EDGE PROGRAM PROJECT

Identifying required Entities : identifying attributes and primary key for each entity:

- **Users**
  - `user_id` (PK)
  - `first_name`
  - `last_name`
  - `email`
  - `phone`
  - `address`
  - `driver_license_number`
- **Vehicles**
  - `vehicle_id` (PK)
  - `make`
  - `model`
  - `year`
  - `vin`
  - `registration_number`
  - `status`
  - `type_id` (FK)
- **Vehicle_Type**
  - `type_id` (PK)
  - `type_name`
- **Rentals**
  - `rental_id` (PK)
  - `user_id` (FK)
  - `vehicle_id` (FK)
  - `rental_date`
  - `return_date`
  - `total_amount`
- **Payments**
  - `payment_id` (PK)
  - `rental_id` (FK)
  - `payment_date`

- `amount`
- `payment_method`

- **Reservations**

  - `reservation_id` (PK)
  - `user_id` (FK)
  - `vehicle_id` (FK)
  - `reservation_date`
  - `status`

- **Vehicle_Maintenance**

  - `maintenance_id` (PK)
  - `vehicle_id` (FK)
  - `maintenance_date`
  - `details`
  - `cost`

- **Locations**

  - `location_id` (PK)
  - `location_name`
  - `address`
  - `phone`

- **Vehicle_Location**

  - `vehicle_location_id` (PK)
  - `vehicle_id` (FK)
  - `location_id` (FK)
  - `date_moved`

Reviews

- `review_id` (PK)
- `user_id` (FK)
- `vehicle_id` (FK)
- `review_date`
- `rating`
- `comments`

Identifying the required relationship;

- **Users** to **Rentals**: One-to-Many
  - One user can have multiple rentals.
  - Foreign Key: `user_id` in Rentals references `user_id` in Users.
- **Users** to **Reservations**: One-to-Many
  - One user can have multiple reservations.
  - Foreign Key: `user_id` in Reservations references `user_id` in Users.
- **Users** to **Reviews**: One-to-Many
  - One user can write multiple reviews.
  - Foreign Key: `user_id` in Reviews references `user_id` in Users.
- **Vehicles** to **Rentals**: One-to-Many
  - One vehicle can be rented multiple times.
  - Foreign Key: `vehicle_id` in Rentals references `vehicle_id` in Vehicles.
- **Vehicles** to **Reservations**: One-to-Many
  - One vehicle can have multiple reservations.
  - Foreign Key: `vehicle_id` in Reservations references `vehicle_id` in Vehicles.
- **Vehicles** to **Vehicle_Maintenance**: One-to-Many
  - One vehicle can have multiple maintenance records.
  - Foreign Key: `vehicle_id` in Vehicle_Maintenance references `vehicle_id` in Vehicles.
- **Vehicles** to **Vehicle_Location**: One-to-Many
  - One vehicle can have multiple location records over time.

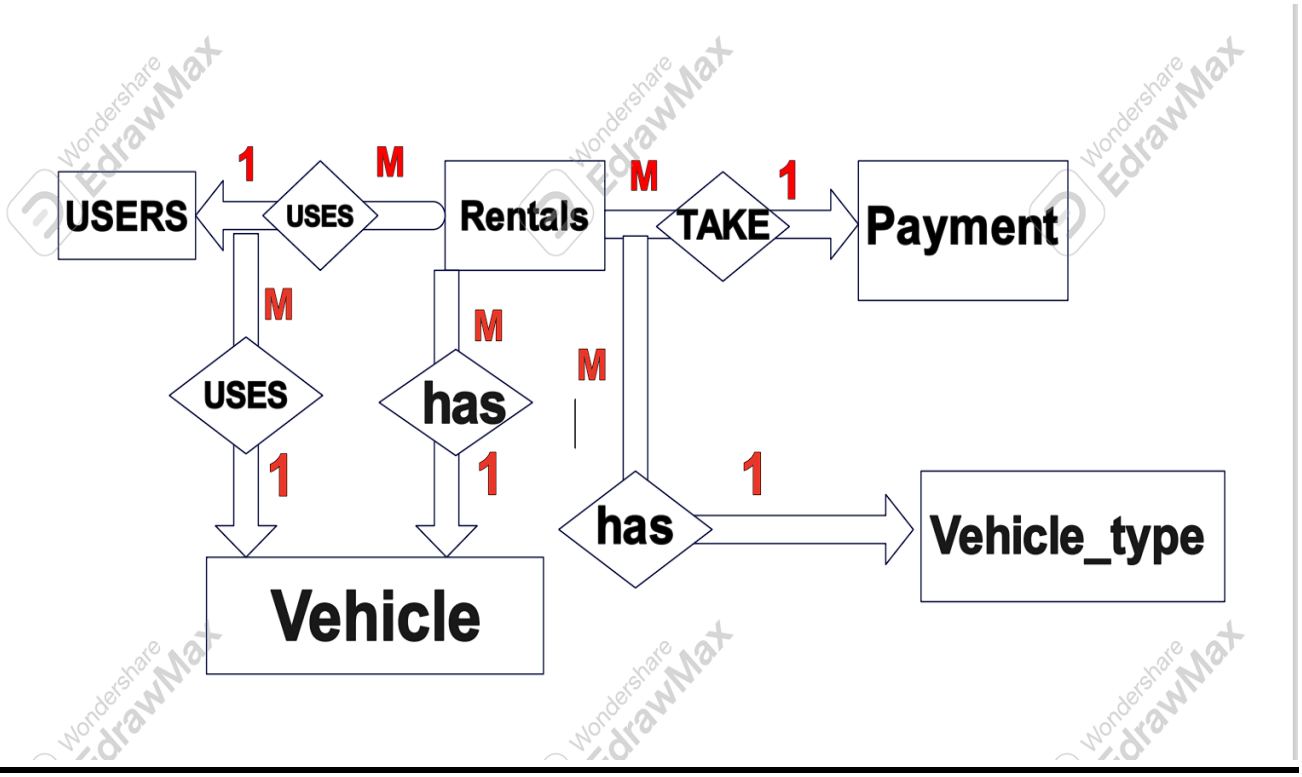
- Foreign Key: `vehicle_id` in Vehicle_Location references `vehicle_id` in Vehicles.

- **Vehicle_Type** to **Vehicles**: One-to-Many
  - One vehicle type can be assigned to multiple vehicles.
  - Foreign Key: `type_id` in Vehicles references `type_id` in Vehicle_Type.

- **Locations** to **Vehicle_Location**: One-to-Many
  - One location can have multiple vehicle location records.
  - Foreign Key: `location_id` in Vehicle_Location references `location_id` in Locations.

- **Rentals** to **Payments**: One-to-Many
  - One rental can have multiple payments.
  - Foreign Key: `rental_id` in Payments references `rental_id` in Rentals.

ERD Diagram



Reducing diagram to schema:

Users (user_id, first_name, last_name, email, phone, address, driver_license_number)

Vehicles (vehicle_id, make, model, year, vin, registration_number, status)

Vehicle_Type (type_id, type_name)

Rentals (rental_id, user_id (FK), vehicle_id (FK), rental_date, return_date, total_amount)

Payments (payment_id, rental_id (FK), payment_date, amount, payment_method)

Reservations (reservation_id, user_id (FK), vehicle_id (FK), reservation_date, status)

Vehicle_Maintenance (maintenance_id, vehicle_id (FK), maintenance_date, details, cost)

Locations (location_id, location_name, address, phone)

Vehicle_Location (vehicle_location_id, vehicle_id (FK), location_id (FK), date_moved)

Reviews (review_id, user_id (FK), vehicle_id (FK), review_date, rating, comments)

- `review_id` (PK)
- `user_id` (FK)
- `vehicle_id` (FK)
- `review_date`
- `rating`
- `comments`