Assignment

1. **Write a query to display the employee full name**

create or replace function display\_empName()

returns table(

fullName text

)

language plpgsql

as

$$

begin

return query

select concat(first\_name,' ',last\_name) from staff;

end;$$

select \* from display\_empName();

1. **Write a query to get the last\_update date. Check if you have functions to get the year alone**

create or replace function getYearOfUpdate()

returns table(

yearOfLastUpdate numeric

)

language plpgsql

as $$

begin

return query

select extract(year from last\_update) from film;

end;$$

select \* from getYearOfUpdate();

1. **Write a query to display the staff\_name and the address of the store**

select \* from getFilmNameAndActorName('The%');

select concat(address.address,' ',address.address2,' ',address.district,' ',address.postal\_code),

concat(staff.first\_name,' ',staff.last\_name)emp\_name from staff join address on staff.address\_id=address.address\_id

1. **Display the number of customer in every city with the city name**

select city.city city\_name,count(customer) totalcustomer from city join address on city.city\_id= address.city\_id

join customer on address.address\_id = customer.address\_id group by city.city

1. **A customer has lost the the film he/she rented in that case please display teh film name and the cost he has to pay for that**

select concat(first\_name,' ',last\_name) as customer\_name ,  
film.title as lost\_film,  
film.replacement\_cost as cost\_to\_pay  
from customer  
join rental on customer.customer\_id=rental.customer\_id  
join inventory on rental.inventory\_id=inventory.inventory\_id  
join film on inventory.film\_id=film.film\_id  
left join payment on rental.rental\_id=payment.rental\_id  
where payment.rental\_id is null

anather ways

select title from film join inventory on film.film\_id=inventory.film\_id  
join rental on inventory.inventory\_id=rental.inventory\_id  
join payment on rental.customer\_id=payment.customer\_id;

1. **Take the customer name and get the total cost spent for renting(use joins)**

select concat(first\_name,' ',last\_name) as customer\_name , sum(payment.amount)as total\_spend from customer join payment on customer.customer\_id=payment.customer\_id group by customer.customer\_id,customer\_name

1. **Create a stored procedure that will update the length of the file given the title and print if it was successful. The length should be updated only if the new length is greater than the old one**

create or replace procedure update\_film\_length(n\_title varchar,new\_length int)

language plpgsql

as

$$

begin

update film set length=new\_length where title=n\_title and new\_length>length;

if found then

raise notice 'film length updeted successfull.';

else

raise notice 'not updeted';

end if;

end;

$$;

call update\_film\_length('Chamber Italian',180)

1. **Create a function that will return all the customer names, full address(including city name and country)**

create or replace function get\_customer\_Address()

returns table(customer\_name text,full\_address text)

language plpgsql

as

$$

begin

return query select concat(first\_name,' ',last\_name) as customer\_name,

concat(address.address,' ',city.city,' ',country.country)as full\_address

from customer

join address on customer.address\_id=address.address\_id

join city on address.city\_id=city.city\_id

join country on city.country\_id=country.country\_id;

end;

$$;

select \*from get\_customer\_Address()

**9) Create a procedure that will print the remarks on how much the customer is spending on rental**

**0 : Not great**

**0-5 : Beginner**

**>5 : Good one**

create or replace procedure cst\_spend\_remark(customer\_id int)

language plpgsql

as

$$

declare

total\_spent integer;

remark varchar;

begin

select sum(payment.amount) into total\_spent from payment where payment.customer\_id=payment.customer\_id;

if total\_spent is null then

raise notice 'customer does not have any retal spending';

else

if total\_spent=0 then

remark:='not great';

elseif total\_spent <=5 then

remark:='beginner';

else

remark:='good one';

end if;

raise notice ' remark: %',remark;

end if;

end;

$$;

call cst\_spend\_remark(2);

**10) Print the actor name(full name) of all the actors who have starred in movies that have been rented more than 2 times**

select concat(first\_name,' ',last\_name) from film

join film\_actor on film\_actor.actor\_id=film.film\_id

join actor on film\_actor.actor\_id = actor.actor\_id

where actor.actor\_id in

(select actor\_id from film\_actor group by actor\_id having count(actor\_id)>=2);

**11) Print the name of the actors who has starred in movie that is rented the most**

SELECT CONCAT(actor.first\_name,' ',actor.last\_name) , rental\_rate AS Full\_name

FROM actor JOIN film\_actor ON actor.actor\_id = film\_actor.actor\_id

join film on film\_actor.film\_id=film.film\_id

where film.rental\_rate =

(select max(film.rental\_rate) from film

)

12) **print all the actor names and the total amount of rent collected by them if any**

SELECT CONCAT(actor.first\_name,' ',actor.last\_name) AS Full\_name ,sum(amount)

FROM actor JOIN film\_actor ON actor.actor\_id = film\_actor.actor\_id

join film on film\_actor.film\_id=film.film\_id

join inventory on film.film\_id = inventory.film\_id

join rental on inventory.inventory\_id = rental.inventory\_id

join payment on rental.rental\_id=payment.rental\_id group by actor.actor\_id