**EDGE PROGRAM PROJECT -Vehicle Rental System**

**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.

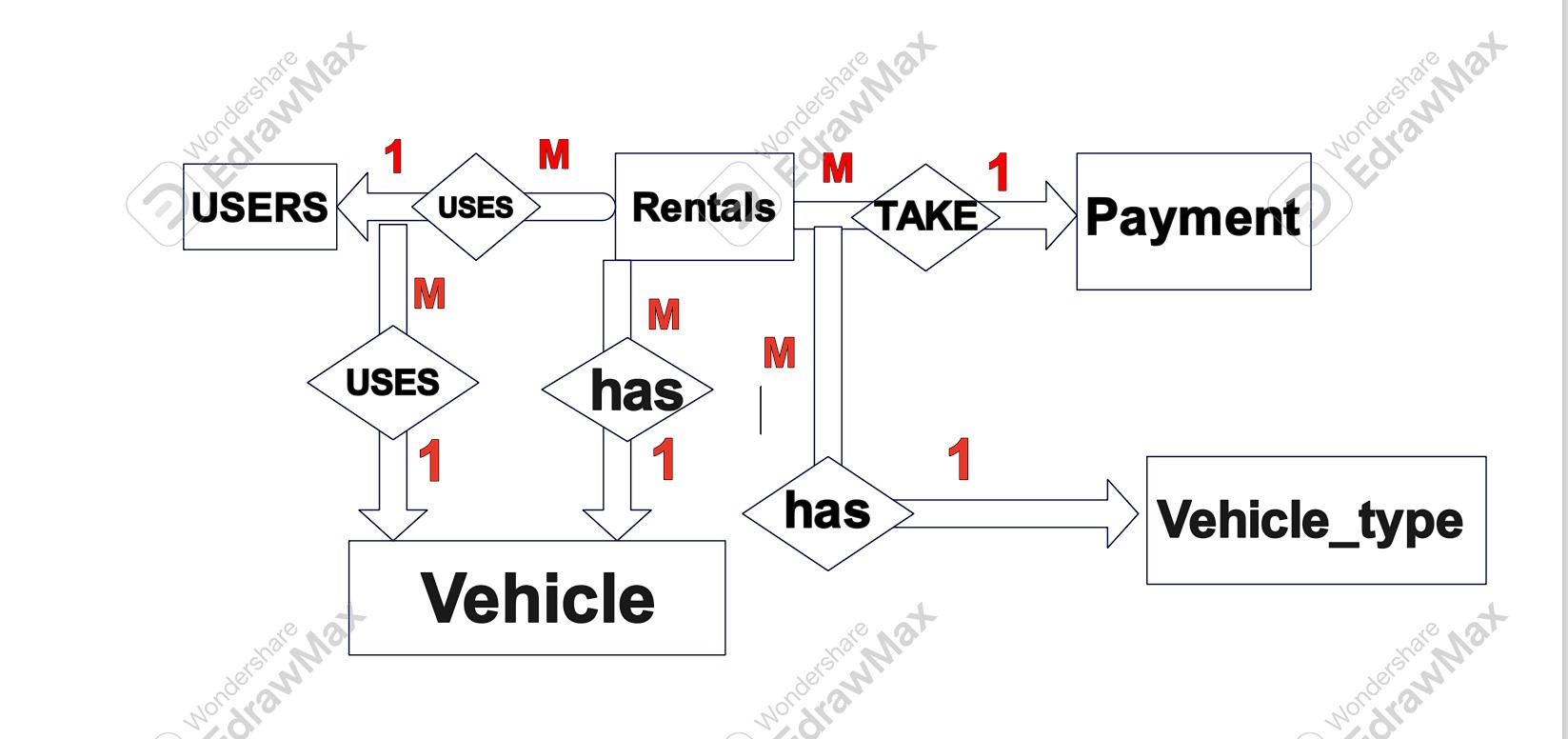
# • Vehicle\_Type to Vehicles: One-to-Many

* 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