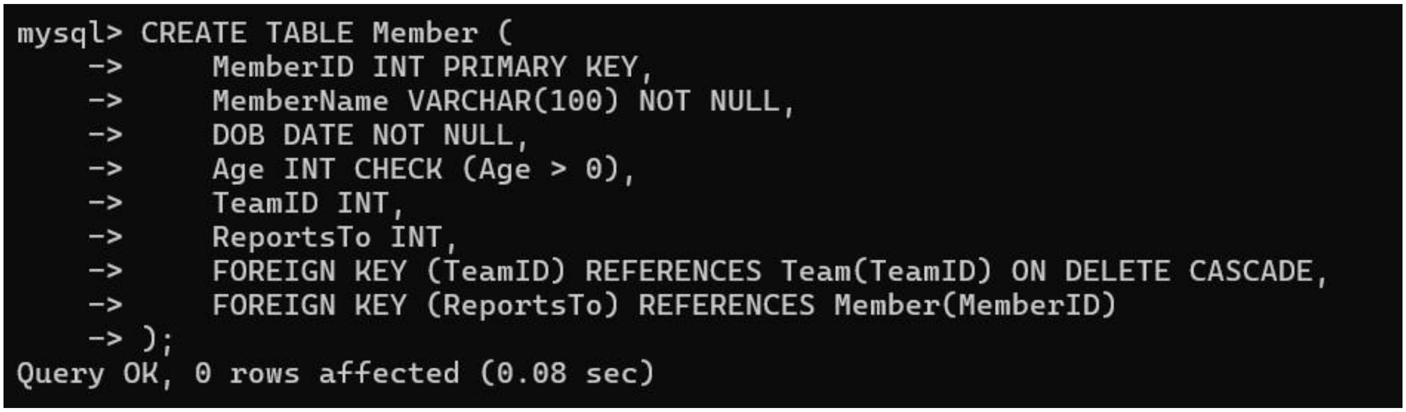
Take the screen shot of every table created along with the create table sql command.



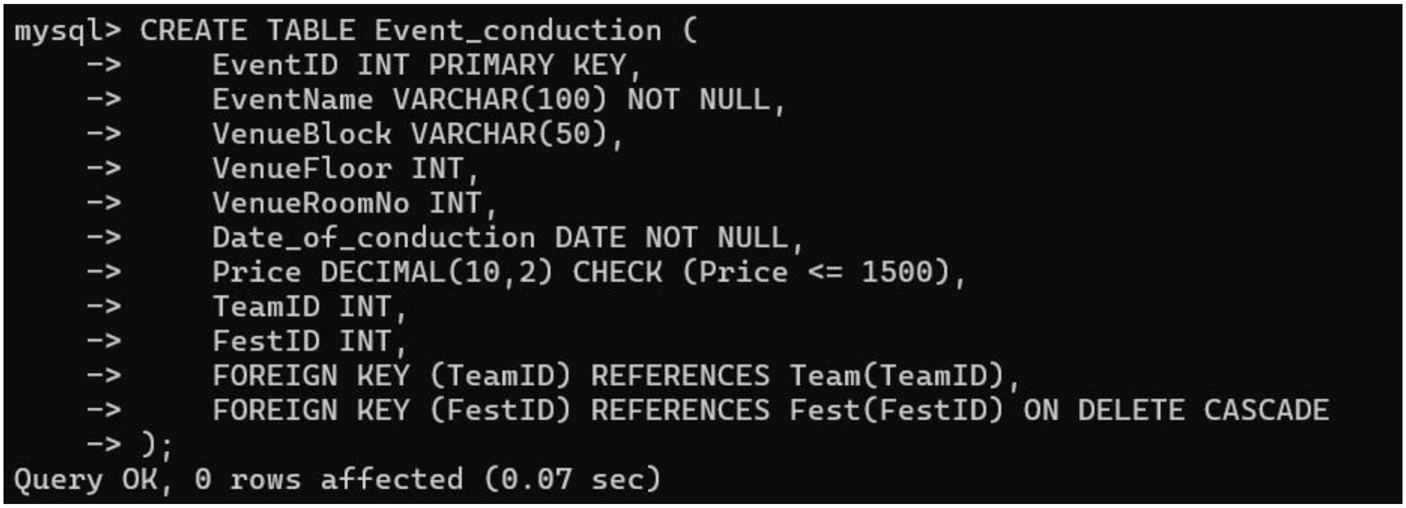
```
mysql> CREATE DATABASE University_fest;
Query OK, 1 row affected (0.01 sec)

mysql> USE University_fest;
Database changed
mysql> CREATE TABLE Fest (
    ->     FestID INT PRIMARY KEY,
    ->     FestName VARCHAR(100) NOT NULL,
    ->     Year INT NOT NULL,
    ->     HeadTeamID INT
    -> );
Query OK, 0 rows affected (0.03 sec)
```



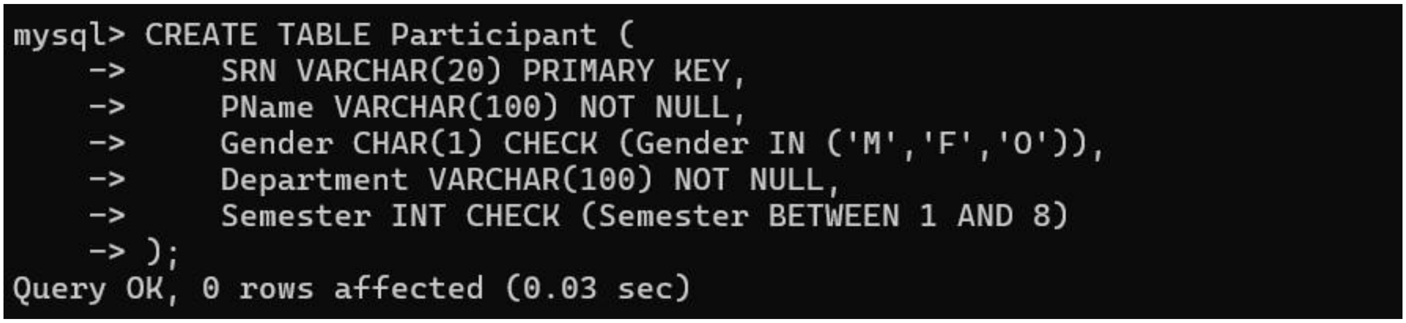
```
mysql> CREATE TABLE Team (
    ->     TeamID INT PRIMARY KEY,
    ->     TeamName VARCHAR(100) NOT NULL,
    ->     NumMembers INT CHECK (NumMembers >= 1),
    ->     TeamType ENUM('ORG','MNG') DEFAULT 'ORG',
    ->     FestID INT,
    ->     FOREIGN KEY (FestID) REFERENCES Fest(FestID) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.08 sec)
```



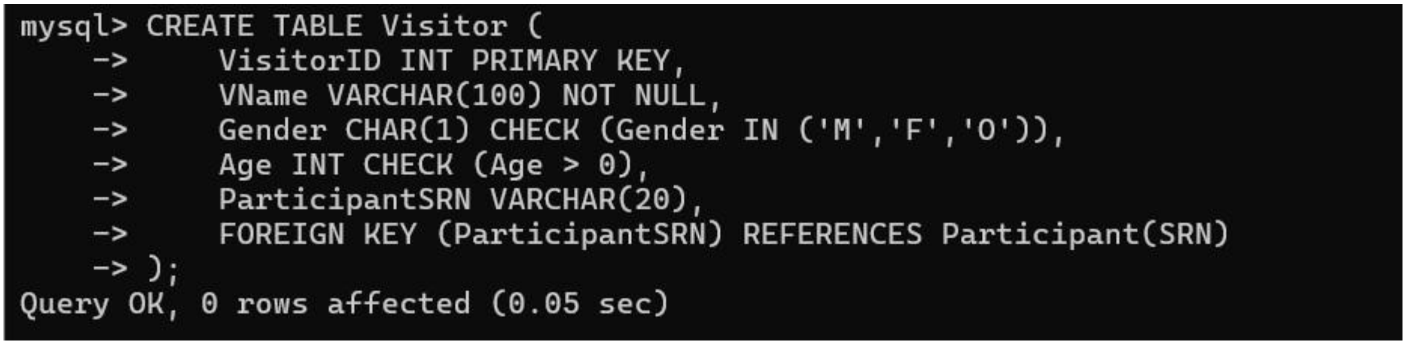
```
mysql> CREATE TABLE Member (
    ->     MemberID INT PRIMARY KEY,
    ->     MemberName VARCHAR(100) NOT NULL,
    ->     DOB DATE NOT NULL,
    ->     Age INT CHECK (Age > 0),
    ->     TeamID INT,
    ->     ReportsTo INT,
    ->     FOREIGN KEY (TeamID) REFERENCES Team(TeamID) ON DELETE CASCADE,
    ->     FOREIGN KEY (ReportsTo) REFERENCES Member(MemberID)
    -> );
Query OK, 0 rows affected (0.08 sec)
```



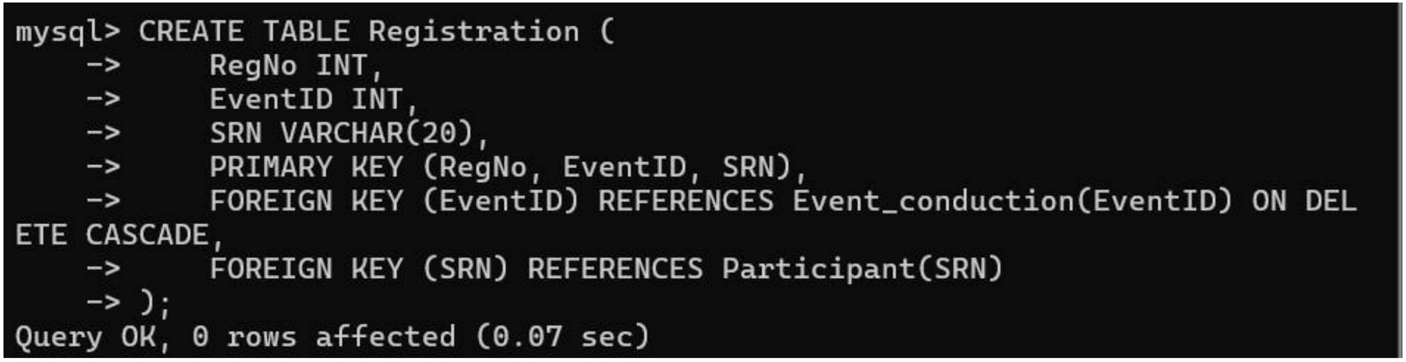
```
mysql> CREATE TABLE Event_conduction (
    ->     EventID INT PRIMARY KEY,
    ->     EventName VARCHAR(100) NOT NULL,
    ->     VenueBlock VARCHAR(50),
    ->     VenueFloor INT,
    ->     VenueRoomNo INT,
    ->     Date_of_conduction DATE NOT NULL,
    ->     Price DECIMAL(10,2) CHECK (Price <= 1500),
    ->     TeamID INT,
    ->     FestID INT,
    ->     FOREIGN KEY (TeamID) REFERENCES Team(TeamID),
    ->     FOREIGN KEY (FestID) REFERENCES Fest(FestID) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.07 sec)
```



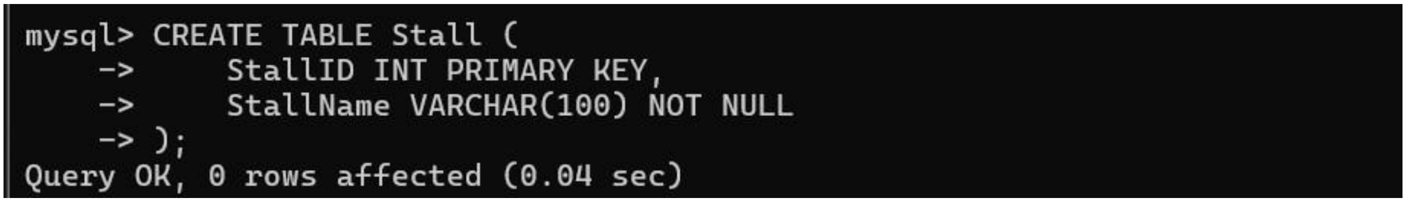
```
mysql> CREATE TABLE Participant (
    ->     SRN VARCHAR(20) PRIMARY KEY,
    ->     PName VARCHAR(100) NOT NULL,
    ->     Gender CHAR(1) CHECK (Gender IN ('M','F','O')),
    ->     Department VARCHAR(100) NOT NULL,
    ->     Semester INT CHECK (Semester BETWEEN 1 AND 8)
    -> );
Query OK, 0 rows affected (0.03 sec)
```



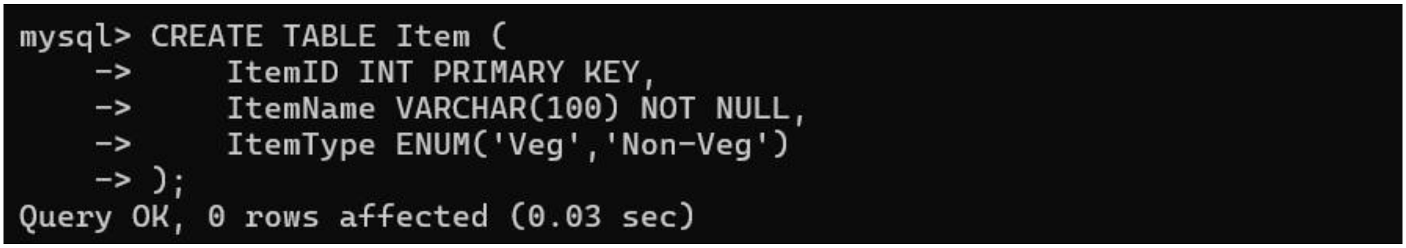
```
mysql> CREATE TABLE Visitor (
    ->     VisitorID INT PRIMARY KEY,
    ->     VName VARCHAR(100) NOT NULL,
    ->     Gender CHAR(1) CHECK (Gender IN ('M','F','O')),
    ->     Age INT CHECK (Age > 0),
    ->     ParticipantSRN VARCHAR(20),
    ->     FOREIGN KEY (ParticipantSRN) REFERENCES Participant(SRN)
    -> );
Query OK, 0 rows affected (0.05 sec)
```



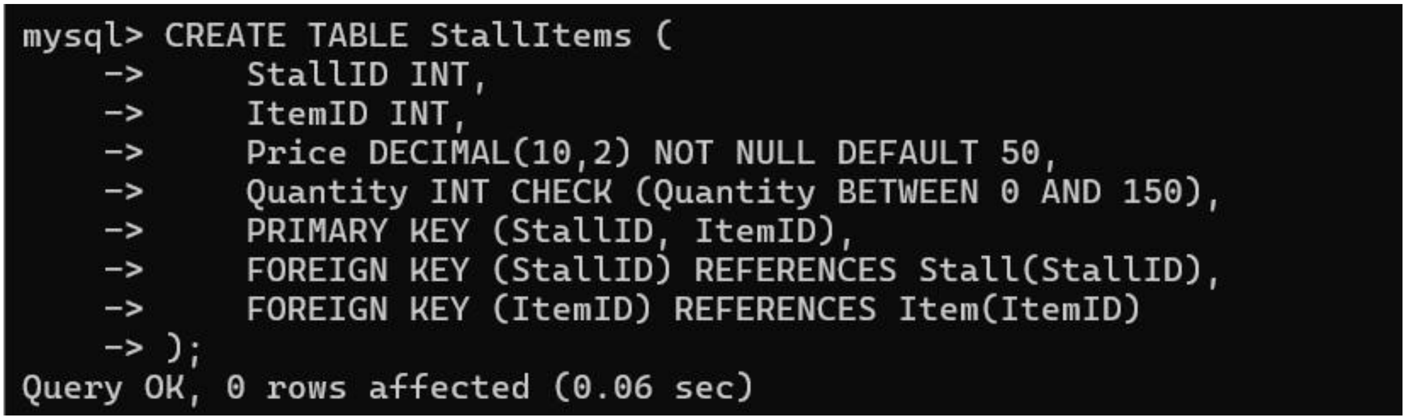
```
mysql> CREATE TABLE Registration (
    ->     RegNo INT,
    ->     EventID INT,
    ->     SRN VARCHAR(20),
    ->     PRIMARY KEY (RegNo, EventID, SRN),
    ->     FOREIGN KEY (EventID) REFERENCES Event_conduction(EventID) ON DEL
ETE CASCADE,
    ->     FOREIGN KEY (SRN) REFERENCES Participant(SRN)
    -> );
Query OK, 0 rows affected (0.07 sec)
```



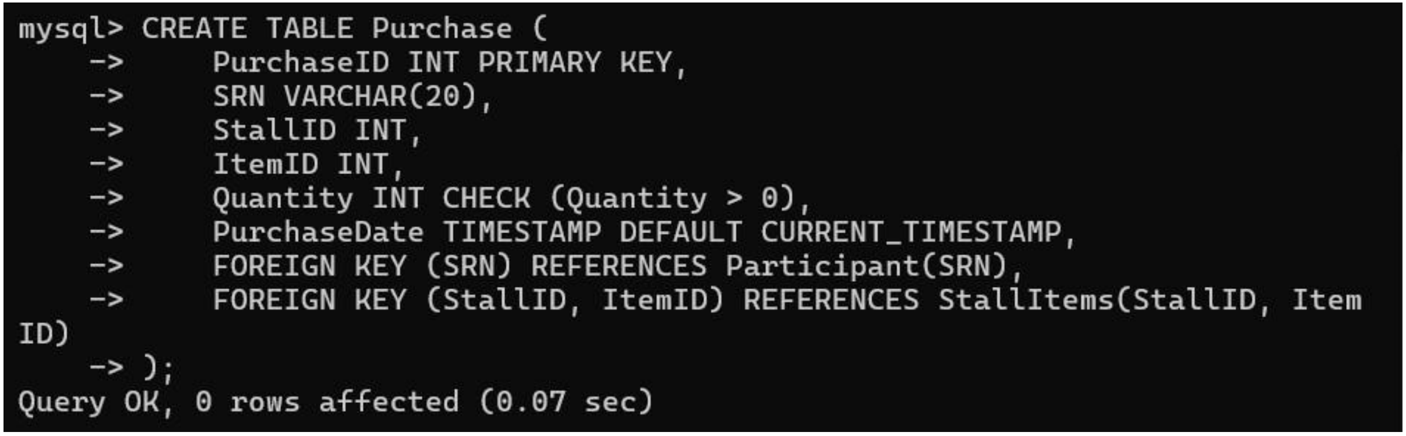
```
mysql> CREATE TABLE Stall (
    ->     StallID INT PRIMARY KEY,
    ->     StallName VARCHAR(100) NOT NULL
    -> );
Query OK, 0 rows affected (0.04 sec)
```



```
mysql> CREATE TABLE Item (
    ->     ItemID INT PRIMARY KEY,
    ->     ItemName VARCHAR(100) NOT NULL,
    ->     ItemType ENUM('Veg','Non-Veg')
    -> );
Query OK, 0 rows affected (0.03 sec)
```



```
mysql> CREATE TABLE StallItems (
    ->     StallID INT,
    ->     ItemID INT,
    ->     Price DECIMAL(10,2) NOT NULL DEFAULT 50,
    ->     Quantity INT CHECK (Quantity BETWEEN 0 AND 150),
    ->     PRIMARY KEY (StallID, ItemID),
    ->     FOREIGN KEY (StallID) REFERENCES Stall(StallID),
    ->     FOREIGN KEY (ItemID) REFERENCES Item(ItemID)
    -> );
Query OK, 0 rows affected (0.06 sec)
```



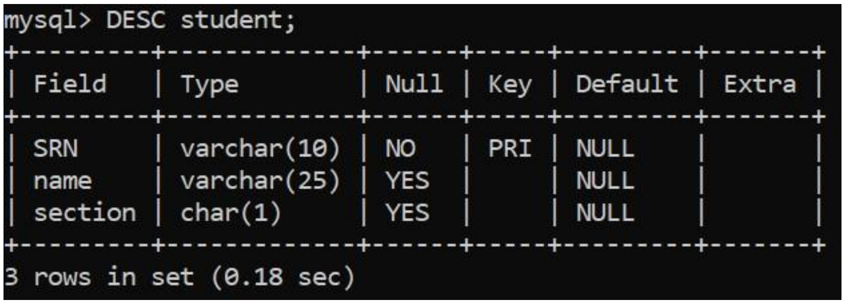
```
mysql> CREATE TABLE Purchase (
    ->     PurchaseID INT PRIMARY KEY,
    ->     SRN VARCHAR(20),
    ->     StallID INT,
    ->     ItemID INT,
    ->     Quantity INT CHECK (Quantity > 0),
    ->     PurchaseDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ->     FOREIGN KEY (SRN) REFERENCES Participant(SRN),
    ->     FOREIGN KEY (StallID, ItemID) REFERENCES StallItems(StallID, Item
ID)
    -> );
Query OK, 0 rows affected (0.07 sec)
```

Task 2:

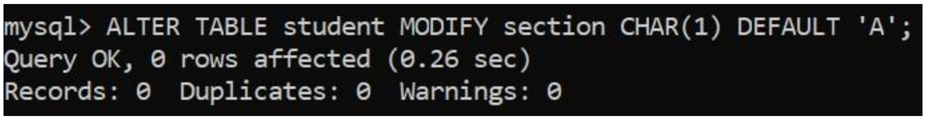
Task 2 involves modifying the structure of the tables created in Task 1. For every modification made, **3 screenshots** are required:

1. Structure of the table **before** modification
2. **DDL statement**
3. Structure of the table **after** modification

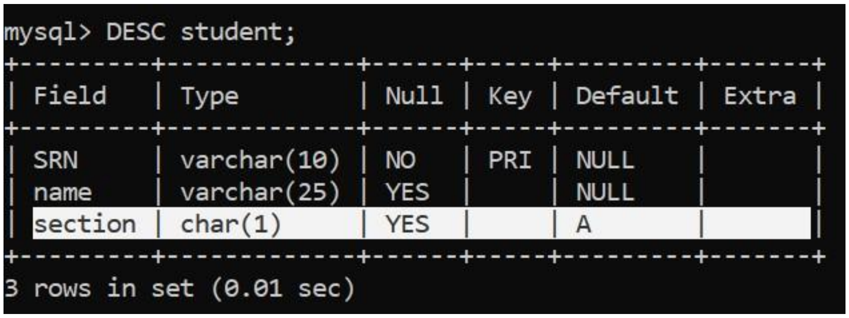
Sample submission:



```
mysql> DESC student;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| SRN     | varchar(10) | NO   | PRI | NULL    |       |
| name    | varchar(25) | YES  |     | NULL    |       |
| section | char(1)     | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.18 sec)
```



```
mysql> ALTER TABLE student MODIFY section CHAR(1) DEFAULT 'A';
Query OK, 0 rows affected (0.26 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



```
mysql> DESC student;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| SRN     | varchar(10) | NO   | PRI | NULL    |       |
| name    | varchar(25) | YES  |     | NULL    |       |
| section | char(1)     | YES  |     | A       |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

TASK 1:

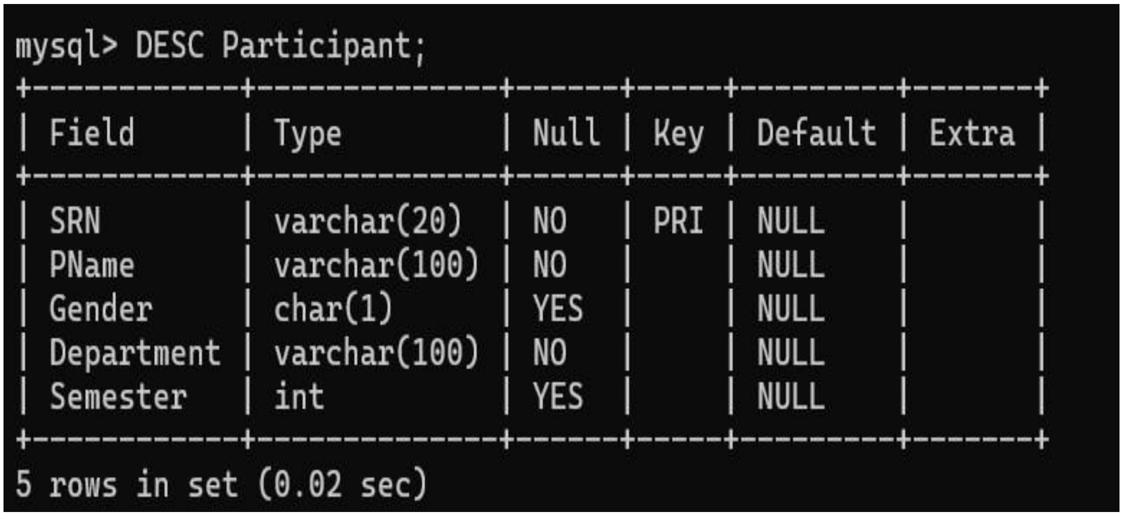
Identify all the constraints (domain, key, constraint on null, primary key, foreign key and check etc) based on the ER and description given and execute the DDL commands for University_fest database.

Add all the screenshots of sql command and desc table.

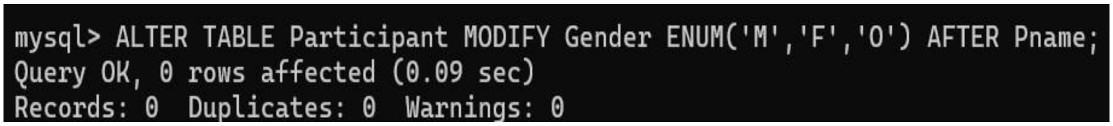
TASK 2:

1. **Modify the datatype of the gender attribute to make sure that the only values that can be stored are M: for male, F: for female, and O: for other. Also, make sure that the gender attribute is positioned after the "name" column.**

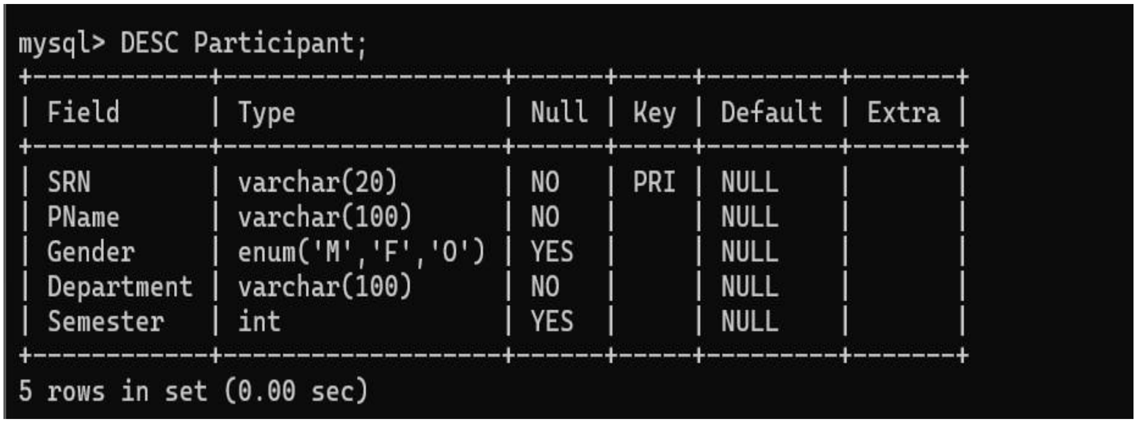
Before: --describe the table/s
Command – sql command
Afer: --describe the table/s



```
mysql> DESC Participant;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| SRN        | varchar(20)  | NO   | PRI | NULL    |       |
| PName      | varchar(100) | NO   |     | NULL    |       |
| Gender     | char(1)      | YES  |     | NULL    |       |
| Department | varchar(100) | NO   |     | NULL    |       |
| Semester   | int          | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
5 rows in set (0.02 sec)
```



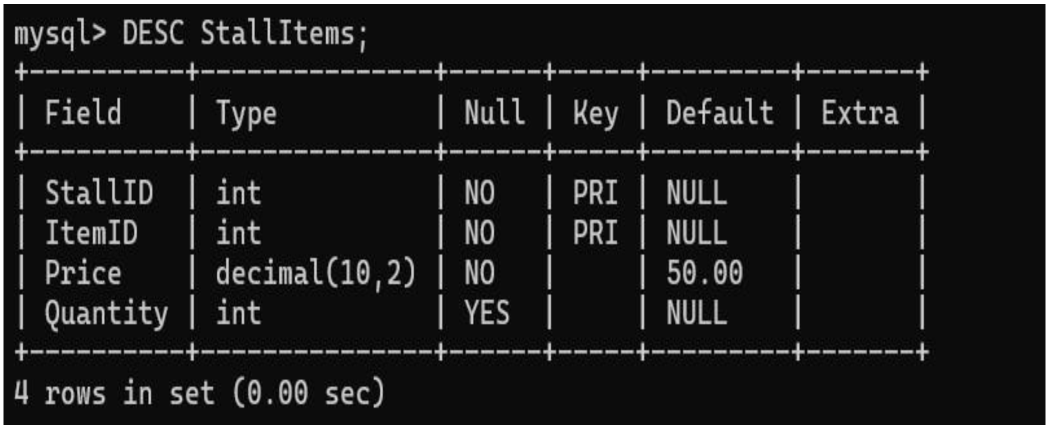
```
mysql> ALTER TABLE Participant MODIFY Gender ENUM('M','F','O') AFTER Pname;
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



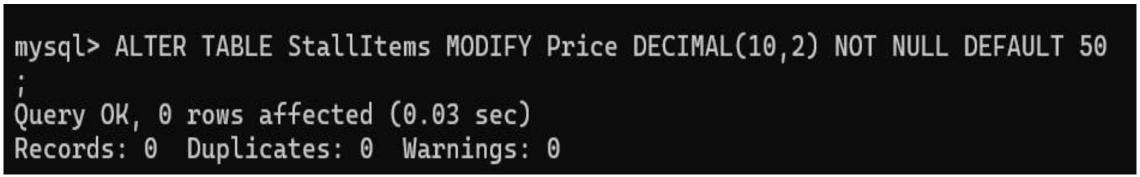
```
mysql> DESC Participant;
+------------+-------------------+------+-----+---------+-------+
| Field      | Type              | Null | Key | Default | Extra |
+------------+-------------------+------+-----+---------+-------+
| SRN        | varchar(20)       | NO   | PRI | NULL    |       |
| PName      | varchar(100)      | NO   |     | NULL    |       |
| Gender     | enum('M','F','O') | YES  |     | NULL    |       |
| Department | varchar(100)      | NO   |     | NULL    |       |
| Semester   | int               | YES  |     | NULL    |       |
+------------+-------------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

2. **Every stall would offer items for different prices, it is found in the previous fests that the price of most of the items was 50. Therefore as a DB designer, set the default value of the prices of items to be 50 rupees and also make sure that every item has a price associated with it meaning we cannot have a null value entered into the price.**

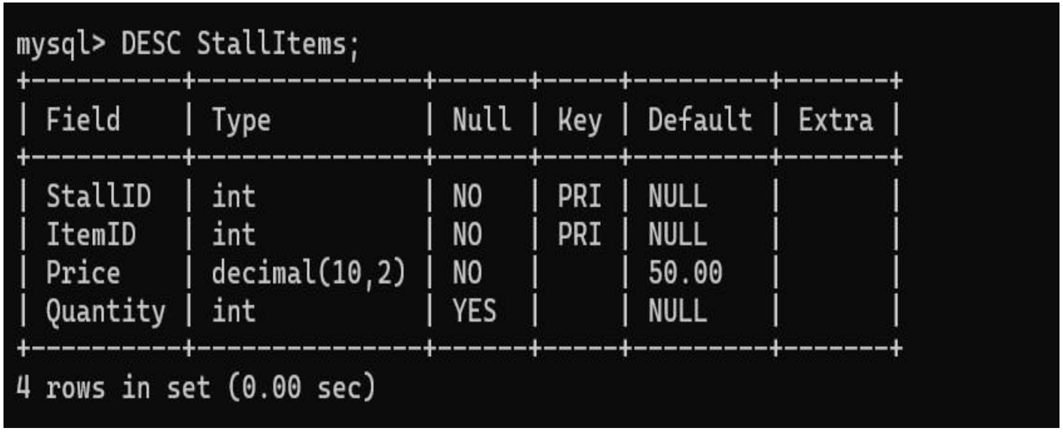
   Before: --describe the table/s
   Command – sql command
   Afer: --describe the table/s



```
mysql> DESC StallItems;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| StallID  | int           | NO   | PRI | NULL    |       |
| ItemID   | int           | NO   | PRI | NULL    |       |
| Price    | decimal(10,2) | NO   |     | 50.00   |       |
| Quantity | int           | YES  |     | NULL    |       |
+----------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```



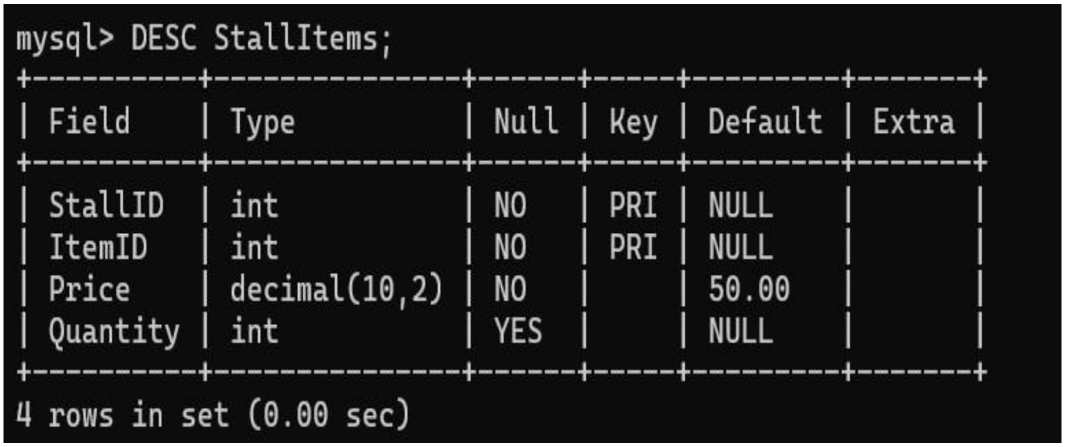
```
mysql> ALTER TABLE StallItems MODIFY Price DECIMAL(10,2) NOT NULL DEFAULT 50
;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



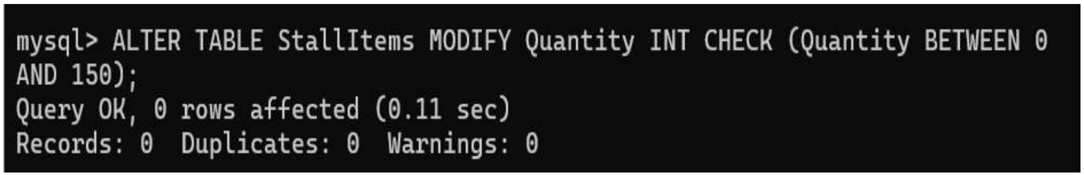
```
mysql> DESC StallItems;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| StallID  | int           | NO   | PRI | NULL    |       |
| ItemID   | int           | NO   | PRI | NULL    |       |
| Price    | decimal(10,2) | NO   |     | 50.00   |       |
| Quantity | int           | YES  |     | NULL    |       |
+----------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

3. **Considering that the stalls have a limited space for storing the various items they sell, create a max_stocks condition that ensures that a particular stall can at max have 150 units of each item that they sell.**

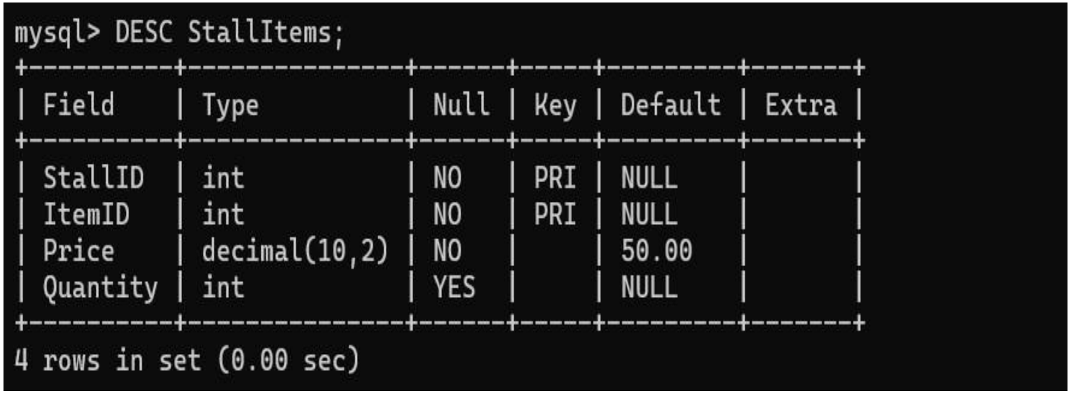
Before: --describe the table/s
Command – sql command
Afer: --describe the table/s



```
mysql> DESC StallItems;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| StallID  | int           | NO   | PRI | NULL    |       |
| ItemID   | int           | NO   | PRI | NULL    |       |
| Price    | decimal(10,2) | NO   |     | 50.00   |       |
| Quantity | int           | YES  |     | NULL    |       |
+----------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```



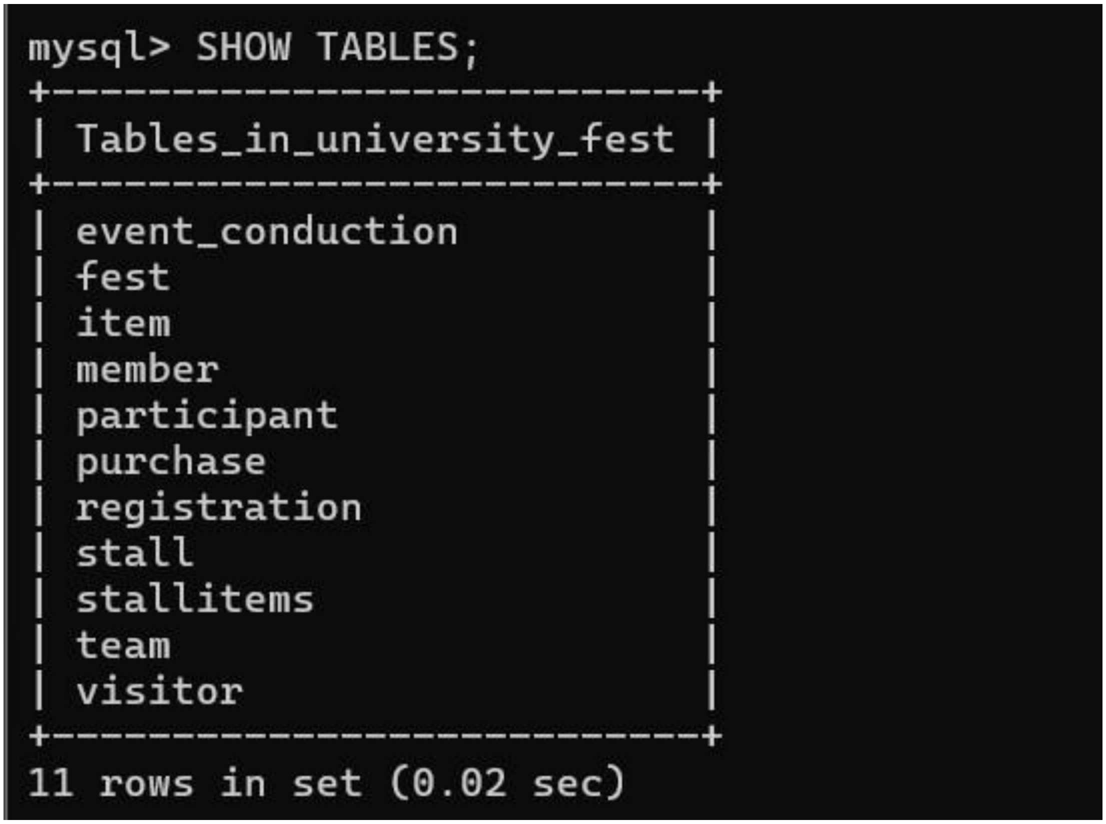
```
mysql> ALTER TABLE StallItems MODIFY Quantity INT CHECK (Quantity BETWEEN 0
AND 150);
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



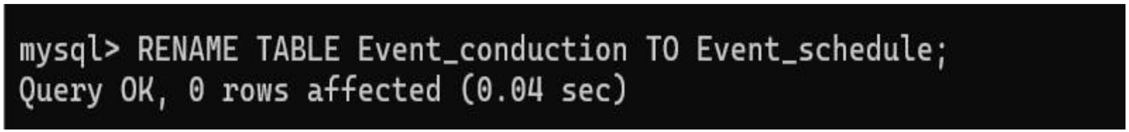
```
mysql> DESC StallItems;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| StallID  | int           | NO   | PRI | NULL    |       |
| ItemID   | int           | NO   | PRI | NULL    |       |
| Price    | decimal(10,2) | NO   |     | 50.00   |       |
| Quantity | int           | YES  |     | NULL    |       |
+----------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

4: Rename the table “Event_conduction” to “Event_schedule”

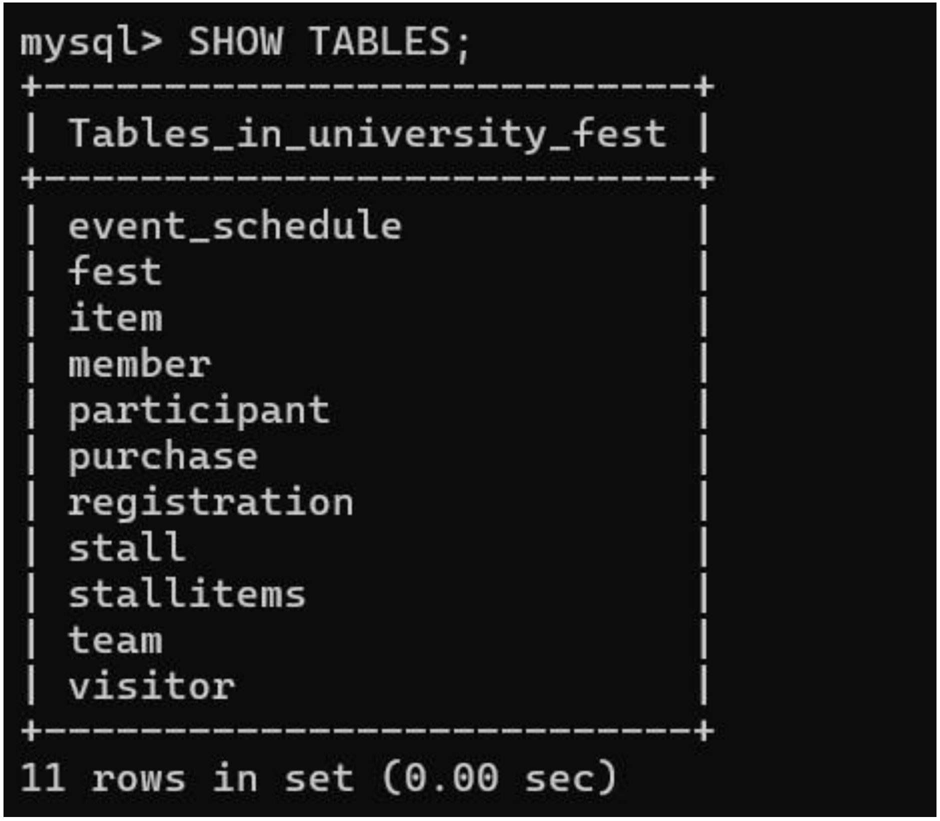
Before: --describe the table/s
Command – sql command
Afer: --describe the table/s



```
mysql> SHOW TABLES;
+---------------------------+
| Tables_in_university_fest |
+---------------------------+
| event_conduction          |
| fest                      |
| item                      |
| member                    |
| participant               |
| purchase                  |
| registration              |
| stall                     |
| stallitems                |
| team                      |
| visitor                   |
+---------------------------+
11 rows in set (0.02 sec)
```



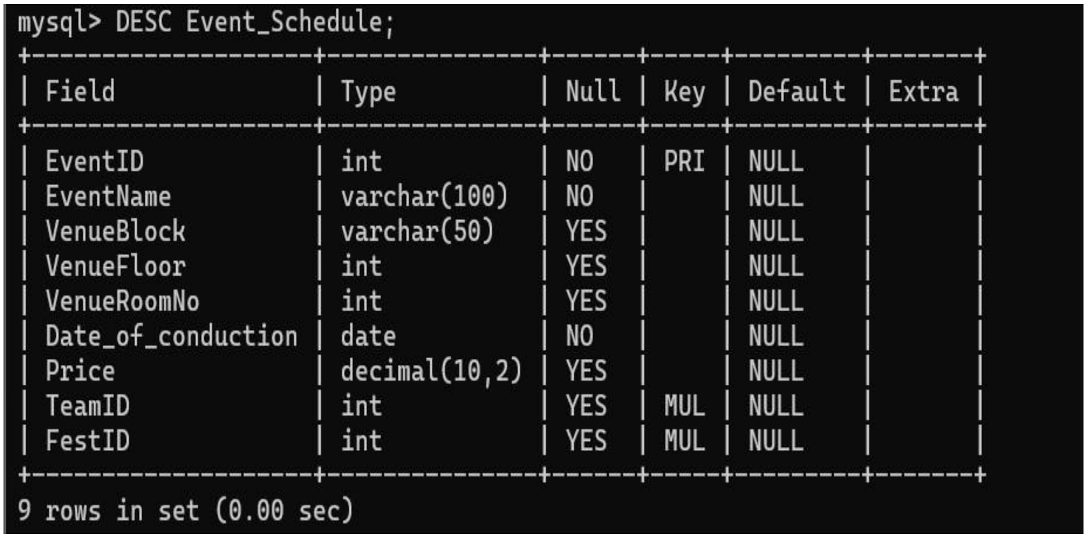
```
mysql> RENAME TABLE Event_conduction TO Event_schedule;
Query OK, 0 rows affected (0.04 sec)
```



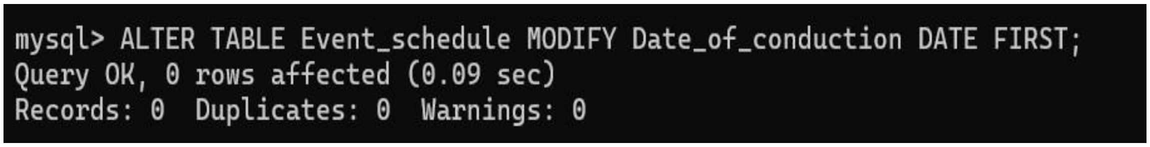
```
mysql> SHOW TABLES;
+---------------------------+
| Tables_in_university_fest |
+---------------------------+
| event_schedule            |
| fest                      |
| item                      |
| member                    |
| participant               |
| purchase                  |
| registration              |
| stall                     |
| stallitems                |
| team                      |
| visitor                   |
+---------------------------+
11 rows in set (0.00 sec)
```

5: Move the column "Date_of_conduction" such that it’s the first column

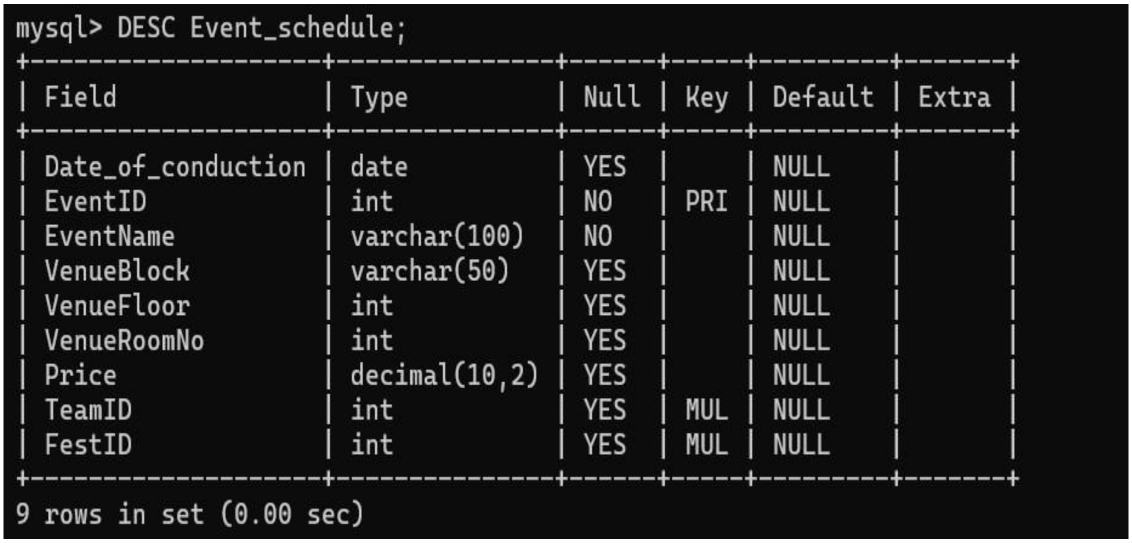
Before: --describe the table/s
Command – sql command
Afer: --describe the table/s



```
mysql> DESC Event_Schedule;
+--------------------+---------------+------+-----+---------+-------+
| Field              | Type          | Null | Key | Default | Extra |
+--------------------+---------------+------+-----+---------+-------+
| EventID            | int           | NO   | PRI | NULL    |       |
| EventName          | varchar(100)  | NO   |     | NULL    |       |
| VenueBlock         | varchar(50)   | YES  |     | NULL    |       |
| VenueFloor         | int           | YES  |     | NULL    |       |
| VenueRoomNo        | int           | YES  |     | NULL    |       |
| Date_of_conduction | date          | NO   |     | NULL    |       |
| Price              | decimal(10,2) | YES  |     | NULL    |       |
| TeamID             | int           | YES  | MUL | NULL    |       |
| FestID             | int           | YES  | MUL | NULL    |       |
+--------------------+---------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```



```
mysql> ALTER TABLE Event_schedule MODIFY Date_of_conduction DATE FIRST;
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



```
mysql> DESC Event_schedule;
+--------------------+---------------+------+-----+---------+-------+
| Field              | Type          | Null | Key | Default | Extra |
+--------------------+---------------+------+-----+---------+-------+
| Date_of_conduction | date          | YES  |     | NULL    |       |
| EventID            | int           | NO   | PRI | NULL    |       |
| EventName          | varchar(100)  | NO   |     | NULL    |       |
| VenueBlock         | varchar(50)   | YES  |     | NULL    |       |
| VenueFloor         | int           | YES  |     | NULL    |       |
| VenueRoomNo        | int           | YES  |     | NULL    |       |
| Price              | decimal(10,2) | YES  |     | NULL    |       |
| TeamID             | int           | YES  | MUL | NULL    |       |
| FestID             | int           | YES  | MUL | NULL    |       |
+--------------------+---------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```

Task 3: Few questions for you to answer:

1: Which is the sql command to know the current database in MySQL?

SELECT DATABASE();

2: Which is the sql command to clear the command prompt window of MySQL?

system cls;

3: Can you rename the database in MySQL?

CREATE DATABASE new_db;

RENAME TABLE old_db.table1 TO new_db.table1;

DROP DATABASE old_db;

4: What is the command to remove a table along with its structure?

DROP TABLE table_name;

5: Specify the difference between drop table and truncate table?

DROP TABLE- Removes the table **and its structure** permanently.

TRUNCATE TABLE- Deletes **all rows** but keeps the **table structure** for reuse.

6: Can a table have more than one primary key?

No.

7: Can a foreign key value be null?

Yes.

8: Can a primary key value be null? Which constraint is this?

No.

9: Upon describing the table using the command “desc tablename” what information about the table is given.

Column name

Data type

NULL / NOT NULL info

Key type (Primary, Foreign, Unique, etc.)

Default value

Extra info (e.g., AUTO_INCREMENT)

10: Can a primary key for a table be changed? If yes how?

Yes, By dropping the existing primary key and adding a new one:

ALTER TABLE table_name DROP PRIMARY KEY;

ALTER TABLE table_name ADD PRIMARY KEY (new_column);