--Section B

--

--Write a SQL code that creates the tables to hold your report sections

--Create and display the Detailed Table

Drop Table If Exists detailed;

Create Table detailed

(

customer\_id int,

full\_name VarChar(50),

email VarChar(50),

total\_rentals int,

address VarChar(50),

full\_address VarChar(50)

);

Select \* From detailed

--Create and display the Summary Table

Drop Table If Exists summary;

Create Table Summary

(

full\_name VarChar(50),

total\_rentals int,

email VarChar(50),

full\_address VarChar(50)

);

Select \* From summary

--Section C

--

--Write a SQL query that will extract the raw data needed for the detailed

--section of your report from the source data and verify the data's accuracy

Insert into Detailed

select r.customer\_id,

Concat(c.first\_name, ' ', c.last\_name) As full\_name,

c.email,

count(r.customer\_id) As total\_rentals,

a.address,

concat(city.city, ', ', country.country) As full\_address

From rental r

inner join customer c on r.customer\_id = c.customer\_id

inner join address a on c.address\_id = a.address\_id

inner join city on a.city\_id = city.city\_id

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

group by r.customer\_id,c.first\_name, c.last\_name, c.email, a.address, city.city, country.country

order by total\_rentals desc

limit 50;

select \* from detailed

--Section D

--Insert Data into Summary Report

insert into summary

select full\_name, total\_rentals, email, full\_address

from detailed;

commit;

end;

select \* from summary

--Section E

--Write a SQL Code that creates a trigger on the detailed table

--of the report that will continually update the summary table

--as data is added to the detailed table

Drop Function if exists update\_summary\_table() cascade;

Create Function update\_summary\_table()

Returns Trigger

As $$

Begin

Insert Into summary

Values(New.full\_name, New.total\_rentals, New.email, new.full\_address);

Return New;

End

$$

Language PLPGSQL;

--Insert Data into Detailed and Create Trigger

Drop trigger if exists Update\_summary on detailed;

Create Trigger Update\_Summary

after insert on detailed

for each row

execute procedure update\_summary\_table();

Insert Into detailed

Values(9991, 'Ellen Degeneres', 'edegeneres@gmail.com', 145 ,'123 Fake St', 'New York, USA');

select \* from summary

order by total\_rentals desc;

--Section F

--Create a stored procedure that can be used to refresh the

--data in both your detailed and summary tables. The

--procedure should clear the contents of the detailed

--and summary tables and perform the ETL load process from

--part C and include comments that identify how often the

--stored procedure should be executed

Create Procedure Refresh\_data()

Language SQL

As $$

Delete from detailed;

Delete from summary;

Insert into Detailed

select r.customer\_id,

Concat(c.first\_name, ' ', c.last\_name) As full\_name,

c.email,

count(r.customer\_id) As total\_rentals,

a.address,

concat(city.city, ', ', country.country) As full\_address

From rental r

inner join customer c on r.customer\_id = c.customer\_id

inner join address a on c.address\_id = a.address\_id

inner join city on a.city\_id = city.city\_id

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

group by r.customer\_id,c.first\_name, c.last\_name, c.email, a.address, city.city, country.country

order by total\_rentals desc

limit 50;

$$;

Call refresh\_data();

Select \* from Summary;