Robert Pavlik

D597 – Data Management

MKN1 – Task 1: Relational Database Design and Implementation

PART 1: Design Documentation

A

1. HealthFit Innovations, through its platform HealthTrack, collects and processes vast amounts of diverse health-related data from multiple sources such as wearable devices, electronic health records (EHR), and patient-reported outcomes. The current database management system struggles with handling the increasing volume and variety of data, leading to several key challenges:
    - Scalability Issues: As more users adopt HealthTrack and input data from devices like fitness trackers, smartwatches, and medical devices, the volume of real-time data is growing exponentially. The current system is not capable of efficiently managing this data at scale.
    - Performance Bottlenecks: With the rising volume of data, the database faces slow query performance, affecting the delivery of real-time insights and predictive analytics. The system struggles to integrate and process data in real-time
    - Data Silos: HealthTrack aims to integrate multiple types of health data, but the current system leads to isolated data repositories, hindering a unified view of patient health.
    - Integration Challenges: The platform needs to integrate diverse data sources including wearables, medical imaging systems, and EHRs. The existing system faces difficulties in unifying and querying this data effectively.

    A database solution that efficiently integrates, stores, and scales with the growing data demands will solve the business problem. The proposed database should handle large volumes of structured and unstructured health-related data, ensuring performance and scalability. This new system will support HealthFit's objectives of offering personalized health insights, real-time monitoring, and predictive analytics for better patient care.

2. The current database is spread across two tables – Fitness_Trackers & Patients – we can improve the design by normalizing the database structure. This would help avoid redundancy, ensure scalability, and facilitate easier queries.
    - Eliminate Redundancy – Remove repetitive data entries.
    - Create Relationships – Establish clear relationships between tables using foreign keys
    - Normalize Data – Break the existing tables into smaller, well-structured tables
    - Maintain Data Integrity – Use appropriate data types and constraints to ensure data accuracy and consistency.

    Trackers Table (Former Fitness_Trackers Table)

    - This table will store information about different fitness trackers, but instead of having all the attributes in one table, some fields that can belong in their table will be separated

    Pricing Table

    - The pricing data, which includes the 'selling price' and 'original price' can be separated to avoid mixing financial data with device specifications.

    Reviews Table

    - Reviews should be in a separate table to avoid redundancy and allow for multiple reviews per tracker.

    Patients Table (Updated)

    - The Patients table will also benefit from normalization. Medical Conditions, Medications, and Allergies can be separated into their tables to facilitate easier querying.

    Patient_Trackers Table

    - Patients may use multiple trackers, so a table is needed to link patients to the devices they are using. This table will also include data such as the purchase date or assigned date.

    Medical_Conditions Table

    - This table will store medical condition data for patients, allowing for multiple conditions per patient.

Medications Table

- Medications can be stored in a separate table to accommodate changes in medication over time

Allergies

- A table to store allergy information for patients, allowing for multiple allergies per patient.

By breaking down the existing tables into smaller, focused tables, we make it easier to query the data and maintain it as the system scales. This normalization also avoids data redundancy, ensuring consistent and efficient data storage.

3. Implementing this database solution addresses HealthFit Innovations' business problem by providing a scalable, efficient, and secure system for managing and analyzing diverse health-related data. This solution ensures data integrity, supports real-time insights, and enhances overall operational efficiency, ultimately leading to improved patient care and better health management outcomes.

- Centralized Data Management – Integrating data from multiple sources, including wearables, electronic health records, and patient-reported outcomes. By centralizing this data, HealthTrack can offer a unified view of health information, eliminating data silos and improving accessibility. A well-designed database ensures data consistency and integrity, reducing discrepancies that might arise from having multiple different data sources.

- Scalability – The new database design supports scalability, allowing HealthTrack to handle the growing volume of health-related data from an expanding user base. The normalized structure and appropriate indexing will ensure that the database performs efficiently even as data volume increases. The database design can accommodate future additions of new data types or sources, ensuring that HealthTrack remains adaptable to evolving technological advancements and user needs.

- Improved Data Quality and Accuracy – By normalizing the data, the database reduces redundancy and minimizes errors. This ensures that health data is accurate and up to date, which is crucial for providing reliable health insights. Database constraints and data types help enforce data validation rules, ensuring that only valid and correctly formatted data is entered into the system.

- Efficient Data Retrieval and Analysis – The database structure allows for efficient querying and data retrieval. Indexes and optimized queries enable quick access to relevant information, which is essential for real-time monitoring and personalized health recommendations. With a well-organized database, HealthTrack can generate detailed reports and perform advanced analytics. This supports predictive analytics and helps healthcare providers make data-driven decisions.

- Enhanced Data Security and Privacy – The database solution can implement access control measures to restrict data access based on user roles. This ensures that sensitive health information is only accessible to authorized personnel. It can incorporate security measures to comply with data protection regulations such as HIPAA (Health Insurance Portability and Accountability Act), ensuring the privacy and security of patient data.

4. In the HealthTrack database solution, the business data will be utilized in several key ways to enhance health management and provide valuable insights.

- The Patient_Trackers table will link patients with their assigned fitness trackers and record the assignment dates. This helps track device usage and correlates it with health data.

- The Pricing table records the sales and original prices of fitness trackers. This data can be used for analyzing pricing trends, evaluating cost-effectiveness, and providing cost-related insights to users and healthcare providers.

- The Patient_Trackers table includes assignment dates for trackers. This helps in tracking when a patient started using a specific device, which can be useful when evaluating the impact of device usage over time.

- The Reviews table stores user reviews and ratings for different trackers. This information helps in assessing the performance and user satisfaction of the devices, which can be used to make recommendations and improve device offerings.

- The Medical_Conditions table records the medical conditions associated with patients. This data is essential for personalized health recommendations and identifying potential correlations between medical conditions and health metrics.

- The Allergies table captures information about patient allergies. This information is important for ensuring that health recommendations do not conflict with known allergies and for providing safe and effective health management strategies.

- By analyzing data from various tables, including Patients, Trackers, and Tracker_Usage, HealthTrack can generate personalized health insights, recommendations, and alerts based on users' health metrics and device usage.

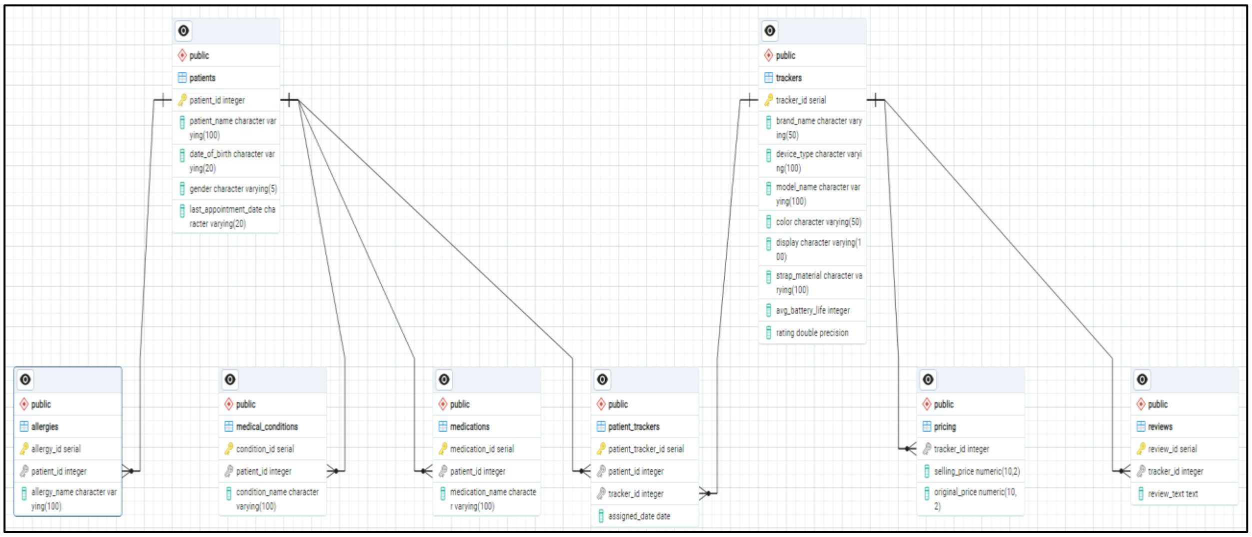
- The data from Pricing, Reviews, and Trackers can be used to identify trends in device performance, pricing changes, and user feedback, helping the company to make informed business decisions.

- The integrated data from Patients, Trackers, and Patient_Trackers provides a comprehensive view of each patient's health management and tracker usage. This supports more accurate and effective health management strategies.

- The database supports decision-making by providing access to relevant health data, enabling healthcare providers to make data-driven decisions and tailor their care approaches based on real-time insights.

The business data within the HealthTrack database solution will be used to manage patient profiles, monitor health metrics, analyze device performance and pricing, and provide personalized health recommendations. The integration and analysis of this data will enhance health management, improve patient care, and support informed decision-making.

B.



C.

- Patients Table
    - Patient_ID: INT, Primary Key
    - Patient_Name: VARCHAR (100)
    - Date_of_Birth: DATE
    - Gender: CHAR (1)
    - Last_Appointment_Date: DATE

- Trackers Table
    - Tracker_ID: INT, Primary Key
    - Brand_Name: VARCHAR (50)
    - Device_Type: VARCHAR (100)
    - Model_Name: VARCHAR (100)
    - Color: VARCHAR (100)
    - Display: VARCHAR (100)
    - Strap_Material: VARCHAR (100)
    - Avg_Battery_Life: INT
    - Rating: DOUBLE PRECISION

- Pricing Table
    - Tracker_ID: INT, FOREIGN KEY REFERENCING Trackers (Tracker_ID)
    - Selling_Price: DECIMAL (10,2)
    - Original_Price: DECIMAL (10,2)

- Reviews Table
    - Tracker_ID: INT, FOREIGN KEY REFERENCING Trackers (Tracker_ID)
    - Review_ID: INT, PRIMARY KEY
    - Review_Text: TEXT

- Patient_Trackers Table
    - Patient_ID: INT, FOREIGN KEY REFERENCING Patients (Patient_ID)
    - Tracker_ID: INT, FOREIGN KEY REFERENCING Trackers (Tracker_ID)
    - Assigned_Date: DATE

- Medical_Conditions Table
    - Condition_ID: INT, PRIMARY KEY
    - Patient_ID: INT, FOREIGN KEY REFERENCING Patients (Patient_ID)
    - Condition: VARCHAR (100)

- Medications Table
    - Medication_ID: INT, PRIMARY KEY
    - Patient_ID: INT, FOREIGN KEY REFERENCING Patients (Patient_ID)
    - Medicaiton_Name: VARCHAR (100)
    - Dosage: VARCHAR (50)

- Allergies Table
    - Allergy_ID: INT, PRIMARY KEY
    - Patient_ID: INT, FOREIGN KEY REFERENCING Patients (Patient_ID)
    - Allergy: VARCHAR (100)

The database objects include tables for managing patients, trackers, pricing, reviews, and related entities, each with specific columns and data types. The DBMS handles file attributes such as table structure, indexes, data types, and storage efficiency. Constraints and transaction logs ensure data integrity and recovery. This organization supports efficient data management and retrieval, essential for the HealthTrack platform.

D.

- Efficient Table Structure and Normalization
    - Normalization – The database design is normalized to eliminate redundancy and avoid unnecessary data duplication. By breaking down the data into separate related tables, we ensure that each table stores only relevant data. This reduces storage requirements and ensures faster query performance as the system grows.
    - Indexed Primary and Foreign Keys – By using primary keys and foreign keys across the tables, we maintain data integrity and optimize query performance. Indexes allow the system to quickly retrieve data, even as the tables grow.

- Modular Data Architecture
    - Separation of Concerns – The proposed database design separates different types of data into dedicated tables. This modular structure improves performance by allowing queries to focus on specific parts of the database rather than scanning larger singular tables.
    - Horizontally Scalable Design – With separate tables for entities like Trackers, patients, Pricing, and Reviews, the system can scale horizontally. As the dataset grows, tables can be partitioned across different servers, which is essential for handling massive amounts of health-related data.

- Indexing and Query Optimization
    - Creating Indexes on Frequently Queried Columns – Indexes on commonly queried fields will significantly improve query performance. As the database grows, these indexes will help retrieve records without having to scan entire tables, ensuring scalability even with increasing data volumes.
    - Composite Indexes – In some cases, composite indexes can be created to optimize queries involving multiple columns. This speeds up complex queries that combine data from multiple tables.

- Handling High Write and Read Load
    - Read-Write Separation – The database design can leverage read-write separation by distributing read-heavy queries to read replicas. This prevents bottlenecks when large numbers of users are

querying the system simultaneously, especially when pulling health insights or device data from wearables.
    - Batch Inserts and Updates – For large-scale data inputs from wearables and medical records, batch processing can be implemented to insert or update records in bulk. This reduces the load on the database and enhances its ability to handle real-time data inputs efficiently.

E.

Given the sensitive nature of health data, such as patient information and wearable device data, implementing robust privacy and security measures is critical for the HealthTrack platform.

- Data Encryption
    - Encryption at Risk – All sensitive data, including personal information, should be encrypted when stored in the database. This ensures that even if unauthorized individuals gain access to the physical storage, the data remains unreadable without the encryption keys.
    - Encryption in Transit – Data should be encrypted while being transmitted between the client application and the database. SSL/TLS protocols should be used to protect data in transit, ensuring that intercepted data cannot be easily read or altered by attackers.

- Role-Based Access Control (RBAC)
    - Least Privilege Principle – Only authorized users should have access to specific parts of the database, and they should only be granted the minimum permissions required to perform their duties.
    - Granular Access Control – Define roles with carrying access levels such as Admin, Data Analyst, Healthcare Provider, and Patient. Each role should have specific access rights to prevent unauthorized data modifications or access to sensitive information.

- Audit Logs
    - Comprehensive Logging – Implement detailed audit logging for all database transactions. This should track user activity, such as login attempts, data access, modifications, and deletions. Each log entry should capture the user ID, timestamp, action performed, and data affected.

    - Tamper-Proof Logs – Logs should be immutable and stored securely to prevent tampering. A log management system or third-party security solution can be used to maintain the integrity of the audit logs.

- Multi-Factor Authentication (MFA)
    - Enhanced User Authentication – All users accessing the database, especially administrators and healthcare providers, should be required to use multi-factor authentication to ensure the highest level of identity verification. This adds an extra layer of protection beyond traditional username/password credentials.

- Compliance with Regulatory Standards
    - HIPAA Compliance – Ensure that the database design and management practices comply with HIPAA regulations, which dictate how healthcare data must be secured and stored. This includes encryption, access control, and audit requirements.

PART 2: Implementation

F.

1. Script to create a database instance named “D597 Task1”

```
CREATE DATABASE "D597 Task1"
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    LC_COLLATE = 'English_United States.1252'
    LC_CTYPE = 'English_United States.1252'
    LOCALE_PROVIDER = 'libc'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1
    IS_TEMPLATE = False;
```

2. Script to import the data records from the chosen scenario CSV files into the database instance

```sql
CREATE TABLE Fitness_Tracker (
    Tracker_ID SERIAL PRIMARY KEY,
    Brand_Name VARCHAR(50),
    Device_Type VARCHAR(100),
    Model_Name VARCHAR(100),
    Color VARCHAR(100),
    Selling_Price VARCHAR(50),
    Original_Price VARCHAR(50),
    Display VARCHAR(100),
    Rating FLOAT,
    Strap_Material VARCHAR(100),
    Avg_Battery_Life INT,
    Reviews VARCHAR(50)
);
CREATE TABLE Patients (
    Patient_ID INT PRIMARY KEY,
    Patient_Name VARCHAR(100),
    Date_of_Birth VARCHAR(20),
    Gender VARCHAR(5),
    Medical_Conditions TEXT,
    Medications TEXT,
    Allergies TEXT,
    Last_Appointment_Date VARCHAR(20),
    Tracker VARCHAR(100)
);
```

			PID	Type	Server	Object
			1144	Import Data	PostgreSQL 16 (localhost:5432)	D597 Task1/public.fitness_tracker
			3208	Import Data	PostgreSQL 16 (localhost:5432)	D597 Task1/public.patients

Process Watcher - Import - Copying table data

Copying table data 'public.fitness_tracker' on database 'D597 Task1' and server 'PostgreSQL 16 (localhost:5432)'
Running command:

```
--command " "\\copy public.fitness_tracker (brand_name, device_type, model_name, color, selling_price, original_price, display, rating, strap_material, avg_battery_life, reviews) FROM 'C:/WGU/D597/TASK1~1/SCENAR~1/D597TA~3.CSV' DELIMITER ',' CSV HEADER QUOTE '\"' ESCAPE '''';""
```

Start time: Fri Sep 27 2024 01:22:06 GMT+0000 (Coordinated Universal Time) Stop Process

COPY 565

Successfully completed. Execution time: 0.19 seconds

Process Watcher - Import - Copying table data

Copying table data 'public.patients' on database 'D597 Task1' and server 'PostgreSQL 16 (localhost:5432)'
Running command:

```
--command " "\\copy public.patients (patient_id, patient_name, date_of_birth, gender, medical_conditions, medications, allergies, last_appointment_date, tracker) FROM 'C:/WGU/D597/TASK1~1/SCENAR~1/D597TA~1.CSV' DELIMITER ',' CSV HEADER QUOTE '\"' ESCAPE '''';""
```

Start time: Fri Sep 27 2024 01:21:47 GMT+0000 (Coordinated Universal Time) Stop Process

COPY 100000

Successfully completed. Execution time: 1.1 seconds

```sql
CREATE TABLE Trackers (
    Tracker_ID SERIAL PRIMARY KEY,
    Brand_Name VARCHAR(50),
    Device_Type VARCHAR(100),
    Model_Name VARCHAR(100),
    Color VARCHAR(100),
    Display VARCHAR(100),
    Strap_Material VARCHAR(100),
    Avg_Battery_Life INT,
    Rating DOUBLE PRECISION
);
CREATE TABLE Pricing (
    Tracker_ID INT,
    Selling_Price DECIMAL(10,2),
    Original_Price DECIMAL(10,2),
    FOREIGN KEY (Tracker_ID) REFERENCES Trackers(Tracker_ID)
);
CREATE TABLE Reviews (
    Review_ID SERIAL PRIMARY KEY,
    Tracker_ID INT,
    Review_Text TEXT,
    FOREIGN KEY (Tracker_ID) REFERENCES Trackers(Tracker_ID)
);
CREATE TABLE Patient_Trackers (
    Patient_ID INT,
    Tracker_ID INT,
    Assigned_Date DATE,
    PRIMARY KEY (Patient_ID, Tracker_ID),
    FOREIGN KEY (Patient_ID) REFERENCES Patients(Patient_ID),
    FOREIGN KEY (Tracker_ID) REFERENCES Trackers(Tracker_ID)
);
CREATE TABLE Medical_Conditions (
    Condition_ID SERIAL PRIMARY KEY,
    Patient_ID INT,
    Condition VARCHAR(100),
    FOREIGN KEY (Patient_ID) REFERENCES Patients(Patient_ID)
);
CREATE TABLE Medications (
    Medication_ID SERIAL PRIMARY KEY,
    Patient_ID INT,
    Medication_Name VARCHAR(100),
    Dosage VARCHAR(50),
    FOREIGN KEY (Patient_ID) REFERENCES Patients(Patient_ID)
);
CREATE TABLE Allergies (
    Allergy_ID SERIAL PRIMARY KEY,
    Patient_ID INT,
    Allergy VARCHAR(100),
    FOREIGN KEY (Patient_ID) REFERENCES Patients(Patient_ID)
);
```

```sql
INSERT INTO Trackers (Brand_Name, Device_Type, Model_Name, Color, Display, Strap_Material, Avg_Battery_Life, Rating
SELECT Brand_Name, Device_Type, Model_Name, Color, Display, Strap_Material, Avg_Battery_Life, Rating
FROM Fitness_Tracker;

INSERT INTO Pricing (Tracker_ID, Selling_Price, Original_Price)
SELECT Tracker_ID,
       REPLACE(Selling_Price, ',', '')::DECIMAL(10, 2),
       REPLACE(Original_Price, ',', '')::DECIMAL(10, 2)
FROM Fitness_Tracker;

INSERT INTO Reviews (Tracker_ID, Review_Text)
SELECT Tracker_ID, Reviews
FROM Fitness_Tracker
WHERE Reviews IS NOT NULL AND Reviews <> '';

INSERT INTO Patient_Trackers (Patient_ID, Tracker_ID, Assigned_Date)
SELECT p.Patient_ID, t.Tracker_ID, TO_DATE(p.Last_Appointment_Date, 'DD/MM/YYYY')
FROM Patients p
JOIN Trackers t ON p.Tracker = t.Model_Name;

INSERT INTO Medical_Conditions (Patient_ID, Condition)
SELECT Patient_ID, Medical_Conditions
FROM Patients
WHERE Medical_Conditions IS NOT NULL AND Medical_Conditions <> 'None';

INSERT INTO Allergies (Patient_ID, Allergy)
SELECT Patient_ID, Allergies
FROM Patients
WHERE Allergies IS NOT NULL AND Allergies <> 'None';

ALTER TABLE Patients
DROP COLUMN Medical_Conditions,
DROP COLUMN Medications,
DROP COLUMN Allergies,
DROP COLUMN Last_Appointment_Date,
DROP COLUMN Tracker;

DROP TABLE Fitness_Tracker;
```

3. Script for three queries to retrieve specific information from the database that will help solve the identified business problems.

- **Query 1**

Query   Query History   Scratch Pad

```
SELECT
    p.Patient_Name,
    p.Date_of_Birth,
    p.Gender,
    t.Brand_Name,
    t.Model_Name,
    pt.Assigned_Date
FROM
    Patients p
JOIN
    Patient_Trackers pt ON p.Patient_ID = pt.Patient_ID
JOIN
    Trackers t ON pt.Tracker_ID = t.Tracker_ID;
```

Data Output   Messages   Notifications

	patient_name character varying (100)	date_of_birth character varying (20)	gender character varying (5)	brand_name character varying (50)	model_name character varying (100)	assigned_date date
1	Scott Webb	28/04/1967	F	Honor	Band 4	2022-07-26
2	Scott Webb	28/04/1967	F	Honor	Band 4	2022-07-26
3	Scott Webb	28/04/1967	F	Huawei	Band 4	2022-07-26
4	Rachel Frederick	04/04/1977	M	Honor	Band 3	2023-02-14
5	Rachel Frederick	04/04/1977	M	Honor	Band 3	2023-02-14
6	Rachel Frederick	04/04/1977	M	Honor	Band 3	2023-02-14
7	Rachel Frederick	04/04/1977	M	Xiaomi	Band 3	2023-02-14
8	Eric Kline	18/05/1926	F	Honor	Band 5i	2021-04-24
9	Eric Kline	18/05/1926	F	Honor	Band 5i	2021-04-24
10	James Rodriguez	20/07/1954	M	Honor	Band 3	2022-05-26
11	James Rodriguez	20/07/1954	M	Honor	Band 3	2022-05-26
12	James Rodriguez	20/07/1954	M	Honor	Band 3	2022-05-26
13	James Rodriguez	20/07/1954	M	Xiaomi	Band 3	2022-05-26
14	David Scott	07/12/2015	M	Infinix	Band 5	2021-05-17
15	David Scott	07/12/2015	M	Honor	Band 5	2021-05-17
16	David Scott	07/12/2015	M	Honor	Band 5	2021-05-17
17	David Scott	07/12/2015	M	Honor	Band 5	2021-05-17
18	Dawn Roach	02/06/1967	M	Infinix	Band 5	2022-02-02
19	Dawn Roach	02/06/1967	M	Honor	Band 5	2022-02-02
20	Dawn Roach	02/06/1967	M	Honor	Band 5	2022-02-02
21	Dawn Roach	02/06/1967	M	Honor	Band 5	2022-02-02
22	Mary Harris	27/08/2013	M	Infinix	Band 5	2021-06-02
23	Mary Harris	27/08/2013	M	Honor	Band 5	2021-06-02
24	Mary Harris	27/08/2013	M	Honor	Band 5	2021-06-02
25	Mary Harris	27/08/2013	M	Honor	Band 5	2021-06-02
26	Sandy Brown	23/06/1927	F	Honor	Band 4	2021-05-05

Total rows: 1000 of 202786   Query complete 00:00:00.559   Ln 14, Col 1

This query retrieves patient details along with information about the fitness tracker they are using. It helps link patients to the devices tracking their health metrics.

- **Query 2**

Query | Query History | Scratch Pad

```sql
SELECT
    t.Brand_Name,
    t.Model_Name,
    t.Rating,
    t.Avg_Battery_Life
FROM
    Trackers t
WHERE
    t.Rating >= 4.0
ORDER BY
    t.Rating DESC;
```

Data Output | Messages | Notifications

	brand_name character varying (50)	model_name character varying (100)	rating double precision	avg_battery_life integer
1	APPLE	Series 7 GPS 41 mm Aluminium Case	5	1
2	GARMIN	Instinct	5	7
3	FOSSIL	Neutra Hybrid	5	12
4	APPLE	Series 7 GPS 45 mm	5	1
5	GARMIN	vivoactive 4S 40mm	5	14
6	SAMSUNG	Galaxy Classic 4 LTE	5	14
7	FOSSIL	FTW1159 Hybrid	5	2
8	GARMIN	Forerunner 745 Black	5	14
9	FOSSIL	Neutra Hybrid	5	12
10	FOSSIL	Carlie	4.8	10
11	FOSSIL	FTW4010 Gold HR	4.8	6
12	GARMIN	Vivoactive	4.8	14
13	FOSSIL	FTW5011 Hybrid	4.8	2
14	GARMIN	Forerunner 935	4.7	14
15	APPLE	42 mm White Ceramic Case with Cloud Sport	4.7	1
16	GARMIN	Forerunner 935	4.7	14
17	GARMIN	Forerunner 235	4.7	14
18	APPLE	SE 44 mm Space Grey Aluminium Case	4.7	1
19	FOSSIL	Neely	4.7	1
20	APPLE	Nike Series 5 GPS + Cellular 44 mm	4.7	1
21	GARMIN	Instinct Solar Camo Edition	4.7	14
22	APPLE	SE GPS 40 mm Aluminium Case	4.7	1
23	APPLE	Series 5 GPS + Cellular 44 mm Gold Aluminium Case	4.7	1
24	APPLE	Series 2 - 42 mm Stainless Steel Case	4.7	1
25	FitBit	Lunar	4.7	7
26	GARMIN	Forerunner 935	4.7	14

Total rows: 419 of 419 Query complete 00:00:00.149 Ln 12, Col 1

This query helps identify which trackers are most preferred by users based on ratings, allowing HealthFit to focus on promoting or further analyzing high-rated devices

- **Query 3**

Query   Query History   Scratch Pad

```sql
SELECT
    p.Patient_Name,
    mc.Condition,
    t.Brand_Name,
    t.Model_Name
FROM
    Patients p
JOIN
    Medical_Conditions mc ON p.Patient_ID = mc.Patient_ID
JOIN
    Patient_Trackers pt ON p.Patient_ID = pt.Patient_ID
JOIN
    Trackers t ON pt.Tracker_ID = t.Tracker_ID
ORDER BY
    p.Patient_Name;
```

Data Output   Messages   Notifications

	patient_name character varying (100)	condition character varying (100)	brand_name character varying (50)	model_name character varying (100)
1	Aaron Allen	Watch	huami	Amazfit Verge
2	Aaron Allen	Watch	huami	Amazfit Verge
3	Aaron Allen	Watch	huami	Amazfit Verge
4	Aaron Allen	Watch	huami	Amazfit Verge
5	Aaron Andrews	Watch	Honor	Band 5
6	Aaron Andrews	Watch	Infinix	Band 5
7	Aaron Andrews	Watch	Honor	Band 5
8	Aaron Andrews	Watch	Honor	Band 5
9	Aaron Ashley	Watch	Oppo	41mm
10	Aaron Baker	Watch	Honor	Band 4
11	Aaron Baker	Watch	Huawei	Band 4
12	Aaron Baker	Watch	Honor	Band 4
13	Aaron Baker	Watch	huami	Amazfit Bip
14	Aaron Baker	Watch	Infinix	Band 5
15	Aaron Baker	Watch	Honor	Band 5
16	Aaron Baker	Watch	Honor	Band 5
17	Aaron Baker	Watch	Honor	Band 5
18	Aaron Baker	Watch	huami	Amazfit Bip
19	Aaron Baker	Watch	huami	Amazfit Bip
20	Aaron Baker	Watch	huami	Amazfit Bip
21	Aaron Baker	Watch	huami	Amazfit Bip
22	Aaron Barber	Watch	huami	Amazfit Bip Lite
23	Aaron Bauer	Watch	huami	Bip Lite On
24	Aaron Boone	Watch	huami	Amazfit GTR 2 Aluminium

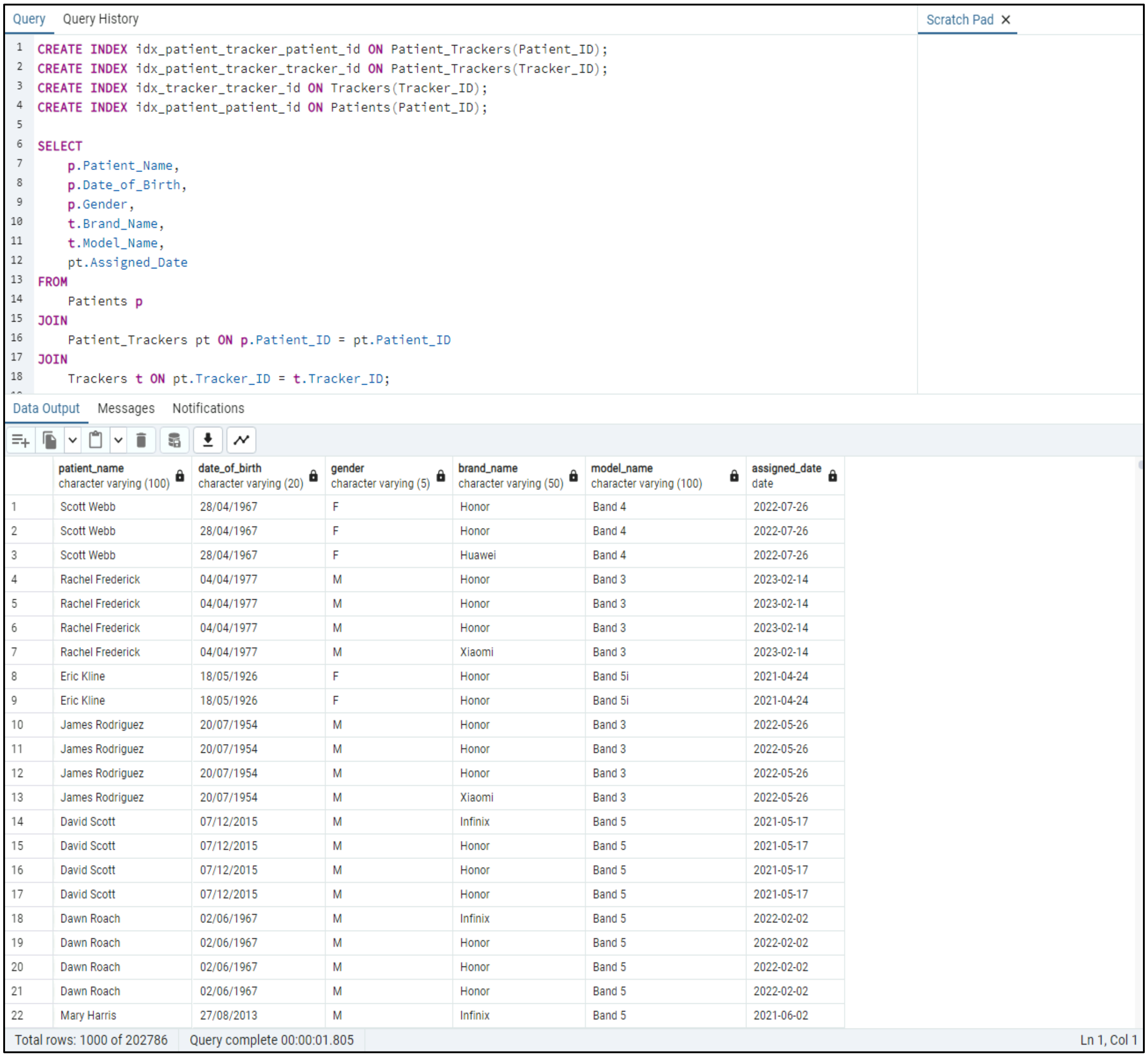
✓ Successfully run. Total query runtime: 597 msec. 77422 rows affected.

Total rows: 1000 of 77422   Query complete 00:00:00.597   Ln 16, Col 1

This query retrieves patients who have specific medical conditions and the trackers they are using. This can help in understanding how different health conditions might correlate with tracker usage and performance.

4. Apply optimization techniques to improve the run time of the queries from F3.

- **Optimized Query 1**



Query | Query History | Scratch Pad

```sql
CREATE INDEX idx_patient_tracker_patient_id ON Patient_Trackers(Patient_ID);
CREATE INDEX idx_patient_tracker_tracker_id ON Patient_Trackers(Tracker_ID);
CREATE INDEX idx_tracker_tracker_id ON Trackers(Tracker_ID);
CREATE INDEX idx_patient_patient_id ON Patients(Patient_ID);

SELECT
    p.Patient_Name,
    p.Date_of_Birth,
    p.Gender,
    t.Brand_Name,
    t.Model_Name,
    pt.Assigned_Date
FROM
    Patients p
JOIN
    Patient_Trackers pt ON p.Patient_ID = pt.Patient_ID
JOIN
    Trackers t ON pt.Tracker_ID = t.Tracker_ID;
```

Data Output | Messages | Notifications

	patient_name character varying (100)	date_of_birth character varying (20)	gender character varying (5)	brand_name character varying (50)	model_name character varying (100)	assigned_date date
1	Scott Webb	28/04/1967	F	Honor	Band 4	2022-07-26
2	Scott Webb	28/04/1967	F	Honor	Band 4	2022-07-26
3	Scott Webb	28/04/1967	F	Huawei	Band 4	2022-07-26
4	Rachel Frederick	04/04/1977	M	Honor	Band 3	2023-02-14
5	Rachel Frederick	04/04/1977	M	Honor	Band 3	2023-02-14
6	Rachel Frederick	04/04/1977	M	Honor	Band 3	2023-02-14
7	Rachel Frederick	04/04/1977	M	Xiaomi	Band 3	2023-02-14
8	Eric Kline	18/05/1926	F	Honor	Band 5i	2021-04-24
9	Eric Kline	18/05/1926	F	Honor	Band 5i	2021-04-24
10	James Rodriguez	20/07/1954	M	Honor	Band 3	2022-05-26
11	James Rodriguez	20/07/1954	M	Honor	Band 3	2022-05-26
12	James Rodriguez	20/07/1954	M	Honor	Band 3	2022-05-26
13	James Rodriguez	20/07/1954	M	Xiaomi	Band 3	2022-05-26
14	David Scott	07/12/2015	M	Infinix	Band 5	2021-05-17
15	David Scott	07/12/2015	M	Honor	Band 5	2021-05-17
16	David Scott	07/12/2015	M	Honor	Band 5	2021-05-17
17	David Scott	07/12/2015	M	Honor	Band 5	2021-05-17
18	Dawn Roach	02/06/1967	M	Infinix	Band 5	2022-02-02
19	Dawn Roach	02/06/1967	M	Honor	Band 5	2022-02-02
20	Dawn Roach	02/06/1967	M	Honor	Band 5	2022-02-02
21	Dawn Roach	02/06/1967	M	Honor	Band 5	2022-02-02
22	Mary Harris	27/08/2013	M	Infinix	Band 5	2021-06-02

Total rows: 1000 of 202786 | Query complete 00:00:01.805 | Ln 1, Col 1

This query was optimized by creating Indexes on Patient_Trackers.Patient_ID, Trackers.Tracker_ID, Patients.Patient_ID to speed up the join operations.

- **Optimized Query 2**

Query | Query History | Scratch Pad

```sql
CREATE INDEX idx_tracker_rating ON Trackers(Rating);

SELECT
    t.Brand_Name,
    t.Model_Name,
    t.Rating,
    t.Avg_Battery_Life
FROM
    Trackers t
WHERE
    t.Rating >= 4.0
ORDER BY
    t.Rating DESC;

```

Data Output | Messages | Notifications

	brand_name character varying (50)	model_name character varying (100)	rating double precision	avg_battery_life integer
1	APPLE	Series 7 GPS 41 mm Aluminium Case	5	1
2	GARMIN	Instinct	5	7
3	FOSSIL	Neutra Hybrid	5	12
4	APPLE	Series 7 GPS 45 mm	5	1
5	GARMIN	vivoactive 4S 40mm	5	14
6	SAMSUNG	Galaxy Classic 4 LTE	5	14
7	FOSSIL	FTW1159 Hybrid	5	2
8	GARMIN	Forerunner 745 Black	5	14
9	FOSSIL	Neutra Hybrid	5	12
10	FOSSIL	Carlie	4.8	10
11	FOSSIL	FTW4010 Gold HR	4.8	6
12	GARMIN	Vivoactive	4.8	14
13	FOSSIL	FTW5011 Hybrid	4.8	2
14	GARMIN	Forerunner 935	4.7	14
15	APPLE	42 mm White Ceramic Case with Cloud Sport	4.7	1
16	GARMIN	Forerunner 935	4.7	14
17	GARMIN	Forerunner 235	4.7	14
18	APPLE	SE 44 mm Space Grey Aluminium Case	4.7	1
19	FOSSIL	Neely	4.7	1
20	APPLE	Nike Series 5 GPS + Cellular 44 mm	4.7	1
21	GARMIN	Instinct Solar Camo Edition	4.7	14
22	APPLE	SE GPS 40 mm Aluminium Case	4.7	1
23	APPLE	Series 5 GPS + Cellular 44 mm Gold Aluminium Case	4.7	1
24	APPLE	Series 2 - 42 mm Stainless Steel Case	4.7	1
25	FitBit	Lunar	4.7	7
26	GARMIN	Forerunner 935	4.7	14

✓ Successfully run. Total query runtime: 144 msec. 419 rows affected.

Total rows: 419 of 419 Query complete 00:00:00.144 Ln 2, Col 1

This query was optimized by creating an Index on Trackers.Rating to quickly filter high-rated devices.

- **Optimized Query 3**

Query Query History

Scratch Pad

```sql
CREATE INDEX idx_medical_condition_patient_id ON Medical_Conditions(Patient_ID);

SELECT
    p.Patient_Name,
    mc.Condition,
    t.Brand_Name,
    t.Model_Name
FROM
    Patients p
JOIN
    Medical_Conditions mc ON p.Patient_ID = mc.Patient_ID
JOIN
    Patient_Trackers pt ON p.Patient_ID = pt.Patient_ID
JOIN
    Trackers t ON pt.Tracker_ID = t.Tracker_ID
ORDER BY
    p.Patient_Name;
```

Data Output Messages Notifications

	patient_name character varying (100)	condition character varying (100)	brand_name character varying (50)	model_name character varying (100)
1	Aaron Allen	Watch	huami	Amazfit Verge
2	Aaron Allen	Watch	huami	Amazfit Verge
3	Aaron Allen	Watch	huami	Amazfit Verge
4	Aaron Allen	Watch	huami	Amazfit Verge
5	Aaron Andrews	Watch	Honor	Band 5
6	Aaron Andrews	Watch	Honor	Band 5
7	Aaron Andrews	Watch	Infinix	Band 5
8	Aaron Andrews	Watch	Honor	Band 5
9	Aaron Ashley	Watch	Oppo	41mm
10	Aaron Baker	Watch	Honor	Band 5
11	Aaron Baker	Watch	Honor	Band 5
12	Aaron Baker	Watch	huami	Amazfit Bip
13	Aaron Baker	Watch	Honor	Band 5
14	Aaron Baker	Watch	Infinix	Band 5
15	Aaron Baker	Watch	huami	Amazfit Bip
16	Aaron Baker	Watch	huami	Amazfit Bip
17	Aaron Baker	Watch	huami	Amazfit Bip
18	Aaron Baker	Watch	huami	Amazfit Bip
19	Aaron Baker	Watch	Huawei	Band 4
20	Aaron Baker	Watch	Honor	Band 4
21	Aaron Baker	Watch	Honor	Band 4
22	Aaron Barber	Watch	huami	Amazfit Bip Lite
23	Aaron Bauer	Watch	huami	Bip Lite On

Successfully run. Total query runtime: 654 msec. 77422 rows affected.

Total rows: 1000 of 77422 Query complete 00:00:00.654 Ln 1, Col 1

This query was optimized on Medical_Conditions.Patient_ID, Patient_Tracker.Patient_ID, and Patient_Trackers.Tracker_ID for faster joins.