Ajanta
Page No. :
Date : 25/3/23

EXPERIMENT - 2

AIM : Implementation of DDL and DML commands in SQL.

THEORY :

DDL stands for Data Definition Language. DML stands for Data Manipulation Language. These two categories of commands are used to create, modify and manipulate the structure and data of relational databases.

⇒ DDL commands used here are :

1) CREATE : It is used to create new table in the database.

Syntax:
```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ....
);
```

The column parameters specify the names of the columns of the table.

Teacher's Signature :

The datatype parameter specifies the type of data the column can hold (e.g varchar, integer, date, etc).

Example:

CREATE TABLE Employee (Name varchar(20), Email varchar2(100));

2) DROP: It is used to delete both the structure and record stored in the table.

Syntax: DROP TABLE table_name;

Example: DROP TABLE Employee;

⇒ DML commands used here are:

1) DELETE: DELETE FROM is a DML command that is used to delete one or more rows from a table.
The syntax is as follows:

DELETE FROM table-name
WHERE condition;

EXAMPLE: DELETE FROM Employee
WHERE Name = "Arun";

Teacher's Signature :

2) INSERT : The INSERT statement is a SQL query. It is used to insert data into the row of a table.

Syntax: INSERT INTO TABLE_NAME
(col 1, col 2, col 3 ... col N)
VALUES (value 1, value 2, value 3 ... value N);

Example:
INSERT INTO Employee
VALUES ("Arun", "Aurangabad");

3) UPDATE : This command is used to update or modify the value of a column in the table.

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

Example: UPDATE Employee
SET city = 'Nagpur'
WHERE Name = 'Arun';

⇒ The EXISTS keyword is used to test for the existance of a subquery. A subquery is usually a select statement that returns a set of rows and the exists keyword is used to check if any rows are returned by the subquery.

Teacher's Signature :

Syntax of EXIST :

```
SELECT column_name(s);
FROM table_name
WHERE EXISTS (subquery);
```

It is usually used with the NOT keyword. For example:

```
SELECT * FROM Employee
WHERE NOT EXISTS (
    SELECT * FROM Company
WHERE Company.city = city);
```

→ This query returns all the companies that do not have any associated city.

Ajanta
Page No. :
Date :

⇒ QUERIES : (to create initial tables)

```
DROP TABLE IF EXISTS Employee;
CREATE TABLE Employee (ename varchar2(10), city varchar2(10));
INSERT INTO Employee VALUES ("Sunil","Madras"),("Vijay","Madras");
SELECT * FROM Employee;

DROP TABLE IF EXISTS Emp_Company;
CREATE TABLE Emp_Company (ename varchar2(10), cname varchar2(10),
  Salary number(7,2), jdate date);
INSERT INTO Emp_Company VALUES('Sunil','ACC',50000,'01-SEP-2027'),
('Vijay','ACC',60000,'01-SEP-2027'),('Amar','Microsft',60000,
'01-SEP-2027'),('Atharva','Google',60000,'01-SEP-2027');
SELECT * FROM Emp_Company;

DROP TABLE IF EXISTS Company;
CREATE TABLE Company (cname varchar2(10), city varchar2(10));
INSERT INTO Company VALUES ('ACC','Madras'),('TATA','Bengaluru'),
('Microsft','Pune'),('Google','Bengalura');
SELECT * FROM Company;

DROP TABLE IF EXISTS Manager;
CREATE TABLE Manager (ename varchar2(10),mname varchar2(10));
INSERT INTO Manager VALUES('Sunil','Sharrari'),('Vijay','Sunil'),
('Amar','Mahi'),('Atharva','Rucha');
SELECT * FROM Manager;

DROP TABLE IF EXISTS Emp_Shift;
CREATE TABLE Emp_Shift (ename varchar2(10), shift varchar2(10));
```

Teacher's Signature :

```
INSERT INTO Emp_Shift VALUES ('Sunil', 'A'), ('Vijay', 'A'),
('Amar', 'B'), ('Atharva', 'C');
SELECT * FROM Emp_Shift;
```

DATABASES :

Employee

ename	city
Sunil	Madras
Vijay	Madras
Amar	Pune
Atharva	Bengaluru

Company

cname	city
ACC	Madras
TATA	Bengaluru
Microsoft	Pune
Google	Bengaluru

Emp_Company

ename	cname	salary	jdate
Sunil	ACC	5000	01-SEP-2027
Vijay	ACC	40000	01-SEP-2027
Amar	Microsoft	6000	01-SEP-2027
Atharva	Google	60000	01-SEP-2027

Manager

ename	mname
Sunil	Sharvari
Vijay	Sunil
Amar	Mahi
Atharva	Rucha

Emp_Shift

ename	shift
Sunil	A
Vijay	A
Amar	B
Atharva	C

QUERIES :

1) Decrease the salary of Vijay by 100 if Sunil and Vijay are living in city Madras.

➔ Before running the query :

ename	cname	salary	jdate
Sunil	ACC	5000	01-SEP-2027
Vijay	ACC	40000	01-SEP-2027
Amar	Microsoft	6000	01-SEP-2027
Atharva	Google	60000	01-SEP-2027

After running the following query :

```
UPDATE Emp_Company
SET salary = salary - 100
WHERE ename = 'Vijay' and
EXISTS(SELECT ename FROM Employee
WHERE ename = 'Vijay' AND city = 'Madras' AND city IN
(SELECT city FROM Employee
WHERE ename = 'Sunil'));
SELECT * FROM Emp_Company;
```

→

ename	cname	salary	jdate
Sunil	ACC	5000	01-SEP-2027
Vijay	ACC	39900	01-SEP-2027
Amar	Microsoft	6000	01-SEP-2027
Atharva	Google	60000	01-SEP-2027

2) All employees of ‘ACC’ having salary greater than 8000 are shifted to ‘TATA’.

→ Before running the query :

ename	cname	salary	jdate
Sunil	ACC	5000	01-SEP-2027
Vijay	ACC	39900	01-SEP-2027
Amar	Microsoft	6000	01-SEP-2027
Atharva	Google	60000	01-SEP-2027

After running the following query :

```
UPDATE Emp_Company
SET cname = 'TATA'
WHERE salary > 8000 and cname = 'ACC';
SELECT * FROM Emp_Company;
```

→

ename	cname	salary	jdate
Sunil	ACC	5000	01-SEP-2027
Vijay	TATA	39900	01-SEP-2027
Amar	Microsoft	6000	01-SEP-2027
Atharva	Google	60000	01-SEP-2027

3)Decrease the salary of employee Vijay By 100 and increase the salary of employee Sunil by 100.

→ Before running the query :

ename	cname	salary	jdate
Sunil	ACC	5000	01-SEP-2027
Vijay	TATA	39900	01-SEP-2027
Amar	Microsoft	6000	01-SEP-2027
Atharva	Google	60000	01-SEP-2027

After running the following query :

```
UPDATE Emp_Company
SET salary = salary - 100
WHERE ename = 'Vijay';
UPDATE Emp_Company
SET salary = salary + 100
WHERE ename = 'Sunil';
SELECT * FROM Emp_Company;
```

→

ename	cname	salary	jdate
Sunil	ACC	5100	01-SEP-2027
Vijay	TATA	39800	01-SEP-2027
Amar	Microsoft	6000	01-SEP-2027
Atharva	Google	60000	01-SEP-2027

4) Delete the rows of Emp_Company having salary Greater than 8000.

→ Before running the query :

ename	cname	salary	jdate
Sunil	ACC	5100	01-SEP-2027
Vijay	TATA	39800	01-SEP-2027
Amar	Microsoft	6000	01-SEP-2027
Atharva	Google	60000	01-SEP-2027

After running the following query :

```
DELETE FROM Emp_Company
WHERE salary > 8000;
SELECT * FROM Emp_Company;
```

→

ename	cname	salary	jdate
Sunil	ACC	5100	01-SEP-2027
Amar	Microsoft	6000	01-SEP-2027

CONCLUSION : DDL and DML Commands in SQL are being understood and implemented.