Data Management

Task 1

Nathan Hefner

A. Select one of the provided scenarios to complete the following:

For this task, I chose scenario 1, HealthFit Innovations, to help with their new platform, HealthTrack

A1. Describe a business problem that can be solved with a database solution and is in alignment with the chosen scenario

The business problem HealthFit Innovations is currently facing is the massive upload of data growing exponentially. Their current database needs the proper infrastructure to handle these data loads, which leads to more problems, such as bottlenecks and slow upload and processing times, along with needing to be more scalable, which only heightens their existing issues to new levels. Without something changing soon, the whole database will fail.

A2. Propose a data structure to solve the identified business problem.

A data structure I would implement to solve the problem is a Relational database backed by PostgreSQL for integrity and scalability. PostgreSQL is built to manage, store, and find large amounts of data quickly and create relationships between the data. To normalize the data to simplify and reduce redundancies, I will create a schema that will hold tiers of standard form. The first tier will keep all the data, the second will show data for specific values, and the third will ensure that the column data types depend on the table's unique identifier. An example of what the tables will be made up of is below.

Patients(Patient_ID, First_Name, Last_Name, Date_of_Birth, Gender, Last_Appointment_Date, Health_ID)

Health(Health_ID, Medical_conditions, Medications, Allergies, Tracker_ID)

Tracker(Tracker_ID, Tracker_Type, Brand_Name, Device_Type, Model_Name, Tracker_Specifications_ID)

Tracker_Specifications(Tracker_Specifications_ID, Color, Display, Strap_Material, Average_Battery_Life)

Sales(Sales_ID, Selling_Price, Original_Price, Rating, Reviews, Tracker_ID)

This relational database holds primary keys, which will act as unique identifiers for every patient and health tracking band to eliminate the possibility of repeats, thus removing the scalability problem. Foreign keys will also help link the data together.

A3. Justify why a database solution will solve the identified business problem.

This database solution will solve the scalability and infrastructure issues by cleaning and organizing the data efficiently and optimizing the workflow by reducing redundancies null values, and clearing errors. This will improve the system, allowing for more data inputs without lowering the output times and eliminating any bottlenecks.

A4. Explain how the business data will be used within the database solution.

The business data will be used to provide personal health inspections and real-time monitoring, predict possible health-related problems or issues, diagnose early symptoms, and better help manage one's health. Building a better database model will all be possible even with a more significant growing data input.

B. Create a logical data model for storing data in the database solution.

For this scenario, I created five tables holding data that were relevant to each. This method breaks down massive amounts of data into simpler, easier-to-read, smaller tables. Each of these has a primary and foreign key that connects each table to the other. Thus creating relationships between these tables.

C. Describe the database objects and storage, identifying the file attributes within the database solution.

The objects in the storage are entities, identifiers, or primary keys/foreign keys. The keys are the most important as they help improve scalability by always being unique for each row, thus never allowing an exact copy of multiple data points. The names and any kind of description are the 'VARCHAR' data type, meaning you can put a certain amount of characters to describe the selected data. Any data cell holding numbers as information, such as date, selling price,

reviews, or primary keys, are integer data types. Only allow numbers to be inputted. These restrictions help keep the data clean and consistent and make building reports easier.

D. Discuss how the proposed database design addresses scalability concerns, including strategies that align with the chosen scenario.

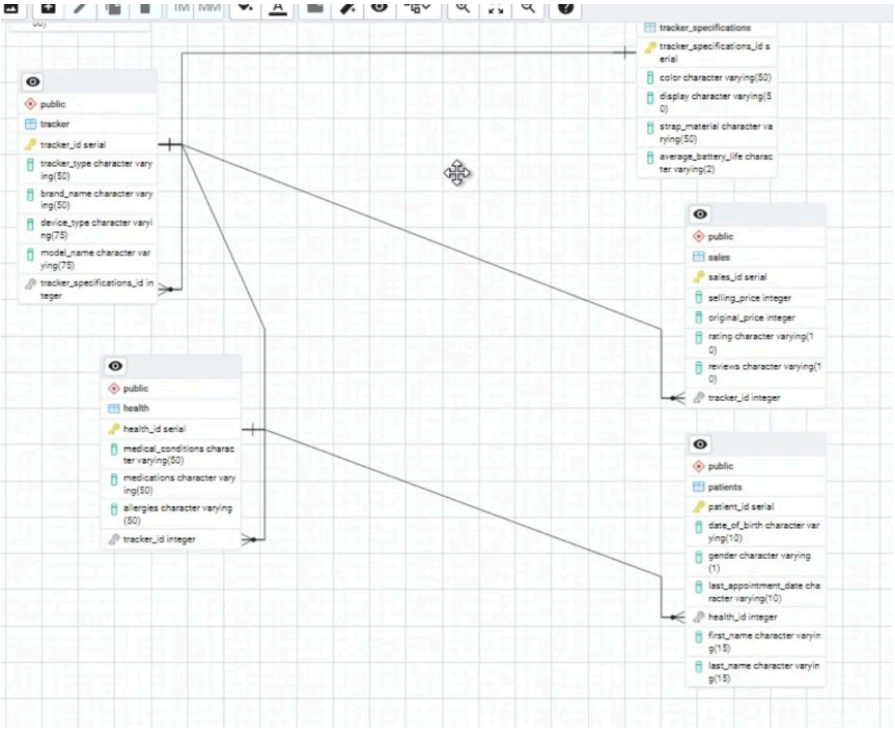
The proposed database design addresses scalability by including primary keys and foreign keys. These keys make every row of data unique, thus making it impossible to have copies of data points. This will then make creating reports more accessible and more consistent. The database is also separated into different categorical tables to keep the organization simple and easier to find, and they are all linked with foreign keys. Foreign keys are also unique in every row; they link tables together to reduce redundancy and clutter. This also optimizes workflow and creates faster generation times for reports. This will help the business as its data is cluttered, unorganized, and vulnerable to duplicates without unique identifiers.

E. Outline the privacy and security measures that should be implemented in the proposed database design.

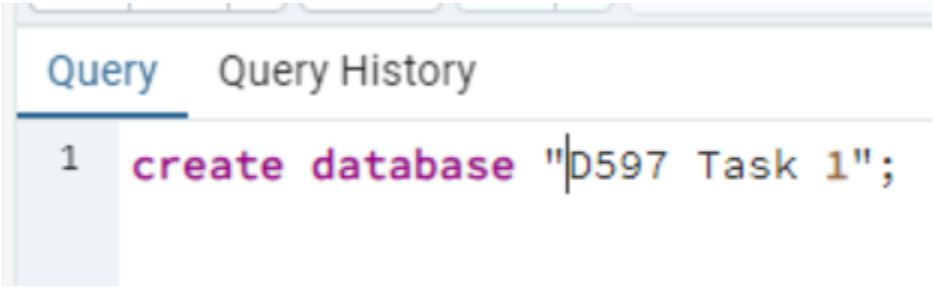
The privacy and security measures to be implemented into our database design would be multiple layers of protection. By implementing this layer of security, multi-factor authentication ensures that only verified users can log in and access the database. PGAdmin has a built-in setting that can be configured for this exact reason. Configure the authentication settings in config_local.py in server mode (*PgAdmin*). Our database will use cloud services as the company isn't big enough to have an in-house physical database server. Both options have pros and cons, but having it offsite and in the cloud increases security and reduces the risk of attack while creating backups in case of a major problem. The data will comply with regulations and rules across multiple nations, ensuring the data is safe and secure. Finally, the data will be encrypted by a 256-bit encryption key, helping keep it safe from brute-force hack attacks and stopping outside access. All these safeguards will ensure that the data is secure and protected.

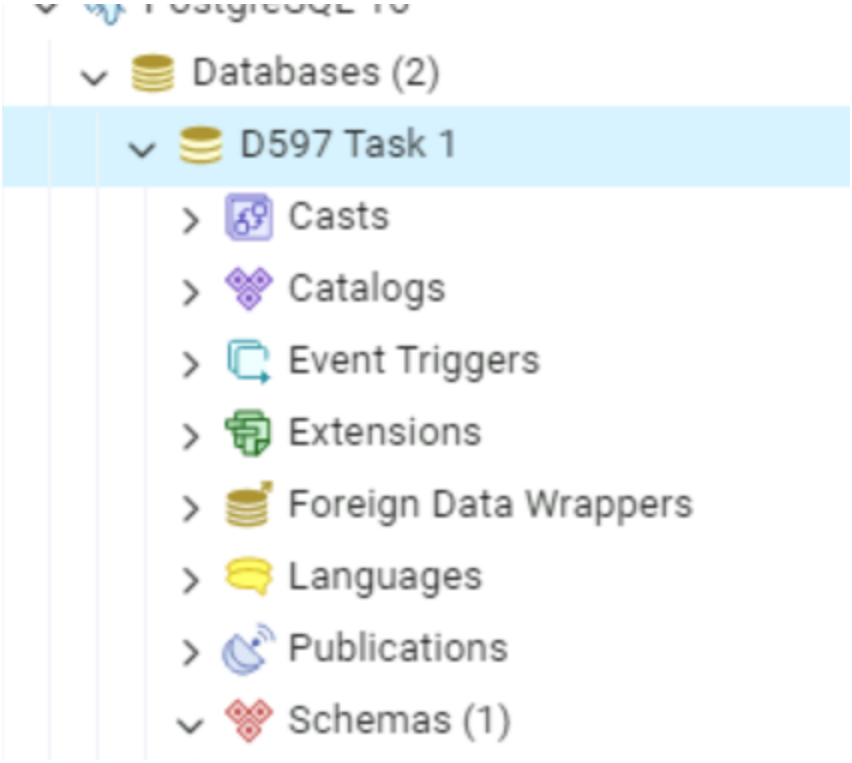
F. Implement the proposed database design in the WGU Virtual Lab environment by completing the following:

I implemented the proposed database design. Below are my screenshots of the database.



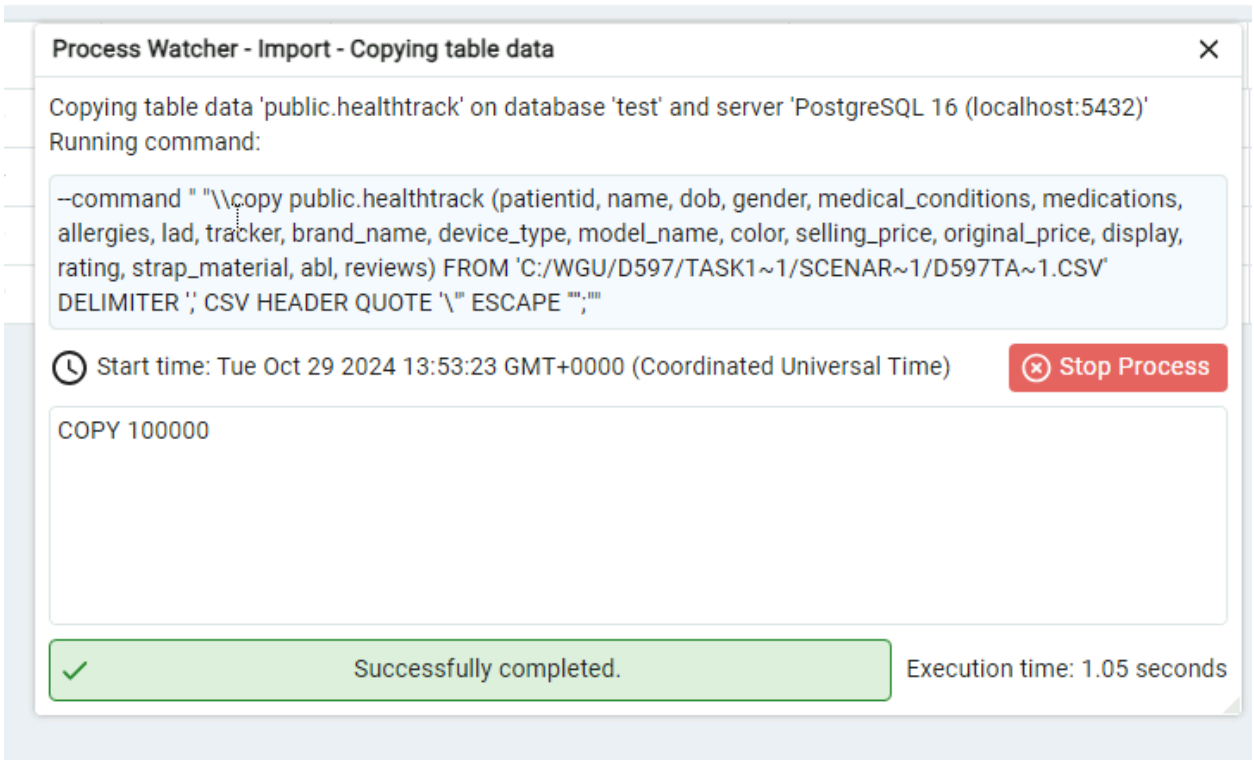
F1. Write a script to create a database instance named “D597 Task 1” using the appropriate query language based on the logical data model in part B. Provide a screenshot showing the script and the database instance in the platform.

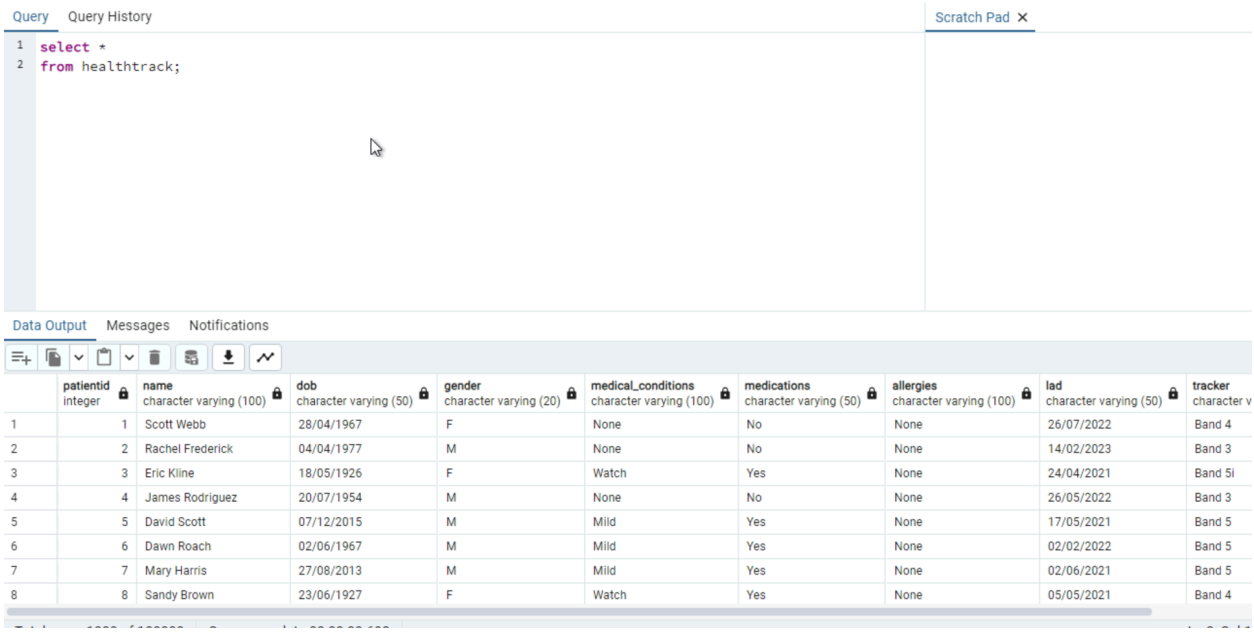




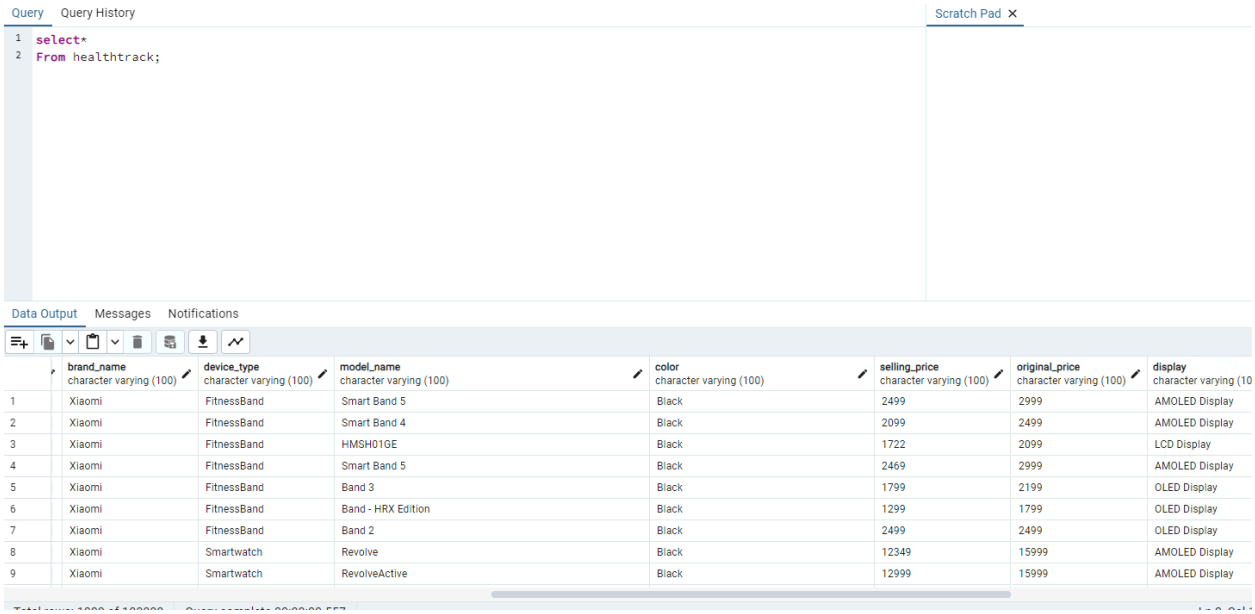
Here, I created the database for the exercise. This is where all our work will be done.

F2. Write a script to import the data records from the chosen scenario CSV files into the database instance. Provide a screenshot showing the script and the data correctly inserted or mapped into the database.





	patientid integer	name character varying (100)	dob character varying (50)	gender character varying (20)	medical_conditions character varying (100)	medications character varying (50)	allergies character varying (100)	lad character varying (50)	tracker character v
1	1	Scott Webb	28/04/1967	F	None	No	None	26/07/2022	Band 4
2	2	Rachel Frederick	04/04/1977	M	None	No	None	14/02/2023	Band 3
3	3	Eric Kline	18/05/1926	F	Watch	Yes	None	24/04/2021	Band 5i
4	4	James Rodriguez	20/07/1954	M	None	No	None	26/05/2022	Band 3
5	5	David Scott	07/12/2015	M	Mild	Yes	None	17/05/2021	Band 5
6	6	Dawn Roach	02/06/1967	M	Mild	Yes	None	02/02/2022	Band 5
7	7	Mary Harris	27/08/2013	M	Mild	Yes	None	02/06/2021	Band 5
8	8	Sandy Brown	23/06/1927	F	Watch	Yes	None	05/05/2021	Band 4



	brand_name character varying (100)	device_type character varying (100)	model_name character varying (100)	color character varying (100)	selling_price character varying (100)	original_price character varying (100)	display character varying (10
1	Xiaomi	FitnessBand	Smart Band 5	Black	2499	2999	AMOLED Display
2	Xiaomi	FitnessBand	Smart Band 4	Black	2099	2499	AMOLED Display
3	Xiaomi	FitnessBand	HMSH01GE	Black	1722	2099	LCD Display
4	Xiaomi	FitnessBand	Smart Band 5	Black	2469	2999	AMOLED Display
5	Xiaomi	FitnessBand	Band 3	Black	1799	2199	OLED Display
6	Xiaomi	FitnessBand	Band - HRX Edition	Black	1299	1799	OLED Display
7	Xiaomi	FitnessBand	Band 2	Black	2499	2499	OLED Display
8	Xiaomi	Smartwatch	Revolve	Black	12349	15999	AMOLED Display
9	Xiaomi	Smartwatch	RevolveActive	Black	12999	15999	AMOLED Display

After creating a “HealthTrack” table, I imported all the data from both cvs files. I then normalized the data by separating the data into smaller tables and creating primary and foreign keys to add unique identifiers to each table.

F3. Write a script for three queries to retrieve specific information from the database to help solve the identified business problem. Provide a screenshot showing the script for each query successfully executed.

My first query is ranking the top 10 lowest tracker band reviews, to see which bands aren’t performing as well or expected and to possibly stop producing them or improve their quality to improve business revenue.

```
SELECT sales_id, rating
FROM sales
WHERE rating IS NOT NULL
ORDER BY rating ASC
LIMIT 10;
```

	sales_id [PK] integer	rating character varying (10)
1	541	2
2	207	2.3
3	340	2.8
4	88	2.8
5	218	3
6	351	3
7	354	3
8	485	3
9	90	3.1
10	134	3.1

My second query was subtracting the original price from the selling price to see which band has the most net profit. Before this, though, I had to clean the data by removing commas and changing the data to integer type.

```
SELECT sales_id, (original_price - selling_price) AS price_difference
FROM sales
WHERE original_price IS NOT NULL AND selling_price IS NOT NULL
ORDER BY price_difference DESC
LIMIT 10;
```

	sales_id [PK] integer	price_difference integer
1	21	29693
2	483	20000
3	342	17596
4	192	17009
5	307	15996
6	219	15551
7	464	15500
8	504	15248
9	314	15196
10	527	15000

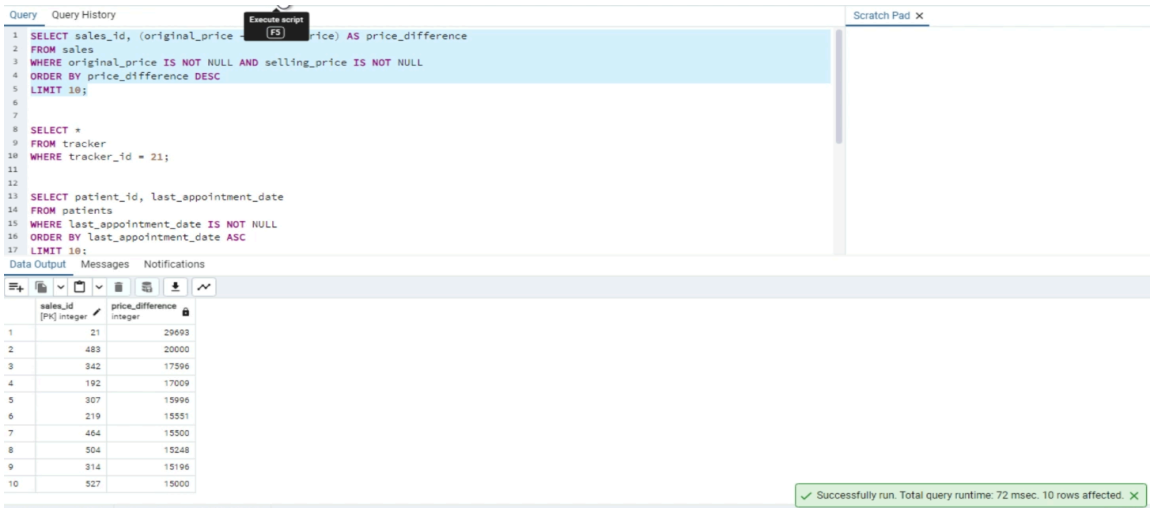
My third query is finding the top 10 latest last appointment dates to see who hasn't had an appointment in the longest time. This could be used to set up checks and/or trackers to check up on the patients who haven't had an appointment in a certain period to remind them that they will need another one soon.

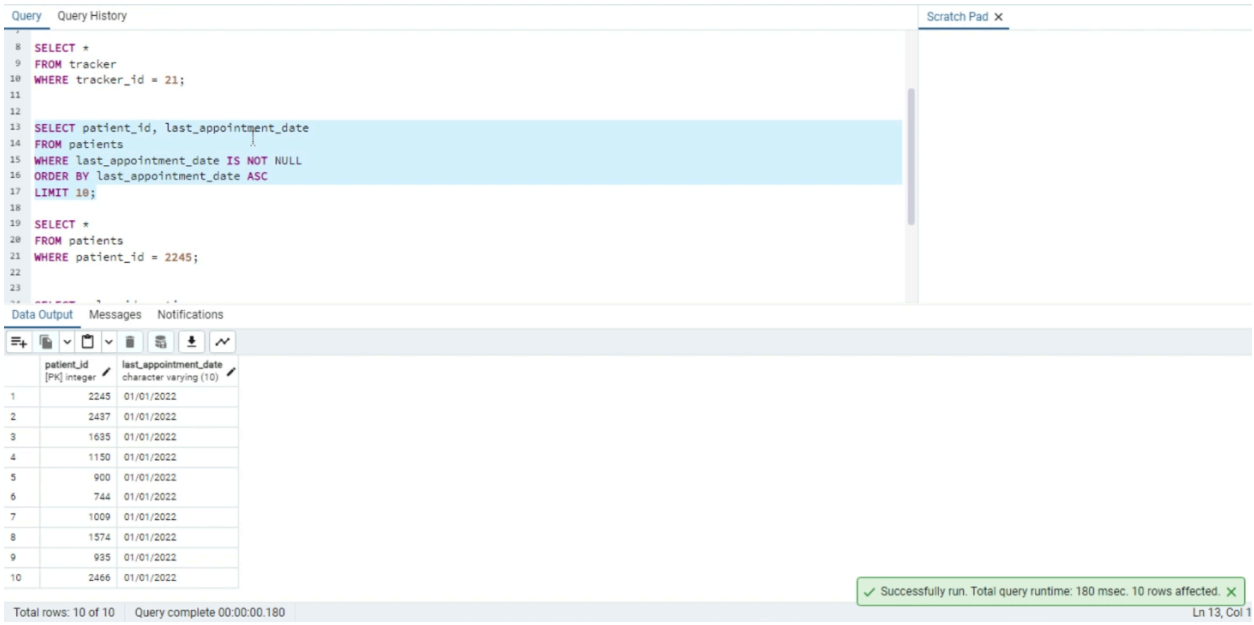
```
SELECT patient_id, last_appointment_date
FROM patients
WHERE last_appointment_date IS NOT NULL
ORDER BY last_appointment_date ASC
LIMIT 10;
```

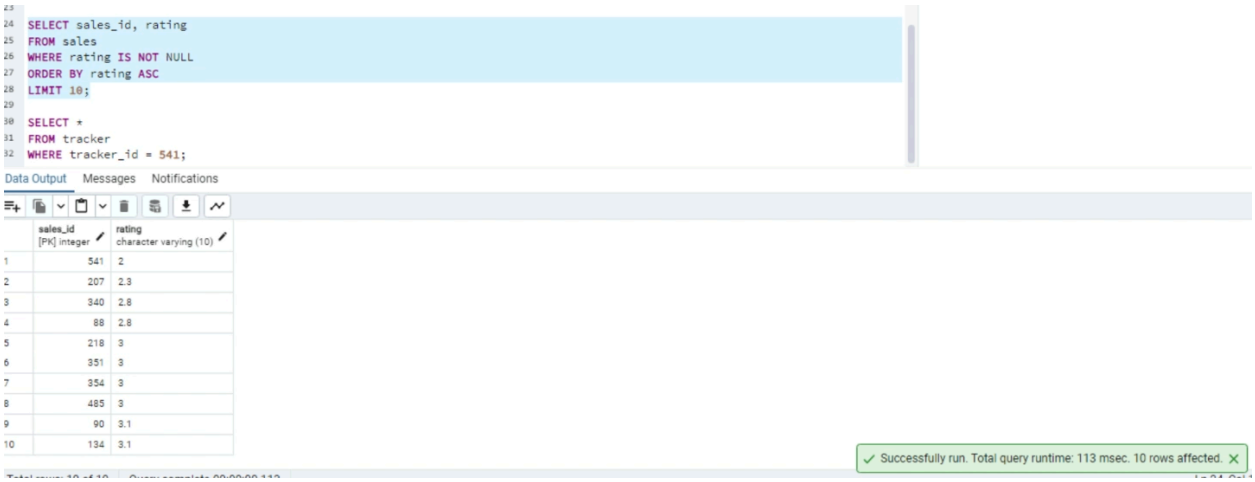
	patient_id [PK] integer	last_appointment_date character varying (10)
1	2245	01/01/2022
2	2437	01/01/2022
3	1635	01/01/2022
4	1150	01/01/2022
5	900	01/01/2022
6	744	01/01/2022
7	1009	01/01/2022
8	1574	01/01/2022
9	935	01/01/2022
10	2466	01/01/2022

F4. Apply optimization techniques to improve the run time of your queries from part F3, providing output results via a screenshot.

I apply optimization techniques by cleaning the data, changing the data objects to accurately reflect the data it is holding inside the columns, and changing the size of the data columns to also accurately reflect the data being held inside while simultaneously adding constraints to the data columns to keep the data inputs consistent.







By using the techniques above, simple queries complete on average in 120 milliseconds.

Sources

"PgAdmin." *pgAdmin*, www.pgadmin.org/.