Experiment No.: 04

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

Batch:A2 **Roll No.:16010421063** **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

```
fnamemnamelnamessn         salary      ecitydno
-------------------------------------- ----------------------------- ---------------------------------
anita            m         sharma       emp0001     20000       mumbai101
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     mumbai104
```

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          delhidocumentaion         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.

Example:

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

Select fname, mname, lname, ssn from employee;

```
fnamemnamelnamessn
-------------------------------------- -------------------------------------- ---------------------------
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 then 10000:**

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

```
fnamemnamelname             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:**

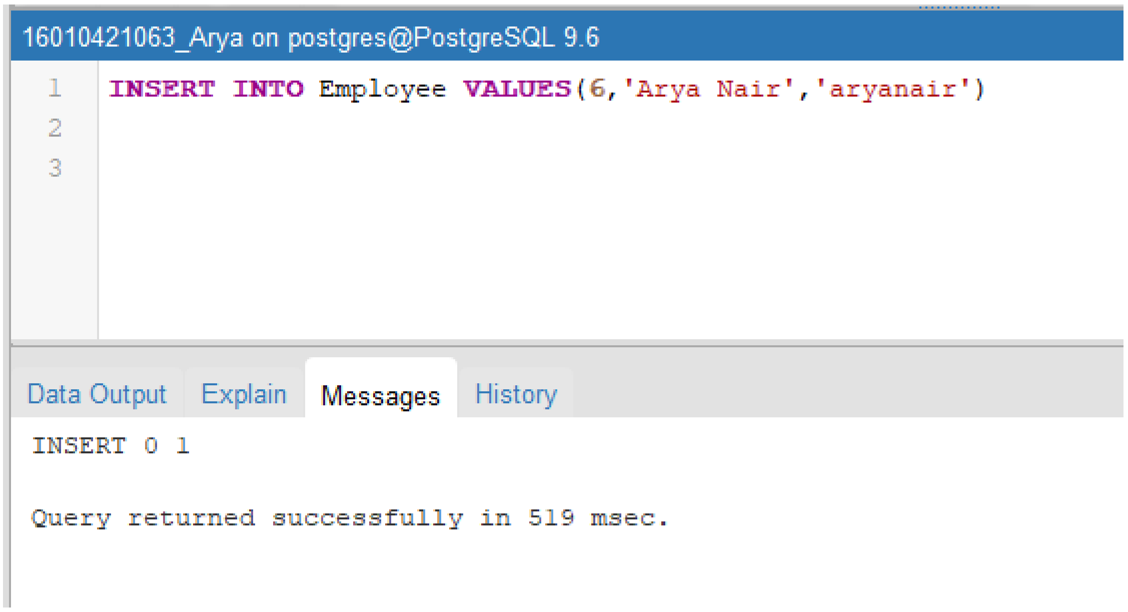
delete form dependent where relation='mother';

```
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;

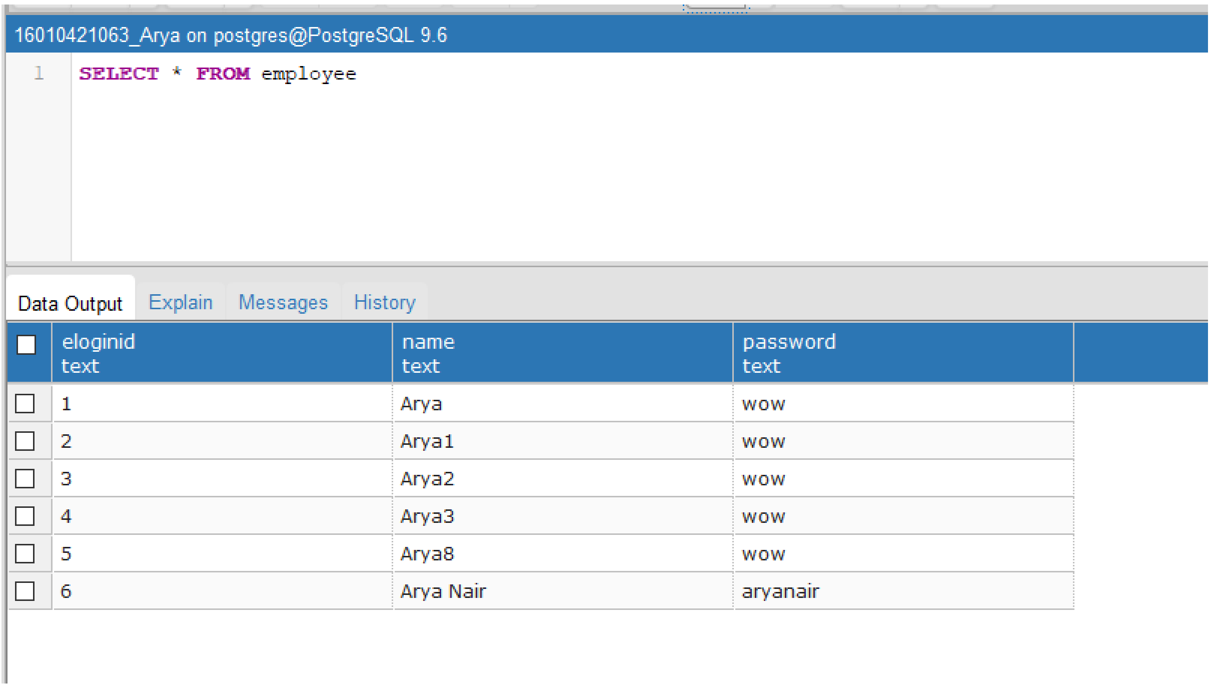
```
fnamemnamelnamessn         salary      ecitydno
---------------------------------------- ------------------------------ ---------------------------------
anita           m         sharma       emp0001      30000       mumbai101
juhi             r         verma         emp0002     15000       delhi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



```
INSERT INTO Employee VALUES(6,'Arya Nair','aryanair')
```

```
INSERT 0 1

Query returned successfully in 519 msec.
```

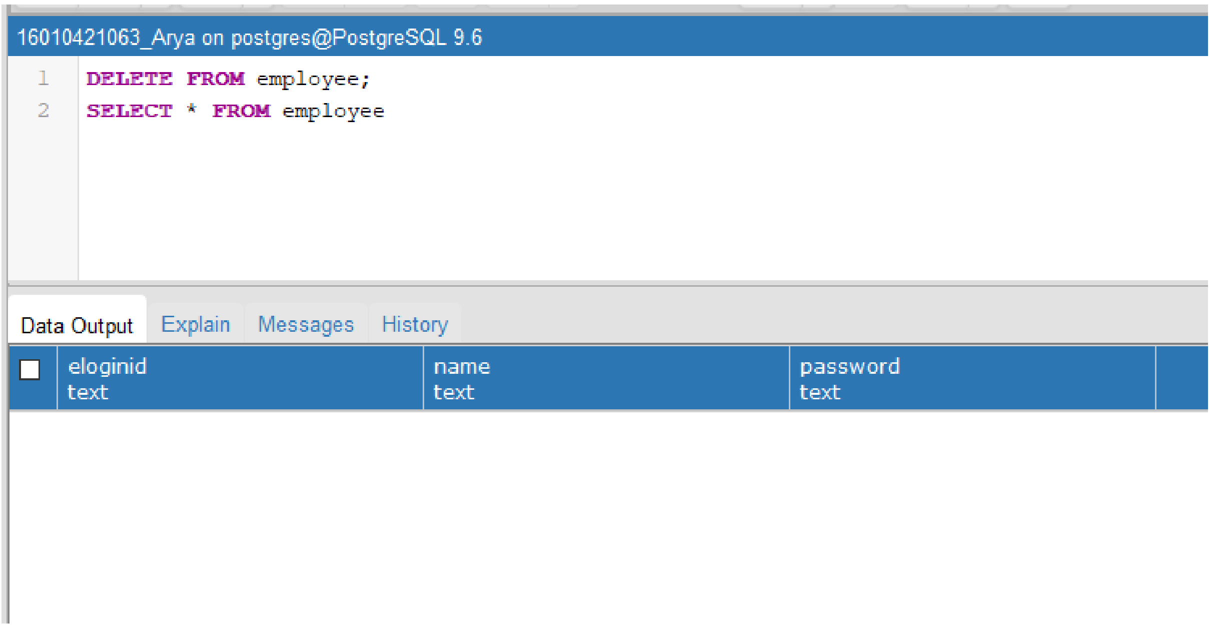


16010421063_Arya on postgres@PostgreSQL 9.6

```
SELECT * FROM employee
```

Data Output | Explain | Messages | History

eloginid text	name text	password text
1	Arya	wow
2	Arya1	wow
3	Arya2	wow
4	Arya3	wow
5	Arya8	wow
6	Arya Nair	aryanair

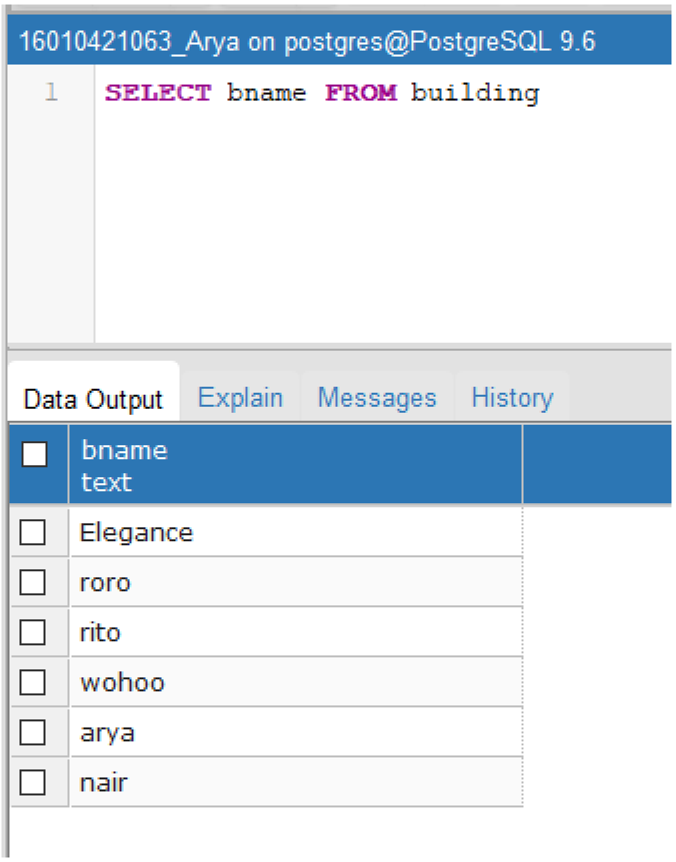


16010421063_Arya on postgres@PostgreSQL 9.6

```
DELETE FROM employee;
SELECT * FROM employee
```

Data Output | Explain | Messages | History

eloginid text	name text	password text

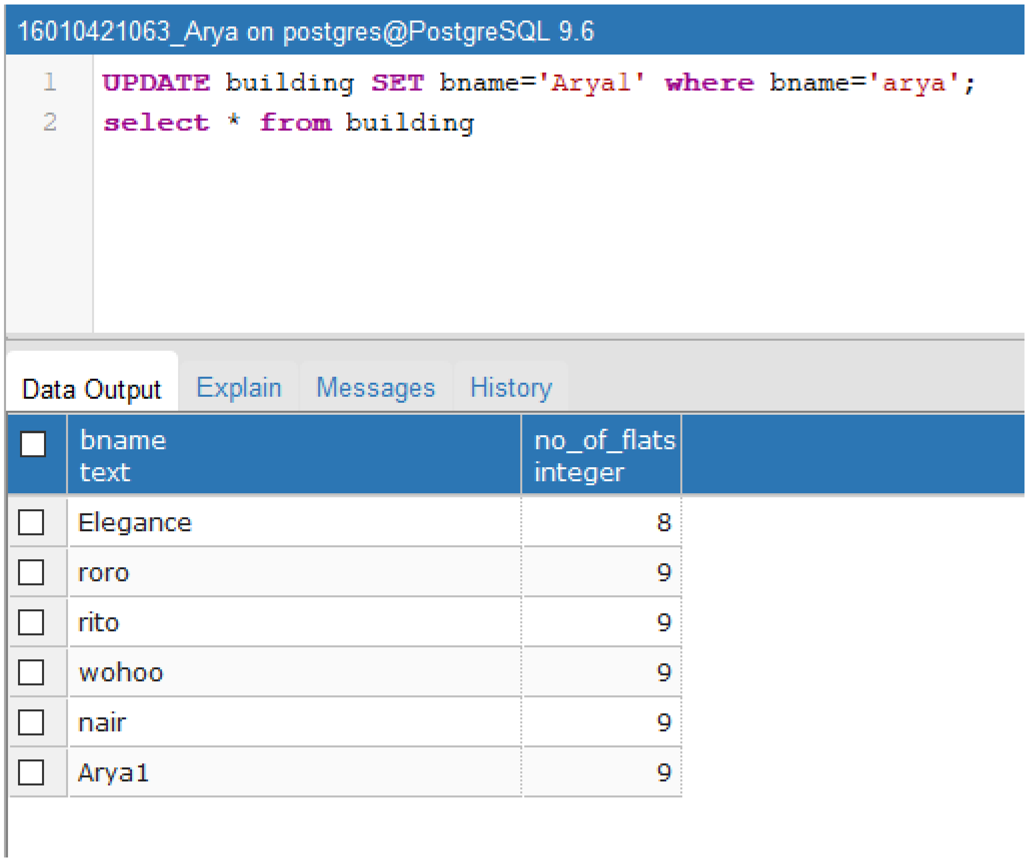


16010421063_Arya on postgres@PostgreSQL 9.6

```sql
SELECT bname FROM building
```

Data Output | Explain | Messages | History

bname text
Elegance
roro
rito
wohoo
arya
nair

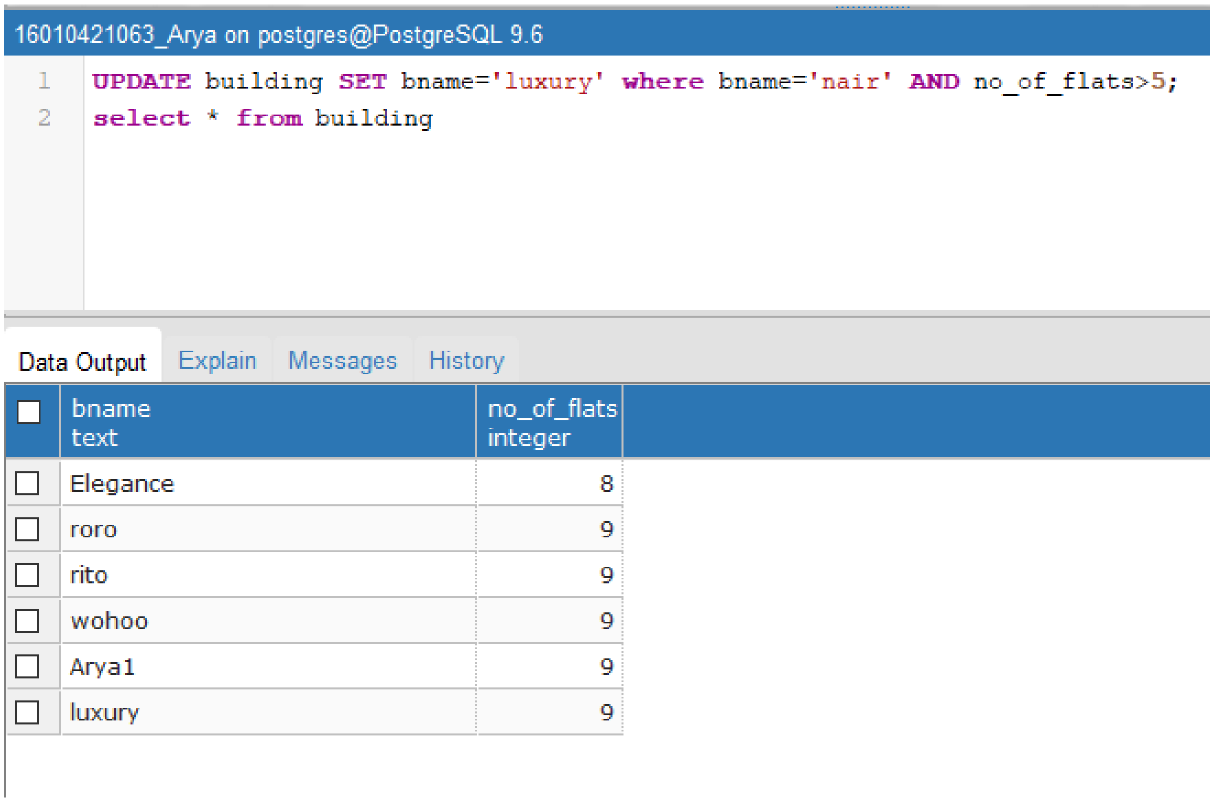


16010421063_Arya on postgres@PostgreSQL 9.6

```sql
UPDATE building SET bname='Arya1' where bname='arya';
select * from building
```

Data Output | Explain | Messages | History

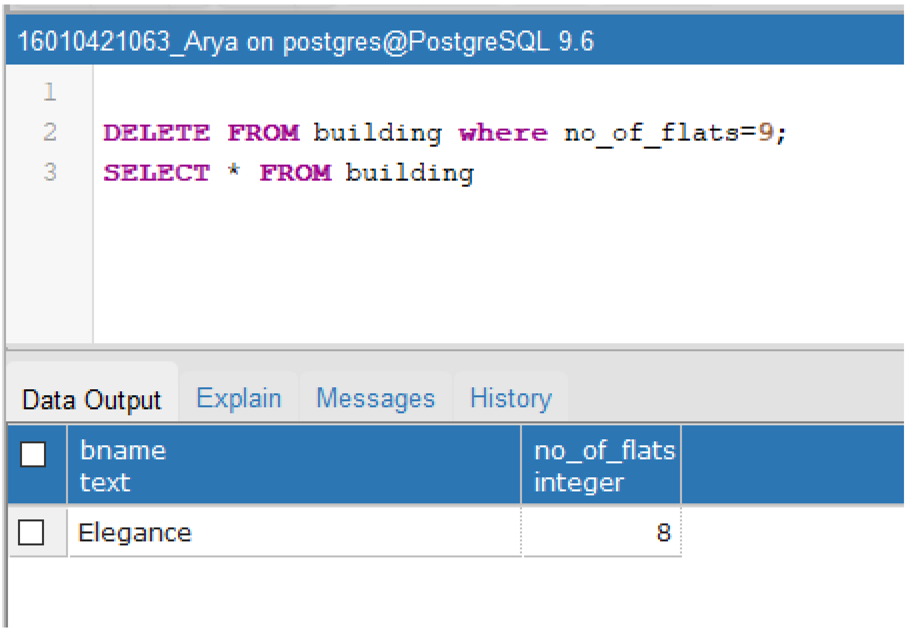
bname text	no_of_flats integer
Elegance	8
roro	9
rito	9
wohoo	9
nair	9
Arya1	9



16010421063_Arya on postgres@PostgreSQL 9.6

```
UPDATE building SET bname='luxury' where bname='nair' AND no_of_flats>5;
select * from building
```

bname text	no_of_flats integer
Elegance	8
roro	9
rito	9
wohoo	9
Arya1	9
luxury	9

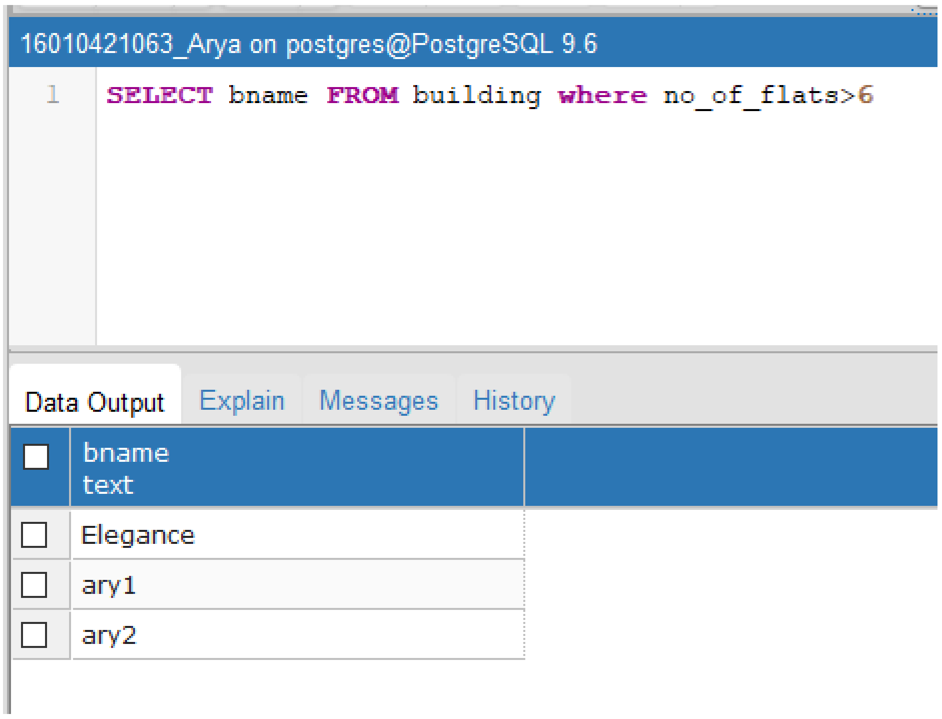


16010421063_Arya on postgres@PostgreSQL 9.6

```

DELETE FROM building where no_of_flats=9;
SELECT * FROM building
```

bname text	no_of_flats integer
Elegance	8



16010421063_Arya on postgres@PostgreSQL 9.6

```
SELECT bname FROM building where no_of_flats>6
```

Data Output | Explain | Messages | History

```
bname
text
Elegance
ary1
ary2
```

Outcomes:

CO2:Apply data models to real world scenario

Questions:

Q1 Explain various data types used in SQL

String:

CHAR(size) A FIXED length string (can contain letters, numbers, and special characters). The size parameter specifies the column length in characters - can be from 0 to 255. Default is 1

CHAR(size) A FIXED length string (can contain letters, numbers, and special characters). The size parameter specifies the column length in characters - can be from 0 to 255. Default is 1

VARCHAR(size) A VARIABLE length string (can contain letters, numbers, and special characters). The size parameter specifies the maximum column length in characters - can be from 0 to 65535

Numeric:

BIT(size) A bit-value type. The number of bits per value is specified in size. The size parameter can hold a value from 1 to 64. The default value for size is 1.

BOOL Zero is considered as false, nonzero values are considered as true.

INT(size) A medium integer. Signed range is from -2147483648 to 2147483647.

INTEGER(size) Equal to INT(size)

FLOAT(size, d) A floating point number. The total number of digits is specified in size. The number of digits after the decimal point is specified in the d parameter. This syntax is deprecated in MySQL 8.0.17, and it will be removed in future MySQL versions

FLOAT(p) A floating point number. MySQL uses the p value to determine whether to use FLOAT or DOUBLE for the resulting data type. If p is from 0 to 24, the data type becomes FLOAT(). If p is from 25 to 53, the data type becomes DOUBLE()

Date and Time Data Types

DATE A date. Format: YYYY-MM-DD. The supported range is from '1000-01-01' to '9999-12-31'

TIMESTAMP(fsp) A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC. Automatic initialization and updating to the current date and time can be specified using

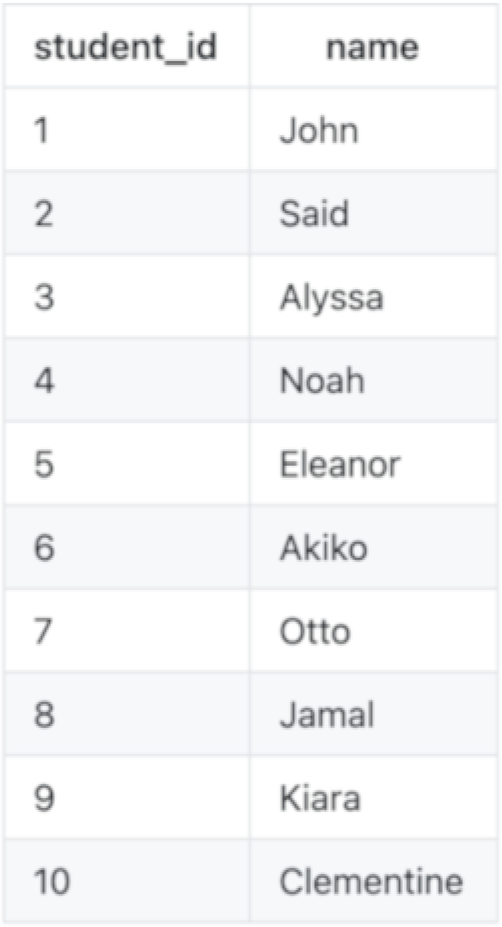
Q2 what is outer JOIN and why it is used? Explain its type with example

The FULL OUTER JOIN (aka OUTER JOIN) is used to return all of the records that have values in either the left or right table.For example, a full outer join of a table of customers and a table of orders might return all customers, including those without any orders, as well as all of the orders. Customers who have made orders would be united with their orders using their customer id number.

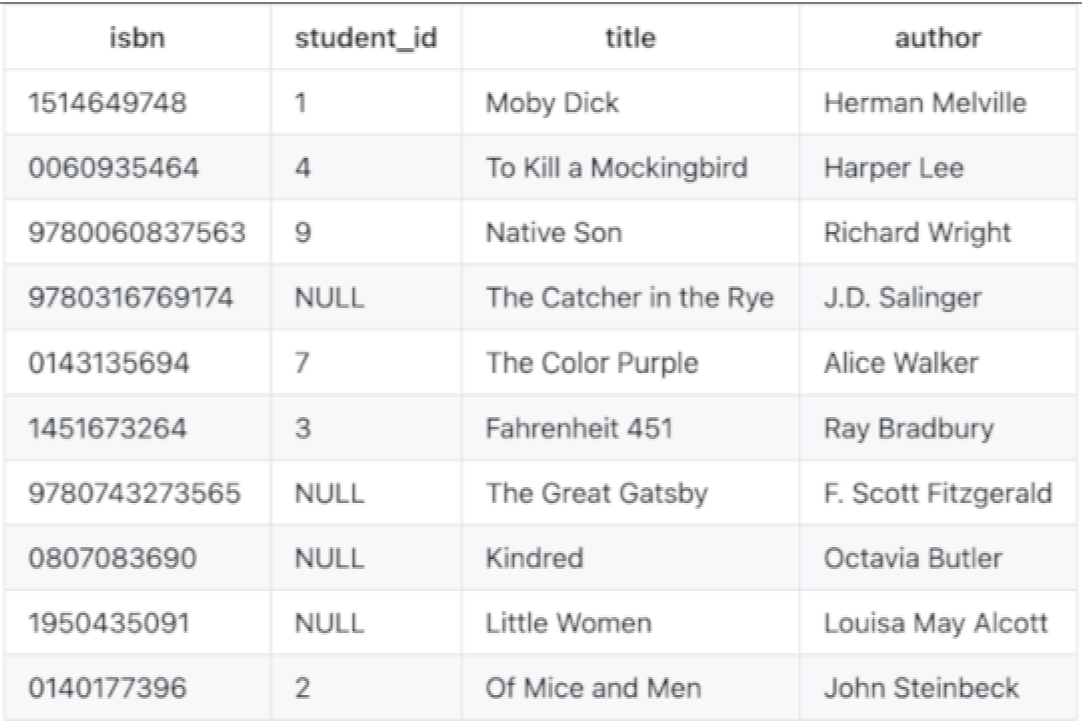
A full outer join can return a lot of data, so before you use it, consider whether a more conservative method might meet your needs.

Imagine that you are teaching an American Literature class. You have ten students, and you want each of them to read a different book from a list of pre-approved classic American novels. Some students have chosen the book they will read, while others have not done so yet.

You have created a table that lists the students along with their student ID numbers, and another table that lists books with their title, author, ISBN, and the ID of the student who will be reading the book, if someone has chosen



student_id	name
1	John
2	Said
3	Alyssa
4	Noah
5	Eleanor
6	Akiko
7	Otto
8	Jamal
9	Kiara
10	Clementine



isbn	student_id	title	author
1514649748	1	Moby Dick	Herman Melville
0060935464	4	To Kill a Mockingbird	Harper Lee
9780060837563	9	Native Son	Richard Wright
9780316769174	NULL	The Catcher in the Rye	J.D. Salinger
0143135694	7	The Color Purple	Alice Walker
1451673264	3	Fahrenheit 451	Ray Bradbury
9780743273565	NULL	The Great Gatsby	F. Scott Fitzgerald
0807083690	NULL	Kindred	Octavia Butler
1950435091	NULL	Little Women	Louisa May Alcott
0140177396	2	Of Mice and Men	John Steinbeck

In this example, we are selecting the names from the students table and the book titles from the books table. Records are matched using the student_id column in both tables.
With the full outer join, we are able to see all of the students, including those who have not chosen a book yet. We can also see all of the books, including those that have not yet been

chosen.

Conclusion: Used to INSERT,SELECT,DELETE and UPDATE to manipulate the data

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", 6th 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/