https://drive.google.com/drive/folders/1W4gzaImfw_sh5kaB7XqiXsYDVUGrTwTE
lab manual refrence^ ignore this

aim To run Simple SQL Queries

DATA MANIPULATION LANGUAGE (DML): The Data Manipulation Language (DML) is used to retrieve, insert, and modify database information. These commands are used by all database users during routine database operations. Here are the basic DML commands:

1. **INSERT**
2. **UPDATE**
3. **DELETE**

1. INSERT INTO: This is used to add records into a table. There are several types of INSERT INTO queries:

a) **Inserting a single record**:

Syntax:

```
INSERT INTO <relation/table name> (field_1, field_2, ..., field_n) VALUES (data_1, data_2, ..., data
```

Example:

```
INSERT INTO student (sno, sname, class, address) VALUES (1, 'Ravi', 'M.Tech', 'Palakol');
```

b) **Inserting a single record without column names**:

Syntax:

```
INSERT INTO <relation/table name> VALUES (data_1, data_2, ..., data_n);
```

Example:

```
INSERT INTO student VALUES (1, 'Ravi', 'M.Tech', 'Palakol');
```

c) **Inserting all records from another table**:

Syntax:

```
INSERT INTO relation_name_1 SELECT field_1, field_2, ..., field_n FROM relation_name_2 WHER
```

Example:

```
INSERT INTO std SELECT sno, sname FROM student WHERE name = 'Ramu';
```

d) **Inserting multiple records**:

Syntax:

```
INSERT INTO relation_name (field_1, field_2, ..., field_n) VALUES (&data_1, &data_2, ..., &data_n)
```

Example:

```
INSERT INTO student (sno, sname, class, address) VALUES (&sno, '&sname', '&class', '&address');
```

Prompt Example:

```
Enter value for sno: 101
Enter value for name: Ravi
Enter value for class: M.Tech
Enter value for address: Palakol
```

2. UPDATE-SET-WHERE: This is used to update the content of a record in a relation.

Syntax:

```
UPDATE relation_name SET Field_name1 = data, field_name2 = data WHERE field_name = data;
```

Example:

```
UPDATE student SET sname = 'Kumar' WHERE sno = 1;
```

3. DELETE-FROM: This is used to delete all the records of a relation but it will retain the structure of that relation.

a) **DELETE-FROM**: This is used to delete all the records of a relation.

Syntax:

```
DELETE FROM relation_name;
```

Example:

```
DELETE FROM std;
```

b) **DELETE-FROM-WHERE**: This is used to delete a selected record from a relation.

Syntax:

```
DELETE FROM relation_name WHERE condition;
```

Example:

```
DELETE FROM student WHERE sno = 2;
```

TRUNCATE: This command will remove the data permanently. But structure will not be removed.

Difference between TRUNCATE and DELETE:

- TRUNCATE removes data permanently, while DELETE removes data temporarily and can be rolled back.
- DELETE allows for conditional removal, whereas TRUNCATE does not.

- TRUNCATE is a DDL command, DELETE is a DML command.

Syntax:

```
TRUNCATE TABLE <Table_name>;
```

Example:

```
TRUNCATE TABLE student;
```

1. SELECT FROM: To display all fields for all records.

Syntax:

```
SELECT * FROM relation_name;
```

Example:

```
SELECT * FROM dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

2. SELECT FROM: To display a set of fields for all records of a relation.

Syntax:

```
SELECT a set of fields FROM relation_name;
```

Example:

```
SELECT deptno, dname FROM dept;
```

DEPTNO	DNAME
10	ACCOUNTING
20	RESEARCH
30	SALES

3. SELECT - FROM - WHERE: To display a selected set of fields for a selected set of records.

Syntax:

```
SELECT a set of fields FROM relation_name WHERE condition;
```

Example:

```
SELECT * FROM dept WHERE deptno <= 20;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS

Conclusion: hence ,we successfully performed running of simple sql queries