

## DBMS ASSIGNMENT - 3

### Topic : Vehicle Showroom Management System

#### Section : C

#### Team Details:

Gauthami S - PES1UG19CS166

Gitika Jain - PES1UG19CS169

Janya Mahesh - PES1UG19CS194

#### Part - 1 (queries)

#### Simple Queries:

1) Show the employees with their total number of sales(so)

```
>> select emp_id,count(*) from sales group by emp_id;
```

```
you are now connected to database vehicle_mgmt as user postgres
vehicle_mgmt=# select emp_id,count(*) from sales group by emp_id;
 emp_id | count
-----+-----
 60008.00 | 1
 60009.00 | 1
 600014.00 | 2
 60005.00 | 3
 60006.00 | 3
 60001.00 | 2
 600020.00 | 1
 60002.00 | 2
 60003.00 | 2
 600019.00 | 1
 60004.00 | 1
 600013.00 | 1
(12 rows)

vehicle_mgmt=#
```

2) Show the customer id whose payment is pending.(e)

>> select cust\_id from sales where status='PENDING' ;

```
vehicle_mgmt=# select cust_id from sales where status='PENDING';
cust_id
-----
10002.00
10005.00
10006.00
10009.00
10004.00
10004.00
10008.00
10006.00
10002.00
10004.00
(10 rows)
```

3) Show the details of the showroom where the location includes

"MG Roads".(a)

>> select s\_id,location,contact\_no from showroom where location like '%MG Road%';

```
vehicle_mgmt=# select s_id,location,contact_no from showroom where location like '%MG Road%';
 s_id | location | contact_no
-----+-----+-----
40006.00 | 3321 MG Road | 9662891236
(1 row)
```

4) Show the vehicle details where fuel type is petrol(e).

>> select mfg\_id,v\_name,s\_id from vehicle where FType like 'Petrol' order by s\_id;

```
vehicle_mgmt=# select s_id,v_name,mfg_id from vehicle where FType = 'Petrol' order by s_id;
 s_id | v_name | mfg_id
-----+-----+-----
40001.00 | Eco Sport | FD-ES-TTN-MN-P-1
40001.00 | Swift | SZ-SW-LXI-MN-P-9811
40001.00 | Kwid | RN-KW-KRS-MN-P-2
40001.00 | Swift Desire | SZ-SD-ZXI-MN-P-2
40002.00 | Swift Desire | SZ-SD-ZXI-MN-P-3
40002.00 | Alto 800 | SZ-A80-LXI-MN-P-8873
40002.00 | VERNA | HU-VN-CRT-AT-P-3
40003.00 | Ertiga | SZ-ER-VXI-MN-P-21
40004.00 | Alto 800 | SZ-A80-VXI-MN-P-8871
40004.00 | Swift | SZ-SW-LXI-MN-P-41
40004.00 | Nissan GTR | NM-GTR-TIO-MN-P-964
40005.00 | Alto 800 | SZ-A80-VXI-MN-P-21
40005.00 | Alto 800 | SZ-A80-LXI-MN-P-001
40005.00 | R8 | AD-R8-SPY-AT-P-94
40006.00 | Nissan GTR | NM-GTR-TIO-MN-P-844
40006.00 | Eco Sport | FD-ES-TTN-MN-P-593
40007.00 | Swift Desire | SZ-SD-ZXI-MN-P-4006
40007.00 | Kwid | RN-KW-KRS-MN-P-996
40007.00 | VERNA | HU-VN-CRT-AT-P-123
40008.00 | Ertiga | SZ-ER-VXI-MN-P-2901
(20 rows)
```

5) Show number of sales persons in each showroom(so).

>> select count(\*),s\_id from employee where designation = 'Sales Person' group by s\_id;

```
vehicle_mgmt=# select count(*),s_id from employee where designation = 'Sales Person' group by s_id;
 count | s_id
-----+-----
1 | 40001.00
3 | 40002.00
1 | 40003.00
3 | 40004.00
3 | 40005.00
1 | 40007.00
2 | 40008.00
(7 rows)
```

6)

>>select sale\_id from insurance where amount = (select max(amount) from insurance);

```
vehicle_mgmt=# select sale_id,amount from insurance where amount = (select max(amount) from insurance);
 sale_id | amount
-----+-----
1000015.00 | 25000.00
(1 row)
```

Complex Queries:

1) Show the name of the customer and vehicle bought by customers before a particular date 2019-12-12.

select fname,minit,lname,V\_name from customer,sales,vehicle  
where order\_date <= '2019-12-12' and sales.cust\_id = customer.cust\_id  
and sales.mfg\_id = vehicle.mfg\_id;

```
vehicle_mgmt=# select fname,minit,lname,V_name from customer,sales,vehicle where order_date <= '2019-12-12' and sales.cust_id = customer.cust_id and sales.mfg_id = vehicle.mfg_id;
fname | minit | lname | v_name
-----+-----+-----+-----
Casey | C     | Park  | Eco Sport
John  | B     | Smith | Baleno
Maddy | B     | Kurk  | I20
Raj   | B     | Prakash | Nissan GTR
Tara  | A     | Rose  | RD
      |      |      |
Text Editor | Ham | Nexon
Maddy | B     | Kurk  | Creta
John  | B     | Smith | Nexon
(8 rows)
```

2) Show the number of vehicles sold by each showroom in year 2020

```
>>select s_id,count(*) from
sales,employee
where sales.emp_id = employee.emp_id and cast(sales.order_date
as varchar) like '2020%' group by s_id;
```

```
vehicle_mgmt=# select s_id,count(*)
vehicle_mgmt=# from sales,employee
vehicle_mgmt=# where sales.emp_id = employee.emp_id and
vehicle_mgmt=# cast(sales.order_date as varchar) like '2020%' group by s_id;
 s_id | count
-----+-----
40001.00 | 1
40002.00 | 1
40003.00 | 1
40004.00 | 3
(4 rows)
```

3) Show the details of the showroom having vehicle 'CRETA'

```
>>select showroom.s_id,location,state,city,contact_no
from showroom join vehicle on showroom.s_id =
vehicle.s_id and v_name = 'Creta';
```

```
vehicle_mgmt=# select showroom.s_id,location,state,city,contact_no
vehicle_mgmt=# from showroom join vehicle on
vehicle_mgmt=# showroom.s_id = vehicle.s_id and v_name = 'Creta';
 s_id | location | state | city | contact_no
-----+-----+-----+-----+-----
40002.00 | 332 Parkson Street | Karnataka | Bangalore | 9449869880
40007.00 | 3321 Indi Street | Karnataka | Bangalore | 8972557202
(2 rows)
```

4) Display the total number of vehicles of different company being sold.

```
>>select substr(mfg_id,1,2) as alpha,count(*) from sales group by
substr(mfg_id,1,2) having count(*) >= 3;
```

```
vehicle_mgmt=# select substr(mfg_id,1,2) as alpha,count(*) from sales group by substr(mfg_id,1,2) having count(*) >= 3;
alpha | count
-----+-----
HU    |      4
FD    |      3
SZ    |      5
TM    |      3
(4 rows)
```

6) show the details of showroom with maximum sales.

```
>>select s_id,count(*) from sales,employee where
      sales.emp_id = employee.emp_id group by s_id order
      by count desc
      fetch FIRST ROW only;
```

```
vehicle_mgmt=# select s_id,count(*) from sales,employee where
vehicle_mgmt=# sales.emp_id = employee.emp_id group by s_id
vehicle_mgmt=# order by count desc
vehicle_mgmt=# fetch FIRST ROW only;
s_id | count
-----+-----
40004.00 |      6
(1 row)

vehicle_mgmt=# select s_id,count(*) from sales,employee where
sales.emp_id = employee.emp_id group by s_id
order by count desc;
s_id | count
-----+-----
40004.00 |      6
40002.00 |      6
40001.00 |      2
40003.00 |      2
40008.00 |      1
40007.00 |      1
40005.00 |      1
40006.00 |      1
(8 rows)
```

**Nested Queries:**

1) Show the female customer details along with their tax percentage from the vehicle.

```
>>select s.cust_id,fname,minit,lname,s.mfg_id,tax_per from sales as s,customer as c
where s.cust_id = c.cust_id and exists (select * from customer where gender = 'F');
```

```
vehicle_mgmt=# select s.cust_id,fname,minit,lname,s.mfg_id,tax_per from sales as s,customer as c where s.cust_id = c.cust_id and exists (select * from customer where gender = 'F');
cust_id | fname | minit | lname | mfg_id | tax_per
-----+-----+-----+-----+-----+-----
10002.00 | John  | B      | Smith | FD-ES-TRN-AT-D-1 | 5
10003.00 | Casey | C      | Park  | FD-ES-TTN-MN-P-1 | 10
10005.00 | Sally | B      | Simon | SZ-SD-ZXI-MN-P-2 | 8
10002.00 | John  | B      | Smith | SZ-BL-B56-MN-D-1 | 10
10006.00 | Raddy | B      | Kurk  | HU-ZB-CRT-AT-P-2681 | 15
10001.00 | James | E      | Borg  | HU-VN-CRT-MN-D-1731 | 8
10007.00 | Sasha | B      | Han   | NM-GTR-TIO-AT-D-4 | 6
10007.00 | Sasha | B      | Han   | RM-KM-KRS-AT-D-30 | 9
10009.00 | Raj   | B      | Prakash | NM-GTR-TIO-MN-P-964 | 13
10004.00 | Tara  | A      | Rose  | FD-ES-TRN-AT-D-3990 | 18
10004.00 | Tara  | A      | Rose  | AD-RB-SPY-AT-D-7454 | 20
10007.00 | Sasha | B      | Han   | TM-NX-XZ-MN-E-310 | 5
10008.00 | Leonard | B    | Tard  | SZ-BL-B56-MN-D-21 | 14
10003.00 | Casey | C      | Park  | HU-VN-CRT-MN-D-53 | 10
10006.00 | Raddy | B      | Kurk  | HU-CR-CRT-AT-D-2 | 12
10001.00 | James | E      | Borg  | SZ-AB0-VXI-MN-P-8873 | 16
10001.00 | James | E      | Borg  | SZ-AB0-VXI-MN-P-8871 | 10
10009.00 | Raj   | B      | Prakash | TM-NX-XZ-MN-D-34775 | 8
10002.00 | John  | B      | Smith | TM-NX-XZ-MN-E-915 | 13
10004.00 | Tara  | A      | Rose  | AD-RB-SPY-AT-P-94 | 20
(20 rows)
```

2) display to which showroom the best employee belongs to.

```
>>select s_id,emp_id from employee where emp_id = (select
emp_id from (select emp_id,count(sale_id) as sc from sales
group by emp_id) sub order by sc desc fetch FIRST ROW only);
```

```
vehicle_mgmt=# select s_id,emp_id from employee where
vehicle_mgmt=# emp_id = (select emp_id from (select emp_id,count(sale_id) as sc
vehicle_mgmt=# from sales group by emp_id) sub order by sc desc
vehicle_mgmt=# fetch FIRST ROW only);
s_id | emp_id
-----+-----
40004.00 | 60005.00
(1 row)
```

3) Show all the sales along with their respective customers who purchased vehicles except year 2018.

```
>>select sale_id,cust_id from sales where sale_id not in (select sale_id from sales
where cast(order_date as varchar) like '2018%');
```

```
vehicle_mgmt=# select sale_id,cust_id from sales where sale_id not in (select sale_id from sales where cast(order_date as varchar) like '2018%');
sale_id | cust_id
-----+-----
100001.00 | 10002.00
100003.00 | 10005.00
100004.00 | 10002.00
100006.00 | 10001.00
100007.00 | 10007.00
100008.00 | 10007.00
100009.00 | 10009.00
100010.00 | 10004.00
100011.00 | 10004.00
100013.00 | 10008.00
100014.00 | 10003.00
100016.00 | 10001.00
100017.00 | 10001.00
100018.00 | 10009.00
100019.00 | 10002.00
100020.00 | 10004.00
(16 rows)
```

4) Show all the employees with their designation working in a particular showroom.

```
>>select emp_id,designation from employee where emp_id in
```



(select sales.emp\_id from sales,employee where s\_id =  
40002 and sales.emp\_id = employee.emp\_id group by sales.emp\_id);

```
vehicle_mgmt=# select emp_id,designation from employee where emp_id in  
vehicle_mgmt-# (select sales.emp_id from sales,employee where  
vehicle_mgmt-# s_id = 40002 and sales.emp_id = employee.emp_id  
vehicle_mgmt-# group by sales.emp_id);  
emp_id | designation  
-----+-----  
60003.00 | Sales Person  
60004.00 | General Manager  
600013.00 | Sales Person  
600014.00 | Sales Person  
(4 rows)
```

5) Show the most often customer car buyers(i.e who bought more than 2 cars).

>>select cust\_id , fname,minit,lname from customer where cust\_id  
in( select cust\_id from sales group by cust\_id having count(\*) > 2);

```
vehicle_mgmt=# select cust_id , fname,minit,lname from customer where cust_id in( select cust_id from sales group by cust_id having count(*) > 2);  
cust_id | fname | minit | lname  
-----+-----+-----+-----  
10001.00 | James | E | Borg  
10002.00 | John | B | Smith  
10004.00 | Tara | A | Rose  
10007.00 | Sasha | B | Han  
(4 rows)
```

## PART - 2

(user access authority)

Four Types of User:

- **Showroom owner:** The Showroom owner is given access to all the privileges except insertion, deletion and updation of the showroom.

```
vehicle_mgmt=# create user dealer1 with password 'deal1' createdb;
CREATE ROLE
```

```
vehicle_mgmt=# grant all privileges on all tables in schema public to dealer1;
GRANT
```

```
vehicle_mgmt=# exit
pes1ug19cs169@pes1ug19cs169:~$ psql -h localhost -d vehicle_mgmt -U dealer1 -p 5432
Password for user dealer1:
psql (13.4 (Ubuntu 13.4-1.pgdg20.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

vehicle_mgmt=> select * from sales;
```

sale_id	emp_id	mfg_id	cust_id	status	tax_per	order_date
100001.00	60001.00	FD-ES-TRN-AT-D-1	10002.00	PENDING	5	2020-10-10
100002.00	60002.00	FD-ES-TRN-AT-D-1	10003.00	PAID	10	2018-12-10
100003.00	60001.00	SZ-SD-ZXI-MN-P-2	10005.00	PENDING	8	2021-10-08
100004.00	60002.00	SZ-BL-BS6-MN-D-1	10002.00	PAID	10	2019-10-06
100005.00	60004.00	HU-20-CRT-AT-P-2681	10006.00	PENDING	15	2018-08-10
100006.00	60005.00	HU-VN-CRT-MN-D-1731	10001.00	PAID	8	2020-10-30
100007.00	60006.00	NM-GTR-TIO-AT-D-4	10007.00	PAID	6	2020-10-10
100008.00	60005.00	RN-KW-KRS-AT-D-30	10007.00	PAID	9	2021-06-15
100009.00	60006.00	NM-GTR-TIO-MN-P-964	10009.00	PENDING	13	2019-10-12
1000010.00	60001.00	FD-ES-TRN-AT-D-3990	10004.00	PENDING	18	2021-08-10
1000011.00	60003.00	AD-R8-SPY-AT-D-7454	10004.00	PENDING	20	2019-10-10
1000012.00	60005.00	TM-NX-XZ-MN-E-310	10007.00	PAID	5	2018-07-08
1000013.00	60003.00	SZ-BL-BS6-MN-D-21	10008.00	PENDING	14	2021-03-15
1000014.00	60002.00	HU-VN-CRT-MN-D-53	10003.00	PAID	10	2020-10-06
1000015.00	60004.00	HU-CR-CRT-AT-D-2	10006.00	PENDING	12	2018-06-10
1000016.00	60004.00	SZ-A80-LXI-MN-P-8873	10001.00	PAID	16	2020-09-30
1000017.00	60006.00	SZ-A80-VXI-MN-P-8871	10001.00	PAID	10	2020-07-10
1000018.00	60003.00	TM-NX-XZ-MN-D-34775	10009.00	PAID	8	2021-01-05
1000019.00	60008.00	TM-NX-XZ-MN-E-915	10002.00	PENDING	13	2019-02-18
1000020.00	60009.00	TM-NX-XZ-MN-D-34775	10004.00	PENDING	20	2021-08-14

```
(20 rows)
```

```
vehicle_mgmt=# revoke insert, update, delete on showroom, company from dealer1;
REVOKE
```

```
vehicle_mgmt=> INSERT into Showroom values (40012,20005,9449869880,'332 Parkson Street','Karnataka','Bangalore', 583222);
ERROR: permission denied for table showroom
vehicle_mgmt=>
```

- **Employee:** The employee is allowed to make changes to the vehicles, sales, customer and insurance.



```
postgres=# \c vehicle_mgmt
You are now connected to database "vehicle_mgmt" as user "postgres".
vehicle_mgmt=# create user employee1 with password 'emp1' createdb;
CREATE ROLE
```

```
vehicle_mgmt=# grant select, insert, update, delete on vehicle, sales , insurance to employee1;
GRANT
vehicle_mgmt=#
```

```
logout
pesiug19cs169@pesiug19cs169:~$ psql -h localhost -d vehicle_mgmt -U employee1 -p 5432
Password for user employee1:
psql (13.4 (Ubuntu 13.4-1.pgdg20.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.
```

```
vehicle_mgmt=> select * from sales;
```

sale_id	emp_id	mfg_id	cust_id	status	tax_per	order_date
100001.00	60001.00	FD-ES-TRN-AT-D-1	10002.00	PENDING	5	2020-10-10
100002.00	60002.00	FD-ES-TRN-AT-D-1	10003.00	PAID	10	2018-12-10
100003.00	60001.00	SZ-SD-ZXI-MN-P-2	10005.00	PENDING	8	2021-10-08
100004.00	60002.00	SZ-BL-BS6-MN-D-1	10002.00	PAID	10	2019-10-06
100005.00	60004.00	HU-20-CRT-AT-P-2681	10006.00	PENDING	15	2018-08-10
100006.00	60005.00	HU-VN-CRT-MN-D-1731	10001.00	PAID	8	2020-10-30
100007.00	60006.00	NM-GTR-TIO-AT-D-4	10007.00	PAID	6	2020-10-10
100008.00	60005.00	RN-KW-KRS-AT-D-30	10007.00	PAID	9	2021-06-15
100009.00	60006.00	NM-GTR-TIO-MN-P-964	10009.00	PENDING	13	2019-10-12
100010.00	60001.00	FD-ES-TRN-AT-D-3990	10004.00	PENDING	18	2021-08-10
100011.00	60003.00	AD-R8-SPY-AT-D-7454	10004.00	PENDING	20	2019-10-10
100012.00	60005.00	TM-NX-XZ-MN-E-310	10007.00	PAID	5	2018-07-08
100013.00	60003.00	SZ-BL-BS6-MN-D-21	10008.00	PENDING	14	2021-03-15
100014.00	60002.00	HU-VN-CRT-MN-D-53	10003.00	PAID	10	2020-10-06
100015.00	60004.00	HU-CR-CRT-AT-D-2	10006.00	PENDING	12	2018-06-10
100016.00	60004.00	SZ-A80-LXI-MN-P-8873	10001.00	PAID	16	2020-09-30
100017.00	60006.00	SZ-A80-VXI-MN-P-8871	10001.00	PAID	10	2020-07-10
100018.00	60003.00	TM-NX-XZ-MN-D-34775	10009.00	PAID	8	2021-01-05
100019.00	60008.00	TM-NX-XZ-MN-E-915	10002.00	PENDING	13	2019-02-18
100020.00	60009.00	TM-NX-XZ-MN-D-34775	10004.00	PENDING	20	2021-08-14

```
(20 rows)
```

```
vehicle_mgmt=> select * from showroom;
ERROR: permission denied for table showroom
vehicle_mgmt=>
```

- **Customer:** The customer can select the respective showroom and vehicle.

```
You are now connected to database "company" as user "postgres".
company=# \c vehicle_mgmt
You are now connected to database "vehicle_mgmt" as user "postgres".
vehicle_mgmt=# create user customer1 with password 'cust1' createdb;
CREATE ROLE
```

```
vehicle_mgmt=# grant select on showroom, vehicle to customer1;
GRANT
vehicle_mgmt=#
```

```

pesiug19cs169@pesiug19cs169:~$ psql -h localhost localhost -d vehicle_mgmt -U customer1 -p 5432
psql: warning: extra command-line argument "localhost" ignored
Password for user customer1:
psql (13.4 (Ubuntu 13.4-1.pgdg20.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

vehicle_mgmt=> select * from showroom
vehicle_mgmt-> ;

```

s_id	d_id	contact_no	location	state	city	pincode
40001.00	20001.00	9998877707	3321 Castle Spring	Karnataka	Bangalore	583221.00
40002.00	20005.00	9449869880	332 Parkson Street	Karnataka	Bangalore	583222.00
40003.00	20007.00	9998874459	3321 Red Road	Karnataka	Bangalore	583223.00
40004.00	20001.00	9930974829	3321 Richmond Road	Karnataka	Bangalore	583224.00
40005.00	20006.00	9779823098	3321 Cashmer Street	Karnataka	Bangalore	583225.00
40006.00	20004.00	9662891236	3321 MG Road	Karnataka	Bangalore	583226.00
40007.00	20005.00	8972557202	3321 Indi Street	Karnataka	Bangalore	583227.00
40008.00	20008.00	9822780442	3321 Strend Street	Karnataka	Bangalore	583228.00

```

(8 rows)

vehicle_mgmt=> select * from employee;
ERROR: permission denied for table employee
vehicle_mgmt=>

```

- **Owner:** The owner is given all privileges mentioned below.

```

vehicle_mgmt=# create user owner with password 'owner' createdb;
CREATE ROLE
vehicle_mgmt=# grant select,insert, update, delete on showroom, company to owner;
GRANT
vehicle_mgmt=#

```

```

vehicle_mgmt=# grant all privileges on all tables in schema public to owner;
GRANT

```

- **U7 and U8** are granted insert permission and they insert into the database simultaneously .

```

janya@janya-desktop:~$ psql -h localhost -d vehicle_mgmt -U u7 -p 5432
Password for user u7:
psql (12.8 (Ubuntu 12.8-0ubuntu0.20.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

vehicle_mgmt=> INSERT into Company values (30010, 'Kia','KI');
INSERT 0 1
vehicle_mgmt=>

```

```

janya@janya-desktop:~$ psql -h localhost -d vehicle_mgmt -U u7 -p 5432
Password for user u7:
[1]+  Stopped                  psql -h localhost -d vehicle_mgmt -U u7 -p 5432

```

```

janya@janya-desktop:~$ psql -h localhost -d vehicle_mgmt -U u8 -p 5432
Password for user u8:
psql (12.8 (Ubuntu 12.8-0ubuntu0.20.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

vehicle_mgmt=> INSERT into Company values (30009, 'Mahindra','MH');
INSERT 0 1
vehicle_mgmt=>

```

c_id	company_name	company_short_name
30001.00	Ford	FD
30002.00	Nissan Motors	NM
30003.00	Tata Motors	TM
30004.00	Toyota	TO
30005.00	Suzuki	SZ
30006.00	Renault	RN
30007.00	Hyundai	HU
30008.00	Audi	AD
30009.00	Mahindra	MH
30010.00	Kia	KI

```

(10 rows)

vehicle_mgmt=#
vehicle_mgmt=#
vehicle_mgmt=#
vehicle_mgmt=#
vehicle_mgmt=#

```



## PART - 3

### (transaction operations)

```
postgres=# end transaction
;
postgres=# \c vehicle_mngt
FATAL: database "vehicle_mngt" does not exist
Previous connection kept
postgres=# \c vehicle_mngt
You are now connected to database "vehicle_mngt" as user "postgres".
vehicle_mngt=# Begin;
BEGIN
vehicle_mngt=# update Vehicle set FType = "Petrol" Where Mfg_Id in (select Mfg_Id
from Vehicle where FType="PETROL");
ERROR:  relation "vehicle" does not exist
LINE 1: update Vehicle set FType = "Petrol" Where Mfg_Id in (select ...
^
postgres=# end transaction
;
ROLLBACK
postgres=# \c vehicle_mngt
FATAL: database "vehicle_mngt" does not exist
Previous connection kept
postgres=# \c vehicle_mngt
You are now connected to database "vehicle_mngt" as user "postgres".
vehicle_mngt=# Begin;
BEGIN
vehicle_mngt=# update Vehicle set FType = 'Petrol' Where Mfg_Id in (select Mfg
_Id from Vehicle where FType='PETROL');
UPDATE 2
vehicle_mngt=#
```

```
pes1ug19cs308@pes1ug19cs308-VirtualBox: ~/Downloads$ sudo su - postgres
[sudo] password for pes1ug19cs308:
postgres@pes1ug19cs308-VirtualBox:~$ psql
psql (13.4 (Ubuntu 13.4-1.pgdg20.04+1))
Type "help" for help.

postgres=# \c vehicle_mngt
You are now connected to database "vehicle_mngt" as user "postgres".
vehicle_mngt=# Begin;
BEGIN
vehicle_mngt=# update Vehicle set FType = "Petrol" Where Mfg_Id in (select Mfg_
Id from Vehicle where FType="PETROL")
vehicle_mngt-## ;
ERROR:  column "PETROL" does not exist
LINE 1: ... Mfg_Id in (select Mfg_Id from Vehicle where FType="PETROL")
^
vehicle_mngt=# end transaction;
ROLLBACK
vehicle_mngt=# Begin;
BEGIN
vehicle_mngt=# update Vehicle set FType = 'Petrol' Where Mfg_Id in (select Mfg_
Id from Vehicle where FType='PETROL');
UPDATE 2
vehicle_mngt=#
```

```
postgres=# ;
ROLLBACK
postgres=# Begin;
BEGIN
postgres=# update Vehicle set FType = "Petrol" Where Mfg_Id in (select Mfg_Id
from Vehicle where FType="PETROL");
ERROR:  relation "vehicle" does not exist
LINE 1: update Vehicle set FType = "Petrol" Where Mfg_Id in (select ...
^
postgres=# end transaction
;
ROLLBACK
postgres=# \c vehicle_mngt
FATAL: database "vehicle_mngt" does not exist
Previous connection kept
postgres=# \c vehicle_mngt
You are now connected to database "vehicle_mngt" as user "postgres".
vehicle_mngt=# Begin;
BEGIN
vehicle_mngt=# update Vehicle set FType = 'Petrol' Where Mfg_Id in (select Mfg
_Id from Vehicle where FType='PETROL');
UPDATE 2
vehicle_mngt=#
```

```
pes1ug19cs308@pes1ug19cs308-VirtualBox: ~/Downloads$ sudo su - postgres
[sudo] password for pes1ug19cs308:
postgres@pes1ug19cs308-VirtualBox:~$ psql
psql (13.4 (Ubuntu 13.4-1.pgdg20.04+1))
Type "help" for help.

postgres=# \c vehicle_mngt
You are now connected to database "vehicle_mngt" as user "postgres".
vehicle_mngt=# Begin;
BEGIN
vehicle_mngt=# update Vehicle set FType = "Petrol" Where Mfg_Id in (select Mfg_
Id from Vehicle where FType="PETROL")
vehicle_mngt-## ;
ERROR:  column "PETROL" does not exist
LINE 1: ... Mfg_Id in (select Mfg_Id from Vehicle where FType="PETROL")
^
vehicle_mngt=# end transaction;
ROLLBACK
vehicle_mngt=# Begin;
BEGIN
vehicle_mngt=# update Vehicle set FType = 'Petrol' Where Mfg_Id in (select Mfg_
Id from Vehicle where FType='PETROL');
UPDATE 2
vehicle_mngt=# end transaction;
COMMIT
vehicle_mngt=#
```

## PERFORMANCE ANALYSIS:

```
vehicle_mgmt=# explain analyze select sale_id from insurance order by amount desc fetch first row only;  
QUERY PLAN
```

```
-----  
Limit (cost=18.40..18.40 rows=1 width=32) (actual time=0.020..0.021 rows=1 loops=1)  
  -> Sort (cost=18.40..19.80 rows=560 width=32) (actual time=0.019..0.020 rows=1 loops=1)  
        Sort Key: amount DESC  
        Sort Method: top-N heapsort  Memory: 25kB  
        -> Seq Scan on insurance (cost=0.00..15.60 rows=560 width=32) (actual time=0.006..0.009 rows=20 loops=1)  
Planning Time: 0.046 ms  
Execution Time: 0.037 ms  
(7 rows)
```

```
vehicle_mgmt=# explain analyze select sale_id from insurance where amount = (select max(amount) from insurance);  
QUERY PLAN
```

```
-----  
Seq Scan on insurance (cost=17.01..34.01 rows=3 width=16) (actual time=0.042..0.044 rows=1 loops=1)  
  Filter: (amount = $0)  
  Rows Removed by Filter: 19  
  InitPlan 1 (returns $0)  
    -> Aggregate (cost=17.00..17.01 rows=1 width=32) (actual time=0.031..0.031 rows=1 loops=1)  
          -> Seq Scan on insurance insurance_1 (cost=0.00..15.60 rows=560 width=16) (actual time=0.002..0.004 rows=20 loops=1)  
Planning Time: 0.080 ms  
Execution Time: 0.059 ms  
(8 rows)
```