### **1. Customers Table**

CREATE TABLE Customers (

Customer\_ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

First\_Name VARCHAR2(50) NOT NULL,

Last\_Name VARCHAR2(50) NOT NULL,

Email VARCHAR2(100) NOT NULL,

Phone\_Number VARCHAR2(15),

License\_Number VARCHAR2(20) UNIQUE NOT NULL,

Address VARCHAR2(255)

);

Primary Key: Customer\_ID

Unique Identifier: License\_Number (can serve as an alternate key)

Functional Dependencies:

Customer\_ID → First\_Name, Last\_Name, Email, Phone\_Number, License\_Number, Address

License\_Number → First\_Name, Last\_Name, Email, Phone\_Number, Address

#### 3NF Verification:

No Partial Dependencies: All non-key attributes depend on the primary key Customer\_ID.

No Transitive Dependencies: Each non-key attribute is directly determined by Customer\_ID or License\_Number. No attribute depends on another non-key attribute.

Conclusion: This table is in 3NF

### 

### 

### 

### 

### 

### 

### 

### 

### **2. Cars Table**

CREATE TABLE Cars (

Car\_ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

Make VARCHAR2(50) NOT NULL,

Model VARCHAR2(50) NOT NULL,

Year NUMBER(4) CHECK (Year >= 1886),

License\_Plate VARCHAR2(10) UNIQUE NOT NULL,

VIN VARCHAR2(17) UNIQUE NOT NULL,

Daily\_Rental\_Price NUMBER(10, 2) NOT NULL,

Availability\_Status VARCHAR2(20) CHECK (Availability\_Status IN ('Available', 'Rented', 'Under\_Maintenance')),

Location\_ID NUMBER REFERENCES Locations(Location\_ID)

);

Primary Key: Car\_ID

Unique Identifiers: License\_Plate and VIN (can serve as candidate keys)

Functional Dependencies:

Car\_ID → Make, Model, Year, License\_Plate, VIN, Daily\_Rental\_Price, Availability\_Status, Location\_ID

License\_Plate → Car\_ID, Make, Model, Year, VIN, Daily\_Rental\_Price, Availability\_Status, Location\_ID

VIN → Car\_ID, Make, Model, Year, License\_Plate, Daily\_Rental\_Price, Availability\_Status, Location\_ID

#### 3NF Verification:

No Partial Dependencies: All attributes are fully dependent on the primary key Car\_ID.

No Transitive Dependencies: All attributes are directly dependent on the primary key or candidate keys (License\_Plate and VIN). No attribute depends on another non-key attribute.

Conclusion: This table is in 3NF.

### 

### 

### 

### 

### **3. Rental\_Transactions Table**

CREATE TABLE Rental\_Transactions (

Rental\_ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

Customer\_ID NUMBER REFERENCES Customers(Customer\_ID),

Car\_ID NUMBER REFERENCES Cars(Car\_ID),

Rental\_Start\_Date DATE NOT NULL,

Rental\_End\_Date DATE,

Total\_Cost NUMBER(10, 2),

Status VARCHAR2(20) CHECK (Status IN ('Active', 'Completed', 'Cancelled'))

);

Primary Key: Rental\_ID

Functional Dependencies:

Rental\_ID → Customer\_ID, Car\_ID, Rental\_Start\_Date, Rental\_End\_Date, Total\_Cost, Status

#### 3NF Verification:

No Partial Dependencies: All attributes depend entirely on the primary key Rental\_ID.

No Transitive Dependencies: Each attribute is directly dependent on Rental\_ID, with no dependencies among non-key attributes.

Conclusion: This table is in 3NF.

### **4. Payments Table**

CREATE TABLE Payments (

Payment\_ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

Rental\_ID NUMBER REFERENCES Rental\_Transactions(Rental\_ID),

Payment\_Amount NUMBER(10, 2) NOT NULL,

Payment\_Date DATE NOT NULL,

Payment\_Method VARCHAR2(20) CHECK (Payment\_Method IN ('Credit Card', 'Cash', 'Debit'))

);

Primary Key: Payment\_ID

Functional Dependencies:

Payment\_ID → Rental\_ID, Payment\_Amount, Payment\_Date, Payment\_Method

#### 3NF Verification:

No Partial Dependencies: All attributes are fully dependent on the primary key Payment\_ID.

No Transitive Dependencies: Each attribute depends directly on Payment\_ID, and there are no dependencies among non-key attributes.

Conclusion: This table is in 3NF.

### **5. Car\_Maintenance Table**

CREATE TABLE Car\_Maintenance (

Maintenance\_ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

Car\_ID NUMBER REFERENCES Cars(Car\_ID),

Maintenance\_Date DATE NOT NULL,

Description VARCHAR2(255),

Maintenance\_Cost NUMBER(10, 2)

);

Primary Key: Maintenance\_ID

Functional Dependencies:

Maintenance\_ID → Car\_ID, Maintenance\_Date, Description, Maintenance\_Cost

#### 3NF Verification:

No Partial Dependencies: All attributes are fully dependent on the primary key Maintenance\_ID.

No Transitive Dependencies: All non-key attributes depend directly on Maintenance\_ID, with no dependencies among them.

Conclusion: This table is in 3NF.

### **6. Locations Table**

CREATE TABLE Locations (

Location\_ID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

Location\_Name VARCHAR2(100) NOT NULL,

Address VARCHAR2(255) NOT NULL,

Phone\_Number VARCHAR2(15)

);

Primary Key: Location\_ID

Functional Dependencies:

Location\_ID → Location\_Name, Address, Phone\_Number

Location\_Name → Location\_ID, Address, Phone\_Number

#### 3NF Verification:

No Partial Dependencies: All attributes are fully dependent on the primary key Location\_ID.

No Transitive Dependencies: Each attribute is dependent on the primary key or candidate key (Location\_Name) with no transitive dependencies.

Conclusion: This table is in 3NF.

### **Algorithm**

### 

### **1. Identify Functional Dependencies**

### **2. Make Sure the Table is in First Normal Form (1NF)**

It should not have repeated groups or arrays.

### **3. Ensure Second Normal Form (2NF)**

For tables with composite keys, make sure that every non-key column is fully dependent on the entire primary key.

### **4. Remove Transitive Dependencies to Achieve 3NF**

Look for transitive dependencies. A transitive dependency is when a non-key column depends on another non-key column, rather than the primary key.

### **5. Add Foreign Keys to Preserve Relationships**

After breaking out any dependencies into new tables, add foreign keys to keep the tables connected.