Database Management Group 9

Group Members

1. Abdul Samed Al-Hassan- 11232733

2. Brandford Mettle-11259497

3. Borketey Pearl Borley-11015547

Table of Contents

Wearable Health Device Data Hub
Complete System Documentation

System Overview and Purpose

Brief Summary of the System

The Wearable Health Device Data Hub is a centralized platform designed to collect, process, and analyze health data from wearable devices such as fitness trackers and smartwatches. It continuously monitors key health metrics including heart rate, sleep patterns, physical activity, and calories burned. By aggregating data from multiple devices and applying intelligent analytics, the system delivers personalized insights and health recommendations. Its goal is to empower users to better understand and manage their health, support early detection of potential issues, and encourage healthier lifestyle choices through data-driven feedback.

Key Features

Expected Outcomes

- **User Management**
  Stores essential user information including name, email, age, gender, and registration details.
- **Device Integration**
  Tracks multiple health devices per user, supporting device-specific data collection.
- **Health Data Collection**
  Captures timestamped health data (e.g., heart rate, sleep, steps) linked to both the user and the device that recorded it.
- **Metric-Based Tracking**
  Each health data entry is categorized by metric type (e.g., heart rate, calories burned) with associated units for accurate interpretation.
- **Personalized Recommendations**
  Generates health recommendations for users based on collected data and device usage.

- **Relational Data Model**
  Strongly linked entities (Users, Devices, Health Metrics, Health Data, Recommendations) ensure data integrity and logical structure.
- **Scalability**
  Easily expandable to accommodate new devices, metrics, or users without altering the core structure.
- **Analytics-Ready Design**
  Optimized schema to support advanced queries, trend analysis, and future integration with AI or machine learning tools.

Expected Outcomes

- **Centralized Health Data Storage**: All user health data from various devices is securely stored and easily accessible in one unified database.
- **Personalized Recommendations**: The system can generate tailored health recommendations for users based on their historical health data and device usage.
- **Device Usage Tracking**: The system provides a clear mapping of which user used which device, enabling accurate tracking of device performance and data sources.
- **Comprehensive Health Monitoring**: Users' health can be monitored over time using various metrics (e.g., heart rate, steps, sleep), enabling early detection of anomalies or trends.
- **Improved Data Accuracy and Integrity**: By linking health data to specific users, devices, and metrics, the system ensures high data accuracy and traceability.
- **Scalable and Extensible System**: The database structure supports the integration of additional metrics, devices, or users without major restructuring.

Key Actors And Users

- **Users (Customers)**: These are the primary users who own and wear the health devices. They interact with a dashboard or mobile app to view their health metrics, track progress, receive recommendations, set wellness goals and connect new devices to their profile.

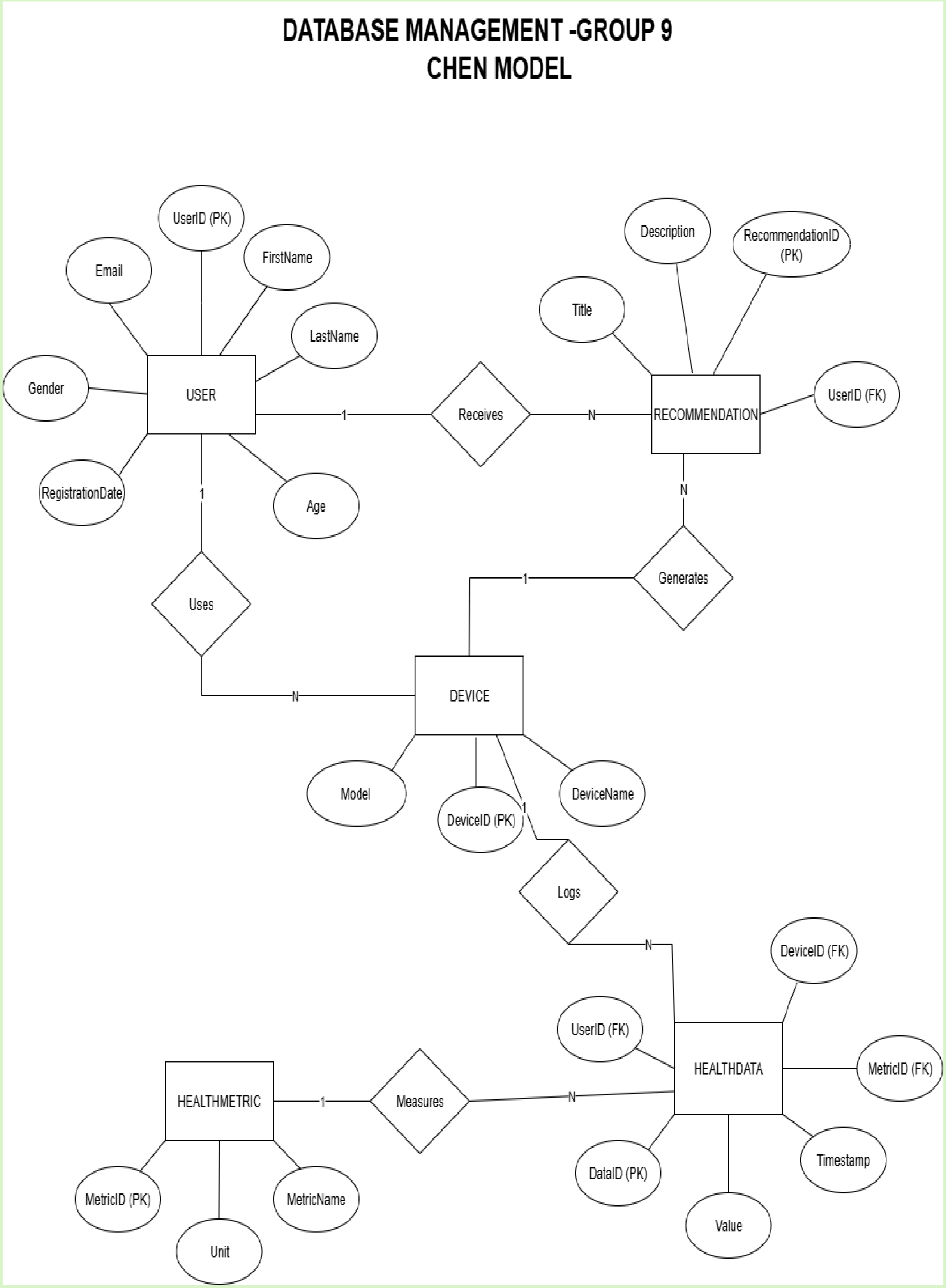
- **System Administrators**: They are responsible for the system's performance, security, and data integrity. They use an admin interface to monitor the system, manage user accounts, review system-wide analytics and handling issues such as data breaches or system errors
- **Healthcare Professionals (optional integration**): They support medical guidance and review of user health data (with user consent). They might access a patient's shared health metrics, monitor real-time or historical health data for clinical insights and suggest further tests or lifestyle changes based on system analytics.
- **Third-Party Developers / API Consumers**: These are external software developers or companies that create new apps or services that work with the system. They use the system's API to connect their own apps or tools. (e.g., integration with insurance apps, fitness platforms).
- **Device Vendors / Manufacturers:** These are the companies that make the wearable devices, like smartwatches and fitness trackers. They make sure the health data collected by their devices (like heart rate or steps) can be correctly shared with the system.

Database Design

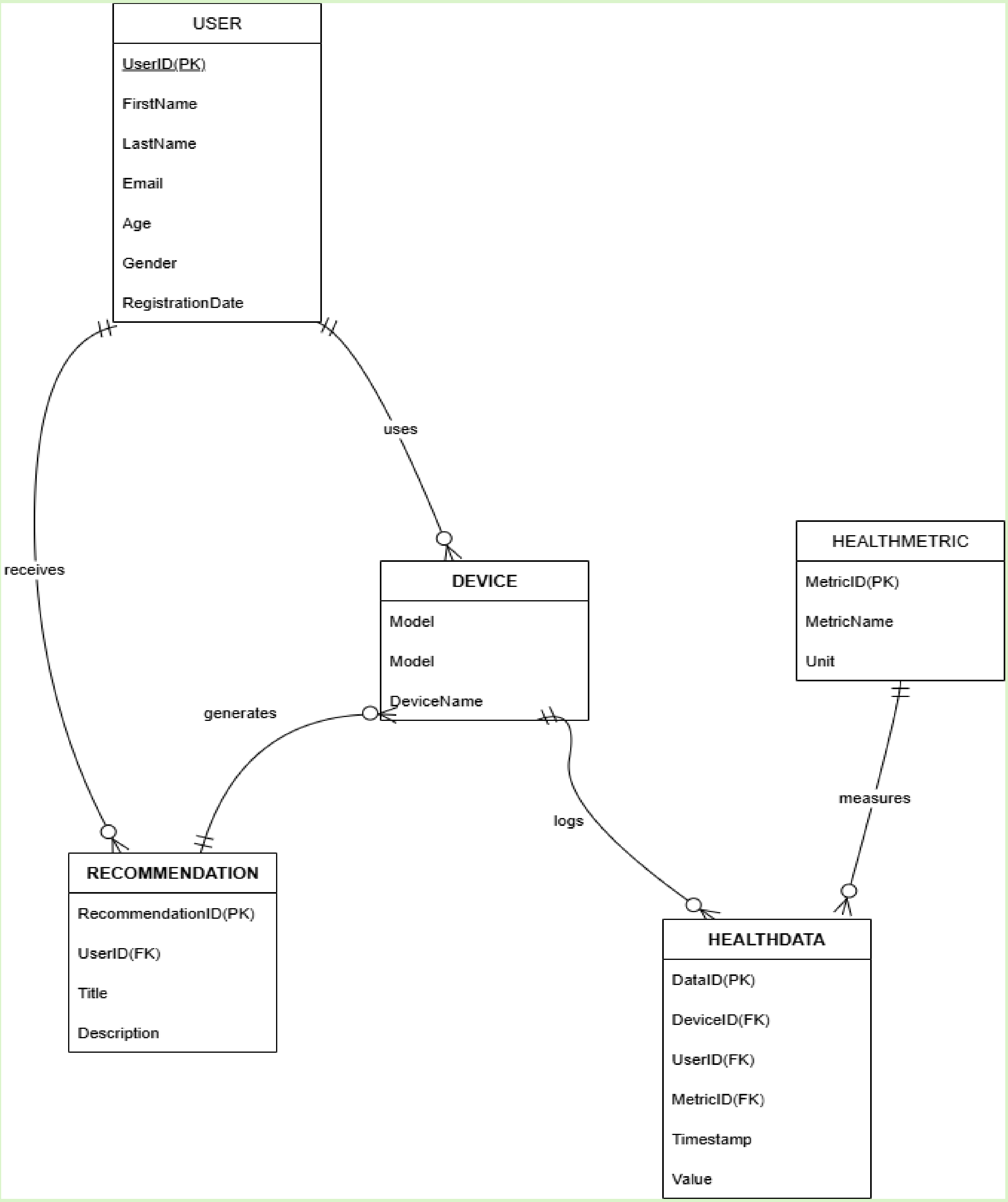
The database was designed with a clear goal: to create an efficient and reliable system for managing health data. The tables were chosen to avoid a common problem called data redundancy, where the same information is stored in multiple places. By breaking down the data into these specific tables, we ensure that each piece of information is stored only once, which makes the database easier to manage and keeps the data consistent. We decided to use:

Use:	Description
1.Users	Stores information about all registered users of the wearable health devices. This includes their personal details, age, and registration date.
2.Device	Stores information about the wearable health devices themselves, such as the model and a user-friendly name.
3.HealthMetric	A lookup table that defines the different types of health metrics that can be tracked. It includes the metric's name (e.g., "Heart Rate") and the unit of measurement (e.g., "bpm").
4.HealthData	The core table for storing the actual time-series health data collected from the devices. Each row represents a single data point, linked to a specific user, metric, and device.
5.Recommendation	Stores personalized health recommendations for users. Each recommendation is tied to a specific user.

DATABASE MANAGEMENT GROUP 9 CHEN MODEL



DATABASE MANAGEMENT GROUP 9 CROW FOOT MODEL



Before we explain the database component in depth lets show you how we normalized from its raw state.

Normalization Process and Results

Unnormalized Form (UNF)

In the beginning, all the data existed as a flat file. This structure was unnormalized and contains repeating groups and redundant data. A single row captured everything about a user, including all their health readings from all their devices.

Sample Table:

UserID	Name	Email	Age	Device Model	Device Name	Health Readings (Repeating Group)	Recommenda-tions (Repeating Group)
1	Alice Smith	alice.s@e.com	30	FitBit Charge 5	Alice's FitBit	('Heart Rate', 'bpm', 75, '2025-07-31 08:00'), ('Steps', 'steps', 10245, '2025-07-31 18:30')	('Increase Steps', 'Aim for 10,000 steps...'), ('Improve Sleep', 'Go to bed earlier...')

Problems with the unnormalized data

- **Repeating Groups (Violates 1NF)**

  Both HealthReadings and Recommendations columns contain multiple values in a single field (repeating groups). The Name field also stores both first name and last name in a single column (e.g., "Alice Smith"), This also violates 1NF because the field is not atomic

- **Data Redundancy**

  User information (Username, Email) was repeated for every entry, meaning If Alice has more devices or readings, you'll have to repeat her personal data (name, email, etc.) for each new entry.

  Hard to Query.

- You can't also easily query things like “show all readings of heart rate above 80 bpm” without parsing a text blob. Users may also enter their names in different formats, making it hard query reliably when they aren't split into first and last names

- **No Primary Keys for Repeating Entities**

  You can't uniquely identify each individual HealthReading or Recommendation.

- **Mixed Data Types in One Field**

  For example, ('Heart Rate', 'bpm', 75, '2025-07-31 08:00') mixes a metric name, unit, value, and timestamp in one field — violating atomicity.

- **Device Data Not Normalized**

  The device info (DeviceModel, DeviceName) is embedded in the same row as the user, instead of being stored in a separate device table.

First Normal Form (1NF)

In the first normal form our aim is to eliminate repeating groups and ensure every column holds a single, atomic value. Here, we separated Recommendations into its own table at this stage since it repeats per user, not per health reading. we also split the Name field into FirstName and LastName columns

Table 1: HealthLog_1NF

- **Primary Key:** A combination of attributes is needed to uniquely identify a row. A logical choice is (UserID, DeviceID, MetricName, Timestamp). This is a **composite primary key**.

UserID	FirstName	LastName	Email	DeviceID	DeviceModel	DeviceName	MetricName	MetricUnit	Value	Timestamp
1	Alice	Smith	alice.s@e.com	101	FitBit Charge 5	Alice's FitBit	Heart Rate	bpm	75.00	2025-07-31 08:00
1	Alice	Smith	alice.s@e.com	101	FitBit Charge 5	Alice's FitBit	Steps Count	steps	10245.00	2025-07-31 18:30
2	Bob	Johnson	bob.j@e.com	102	Apple Watch 8	Bob's Watch	Heart Rate	bpm	68.00	2025-07-31 09:00

The Recommendations table.

- **Primary Key:** (UserID, Title)

UserID	FirstName	LastName	Title	Description
1	Alice	Smith	Increase Steps	Aim for 10,000 steps...
1	Alice	Smith	Improve Sleep	Go to bed earlier...

Status: The data is now in 1NF. All repeating groups are gone, but there is still significant data redundancy, which leads to update, insertion, and deletion anomalies.

Second Normal Form (2NF)

Aim: Since the data is in 1NF, the next focus is to remove **partial dependencies**. A partial dependency occurs when a non-key attribute is dependent on only a part of the composite primary key.

Analyzing our HealthLog_1NF table we notice the primary key (UserID, DeviceID, MetricName, Timestamp).

- FirstName, LastName, Email depend only on **UserID**. (Partial Dependency)
- DeviceModel and DeviceName depend only on **DeviceID**. (Partial Dependency)
- MetricUnit depends only on **MetricName**. (Partial dependency)
- Value is the only attribute that depends on the *entire* primary key.

To resolve this, we created separate tables for these dependent attributes.

1. **Users Table:** Pulled out all attributes depending only on UserID.

- Users (UserID (PK)**, FirstName, LastName, Email, Age, Gender, RegistrationDate)

2. **Device Table:** Pull out attributes dependent only on DeviceID.
   - **Device (DeviceID (PK)**, Model, DeviceName)
3. **HealthMetric Table:** For attributes dependent only on MetricName.
   HealthMetric (MetricID (PK), MetricName, Unit)
4. **The HealthLog table is now reduced** to the primary key components and the fully dependent Value attribute.
   - **HealthData (DataID (PK)**, UserID (FK), DeviceID (FK), MetricID (FK), Timestamp, Value)

We do the same for the Recommendations table, creating the final Recommendation table in 3NF.

Status: After this process, all partial dependencies are removed. All tables are in 2NF.

Third Normal Form (3NF)

Aim: Data must be in 2NF and remove **transitive dependencies**. A transitive dependency exists when a non-key attribute depends on another non-key attribute, rather than directly on the primary key.

- **Users (UserID (PK), FirstName, LastName, Email, ...):** No non-key attribute (e.g., FirstName) determines another non-key attribute (e.g., Email). So this table is in **3NF**.
- **Device (DeviceID (PK), Model, DeviceName):** Model does not determine DeviceName or vice versa. This is in 3NF.
- **HealthMetric (MetricID (PK), MetricName, Unit):** MetricName does not determine Unit independently of the key. Hence, this is in 3NF. (Technically, MetricName determines Unit, but MetricName is a candidate key, so this doesn't violate 3NF).
- **HealthData (DataID (PK), ...):** The only non-key attributes are Timestamp and Value, which depend directly on the key. This is in **3NF**.

- **Recommendation (RecommendationID (PK), Title, Description, UserID (FK)):** Title and Description are dependent on the RecommendationID, not on each other or other non-key attributes. This is in **3NF**.

Results

- **Users** (Stores user info)
- **Device** (Stores device info)
- **HealthMetric** (Defines the types of metrics)
- **Recommendation** (Stores user-specific advice)
- **HealthData** (Acts as a junction table, linking users, devices, and metrics to store the actual data points)

After normalization, each table serves a single purpose, data redundancy is minimized, and the data integrity is protected from anomalies.

Database design continuation

Since we now understand how we come up with these components lets explain them

Database Schema

USER

- UserID: Primary Key (PK).
- Email: User's email address.
- Name: User's full name.
- Age: User's age.
- Gender: User's gender.
- RegistrationDate: The date the user registered.

DEVICE

- DeviceID: Primary Key (PK). A unique identifier for each device.
- Model: The model of the device (e.g., "Fitbit Charge 5").

- DeviceName: A user-friendly name for the device.

- **HEALTHMETRIC**
- MetricID: Primary Key (PK). A unique identifier for each type of metric.
- MetricName: The name of the metric (e.g., "Heart Rate", "Steps", "Blood
- Pressure").
- Unit: The unit of measurement for the metric (e.g., "bpm", "steps"

HEALTHDATA

- DataID: Primary Key (PK). A unique identifier for each data log entry.
- Timestamp: The date and time the data was logged.
- Value: The measured value of the metric (e.g., 85 for heart rate, 10000 for steps).
- DeviceID: Foreign Key (FK) referencing DEVICE. Links a data entry to the device that logged it.
- UserID: Foreign Key (FK) referencing USER. Links a data entry to the user it belongs to.
- MetricID: Foreign Key (FK) referencing HEALTHMETRIC. Links a data entry to the type of metric it represents

RECOMMENDATION

- RecommendationID: Primary Key (PK). A unique identifier for each recommendation.
- Title: The title of the recommendation.
- Description: A detailed description of the recommendation.
- UserID: Foreign Key (FK) referencing USER. This links a recommendation to a specific user.

Entity Relationship Analysis

Overview of Relationships

The schema connects key entities related to users, wearable devices, health data, and recommendations. Below is a structured explanation of each relationship

1.USER⟶DEVICE(One-to-Many)
One USER can use multiple DEVICES, or none.

Each device is used by one and only one USER.
This suggests ownership or exclusive usage of a device by a user.

2. USER ⟶ RECOMMENDATION (One-to-Many)

A single User can receive zero or many recommendations.

Each recommendation must be for one and only one user. This means the user may receive multiple personalized health recommendations over time.

3. DEVICE ⟶ HEALTHDATA (One-to-Many)

A device can log many healthdata entries. Each healthdata record is from one and only one device.
This means a single device can generate many health data records as it continuously collects metrics

4. HEALTHMETRIC ⟶ HEALTHDATA (One-to-Many)

Each Healthmetric (like heart rate, temperature, etc.) can be associated with many healthdata records. This means Health metrics define what type of data was collected (e.g., blood pressure).

5. RECOMMENDATION ⟶ DEVICE (Many-to-One)

RECOMMENDATIONs can be generated from one DEVICE. This implies that recommendations can be device-specific — perhaps generated based on the data that device collected.

In Summary Health data is collected by a device used by a user, measured according to a specific health metric, stored in the HealthData table, and then used to generate personalized recommendations for that user.

Future Improvements

- **Smarter Recommendations with Machine Learning**

Upgrade the recommendation engine to use machine learning so it can give more accurate and personalized health advice by learning from user data and behavior over time.

- **User Feedback on Recommendations**

Allow users to give feedback (like thumbs up/down) on the health tips they receive. This helps the system learn what works and improve future suggestions.

- **Support More Devices and Health Metrics**

Connect the system to more types of health-tracking devices (like different smartwatches or medical wearables) and track more health indicators like blood oxygen, stress, or ECG.

- **Build a Visual Dashboard for Users**

Create a simple, interactive dashboard where users can see graphs and trends of their health data (e.g., sleep, heart rate, activity) over time.

Conclusion

The **Wearable Health Device Data Hub** offers a robust and scalable platform for collecting, managing, and analyzing health data from various wearable devices. Through a carefully normalized database design, it ensures data integrity, eliminates redundancy, and supports efficient processing of large volumes of user-generated health metrics.

By leveraging structured relationships between users, devices, metrics, and recommendations, the system not only provides personalized insights but also creates opportunities for real-time health monitoring and early anomaly detection. Key features such as recommendation generation, user-specific analytics, and data logging form the foundation of a powerful health management tool.

Looking ahead, enhancements such as machine learning–based recommendations, user feedback integration, support for a wider range of health devices, and intuitive data visualization will further strengthen the platform. These improvements aim to deliver smarter, more personalized, and user-friendly health experiences—enabling users to take proactive control of their well-being through technology-driven insights.

Ultimately, the system demonstrates how thoughtful database design, combined with modern analytics, can transform raw health data into meaningful, actionable guidance for better health outcomes.