GitHub ↗

KARSAFAR PROGRESS

- ## Initial Database Schema:

1. **User** - userId, firstName, lastName, phoneNo, Email, Password, Street, City, State, Pincode, Country, picture
2. **Booking** - bookingId, price, status, userId, paymentId, paymentMethod, tripId, vehicleItemId, accomodationItemId
3. **VehicleItem** - vehicleItemId, vehicleId, onboardingLocation, deboardingLocation, onboardingTime, deboardingTime, coachType, seatId
4. **Seat** - vehicleId, seatId, seatNumber, name, age, gender, food
5. **VechicleItemCoach** - vehicleItemCoachId, coachType, noOfSeats
6. **Station** - vehicleId, stationName, time, stoppage
7. **Coach** - vehicleId, coachType, seatsAvailable, price
8. **Train** - vehicleId, trainName
9. **Flight** - vehicleId, flightName, photo
10. **Bus** - vehicleId, busName, driverName, driverPhoneNo, photo
11. **Cab** - vehicleId, driverName, driverPhoneNo, carModel, photo
12. **Cruise** - vehicleId, cruiseName, photo
13. **Review** - itemId, reviewId, comment, rating
14. **Pictures** - accomId, picture
15. **AccomItem** - accomItemId, accomId, checkInDate, checkoutDate, name , phoneNo, email
16. **AccomItemRoom** - accomItemId, accomId, roomType, roomNumber
17. **Hotel** - accomId, name, phoneNo, email, Street, City, State, Pincode, Country, description, breakfastInc, AC/non-AC
18. **Rooms** - accomId, roomType, roomsAvailable, pplAccomodated, roomDescr // rooom type should inc AC, non AC, suite ke b type, bunk bed
19. **Airbnb** - accomId, name, phoneNo, email, Street, landmark, City, State, Pincode, Country, description, maxAllowedGuests
20. **AirbnbAmen** - accomId, amenityType

NORMALIZED DATABASE FINAL SCHEMA IN BCNF:

1. **User Management:**

a) **User** (userId PK, firstName, lastName, phoneNo, email, password, profilePicture)
b) **UserAddress** (addressId PK, userId FK, street, city, state, pinCode, country)

2. **Vehicle Management:**

a) **Vehicle** (vehicleId PK, vehicleType, status, availableSeats)
b) **Train** (vehicleId PK/FK, trainName)
c) **Flight** (vehicleId PK/FK, flightName)
d) **Bus** (vehicleId PK/FK, busName, photo)
e) **Cab** (vehicleId PK/FK, carModel, photo)
f) **Cruise** (vehicleId PK/FK, cruiseName, photo)
g) **VehicleDriver** (driverId PK, vehicleId FK, driverName, driverPhoneNo)
h) **VehicleCoach** (coachId PK, vehicleId FK, coachType, seatsAvailable, price)
i) **VehicleStation** (stationId PK, vehicleId FK, stationName, arrivalTime, departureTime, stoppage, stationOrder)
j) **Seat** (seatId PK, vehicleId FK, coachId FK, seatNumber)

3. **Accommodation Management:**

a) **Accommodation** (accomId PK, accomType, name, phoneNo, email, description)
b) **AccommodationAddress** (addressId PK, accomId FK, street, landmark, city, state, pinCode, country)
c) **AccommodationPhoto** (photoId PK, accomId FK, photoUrl)
d) **AccommodationAmenity** (amenityId PK, amenityType)
e) **AccomAmenityMap** (accomId FK, amenityId FK, PK(accomId, amenityId))
f) **Hotel** (accomId PK/FK, breakfastIncluded, acType)

g) **Airbnb** (accomId PK/FK, maxAllowedGuests)

h) **Room** (roomId PK, accomId FK, roomType, roomsAvailable, pplAccommodated, roomDescription, price)

4. Booking Management:

a) **Trip** (tripId PK, userId FK, name, startDate, endDate, status)

b) **VehicleBookingItem** (vehicleItemId PK, vehicleId FK, onboardingLocation, deboardingLocation, onboardingTime, deboardingTime, coachType, price, status)

c) **AccommodationBookingItem** (accomItemId PK, accomId FK, checkInDate, checkOutDate, contactName, contactPhoneNo, contactEmail, price, status)

d) **AccomBookingRoom** (bookingRoomId PK, accomItemId FK, roomId FK, roomNumber)

e) **PassengerSeat** (passengerId PK, vehicleItemId FK, seatId FK, name, age, gender, foodPreference)

f) **Booking** (bookingId PK, userId FK, tripId FK, totalPrice, status, createDate)

g) **BookingItem** (bookingItemId PK, bookingId FK, itemType, vehicleItemId FK, accomItemId FK, price)

h) **Payment** (paymentId PK, bookingId FK, amount, paid, paymentMethod, transactionId, paymentDate, status)

5. Reviews:

a) **Review** (reviewId PK, userId FK, itemType, itemId, rating, comment, reviewDate)

Functional Dependencies in the Trip Database Schema

1. User Management:

a) User:

i. userId → firstName, lastName, phoneNo, email, password, profilePicture
ii. email → userId (assuming email is unique)
iii. UserAddress:
iv. addressId → userId, street, city, state, pinCode, country
v. (userId, addressId) → street, city, state, pinCode, country

2. Vehicle Management:

a) Vehicle:

i. vehicleId → vehicleType, status, availableSeats
ii. Train/Flight/Bus/Cab/Cruise (Inheritance):
iii. vehicleId → all attributes specific to that vehicle type
iv. For Train: vehicleId → trainName
v. For Flight: vehicleId → flightName
vi. For Bus: vehicleId → busName, photo
vii. For Cab: vehicleId → carModel, photo
viii. For Cruise: vehicleId → cruiseName, photo

b) VehicleDriver:

i. driverId → vehicleId, driverName, driverPhoneNo
ii. vehicleId → driverId (assuming one driver per vehicle)
iii. VehicleCoach:
iv. coachId → vehicleId, coachType, seatsAvailable, price
v. (vehicleId, coachType) → coachId, seatsAvailable, price

c) **VehicleStation:**

   i. stationId → vehicleId, stationName, arrivalTime, departureTime, stoppage, stationOrder

   ii. (vehicleId, stationOrder) → stationId, stationName, arrivalTime, departureTime, stoppage

d) **Seat:**

   i. seatId → vehicleId, coachId, seatNumber

   ii. (vehicleId, coachId, seatNumber) → seatId

3. Accommodation Management:

a) **Accommodation:**

   i. accomId → accomType, name, phoneNo, email, description

b) **AccommodationAddress:**

   i. addressId → accomId, street, landmark, city, state, pinCode, country

   ii. accomId → addressId (assuming one address per accommodation)

c) **AccommodationPhoto:**

   i. photoId → accomId, photoUrl

d) **AccommodationAmenity:**

   i. amenityId → amenityType

   ii. AccomAmenityMap (M:M relationship):

   iii. (accomId, amenityId) → No other attributes (just the relationship)

e) **Hotel/Airbnb (Inheritance):**

   i. accomId → attributes specific to that accommodation type

   ii. For Hotel: accomId → breakfastIncluded, acType

   iii. For Airbnb: accomId → maxAllowedGuests

f) **Room:**

   i. roomId → accomId, roomType, roomsAvailable, pplAccommodated, roomDescription, price

   ii. (accomId, roomType) → roomsAvailable, pplAccommodated, roomDescription, price

4. Booking Management:

a) Trip:

i. tripId → userId, name, startDate, endDate, status

b) VehicleBookingItem:

i. vehicleItemId → vehicleId, onboardingLocation, deboardingLocation, onboardingTime, deboardingTime, coachType, price, status

c) AccommodationBookingItem:

i. accomItemId → accomId, checkInDate, checkOutDate, contactName, contactPhoneNo, contactEmail, price, status

d) AccomBookingRoom:

i. bookingRoomId → accomItemId, roomId, roomNumber

ii. (accomItemId, roomId) → roomNumber

e) PassengerSeat:

i. passengerId → vehicleItemId, seatId, name, age, gender, foodPreference

ii. (vehicleItemId, seatId) → passengerId, name, age, gender, foodPreference

f) Booking:

i. bookingId → userId, tripId, totalPrice, status, createDate

g) BookingItem:

i. bookingItemId → bookingId, itemType, vehicleItemId, accomItemId, price

ii. Note: vehicleItemId and accomItemId will be NULL depending on itemType

h) Payment:

i. paymentId → bookingId, amount, paid, paymentMethod, transactionId, paymentDate, status

ii. bookingId → paymentId (assuming one payment per booking)

5. Reviews

a) Review:

i. reviewId → userId, itemType, itemId, rating, comment, reviewDate

ii. (userId, itemType, itemId) → reviewId, rating, comment, reviewDate

SQL QUERIES

```
CREATE TABLE users (
    userId BINARY(16) PRIMARY KEY,
    firstName VARCHAR(50) NOT NULL,
    lastName VARCHAR(50),
    phoneNo VARCHAR(15) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    profilePicture VARCHAR(255)
);
CREATE TABLE userAddresses (
    addressId BINARY(16) PRIMARY KEY,
    userId INT NOT NULL,
    street VARCHAR(100),
    city VARCHAR(50) NOT NULL,
    state VARCHAR(50),
    pinCode VARCHAR(10),
    country VARCHAR(50) NOT NULL,
    FOREIGN KEY (userId) REFERENCES users(userId) ON DELETE CASCADE
);
```

```
CREATE TABLE vehicles (
    vehicleId BINARY(16) PRIMARY KEY,
    vehicleType ENUM('train', 'flight', 'bus', 'cab', 'cruise') NOT NULL,
    status ENUM('active', 'maintenance', 'cancelled') DEFAULT 'active',
    availableSeats INT NOT NULL
);
CREATE TABLE trains (
    vehicleId BINARY(16) KEY,
    trainName VARCHAR(100) NOT NULL,
    FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);
CREATE TABLE flights (
    vehicleId BINARY(16) PRIMARY KEY,
    flightName VARCHAR(100) NOT NULL,
    FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);
CREATE TABLE buses (
    vehicleId BINARY(16) PRIMARY KEY,
    busName VARCHAR(100) NOT NULL,
    photo VARCHAR(255),
    FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);
```

```
CREATE TABLE cabs (
    vehicleId BINARY(16) PRIMARY KEY,
    carModel VARCHAR(100) NOT NULL,
    photo VARCHAR(255),
    FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);
CREATE TABLE cruises (
    vehicleId BINARY(16) PRIMARY KEY,
    cruiseName VARCHAR(100) NOT NULL,
    photo VARCHAR(255),
    FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);
CREATE TABLE vehicleDrivers (
    driverId BINARY(16) PRIMARY KEY,
    vehicleId INT NOT NULL,
    driverName VARCHAR(100) NOT NULL,
    driverPhoneNo VARCHAR(15) NOT NULL,
    FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);
```

```
CREATE TABLE vehicleCoaches (
    coachId BINARY(16) PRIMARY KEY,
    vehicleId INT NOT NULL,
    coachType VARCHAR(50) NOT NULL,
    seatsAvailable INT NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE,
    UNIQUE KEY unique_vehicle_coach (vehicleId, coachType)
);
CREATE TABLE vehicleStations (
    stationId BINARY(16) PRIMARY KEY,
    vehicleId INT NOT NULL,
    stationName VARCHAR(100) NOT NULL,
    arrivalTime DATETIME,
    departureTime DATETIME,
    stoppage INT, -- Duration in minutes
    stationOrder INT NOT NULL,
    FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE,
    UNIQUE KEY unique_vehicle_station_order (vehicleId, stationOrder)
);
```

```
CREATE TABLE seats (
    seatId BINARY(16) PRIMARY KEY,
    vehicleId INT NOT NULL,
    coachId INT NOT NULL,
    seatNumber VARCHAR(10) NOT NULL,
    FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE,
    FOREIGN KEY (coachId) REFERENCES vehicleCoaches(coachId) ON DELETE CASCADE,
    UNIQUE KEY unique_seat (vehicleId, coachId, seatNumber)
);
CREATE TABLE accommodations (
    accomId BINARY(16) PRIMARY KEY,
    accomType ENUM('hotel', 'airbnb') NOT NULL,
    name VARCHAR(100) NOT NULL,
    phoneNo VARCHAR(15) NOT NULL,
    email VARCHAR(100),
    description TEXT
);
```

```
CREATE TABLE accommodationAddresses (
   addressId BINARY(16) PRIMARY KEY,
   accomId INT NOT NULL,
   street VARCHAR(100) NOT NULL,
   landmark VARCHAR(100),
   city VARCHAR(50) NOT NULL,
   state VARCHAR(50),
   pinCode VARCHAR(10),
   country VARCHAR(50) NOT NULL,
   FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE
CASCADE
);
CREATE TABLE accommodationPhotos (
   photoId BINARY(16) PRIMARY KEY,
   accomId INT NOT NULL,
   photoUrl VARCHAR(255) NOT NULL,
   FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE
CASCADE
);
```

```
CREATE TABLE accommodationAmenities (
    amenityId BINARY(16) PRIMARY KEY,
    amenityType VARCHAR(50) NOT NULL UNIQUE
);
CREATE TABLE accomAmenityMap (
    accomId INT NOT NULL,
    amenityId INT NOT NULL,
    PRIMARY KEY (accomId, amenityId),
    FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE CASCADE,
    FOREIGN KEY (amenityId) REFERENCES accommodationAmenities(amenityId) ON DELETE CASCADE
);
CREATE TABLE hotels (
    accomId BINARY(16) PRIMARY KEY,
    breakfastIncluded BOOLEAN DEFAULT FALSE,
    acType ENUM('AC', 'NON-AC', 'BOTH') NOT NULL,
    FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE CASCADE
);
```

```
CREATE TABLE airbnbs (
   accomId BINARY(16) PRIMARY KEY,
   maxAllowedGuests INT NOT NULL,
   FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE CASCADE
);
CREATE TABLE rooms (
   roomId BINARY(16) PRIMARY KEY,
   accomId INT NOT NULL,
   roomType VARCHAR(50) NOT NULL,
   roomsAvailable INT NOT NULL,
   pplAccommodated INT NOT NULL,
   roomDescription TEXT,
   price DECIMAL(10,2) NOT NULL,
   FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE CASCADE
);
```

```
CREATE TABLE trips (
    tripId BINARY(16) PRIMARY KEY,
    userId INT NOT NULL,
    name VARCHAR(100) NOT NULL,
    startDate DATE NOT NULL,
    endDate DATE NOT NULL,
    status ENUM('planning', 'booked', 'ongoing', 'completed', 'cancelled') DEFAULT 'planning',
    FOREIGN KEY (userId) REFERENCES users(userId) ON DELETE CASCADE
);
CREATE TABLE vehicleBookingItems (
    vehicleItemId BINARY(16) PRIMARY KEY,
    vehicleId INT NOT NULL,
    onboardingLocation VARCHAR(100) NOT NULL,
    deboardingLocation VARCHAR(100) NOT NULL,
    onboardingTime DATETIME NOT NULL,
    deboardingTime DATETIME NOT NULL,
    coachType VARCHAR(50),
    price DECIMAL(10,2) NOT NULL,
    status ENUM('confirmed', 'pending', 'cancelled') DEFAULT 'pending',
    FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);
```

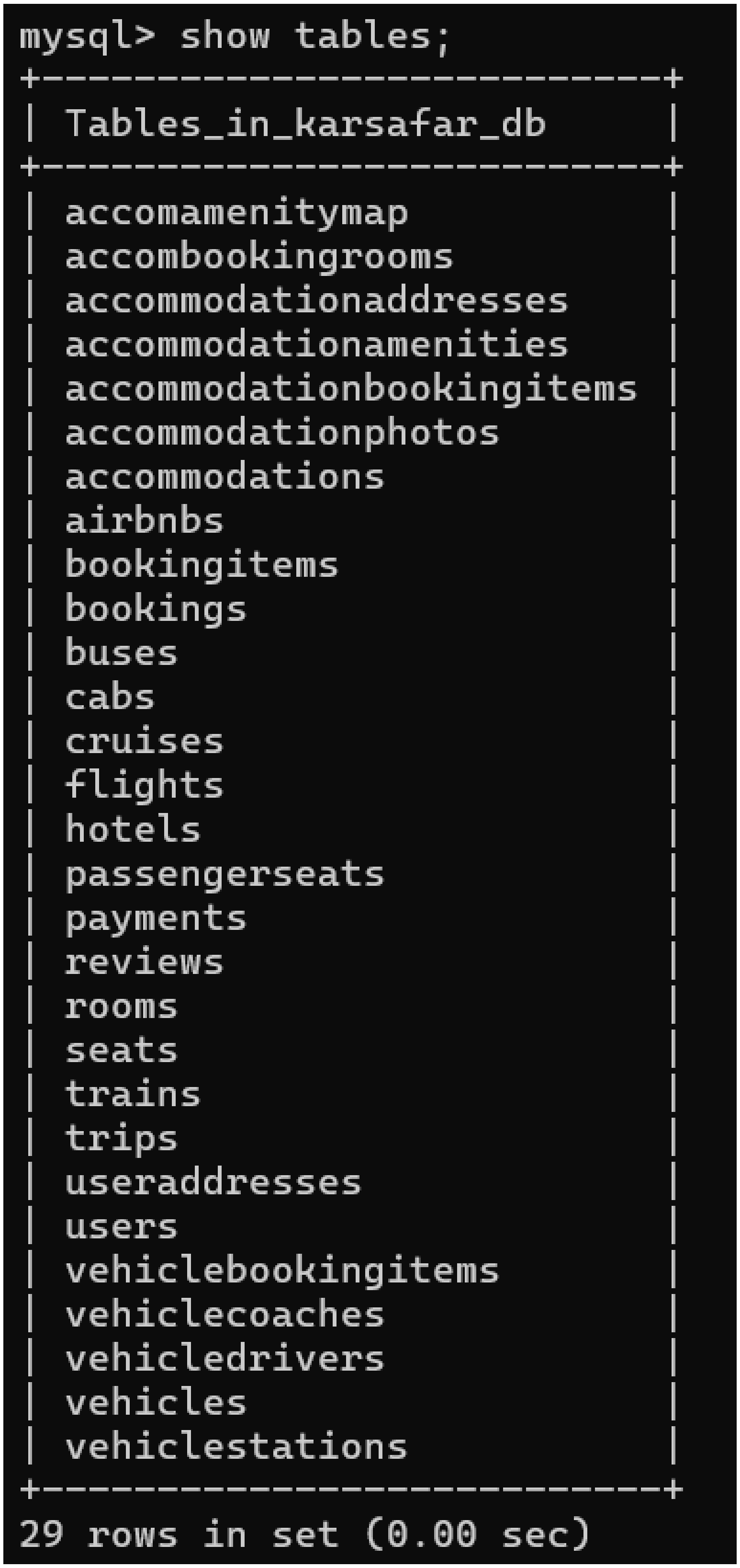
```
CREATE TABLE accommodationBookingItems (
   accomItemId BINARY(16) PRIMARY KEY,
   accomId INT NOT NULL,
   checkInDate DATE NOT NULL,
   checkOutDate DATE NOT NULL,
   contactName VARCHAR(100) NOT NULL,
   contactPhoneNo VARCHAR(15) NOT NULL,
   contactEmail VARCHAR(100),
   price DECIMAL(10,2) NOT NULL,
   status ENUM('confirmed', 'pending', 'cancelled') DEFAULT 'pending',
   FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE CASCADE
);
CREATE TABLE accomBookingRooms (
   bookingRoomId BINARY(16) PRIMARY KEY,
   accomItemId INT NOT NULL,
   roomId INT NOT NULL,
   roomNumber VARCHAR(20),
   FOREIGN KEY (accomItemId) REFERENCES accommodationBookingItems(accomItemId) ON DELETE CASCADE,
   FOREIGN KEY (roomId) REFERENCES rooms(roomId) ON DELETE CASCADE
);
```

```
CREATE TABLE passengerSeats (
   passengerId BINARY(16) PRIMARY KEY,
   vehicleItemId INT NOT NULL,
   seatId INT NOT NULL,
   name VARCHAR(100) NOT NULL,
   age INT NOT NULL,
   gender ENUM('male', 'female', 'other') NOT NULL,
   foodPreference ENUM('veg', 'non-veg', 'vegan', 'none') DEFAULT 'none',
   FOREIGN KEY (vehicleItemId) REFERENCES vehicleBookingItems(vehicleItemId) ON DELETE CASCADE,
   FOREIGN KEY (seatId) REFERENCES seats(seatId) ON DELETE CASCADE,
   UNIQUE KEY unique_vehicle_seat (vehicleItemId, seatId)
);
CREATE TABLE bookings (
   bookingId BINARY(16) PRIMARY KEY,
   userId INT NOT NULL,
   tripId INT,
   totalPrice DECIMAL(10,2) NOT NULL,
   status ENUM('confirmed', 'pending', 'cancelled') DEFAULT 'pending',
   createDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   FOREIGN KEY (userId) REFERENCES users(userId) ON DELETE CASCADE,
   FOREIGN KEY (tripId) REFERENCES trips(tripId) ON DELETE SET NULL
);
```

```
CREATE TABLE bookingItems (
    bookingItemId BINARY(16) PRIMARY KEY,
    bookingId INT NOT NULL,
    itemType ENUM('vehicle', 'accommodation') NOT NULL,
    vehicleItemId INT,
    accomItemId INT,
    price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (bookingId) REFERENCES bookings(bookingId) ON DELETE CASCADE,
    FOREIGN KEY (vehicleItemId) REFERENCES vehicleBookingItems(vehicleItemId) ON DELETE SET NULL,
    FOREIGN KEY (accomItemId) REFERENCES accommodationBookingItems(accomItemId) ON DELETE SET NULL,
    CHECK (
        (itemType = 'vehicle' AND vehicleItemId IS NOT NULL AND accomItemId IS NULL) OR
        (itemType = 'accommodation' AND accomItemId IS NOT NULL AND vehicleItemId IS NULL)
    )
);
```

```sql
CREATE TABLE payments (
   paymentId BINARY(16) PRIMARY KEY,
   bookingId INT NOT NULL,
   amount DECIMAL(10,2) NOT NULL,
   paid BOOLEAN DEFAULT FALSE,
   paymentMethod VARCHAR(50),
   transactionId VARCHAR(100),
   paymentDate TIMESTAMP,
   status ENUM('pending', 'completed', 'failed', 'refunded') DEFAULT 'pending',
   FOREIGN KEY (bookingId) REFERENCES bookings(bookingId) ON DELETE
CASCADE
);
CREATE TABLE reviews (
   reviewId BINARY(16) PRIMARY KEY,
   userId INT NOT NULL,
   itemType ENUM('vehicle', 'accommodation', 'trip') NOT NULL,
   itemId INT NOT NULL,
   rating DECIMAL(3,2) NOT NULL CHECK (rating BETWEEN 0 AND 5),
   comment TEXT,
   reviewDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   FOREIGN KEY (userId) REFERENCES users(userId) ON DELETE CASCADE,
   UNIQUE KEY unique_user_review (userId, itemType, itemId)
);
```

PROGRESS SCREENSHOTS



```
mysql> show tables;
+---------------------------+
| Tables_in_karsafar_db     |
+---------------------------+
| accomamenitymap           |
| accombookingrooms         |
| accommodationaddresses    |
| accommodationamenities    |
| accommodationbookingitems |
| accommodationphotos       |
| accommodations            |
| airbnbs                   |
| bookingitems              |
| bookings                  |
| buses                     |
| cabs                      |
| cruises                   |
| flights                   |
| hotels                    |
| passengerseats            |
| payments                  |
| reviews                   |
| rooms                     |
| seats                     |
| trains                    |
| trips                     |
| useraddresses             |
| users                     |
| vehiclebookingitems       |
| vehiclecoaches            |
| vehicledrivers            |
| vehicles                  |
| vehiclestations           |
+---------------------------+
29 rows in set (0.00 sec)
```