| Ex.No : 1 | CREATION OF BASE TABLE AND DML OPERATIONS |
|---|---|
| Date: 26/07/2024 | |

AIM:

**ALGORITHM:**

**STEP-1:** Start.

**STEP-2:** Create a base Table

Syntax:

CREATE TABLE <table name> (column1 type, column2 type, ...);

**STEP-3:** Describe the Table structure

Syntax:

DESC <table name>

**STEP-4:** Add a new row to a Table using INSERT statement.

Syntax:

- INSERT INTO <table name> VALUES (value1, value2..);
- INSERT INTO <table name> (column1, column2..)
  VALUES (value1, value2..);
- INSERT INTO <table name>VALUES (&column1,'&column');

**STEP-5:** Modify the existing rows in the base Table with UPDATE statement.

Syntax:

UPDATE <table name> SET column1=value, column2 = `value`
WHERE (condition);

**STEP-6:** Remove the existing rows from the Table using DELETE statement.

Syntax:

DELETE FROM <table name> WHERE <condition>;

**STEP-7:** Perform a Query using SELECT statement.

Syntax:

SELECT [DISTINCT] {*,<column1,...>} FROM <table name>
WHERE <condition>;

**STEP-8:** The truncate command deletes all rows from the table. Only the structure of the table remains.

Syntax:

TRUNCATE TABLE <table name>;

**STEP-9:** Alter the existing table using ALTER statement.

Syntax:

Add Column:

ALTER TABLE <table name> ADD (column data type [DEFAULTexpr][,column data type]);

Modify Column:

ALTER TABLE <table name> MODIFY (column data type [DEFAULT expr], [,column data type]);

Drop Column:

ALTER TABLE <table name> DROP COLUMN <column name>;

**STEP-10:** To drop the entire table using DROP statement.

Syntax:

DROP TABLE <table name>;

**STEP-11:** Exit.

1. Create MY_EMPLOYEE table with the following structure

| NAME | NULL? | TYPE |
|---|---|---|
| ID | Not null | Number(4) |
| Last_name | | Varchar(25) |
| First_name | | Varchar(25) |
| Userid | | Varchar(25) |
| Salary | | Number(9,2) |

```
CREATE TABLE MY-EMPLOYEE ( ID number (4) NOT NULL, Last_name varchar (25),
First_name varchar (25), Userid varchar (25), Salary number (9,2) );
```

2. Add the first and second rows data to MY_EMPLOYEE table from the following sample data.

| ID | Last_name | First_name | Userid | salary |
|---|---|---|---|---|
| 1 | Patel | Ralph | rpatel | 895 |
| 2 | Dancs | Betty | bdancs | 860 |
| 3 | Biri | Ben | bbiri | 1100 |
| 4 | Newman | Chad | Cnewman | 750 |
| 5 | Ropebur | Audrey | aropebur | 1550 |

```
INSERT INTO my-employee VALUES (1, 'Patel', 'Ralph', 'rpatel', 895);

INSERT INTO my-employee VALUES (2, 'Dancs', 'Betty', 'bdancs', 860);
```

3. Display the table with values.

```
SELECT * FROM MY_EMPLOYEE ;
```

4. Populate the next two rows of data from the sample data. Concatenate the first letter of the first_name with the first seven characters of the last_name to produce Userid.

```
INSERT INTO MY_EMPLOYEE VALUES (3, 'Biri', 'Ben', 'bbiri', 1100);

INSERT INTO MY-EMPLOYEE VALUES (4, 'Newman', 'chad', 'cnewman', 750);
```

5. Delete Betty dancs from MY _EMPLOYEE table.

```
DELETE FROM MY-EMPLOYEE WHERE ID = 2;
```

6. Empty the fourth row of the emp table.

```
DELETE FROM MY_EMPLOYEE WHERE ID=4;
```

7. Make the data additions permanent.

```
COMMIT;
```

8. Change the last name of employee 3 to Drexler.

```
UPDATE MY_EMPLOYEE SET Last_name='Drexler' WHERE ID = 3;
```

9. Change the salary to 1000 for all the employees with a salary less than 900.

```
UPDATE MY_EMPLOYEE SET salary= 1000 WHERE salary < 900;
```

| Evaluation Procedure | Marks awarded |
|---|---|
| Query(5) | 5 |
| Execution (5) | 5 |
| Viva(5) | 5 |
| Total (15) | 15 |
| Faculty Signature | |