# 

CAR RENTAL DATABASE SYSTEM

Database Project

A logo of a college

AI-generated content may be incorrect.

**Deliverable 2**

**Submitted by**

|  |  |
| --- | --- |
| **Member Name** | **Registration Number** |
| Farah Tanveer | (BCSF24M016) |
| Ayesha Bint E Younas | (BCSF24M032) |
| Gulshan Mushtaq | (BCSF24M046) |
| Hafsa Kokab | (BCSF24M051) |

**Submitted To**

Dr. Irfana Bibi

# 

# ERD Normalization: Step-by-step 1NF / 2NF / 3NF / BCNF Justification

This document performs a table-by-table normalization check for the above ERD:

1. 1NF — atomic values, no repeating groups, unique rows
2. 2NF (only relevant when PK is composite) no partial dependencies
3. 3NF — no transitive dependencies; every non-key attribute depends only on the PK
4. BCNF — every determinant is a candidate key.

**Notation:**

**BCNF**

* Only determinants are Customer\_id (candidate key). No other FD shown.

This keeps CUSTOMER in BCNF and makes data 1NF/3NF-friendly and auditable.

## TABLE: VEHICLE

**Schema:**

VEHICLE(ID PK, Company\_id FK, PlateNumber, Model, Year, CurrentMilage, Color,

LastServiceDate, DailyRate, SeatingCapacity ,Status, Description)

**1NF**

* Fields are atomic. Hence, in 1NF.

**2NF**

* PK is single attribute ID → no partial dependencies.

**3NF**

Non-key attributes (Company\_id, PlateNumber, Model, Year, CurrentMilage, Color, LastServiceDate, DailyRate, SeatingCapacity ,Status, Description) all describe the vehicle and depend on ID.

* No transitive dependencies.

**BCNF**

* If there existed a dependency such as PlateNumber → id (plate uniquely identifies vehicle), then PlateNumber could be a candidate key. Current diagram doesn't state that, but if PlateNumber is unique it would be a candidate key; still BCNF holds as long as every determinant is a candidate key.

**Solution :**

Separate table for PLATENUMBER.

**Revised:**

VEHICLE(id PK, Company\_id FK, CurrentMilage, Color, LastServiceDate,DailyRate,SeatingCapacity,Status, description)

PLATENUMBER([PlateNumber , Vehicle\_id] PK(COMPOSITE PK) , year, model)

## TABLE: COMPANY

**Schema:**

COMPANY(ID PK, Name)

**1NF**

* All attributes are atomic.

**2NF**

* Single-attribute PK → no partial dependencies.

**3NF**

Non-key attributes depends on key attributes. No transitive dependencies visible.

**BCNF**

* No non-key determinants apparent.

**Status:** COMPANY is in BCNF.

## TABLE: ROLE

**Schema:**

ROLE(ID PK, Name, Description)

**1NF**

* Atomic attributes.

**2NF**

* PK is single attribute no partial dependency.

**3NF**

* Each attribute depends only on ID no transitive dependency exists.

**BCNF**

* No other determinants shown.

## TABLE: STAFF

**Schema:**

(id PK, Role\_id, FirstName, LastName, HireDate, Salary, ContactNo,

Email,LisenceNo,ExperienceYears)

**1NF**

* Atomic attributes. 1NF satisfied.

**2NF**

* Single-attribute PK so no partial dependencies.

**3NF**

* Attributes describe employee and depend on ID. No transitive dependency shown.

**BCNF**

* No non-key determinants shown.

## TABLE: MAINTENANCE

**Schema:**

MAINTENANCE(id PK, Vehicle\_id FK, Staff\_id FK, MaintanaceDate, Description, Cost)

**1NF**

* Atomic attributes.1NF satisfied.

**2NF**

* PK is single attribute 2NF satisfied.

**3NF**

* All non-key attributes (Vehicle\_id, Staff\_id, MaintanaceDate, Description, Cost) describe the maintenance event. They all depend on id. No transitive dependency appears.

**BCNF**

If the real-world functional dependency (Vehicle\_id,MaintenanceDate)→Employee\_id exists, then (Vehicle\_id, MaintenanceDate) acts as a determinant. However, in the current design, Maintenance\_id is the only candidate key, so this natural dependency is hidden and not enforced. This means the schema is in 3NF but may violate BCNF.

**Problem:**

Using a surrogate PK alone does not prevent storing two maintenance records for the same vehicle at the same timestamp.

**Solution:**

Keep the surrogate PK but add a UNIQUE constraint on (Vehicle\_id, MaintenanceDate).

MAINTENANCE(ID PK,Vehicle\_id , MaintenanceDate , STAFF\_id FK, Description, Cost) or add a unique constraint on (Vehicle\_id, MaintenanceDate) while keeping surrogate PK.

**Normalization status:** current design is in 3NF. Use unique constraints for stronger real-world guarantees.

## TABLE: RESERVATION

**Schema:**

RESERVATION(id PK, Customer\_id FK, Vehicle\_id FK, Staff\_id FK, Driver\_id FK,DriverRequired,

RateAtBooking, PriceBasis, Rent\_SatrtDate, Rent\_EndDate, TotalAmount, Status ,

PaymentStatus)

**1NF**

* All attributes atomic. TotalAmount is a single numeric value.

**2NF**

* PK is single attribute, no partial dependencies.

**3NF**

* **Potential issue of derived attribute**

TotalAmount is **derived** from:

Daily Rate × number of rental days

So it is **functionally dependent on other attributes**, meaning:

If it is not stored as a permanent transaction snapshot,then storing it violates strict 3NF because it introduces redundancy.

**So, we remove TotalAmount attribute.**

**Conclusion for 3NF**

3NF is technically violated **only if** TotalAmount is always derivable and not meant as transactional history. **BCNF**

BCNF SATISFIED.

## TABLE: PAYMENT

**Schema:**

PAYMENT(id PK, Reservation\_id FK, PaymentDate, Amount,

PaymentMethod)

**1NF**

* Atomic values.

**2NF**

* Single PK 2NF satisfied.

**3NF**

* No transitive dependency

**BCNF**

* BCNF holds. **Fix**

## TABLE: DAMAGE\_REPORT

**Schema:**

DAMAGE\_REPORT(id PK, Resrevation\_id FK, Vehicle\_id FK, SeverityLevel, Description,

EstimatedCost, ReportDate)

**1NF**

* Atomic attributes.

**2NF**

* Single PK , no partial dependency.

**3NF**

id determines all non-key attributes.

* Determinant (id) is primary key

**But** Reservation\_id → Vehicle\_id

* Both are **non-key attributes**
* This means:

**id → Reservation\_id → Vehicle\_id** This is a **transitive dependency**.

**3NF Violation**

Because:

* Reservation\_id is a non-key attribute
* It determines another non-key attribute (Vehicle\_id)

**Solution:**

Vehicle\_id should not be stored in DAMAGE\_REPORT because it can be obtained from Reservation table.

**Revised Schema:**

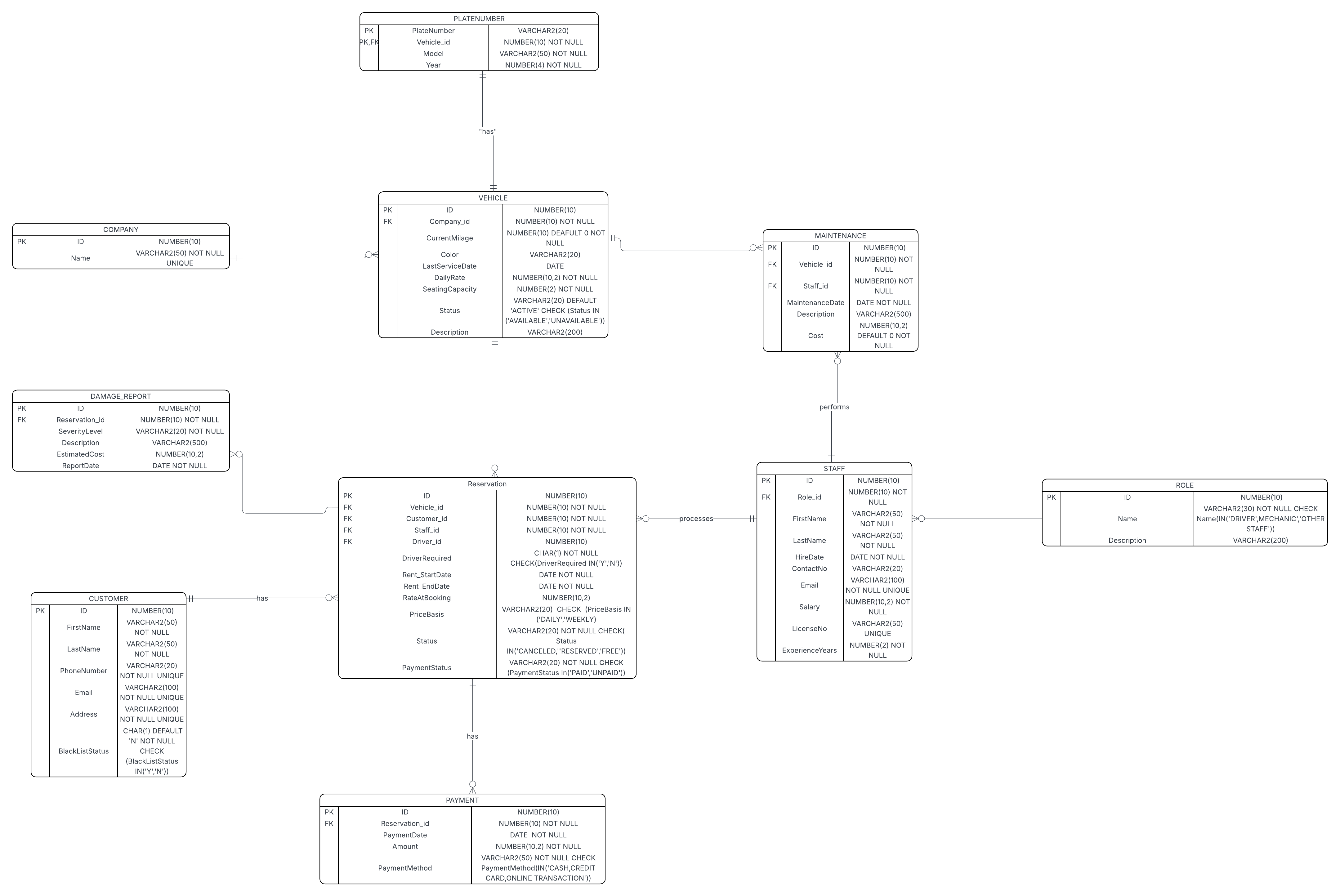
DAMAGE\_REPORT(id PK, Resrevation\_id FK, SeverityLevel, Description, EstimatedCost, ReportDate)

**BCNF :**

The table is already in BCNF after fixing the 3NF violation.

There is no BCNF violation remaining and no further decomposition is required.

# 4.NORMALIZED ERD DIAGRAM



# 5.DENORMALIZATION

During normalization analysis, we identified **functional dependencies crossing tables**, such as:

Reservation\_id → Customer\_id, Vehicle\_id

(Meaning Customer and Vehicle can be inferred from reservation)

Vehicle\_id → Company\_id

(Meaning company can be derived from vehicle)

These dependencies showed that some stored attributes were **redundant copies** of data that could be derived elsewhere, creating risk of:

* inconsistency
* unnecessary duplication
* update anomalies

Therefore, tables were normalized (reducing redundancy).

Later, we evaluated whether denormalization improves:

* query speed
* application logic
* reporting needs

## Denormalization Decisions

Although the schema is logically correct in BCNF, operational considerations may justify re-introducing some redundant attributes.

## Case Example 1: Reporting Speed

Queries like:

Show all damages with vehicle details become expensive if every lookup requires:

Damage\_Report → Reservation → Vehicle

So we may **denormalize DAMAGE\_REPORT** by storing Vehicle\_id again to optimize reporting.

* Improves analyst queries
* Reduces join cost

## Case Example 2: Rental Flow Logic

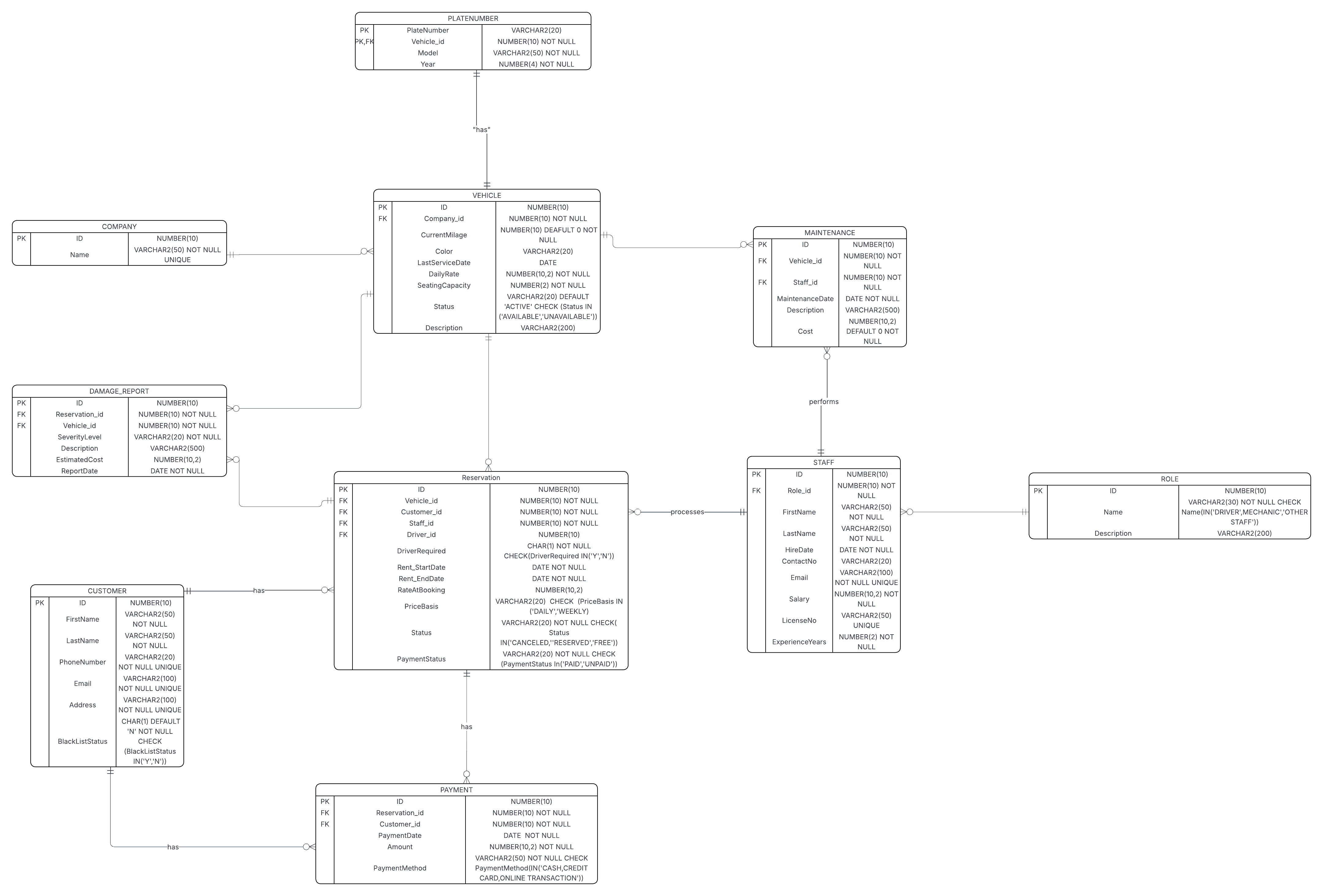
Even though:

Reservation\_id → Customer\_id

As the business allows **walk-in rentals without reservations**, Customer\_id must remain *directly stored* in RESERVATION.

So reintroducing Customer\_id IN PAYMENT TABLE could be justified.

# 6.FINAL ERD



# 7.DESCRIPTION OF THE RELATIONS

## 1. CUSTOMER MODULE

**TABLE: CUSTOMER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Size | | Constraints |
| ID | NUMBER | 10 | | PRIMARY KEY |
| FirstName | VARCHAR2 | 50 | | NOT NULL |
| LastName | VARCHAR2 | 50 | | NOT NULL |
| BlacklistStatus | CHAR | 1 | | NOT NULL, DEFAULT ‘N’,  CHECK (‘Y’,’N’) |
| Email | VARCHAR2 | 100 | NOT NULL, UNIQUE | |
| PhoneNumber | VARCHAR2 | 20 | NOTNULL, UNIQUE | |
| Address | VARCHAR2 | 100 | NOT NULL, UNIQUE | |

## 2. VEHICLE MODULE

**TABLE: COMPANY**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Size | Constraints |
| ID | NUMBER | 10 | PRIMARY KEY |
| Name | VARCHAR2 | 50 | UNIQUE, NOT NULL |

**TABLE: VEHICLE**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name Data Type | | Size Constraints | |
| ID | NUMBER | 10 PRIMARY KEY | |
| Company\_id | NUMBER | 10 FOREIGN KEY ,NOT NULL | |
| DailyRate | NUMBER | 10,2 NOT NULL | |
| SeatingCapacity | NUMBER | 2 | NOT NULL |
| Status | VARCHAR2 | 20 | DEFAULT ‘ACTIVE’, NOT NULL  CHECK(‘AVAILABLE’, ’UNAVAILABLE’) |
| CurrentMilage | NUMBER | 10 | DEFAULT 0, NOT NULL |
| Color | VARCHAR2 | 20 |  |
| Description | VARCHAR2 | 200 |  |
| LastServiceDate | DATE | — |  |

**TABLE: PLATENUMBER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Size |  | Constraints |

|  |  |  |  |
| --- | --- | --- | --- |
| PlateNumber | VARCHAR2 | 20 | PRIMARY KEY |
| Vehicle\_id | NUMBER | 10 | PRIMARY KEY, FOREIGN KEY,  NOT NULL |
| Model | VARCHAR2 | 50 | NOT NULL |
| Year | NUMBER | 4 | NOT NULL |

## 3. STAFF MODULE

**TABLE: STAFF**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Size | Constraints |
| ID | NUMBER | 10 | PRIMARY KEY |
| Role\_id | NUMBER | 10 | FOREIGN KEY, NOT NULL |
| FirstName | VARCHAR2 | 50 | NOT NULL |
| LastName | VARCHAR2 | 50 | NOT NULL |
| HireDate | DATE | — | NOT NULL |
| Salary | NUMBER | 10,2 | NOT NULL |
| ContactNo | VARCHAR2 | 20 |  |
| Email | VARCHAR2 | 100 | UNIQUE, NOT NULL |
| LisenceNo | VARCHAR2 | 50 | UNIQUE |
| ExperienceYears | NUMBER | 2 | NOT NULL |

**TABLE: ROLE**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name Data Type | | Size | Constraints |
| ID | NUMBER | 10 | PRIMARY KEY |
| Name | VARCHAR2 | 30 | NOT NULL , CHECK(‘DRIVER’, ’MECHANIC’, ’OTHER STAFF’) |
| Description | VARCHAR2 | 200 |  |

## 4. MAINTENANCE MODULE

**TABLE: MAINTENANCE**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Size Constraints | |
| ID | NUMBER | 10 | PRIMARY KEY |
| Vehicle\_id | NUMBER | 10 | FOREIGN KEY , NOT NULL |
| Staff\_id | NUMBER | 10 | FOREIGN KEY , NOT NULL |
| MaintenanceDate | DATE | — | NOT NULL |
| Description | VARCHAR2 | 500 |  |
| Cost | NUMBER | 10,2 | DEFAULT 0, NOT NULL |

## 5. RESERVATION MODULE

TABLE: RESERVATION

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | | Size | | Constraints |
| ID | NUMBER | | 10 | | PRIMARY KEY |
| Customer\_id | NUMBER | | 10 | | FOREIGN KEY ,NOT NULL |
| Vehicle\_id | NUMBER | | 10 | | FOREIGN KEY , NOT NULL |
| Staff\_id | NUMBER | 10 | | FOREIGN KEY, NOT  NULL | |
| Driver\_id | NUMBER | 10 | | FOREIGN KEY | |
| DriverRequired | CHAR | 1 | | NOT NULL,  CHECK(‘Y’, ‘N’) | |
| Rent\_StartDate | DATE | — | | NOT NULL | |
| Rent\_EndDate | DATE | — | | NOT NULL | |
| RateAtBooking | NUMBER | 10,2 | |  | |
| PriceBasis | VARCHAR2 | 20 | | CHECK (‘DAILY’,’WEEKLY’) | |
| Status | VARCHAR2 | 20 | | NOT NULL ,  CHECK(‘CANCELLED’,  ‘RESERVED’, ‘FREE’) | |
| PaymentStatus | VARCHAR2 | 20 | | NOT NULL,  CHECK(‘PAID’,  ‘UNPAID’) | |

## 6. PAYMENT MODULE

**TABLE: PAYMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Size | Constraints |
| ID | NUMBER | 10 | PRIMARY KEY |
| Customer\_id | NUMBER | 10 | FOREIGN KEY, NOT NULL |
| Reservation\_id | NUMBER | 10 | FOREIGN KEY, NOT NULL |
| PaymentDate | DATE | — | NOT NULL |
| Amount | NUMBER | 10,2 | NOT NULL |
| PaymentMethod | VARCHAR2 | 50 | NOT NULL , CHECK(‘CASH’,  CREDIT CARD’,’ONLINE  TRANSACTION') |

## 7. DAMAGE REPORT MODULE

TABLE: DAMAGE REPORT

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Size | Constraints |
| ID | NUMBER | 10 | PRIMARY KEY |
| Reservation\_id | NUMBER | 10 | FOREIGN KEY, NOT NULL |
| Vehicle\_id | NUMBER | 10 | FOREIGN KEY, NOT NULL |
| SeverityLevel | VARCHAR2 | 20 | NOT NULL |
| Description | VARCHAR2 | 500 |  |
| EstimatedCost | NUMBER | 10,2 |  |
| ReportDate | DATE | — | NOT NULL |

# DDL

-----------------COMPANY—------------------

**CREATE TABLE Company (**

**ID NUMBER(10) PRIMARY KEY,**

**Name VARCHAR2(150) NOT NULL UNIQUE**

**);**

**—----------------VEHICLE—-------------------- CREATE TABLE Vehicle (**

**ID NUMBER(10) PRIMARY KEY,**

**Company\_id NUMBER(10) NOT NULL,**

**CurrentMileage NUMBER DEFAULT 0 NOT NULL,**

**Color VARCHAR2(20),**

**LastServiceDate DATE,**

**DailyRate NUMBER(8,2) NOT NULL,**

**SeatingCapacity NUMBER(2) NOT NULL,**

**Status VARCHAR2(20) DEFAULT 'Active' NOT NULL,**

**Description VARCHAR2(200),**

**CONSTRAINT fk\_vehicle\_company FOREIGN KEY (Company\_id)**

**REFERENCES Company(ID)**

**);**

**—-------------PLATENUMBER—-------------------**

**CREATE TABLE PlateNumber (**

**PlateNumber VARCHAR2(20) PRIMARY KEY,**

**Vehicle\_id NUMBER(10) NOT NULL UNIQUE,**

**Model VARCHAR2(150) NOT NULL,**

**Year NUMBER(4) NOT NULL,**

**CONSTRAINT fk\_plate\_vehicle FOREIGN KEY (Vehicle\_id)**

**REFERENCES Vehicle(ID)**

**);**

**—----------------ROLE—--------------------------**

**CREATE TABLE Role (**

**ID NUMBER(10) PRIMARY KEY,**

**Name VARCHAR2(20) NOT NULL CHECK (Name IN**

**('DRIVER','MECHANIC','OTHER STAFF')),**

**Description VARCHAR2(200)**

**);**

**—----------------STAFF—-----------------**

**CREATE TABLE Staff (**

**ID NUMBER(10) PRIMARY KEY,**

**Role\_id NUMBER(10) NOT NULL,**

**FirstName VARCHAR2(50) NOT NULL,**

**LastName VARCHAR2(50) NOT NULL,**

**HireDate DATE NOT NULL,**

**ContactNo VARCHAR2(20),**

**Email VARCHAR2(100) NOT NULL UNIQUE,**

**Salary NUMBER(10,2) NOT NULL,**

**LicenseNo VARCHAR2(50) UNIQUE,**

**ExperienceYears NUMBER(2) NOT NULL,**

**CONSTRAINT fk\_staff\_role FOREIGN KEY (Role\_id)**

**REFERENCES Role(ID)**

**);**

**—--------------CUSTOMER—----------------**

**CREATE TABLE Customer (**

**ID NUMBER(10) PRIMARY KEY,**

**FirstName VARCHAR2(50) NOT NULL,**

**LastName VARCHAR2(50) NOT NULL,**

**PhoneNumber VARCHAR2(20) NOT NULL,**

**Email VARCHAR2(100) NOT NULL UNIQUE,**

**Address VARCHAR2(200),**

**BlackListStatus CHAR(1) DEFAULT 'N' CHECK (BlackListStatus IN ('Y','N'))**

**);**

**—-------------RESERVATION—--------------**

**CREATE TABLE Reservation (**

**ID NUMBER(10) PRIMARY KEY,**

**Vehicle\_id NUMBER(10) NOT NULL,**

**Customer\_id NUMBER(10) NOT NULL,**

**Staff\_id NUMBER(10) NOT NULL,**

**DriverRequired CHAR(1) NOT NULL CHECK (DriverRequired IN ('Y','N')),**

**Driver\_id NUMBER(10),**

**Rent\_StartDate DATE NOT NULL,**

**Rent\_EndDate DATE NOT NULL,**

**RateNoBooking NUMBER(10,2),**

**PriceBasis VARCHAR2(30),**

**Status VARCHAR2(20) NOT NULL CHECK (Status IN ('Canceled','Reserved','Free')),**

**PaymentStatus VARCHAR2(20) NOT NULL CHECK (PaymentStatus IN ('PAID','PARTIALLY PAID','UNPAID')),**

**CONSTRAINT fk\_res\_vehicle FOREIGN KEY (Vehicle\_id) REFERENCES Vehicle(ID),**

**CONSTRAINT fk\_res\_customer FOREIGN KEY (Customer\_id) REFERENCES Customer(ID),**

**CONSTRAINT fk\_res\_staff FOREIGN KEY (Staff\_id) REFERENCES Staff(ID)**

**);**

**—-----------------PAYMENT—----------------**

**CREATE TABLE Payment (**

**ID NUMBER(10) PRIMARY KEY,**

**Reservation\_id NUMBER(10) NOT NULL,**

**Customer\_id NUMBER(10) NOT NULL,**

**PaymentDate DATE NOT NULL,**

**Amount NUMBER(10,2) NOT NULL,**

**PaymentMethod VARCHAR2(30) NOT NULL CHECK (PaymentMethod IN**

**('CASH','CREDITCARD','CARD','ONLINE TRANSACTION')),**

**CONSTRAINT fk\_payment\_res FOREIGN KEY (Reservation\_id) REFERENCES Reservation(ID),**

**CONSTRAINT fk\_payment\_customer FOREIGN KEY (Customer\_id) REFERENCES Customer(ID)**

**);**

**—-----------------MAINTENANCE—--------------**

**CREATE TABLE Maintenance (**

**ID NUMBER(10) PRIMARY KEY,**

**Vehicle\_id NUMBER(10) NOT NULL UNIQUE,**

**Staff\_id NUMBER(10) NOT NULL,**

**MaintenanceDate DATE NOT NULL,**

**Description VARCHAR2(500),**

**Cost NUMBER(8,2) DEFAULT 0 NOT NULL,**

**CONSTRAINT fk\_maint\_vehicle FOREIGN KEY (Vehicle\_id) REFERENCES**

**Vehicle(ID),**

**CONSTRAINT fk\_maint\_staff FOREIGN KEY (Staff\_id) REFERENCES Staff(ID)**

**);**

**—-----------------DAMAGE REPORT—---------------**

**CREATE TABLE Damage\_Report (**

**ID NUMBER(10) PRIMARY KEY,**

**Reservation\_id NUMBER(10) NOT NULL,**

**Vehicle\_id NUMBER(10) NOT NULL,**

**SeverityLevel VARCHAR2(20) NOT NULL,**

**Description VARCHAR2(500),**

**EstimatedCost NUMBER(10,2),**

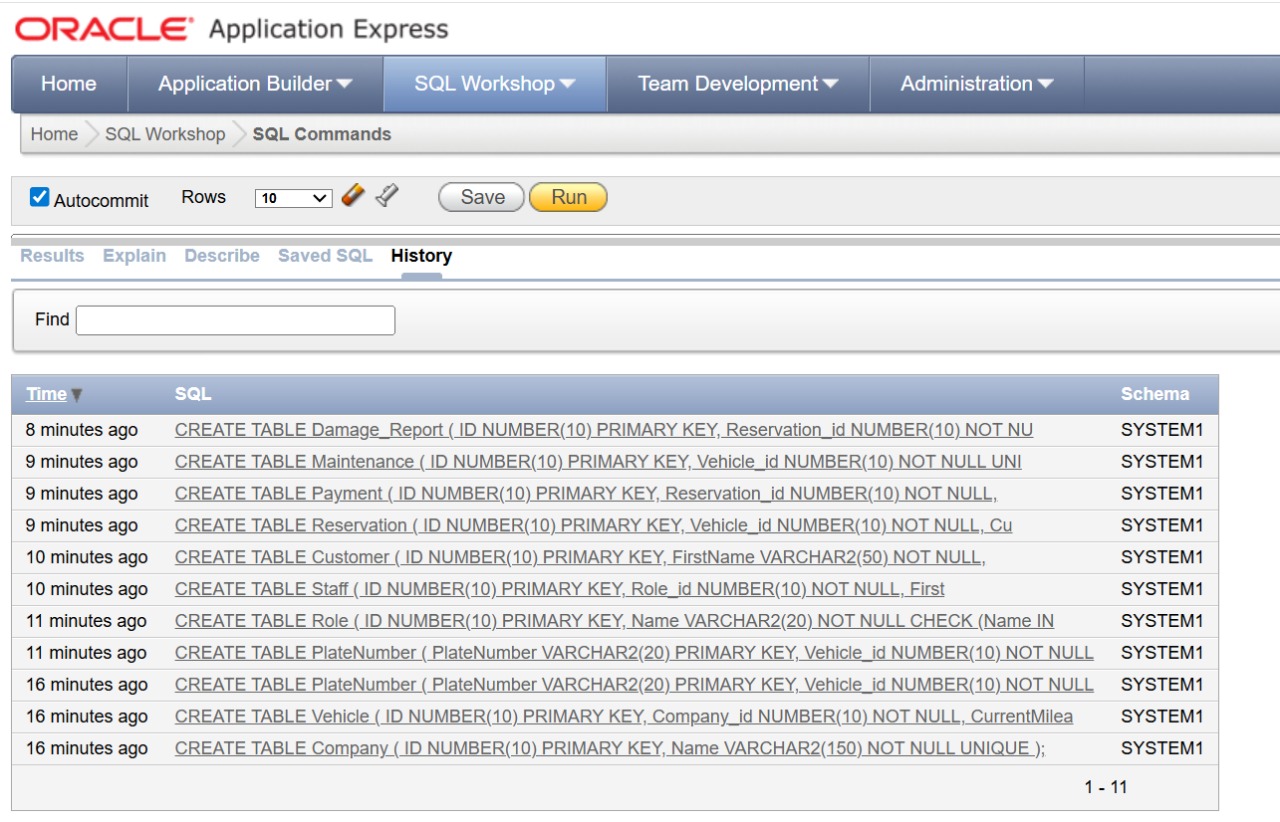
**ReportDate DATE NOT NULL,**

**CONSTRAINT fk\_damage\_reservation FOREIGN KEY (Reservation\_id)**

**REFERENCES Reservation(ID),**

**CONSTRAINT fk\_damage\_vehicle FOREIGN KEY (Vehicle\_id) REFERENCES Vehicle(ID)**

**);**



# →DML

—------------COMPANY—-----------------

**INSERT ALL**

**INTO Company (ID, Name) VALUES (1, 'ZoomRentals')**

**INTO Company (ID, Name) VALUES (2, 'AutoFleet Inc')**

**INTO Company (ID, Name) VALUES (3, 'CityDrive Co')**

**INTO Company (ID, Name) VALUES (4, 'MetroCars')**

**INTO Company (ID, Name) VALUES (5, 'NextRide')**

**INTO Company (ID, Name) VALUES (6, 'UrbanMove')**

**INTO Company (ID, Name) VALUES (7, 'PrimeAuto')**

**INTO Company (ID, Name) VALUES (8, 'DrivePro')**

**INTO Company (ID, Name) VALUES (9, 'CarGo Rentals')**

**INTO Company (ID, Name) VALUES (10, 'RoadKing')**

**SELECT \* FROM dual;**

—-----------------VEHICLE—---------------

**INSERT ALL**

**INTO Vehicle (ID, Company\_id, CurrentMileage, Color, LastServiceDate,**

**DailyRate, SeatingCapacity, Status, Description)**

**VALUES (1,1,12000,'Red',DATE '2024-01-15',45.99,4,'Active','Compact car')**

**INTO Vehicle VALUES (2,2,30000,'Blue',DATE**

**'2024-02-10',55.50,5,'Active','Sedan')**

**INTO Vehicle VALUES (3,3,15000,'Black',DATE**

**'2024-01-20',60.00,7,'Active','SUV')**

**INTO Vehicle VALUES (4,4,50000,'White',DATE**

**'2023-12-15',40.00,4,'Inactive','Old model')**

**INTO Vehicle VALUES (5,5,10000,'Silver',DATE**

**'2024-03-01',80.00,2,'Active','Sports car')**

**INTO Vehicle VALUES (6,6,9000,'Gray',DATE**

**'2024-02-28',70.00,5,'Active','Hybrid')**

**INTO Vehicle VALUES (7,7,22000,'Blue',DATE**

**'2024-01-10',100.00,9,'Active','Van')**

**INTO Vehicle VALUES (8,8,8000,'Green',DATE '2024-02-15',50.00,4,'Active','Eco car')**

**INTO Vehicle VALUES (9,9,12000,'Red',DATE**

**'2024-03-05',90.00,5,'Active','Premium sedan')**

**INTO Vehicle VALUES (10,10,6000,'Black',DATE**

**'2024-02-25',110.00,7,'Active','Luxury SUV')**

**SELECT \* FROM dual;**

—------------PLATENUMBER—-------------

**INSERT ALL**

**INTO PlateNumber (PlateNumber, Vehicle\_id, Model, Year) VALUES**

**('ABC123',1,'Toyota Yaris',2020)**

**INTO PlateNumber VALUES ('XYZ456',2,'Honda Civic',2021)**

**INTO PlateNumber VALUES ('JKL789',3,'Nissan Patrol',2022)**

**INTO PlateNumber VALUES ('LMN321',4,'Ford Fiesta',2019)**

**INTO PlateNumber VALUES ('QWE234',5,'BMW Z4',2023)**

**INTO PlateNumber VALUES ('RTY567',6,'Toyota Prius',2022)**

**INTO PlateNumber VALUES ('UIO890',7,'Mercedes Sprinter',2021)**

**INTO PlateNumber VALUES ('PAS456',8,'Hyundai Ioniq',2023)**

**INTO PlateNumber VALUES ('DFG678',9,'Audi A4',2022)**

**PlateNumber VALUES ('HJK345',10,'Range Rover',2024)**

**SELECT \* FROM dual;**

—-----------ROLE—---------------

**INSERT ALL**

**INTO Role (ID, Name, Description) VALUES (1,'DRIVER','Assigned drivers')**

**INTO Role VALUES (2,'MECHANIC','Vehicle maintenance')**

**INTO Role VALUES (3,'OTHER STAFF','General staff')**

**INTO Role VALUES (4,'DRIVER','Additional driver role')**

**INTO Role VALUES (5,'MECHANIC','Extra mechanic')**

**INTO Role VALUES (6,'OTHER STAFF','Operations')**

**INTO Role VALUES (7,'DRIVER','Fleet driver')**

**INTO Role VALUES (8,'MECHANIC','Workshop mechanic')**

**INTO Role VALUES (9,'OTHER STAFF','Support staff')**

**INTO Role VALUES (10,'DRIVER','Backup driver')**

**SELECT \* FROM dual;**

—------------STAFF—-------------

**INSERT ALL**

**INTO Staff (ID, Role\_id, FirstName, LastName, HireDate, ContactNo, Email,**

**Salary, LicenseNo, ExperienceYears)**

**VALUES (1,1,'John','Smith',DATE**

**'2021-02-10','123456','john1@mail.com',2500,'LIC001',5)**

**INTO Staff VALUES (2,2,'Mike','Brown',DATE**

**'2020-01-05','234567','mike2@mail.com',3000,NULL,8)**

**INTO Staff VALUES (3,3,'Sara','Lee',DATE**

**'2022-03-12','345678','sara3@mail.com',2700,NULL,3)**

**INTO Staff VALUES (4,1,'David','White',DATE**

**'2019-07-18','456789','david4@mail.com',2600,'LIC004',6)**

**INTO Staff VALUES (5,2,'Emma','Clark',DATE**

**'2021-06-01','567890','emma5@mail.com',3100,NULL,7)**

**INTO Staff VALUES (6,3,'Ryan','Kim',DATE**

**'2023-01-20','678901','ryan6@mail.com',2400,NULL,2)**

**INTO Staff VALUES (7,1,'Chris','Stone',DATE**

**'2020-04-25','789012','chris7@mail.com',2550,'LIC007',4)**

**Staff VALUES (8,2,'Alex','Park',DATE**

**'2018-11-12','890123','alex8@mail.com',3200,NULL,10)**  **Staff VALUES (9,3,'Linda','Jones',DATE**

**'2022-09-30','901234','linda9@mail.com',2450,NULL,3)**

**INTO Staff VALUES (10,1,'Tom','Adams',DATE**

**'2023-05-15','012345','tom10@mail.com',2300,'LIC010',1)**

**SELECT \* FROM dual;**

—----------CUSTOMER—---------------

**INSERT ALL**

**INTO Customer (ID, FirstName, LastName, PhoneNumber, Email, Address,**

**BlackListStatus)**

**VALUES (1,'Adam','Stone','123123','adam@mail.com','NY','N')**

**INTO Customer VALUES (2,'Brian','Ray','234234','brian@mail.com','LA','N')**

**INTO Customer VALUES (3,'Cindy','Hope','345345','cindy@mail.com','TX','N')**

**INTO Customer VALUES (4,'Derek','Long','456456','derek@mail.com','NJ','Y')**

**INTO Customer VALUES (5,'Ella','Moon','567567','ella@mail.com','FL','N')**

**INTO Customer VALUES (6,'Frank','Cole','678678','frank@mail.com','AZ','N')**

**INTO Customer VALUES (7,'Grace','Kim','789789','grace@mail.com','WA','N')**

**INTO Customer VALUES (8,'Henry','Lee','890890','henry@mail.com','NV','N')**

**INTO Customer VALUES (9,'Ivy','Park','901901','ivy@mail.com','CO','N')**

**INTO Customer VALUES (10,'Jake','Hill','112112','jake@mail.com','GA','N') SELECT \* FROM dual;**

—-----------RESERVATION—--------------

**INSERT ALL**

**INTO Reservation (ID, Vehicle\_id, Customer\_id, Staff\_id, DriverRequired, Driver\_id,**

**Rent\_StartDate, Rent\_EndDate, RateNoBooking, PriceBasis, Status, PaymentStatus)**

**VALUES (1,1,1,1,'Y',101, DATE '2024-03-01', DATE '2024-03-05',200,'DAILY','Reserved','PAID')**

**INTO Reservation VALUES (2,2,2,2,'N',NULL, DATE '2024-03-10', DATE '2024-03-12',150,'DAILY','Reserved','UNPAID')**

**INTO Reservation VALUES (3,3,3,3,'Y',102, DATE '2024-03-15', DATE '2024-03-18',300,'WEEKLY','Reserved','PAID')**

**INTO Reservation VALUES (4,4,4,4,'N',NULL, DATE '2024-03-20', DATE '2024-03-22',120,'DAILY','Canceled','UNPAID')**

**INTO Reservation VALUES (5,5,5,5,'Y',103, DATE '2024-04-01', DATE '2024-04-02',500,'DAILY','Reserved','PAID')**

**INTO Reservation VALUES (6,6,6,6,'N',NULL, DATE '2024-04-05', DATE '2024-04-07',180,'DAILY','Free','UNPAID')**

**INTO Reservation VALUES (7,7,7,7,'Y',104, DATE '2024-04-10', DATE '2024-04-15',700,'WEEKLY','Reserved','PAID')**

**INTO Reservation VALUES (8,8,8,8,'N',NULL, DATE '2024-04-18', DATE '2024-04-20',120,'DAILY','Reserved','PAID')**

**INTO Reservation VALUES (9,9,9,9,'Y',105, DATE '2024-04-22', DATE '2024-04-25',350,'DAILY','Reserved','UNPAID')**

**INTO Reservation VALUES (10,10,10,10,'N',NULL, DATE '2024-04-28', DATE '2024-05-01',400,'DAILY','Reserved','PAID')**

**SELECT \* FROM dual;**

—-------------PAYMENT—------------

**INSERT ALL**

**INTO Payment (ID, Reservation\_id, Customer\_id, PaymentDate, Amount,**

**PaymentMethod)**

**VALUES (1,1,1,DATE '2024-03-01',200,'CASH')**

**INTO Payment VALUES (2,2,2,DATE '2024-03-10',150,'CREDITCARD')**

**INTO Payment VALUES (3,3,3,DATE '2024-03-15',100,'ONLINE TRANSACTION')**

**INTO Payment VALUES (4,4,4,DATE '2024-03-20',0,'CASH')**

**INTO Payment VALUES (5,5,5,DATE '2024-04-01',250,'CARD')**

**INTO Payment VALUES (6,6,6,DATE '2024-04-05',180,'CASH')**

**INTO Payment VALUES (7,7,7,DATE '2024-04-10',700,'CARD')**

**INTO Payment VALUES (8,8,8,DATE '2024-04-18',120,'CREDITCARD')**

**INTO Payment VALUES (9,9,9,DATE '2024-04-22',200,'ONLINE TRANSACTION')**

**INTO Payment VALUES (10,10,10,DATE '2024-04-28',400,'CARD')**  **SELECT \* FROM dual;**

—-----------MAINTENANCE—-----------

**INSERT ALL**

**Maintenance (ID, Vehicle\_id, Staff\_id, MaintenanceDate, Description, Cost)**

**VALUES (1,1,2,DATE '2024-02-01','Oil change',50)**

**INTO Maintenance VALUES (2,2,5,DATE '2024-02-05','Brake inspection',80)**

**INTO Maintenance VALUES (3,3,8,DATE '2024-02-10','Engine tuning',120)**

**INTO Maintenance VALUES (4,4,2,DATE '2024-02-12','Tire replacement',200)**

**INTO Maintenance VALUES (5,5,5,DATE '2024-02-15','Detailing',150)**

**Maintenance VALUES (6,6,8,DATE '2024-02-18','Hybrid check',180)**

**INTO Maintenance VALUES (7,7,2,DATE '2024-02-20','AC repair',130) INTO Maintenance VALUES (8,8,5,DATE '2024-02-22','Battery replacement',220)**

**INTO Maintenance VALUES (9,9,8,DATE '2024-02-25','Alignment',70)**

**INTO Maintenance VALUES (10,10,2,DATE '2024-02-28','Full service',300) SELECT \* FROM dual;**

—------------DAMAGE REPORT—------------

**INSERT ALL**

**INTO Damage\_Report (ID, Reservation\_id, Vehicle\_id, SeverityLevel,**

**Description, EstimatedCost, ReportDate)**

**VALUES (1,1,1,'LOW','Scratch on bumper',50,DATE '2024-03-05')**

**INTO Damage\_Report VALUES (2,2,2,'MEDIUM','Broken mirror',200,DATE '2024-03-12')**

**INTO Damage\_Report VALUES (3,3,3,'HIGH','Dent on side door',500,DATE '2024-03-18')**

**INTO Damage\_Report VALUES (4,4,4,'LOW','Small crack in light',80,DATE '2024-03-22')**

**INTO Damage\_Report VALUES (5,5,5,'MEDIUM','Interior damage',250,DATE '2024-04-02')**

**INTO Damage\_Report VALUES (6,6,6,'LOW','Windshield chip',100,DATE**

**'2024-04-07')**

**INTO Damage\_Report VALUES (7,7,7,'HIGH','Major collision repair',1500,DATE '2024-04-15')**

**INTO Damage\_Report VALUES (8,8,8,'LOW','Paint peel',60,DATE '2024-04-20')**

**INTO Damage\_Report VALUES (9,9,9,'MEDIUM','Tire damage',150,DATE**

**'2024-04-25')**

**INTO Damage\_Report VALUES (10,10,10,'HIGH','Suspension damage',900, DATE**  **'2024-05-01')**

**SELECT \* FROM dual;**

