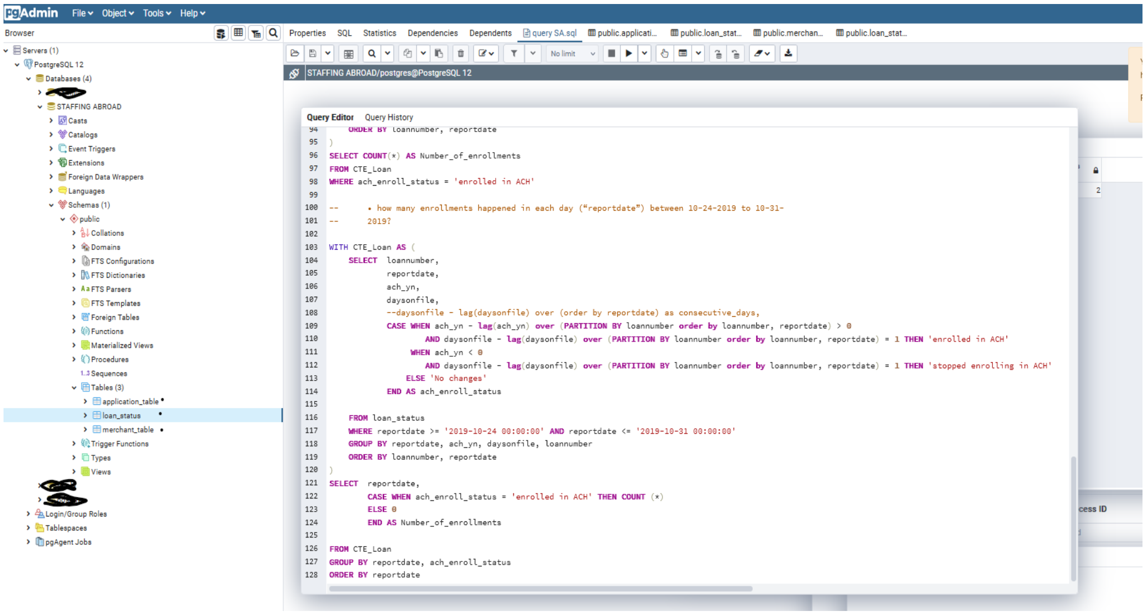
Michael Cabas

Data Analyst | Data Scientist | Consultant | Python | SQL | PowerBI
Medellín, COL
+57 3013748109
mcabas.ingbio@gmail.com
www.linkedin.com/in/michael-cabas
https://github.com/mcabas

```
-- Test: STAFFING ABROAD | SQL Data Analyst process technical questions
```

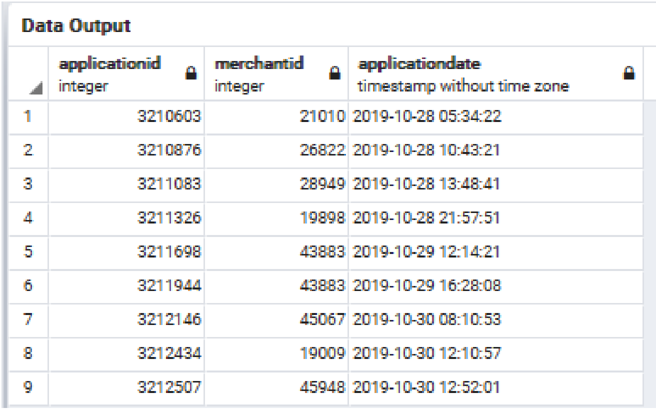


```
    ORDER BY loannumber, reportdate
)
SELECT COUNT(*) AS Number_of_enrollments
FROM CTE_Loan
WHERE ach_enroll_status = 'enrolled in ACH'

--      • how many enrollments happened in each day ("reportdate") between 10-24-2019 to 10-31-
--      2019?

WITH CTE_Loan AS (
    SELECT  loannumber,
            reportdate,
            ach_yn,
            daysonfile,
            --daysonfile - lag(daysonfile) over (order by reportdate) as consecutive_days,
            CASE WHEN ach_yn - lag(ach_yn) over (PARTITION BY loannumber order by loannumber, reportdate) > 0
                    AND daysonfile - lag(daysonfile) over (PARTITION BY loannumber order by loannumber, reportdate) = 1 THEN 'enrolled in ACH'
                 WHEN ach_yn < 0
                    AND daysonfile - lag(daysonfile) over (PARTITION BY loannumber order by loannumber, reportdate) = 1 THEN 'stopped enrolling in ACH'
                ELSE 'No changes'
            END AS ach_enroll_status

    FROM loan_status
    WHERE reportdate >= '2019-10-24 00:00:00' AND reportdate <= '2019-10-31 00:00:00'
    GROUP BY reportdate, ach_yn, daysonfile, loannumber
    ORDER BY loannumber, reportdate
)
SELECT  reportdate,
        CASE WHEN ach_enroll_status = 'enrolled in ACH' THEN COUNT (*)
        ELSE 0
        END AS Number_of_enrollments

FROM CTE_Loan
GROUP BY reportdate, ach_enroll_status
ORDER BY reportdate
```

Img 1. The exercise was performed on pgAdmind 4 using PostgreSQL

```
-- Q1
-- 1. Create tables

--create table application table and merchant table
CREATE TABLE application_table
(applicationid integer,
merchantid integer,
applicationdate timestamp
);


CREATE TABLE merchant_table
```

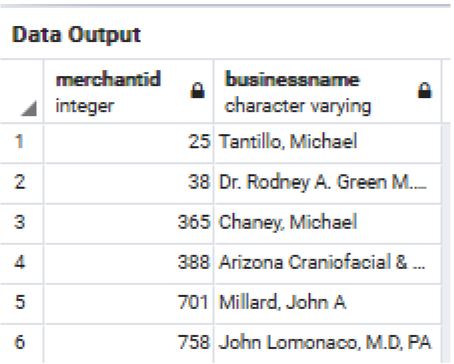
```
(merchantid integer,
businessname varchar
);

-- 2. Import data into the tables
--Step 1. Refresh Database
--Step 2. Go to the application_table table and click to open import/export, in
options select import, and look for the application_table.csv file in downloads,
update the header as ‘yes’ and delimiter as ‘,’, then go to the columns tab and
validate that all columns are selected.
--Step 3. Click on OK.
--Step 4. Repeat the steps for the merchant_table.csv.
```



Data Output

	applicationid integer	merchantid integer	applicationdate timestamp without time zone
1	3210603	21010	2019-10-28 05:34:22
2	3210876	26822	2019-10-28 10:43:21
3	3211083	28949	2019-10-28 13:48:41
4	3211326	19898	2019-10-28 21:57:51
5	3211698	43883	2019-10-29 12:14:21
6	3211944	43883	2019-10-29 16:28:08
7	3212146	45067	2019-10-30 08:10:53
8	3212434	19009	2019-10-30 12:10:57
9	3212507	45948	2019-10-30 12:52:01

Img 2. application_table



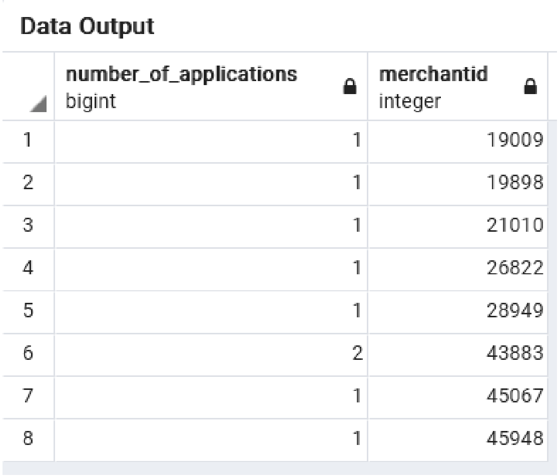
Data Output

	merchantid integer	businessname character varying
1	25	Tantillo, Michael
2	38	Dr. Rodney A. Green M....
3	365	Chaney, Michael
4	388	Arizona Craniofacial & ...
5	701	Millard, John A
6	758	John Lomonaco, M.D, PA

Img 3. merchant_table

```
-- 3. Question: Please construct SQL queries to answer the two questions below:
```

```
-- 3.1. How many applications were received from each merchant between 2019-10-28
and 2019-10-31?


SELECT  COUNT(*) AS Number_of_applications,
      merchantid
FROM application_table
WHERE applicationdate > '2019-10-28 00:00:00' AND applicationdate < '2019-10-31
00:00:00'
GROUP BY merchantid
```

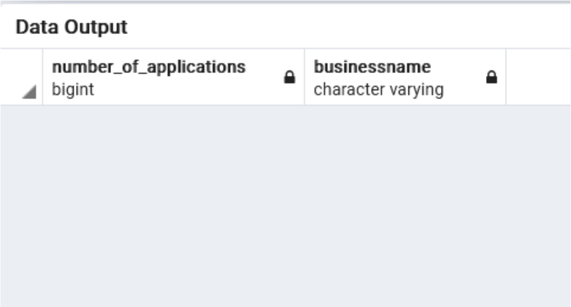


Data Output

	number_of_applications bigint	merchantid integer
1	1	19009
2	1	19898
3	1	21010
4	1	26822
5	1	28949
6	2	43883
7	1	45067
8	1	45948

Img 4. Output Q1 3.1.

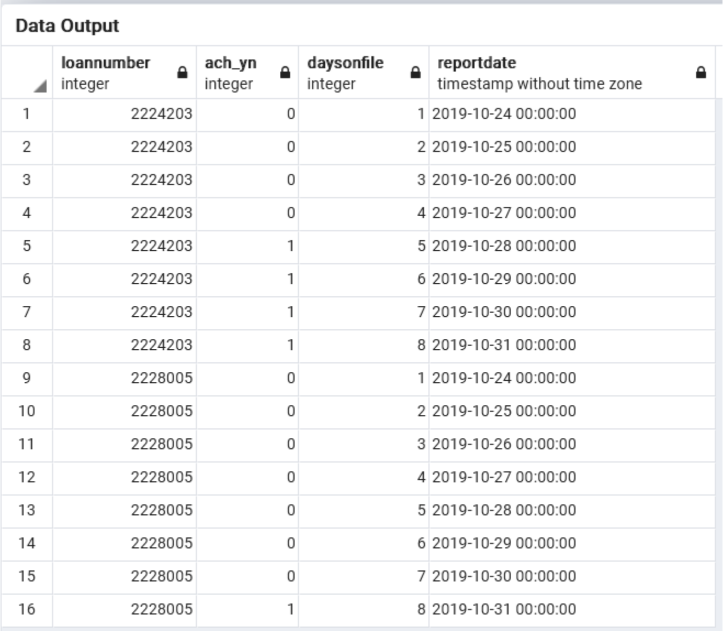
```
-- 3.2. What are the top three merchant names by applications received between
2019-10-28 and 2019-10-31?


SELECT  COUNT(*) AS Number_of_applications,
      me.businessname
FROM application_table AS ap
INNER JOIN merchant_table AS me ON ap.merchantid = me.merchantid
WHERE applicationdate > '2019-10-28 00:00:00' AND applicationdate < '2019-10-31
00:00:00'
GROUP BY me.businessname
ORDER BY Number_of_applications DESC
LIMIT 3
```



number_of_applications bigint	businessname character varying

Img5. Output Q1 3.2. Output is empty due it was found no relation between the merchant ID from the application table and the merchant ID from the merchant table in the dataset provided for the exercise. But, in the case of having a complete database, the query shown is accurate.

```
-- Q2
-- 1. Create Loan_status table

CREATE TABLE loan_status
(loannumber integer,
ach_yn integer,
daysonfile integer,
reportdate timestamp
);

-- 2. import data into the tables
--Step 1. Refresh Database
--Step 2. Go to the loan_status table and click to open import/export, in options
select import, and look for the loan_status.csv file in downloads, update the
header as ‘yes’ and delimiter as ‘,’, then go to the columns tab and validate
that all columns are selected.
--Step 3. Click on OK.
```



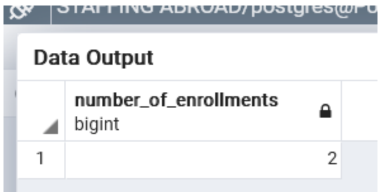
Data Output

	loannumber	ach_yn	daysonfile	reportdate
	integer	integer	integer	timestamp without time zone
1	2224203	0	1	2019-10-24 00:00:00
2	2224203	0	2	2019-10-25 00:00:00
3	2224203	0	3	2019-10-26 00:00:00
4	2224203	0	4	2019-10-27 00:00:00
5	2224203	1	5	2019-10-28 00:00:00
6	2224203	1	6	2019-10-29 00:00:00
7	2224203	1	7	2019-10-30 00:00:00
8	2224203	1	8	2019-10-31 00:00:00
9	2228005	0	1	2019-10-24 00:00:00
10	2228005	0	2	2019-10-25 00:00:00
11	2228005	0	3	2019-10-26 00:00:00
12	2228005	0	4	2019-10-27 00:00:00
13	2228005	0	5	2019-10-28 00:00:00
14	2228005	0	6	2019-10-29 00:00:00
15	2228005	0	7	2019-10-30 00:00:00
16	2228005	1	8	2019-10-31 00:00:00

Img5. loan_status table

```
-- Here is a table containing ACH status of loans from 10-24-2019 to 10-31-2019.
“ach_yn” is the flag for
-- the daily ACH status of a loan (1 = Yes, 0 = No). “ach_yn” = 1 means the
borrower has an active
-- automatic payment on that day.



--  • When “ach_yn” changes from 0 to 1 between two consecutive days, the loan
has enrolled in
--    ACH (enrollment).
--  • When “ach_yn” changes from 1 to 0 between two consecutive days, the loan
has stopped
--    enrolling in ACH.

--  Question: Please construct SQL queries to answer the two questions below:
```

```
--    • how many enrollments happened within this time period (10-24-2019 to
10-31-2019)?

WITH CTE_Loan AS (
      SELECT  loannumber,
            reportdate,
            ach_yn,
            daysonfile,
            CASE WHEN ach_yn - lag(ach_yn) over (PARTITION BY loannumber order
by loannumber, reportdate) > 0
                  AND daysonfile - lag(daysonfile) over (PARTITION BY loannumber
order by loannumber, reportdate) = 1 THEN 'enrolled in ACH'
                  WHEN ach_yn < 0
                  AND daysonfile - lag(daysonfile) over (PARTITION BY loannumber
order by loannumber, reportdate) = 1 THEN 'stopped enrolling in ACH'
            ELSE 'No changes'
            END AS ach_enroll_status

      FROM loan_status
      WHERE reportdate >= '2019-10-24 00:00:00' AND reportdate <= '2019-10-31
00:00:00'
      GROUP BY reportdate, ach_yn, daysonfile, loannumber
      ORDER BY loannumber, reportdate
)
SELECT COUNT(*) AS Number_of_enrollments
FROM CTE_Loan
WHERE ach_enroll_status = 'enrolled in ACH'
```

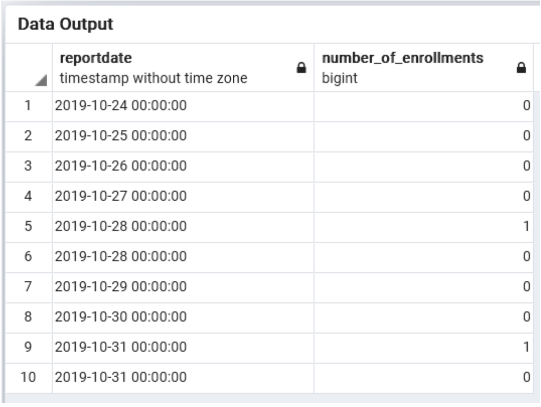


Data Output

	number_of_enrollments bigint
1	2

Img6. Q2 first output. A Common Table Expressions ´CTE_Loan´ is used to build the query.

```
--     • how many enrollments happened in each day (“reportdate”) between
10-24-2019 to 10-31-2019?
WITH CTE_Loan AS (
      SELECT  loannumber,
            reportdate,
            ach_yn,
            daysonfile,
            CASE WHEN ach_yn - lag(ach_yn) over (PARTITION BY loannumber order
by loannumber, reportdate) > 0
                  AND daysonfile - lag(daysonfile) over (PARTITION BY loannumber
order by loannumber, reportdate) = 1 THEN 'enrolled in ACH'
                  WHEN ach_yn < 0
                  AND daysonfile - lag(daysonfile) over (PARTITION BY loannumber
order by loannumber, reportdate) = 1 THEN 'stopped enrolling in ACH'
            ELSE 'No changes'
            END AS ach_enroll_status

      FROM loan_status
      WHERE reportdate >= '2019-10-24 00:00:00' AND reportdate <= '2019-10-31
00:00:00'
      GROUP BY reportdate, ach_yn, daysonfile, loannumber
      ORDER BY loannumber, reportdate
)
SELECT  reportdate,
      CASE WHEN ach_enroll_status = 'enrolled in ACH' THEN COUNT (*)
      ELSE 0
      END AS Number_of_enrollments
FROM CTE_Loan
GROUP BY reportdate, ach_enroll_status
ORDER BY reportdate
```



Data Output

	reportdate timestamp without time zone	number_of_enrollments bigint
1	2019-10-24 00:00:00	0
2	2019-10-25 00:00:00	0
3	2019-10-26 00:00:00	0
4	2019-10-27 00:00:00	0
5	2019-10-28 00:00:00	1
6	2019-10-28 00:00:00	0
7	2019-10-29 00:00:00	0
8	2019-10-30 00:00:00	0
9	2019-10-31 00:00:00	1
10	2019-10-31 00:00:00	0

Img7. Q2 second output.