

Relationship Schema:

Condition(trackable_id:INT [FK to Trackable.trackable_id],
trackable_name:VARCHAR(64), trackable_value:VARCHAR(20))

Symptom(trackable_id:INT [FK to Trackable.trackable_id],
trackable_name:VARCHAR(64), trackable_value:VARCHAR(20))

Treatment(trackable_id:INT [FK to Trackable.trackable_id],
trackable_name:VARCHAR(64), trackable_value:VARCHAR(20))

Tag(trackable_id:INT [FK to Trackable.trackable_id],
trackable_name:VARCHAR(64))

Weather(trackable_id:INT [FK to Trackable.trackable_id],
trackable_name:VARCHAR(64), trackable_value:Decimal)

Trackable(trackable_id:INT [PK], trackable_type:VARCHAR(64),
checkin_date:DATE)

User(user_id:VARCHAR(32) [PK], username:VARCHAR(32),
password:VARCHAR(32), email:VARCHAR(32))

Demographics(user_id:VARCHAR(32) [FK to User.user_id], age:INT,
sex:VARCHAR(32), country:VARCHAR(32))

UserTracks(trackable_id:INT [FK to Trackable.trackable_id],
user_id:VARCHAR(32) [FK to User.user_id])

Entity Description & Assumptions:

Condition(trackable_id, trackable_name, trackable_value)

- The Condition table keeps track of all existing conditions that users input into our app. Condition examples include: Ehlers-Danlos Syndrome, Ovarian Cysts, and other diagnoses. This table comes pre-populated with the symptoms existing in the dataset. We are assuming that conditions will have a name and a value (not null). We are also assuming that one user can have 0 to many conditions associated with the user, and

that a condition does not need a user to exist as a valid condition to have. We are assuming that a condition cannot exist without a valid trackable id.

Symptom(trackable_id, trackable_name, trackable_value)

- The Symptom table keeps track of all existing symptoms that users input into our app. Symptom examples include: stomach cramps, fatigue, and other symptoms. This table comes pre-populated with the symptoms existing in the dataset. We are assuming that symptoms will have a name and a value (not null). We are also assuming that one user can have 0 to many symptoms associated with the user, and that a symptom does not need a user to exist as a valid symptom to have. We are assuming that a symptom cannot exist without a valid trackable id.

Treatment(trackable_id, trackable_name, trackable_value)

- The Treatment table keeps track of all existing treatments that users input into our app. Treatment examples include: Zofran, Klonopin. This table comes pre-populated with the treatments existing in the dataset. We are assuming that treatments will have a name and a value (not null). We are also assuming that one user can have 0 to many treatments associated with the user, and that a treatment does not need a user to exist as a valid condition to have. We are assuming that a treatment cannot exist without a valid trackable id.

Tag(trackable_id, trackable_name)

- The Tag table keeps track of all existing treatments that users input into our app. Tag examples include: "ate breakfast", "good sleep", "comcast >:(". This table comes pre-populated with the tags existing in the dataset. We are assuming that tags will have a name (not null). We are also assuming that one user can have 0 to many tags associated with the user, and that a tag does not need a user to exist as a valid condition to have. We are assuming that a tag cannot exist without a valid trackable id.

Weather(trackable_id, trackable_name, trackable_value)

- The weather table keeps track of the current weather conditions when the user inputted their trackable. The trackable_name holds the current weather condition (ex: rainy, sunny, etc). We are assuming that weather cannot exist without a valid trackable id.

Trackable(trackable_id, trackable_type, checkin_date)

- The trackable table keeps track of the trackable aspects about each user (symptom, treatment, tag etc). These are identified by the trackable_id and trackable_type identifies the aspect. We are assuming that a trackable can exist independently of the user and vice versa.

User(user_id, username, password, email)

- The user table keeps track of user login information and profile data. We are assuming that a user can have 0 to many trackables, and a trackable can have 0 to many users.

Demographics(user_id, age, sex, country)

- The demographics table keeps track of user demographics such as age, gender, and nationality. We are representing the demographics table as a weak entity, and are assuming that a demographic relation cannot exist without a user.

UserTracks(user_id, trackable_id)

- The UserTracks table represents a relationship between the User and the Trackable tables. This table keeps track of what trackables a user logs. We are assuming that one user can have 0 to many trackables, and one trackable can have 0 to many users (many-to-many relationship).

Diagram:

