**Analysis and Documentation of the Prisma Schema**

This document provides a comprehensive analysis of a healthcare-related database schema using Prisma. It includes explanations of tables, an Entity-Relationship Diagram (ERD), and detailed documentation for each entity. Graphical representations are included to enhance clarity.

**1. Explanation of Tables**

**Main Entities**

**User**

Represents the users of the system, which can be pharmacy owners, medicine companies, or regular users based on the role field.

**Relationships:**

* A pharmacy can have one user (Pharmacy table).
* Users can create orders, reviews, search histories, and receive notifications.

**Medicine**

Represents the medicines available in the system.

**Fields:**

* name: Name of the medicine.
* active\_substance: Key ingredient.
* description: Detailed description of the medicine.

**Relationships:**

* Related to **Category** via the MedicineCategory table.
* A pharmacy's inventory stores medicines (**PharmacyInventory**).
* Associated with **Disease** (via DiseaseMedicine) and **Review**.

**Pharmacy**

Represents pharmacies in the system.

**Fields:**

* name: Pharmacy name.
* location: Address or geolocation of the pharmacy.

**Relationships:**

* Associated with a single **User**.
* Maintains an inventory (**PharmacyInventory**) and fulfills orders (**Order**).
* Collects reviews (**Review**).

**Category**

Organizes medicines into groups.

**Relationships:**

* Many-to-many relationship with **Medicine** through the MedicineCategory table.

**Order**

Tracks orders placed by users at pharmacies.

**Fields:**

* total\_price: Total price of the order.
* status: Pending, completed, or cancelled.

**Relationships:**

* Linked to OrderItem for details about each medicine in the order.
* Associated with a single **User** and a single **Pharmacy**.

**Linking Tables**

**MedicineCategory**

Bridges **Medicine** and **Category** in a many-to-many relationship.

**PharmacyInventory**

Links **Pharmacy** and **Medicine**, allowing pharmacies to manage their stocks and prices.

**DiseaseMedicine**

Bridges **Disease** and **Medicine** in a many-to-many relationship.

**Other Supporting Tables**

**Review**

Tracks user reviews for pharmacies and medicines.

**Relationships:**

* Optional association with a pharmacy or medicine.

**SearchHistory**

Tracks user searches, including the query and result count. Useful for analytics and personalizing suggestions.

**Notification**

Tracks notifications sent to users, marking them as read or unread.

**Disease**

Represents diseases that medicines can treat.

**Relationships:**

* Many-to-many relationships with **Medicine**.

**2. Entity-Relationship Diagram (ERD)**

Below is a graphical representation of the schema relationships.

**Relationships Overview:**

* **User ↔ Pharmacy**: One-to-One
* **User ↔ Order**: One-to-Many
* **User ↔ Review**: One-to-Many
* **User ↔ Notification**: One-to-Many
* **User ↔ SearchHistory**: One-to-Many
* **Pharmacy ↔ PharmacyInventory**: One-to-Many
* **Pharmacy ↔ Order**: One-to-Many
* **Pharmacy ↔ Review**: One-to-Many
* **Medicine ↔ Review**: One-to-Many
* **Medicine ↔ Disease**: Many-to-Many (via DiseaseMedicine)
* **Medicine ↔ Category**: Many-to-Many (via MedicineCategory)
* **Medicine ↔ PharmacyInventory**: One-to-Many
* **Order ↔ OrderItem**: One-to-Many

**3. Detailed Documentation**

**User**

* **Primary Key:** user\_id
* **Fields:**
  + email (unique): User email.
  + password\_hash: Encrypted password.
  + role (enum): Determines if the user is a pharmacy, company, or regular user.
* **Relationships:**
  + One-to-One with **Pharmacy**.
  + One-to-Many with **Order**, **Review**, **SearchHistory**, and **Notification**.

**Medicine**

* **Primary Key:** medicine\_id
* **Fields:**
  + name (unique): Name of the medicine.
  + active\_substance: Key ingredient.
  + description: Detailed description.
* **Relationships:**
  + Many-to-Many with **Category**.
  + Many-to-Many with **Disease**.
  + One-to-Many with **PharmacyInventory**.
  + One-to-Many with **Review**.

**Pharmacy**

* **Primary Key:** pharmacy\_id
* **Fields:**
  + name: Pharmacy name.
  + location: Address/geolocation of the pharmacy.
* **Relationships:**
  + One-to-One with **User**.
  + One-to-Many with **PharmacyInventory** and **Order**.
  + One-to-Many with **Review**.

**Order**

* **Primary Key:** order\_id
* **Fields:**
  + total\_price: Total price of the order.
  + status (enum): Pending, completed, or cancelled.
* **Relationships:**
  + One-to-Many with **OrderItem**.

**OrderItem**

* **Primary Key:** order\_item\_id
* **Fields:**
  + quantity: Number of units.
  + price: Price per unit.
* **Relationships:**
  + Many-to-One with **Order**.

**Category**

* **Primary Key:** category\_id
* **Fields:**
  + name (unique): Category name.
* **Relationships:**
  + Many-to-Many with **Medicine**.

**Disease**

* **Primary Key:** disease\_id
* **Fields:**
  + name (unique): Disease name.
* **Relationships:**
  + Many-to-Many with **Medicine**.

**Review**

* **Primary Key:** review\_id
* **Fields:**
  + rating: Rating out of 5.
  + comment: Review comment.
* **Relationships:**
  + Many-to-One with **Pharmacy** and **Medicine**.

**PharmacyInventory**

* **Primary Key:** inventory\_id
* **Fields:**
  + stock\_quantity: Number of items in stock.
  + price: Price per unit.
* **Relationships:**
  + Many-to-One with **Pharmacy**.

**Conclusion**

This document outlines a robust schema designed for managing healthcare-related data efficiently. Each table and relationship is tailored to support core functionalities such as user management, medicine tracking, and pharmacy operations. The inclusion of many-to-many linking tables ensures scalability and flexibility in handling complex associations, while the detailed documentation facilitates understanding and implementation.