Project Title

SmartCar – Car Rental Management System

Project Scenario

SmartCar is a database system designed for car rental companies to manage their vehicles, customers, reservations, maintenance operations, and payment transactions efficiently.

The system allows customers to make reservations, employees to manage vehicle maintenance, and administrators to track rental history, insurance details, and payments.

Project Components

1. Entities

1. Customer
2. Vehicle
3. Reservation
4. Rental
5. Payment
6. Insurance
7. Employee
8. Branch
9. Maintenance
10. Vehicle Model
11. Additional Services

---

2. Relationships

- One **Customer** can make multiple **Reservations** → *One-to-Many*
- Each **Reservation** is made for one **Vehicle** → *Many-to-One*
- One **Vehicle** can have multiple **Maintenance** records → *One-to-Many*
- One **Vehicle Model** can correspond to multiple **Vehicles** → *One-to-Many*
- One **Branch** can have many **Employees** and **Vehicles** → *One-to-Many*
- A **Rental** connects one **Customer** and one **Vehicle**, but over time both can appear in many rentals → *Many-to-Many*

- One **Reservation** can include multiple **Additional Services**, and each Service can belong to multiple **Reservations** → *Many-to-Many*
- Each **Rental** has exactly one **Payment**, and a **Payment** belongs to one **Rental** → *One-to-One*
- Each **Vehicle** must have one **Insurance** record → *One-to-One*

---

3. Attributes

Customer: customerID, firstName, lastName, licenseNo, phone, email, address

Vehicle: vehicleID, plateNo, modelID, branchID, color, kilometer, fuelType, status

Reservation: reservationID, customerID, vehicleID, startDate, endDate, totalPrice, reservationDate, status

Rental: rentalID, reservationID, paymentID, totalDays, totalCost

Payment: paymentID, rentalID, amount, paymentDate, paymentMethod

Insurance: insuranceID, vehicleID, policyType, provider, startDate, expiryDate, cost

Employee: employeeID, branchID, firstName, lastName, role, salary

Branch: branchID, branchName, city, address, phone

Maintenance: maintenanceID, vehicleID, date, description, cost

Vehicle Model: modelID, brand, modelName, year, transmissionType, pricePerDay

Additional Services: serviceID, serviceName, description, cost

---

4. Existence Dependency

The **Reservation** entity shows existence dependency on the **Customer** entity. A reservation can only be created if a customer exists. When a customer is deleted, all their reservations must also be deleted.

Similarly, the **Rental** entity is dependent on **Reservation**. A rental can only occur through a valid reservation.

Additionally, **Maintenance** records are existence-dependent on **Vehicle**. When a vehicle is removed from the system, all its maintenance history must also be deleted.

---

5. ISA Relationship

An ISA relationship is applied for the **Employee** entity. The 'role' attribute in Employee determines the type of employee:

- **Manager:** Responsible for branch management and has additional bonus compensation

- **Technician:** Handles vehicle maintenance and repairs with specific specializations
- **Salesperson:** Manages customer relations and rental operations with commission-based earnings

An employee can have one of these roles, and the role determines their responsibilities within the system.

---

6. Exclusion

A **Vehicle** at a specific time can either be in the **Maintenance** process or in the **Rental** process, but cannot be in both simultaneously. This exclusion constraint is controlled by the vehicle's status attribute (available, rented, maintenance).

When a vehicle is marked as "maintenance", it cannot be included in any active rental, and vice versa.

---

7. Weak Entity

Maintenance is a weak entity because it has no meaning on its own and is completely dependent on the **Vehicle** entity. When a vehicle is deleted, all its maintenance records are also deleted.

The primary key of the Maintenance entity consists of the Vehicle's vehicleID together with maintenanceID (partial key). A maintenance record cannot exist without being associated with a specific vehicle.

---

8. Disjoint

A **disjoint** constraint is applied in the ISA relationship for the **Employee** entity. An employee cannot have multiple roles simultaneously. Each employee can only be either a Manager, a Technician, or a Salesperson at any given time.

This ensures clear responsibility assignment and prevents role conflicts within the organization.

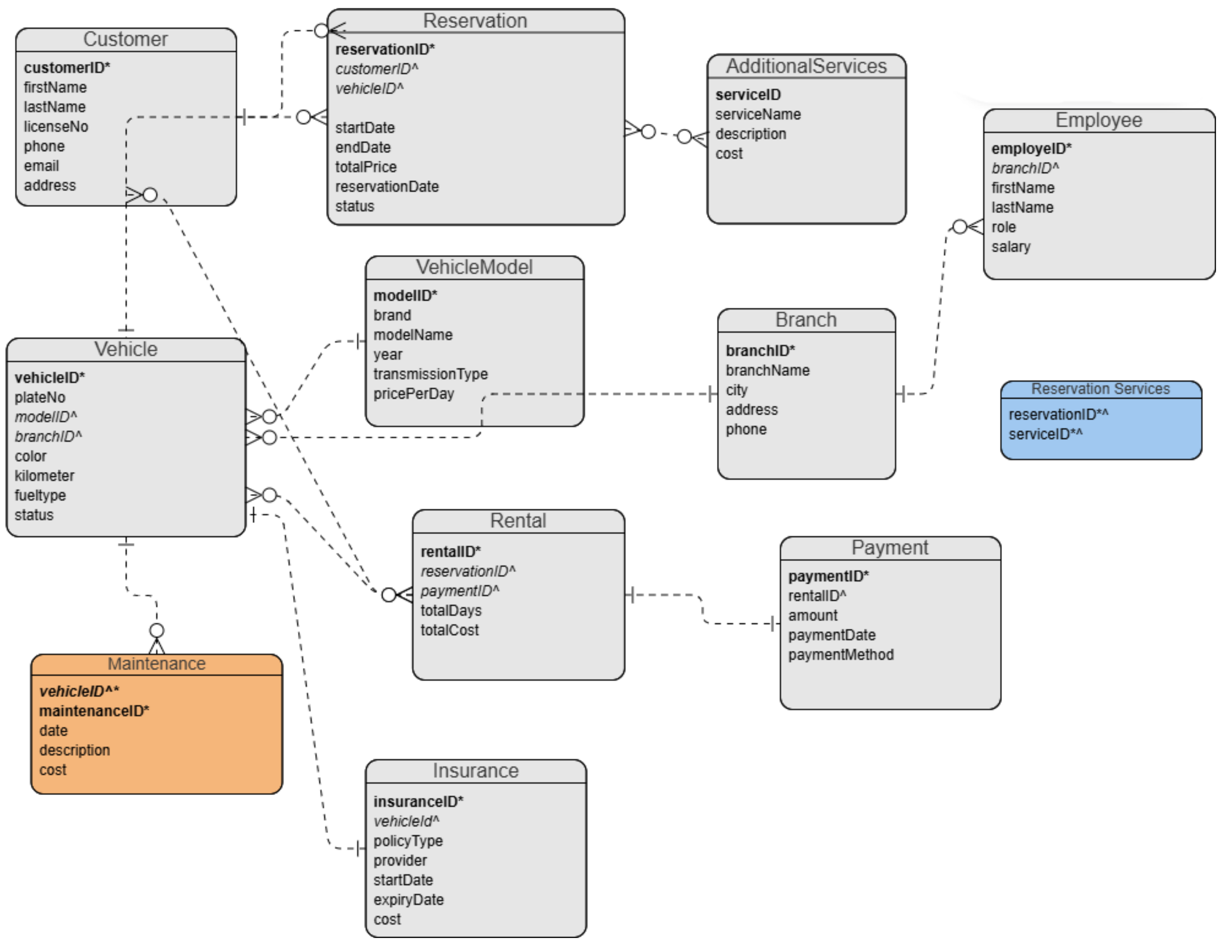
---

9. Overlapping

An overlapping situation exists for the **Additional Services** entity. A reservation can include multiple services simultaneously such as 'GPS Navigation', 'Child Seat', and 'Additional Driver'.

A service can be in multiple reservations at the same time, and a reservation can have multiple services. This Many-to-Many relationship allows flexible service combinations for customers.

10. Relational Schemas



Customer(customerID*, firstName, lastName, licenseNo, phone, email, address)

VehicleModel(modelID*, brand, modelName, year, transmissionType, pricePerDay)

Branch(branchID*, branchName, city, address, phone)

Vehicle(vehicleID*, plateNo, modelID↑, branchID↑, color, kilometer, fuelType, status)

Employee(employeeID*, branchID↑, firstName, lastName, role, salary)

Reservation(reservationID*, customerID↑, vehicleID↑, startDate, endDate, totalPrice, reservationDate, status)

Payment(paymentID*, rentalID↑, amount, paymentDate, paymentMethod)

Rental(rentalID*, reservationID↑, paymentID↑, totalDays, totalCost)

Insurance(insuranceID*, vehicleID↑, policyType, provider, startDate, expiryDate, cost)

Maintenance(vehicleID↑*, maintenanceID*, date, description, cost)

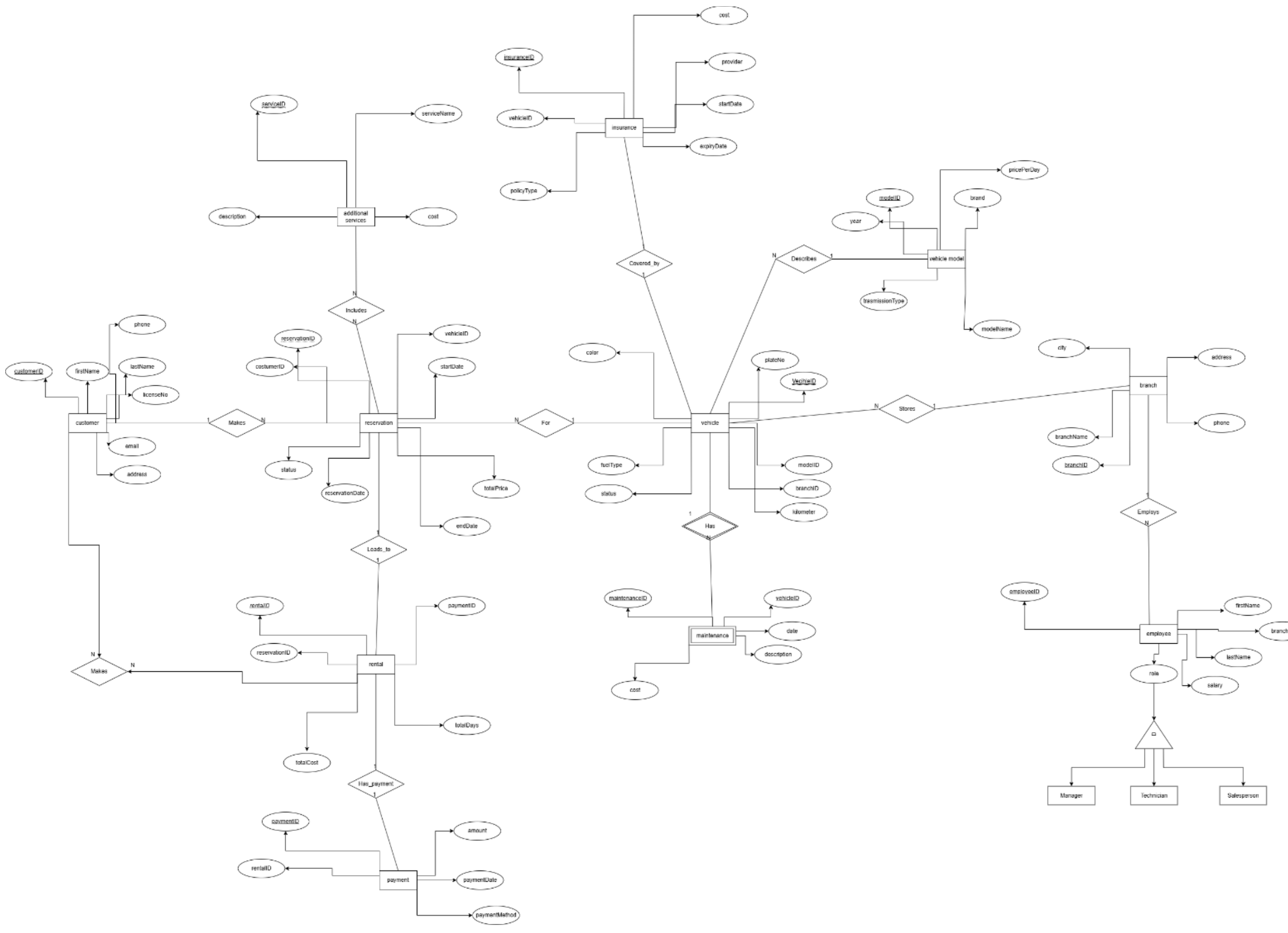
AdditionalServices(serviceID*, serviceName, description, cost)

Reservation_Services(reservationID↑*, serviceID↑*)

Notation:

- (*) Primary key
- (↑) Foreign key
- Maintenance has a composite primary key (vehicleID, maintenanceID) as it is a weak entity
- Reservation_Services is a junction table for the Many-to-Many relationship between Reservation and Additional Services

E-R (Chen) Diagram:



cost
insuranceID
provider
vehicleID
insurance
startDate
expiryDate
policyType
serviceID
serviceName
description
additional services
cost
pricePerDay
modelID
brand
year
Covered_by
1
N
Describes
1
vehicle model
trasmissionType
modelName
N
Includes
N
phone
reservationID
vehicleID
customerID
firstName
lastName
costumerID
startDate
licenseNo
color
plateNo
VecihleID
city
address
branch
customer
1
Makes
N
reservation
N
For
1
vehicle
N
Stores
1
phone
email
address
status
reservationDate
totalPrice
endDate
fuelType
status
modelID
branchID
kilometer
branchName
branchID
1
Has
N
1
Employs
N
1
Loads_to
1
rentalID
paymentID
maintenanceID
vehicleID
date
maintenance
description
cost
employeeID
firstName
employee
branchID
lastName
role
salary
reservationID
N
Makes
N
rental
totalDays
totalCost
1
Has_payment
1
paymentID
amount
rentalID
payment
paymentDate
paymentMethod
d
Manager
Technician
Salesperson