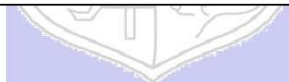




**Experiment No.: 04**

**Title:** To use DML operations and SQL queries to  
Populate the database



**Batch: B1****Roll No.:** 16010420133**Experiment No: 04**

**Aim:** To use DML operations and SQL queries to populate the database .

---

**Resources needed:** PostgreSQL PgAdmin4

---

**Theory:**

The Data Manipulation Language (DML) is used to populate the table with values, modify the table values and remove the rows of the table.

The DML statements are:

SELECT  
INSERT  
UPDATE  
DELETE

---

**Procedure:**

CREATE TABLE products (  
product\_no integer,  
name text,  
price  
numeric );

Let us consider the above products table



**Inserting rows:**

The INSERT command requires the table name and column values

INSERT INTO products VALUES (1, 'Cheese', 9.99);

If we don't have values for all the columns, you can omit some of them. In that case, the columns will be filled with their default values. For example:

INSERT INTO products (product\_no, name) VALUES (1, 'Cheese')

**Updating the values:**

The UPDATE command requires three pieces of information:

1. The name of the table and column to update
2. The new value of the column
3. Which row(s) to update

UPDATE products SET price = 10 WHERE price = 5;

UPDATE products SET price = price \* 1.10;

**Deleting rows:**

The syntax of the DELETE command is similar to the UPDATE command. DELETE FROM products WHERE price = 10;

**Retrieving values:**

The general syntax of the SELECT command is SELECT select\_list FROM table\_expression

SELECT \* FROM table1;

SELECT \* FROM products WHERE price=10;

SELECT product\_no, name FROM products WHERE price=10;

**Example:**

```
insert into department values('IT', 101, 'mumbai');
insert into department values('COMP', 102, 'mumbai');
insert into department values('ETRX', 103, 'delhi');
insert into department values('EXTC', 104, 'chennai');
insert into department values('account', 105, 'mumbai');
```

```
insert into employee values('anita','m','sharma','emp0001',20000,'mumbai',101);
insert into employee values('nita','g','patil','emp0004',10000,'mumbai',101);
insert into employee values('krupita','v','jetali','emp0003',20000,'delhi',103);
insert into employee values('juhi','r','verma','emp0002',15000,'delhi',104);
insert into employee values('anita','m','sharma','emp0005',20000,'mumbai',104);
```

```
insert into project values( 1, 'mumbai','website',101);
insert into project values( 2, 'chennai','coding',101);
insert into project values( 3, 'mumbai','testing',102);
insert into project values( 4, 'delhi','documentaion',103);
```

```
insert into works_on values(1,'emp0001', 12);
insert into works_on values(1,'emp0002', 10);
insert into works_on values(2,'emp0001', 6);
insert into works_on values(3,'emp0004', 2);
```

```
insert into dependent values('emp0001', 'sunita','sister');
insert into dependent values('emp0001', 'nita','mother');
insert into dependent values('emp0002', 'kamal','brother');
insert into dependent values('emp0004', 'krishna','father');
```

```
select * from employee;
select * from department;
select * from project;
select * from dependent;
select * from works_on;
```

**1) employee**

fname	mname	lname	ssn	salary	ecity	dno
anita	m	sharma	emp0001	20000	mumbai	101
juhi	r	verma	emp0002	15000	delhi	104

krupita	v	jetali	emp0003	20000	delhi	103
nita	g	patil	emp0004	10000	mumbai	101
anita	m	sharma	emp0005	20000	mumbai	104

2) department

dnamednodlocation

IT	101	mumbai
COMP	102	mumbai
ETRX	103	delhi
EXTC	104	chennai
account	105	mumbai

4) project

pnoplocationpnamedno

1	mumbai	website	101
2	chennai	coding	101
3	mumbai	testing	102
4	delhi	documentaion	103

5) dependents

ssndepname	relation
emp0001nita	mother
emp0001sunita	sister
emp0002kamal	brother
emp0004krishna	father

6) woks\_on

pnossnno\_of\_hrs

1	emp0001	12
1	emp0002	10
2	emp0001	6
3	emp0004	2

**Results: (Queries printout with output as per the format)**

1. Write 10 queries using 'from' and 'where' clause.

**1. Insert Row:**

```
CREATE TABLE employee (
ELoginID text PRIMARY KEY,
Name text UNIQUE,
Password text
);

INSERT into employee values('22', 'SOUMEN','abcdef');




INSERT into employee values('23', 'MEN','abcde');

INSERT into employee values('24', 'Deva','abcd');

INSERT into employee values('25', 'Dev','abc');

SELECT * from employee;
```

**Output:**

	 <b>elloginid</b> [PK] text	 <b>name</b> text	 <b>password</b> text
1	22	SOUMEN	abcdef
2	23	MEN	abcde
3	24	Deva	abcd
4	25	Dev	abc

## 2. Updating the values:

### A. Specific Value

**CREATE TABLE employee (**

**ELoginID text PRIMARY KEY,**

**Name text UNIQUE,**

**Password text**

**);**

**INSERT into employee values('22', 'SOUMEN','abcdef');**

**INSERT into employee values('23', 'MEN','abcde');**

**INSERT into employee values('24', 'Deva','abcd');**

**INSERT into employee values('25', 'Dev','abc');**

**UPDATE employee SET Name = 'SOUMEN SAMANTA' WHERE Name = 'SOUMEN';**

**SELECT \* from employee;Output:**

	Data Output	Explain	Messages	Notifications
	<div> <div>elloginid</div> <div>[PK] text</div> </div>	<div> <div>name</div> <div>text</div> </div>	<div> <div>password</div> <div>text</div> </div>	
1	23	MEN	abcde	
2	24	Deva	abcd	
3	25	Dev	abc	
4	22	SOUMEN ...	abcdef	

**B. Updating all values:**

```

CREATE TABLE employee (
ELoginID text PRIMARY KEY,
Name text UNIQUE,
Password text
);




INSERT into employee values('22', 'SOUMEN','abcdef');
INSERT into employee values('23', 'MEN','abcde');
INSERT into employee values('24', 'Deva','abcd');
INSERT into employee values('25', 'Dev','abc');

UPDATE employee SET Password = 'xyz';

SELECT * from employee;

```

**Output:**

	 <b>elloginid</b> [PK] text	 <b>name</b> text	 <b>password</b> text
1	22	SOUMEN	xyz
2	23	MEN	xyz
3	24	Deva	xyz
4	25	Dev	xyz

### 3. Deleting rows:

CREATE TABLE employee (

ELoginID text PRIMARY KEY,

Name text UNIQUE,

Password text

);

INSERT into employee values('22', 'SOUMEN','abcdef');

INSERT into employee values('23', 'MEN','abcde');





INSERT into employee values('24', 'Deva','abcd');

INSERT into employee values('25', 'Dev','abc');

DELETE FROM employee WHERE ELoginID = '24';

SELECT \* from employee;Output:



	Data Output	Explain	Messages	Notifications
	 <b>eloinid</b> [PK] text 	<b>name</b> text 	<b>password</b> text 	
1	22	SOUMEN	abcdef	
2	23	MEN	abcde	
3	25	Dev	abc	

#### 4. Retrieving values from table:

**CREATE TABLE employee (**

**ELoginID text PRIMARY KEY,**

**Name text UNIQUE,**

**Password text**

**);**





**INSERT into employee values('22', 'SOUMEN','abcdef');**

**INSERT into employee values('23', 'MEN','abcdef');**

**INSERT into employee values('24', 'Deva','abcd');**





**INSERT into employee values('25', 'Dev','abc');**

**SELECT \* from employee;Output:**

Data Output	Explain	Messages	Notifications
 <b>elloginid</b> [PK] text 	<b>name</b> text 	<b>password</b> text 	
1 22	SOUMEN	abcdef	
2 23	MEN	abcdef	
3 24	Deva	abcd	
4 25	Dev	abc	



SELECT \* FROM employee WHERE Password = 'abcdef';

**Output:**

Data Output	Explain	Messages	Notifications
 <b>elloginid</b> [PK] text 	<b>name</b> text 	<b>password</b> text 	
1 22	SOUMEN	abcdef	
2 23	MEN	abcdef	

SELECT ELoginID FROM employee WHERE Password = 'abcdef';

**Output:**

Data Output	Explain
 <b>elloginid</b> [PK] text 	
1 22	
2 23	

SELECT \* FROM employee WHERE ELoginID > '22';

**Output:**

	<b>elloginid</b> [PK] text	<b>name</b> text	<b>password</b> text
1	23	MEN	abcdef
2	24	Deva	abcd
3	25	Dev	abc

SELECT Name,Password FROM employee WHERE ELoginID > '22';

**Output:**

	<b>name</b> text	<b>password</b> text
1	MEN	abcdef
2	Deva	abcd
3	Dev	abc

SELECT Name,Password FROM employee WHERE ELoginID >= '22';

**Output:**

Data Output	Explain	Messages	Notifications
<div> <div></div> <div>name</div> <div>text</div> <div></div> </div>	<div> <div>password</div> <div>text</div> <div></div> </div>		
1	SOUMEN	abcdef	
2	MEN	abcdef	
3	Deva	abcd	
4	Dev	abc	

SELECT Name,Password FROM employee WHERE ELoginID < '24';

**Output:**

Data Output	Explain	Messages	Notifications
<div> <div></div> <div>name</div> <div>text</div> <div></div> </div>	<div> <div>password</div> <div>text</div> <div></div> </div>		
1	SOUMEN	abcdef	
2	MEN	abcdef	




SELECT Name,Password FROM employee WHERE ELoginID != '24';

**Output:**

Data Output	Explain	Messages	Notifications
<div> <div></div> <div>name</div> <div>text</div> <div></div> </div>	<div> <div>password</div> <div>text</div> <div></div> </div>		
1	SOUMEN	abcdef	
2	MEN	abcdef	
3	Dev	abc	




SELECT Name,Password FROM employee WHERE ELoginID BETWEEN '22' and '23';

**Output:**

Data Output	Explain	Messages	Notifications
 <b>name</b> text 	<b>password</b> text 		
1	SOUMEN	abcdef	
2	MEN	abcdef	

SELECT Name,Password FROM employee WHERE ELoginID = '22' OR ELoginID = '24';

**Output:**

Data Output	Explain	Messages	Notifications
 <b>name</b> text 	<b>password</b> text 		
1	SOUMEN	abcdef	
2	Deva	abcd	

SELECT Name,Password FROM employee WHERE ELoginID IN ('22','23','24');

**Output:**

Data Output	Explain	Messages	Notifications
	name text	password text	
1	SOUMEN	abcdef	
2	MEN	abcdef	
3	Deva	abcd	

**Example:****1) To extract the name and ssn of all the employees:**

Select fname, mname, lname, ssn from employee;

fname mname lname ssn

```

-----
anitasharmam          emp0001
juhiverma             r          emp0002
krupitajetali         v          emp0003
nitapatil             g          emp0004
anitasharma           m          emp0005
  
```

**2) To select names and city of the employees earning salary more than 10000:**

Select fname, mname, lname, ecity from the employee where salary>10000;

fname mname lname ecity

```

-----
anitasharmam          mumbai
juhivermar            delhi
krupitajetaliv        delhi
anitasharma m         mumbai
  
```

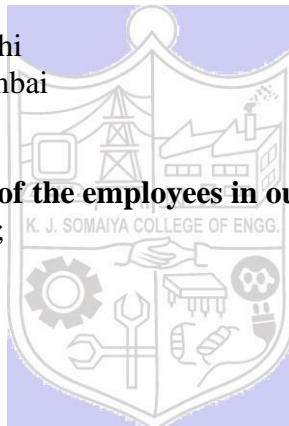
**3) TO get the details of the cities of the employees in our company:**

select distinct ecity from employee;

ecity

```

-----
delhi
mumbai
  
```



**4) To find the name of the department located in Mumbai and with department number 101:**

```
select dname from department where dlocation='Mumbai' and dno=101;
dname
-----
```

**5) To delete all dependent whose relation is mother with employee:**

```
delete form dependent where relation='mother';
```

```
ssndepname          relation
-----
```

```
emp0001sunita      sister
emp0002kamal       brother
emp0004krishna     father
```

**6) Update relation employee to increment salary of all employees working in Department 101 by Rs. 10000:**

```
update employee set salary=salary+10000 where dno=101;
```

```
fnamemnamelnamessn    salary    ecitydno
-----
```

```
anita      m      sharma      emp0001      30000      mumbai101
juhi       r      verma       emp0002      15000      delhi      104
krupita    v      jetali      emp0003      20000      delhi      103
nita       g      patil emp0004      20000      mumbai      101
anita      m      sharma      emp0005      20000      mumbai104
```

---

**Outcomes:** Illustrate the concept of security, Query processing, indexing and Normalization for Relational database

---

**Questions:****Q1 Explain various data types used in SQL**

- Numeric types consist of two-byte, four-byte, and eight-byte integers, four-byte and eight-byte floating-point numbers, and selectable-precision decimals. Eg) smallint, integer, bigint, decimal, numeric, etc
- Date and Time data types such as Date, Time, Datetime etc.
- Character and String data types such as char, varchar, text etc.

- Unicode character string data types, for example nchar, nvarchar, ntext etc.
- Binary data types such as binary, varbinary etc.
- Miscellaneous data types – clob, blob, xml, cursor, table etc.

## Q2 what is outer JOIN and why it is used? Explain its type with example

When performing an inner join, rows from either table that are unmatched in the other table are not returned. In an outer join, unmatched rows in one or both tables can be returned. The most common case for this is when you're matching the foreign key of one table to the primary key of another, such as when using an ID to look up a value.

```
SELECT P.FirstName,
       P.LastName,
       P.Title,
       PH.PhoneNumber
FROM   Person.Person AS P
       LEFT OUTER JOIN
       Person.PersonPhone AS PH
       ON P.BusinessEntityID = PH.BusinessEntityID
       AND PH.PhoneNumberTypeID = 3
ORDER BY P.LastName
```

When a match isn't found, then a NULL is placed in the column.

---

**Conclusion: :** We used DML operations and SQL queries to populate the database, wrote queries to update, delete etc.





**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of faculty in-charge with date**

---

**References:**

**Books:**

1. Elmasri and Navathe, “Fundamentals of Database Systems”, 6<sup>th</sup> Edition, Pearson Education
2. Korth, Slberchatz,Sudarshan, :”Database System Concepts”, 6th Edition, McGraw – Hill.

**WebSite:**

1. <http://www.tutorialspoint.com/postgresql/>
2. <http://sage.virtual-labs.ac.in/home/pub/21/>