

Project Track 1, Stage 2 Entity-Relationship Diagram

1. Here is our Entity-Relationship (ER) diagram:

We have submitted our ER diagram under the doc folder in Github.

2. Entities and assumptions:

1) User:

Attributes: Email, Address, City, State

A “user” represents an individual who uses this application. Each user is uniquely identified by their email address. Address-related attributes are stored under User because they describe where the user lives and are not an entity of their own.

Users must be modeled as an entity because they interact with multiple parts of the system, as they have symptoms, preferred pharmacies, etc.

2) Pharmacy:

Attributes: ID, Address, City, State

A “pharmacy” represents a store that provides medicines. Each pharmacy is uniquely identified by its ID (taken from the pharmacies database).

Location-related attributes are included to allow for users to find pharmacies near them.

Pharmacy is modeled as an entity because it can exist independently, may be preferred by multiple users, and carries multiple medicines.

3) Medicine:

Attributes: Name, Manufacturer, Rating, Composition

A “medicine” represents a drug that can treat one or more symptoms. Each medicine is uniquely identified by its name. Manufacturer, composition, and rating provide information about the medicine’s origin and effectiveness.

Medicine is modeled as an entity because it forms 1-to-many/many-to-many relationships with symptoms, pharmacies, and side effects. Making it a separate entity allows for us to more easily represent these relationships/make it easier to scale.

4) Side effect:

Attributes: Name, Description

A “side effect” represents a negative effect that may result from taking a certain

medicine. Each side effect is uniquely identified by its name.

Side effect is modeled as an entity because a side effect can be caused by many medicines, and many medicines can cause many side effects. It is more effective to model side effect as an entity because of this many-to-many relationship.

5) Symptom:

Attributes: Name, Description

A "symptom" represents a health issue that a user is currently facing. The description explains what the symptom entails.

Symptom is modeled as an entity because a symptom can affect many users, and a user can experience many symptoms. It is more effective to model symptom as an entity because of this many-to-many relationship.

Relationships and assumptions:

1) Prefers (User/Pharmacy):

This is a 1-to-many relationship because our website will only allow users to prefer up to one pharmacy, but a pharmacy can be preferred by many users.

2) Currently Has (User/Symptom):

A user can have multiple symptoms at once, and an individual symptom can occur in many users. This is a many-to-many relationship because symptoms are not unique to individual users.

3) Treated By (Symptom/Medicine):

This is a many-to-many relationship because a symptom can be treated by multiple medicines, and a medicine can treat multiple symptoms.

4) Caused by (Side Effect/Medicine):

This is a many-to-many relationship because a side effect can be caused by multiple medicines, and a single medicine can cause multiple side effects

5) Carries (Pharmacy/Medicine):

This is a many-to-many relationship because a pharmacy can carry multiple medicines, and each medicine can be available at multiple pharmacies.

3. We have 5 entities and 2 types of relationships:

Our 5 entities are:

1. User
2. Pharmacy
3. Medicine
4. Symptoms
5. Side Effects

Our 2 types of relationships are:

1. Many-to-many (user-symptom, symptom-medicine, medicine-side effect, pharmacy-medicine)
2. 1-to-many (user-pharmacy)

4. Normalization

Relation	Functional Dependencies
User	<u>Email</u> → Address, City, State
Symptom	<u>Name</u> → Description
Medicine	<u>Name</u> → Manufacturer, Rating, Composition
Pharmacy	<u>ID</u> → Address, City, State
SideEffect	<u>Name</u> → Description
Carries	<u>(PharmacyID, MedicineName)</u> → ∅

CausedBy (MedicineName, SideEffectName) → ∅

TreatedBy (SymptomName, MedicineName) → ∅

CurrentlyHas (UserEmail, SymptomName) → ∅

Every functional dependency in our relations has the left-hand side to be a super key, so they are already in BCNF.

5. Conversion to relational schema:

Entities:

User(Email: VARCHAR(255) [PK],
Address: VARCHAR(255),
City: VARCHAR(255),
State: VARCHAR(255))

Symptom(Name: VARCHAR(255) [PK],
Description VARCHAR(255))

Medicine(Name: VARCHAR(255) [PK],
Manufacturer: VARCHAR(255),
Rating: DECIMAL,
Composition: VARCHAR(255))

Pharmacy(ID: INT [PK],
Address: VARCHAR(255),
City: VARCHAR(255),
State: VARCHAR(255))

SideEffect(Name: VARCHAR(255) [PK],
Description: VARCHAR(255))

Relationships:

Carries(PharmacyID: INT [PK, FK to Pharmacy.ID] ,
MedicineName: VARCHAR(255) [PK, FK to Medicine.Name])

CausedBy(MedicineName: VARCHAR(255) [PK, FK to Medicine.Name],
SideEffectName: VARCHAR(255) [PK, FK to SideEffect.Name])

TreatedBy(SymptomName: VARCHAR(255) [PK, FK to Symptom.Name],
MedicineName: VARCHAR(255) [PK, FK to Medicine.Name])

CurrentlyHas(UserEmail: VARCHAR(255) [PK, FK to User.Email],
SymptomName: VARCHAR(255) [PK])