

This project was actually a survey by **Obukofe Uririoghene**, he reached out to me on twitter and sent me the link to a survey he was conducting using SQL.

The first task was to create a normalized schema of 11 tables and import the dataset. For this, he provided the queries to run, along with the instructions to follow, which can be found [here](#).

After data validation, which is step 4 of the project, I started with the main business tasks.

This project involves a managing director of the Lagos Metropolitan BRT service, “Jane”, who is having a bit of trouble managing her database. And also needs help answering the following questions with data from her company's database.

The questions were as follows;

1. **\*\*Arrival Location Rankings\*\***. Jane noticed that some arrival locations stand out more than the others and she intends to find out where her passengers head to the most. She needs a table of the top 5 arrival locations sorted in descending order. Can you help her with that?
2. **\*\*Top Performing Drivers\*\***. Jane is keen on employee retention and she's planning on rewarding her top drivers. She needs to know the top 10 performing drivers based on the number of trips they've handled. For this, it would be best if we had a table containing their full names, their NIN's and the number of trips they've handled. Let's help Jane make that work!
3. **\*\*Frequent Passengers\*\***. Jane is a wonderful boss, and she loves her customers. She intends to reward her top 10 customers by offering 50% discounts on their next trips. The problem however is that she doesn't exactly know which passengers are her most frequent. She needs a table containing the passenger's full name, their email address and the number of trips they've taken sorted in descending order.
4. **\*\*Driver - Vehicle Pairings\*\***. Jane lost track of her drivers and their paired vehicles. She needs your help! She wants to know the current driver - vehicle pairings. To help her, let's create a table containing the full names of the driver and the buses they are paired to. She'd also like to see some details of the vehicle such as the plate number and the model.
5. **\*\*Highest Issue Dates\*\***. Jane has been impressed by your work so far but she needs one more thing. She wants to know the issue dates where her customers had the highest number of card issues for trips payment. She wants the list sorted by top 5 issue dates in descending order and also the number of cards issued. Let's help her with that.











I was also required to provide “open ended queries as you explore more on the database. The queries can be any query of your choice. Think of it as extra recommendations to Jane as she intends to find the answers to more questions in the future regarding her company.”

I started with the original business questions.

### --1. top 5 arrival locations

```
SELECT arrival_location, COUNT(*)
FROM brt.trips
GROUP BY arrival_location
ORDER BY COUNT(*) DESC
LIMIT 5;
```

\*\*For this, it was pretty straight forward, the data was in the trips table, so I used the COUNT function to gather the number of times each location appeared in the arrival\_location column, and ordered it in descending order of the counts, then limited it by 5, to only show the 5 highest values.

Data Output			Messages	Notifications
<div></div>				
	arrival_location character varying (20) 	count bigint 		
1	Abule Egba	41		
2	Inner Marina	41		
3	Aruna	41		
4	Ketu	40		
5	Ajah	38		

### --2. top performing drivers

```
ALTER TABLE brt.drivers
ADD COLUMN full_name text;
```

```
UPDATE brt.drivers
SET full_name = concat(first_name, ' ', last_name);
```












```
SELECT drivers.full_name,
       drivers.national_identity_number AS NIN,
       COUNT(trips.driver_id) AS total_trips
FROM brt.drivers AS drivers
INNER JOIN brt.trips AS trips
ON drivers.id = trips.driver_id
GROUP BY drivers.full_name, drivers.national_identity_number
ORDER BY total_trips DESC
```

LIMIT 10;

\*\*Here, the requirement was to select the top performing drivers' full names, NIN, and show the number of trips they have handled. We were asked to provide the drivers' full names, so I created a new column for their full names, and use the concat() function on the first and last name columns to update their full names.

The required data were in two different tables, so I used INNER JOIN to join the "trips" and "drivers" tables on the condition that the drivers' id and trips' driver\_id were the same.

I then selected the required columns, counting the "driver\_id" column for the trips as "total\_trips" to see how many times a driver went on a trip, then I ordered them by total trips in descending order and limited them by 10, to see only the top 10 performing drivers.

Data Output   Messages   Notifications			
<div>       </div>			
	full_name text 	nin character (11) 	total_trips bigint 
1	Babajide Olujimi	11180826634	81
2	Bukola Kyauta	36085056682	79
3	Umar Azubuike	27076178042	79
4	Aminu Chibueze	48327814351	78
5	Abimbola Jum...	62651357218	74
6	Ebere Bukar	30979759615	73
7	Chuks Ayuba	74086881815	72
8	Hadizatu Eniola	84977638233	71
9	Ola Chioma	15103683041	71
10	Hamisu Bolanle	68883882742	70

### --3. frequent passengers

```
ALTER TABLE brt.passengers  
ADD COLUMN full_name text;
```

```
UPDATE brt.passengers  
SET full_name = concat(first_name, ' ', last_name);
```

```
SELECT passengers.full_name, passengers.email_address,  
COUNT(passenger_trips.passenger_id) AS total_trips
```

```

FROM brt.passengers AS passengers
INNER JOIN brt.passenger_trips AS passenger_trips
ON passengers.id = passenger_trips.passenger_id
GROUP BY passengers.full_name, passengers.email_address
ORDER BY total_trips DESC
LIMIT 10;

```

\*\*In this question, the task was to select the full names and emails of the top 10 passengers, as in the previous question, I created a full name column for the passengers and updated it, then selected the information needed from both tables, using the COUNT() function again to select the top 10 appearing passenger IDs as total\_trips and ordered them by the top 10 in descending order.

Data Output   Messages   Notifications			
<div> <div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div> </div>			
	full_name text	email_address character varying (50)	total_trips bigint
1	Chibuike Afolayan	Chibuike.Afolayan@example.com	13
2	Chinyere Chikere	Chinyere.Chikere@example.com	12
3	Ekene Olujimi	Ekene.Olujimi@example.com	10
4	Udo Dayo	Udo.Dayo@example.com	10
5	Chidike Afolabi	Chidike.Afolabi@example.com	10
6	Gbemisola Uju	Gbemisola.Uju@example.com	9
7	Ogechukwu Yejide	Ogechukwu.Yejide@example.com	9
8	Alheri Abeni	Alheri.Abeni@example.com	9
9	Nnamdi Chizoba	Nnamdi.Chizoba@example.com	9
10	Ebele Chikelu	Ebele.Chikelu@example.com	8

```

--4.driver - vehicle pairings
SELECT drivers.full_name as driver_name,
logs.driver_id,
vehicles.id as vehicle_id,
vehicles.model, vehicles.plate_number
FROM brt.drivers as drivers
JOIN brt.driver_vehicle_logs AS logs
ON drivers.id = logs.driver_id
JOIN brt.vehicles as vehicles

```

ON logs.vehicle\_id = vehicles.id;

\*\*Here, the task was to find what drivers were paired to each vehicle, providing their full names, driver id, their vehicle id, the vehicle model, and its plate number.









The information I needed for this was in 3 different tables, so I used two JOIN statements to draw information from the “drivers”, “driver\_vehicle\_logs”, and the “vehicles” tables.

Data Output Messages Notifications					
	driver_name text	driver_id integer	vehicle_id integer	model character varying (50)	plate_number character varying (10)
1	Bukola Kyauta	1	3	Marcopolo Ideale	LSR-468BS
2	Umar Azubuike	2	15	Marcopolo Ideale	LSR-575PL
3	Chuks Ayuba	3	21	Marcopolo Scania	FST-941VR
4	Chinyelu Onyinyechi	4	22	Marcopolo Scania	FST-264BX
5	Chinwendu Yetunde	5	24	Primero	AAA-583LN
6	Hadizatu Eniola	6	25	Marcopolo Ideale	LSR-653GS
7	Ogochukwu Ife	7	26	Marcopolo Scania	FST-757VV
8	Hamisu Bolanle	8	27	Primero	AAA-454HO
9	Chidiebele Oluwayemisi	9	34	Primero	AAA-712UF
10	Aminu Chibueze	10	39	Marcopolo Ideale	LSR-556AL
11	Abimbola Jummai	11	43	Marcopolo Scania	FST-597RM
12	Ola Chioma	12	44	Primero	AAA-613MN
13	Ebere Bukar	13	58	Primero	AAA-106RO
14	Babajide Olujimi	14	64	Primero	AAA-118TH

--5.highest issue dates

```
SELECT DISTINCT (issue_date), COUNT (*) as cards_issued
FROM brt.passenger_cards
GROUP BY issue_date
ORDER BY COUNT (*) DESC
LIMIT 5
```

\*\*The task here was to select the dates where the most cards were issued, as well as the amount of cards issued on that date. The query here was similar to that of question 1; select issue\_date column and count the times each date appeared, order it by the count in descending order and limit it to the top 5 dates.









Data Output	Messages	Notifications
		
		
		
	issue_date date	cards_issued bigint
1	2022-05-19	5
2	2022-06-26	5
3	2022-05-17	3
4	2022-06-12	3
5	2022-02-11	3

That was it for the original business questions, next, I had to come up with further questions that could be useful for Jane to have further business insights for her company. After scanning the schema and the table contents, I came up with the following questions;

#### --1. no of vehicles in use, available, under maintenance and out of use

```
SELECT status, COUNT (*)
FROM brt.vehicles
GROUP BY status
```

\*\*This query selected the “status” column from the “vehicles” table, and counted how many times each status appeared.











Data Output	Messages	Notifications
		
		
		
	status character varying (20)	count bigint
1	In use	14
2	Available	17
3	Under maintenance	21
4	Out of use	18

#### --2. top 5 departure locations

```
SELECT departure_location, COUNT(*)
FROM brt.trips
GROUP BY departure_location
```

ORDER BY COUNT(\*) DESC  
LIMIT 5;

\*\*I ran a query similar to the arrival location one in the original business question, these are the results

Data Output		Messages	Notifications
       			
	departure_location character varying (20) 	count bigint 	
1	Alapere	41	
2	Onipanu	39	
3	Aruna	39	
4	Irawo	38	
5	Idera	38	

**--3. drivers with licenses to be renewed within one year (current date is feb 14, 2023)**

```
SELECT license.expiry_date, license.driver_id, drivers.full_name  
FROM brt.license as license  
JOIN brt.drivers as drivers  
ON license.driver_id = drivers.id  
WHERE license.expiry_date BETWEEN CURRENT_DATE AND CURRENT_DATE +  
INTERVAL '1 year';
```

\*\*For this, I used a JOIN statement to draw data from the “license” and “drivers” tables, to show drivers’ full names, IDs, and the expiry date.

I also used a WHERE statement to filter the expiry\_date column to select the dates between the current date(February 14th, 2023) and one year from that date, using the “+ INTERVAL ‘1 year’ clause”.

Data Output

Messages

Notifications

≡+

▼

	<div>expiry_date</div> <div>date</div> <div>🔒</div>	<div>driver_id</div> <div>integer</div> <div>🔒</div>	<div>full_name</div> <div>text</div> <div>🔒</div>
1	2024-01-06	1	Bukola Kyauta
2	2024-01-07	2	Umar Azubuike
3	2024-01-08	3	Chuks Ayuba
4	2024-01-09	4	Chinyelu Onyinyechi
5	2024-01-10	5	Chinwendu Yetunde
6	2024-01-11	6	Hadizatu Eniola
7	2024-01-12	7	Ogochukwu Ife

#### --4. drivers with the biggest buses (trip\_capacity)

##### --viewing highest trip capacity

```
SELECT trip_capacity
FROM brt.trips
ORDER BY trip_capacity DESC
```

\*\*I first ran this query to determine what the highest capacities were for the buses

Data Output

Messages

Notifications

≡+

📄

▼


📋

🗑️

🗄️

⬇️

📈

	trip_capacity 
1	55
2	55
3	55
4	55
5	55
6	55
7	55
8	55

##### --viewing drivers details



```

SELECT drivers.full_name as driver_name,
       drivers.id,
       trips.trip_capacity
FROM brt.drivers as drivers
JOIN brt.trips as trips
ON drivers.id = trips.driver_id
WHERE trip_capacity = 55

```

**\*\*Then I used this JOIN query to select the buses with the highest capacity, linking them to their drivers information.**

Data Output Messages Notifications			
	driver_name text	id integer	trip_capacity integer
1	Umar Azubuike	2	55
2	Ola Chioma	12	55
3	Chinyelu Onyinyechi	4	55
4	Chuks Ayuba	3	55
5	Hamisu Bolanle	8	55
6	Ogochukwu Ife	7	55
7	Hamisu Bolanle	8	55
8	Chidiebele Oluwayemisi	9	55
9	Chinyelu Onyinyechi	4	55
10	Aminu Chibueze	10	55
11	Aminu Chibueze	10	55
12	Umar Azubuike	2	55
13	Aminu Chibueze	10	55
14	Chidiebele Oluwayemisi	9	55
15	Babajide Olujimi	14	55
16	Bukola Kyauta	1	55
17	Ola Chioma	12	55
18	Umar Azubuike	2	55
19	Babajide Olujimi	14	55
20	Babajide Olujimi	14	55

### --5. 3 lowest performing drivers

```

SELECT drivers.full_name, drivers.email_address, COUNT(trips.driver_id) AS total_trips
FROM brt.drivers AS drivers
INNER JOIN brt.trips AS trips
ON drivers.id = trips.driver_id
GROUP BY drivers.full_name, drivers.email_address

```

```
ORDER BY total_trips
LIMIT 3;
```

\*\*This query was simply changing the order from descending to ascending for the query I ran in the original business question number 2, but I decided to limit it to the lowest 3 performing drivers instead, changing it to LIMIT 10. I also decided that their email addresses would be more useful information, to contact them and discuss ways to move forward with regards to their performances.

Data Output

Messages

Notifications

	<div>full_name</div> <div>text</div> <div></div>	<div>email_address</div> <div>character varying (50)</div> <div></div>	<div>total_trips</div> <div>bigint</div> <div></div>
1	Chinyelu Onyinyechi	Chinyelu.Onuauta@example.com	60
2	Chinwendu Yetunde	Chinwendu.Yeuauta@example.com	62
3	Ogochukwu Ife	Ogochukwu.Ifuaута@example.com	63