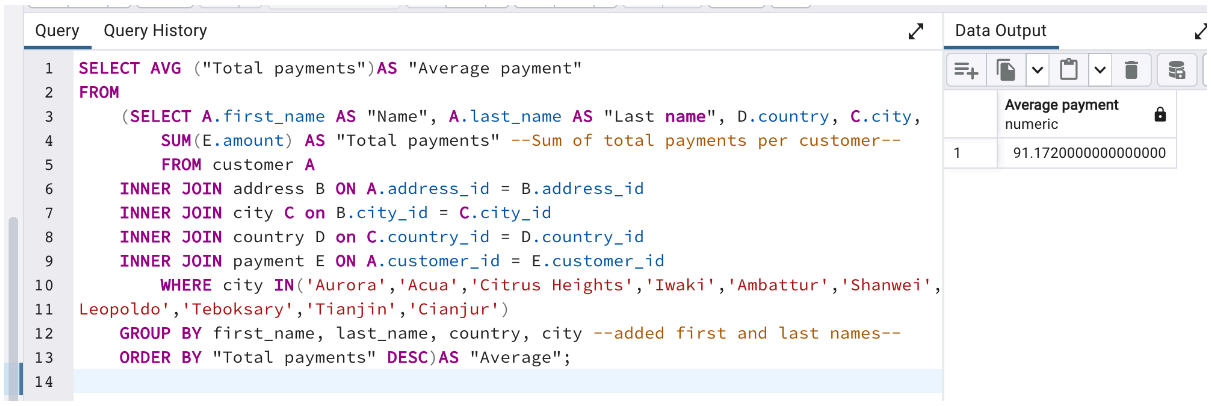
E.Christian Cotterman
Friday, 2. June 2023

Ex. 3.8

Performing Subqueries

- 1: Using the previous lesson, I used the **SELECT AVG** function, renamed it as 'Average Payment' **FROM** (subquery) **AS** "Average"



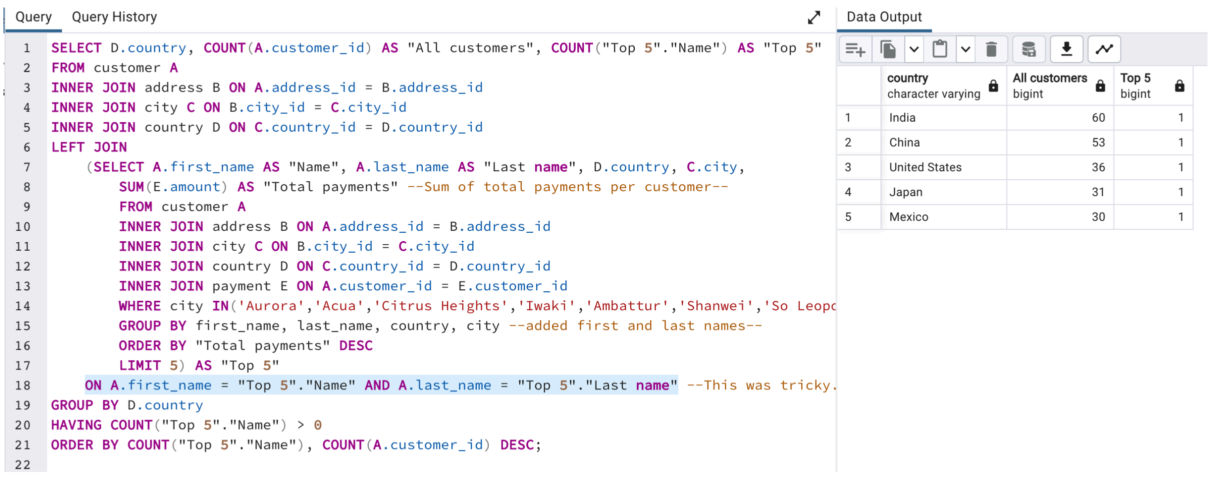
Query | Query History

```
1   SELECT AVG ("Total payments")AS "Average payment"
2   FROM
3       (SELECT A.first_name AS "Name", A.last_name AS "Last name", D.country, C.city,
4           SUM(E.amount) AS "Total payments" --Sum of total payments per customer--
5           FROM customer A
6       INNER JOIN address B ON A.address_id = B.address_id
7       INNER JOIN city C on B.city_id = C.city_id
8       INNER JOIN country D on C.country_id = D.country_id
9       INNER JOIN payment E ON A.customer_id = E.customer_id
10          WHERE city IN('Aurora','Acua','Citrus Heights','Iwaki','Ambattur','Shanwei',
11  Leopoldo','Teboksary','Tianjin','Cianjur')
12      GROUP BY first_name, last_name, country, city --added first and last names--
13      ORDER BY "Total payments" DESC)AS "Average";
14
```

Data Output

	Average payment numeric
1	91.1720000000000000

2: This was tricky for me. I renamed some tables for the query (*see line 1*) which was giving me an error after the **JOIN** statement. After researching, I found I needed to use **ON A**.first_name = "Top 5"."Name" **AND A**.last_name = "Top 5"."Last name" to correct it, due to my renaming. I don't think I will do that again. ***Names_will_stay_this_way_forever_now.***



Query | Query History

```
1   SELECT D.country, COUNT(A.customer_id) AS "All customers", COUNT("Top 5"."Name") AS "Top 5"
2   FROM customer A
3   INNER JOIN address B ON A.address_id = B.address_id
4   INNER JOIN city C ON B.city_id = C.city_id
5   INNER JOIN country D ON C.country_id = D.country_id
6   LEFT JOIN
7       (SELECT A.first_name AS "Name", A.last_name AS "Last name", D.country, C.city,
8           SUM(E.amount) AS "Total payments" --Sum of total payments per customer--
9           FROM customer A
10          INNER JOIN address B ON A.address_id = B.address_id
11          INNER JOIN city C ON B.city_id = C.city_id
12          INNER JOIN country D ON C.country_id = D.country_id
13          INNER JOIN payment E ON A.customer_id = E.customer_id
14          WHERE city IN('Aurora','Acua','Citrus Heights','Iwaki','Ambattur','Shanwei','So Leopo
15          GROUP BY first_name, last_name, country, city --added first and last names--
16          ORDER BY "Total payments" DESC
17          LIMIT 5) AS "Top 5"
18      ON A.first_name = "Top 5"."Name" AND A.last_name = "Top 5"."Last name" --This was tricky.
19  GROUP BY D.country
20  HAVING COUNT("Top 5"."Name") > 0
21  ORDER BY COUNT("Top 5"."Name"), COUNT(A.customer_id) DESC;
22
```

Data Output

	country character varying	All customers bigint	Top 5 bigint
1	India	60	1
2	China	53	1
3	United States	36	1
4	Japan	31	1
5	Mexico	30	1

- 3: With databases being updated often, if not continuously, the **JOIN** is relevant to finding information that is needed, but not necessarily on-the-minute accurate. **JOIN** can be used to find more specific data over multiple databases/tables, so it is useful for '*Deep Dives*' into that information. If you wanted top 10 cities and the average payments; If you are looking for top rentals per city to compare with top actors also.
  I think both of these are better or more accurate answers to the question with **JOIN** statements. If you simply did a top results, you may get the same answers, but refining them with **JOIN** assures more accurate data.

  Here is the query for #2 in full. ***Sorry about the lack of formatting.***

```
SELECT D.country, COUNT(A.customer_id) AS "All customers", COUNT("Top 5"."Name") AS
"Top 5"
FROM customer A
INNER JOIN address B ON A.address_id = B.address_id
INNER JOIN city C ON B.city_id = C.city_id
INNER JOIN country D ON C.country_id = D.country_id
LEFT JOIN
   (SELECT A.first_name AS "Name", A.last_name AS "Last name", D.country, C.city,
      SUM(E.amount) AS "Total payments" --Sum of total payments per customer--       FROM
customer A
      INNER JOIN address B ON A.address_id = B.address_id
      INNER JOIN city C ON B.city_id = C.city_id
      INNER JOIN country D ON C.country_id = D.country_id
      INNER JOIN payment E ON A.customer_id = E.customer_id
      WHERE city IN('Aurora','Acua','Citrus Heights','Iwaki','Ambattur','Shanwei','So
Leopoldo','Teboksary','Tianjin','Cianjur')       GROUP BY first_name, last_name, country, city --
added first and last names--       ORDER BY "Total payments" DESC
      LIMIT 5) AS "Top 5"
   ON A.first_name = "Top 5"."Name" AND A.last_name = "Top 5"."Last name" --This was tricky.
I made aliases and then had to reference the aliases with other aliases. I don't think I will do this
again--
GROUP BY D.country
HAVING COUNT("Top 5"."Name") > 0
ORDER BY COUNT("Top 5"."Name"), COUNT(A.customer_id) DESC;
```