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)

);

Functional Dependencies:

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

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

Candidate Keys:

Customer\_ID, License\_Number (both are unique)

BCNF Verification:

Both FDs have determinants (Customer\_ID and License\_Number) that are candidate keys

Conclusion:

This table is in BCNF

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)

);

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

Candidate Keys:

Car\_ID, License\_Plate, VIN (all are unique).

BCNF Verification:

In each FD, the left side (Car\_ID, License\_Plate, VIN) is a candidate key, satisfying BCNF.

Conclusion:

This table is in BCNF

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'))

);

Functional Dependencies:

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

Candidate Key:

Rental\_ID (unique).

BCNF Verification:

The only FD (Rental\_ID → all other columns) has Rental\_ID as its determinant, which is a candidate key.

Conclusion:

This table is in BCNF.

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'))

);

Functional Dependencies:

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

Candidate Key:

Payment\_ID (unique).

BCNF Verification:

The FD (Payment\_ID → all other columns) has Payment\_ID as its determinant, which is a candidate key.

Conclusion:

This table is in BCNF.

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)

);

Functional Dependencies:

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

Candidate Keys:

Maintenance\_ID, Car\_ID + Maintenance\_Date (unique).

BCNF Verification:

Each FD (Maintenance\_ID or Car\_ID + Maintenance\_Date) has a candidate key on the left side, satisfying BCNF.

Conclusion:

This table is in BCNF.

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)

);

Functional Dependencies:

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

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

Candidate Keys:

Location\_ID, Location\_Name (unique).

BCNF Verification:

Each FD (Location\_ID or Location\_Name) has a candidate key on the left side, satisfying BCNF.

Conclusion:

This table is in BCNF.

**Algorithm**

1. Identify Functional Dependencies (FDs) for each table.
2. Identify Candidate Keys for each table.
3. For each FD, verify if the left side is a superkey (a key that uniquely identifies a row). If all FDs satisfy this, the table is in BCNF.
4. If any FD does not satisfy this rule, decompose the table based on that FD until all tables meet BCNF.

python -m venv .venv

source .venv/bin/activate # Activate the virtual environment

pip install -r requirements.txt # Install dependencies