ASSIGNMENT-3

ART GALLERY MANAGEMENT DATABASE MANAGEMENT SYSTEM

SUBMITTED BY

-DEEKSHITHA R PES2UG19CS104

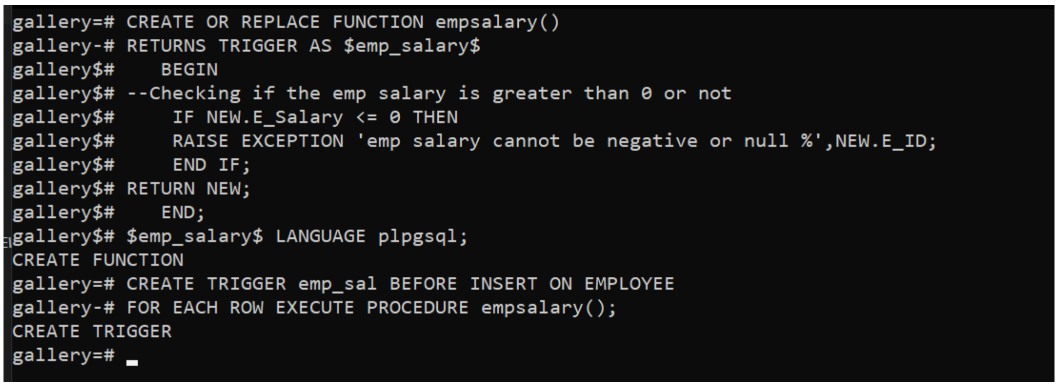
-DEEPA SHREE C V PES2UG19CS105

Department of Computer Science and Engineering

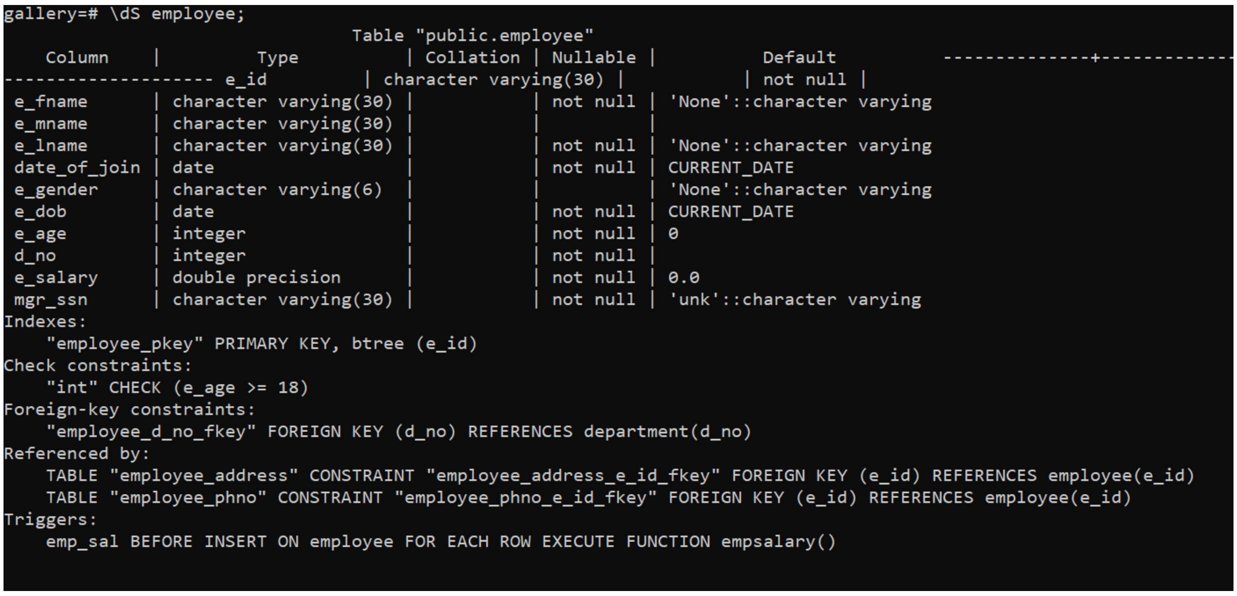
PES UNIVERSITY

Triggers:

1. A trigger to ensure the salary of the employee is a valid number

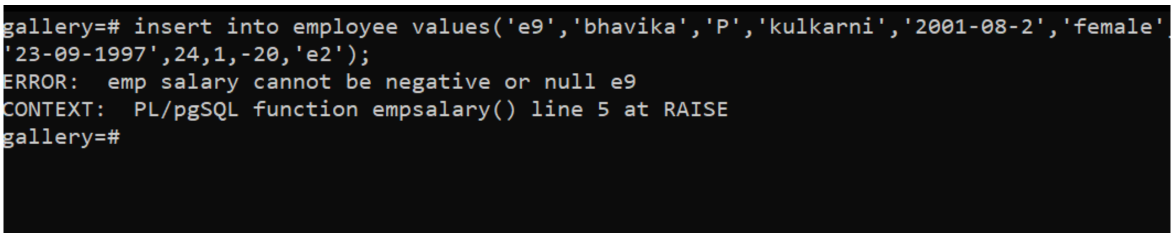


```
gallery=# CREATE OR REPLACE FUNCTION empsalary()
gallery-# RETURNS TRIGGER AS $emp_salary$
gallery$#    BEGIN
gallery$# --Checking if the emp salary is greater than 0 or not
gallery$#     IF NEW.E_Salary <= 0 THEN
gallery$#     RAISE EXCEPTION 'emp salary cannot be negative or null %',NEW.E_ID;
gallery$#     END IF;
gallery$# RETURN NEW;
gallery$#    END;
gallery$# $emp_salary$ LANGUAGE plpgsql;
CREATE FUNCTION
gallery=# CREATE TRIGGER emp_sal BEFORE INSERT ON EMPLOYEE
gallery-# FOR EACH ROW EXECUTE PROCEDURE empsalary();
CREATE TRIGGER
gallery=#
```



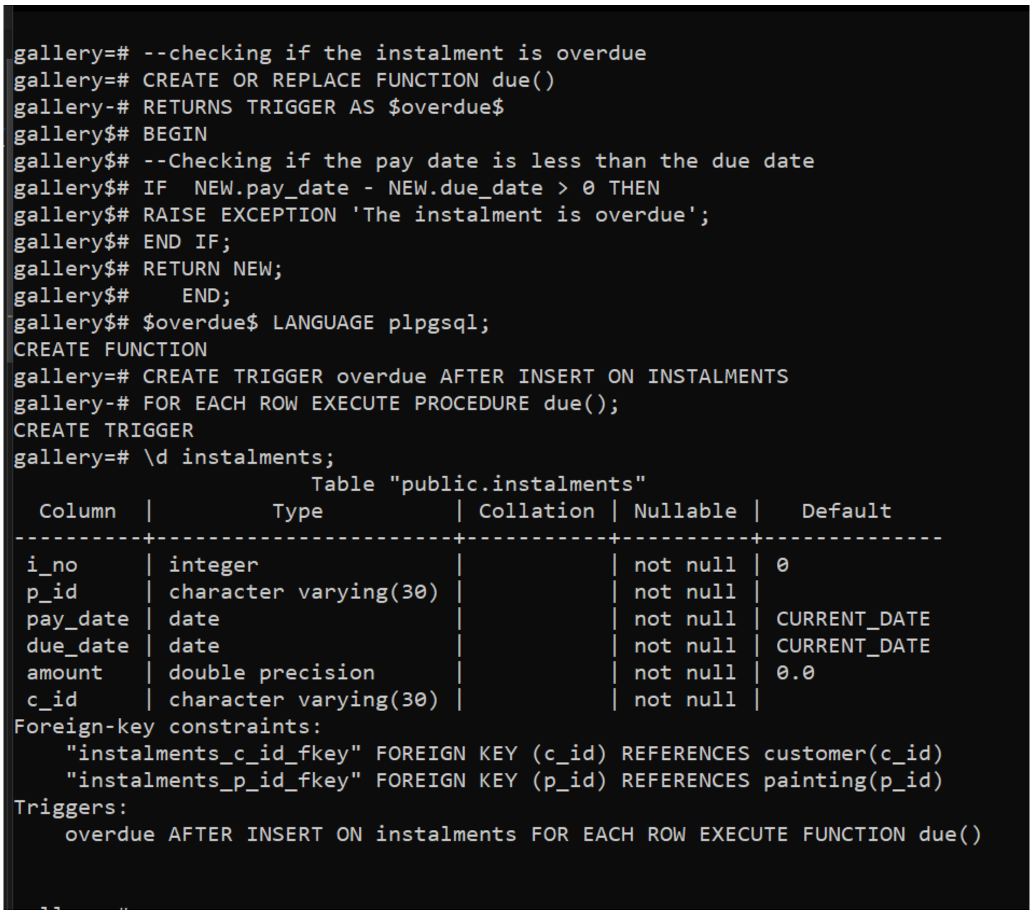
```
gallery=# \dS employee;
                                 Table "public.employee"
    Column    |         Type          | Collation | Nullable |          Default         --------------+------------
------------------- e_id         | character varying(30) |           | not null |
 e_fname      | character varying(30) |           | not null | 'None'::character varying
 e_mname      | character varying(30) |           |          |
 e_lname      | character varying(30) |           | not null | 'None'::character varying
 date_of_join | date                  |           | not null | CURRENT_DATE
 e_gender     | character varying(6)  |           |          | 'None'::character varying
 e_dob        | date                  |           | not null | CURRENT_DATE
 e_age        | integer               |           | not null | 0
 d_no         | integer               |           | not null |
 e_salary     | double precision      |           | not null | 0.0
 mgr_ssn      | character varying(30) |           | not null | 'unk'::character varying
Indexes:
    "employee_pkey" PRIMARY KEY, btree (e_id)
Check constraints:
    "int" CHECK (e_age >= 18)
Foreign-key constraints:
    "employee_d_no_fkey" FOREIGN KEY (d_no) REFERENCES department(d_no)
Referenced by:
    TABLE "employee_address" CONSTRAINT "employee_address_e_id_fkey" FOREIGN KEY (e_id) REFERENCES employee(e_id)
    TABLE "employee_phno" CONSTRAINT "employee_phno_e_id_fkey" FOREIGN KEY (e_id) REFERENCES employee(e_id)
Triggers:
    emp_sal BEFORE INSERT ON employee FOR EACH ROW EXECUTE FUNCTION empsalary()
```

On inserting -20 as the salary, the trigger is aroused.



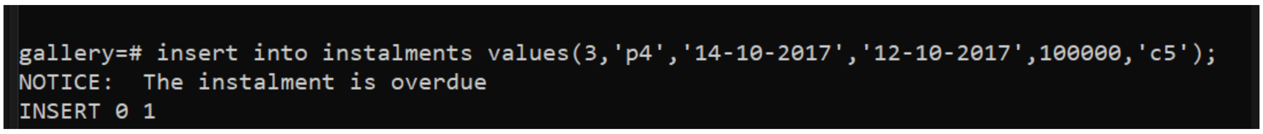
```
gallery=# insert into employee values('e9','bhavika','P','kulkarni','2001-08-2','female'
'23-09-1997',24,1,-20,'e2');
ERROR:  emp salary cannot be negative or null e9
CONTEXT:  PL/pgSQL function empsalary() line 5 at RAISE
gallery=#
```

2. To raise a notice when the payment is overdue



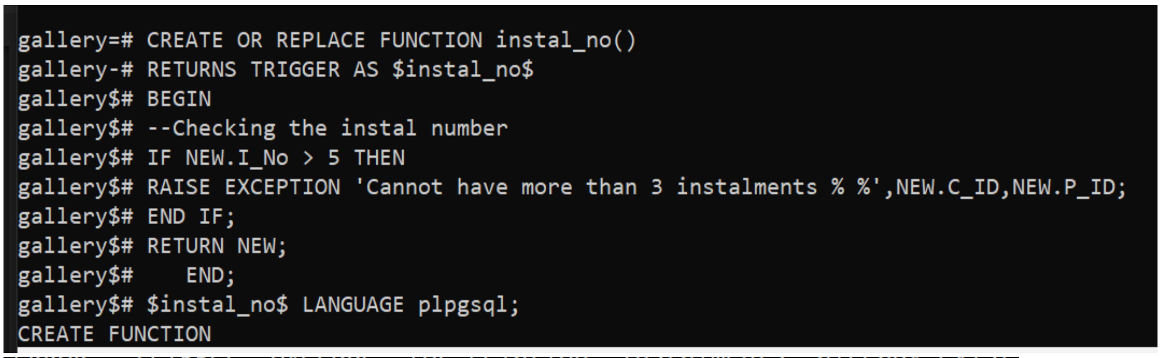
```
gallery=# --checking if the instalment is overdue
gallery=# CREATE OR REPLACE FUNCTION due()
gallery-# RETURNS TRIGGER AS $overdue$
gallery$# BEGIN
gallery$# --Checking if the pay date is less than the due date
gallery$# IF  NEW.pay_date - NEW.due_date > 0 THEN
gallery$# RAISE EXCEPTION 'The instalment is overdue';
gallery$# END IF;
gallery$# RETURN NEW;
gallery$#    END;
gallery$# $overdue$ LANGUAGE plpgsql;
CREATE FUNCTION
gallery=# CREATE TRIGGER overdue AFTER INSERT ON INSTALMENTS
gallery-# FOR EACH ROW EXECUTE PROCEDURE due();
CREATE TRIGGER
gallery=# \d instalments;
                       Table "public.instalments"
  Column  |         Type          | Collation | Nullable |   Default
----------+-----------------------+-----------+----------+--------------
 i_no     | integer               |           | not null | 0
 p_id     | character varying(30) |           | not null |
 pay_date | date                  |           | not null | CURRENT_DATE
 due_date | date                  |           | not null | CURRENT_DATE
 amount   | double precision      |           | not null | 0.0
 c_id     | character varying(30) |           | not null |
Foreign-key constraints:
    "instalments_c_id_fkey" FOREIGN KEY (c_id) REFERENCES customer(c_id)
    "instalments_p_id_fkey" FOREIGN KEY (p_id) REFERENCES painting(p_id)
Triggers:
    overdue AFTER INSERT ON instalments FOR EACH ROW EXECUTE FUNCTION due()
```

A notice is raised when the due date is greater than the pay date

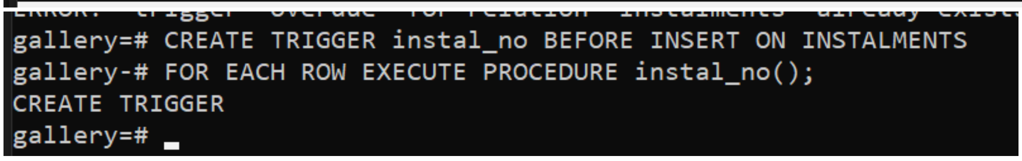


```
gallery=# insert into instalments values(3,'p4','14-10-2017','12-10-2017',100000,'c5');
NOTICE:  The instalment is overdue
INSERT 0 1
```

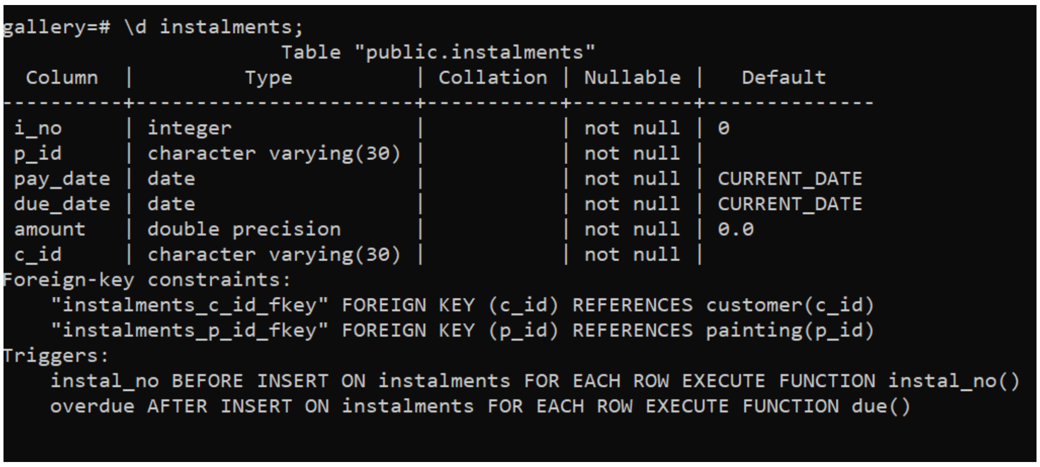
3. **To check the instalment number. It should always be less than or equal to 5.**



```
gallery=# CREATE OR REPLACE FUNCTION instal_no()
gallery-# RETURNS TRIGGER AS $instal_no$
gallery$# BEGIN
gallery$# --Checking the instal number
gallery$# IF NEW.I_No > 5 THEN
gallery$# RAISE EXCEPTION 'Cannot have more than 3 instalments % %',NEW.C_ID,NEW.P_ID;
gallery$# END IF;
gallery$# RETURN NEW;
gallery$#    END;
gallery$# $instal_no$ LANGUAGE plpgsql;
CREATE FUNCTION
```

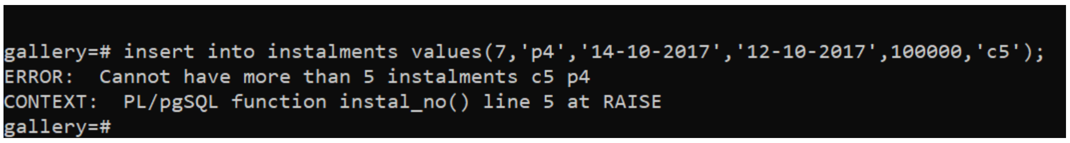


```
ERROR:  trigger overdue for relation instalments already exist
gallery=# CREATE TRIGGER instal_no BEFORE INSERT ON INSTALMENTS
gallery-# FOR EACH ROW EXECUTE PROCEDURE instal_no();
CREATE TRIGGER
gallery=#
```



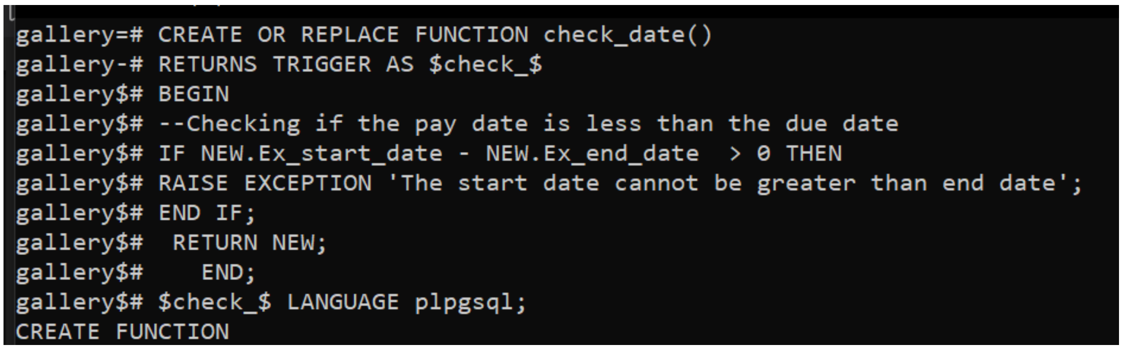
```
gallery=# \d instalments;
                        Table "public.instalments"
  Column  |         Type          | Collation | Nullable |   Default
----------+-----------------------+-----------+----------+--------------
 i_no     | integer               |           | not null | 0
 p_id     | character varying(30) |           | not null |
 pay_date | date                  |           | not null | CURRENT_DATE
 due_date | date                  |           | not null | CURRENT_DATE
 amount   | double precision      |           | not null | 0.0
 c_id     | character varying(30) |           | not null |
Foreign-key constraints:
    "instalments_c_id_fkey" FOREIGN KEY (c_id) REFERENCES customer(c_id)
    "instalments_p_id_fkey" FOREIGN KEY (p_id) REFERENCES painting(p_id)
Triggers:
    instal_no BEFORE INSERT ON instalments FOR EACH ROW EXECUTE FUNCTION instal_no()
    overdue AFTER INSERT ON instalments FOR EACH ROW EXECUTE FUNCTION due()
```

An exception is raised when the instalment number is 7

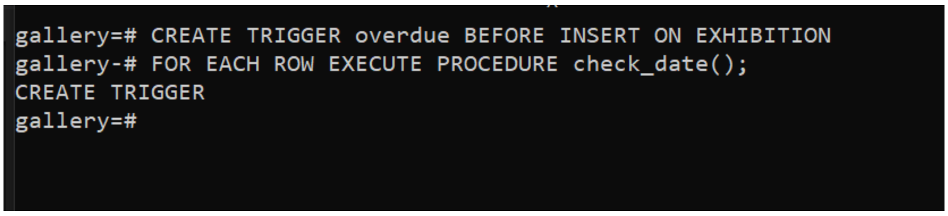


```
gallery=# insert into instalments values(7,'p4','14-10-2017','12-10-2017',100000,'c5');
ERROR:  Cannot have more than 5 instalments c5 p4
CONTEXT:  PL/pgSQL function instal_no() line 5 at RAISE
gallery=#
```

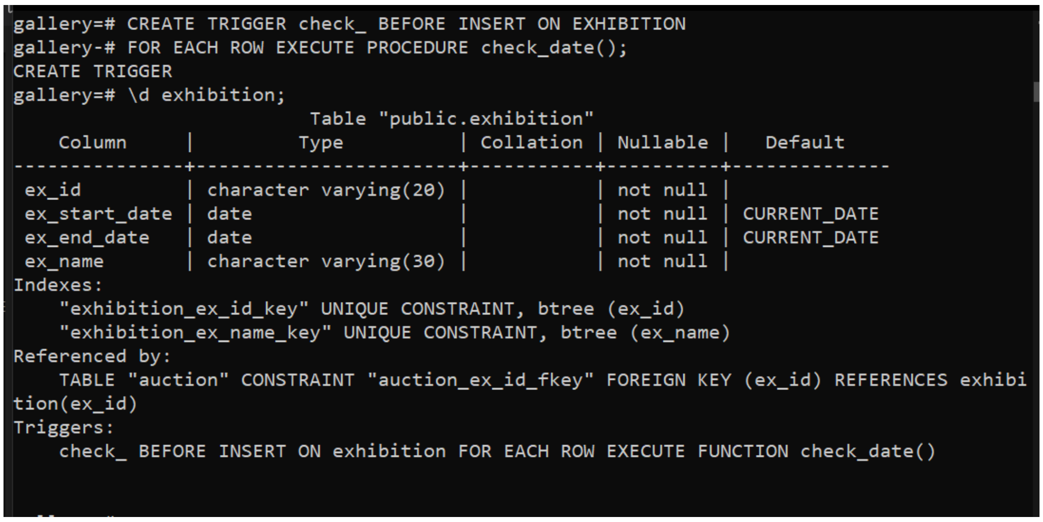
4. A trigger to check the start date is less than the end date of the exhibition



```
gallery=# CREATE OR REPLACE FUNCTION check_date()
gallery-# RETURNS TRIGGER AS $check_$
gallery$# BEGIN
gallery$# --Checking if the pay date is less than the due date
gallery$# IF NEW.Ex_start_date - NEW.Ex_end_date  > 0 THEN
gallery$# RAISE EXCEPTION 'The start date cannot be greater than end date';
gallery$# END IF;
gallery$#  RETURN NEW;
gallery$#    END;
gallery$# $check_$ LANGUAGE plpgsql;
CREATE FUNCTION
```

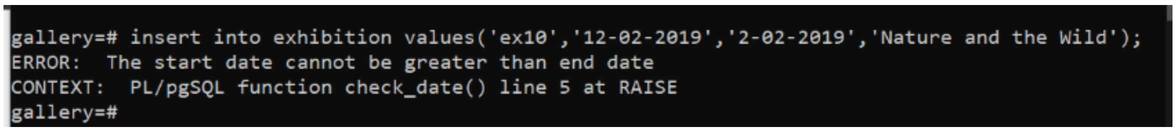


```
gallery=# CREATE TRIGGER overdue BEFORE INSERT ON EXHIBITION
gallery-# FOR EACH ROW EXECUTE PROCEDURE check_date();
CREATE TRIGGER
gallery=#
```



```
gallery=# CREATE TRIGGER check_ BEFORE INSERT ON EXHIBITION
gallery-# FOR EACH ROW EXECUTE PROCEDURE check_date();
CREATE TRIGGER
gallery=# \d exhibition;
                         Table "public.exhibition"
    Column     |         Type          | Collation | Nullable |   Default
---------------+-----------------------+-----------+----------+--------------
 ex_id         | character varying(20) |           | not null |
 ex_start_date | date                  |           | not null | CURRENT_DATE
 ex_end_date   | date                  |           | not null | CURRENT_DATE
 ex_name       | character varying(30) |           | not null |
Indexes:
    "exhibition_ex_id_key" UNIQUE CONSTRAINT, btree (ex_id)
    "exhibition_ex_name_key" UNIQUE CONSTRAINT, btree (ex_name)
Referenced by:
    TABLE "auction" CONSTRAINT "auction_ex_id_fkey" FOREIGN KEY (ex_id) REFERENCES exhibi
tion(ex_id)
Triggers:
    check_ BEFORE INSERT ON exhibition FOR EACH ROW EXECUTE FUNCTION check_date()
```

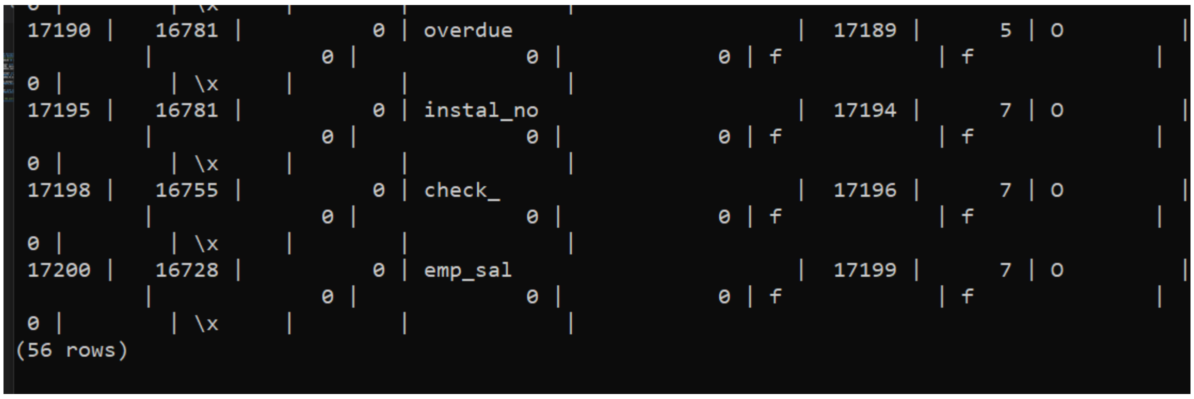
An exception is raised on inserting a start date greater than end date



```
gallery=# insert into exhibition values('ex10','12-02-2019','2-02-2019','Nature and the Wild');
ERROR:  The start date cannot be greater than end date
CONTEXT:  PL/pgSQL function check_date() line 5 at RAISE
gallery=#
```

A list of all the four triggers that are implemented.

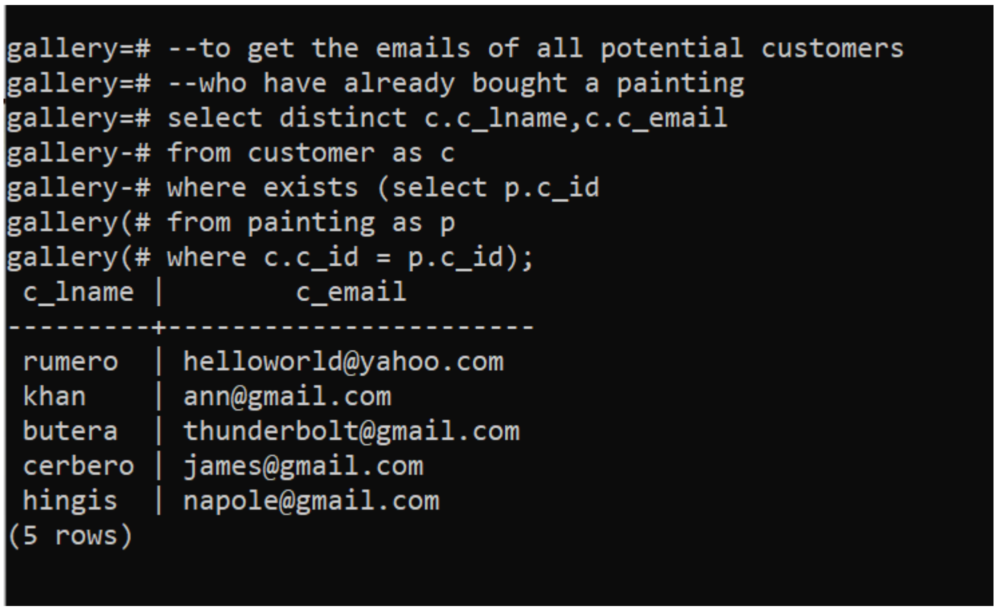
- overdue
- instal_no
- check_
- emp_sal



```
 17190 |   16781 |           0 | overdue                      |  17189 |      5 | O          |
          |             0 |              0 |              0 | f          | f            |
 0 |        | \x     |        |              |
 17195 |   16781 |           0 | instal_no                    |  17194 |      7 | O          |
          |             0 |              0 |              0 | f          | f            |
 0 |        | \x     |        |              |
 17198 |   16755 |           0 | check_                       |  17196 |      7 | O          |
          |             0 |              0 |              0 | f          | f            |
 0 |        | \x     |        |              |
 17200 |   16728 |           0 | emp_sal                      |  17199 |      7 | O          |
          |             0 |              0 |              0 | f          | f            |
 0 |        | \x     |        |              |
(56 rows)
```

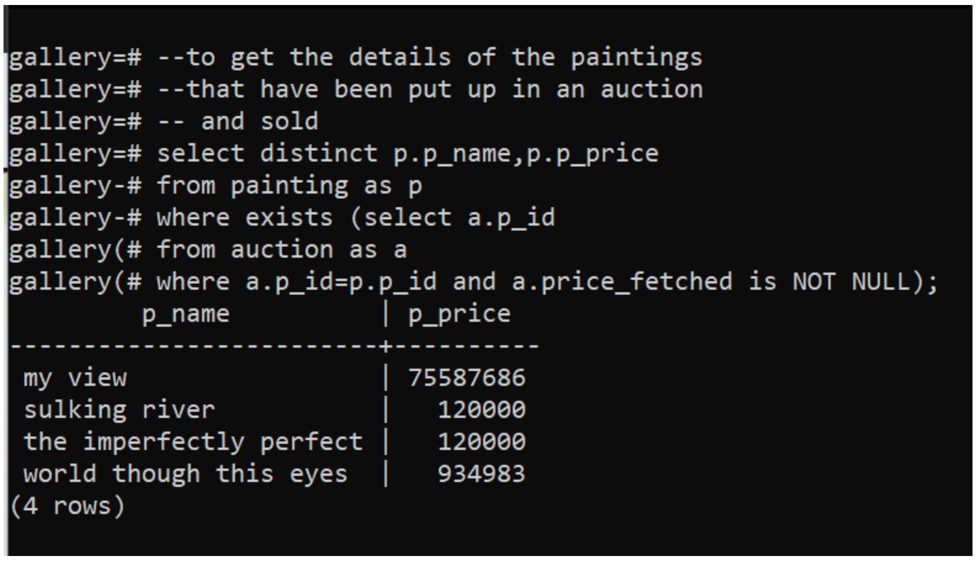
Queries:

1. **Obtain the last name and the email of all the customers who have bought at least one painting.**



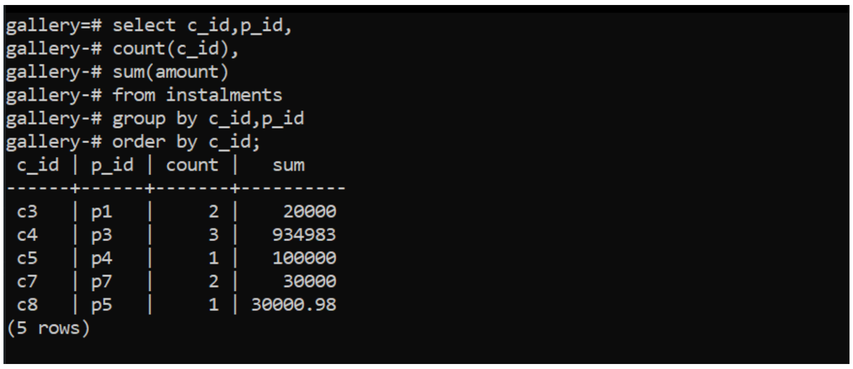
```
gallery=# --to get the emails of all potential customers
gallery=# --who have already bought a painting
gallery=# select distinct c.c_lname,c.c_email
gallery-# from customer as c
gallery-# where exists (select p.c_id
gallery(# from painting as p
gallery(# where c.c_id = p.c_id);
 c_lname |        c_email
---------+-----------------------
 rumero  | helloworld@yahoo.com
 khan    | ann@gmail.com
 butera  | thunderbolt@gmail.com
 cerbero | james@gmail.com
 hingis  | napole@gmail.com
(5 rows)
```

2. To get the details of all the paintings that have been sold in auction



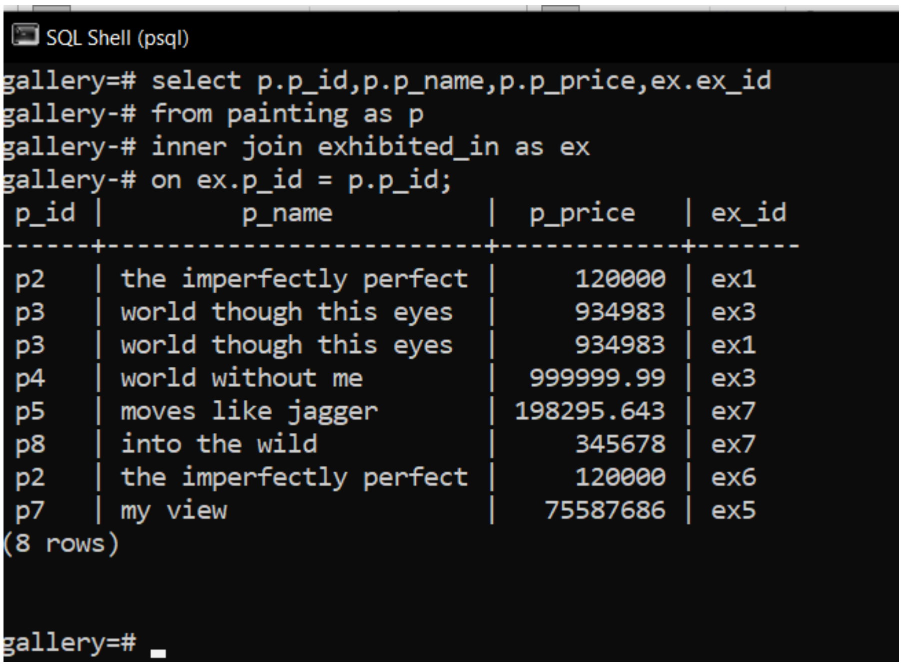
```
gallery=# --to get the details of the paintings
gallery=# --that have been put up in an auction
gallery=# -- and sold
gallery=# select distinct p.p_name,p.p_price
gallery-# from painting as p
gallery-# where exists (select a.p_id
gallery(# from auction as a
gallery(# where a.p_id=p.p_id and a.price_fetched is NOT NULL);
         p_name          | p_price
-------------------------+----------
 my view                 | 75587686
 sulking river           |   120000
 the imperfectly perfect |   120000
 world though this eyes  |   934983
(4 rows)
```

3. To get the number of instalments and the total amount of of each customer



```
gallery=# select c_id,p_id,
gallery-# count(c_id),
gallery-# sum(amount)
gallery-# from instalments
gallery-# group by c_id,p_id
gallery-# order by c_id;
 c_id | p_id | count |   sum
------+------+-------+---------
 c3   | p1   |     2 |    20000
 c4   | p3   |     3 |   934983
 c5   | p4   |     1 |   100000
 c7   | p7   |     2 |    30000
 c8   | p5   |     1 | 30000.98
(5 rows)
```

4. To get the paintings that have been exhibited atleast once



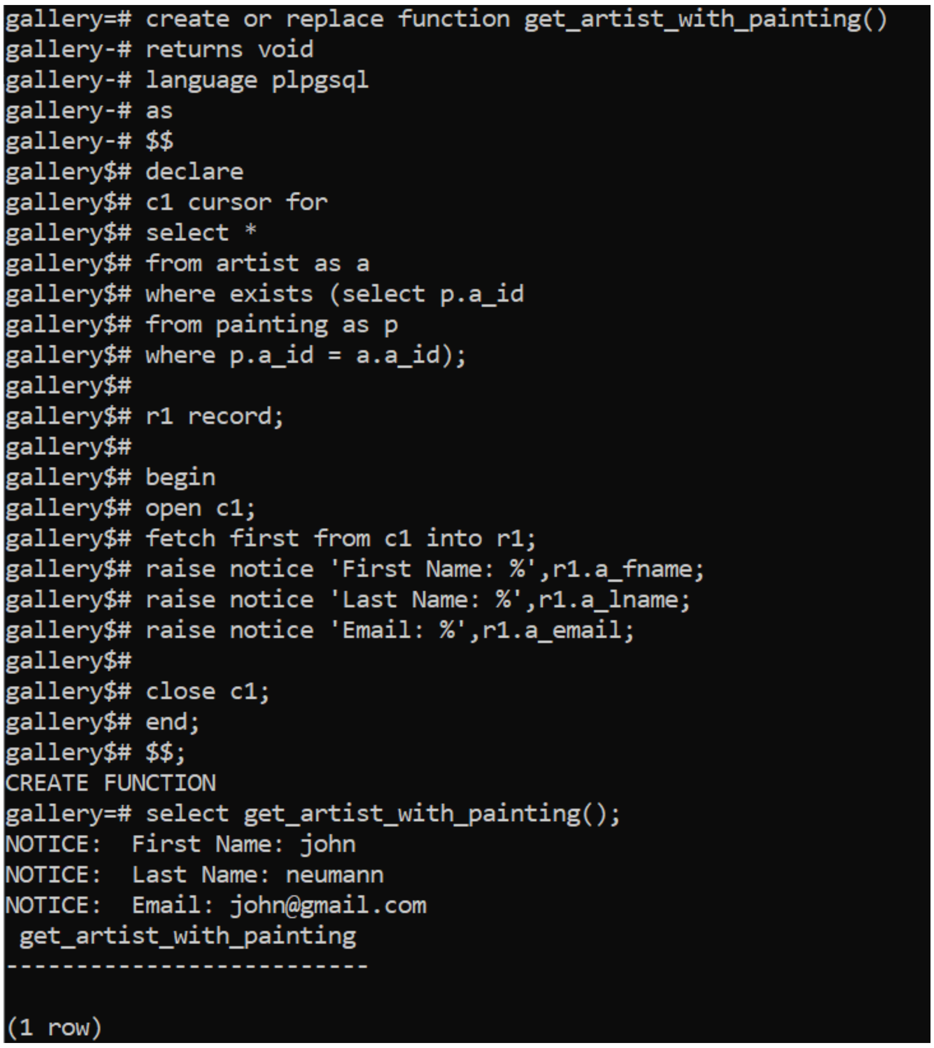
SQL Shell (psql)

```
gallery=# select p.p_id,p.p_name,p.p_price,ex.ex_id
gallery-# from painting as p
gallery-# inner join exhibited_in as ex
gallery-# on ex.p_id = p.p_id;
 p_id |         p_name         |  p_price   | ex_id
------+------------------------+------------+-------
 p2   | the imperfectly perfect |     120000 | ex1
 p3   | world though this eyes |     934983 | ex3
 p3   | world though this eyes |     934983 | ex1
 p4   | world without me       |  999999.99 | ex3
 p5   | moves like jagger      | 198295.643 | ex7
 p8   | into the wild          |     345678 | ex7
 p2   | the imperfectly perfect |     120000 | ex6
 p7   | my view                |   75587686 | ex5
(8 rows)


gallery=# _
```

Cursors:

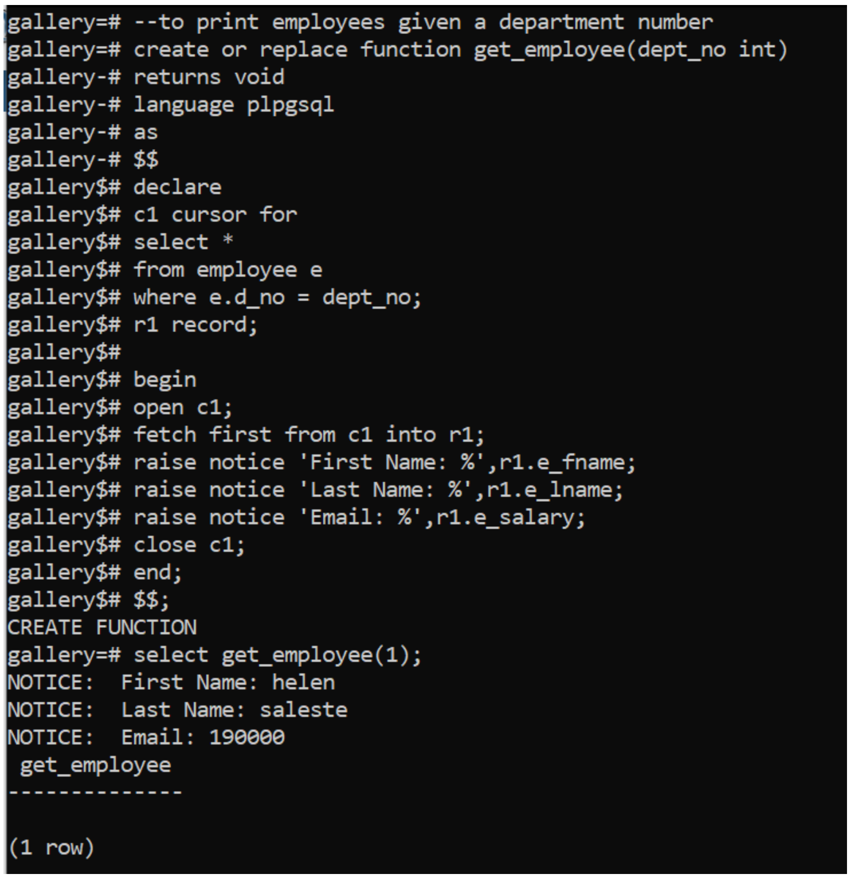
1.To get artists with paintings in the gallery



```
gallery=# create or replace function get_artist_with_painting()
gallery-# returns void
gallery-# language plpgsql
gallery-# as
gallery-# $$
gallery$# declare
gallery$# c1 cursor for
gallery$# select *
gallery$# from artist as a
gallery$# where exists (select p.a_id
gallery$# from painting as p
gallery$# where p.a_id = a.a_id);
gallery$#
gallery$# r1 record;
gallery$#
gallery$# begin
gallery$# open c1;
gallery$# fetch first from c1 into r1;
gallery$# raise notice 'First Name: %',r1.a_fname;
gallery$# raise notice 'Last Name: %',r1.a_lname;
gallery$# raise notice 'Email: %',r1.a_email;
gallery$#
gallery$# close c1;
gallery$# end;
gallery$# $$;
CREATE FUNCTION
gallery=# select get_artist_with_painting();
NOTICE:  First Name: john
NOTICE:  Last Name: neumann
NOTICE:  Email: john@gmail.com
 get_artist_with_painting
--------------------------

(1 row)
```

2.To get employees from a given department number:



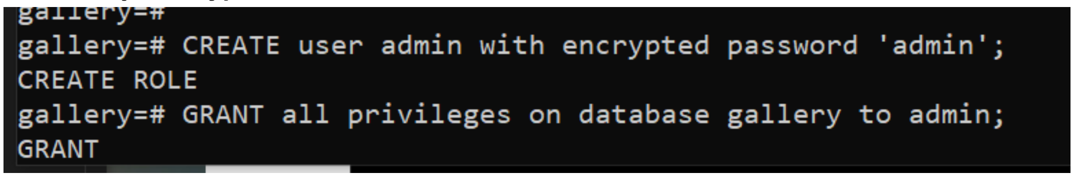
```
gallery=# --to print employees given a department number
gallery=# create or replace function get_employee(dept_no int)
gallery-# returns void
gallery-# language plpgsql
gallery-# as
gallery-# $$
gallery$# declare
gallery$# c1 cursor for
gallery$# select *
gallery$# from employee e
gallery$# where e.d_no = dept_no;
gallery$# r1 record;
gallery$#
gallery$# begin
gallery$# open c1;
gallery$# fetch first from c1 into r1;
gallery$# raise notice 'First Name: %',r1.e_fname;
gallery$# raise notice 'Last Name: %',r1.e_lname;
gallery$# raise notice 'Email: %',r1.e_salary;
gallery$# close c1;
gallery$# end;
gallery$# $$;
CREATE FUNCTION
gallery=# select get_employee(1);
NOTICE:  First Name: helen
NOTICE:  Last Name: saleste
NOTICE:  Email: 190000
 get_employee
--------------

(1 row)
```

User privileges:

A total of five users are created to manage the whole database.

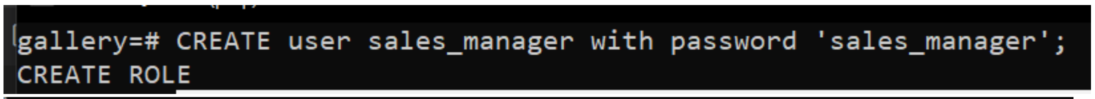
1. **Admin: Has all privileges on the database(Create,Connect and Temporary)**



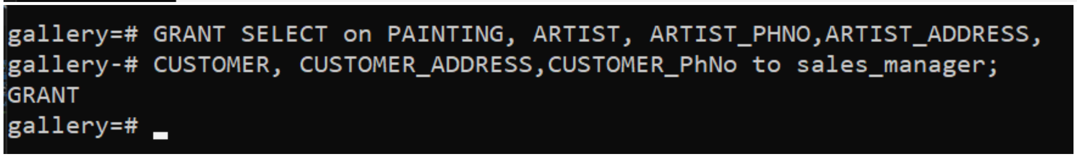
```
gallery=#
gallery=# CREATE user admin with encrypted password 'admin';
CREATE ROLE
gallery=# GRANT all privileges on database gallery to admin;
GRANT
```

2. **Sales manager is given select privileges on:**

PAINTING, ARTIST, ARTIST_PHNO,ARTIST_ADDRESS, CUSTOMER, CUSTOMER_ADDRESS,CUSTOMER_PhNo

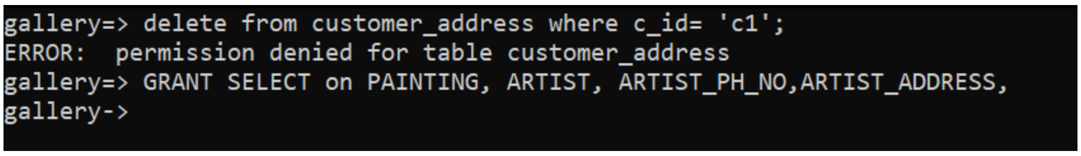


```
gallery=# CREATE user sales_manager with password 'sales_manager';
CREATE ROLE
```



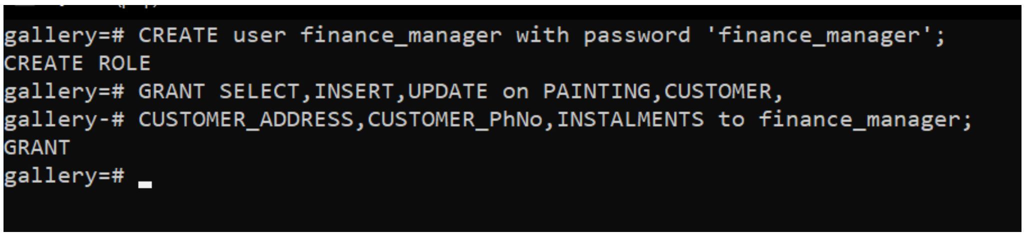
```
gallery=# GRANT SELECT on PAINTING, ARTIST, ARTIST_PHNO,ARTIST_ADDRESS,
gallery-# CUSTOMER, CUSTOMER_ADDRESS,CUSTOMER_PhNo to sales_manager;
GRANT
gallery=# _
```

The user can't delete a record because only select is granted.



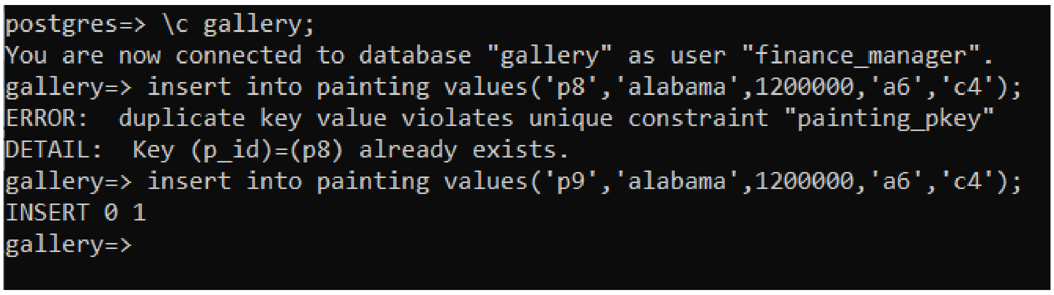
```
gallery=> delete from customer_address where c_id= 'c1';
ERROR:  permission denied for table customer_address
gallery=> GRANT SELECT on PAINTING, ARTIST, ARTIST_PH_NO,ARTIST_ADDRESS,
gallery->
```

3. **Finance Manager is given SELECT, INSERT, UPDATE PRIVILEGES ON PAINTING,CUSTOMER,CUSTOMER_ADDRESS,CUSTOMER_PhNo, INSTALMENTS**



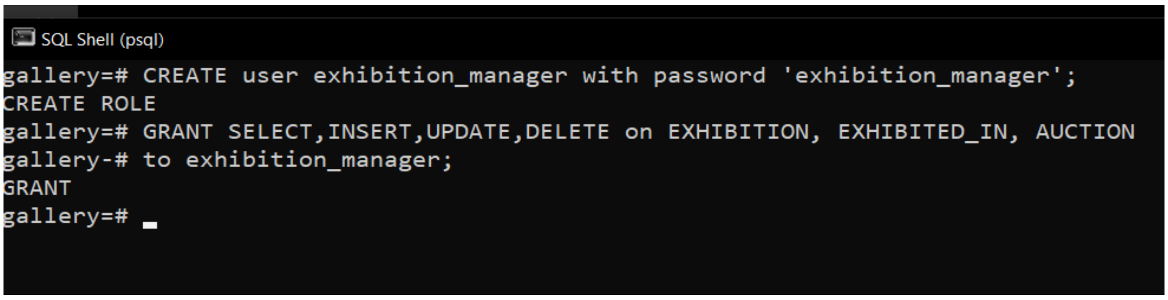
```
gallery=# CREATE user finance_manager with password 'finance_manager';
CREATE ROLE
gallery=# GRANT SELECT,INSERT,UPDATE on PAINTING,CUSTOMER,
gallery-# CUSTOMER_ADDRESS,CUSTOMER_PhNo,INSTALMENTS to finance_manager;
GRANT
gallery=# _
```

Inserting a new value into the relation painting



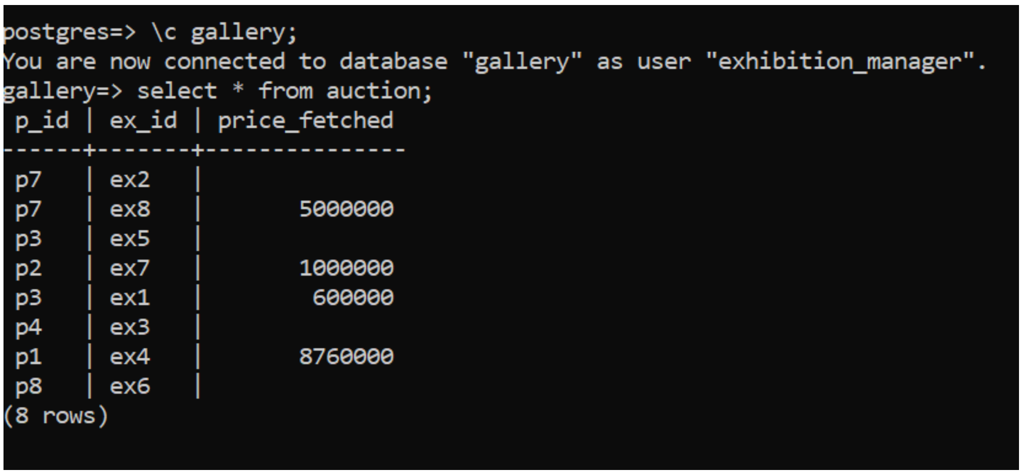
```
postgres=> \c gallery;
You are now connected to database "gallery" as user "finance_manager".
gallery=> insert into painting values('p8','alabama',1200000,'a6','c4');
ERROR:  duplicate key value violates unique constraint "painting_pkey"
DETAIL:  Key (p_id)=(p8) already exists.
gallery=> insert into painting values('p9','alabama',1200000,'a6','c4');
INSERT 0 1
gallery=>
```

4. **Exhibition manager:**
   Has select, insert, update, delete privileges on exhibition, exhibited_in, auction



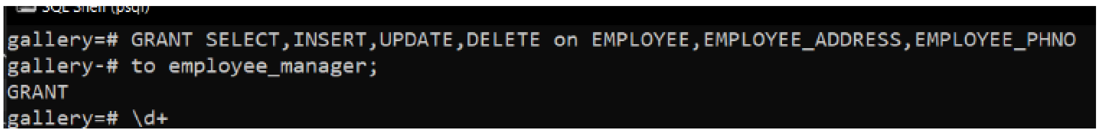
```
gallery=# CREATE user exhibition_manager with password 'exhibition_manager';
CREATE ROLE
gallery=# GRANT SELECT,INSERT,UPDATE,DELETE on EXHIBITION, EXHIBITED_IN, AUCTION
gallery-# to exhibition_manager;
GRANT
gallery=# _
```

Select * on auction by exhibition manager:



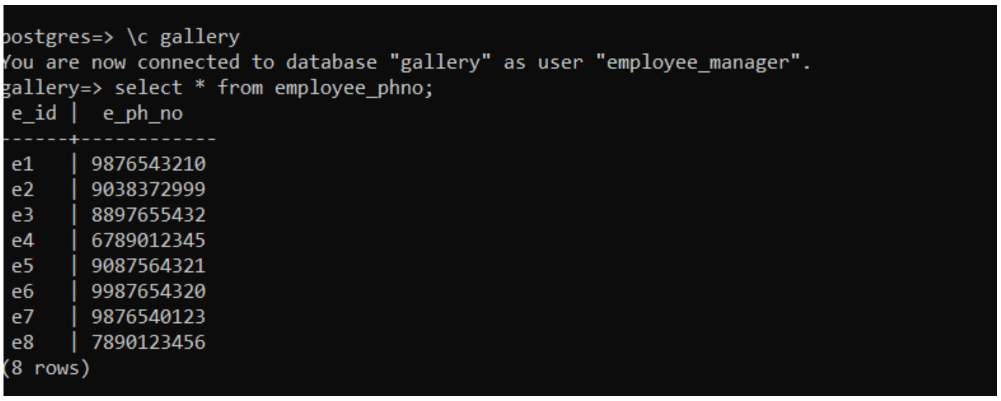
```
postgres=> \c gallery;
You are now connected to database "gallery" as user "exhibition_manager".
gallery=> select * from auction;
 p_id | ex_id | price_fetched
------+-------+---------------
 p7   | ex2   |
 p7   | ex8   |       5000000
 p3   | ex5   |
 p2   | ex7   |       1000000
 p3   | ex1   |        600000
 p4   | ex3   |
 p1   | ex4   |       8760000
 p8   | ex6   |
(8 rows)
```

5. **Employee manager:**
   Has select, insert, update, delete privileges on:
   EMPLOYEE,EMPLOYEE_ADDRESS,EMPLOYEE_PHNO



```
gallery=# GRANT SELECT,INSERT,UPDATE,DELETE on EMPLOYEE,EMPLOYEE_ADDRESS,EMPLOYEE_PHNO
gallery-# to employee_manager;
GRANT
gallery=# \d+
```

select * from employee_phno by employee_manager:



```
postgres=> \c gallery
You are now connected to database "gallery" as user "employee_manager".
gallery=> select * from employee_phno;
 e_id |   e_ph_no
------+------------
 e1   | 9876543210
 e2   | 9038372999
 e3   | 8897655432
 e4   | 6789012345
 e5   | 9087564321
 e6   | 9987654320
 e7   | 9876540123
 e8   | 7890123456
(8 rows)
```

Individual Contributions:
Deekshitha: 2 queries and cursors
Deepa: Triggers, 2 queries and user