DATABASE MANAEMENT SYSTEM

Assignment – 3

LG STORE MANAGEMENT SYSTEM

Achyut Jagini – PES2UG19CS013

Ajith Vivekanandan – PES2UG19CS021

Akif Iqbal – PES2UG19CS027

1. Simple Queries

1. select count(*) from lg_product;

```
                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------
Aggregate  (cost=11.75..11.76 rows=1 width=8) (actual time=0.023..0.024 rows=1 loops=1)
  ->  Seq Scan on lg_product  (cost=0.00..11.40 rows=140 width=0) (actual time=0.015..0.016 rows=16 loops=1)
Planning Time: 0.039 ms
Execution Time: 0.036 ms
(4 rows)
```

```
lg=# select count(*) from lg_product;
 count
-------
    16
(1 row)
```

2. select lgname, count(lgname) from lg_product group by lgname;

```
                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------
HashAggregate  (cost=12.10..13.50 rows=140 width=524) (actual time=0.026..0.028 rows=4 loops=1)
  Group Key: lgname
  Batches: 1  Memory Usage: 40kB
  ->  Seq Scan on lg_product  (cost=0.00..11.40 rows=140 width=516) (actual time=0.018..0.018 rows=16 loops=1)
Planning Time: 0.052 ms
Execution Time: 0.046 ms
(6 rows)
```

```
lg=# select lgname, count(lgname) from lg_product group by lgname;
     lgname      | count
-----------------+-------
 TV              |     4
 AC              |     4
 Washing Machine |     4
 Fridge          |     4
(4 rows)
```

3. select ename, cname from employee, customer where employee.eid=customer.empid;

```
                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------
Hash Join  (cost=12.03..23.42 rows=110 width=236) (actual time=0.025..0.027 rows=4 loops=1)
  Hash Cond: (customer.empid = employee.eid)
  ->  Seq Scan on customer  (cost=0.00..11.10 rows=110 width=126) (actual time=0.009..0.009 rows=4 loops=1)
  ->  Hash  (cost=10.90..10.90 rows=90 width=126) (actual time=0.013..0.013 rows=4 loops=1)
        Buckets: 1024  Batches: 1  Memory Usage: 9kB
        ->  Seq Scan on employee  (cost=0.00..10.90 rows=90 width=126) (actual time=0.008..0.008 rows=4 loops=1)
Planning Time: 0.086 ms
Execution Time: 0.040 ms
(8 rows)
```

```
lg=# select ename, cname from employee, customer where employee.eid=customer.empid;
   ename   | cname
-----------+-------
 Kayling   | Tim
 Josh      | Alice
 Alex      | Ella
 Sebastian | Anna
(4 rows)
```

4. select avg(esalary) from employee;

```
                                                  QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Aggregate  (cost=11.13..11.14 rows=1 width=32) (actual time=0.017..0.018 rows=1 loops=1)
   ->  Seq Scan on employee  (cost=0.00..10.90 rows=90 width=8) (actual time=0.010..0.010 rows=4 loops=1)
 Planning Time: 0.044 ms
 Execution Time: 0.030 ms
(4 rows)
```

```
lg=# select avg(esalary) from employee;
          avg
-----------------------
 6000.0000000000000000
(1 row)
```

5. select cname,count(*) from customer, lg_product as l where customer.cid=l.custid group by cname;

```
                                                  QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 HashAggregate  (cost=24.95..26.05 rows=110 width=126) (actual time=0.034..0.036 rows=4 loops=1)
   Group Key: customer.cname
   Batches: 1  Memory Usage: 24kB
   ->  Hash Join  (cost=12.47..24.25 rows=140 width=118) (actual time=0.025..0.029 rows=16 loops=1)
         Hash Cond: (l.custid = customer.cid)
         ->  Seq Scan on lg_product l  (cost=0.00..11.40 rows=140 width=8) (actual time=0.010..0.011 rows=16 loops=1)
         ->  Hash  (cost=11.10..11.10 rows=110 width=122) (actual time=0.009..0.009 rows=4 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 9kB
               ->  Seq Scan on customer  (cost=0.00..11.10 rows=110 width=122) (actual time=0.006..0.006 rows=4 loops=1)
 Planning Time: 0.089 ms
 Execution Time: 0.058 ms
(11 rows)
```

```
lg=# select cname,count(*) from customer, lg_product as l where customer.cid=l.custid group by cname;
 cname | count
-------+-------
 Alice |     4
 Tim   |     4
 Ella  |     4
 Anna  |     4
(4 rows)
```

2. Complex Queries

1. (select distinct ename as females from employee where ename like 'Kayling')

   union

   (select distinct cname from customer where cgender like 'F');

```
                                                          QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------
 Unique  (cost=22.58..22.59 rows=2 width=118) (actual time=0.032..0.036 rows=4 loops=1)
   ->  Sort  (cost=22.58..22.58 rows=2 width=118) (actual time=0.032..0.035 rows=4 loops=1)
         Sort Key: employee.ename
         Sort Method: quicksort  Memory: 25kB
         ->  Append  (cost=11.13..22.57 rows=2 width=118) (actual time=0.014..0.032 rows=4 loops=1)
               ->  Unique  (cost=11.13..11.14 rows=1 width=118) (actual time=0.014..0.017 rows=1 loops=1)
                     ->  Sort  (cost=11.13..11.14 rows=1 width=118) (actual time=0.014..0.016 rows=1 loops=1)
                           Sort Key: employee.ename
                           Sort Method: quicksort  Memory: 25kB
                           ->  Seq Scan on employee  (cost=0.00..11.13 rows=1 width=118) (actual time=0.011..0.011 rows=1 loops=1)
                                 Filter: ((ename)::text ~~ 'Kayling'::text)
                                 Rows Removed by Filter: 3
               ->  Unique  (cost=11.38..11.39 rows=1 width=118) (actual time=0.013..0.014 rows=3 loops=1)
                     ->  Sort  (cost=11.38..11.39 rows=1 width=118) (actual time=0.013..0.013 rows=3 loops=1)
                           Sort Key: customer.cname
                           Sort Method: quicksort  Memory: 25kB
                           ->  Seq Scan on customer  (cost=0.00..11.38 rows=1 width=118) (actual time=0.005..0.005 rows=3 loops=1)
                                 Filter: ((cgender)::text ~~ 'F'::text)
                                 Rows Removed by Filter: 1
 Planning Time: 0.081 ms
 Execution Time: 0.051 ms
(21 rows)
```

```
lg=# (select distinct ename as females from employee where ename like 'Kayling')
lg-# union
lg-# (select distinct cname from customer where cgender like 'F');
 females
---------
 Alice
 Anna
 Ella
 Kayling
(4 rows)
```

2. (select lgid, cname from lg_product,customer) except (select lgid,cname from lg_product,customer where customer.cname like 'Tim');

```
                                                          QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------
 HashSetOp Except  (cost=0.00..550.25 rows=15400 width=126) (actual time=0.064..0.117 rows=48 loops=1)
   ->  Append  (cost=0.00..472.55 rows=15540 width=126) (actual time=0.019..0.049 rows=80 loops=1)
         ->  Subquery Scan on "*SELECT* 1"  (cost=0.00..369.27 rows=15400 width=126) (actual time=0.019..0.031 rows=64 loops=1)
               ->  Nested Loop  (cost=0.00..215.28 rows=15400 width=122) (actual time=0.019..0.029 rows=64 loops=1)
                     ->  Seq Scan on lg_product  (cost=0.00..11.40 rows=140 width=4) (actual time=0.009..0.010 rows=16 loops=1)
                     ->  Materialize  (cost=0.00..11.65 rows=110 width=118) (actual time=0.001..0.001 rows=4 loops=16)
                           ->  Seq Scan on customer  (cost=0.00..11.10 rows=110 width=118) (actual time=0.005..0.006 rows=4 loops=1)
         ->  Subquery Scan on "*SELECT* 2"  (cost=0.00..25.57 rows=140 width=126) (actual time=0.011..0.014 rows=16 loops=1)
               ->  Nested Loop  (cost=0.00..24.17 rows=140 width=122) (actual time=0.011..0.013 rows=16 loops=1)
                     ->  Seq Scan on customer customer_1  (cost=0.00..11.38 rows=1 width=118) (actual time=0.005..0.005 rows=1 loops=1)
                           Filter: ((cname)::text ~~ 'Tim'::text)
                           Rows Removed by Filter: 3
                     ->  Seq Scan on lg_product lg_product_1  (cost=0.00..11.40 rows=140 width=4) (actual time=0.006..0.007 rows=16 loops=1)
 Planning Time: 0.103 ms
 Execution Time: 0.481 ms
(15 rows)
```

```
lg=# (select lgid, cname from lg_product,customer) except (select lgid,cname from lg_product,customer where customer.cname like 'Tim');
 lgid | cname
------+-------
  457 | Alice
  454 | Anna
  461 | Anna
  462 | Anna
  464 | Ella
  456 | Anna
  466 | Ella
  454 | Alice
  464 | Anna
  468 | Anna
  466 | Alice
  455 | Ella
  464 | Alice
  453 | Anna
  458 | Alice
  468 | Ella
  465 | Anna
  468 | Alice
  459 | Anna
  463 | Anna
  467 | Anna
  453 | Alice
  455 | Alice
  455 | Anna
  460 | Anna
  463 | Alice
  460 | Alice
  453 | Ella
  458 | Anna
  460 | Ella
  459 | Ella
  467 | Ella
  458 | Ella
  456 | Ella
  457 | Anna
  467 | Alice
  459 | Alice
  456 | Alice
  463 | Ella
  457 | Ella
  461 | Ella
  462 | Alice
  465 | Alice
  466 | Anna
  461 | Alice
  454 | Ella
  462 | Ella
  465 | Ella
(48 rows)
```

3. select employee.ename, customer.cemail from employee

   inner join customer on employee.eid=customer.empid where esalary>5500 order by ename asc;

```
                                                                QUERY PLAN
------------------------------------------------------------------------------------------------------------------------------------------
 Sort  (cost=23.86..23.96 rows=37 width=634) (actual time=0.062..0.062 rows=2 loops=1)
   Sort Key: employee.ename
   Sort Method: quicksort  Memory: 25kB
   ->  Hash Join  (cost=11.50..22.90 rows=37 width=634) (actual time=0.052..0.053 rows=2 loops=1)
         Hash Cond: (customer.empid = employee.eid)
         ->  Seq Scan on customer  (cost=0.00..11.10 rows=110 width=524) (actual time=0.023..0.024 rows=4 loops=1)
         ->  Hash  (cost=11.13..11.13 rows=30 width=126) (actual time=0.017..0.017 rows=2 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 9kB
               ->  Seq Scan on employee  (cost=0.00..11.13 rows=30 width=126) (actual time=0.012..0.013 rows=2 loops=1)
                     Filter: (esalary > 5500)
                     Rows Removed by Filter: 2
 Planning Time: 0.088 ms
 Execution Time: 0.079 ms
(13 rows)
```

```
lg=# select employee.ename, customer.cemail from employee
lg-# inner join customer on employee.eid=customer.empid where esalary>5500 order by ename asc;
  ename  |     cemail
---------+-----------------
 Josh    | alice@gmail.com
 Kayling | tim@gmail.com
(2 rows)
```

4. select cname from customer where
(select count(*) from lg_product where cid=custid)>=3;

```
                                                                QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------------------
 Seq Scan on customer  (cost=0.00..1305.25 rows=37 width=118) (actual time=0.020..0.038 rows=4 loops=1)
   Filter: ((SubPlan 1) >= 3)
   SubPlan 1
     ->  Aggregate  (cost=11.75..11.76 rows=1 width=8) (actual time=0.006..0.006 rows=1 loops=4)
           ->  Seq Scan on lg_product  (cost=0.00..11.75 rows=1 width=0) (actual time=0.002..0.003 rows=4 loops=4)
                 Filter: (customer.cid = custid)
                 Rows Removed by Filter: 12
 Planning Time: 0.068 ms
 Execution Time: 0.053 ms
(9 rows)
```

```
lg=# select cname from customer where
lg-# (select count(*) from lg_product where cid=custid)>=3;
 cname
-------
 Tim
 Alice
 Ella
 Anna
(4 rows)
```

5. select ename, cname from employee, customer where
not exists (select * from customer where eID=cID);

```
                                                                QUERY PLAN
-------------------------------------------------------------------------------------------------------------------------------------------
 Nested Loop  (cost=12.47..35.91 rows=110 width=236) (actual time=0.029..0.044 rows=16 loops=1)
   ->  Hash Anti Join  (cost=12.47..23.71 rows=1 width=118) (actual time=0.024..0.026 rows=4 loops=1)
         Hash Cond: (employee.eid = customer_1.cid)
         ->  Seq Scan on employee  (cost=0.00..10.90 rows=90 width=126) (actual time=0.012..0.012 rows=4 loops=1)
         ->  Hash  (cost=11.10..11.10 rows=110 width=4) (actual time=0.008..0.008 rows=4 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 9kB
               ->  Seq Scan on customer customer_1  (cost=0.00..11.10 rows=110 width=4) (actual time=0.005..0.006 rows=4 loops=1)
   ->  Seq Scan on customer  (cost=0.00..11.10 rows=110 width=118) (actual time=0.001..0.001 rows=4 loops=4)
 Planning Time: 0.118 ms
 Execution Time: 0.067 ms
(10 rows)
```

```
lg=# select ename, cname from employee, customer where
lg-# not exists (select * from customer where eID=cID);
   ename    | cname
------------+-------
 Kayling    | Tim
 Kayling    | Alice
 Kayling    | Ella
 Kayling    | Anna
 Josh       | Tim
 Josh       | Alice
 Josh       | Ella
 Josh       | Anna
 Alex       | Tim
 Alex       | Alice
 Alex       | Ella
 Alex       | Anna
 Sebastian  | Tim
 Sebastian  | Alice
 Sebastian  | Ella
 Sebastian  | Anna
(16 rows)
```

3. Multiple Users with different access privilege levels:

```
lg=# create user akif;
CREATE ROLE
lg=# grant all privileges on sale, employee, customer, lg_product to akif;
GRANT
lg=# create user ajith;
CREATE ROLE
lg=# grant delete, update on employee, customer to ajith;
GRANT
lg=# create user achyut;
CREATE ROLE
lg=# grant select, update on sale, payments, lg_product to achyut;
GRANT
```

4. Triggers:

```
drop trigger if exists archive_employee on employee;
drop function if exists ar_employee();

create function ar_employee()
    returns trigger as $$
    begin
        insert into archive_employee (eid, ename, ephone, edob, ehiredate,
esalary, eaccno, eemail, ebank, ebillingid)
        values (old.eid, old.ename, old.ephone, old.edob, old.ehiredate,
old.esalary, old.eaccno, old.eemail, old.ebank, old.ebillingid);
        return old;
    end;
    $$ LANGUAGE 'plpgsql';

create trigger archive_employee
    before delete on employee
    for each row
    execute procedure ar_employee();

create function ar_product()
    returns trigger as $$
    begin
        insert into archive_lg_product (lgid, lgname, custid)
        values (old.lgid, old.lgname, old.custid);
        return old;
    end;
    $$ LANGUAGE 'plpgsql';

create trigger archive_product
    before delete on lg_product
    for each row
    execute procedure ar_product();
```

```
create function log_product_insert()
    returns trigger as $$
    begin
        insert into lg_product_log (lgid, lgname, custid, logger, action)
        values (new.lgid, new.lgname, new.custid, current_date, 'insert');
        return new;
    end;
    $$ language 'plpgsql';

create trigger logging_prod_ins
    after insert on lg_product
    for each row
    execute procedure log_product_insert();

create function log_product_del()
    returns trigger as $$
    begin
        insert into lg_product_log (lgid, lgname, custid, logger, action)
        values (old.lgid, old.lgname, old.custid, current_date, 'delete');
        return old;
    end;
    $$ language 'plpgsql';

create trigger logging_prod_del
    after delete on lg_product
    for each row
    execute procedure log_product_del();

select * from archive_employee;
select * from archive_lg_product;
select * from lg_product_log;
delete from lg_AC where acid=454;
delete from LG_WASHING_MACHINE where wmid=459;
delete from lg_tv where tvid=464;
delete from lg_refrigerator where rid=465;
delete from lg_product where custid=147318;
delete from customer where cid=147318;
delete from employee where ebillingid=113457;
delete from payments where pbillingid=113457;
delete from sale where sbillingid=113457;
select * from archive_employee;
select * from archive_lg_product;
insert into sale values (113465,80000);
```

```
insert into employee values (456,'Eren','6345','1991-11-18','2002-12-
10',45000,5056,'eren@gmail.com','axis',113465);
insert into customer values (134982,'Levi','1987-12-
11',672121,'levi@gmail.com','M','2021-04-21',456);
insert into payments values (0,'card',113465);
insert into lg_product values (678,'Fridge',134982);
insert into lg_product values (690,'TV',134982);
insert into lg_refrigerator values (678,'LG_6_Star',58981,137,0,30,'2029-
12-03',120,700000,100,'Bottom');
insert into LG_TV values (690,'LG_lka',78254,35,'2025-02-
03',300000,95,'8K','AndoidTV');
select * from lg_product_log;
```

```
CREATE FUNCTION
CREATE TRIGGER
CREATE FUNCTION
CREATE TRIGGER
CREATE FUNCTION
CREATE TRIGGER
CREATE FUNCTION
CREATE TRIGGER
 eid | ename | ephone | edob | ehiredate | esalary | eaccno | eemail | ebank | ebillingid
-----+-------+--------+------+-----------+---------+--------+--------+-------+------------
(0 rows)


 lgid | lgname | custid
------+--------+--------
(0 rows)


 lgid | lgname | custid | logger | action
------+--------+--------+--------+--------
(0 rows)


DELETE 1
DELETE 1
DELETE 1
DELETE 1
DELETE 4
DELETE 1
DELETE 1
DELETE 1
DELETE 1
 eid | ename | ephone |    edob    | ehiredate  | esalary | eaccno |     eemail     | ebank | ebillingid
-----+-------+--------+------------+------------+---------+--------+----------------+-------+------------
 600 | Josh  |   9945 | 1987-04-20 | 2001-02-12 |    8000 |   5043 | josh@gmail.com | hdfc  |     113457
(1 row)
```

```
 lgid |     lgname      | custid
------+-----------------+--------
  454 | AC              | 147318
  459 | Washing Machine | 147318
  464 | TV              | 147318
  465 | Fridge          | 147318
(4 rows)


INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
 lgid |     lgname      | custid |   logger   | action
------+-----------------+--------+------------+--------
  454 | AC              | 147318 | 2021-11-07 | delete
  459 | Washing Machine | 147318 | 2021-11-07 | delete
  464 | TV              | 147318 | 2021-11-07 | delete
  465 | Fridge          | 147318 | 2021-11-07 | delete
  678 | Fridge          | 134982 | 2021-11-07 | insert
  690 | TV              | 134982 | 2021-11-07 | insert
(6 rows)
```

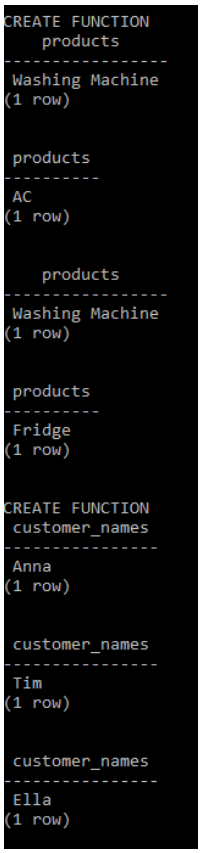
5. Cursors:

```
create or replace function get_product(pid int)
returns table(
    products varchar(255)
)
language plpgsql
as
$$
declare
    cur CURSOR for select lgname from lg_product where lgid=pid;
begin
    open cur;
    return query fetch from cur;
end;
$$;

select * from get_product(463);
select * from get_product(466);
select * from get_product(467);
select * from get_product(453);


create or replace function get_cust_name(pid int)
returns table(
    customer_names varchar(255)
)
language plpgsql
as
$$
declare
    curs CURSOR for select cname from customer where cid=pid;
begin
    open curs;
    return query fetch from curs;
end;
$$;

select * from get_cust_name(119574);
select * from get_cust_name(134252);
select * from get_cust_name(178492);
```



Contributions and Time Spent

Achyut Jagini – Simple & Complex queries – 2 hrs

Ajith Vivekanandan – Simple & Complex queries – 2 hrs

Akif Iqbal – Simple & Complex queries, multiple users with different access privileges levels, 4 Triggers, 2 Cursors and their respective Functions – 5 hrs