

Business Question: Who are the 10 most recent customers to my store?

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

Drop Table if exists detailed\_report;

Create Table detailed\_report(

customer\_id int NOT NULL,

rental\_id int Not Null,

first\_name varchar,

last\_name varchar,

return\_date date,

email varchar,

Primary Key (customer\_id, rental\_id),

Foreign Key (rental\_id)

REFERENCES rental(rental\_id),

Foreign Key (customer\_id)

REFERENCES customer(customer\_id)

);

--CREATE summary table

Drop Table if exists summary\_report;

Create Table summary\_report(

first\_name varchar,

last\_name varchar,

return\_date date,

email varchar

);

--C. Write a SQL query that will extract the raw data needed for the Detailed section of your report from the source database

INSERT INTO detailed\_report (

customer\_id,

first\_name,

last\_name,

email,

rental\_id,

return\_date)

SELECT

c.customer\_id, c.first\_name, c.last\_name, c.email,

r.rental\_id, r.return\_date

FROM rental AS r

INNER JOIN customer AS c ON c.customer\_id = r.customer\_id;

--D. Write code for function(s) that perform the transformation(s) you identified in part A4.

drop table if exists summary\_report;

create table summary\_report( full\_name varchar, email varchar, return\_date date);

insert into summary\_report

select concat(first\_name, " ", last\_name)

as full\_name

FROM detailed\_report;

commit;

end;

CREATE FUNCTION Update\_return\_date()

RETURNS TRIGGER AS \$\$

Begin

DELETE FROM summary\_report;

INSERT INTO summary\_report (

SELECT CONCAT\_(name, " ", last\_name) as full\_name

return\_date,

email

FROM detailed\_report

group by first\_name, last\_name, return\_date , email

ORDER BY return\_date desc Limit 10);

RETURN NEW;

LANGUAGE PLPGSQL \$\$;END;

--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.

Create Trigger summary\_update

After Insert on detailed\_report

For each STATEMENT

Execute Procedure Update\_return\_date();

--F. Create a stored procedure that can be used to refresh the data in both your detailed and summary tables.

CREATE PROCEDURE reports()

LANGUAGE PLPGSQL

AS \$\$

BEGIN

DELETE FROM detailed\_report;

INSERT INTO detailed\_report (

customer\_id,

first\_name,

last\_name,

email,

```
        rental_id,  
        return_date)  
SELECT  
        c.customer_id, c.first_name, c.last_name, c.email,  
        r.rental_id, r.return_date  
FROM rentals AS r  
INNER JOIN customers AS c ON c.customer_id = r.customer_id;  
END; $$;
```

--The stored procedure can be ran at the beginning of the month. To call stored  
--procedure query CALL reports(); To view results query SELECT \* FROM detailed\_report;  
--SELECT \* FROM summary\_report;. Freshness will be consistent if ran on the 1st of every month.

--1. Explain how the stored procedure can be run on a schedule to ensure data freshness.

The stored procedure can be called through PGAGENT. To accomplish this the user must have downloaded, installed and have set up PGAGENT. The user can define the monthly schedule in the scheduling tab of PGAGENT to run the stored procedure to automatically update detailed and summary reports every 1st of the month.