Health Metric Tracker Database Design

1. Database Technology Choice

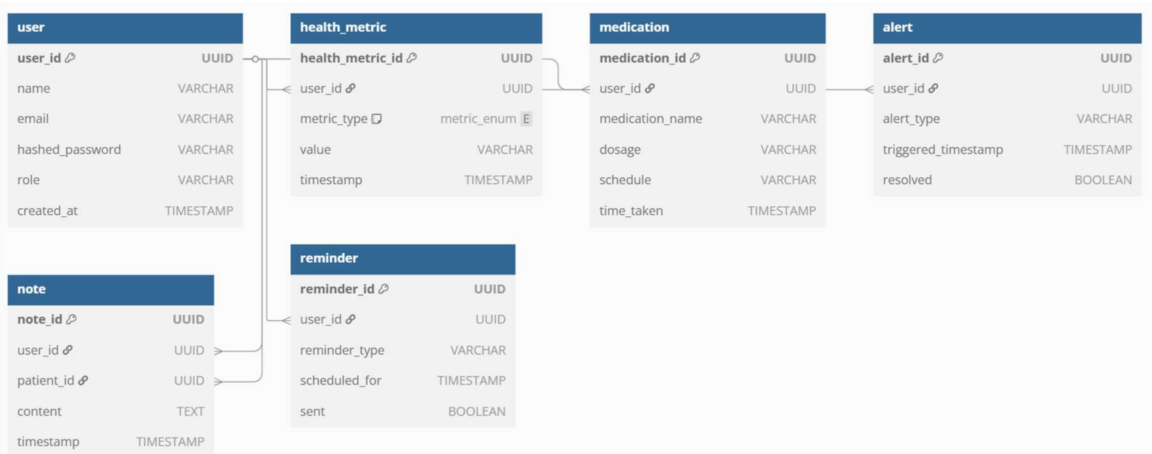
Selected Database: PostgreSQL (Relational Database)

Justification:
We selected PostgreSQL because our application involves:

- Highly structured and normalized data (e.g., users, health metrics, medications, alerts).
- Clear one-to-many relationships between users and various logged data types.
- Relational integrity and scalability, as we may introduce more complex querying/reporting in future versions.
- Transactional safety for logging critical health events and medications.
- Security features like role-based access and encrypted passwords.
- PK – Primary Key; FK – Foreign Key

2. Entity-Relationship (ER) Diagram

Figure 1: Entity-Relationship Diagram for Health Metric Tracker (Made using dbdiagram.io)



user	
user_id	**UUID**
name	VARCHAR
email	VARCHAR
hashed_password	VARCHAR
role	VARCHAR
created_at	TIMESTAMP

health_metric	
health_metric_id	**UUID**
user_id	UUID
metric_type	metric_enum E
value	VARCHAR
timestamp	TIMESTAMP

medication	
medication_id	**UUID**
user_id	UUID
medication_name	VARCHAR
dosage	VARCHAR
schedule	VARCHAR
time_taken	TIMESTAMP

alert	
alert_id	**UUID**
user_id	UUID
alert_type	VARCHAR
triggered_timestamp	TIMESTAMP
resolved	BOOLEAN

note	
note_id	**UUID**
user_id	UUID
patient_id	UUID
content	TEXT
timestamp	TIMESTAMP

reminder	
reminder_id	**UUID**
user_id	UUID
reminder_type	VARCHAR
scheduled_for	TIMESTAMP
sent	BOOLEAN

3. Tables & Descriptions

User Table

Column Name	Type	Description
user_id	UUID (PK)	Unique user ID
Name	VARCHAR	Full name
Email	VARCHAR	Email address (unique)
hashed_password	VARCHAR	Securely stored password
Role	VARCHAR	Role (patient, caregiver, doctor)
created_at	TIMESTAMP	Account creation timestamp

Purpose: Describes the role of the users in the app.

Health_Metric Table

Column Name	Type	Description
user_id	UUID (PK)	Unique metric ID
user_id	UUID (FK)	Reference to User table
metric_type	VARCHAR	Type (e.g., BP, glucose, heart rate)
Value	VARCHAR	Recorded value (e.g., 140/90)
Timestamp	TIMESTAMP	When the metric was logged

Purpose: Describes the role of the health_metrics in the app.

Medication Table

Column Name	Type	Description
user_id	UUID (PK)	Unique medication entry
user_id	UUID (FK)	Reference to User table
medication_name	VARCHAR	Name of medication
Dosage	VARCHAR	Dosage information (e.g., 10 mg)
Schedule	VARCHAR	Timing/frequency (e.g., 2x daily)
time_taken	TIMESTAMP	Actual time medication was taken

Purpose: Describes the role of the medications in the app.

Alert Table

Column Name	Type	Description
user_id	UUID (PK)	Unique alert ID
user_id	UUID (FK)	Reference to User table
alert_type	VARCHAR	Reason for alert (e.g., High BP)
triggered_timestamp	TIMESTAMP	When the alert was created
Resolved	BOOLEAN	Whether the alert has been resolved

Purpose: Describes the role of the alerts in the app.

Note Table

Column Name	Type	Description
user_id	UUID (PK)	Unique note ID
user_id	UUID (FK)	Reference to user (caregiver/doctor)
patient_id	UUID (FK)	Reference to the user (the patient)
Content	TEXT	Note content
Timestamp	TIMESTAMP	When the note was added

Purpose: Describes the role of the notes in the app.

Reminder Table

Column Name	Type	Description
user_id	UUID (PK)	Unique reminder ID
user_id	UUID (FK)	Associated user
reminder_type	VARCHAR	Type (medication, BP check, follow-up)
scheduled_for	TIMESTAMP	Date/time of the reminder
Sent	BOOLEAN	Whether reminder has been sent

Purpose: Describes the role of the reminders in the app.

4. Relationships

- User ↔ Health_Metric: One-to-Many (each user logs many entries).
- User ↔ Medication: One-to-Many (users can track multiple meds).
- User ↔ Alert: One-to-Many (users can have multiple alerts).
- User ↔ Note (Caregiver/Doctor to Patient): Many-to-Many with context.
- User ↔ Reminder: One-to-Many (reminders for check-ins or medications).

5. Sample Usage Scenarios

Scenario 1: John Logs a BP Reading

1. John logs into the system.
2. He enters "140/90" in the Health_Metric table.
3. The system detects it's elevated and inserts a new alert into the Alert table.
4. Sarah (his caregiver) receives an alert notification.
5. She adds a note via the Note table.

Scenario 2: Doctor Reviews Patient Report

1. Dr. Patel accesses the user's Reports page.
2. The system queries Health_Metric and Alert tables.
3. Trends are generated dynamically and displayed.

6. Security and Access

- Roles: Patient, Caregiver, Doctor, Admin.
- Access Control:
  - Patients can only view/edit their own data.
  - Caregivers can view patient data they're linked to.
  - Doctors can access all patient records for review.
  - Admins can manage all users.

Enumerated Types

The `metric_type` field in the `Health_Metric` table now uses a PostgreSQL ENUM called `metric_enum` to restrict allowed metric types.
Defined Enum Values:

- 'BP' (Blood Pressure)
- 'O2' (Oxygen Saturation)
- 'heart_rate'
- 'weight'
- 'height'
- 'BMI'
- 'blood_glucose'

This improves data validation, ensures input consistency, and allows us to map each metric to a corresponding input control in the frontend.

Table Partitioning - Performance Optimization

The `Health_Metric` table may be partitioned by `timestamp` to improve performance for time-based queries and reduce the load on large datasets, in the future. May not be partitioned for MVP – will need to think on this.

Example partitioning strategies:

- Partition by quarter or month using the `timestamp` field.
- Partition by `user_id` for heavy data users.

These partitions are invisible to the application layer and are managed at the database level using PostgreSQL's native partitioning features.

7. Stretch Feature Tables

Symptom Log Table

Purpose: Allows users to track symptoms alongside metrics, providing valuable context for trends and alerts.

Column Name	Type	Description
symptom_log_id	UUID (PK)	Unique ID for each symptom entry
user_id	UUID (FK)	Reference to User table
symptom_type	VARCHAR	Type of symptom (e.g., headache, nausea)
Severity	INTEGER	Severity on a scale (e.g., 1–10)
Notes	TEXT	Optional notes or context
Timestamp	TIMESTAMP	When the symptom was logged

Caregiver Assignment Table

Purpose: Formally tracks which caregiver is assigned to each patient, enabling better access control and notifications.

Column Name	Type	Description
assignment_id	UUID (PK)	Unique assignment ID
caregiver_id	UUID (FK)	References User table (caregiver role)
patient_id	UUID (FK)	References User table (patient role)
assigned_on	TIMESTAMP	Date the caregiver was linked to the patient