Question: Who is the best sales representative, how much he/she sold, and what store is doing the best

Business Report Summary:

1. One type of business report that could be created from the DVD database provided, is a sale's report. The report should give us a bigger and clearer picture of who is our most talented and prize staff member is. This report should define the top sales representatives, as well as how much they sold and the store they come from. With this information the report will give us a better understanding of which rental stores is doing the best so we can reward the store as well as their staffs appropriately.
2. The tables that will provide the necessary information to complete both the detailed & summary section of the report are Store, Address, Staff, & Payment.
3. Specific fields of each table: (Primary key is bolded)

Store	**store_id**, manager_staff_id, address_id
Address	**address_id**, address
Staff	**staff_id**, first_name, last_name, phone_number, total_sale
Payment	**payment_id**, staff_id, amount

4. One field in the detail section that would need a custom transformation would be the email. Being that the DVDs rental company wants to give recognition where it is deserved, they asked us to update the email and append the word "top. ", to the email of their top salesperson. This would allow them to have top salesperson benefits such as extra discounts on DVDs and merchandises. This would also allow other team members to know who the very best salesperson is.
5.
    a. Detail Section: This will give us a deeper look into the top staff members in our company. It will provide the staff's information as well as what store they work at. Their phone numbers will also be provided so the executives could call and personally congratulate them. In addition, the total amount of how much they sold will also be provided.
    b. Summary Section: This will give us a broader look at how each store is doing. The information it will provide is how much income did the store make in total. This will allow us to keep track and come up with improvements for the stores that are at the bottom of the pile.
6. For the report to stay relevant and up to date it should be refreshed every month and not every time a new rental order is placed. Although, when a new rental is made, the total amount of DVD rented from each individual stores will increase making the old report outdated, having the report refreshed and updated every time a new rental is placed is over the top and too frequent. I believe a better approach to gage the staff's performance would be to update the report every month.

B. Write SQL code that creates the table that holds your report sections.

```
CREATE TABLE DetailedStoreReport (
  staff_id integer,
  store_id integer,
  first_name VARCHAR,
  last_name VARCHAR,
  phone_number VARCHAR,
  total_sale DECIMAL
)

CREATE TABLE StoreSummaryReport (
  store_id integer,
  store_address VARCHAR,
  total_sales DECIMAL
)
```

C. Write a SQL query that will extract the raw data needed for the Detailed section of your report from the source database and verify the data's accuracy.

```
DETAILED TABLE
DROP TABLE IF EXISTS DetailedStoreReport;

CREATE TABLE DetailedStoreReport (
  staff_id integer,
store_id integer,
  first_name VARCHAR,
  last_name VARCHAR,
  phone_number VARCHAR,
  total_sale DECIMAL
```

```
);


INSERT INTO DetailedStoreReport  (SELECT
STAFF. Staff_id,
STORE.store_id,
first_name,
last_name,
phone AS phone_num,
SUM(amount) AS total_sale
FROM STAFF
INNER JOIN STORE
        ON STAFF.store_id = STORE.store_id
INNER JOIN ADDRESS
        ON STORE.address_id = ADDRESS.address_id
INNER JOIN PAYMENT
        ON PAYMENT.staff_id = STAFF.staff_id
GROUP BY (STAFF.staff_id, STORE.store_id, ADDRESS.address_id)
ORDER BY total_sale DESC );
```

SUMMARY TABLE

```
DROP TABLE IF EXISTS StoreSummaryReport;

CREATE TABLE StoreSummaryReport (
        store_id integer,
        total_sales DECIMAL,
         store_address VARCHAR
```

```
);


INSERT INTO StoreSummaryReport (
        SELECT
                STORE.store_id,
                ADDRESS.address,
                SUM(amount) AS total_sales
        FROM STORE
                INNER JOIN STAFF
                        ON STORE.store_id = STAFF.store_id
                INNER JOIN ADDRESS
                        ON STORE.address_id = ADDRESS.address_id
                INNER JOIN PAYMENT
                        ON PAYMENT.staff_id = STAFF.staff_id
        GROUP BY (STAFF.staff_id, STORE.store_id, ADDRESS.address_id)
        ORDER BY total_sales DESC
);
```

D. Write code for function(s) that perform the transformation(s) you identified in part A1.

```
CREATE OR REPLACE FUNCTION top_sales_person(top_person_id integer) RETURNS VOID AS
$$
 BEGIN
      UPDATE STAFF SET email = 'top_lead@gmail.com'
      WHERE staff_id = top_person_id;
END;
$$ LANGUAGE plpgsql;
```

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 OR REPLACE FUNCTION UPDATE_STORE_SUMMARY_REPORT()
 RETURNS TRIGGER
 LANGUAGE PLPGSQL
 AS
$$
BEGIN
        IF NEW.total_sale <> OLD.total_sale THEN
                UPDATE StoreSummaryReport
                        SET total_sales = (
                                SELECT SUM(total_sale)
                                FROM DetailedStoreReport WHERE store_id =  OLD.store_id
    ) WHERE store_id = OLD.store_id;
        END IF;
        RETURN NEW;
END;
$$
```

```
CREATE TRIGGER AFTER_SALE_TRIGGER
 AFTER UPDATE
 ON DetailedStoreReport
```

```
FOR EACH ROW
EXECUTE PROCEDURE UPDATE_STORE_SUMMARY_REPORT();
```

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 refreshTable()
LANGUAGE SQL
AS $$
        DROP TABLE IF EXISTS DetailedStoreReport;

        CREATE TABLE DetailedStoreReport (
          staff_id integer,
        store_id integer,
          first_name VARCHAR,
          last_name VARCHAR,
          phone_number VARCHAR,
          total_sale DECIMAL
        );

        INSERT INTO DetailedStoreReport  (SELECT
        STAFF. Staff_id,
        STORE.store_id,
        first_name,
```

```
last_name,
phone AS phone_num,
SUM(amount) AS total_sale
FROM STAFF
INNER JOIN STORE
        ON STAFF.store_id = STORE.store_id
INNER JOIN ADDRESS
        ON STORE.address_id = ADDRESS.address_id
INNER JOIN PAYMENT
        ON PAYMENT.staff_id = STAFF.staff_id
GROUP BY (STAFF.staff_id, STORE.store_id, ADDRESS.address_id)
ORDER BY total_sale DESC );



DROP TABLE IF EXISTS StoreSummaryReport;

CREATE TABLE StoreSummaryReport (
                store_id integer,
                total_sales DECIMAL,
                 store_address VARCHAR
);


INSERT INTO StoreSummaryReport (
SELECT
STORE.store_id,
ADDRESS.address,
SUM(amount) AS total_sale,
```

```
        FROM STORE
        INNER JOIN STAFF
                                ON STORE.store_id = STAFF.store_id
        INNER JOIN ADDRESS
                                ON STORE.address_id = ADDRESS.address_id
        INNER JOIN PAYMENT
                                ON PAYMENT.staff_id = STAFF.staff_id
        GROUP BY (STAFF.staff_id, STORE.store_id, ADDRESS.address_id)
        ORDER BY total_sale DESC
        );
$$;
```

1. This store procedure should be running every month to check out how each individual staff member is doing, as well as the health and profitability of each individual store. It should not be running any earlier or later to get the most accurate information. Running the report more frequently will cause wasted memory usage in our S.Q.L server.
2. The store procedure just basically drops the **DetailedStoreSummary** & **StoreSummaryReport** if it exists, as well as its data, and recreate the table using the provided query. To refresh this store procedure and run it every month you will need to also create a trigger that goes a long with it.