MongoDB Aggregation Queries for Car Pooling Platform

**Users Collection**

**1. Average Rating per User**

**Purpose**: Calculate the average rating for each user based on their embedded ratings.

**Query**:

db.users.aggregate([

{ $match: { "ratings.0": { $exists: true } } }, // Only users with ratings

{ $addFields: { averageRating: { $avg: "$ratings.rating" } } },

{ $project: { name: 1, email: 1, role: 1, averageRating: 1 } }

])

**2. User Count by Role**

**Purpose**: Count the number of users in each role.

**Query**:

db.users.aggregate([

{ $group: { \_id: "$role", count: { $sum: 1 } } }

])

**Rides Collection**

**1. Rides with Passenger Count**

**Purpose**: Show the number of passengers for each ride.

**Query**:

db.rides.aggregate([

{ $addFields: { passengerCount: { $size: "$passengers" } } },

{ $project: { driver: 1, startLocation: 1, endLocation: 1, departureTime: 1, seatsAvailable: 1, status: 1, passengerCount: 1 } }

])

**2. Available Rides**

**Purpose**: List rides with available seats.

**Query**:

db.rides.aggregate([

{ $match: { seatsAvailable: { $gt: 0 }, status: "open" } },

{ $project: { driver: 1, startLocation: 1, endLocation: 1, departureTime: 1, seatsAvailable: 1 } }

])

**Bookings Collection**

**1. Bookings by Status**

**Purpose**: Count bookings by their status.

**Query**:

db.bookings.aggregate([

{ $group: { \_id: "$status", count: { $sum: 1 } } }

])

**2. User Bookings with Ride Details**

**Purpose**: Retrieve detailed ride information for a user's bookings.

**Query**:

db.bookings.aggregate([

{ $match: { rider: "user2" } },

{ $lookup: { from: "rides", localField: "ride", foreignField: "\_id", as: "rideDetails" } },

{ $unwind: "$rideDetails" },

{ $project: { ride: 1, status: 1, "rideDetails.startLocation": 1, "rideDetails.endLocation": 1, "rideDetails.departureTime": 1 } }

])