```
/*
DML stands for Data Manipulation Language and it is a type of SQL command used to
manipulate and query data in a database.
There are four primary DML commands in SQL:

    1. SELECT: This command is used to retrieve data from one or more tables in a
database. It is often used to display data on a website or application.
    2. INSERT: This command is used to add new data to a table in a database. It is often
used to add new records to a database.
    3. UPDATE: This command is used to modify existing data in a table. It is often used
to update records that have changed.
    4. DELETE: This command is used to remove data from a table in a database. It is often
used to delete records that are no longer needed.

It is important to note that DML commands can have a significant impact on a database, so
they should be used with care and \
only by \authorized personnel who understand the consequences of their actions.
*/

CREATE TABLE Emp (
                                eno int,
                ename VARCHAR(30),
                dno int ,
                primary key(eno));

/*
mysql> desc Emp;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| eno   | int         | NO   | PRI | NULL    |       |
| ename | varchar(30) | YES  |     | NULL    |       |
| dno   | int         | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
*/

# 1. INSERT
#=============================================================================================================
/*
        There are 5 methods to insert data into a table

        1. BASIC INSERT
        2. SEQUENTIAL INSERT
        3. MULTIPLE INSERT
        4. IMPORT FROM OTHER TABLE
        5. BULK IMPORT : Import froom .csv or excel files
*/

/*
    1. Basic Insert - both attribute list and value list are specified
                Synatx : INSERT INTO table_name (column1, column2, column3, ...)
                                                                VALUES (value1, value2,
value3, ...);
*/
INSERT INTO Emp (eno, ename, dno) VALUES(1, 'Alice', 100);
INSERT INTO Emp (eno, ename, dno) VALUES(2, 'Bob', 100);
INSERT INTO Emp (eno, ename) VALUES(20, 'Joice');

/*
    2. Sequential Insert : Only attribute values listed
                Synatx : INSERT INTO table_name VALUES (value1, value2, value3, ...);
*/
INSERT INTO Emp VALUES(3, 'Cindy', 100);
INSERT INTO Emp VALUES(4, 'Sam', 100);
```

```
/*
    3. Multiple Insert
                Synatx : INSERT INTO Table_name (column1, column2, column3, ...) VALUES
                                                        (value1, value2, value3, ...),
                                                        (value1, value2, value3, ...),
                                                        ...
                                                        (value1, value2, value3, ...),
*/
INSERT INTO Emp (eno, ename, dno) VALUES
                        (5, 'Alex', 100),
                        (6, 'John', 100);

/*
    4. Import from other tables
                Synatx : INSERT INTO table_name (column1, column2, column3, ...)
                                                                                SELECT
column1, column2, column3, ...
                                                                                FROM
source_table_name
                                                                                WHERE
conditions;
*/

/*
We have one Employee table as below;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| e_no      | int          | YES  |     | NULL    |       |
| e_fname   | varchar(30)  | YES  |     | NULL    |       |
| e_mname   | varchar(1)   | YES  |     | NULL    |       |
| e_lname   | varchar(30)  | YES  |     | NULL    |       |
| e_address | varchar(100) | YES  |     | NULL    |       |
| e_sex     | char(3)      | YES  |     | NULL    |       |
| e_salary  | int          | YES  |     | NULL    |       |
| e_dno     | int          | YES  |     | NULL    |       |
| e_s_ssn   | int          | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+

        Our aim is to populate data from Employee into Emp (eno int, ename varchar(30),
dno int) from Employee.
*/
INSERT INTO Emp (eno, ename, dno)
        SELECT e_no, e_fname, e_dno
        FROM Employee;


/*
    5. BULK IMPORT: Import from .csv or excel

*/
create table Student(
        stud_id INT AUTO_INCREMENT PRIMARY KEY PRIMARY KEY,
        stud_fname VARCHAR(20),
        stud_lname VARCHAR(20),
        stud_email VARCHAR(20),
        stud_ph VARCHAR(10));

LOAD DATA INFILE '/var/lib/mysql-files/student.csv' INTO TABLE Student FIELDS TERMINATED
BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;


# 2. UPDATE
#=============================================================================================================
/*
The UPDATE command in SQL is used to modify existing records in a table. This command
```

```
allows you to
update one or more columns of a specific record or a group of records that meet a certain
condition.

Syntax:
                UPDATE table_name
                        SET column1 = value1, column2 = value2, ...
                        WHERE condition;

*/
UPDATE Emp SET dno = 101 WHERE eno = 1;
UPDATE Emp SET dno = 101 WHERE ename = 'Bob';

# 3. DELETE
#=============================================================================================================
/*
        The DELETE command in SQL is used to delete one or more rows from a table in a
relational database.
    Syntax:
                        DELETE FROM table_name
                                        WHERE conditions;
*/

# delete a single row where eno = 1
DELETE FROM Emp where eno = 1;

# delete multiple row
DELETE FROM Emp where dno = 100;

#delete entire rows
DELETE FROM Emp;

#=============================================================================================================
#-----------------
QUESTIONS-------------------------------------------------------------------------

# 1. what is the difference in DELETE and DROP
/*
        ANS:
                DROP delete the table from database, where DELETE removed rows from table.
*/

# 4. SELECT
#=============================================================================================================

/*
The SELECT command is one of the most fundamental commands in SQL (Structured Query
Language).
It is used to retrieve data from one or more tables in a database. The SELECT statement
can be used to retrieve specific columns of data, filter data based on conditions,
sort the data, and perform various other operations.

Syntax: SELECT column1, column2, ...
                                FROM table_name
                                WHERE condition(s)
                                ORDER BY column_name(s) ASC|DESC;

        SELECT specifies the columns to be retrieved from the table(s).
    FROM specifies the table(s) from which to retrieve the data.
    WHERE specifies the conditions that must be met for a row to be included in the result
set.
    ORDER BY specifies the column(s) by which the result set should be sorted, in
ascending or descending order.
*/

# Quesries
```

```
#1 . Retrieve the names of all employees:
                SELECT ename FROM Emp;
#2.  Retrieve the employee numbers and names of all employees in department 3:
                SELECT eno, ename
                FROM Emp
                WHERE dno = 3;
#3 Retrieve the employee numbers and names of all employees sorted by department number in
ascending order:
                SELECT eno, ename
                FROM Emp
                ORDER BY dno ASC;

#4.Retrieve the employee numbers and names of all employees whose name contains the string
"Smith":
                SELECT eno, ename
                FROM Emp
                WHERE ename LIKE '%Smith%';
#5. retrieve entire table data
                SELECT * FROM Emp;
```