

Entities

User: This entity represents users of the application who create and manage travel itineraries. This must be an entity because it doesn't make sense to make it an attribute on another entity, because it is important to distinguish users.

- Assumption: Each user has personal preferences (food, nature, culture, activity) stored as numeric values on a scale to indicate preference strength.

Itinerary: This entity represents a travel plan for a specific destination and time period. This must be an entity because a single user can have multiple itineraries.

- Assumption: Each itinerary is for a specific destination with a specific start and end date

ItineraryItem: This represents a specific attraction scheduled in an itinerary. This must be its own entity because an Itinerary will have multiple ItineraryItems

- Assumption: Users need to specify not just what attractions to visit but also when to visit them. An itinerary will consist of multiple attractions

Attraction: This entity represents tourist attractions that can be included in itineraries. This is mapped to an ItineraryItem because you can have the same attraction multiple times in one Itinerary at different times.

- Assumption: Attractions should have properties like price, ratings, and categories that help match them to user preferences.

Route: This entity represents travel paths between itinerary items. This must be its own entity because it connects two or more different ItineraryItems

- Assumption: A user may have multiple routes to get between two ItineraryItems depending on mode of travel, traffic, etc.

Relationships

User - Itinerary (Plans): A user can make many itineraries, but an itinerary will always belong to one user

- Cardinality: One-to-many

Itinerary - ItineraryItem (Consists of): An Itinerary will consist of many ItineraryItems with their own start and end times.

- Cardinality: One-to-many

ItineraryItem - Attraction (Describes): An ItineraryItem describes a single attraction.

However, one attraction may be on an Itinerary multiple times.

- Cardinality: Many-to-one

ItineraryItem - Route (Takes): Two ItineraryItems must be connected through a Route. There also may be multiple routes between ItineraryItems

- Cardinality: One-to-many

Normalization to BCNF

User Table FDs:

- $\text{user_id} \rightarrow \text{name, email, password, food_pref, nature_pref, culture_pref, activity_pref}$

Since user_id is the only determinant and it is the primary key, this table is in BCNF

Itinerary Table FDs:

- $\text{itinerary_id} \rightarrow \text{user_id, destination, start_date, end_date, start_time, end_time}$

Since itinerary_id is the only determinant and it is the primary key, this table is in BCNF

ItineraryItem Table FDs:

- $\text{item_id} \rightarrow \text{itinerary_id, attraction_id, day_number, start_time, end_time, notes}$

Since item_id is the only determinant and it is the primary key, this table is in BCNF

Attraction Table FDs:

- $\text{attraction_id} \rightarrow \text{name, description, location, rating, popularity, price_level, category}$

Since attraction_id is the only determinant and it is the primary key, this table is in BCNF

Route Table FDs:

- $\text{route_id} \rightarrow \text{from_item_id, to_item_id, distance, travel_time, transport_mode, price_level}$
- $(\text{from_item_id, to_item_id, transport_mode}) \rightarrow \text{distance, travel_time, price_level}$

route_id is the primary key

$(\text{from_item_id, to_item_id, transport_mode})$ We will assume that the same transport mode between two same items will be the exact same, which makes this a candidate key. Therefore this table is in BCNF.

Relational Schema

```
User(  
  user_id:INT [PK],  
  name:VARCHAR,  
  email:VARCHAR,  
  password:VARCHAR,  
  food_pref:INT,  
  nature_pref:INT,  
  culture_pref:INT,  
  activity_pref:INT  
)
```

```
Itinerary(  
  itinerary_id:INT [PK],  
  user_id:INT [FK to User.user_id],  
  destination:VARCHAR,  
  start_date:DATE,  
  end_date:DATE,  
  start_time:DATETIME,  
  end_time:DATETIME  
)
```

```
ItineraryItem(  
  item_id:INT [PK],  
  itinerary_id:INT [FK to Itinerary.itinerary_id],  
  attraction_id:INT [FK to Attraction.attraction_id],  
  day_number:INT,  
  start_time:DATETIME,  
  end_time:DATETIME,  
  notes:TEXT  
)
```

```
Attraction(  
attraction_id:INT [PK],  
name:VARCHAR,  
description:TEXT,  
location:LOCATION,  
rating:DECIMAL,  
popularity:INT,  
price_level:INT,  
category:ENUM  
)
```

```
Route(  
route_id:INT [PK],  
from_item_id:INT [FK to ItineraryItem.item_id],  
to_item_id:INT [FK to ItineraryItem.item_id],  
distance:DECIMAL,  
travel_time:INT,  
transport_mode:VARCHAR,  
price_level:INT  
)
```