Summarized Business Report

By using the Data provided It can be used to knowing how much profit a store generates in a certain region which can be used to capitalize further in those locations and to pinpoint stores chains in other areas that produces less income so that their productivity can be improved. The Data used in this report would be the store ID for identification, The city and country for location information, the amount that each transaction add together will be the generated profit. The tables include will be the Store, City, Country, Address, Staff, and Payment.

The Detailed table will include:

- Store Id
- Country
- City
- Amount
- Manager Staff Id

While The Summary Table will include:

- Country
- City
- Amount

A Transformation I will include will be to concat the names in country and city together and rename the field to Exact Location in the summary table and to sum of all Payment amounts to give the total Income generated per country.

The detailed report would be good for extensive information concerning not only profit but, the Country, City, Store, and the Manager employed, while the summary will just be an overview of profit by countries and cities for a overview understanding without specific details.

The report should be updated every business quarter for an accurate estimation of the actual profit turnover and order to avoid to short of a time frame which produces high fluctuations of inconsistent data so the stockholder can have data to accurately make decisions based off on.

The stored procedure should be used whenever necessary data is needed to be updated after three months.

--For Detailed Table check Exist to Drop, then creates

```
DROP TABLE IF EXISTS detailed;

CREATE TABLE detailed (
city VARCHAR(50),
country VARCHAR(50),
amount INT,
store_id INT
);

--For Summary table check Exist to Drop, then creates
DROP TABLE IF EXISTS Summary;
CREATE TABLE Summary(
Exact_Location VARCHAR(100),
amount INT
);

--Insert Into statement for Detailed Table
INSERT INTO Detailed(city, country, amount, store_id)
SELECT city, country, amount, store_id
From city c1
INNER JOIN country AS c2 ON c2.country_id = c1.country_id
INNER JOIN address AS a1 ON a1.city_id = c1.city_id
INNER JOIN staff AS s1 ON s1.address_id = a1.address_id
Inner JOIN payment AS p1 ON p1.staff_id = s1.staff_id;

--Functions to Concat and combine the city, and country field in the Summary Table
CREATE FUNCTION city_country ()
RETURNS TRIGGER
LANGUAGE plpgsql AS
```

```
$func$
BEGIN
DELETE FROM Summary;
INSERT INTO Summary(
SELECT concat_ws(',',city,country) AS Exact_Location, SUM(detailed.amount)
FROM Detailed
GROUP BY Exact_Location
);
RETURN NEW;
END
$func$;



--Trigger to respond to Detailed Update
CREATE TRIGGER summary_update
AFTER INSERT ON Detailed
FOR EACH STATEMENT
EXECUTE PROCEDURE city_country ();


CREATE PROCEDURE tables_refresh()
LANGUAGE plpgsql AS
$proc$
BEGIN
DELETE FROM Detailed;
INSERT INTO Detailed(city, country, amount, store_id)
SELECT city, country, amount, store_id
From city c1
```

```
INNER JOIN country AS c2 ON c2.country_id = c1.country_id

INNER JOIN address AS a1 ON a1.city_id = c1.city_id

INNER JOIN staff AS s1 ON s1.address_id = a1.address_id

Inner JOIN payment AS p1 ON p1.staff_id = s1.staff_id;

END

$proc$;
```

--By using a scheduler, you can set the stored procedure to run once every three months. By using the pgAgent tool or the pg_cron command you could make a job that would execute the procedure on the chosen day and time.

No Web Sources or any other third-party code was used.