Title: SQL Table Operations

Objective:

To practice basic SQL operations such as creating tables, modifying columns, adding constraints and managing data using SQL commands.

Description:

This project involves creating and modifying two tables—employee and department—in a database.
Key tasks include adding and altering columns, setting primary and foreign keys and performing delete and truncate operations. It helps build foundational SQL skills for managing relational databases.

1. Create a table called employee with the following structure And 2) Allow NULL for all columns except E_ID & E_NAME:

Column Name	Data Type
E_ID	INT
E_NAME	VARCHAR(20)
E_ADDRESS	VARCHAR(20)
D_ID	INT

SQL Queries:

```
CREATE DATABASE lab1;
USE lab1;

CREATE TABLE employee (
    E_ID INT NOT NULL,
    E_NAME VARCHAR(20) NOT NULL,
    E_ADDRESS VARCHAR(20),
    D_ID INT
);
```

Output

#	Name	Type	Collation	Attributes	Null	Default
1	E_ID	int(11)			No	None
2	E_NAME	varchar(20)	utf8mb4_general_ci		No	None
3	E_ADDRESS	varchar(20)	utf8mb4_general_ci		Yes	NULL
4	D_ID	int(11)			Yes	NULL

3. Add a column joinData (type INT) to the employee table:

SQL Queries:

```
ALTER TABLE employee ADD joinData INT;
```

Output

#	Name	Type	Collation	Attributes	Null	Default
1	E_ID	int(11)			No	None
2	E_NAME	varchar(20)	utf8mb4_general_ci		No	None
3	E_ADDRESS	varchar(20)	utf8mb4_general_ci		Yes	NULL
4	D_ID	int(11)			Yes	NULL
5	joinData	int(11)			Yes	NULL

4. Modify the column joinData to VARCHAR(20):

SQL Queries:

```
ALTER TABLE employee MODIFY joinData VARCHAR(20);
```

⚠ Note: In **MySQL**, MODIFY works. In **SQL Server**, use ALTER COLUMN.

#	Name	Type	Collation	Attributes	Null	Default
1	E_ID	int(11)			No	None
2	E_NAME	varchar(20)	utf8mb4_general_ci		No	None
3	E_ADDRESS	varchar(20)	utf8mb4_general_ci		Yes	NULL
4	D_ID	int(11)			Yes	NULL
5	joinData	varchar(20)	utf8mb4_general_ci		Yes	NULL

5. Create a table called department with the following structure:

Column Name	Data Type
D_ID	INT (PK)
D_NAME	VARCHAR(20)

SQL Queries:

```
CREATE TABLE dep (
    D_ID INT NOT NULL PRIMARY KEY,
    D_NAME VARCHAR(20)
);
```

Output

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	D_ID	int(11)			No	*None*
☐	2	D_NAME	varchar(20)	utf8mb4_general_ci		Yes	*NULL*

6. Add a foreign key constraint to D_ID in employee table:

SQL Queries:

```
ALTER TABLE employee
ADD CONSTRAINT FKEY
FOREIGN KEY (D_ID)
REFERENCES dep(D_ID);
```

Output

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	E_ID	int(11)			No	*None*
☐	2	E_NAME	varchar(20)	utf8mb4_general_ci		No	*None*
☐	3	E_ADDRESS	varchar(20)	utf8mb4_general_ci		Yes	*NULL*
☐	4	D_ID	int(11)			Yes	*NULL*
☐	5	joinData	varchar(20)	utf8mb4_general_ci		Yes	*NULL*

7. Drop the column joinData from the employee table:

SQL Queries:

```
ALTER TABLE employee DROP COLUMN joinData;
```

Output

	#	Name	Type	Collation	Attributes	Null	Default	C
☐	1	E_ID	int(11)			No	*None*	
☐	2	E_NAME	varchar(20)	utf8mb4_general_ci		No	*None*	
☐	3	E_ADDRESS	varchar(20)	utf8mb4_general_ci		Yes	*NULL*	
☐	4	D_ID	int(11)			Yes	*NULL*	

8. Truncate the employee table and drop the department table:

Insertion

INSERT INTO dep (D_ID, D_NAME)

VALUES (101, 'HR'),

(102, 'Finance');

INSERT INTO employee (E_ID, E_NAME, E_ADDRESS, D_ID)

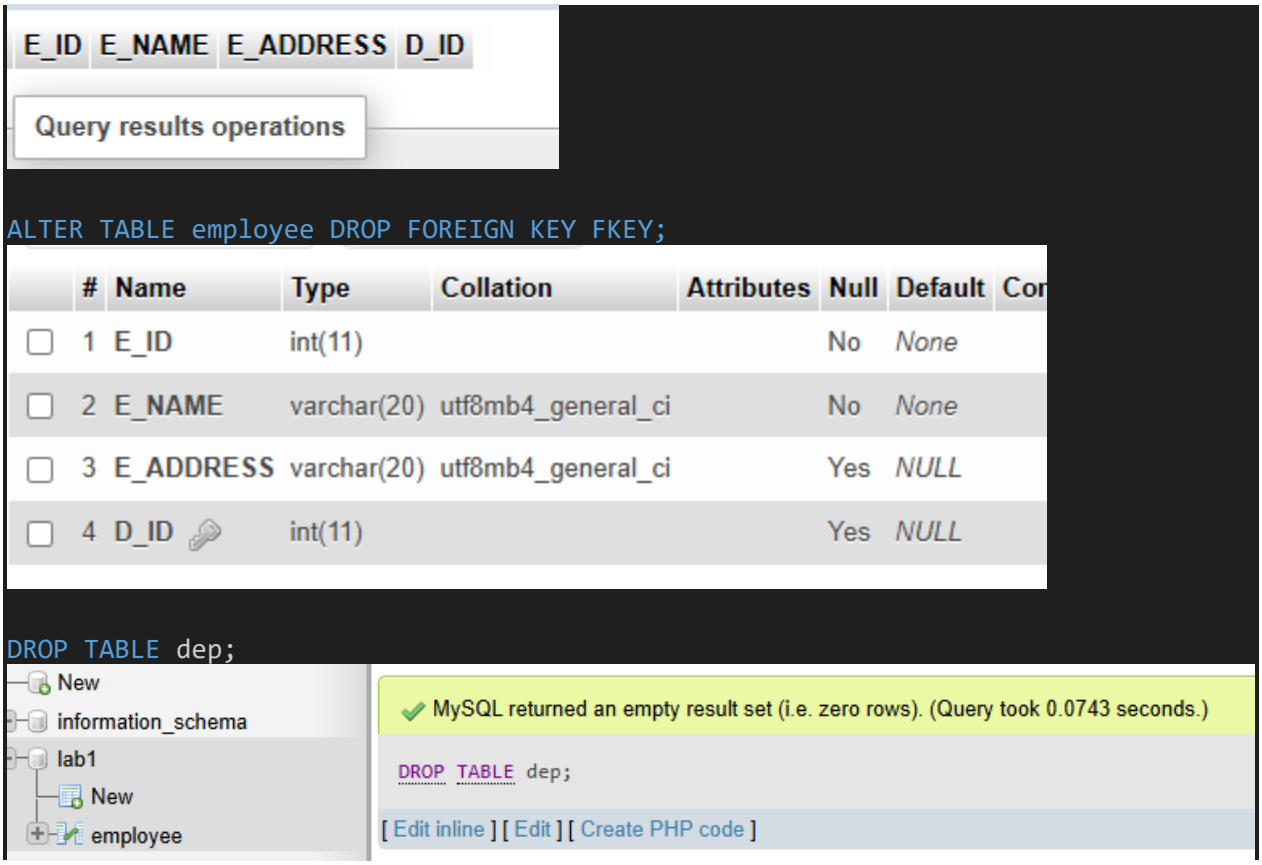
VALUES (1, 'Salim', 'Kathmandu', 101),

(2, 'Santosh', 'Surkhet', 102);

E_ID	E_NAME	E_ADDRESS	D_ID
1	Salim	Kathmandu	101
2	Santosh	Surkhet	102

SQL Queries:

```
TRUNCATE TABLE employee;
```



✅ **Note:**
Table and column names are case-insensitive in most SQL systems.
Ensure the table dep exists before adding a foreign key to avoid errors.
Would you like this as a PDF or Word document as well?

Appendix

```
DROP DATABASE lab1;
```