NWU ®

Phase 3: Physical Design Documentation

Table Definitions

Vehicle Table

```
CREATE TABLE Vehicle (

        VEHICLE_ID INT PRIMARY KEY,

        VEHICLE_COLOUR VARCHAR(10),

        VEHICLE_TYPE VARCHAR(15),

        AVAILABILITY_ID INT,

        VEHICLE_MILEAGE DECIMAL(10, 2),

        VEHICLE_RENTAL_PRICE DECIMAL(10, 2),

        MAINTENANCE_ID INT,

        VEHICLE_MAKE VARCHAR(15)

);
```

ALTER TABLE Vehicle

```
ADD CHECK (VEHICLE_RENTAL_PRICE >= 0);
```

```
ALTER TABLE Vehicle

ADD FOREIGN KEY(AVAILABILITY_ID) REFERENCES
Vehicle_availability(AVAILABILITY_ID)

ON DELETE SET NULL;
```

```
ALTER TABLE Vehicle
```

```
ADD FOREIGN KEY(MAINTENANCE_ID) REFERENCES
Vehicle_maintenance(MAINTENANCE_ID)

ON DELETE SET NULL;
```

Employee Table

```
CREATE TABLE Employee (

        EMPLOYEE_ID INT PRIMARY KEY,

        EMPLOYEE_NAME VARCHAR(20),

        EMPLOYEE_SURNAME VARCHAR(20),

        EMPLOYEE_PHONE VARCHAR(10),

   EMPLOYEE_EMAIL VARCHAR(50),

        EMPLOYEE_POSITION VARCHAR(15),

        BRANCH_ID INT,

        FOREIGN KEY (BRANCH_ID) REFERENCES Branch(BRANCH_ID) ON DELETE
SET NULL

);
```

Customer Table

```
CREATE TABLE Customer (

        CUSTOMER_ID INT PRIMARY KEY,

        CUSTOMER_NAME VARCHAR(20),

        CUSTOMER_SURNAME VARCHAR(20),

        CUSTOMER_PHONE VARCHAR(10),

        CUSTOMER_EMAIL VARCHAR(50),
        ADDRESS_ID INT,

        FOREIGN KEY (ADDRESS_ID) REFERENCES Address(ADDRESS_ID) ON DELETE SET NULL,

        CUSTOMER_DATE_OF_BIRTH DATE NOT NULL,

        CUSTOMER_AGE DECIMAL(5, 0),

        AGENT_ID INT,

        FOREIGN KEY (AGENT_ID) REFERENCES Employee(EMPLOYEE_ID) ON DELETE SET NULL

);

-- Getting the customer's age

UPDATE Customer

SET CUSTOMER_AGE = TRUNC((SYSDATE - CUSTOMER_DATE_OF_BIRTH) / 365);
```

Branch Table

```
CREATE TABLE Branch (

    BRANCH_ID INT PRIMARY KEY,

    ADDRESS_ID INT,

    FOREIGN KEY (ADDRESS_ID) REFERENCES Address(ADDRESS_ID) ON DELETE SET NULL,

    BRANCH_PHONE VARCHAR(10),

    BRANCH_EMAIL VARCHAR(50),

    MANAGER_ID INT,

    FOREIGN KEY (MANAGER_ID) REFERENCES Employee(EMPLOYEE_ID) ON DELETE SET NULL,
        BRANCH_NAME VARCHAR(20)
);
```

Rental Transaction Table

```
 CREATE TABLE Rental_transaction (
        RENTAL_ID INT PRIMARY KEY,
        CUSTOMER_ID INT,
        FOREIGN KEY (CUSTOMER_ID) REFERENCES Customer(CUSTOMER_ID) ON DELETE SET NULL,
        VEHICLE_ID INT,
        FOREIGN KEY (VEHICLE_ID) REFERENCES Vehicle(VEHICLE_ID) ON DELETE SET NULL,
        AGENT_ID INT,
        FOREIGN KEY (AGENT_ID) REFERENCES Employee(EMPLOYEE_ID) ON DELETE SET NULL,
        RENTAL_START_DATE DATE,
        RENTAL_END_DATE DATE,
        RENTAL_DURATION_IN_DAYS AS (TRUNC(RENTAL_END_DATE - RENTAL_START_DATE)),
        RENTAL_COST DECIMAL(10, 2),
        LATE_FEE_RATE DECIMAL(5, 2) DEFAULT 100.00,
        RENTAL_STATUS VARCHAR(15) DEFAULT 'Rented'
);

-- Determines the rental cost with a late fee

UPDATE Rental_transaction
```

```
SET RENTAL_COST = (SELECT VEHICLE_RENTAL_PRICE + (TRUNC(SYSDATE -
RENTAL_END_DATE) * LATE_FEE_RATE)

                    FROM Vehicle

                    WHERE Vehicle.VEHICLE_ID = Rental_transaction.VEHICLE_ID AND
RENTAL_STATUS = 'Returned');

-- Determines the rental cost without a late fee

UPDATE Rental_transaction

SET RENTAL_COST = (SELECT VEHICLE_RENTAL_PRICE

                    FROM Vehicle

                    WHERE Vehicle.VEHICLE_ID = Rental_transaction.VEHICLE_ID AND
RENTAL_STATUS = 'Returned');

ALTER TABLE Rental_transaction

ADD CHECK (RENTAL_COST >= 0);
```

Vehicle Maintenance Table

```
CREATE TABLE Vehicle_availability (

          AVAILABILITY_ID INT,

          VEHICLE_ID INT,

          PRIMARY KEY (AVAILABILITY_ID, VEHICLE_ID),

          FOREIGN KEY (VEHICLE_ID) REFERENCES Vehicle(VEHICLE_ID),

          VEHICLE_CONDITION VARCHAR(10),

          VEHICLE_AVAILABILITY VARCHAR(15),

          UNIQUE (VEHICLE_ID),
        UNIQUE (AVAILABILITY_ID)
);

-- Determines vehicle availability

UPDATE Vehicle_availability

SET VEHICLE_AVAILABILITY = 'Available'

WHERE VEHICLE_CONDITION = 'Good'

AND EXISTS (SELECT 1 FROM Rental_transaction

        WHERE Rental_transaction.RENTAL_STATUS = 'Returned');

UPDATE Vehicle_availability
SET VEHICLE_AVAILABILITY = 'Not available'
WHERE VEHICLE_CONDITION = 'Bad'
OR EXISTS (SELECT 1 FROM Rental_transaction

        WHERE Rental_transaction.RENTAL_STATUS = 'Not returned');
```

Vehicle Availability Table

```
 CREATE TABLE Vehicle_availability (

        AVAILABILITY_ID INT,

        VEHICLE_ID INT,

        PRIMARY KEY (AVAILABILITY_ID, VEHICLE_ID),

        FOREIGN KEY (VEHICLE_ID) REFERENCES Vehicle(VEHICLE_ID),

        VEHICLE_CONDITION VARCHAR(10),

        VEHICLE_AVAILABILITY VARCHAR(15),

        UNIQUE (VEHICLE_ID),

        UNIQUE (AVAILABILITY_ID)

);
```

```
-- Determines vehicle availability

UPDATE Vehicle_availability

SET VEHICLE_AVAILABILITY = 'Available'

WHERE VEHICLE_CONDITION = 'Good'

AND EXISTS (SELECT 1 FROM Rental_transaction

        WHERE Rental_transaction.RENTAL_STATUS = 'Returned');


UPDATE Vehicle_availability

SET VEHICLE_AVAILABILITY = 'Not available'

WHERE VEHICLE_CONDITION = 'Bad'

OR EXISTS (SELECT 1 FROM Rental_transaction

        WHERE Rental_transaction.RENTAL_STATUS = 'Not returned');
```

Address Table

```
 CREATE TABLE Address (

        ADDRESS_ID INT PRIMARY KEY,

        STREET_NAME VARCHAR(20),

        CITY VARCHAR(20),

   PROVINCE VARCHAR(20),

        POSTAL_CODE VARCHAR(4)

);
```

Banking Information Table

```
CREATE TABLE Banking_information (

    CUSTOMER_ID INT PRIMARY KEY,

    BANK_NAME VARCHAR(20),

    CARD_NUMBER VARCHAR(16),

    FOREIGN KEY (CARD_NUMBER) REFERENCES Card(CARD_NUMBER) ON DELETE SET NULL,

    ACCOUNT_NUMBER VARCHAR(10),

    ACCOUNT_TYPE VARCHAR(10)

);
```

Card Table

```
CREATE TABLE Card (

    CARD_NUMBER VARCHAR(16) PRIMARY KEY,

    EXPIRATION_DATE DATE

);
```

Views

Popular Vehicles View

```
CREATE VIEW Popular_Vehicles AS

SELECT VEHICLE_TYPE, COUNT(RENTAL_ID) AS Rental_Count

FROM Rental_transaction

JOIN Vehicle ON Rental_transaction.VEHICLE_ID = Vehicle.VEHICLE_ID

GROUP BY VEHICLE_TYPE

ORDER BY Rental_Count DESC;
```

Available Vehicles View

```
CREATE VIEW Available_vehicles AS
SELECT VEHICLE_ID, VEHICLE_TYPE, VEHICLE_COLOUR
FROM Vehicle
WHERE VEHICLE_ID IN (
        SELECT VEHICLE_ID
        FROM Vehicle_availability
        WHERE VEHICLE_AVAILABILITY = 'Available'
);
```

Overdue Rentals View

```
CREATE VIEW Overdue_Rentals AS
SELECT RENTAL_ID, RENTAL_END_DATE, VEHICLE_TYPE, VEHICLE_COLOUR,
CUSTOMER_NAME, CUSTOMER_SURNAME
FROM Rental_transaction
JOIN Vehicle ON Rental_transaction.VEHICLE_ID = Vehicle.VEHICLE_ID
JOIN Customer ON Rental_transaction.CUSTOMER_ID = Customer.CUSTOMER_ID
WHERE RENTAL_END_DATE < SYSDATE;
```

Revenue Summary View

```
CREATE VIEW Revenue_Summary AS
SELECT SUM(RENTAL_COST) AS Total_Revenue,
        TRUNC(RENTAL_START_DATE) AS Rental_Date
FROM Rental_transaction
```

```
GROUP BY TRUNC(RENTAL_START_DATE);
```

Customer Rentals View

```
CREATE VIEW Customer_Rentals AS

SELECT RENTAL_ID, RENTAL_START_DATE, RENTAL_END_DATE, VEHICLE_TYPE,
VEHICLE_COLOUR

FROM Rental_transaction

JOIN Vehicle ON Rental_transaction.VEHICLE_ID = Vehicle.VEHICLE_ID

JOIN Customer ON Rental_transaction.CUSTOMER_ID = Customer.CUSTOMER_ID;;
```

Total Revenue Per Day over R5000 View

```
CREATE VIEW Total_Revenue_Per_Day_over_R5000 AS

SELECT TRUNC(RENTAL_START_DATE) AS Rental_Date, SUM(RENTAL_COST) AS
Total_Revenue

FROM Rental_transaction

GROUP BY TRUNC(RENTAL_START_DATE)

HAVING SUM(RENTAL_COST) > 5000;
```

Search Employees View

```
CREATE VIEW Search_Employees AS

SELECT EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_SURNAME,
EMPLOYEE_PHONE, EMPLOYEE_EMAIL

FROM Employee

WHERE EMPLOYEE_POSITION LIKE '%Manager%';
```

Search Vehicles View

```
CREATE VIEW Search_Vehicles AS
```

```
SELECT VEHICLE_ID, VEHICLE_TYPE, VEHICLE_COLOUR, VEHICLE_AVAILABILITY

FROM Vehicle

JOIN Vehicle_availability ON Vehicle.VEHICLE_ID = Vehicle_availability.VEHICLE_ID

WHERE VEHICLE_TYPE LIKE '%sedan%' OR VEHICLE_COLOUR = 'Red';
```

Available Sedans View

```
CREATE VIEW Available_sedans AS

SELECT VEHICLE_ID, VEHICLE_TYPE, VEHICLE_COLOUR, VEHICLE_RENTAL_PRICE

FROM Vehicle

JOIN Vehicle_availability ON Vehicle.VEHICLE_ID = Vehicle_availability.VEHICLE_ID

WHERE VEHICLE_TYPE = 'Sedan' AND VEHICLE_AVAILABILITY = 'Available';
```

Indexes

Index on Customer Phone

```
CREATE INDEX idx_customer_phone ON Customer(CUSTOMER_PHONE);
```

Index on Employee Email

```
CREATE INDEX idx_employee_email ON Employee(EMPLOYEE_EMAIL);
```

Index on Rental Transaction Dates

```
CREATE INDEX idx_rental_dates ON Rental_transaction(RENTAL_START_DATE,
RENTAL_END_DATE);
```

Index on Vehicle Type

```
CREATE INDEX idx_vehicle_type ON Vehicle(VEHICLE_TYPE);
```

Sample Data Insertion

Inserting into Address Table

```
INSERT INTO Address (ADDRESS_ID, STREET_NAME, CITY, PROVINCE,
POSTAL_CODE) VALUES (1, 'Main St', 'Springfield', 'IL', '62704');
```

```
INSERT INTO Address (ADDRESS_ID, STREET_NAME, CITY, PROVINCE,
POSTAL_CODE) VALUES (2, '2nd St', 'Springfield', 'IL', '62704');
```

Inserting into Branch Table

```
INSERT INTO Branch (BRANCH_ID, ADDRESS_ID, BRANCH_PHONE, BRANCH_EMAIL,
MANAGER_ID, BRANCH_NAME) VALUES (1, 1, '2175551234', 'branch1@xyz.com', 1,
'Springfield North');
```

```
INSERT INTO Branch (BRANCH_ID, ADDRESS_ID, BRANCH_PHONE, BRANCH_EMAIL,
MANAGER_ID, BRANCH_NAME) VALUES (2, 2, '2175555678', 'branch2@xyz.com', 2,
'Springfield South');
```

Inserting into Employee Table

```
INSERT INTO Employee (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_SURNAME,
EMPLOYEE_PHONE, EMPLOYEE_EMAIL, EMPLOYEE_POSITION, BRANCH_ID)
VALUES (1, 'John', 'Doe', '2175551001', 'johndoe@xyz.com', 'Manager', 1);
```

```
INSERT INTO Employee (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_SURNAME,
EMPLOYEE_PHONE, EMPLOYEE_EMAIL, EMPLOYEE_POSITION, BRANCH_ID)
VALUES (2, 'Jane', 'Smith', '2175551002', 'janesmith@xyz.com', 'Manager', 2);
```

Inserting into Customer Table

```
INSERT INTO Customer (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_SURNAME,
CUSTOMER_PHONE, CUSTOMER_EMAIL, ADDRESS_ID,
CUSTOMER_DATE_OF_BIRTH, CUSTOMER_AGE, AGENT_ID) VALUES (1, 'Alice',
'Johnson', '2175552001', 'alicej@xyz.com', 1, TO_DATE('1990-01-01', 'YYYY-MM-DD'), 34,
1);
```

```
INSERT INTO Customer (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_SURNAME,
CUSTOMER_PHONE, CUSTOMER_EMAIL, ADDRESS_ID,
CUSTOMER_DATE_OF_BIRTH, CUSTOMER_AGE, AGENT_ID) VALUES (2, 'Bob',
'Williams', '2175552002', 'bobw@xyz.com', 2, TO_DATE('1985-05-15', 'YYYY-MM-DD'), 39, 2);
```

Inserting into Vehicle Table

```
INSERT INTO Vehicle (VEHICLE_ID, VEHICLE_COLOUR, VEHICLE_TYPE, AVAILABILITY_ID, VEHICLE_MILEAGE, VEHICLE_RENTAL_PRICE, MAINTENANCE_ID, VEHICLE_MAKE) VALUES (1, 'Red', 'Sedan', 1, 15000, 50.00, 1, 'Toyota');
```

```
INSERT INTO Vehicle (VEHICLE_ID, VEHICLE_COLOUR, VEHICLE_TYPE, AVAILABILITY_ID, VEHICLE_MILEAGE, VEHICLE_RENTAL_PRICE, MAINTENANCE_ID, VEHICLE_MAKE) VALUES (2, 'Blue', 'SUV', 2, 20000, 70.00, 2, 'Honda');
```

Inserting into Vehicle Availability Table

```
INSERT INTO Vehicle_availability (AVAILABILITY_ID, VEHICLE_ID, VEHICLE_CONDITION, VEHICLE_AVAILABILITY) VALUES (1, 1, 'Good', 'Available');
```

```
INSERT INTO Vehicle_availability (AVAILABILITY_ID, VEHICLE_ID, VEHICLE_CONDITION, VEHICLE_AVAILABILITY) VALUES (2, 2, 'Good', 'Available');
```

Inserting into Vehicle Maintenance Table

```
INSERT INTO Vehicle_maintenance (MAINTENANCE_ID, VEHICLE_ID, MAINTENANCE_TYPE, MECHANIC_ID, MAINTENANCE_DATE) VALUES (1, 1, 'Oil Change', 1, TO_DATE('2023-01-01', 'YYYY-MM-DD'));
```

```
INSERT INTO Vehicle_maintenance (MAINTENANCE_ID, VEHICLE_ID, MAINTENANCE_TYPE, MECHANIC_ID, MAINTENANCE_DATE) VALUES (2, 2, 'Brake Inspection', 2, TO_DATE('2023-02-01', 'YYYY-MM-DD'));
```

Inserting into Rental Transaction Table

```
INSERT INTO Rental_transaction (RENTAL_ID, CUSTOMER_ID, VEHICLE_ID, AGENT_ID, RENTAL_START_DATE, RENTAL_END_DATE, RENTAL_COST, RENTAL_STATUS) VALUES (1, 1, 1, 1, TO_DATE('2023-03-01', 'YYYY-MM-DD'), TO_DATE('2023-03-10', 'YYYY-MM-DD'), 500.00, 'Returned');
INSERT INTO Rental_transaction (RENTAL_ID, CUSTOMER_ID, VEHICLE_ID, AGENT_ID,
RENTAL_START_DATE, RENTAL_END_DATE, RENTAL_COST, RENTAL_STATUS)
VALUES (2, 2, 2, 2, TO_DATE('2023-04-01', 'YYYY-MM-DD'), TO_DATE('2023-04-05',
'YYYY-MM-DD'), 350.00, 'Returned');
```

Inserting into Banking Information Table

```
INSERT INTO Banking_information (CUSTOMER_ID, BANK_NAME, CARD_NUMBER,
ACCOUNT_NUMBER, ACCOUNT_TYPE) VALUES (1, 'Bank of America',
'1234567812345678', '100200300', 'Checking');

INSERT INTO Banking_information (CUSTOMER_ID, BANK_NAME, CARD_NUMBER,
ACCOUNT_NUMBER, ACCOUNT_TYPE) VALUES (2, 'Chase', '8765432187654321',
'400500600', 'Savings');
```

Inserting into Card Table

```
INSERT INTO Card (CARD_NUMBER, EXPIRATION_DATE) VALUES
('1234567812345678', TO_DATE('2025-12-31', 'YYYY-MM-DD'));

INSERT INTO Card (CARD_NUMBER, EXPIRATION_DATE) VALUES
('8765432187654321', TO_DATE('2026-06-30', 'YYYY-MM-DD'));
```

Vehicle Table

```
CREATE TABLE Vehicle (

        VEHICLE_ID INT PRIMARY KEY,

        VEHICLE_COLOUR VARCHAR(10),

        VEHICLE_TYPE VARCHAR(15),

        AVAILABILITY_ID INT,

        VEHICLE_MILEAGE DECIMAL(10, 2),

        VEHICLE_RENTAL_PRICE DECIMAL(10, 2),

        MAINTENANCE_ID INT,
        VEHICLE_MAKE VARCHAR(15)

);


ALTER TABLE Vehicle

ADD CHECK (VEHICLE_RENTAL_PRICE >= 0);


ALTER TABLE Vehicle

ADD FOREIGN KEY(AVAILABILITY_ID) REFERENCES
Vehicle_availability(AVAILABILITY_ID)

ON DELETE SET NULL;


ALTER TABLE Vehicle

ADD FOREIGN KEY(MAINTENANCE_ID) REFERENCES
Vehicle_maintenance(MAINTENANCE_ID)

ON DELETE SET NULL;
```

Employee Table

```
CREATE TABLE Employee (

        EMPLOYEE_ID INT PRIMARY KEY,

        EMPLOYEE_NAME VARCHAR(20),

        EMPLOYEE_SURNAME VARCHAR(20),

        EMPLOYEE_PHONE VARCHAR(10),

   EMPLOYEE_EMAIL VARCHAR(50),
	EMPLOYEE_POSITION VARCHAR(15),

	BRANCH_ID INT,

	FOREIGN KEY (BRANCH_ID) REFERENCES Branch(BRANCH_ID) ON DELETE SET NULL

);
```

Customer Table

```
CREATE TABLE Customer (

	CUSTOMER_ID INT PRIMARY KEY,

	CUSTOMER_NAME VARCHAR(20),

	CUSTOMER_SURNAME VARCHAR(20),

	CUSTOMER_PHONE VARCHAR(10),

	CUSTOMER_EMAIL VARCHAR(50),

	ADDRESS_ID INT,

	FOREIGN KEY (ADDRESS_ID) REFERENCES Address(ADDRESS_ID) ON DELETE SET NULL,

	CUSTOMER_DATE_OF_BIRTH DATE NOT NULL,

	CUSTOMER_AGE DECIMAL(5, 0),

	AGENT_ID INT,

	FOREIGN KEY (AGENT_ID) REFERENCES Employee(EMPLOYEE_ID) ON DELETE SET NULL

);



-- Getting the customer's age
```

```
UPDATE Customer

SET CUSTOMER_AGE = TRUNC((SYSDATE - CUSTOMER_DATE_OF_BIRTH) / 365);
```

Branch Table

```
CREATE TABLE Branch (

        BRANCH_ID INT PRIMARY KEY,

        ADDRESS_ID INT,

        FOREIGN KEY (ADDRESS_ID) REFERENCES Address(ADDRESS_ID) ON DELETE SET NULL,

        BRANCH_PHONE VARCHAR(10),

        BRANCH_EMAIL VARCHAR(50),

        MANAGER_ID INT,

        FOREIGN KEY (MANAGER_ID) REFERENCES Employee(EMPLOYEE_ID) ON DELETE SET NULL,

        BRANCH_NAME VARCHAR(20)

);
```

Rental Transaction Table

```
 CREATE TABLE Rental_transaction (

        RENTAL_ID INT PRIMARY KEY,

        CUSTOMER_ID INT,

        FOREIGN KEY (CUSTOMER_ID) REFERENCES Customer(CUSTOMER_ID) ON DELETE SET NULL,

        VEHICLE_ID INT,

        FOREIGN KEY (VEHICLE_ID) REFERENCES Vehicle(VEHICLE_ID) ON DELETE SET NULL,
        AGENT_ID INT,
        FOREIGN KEY (AGENT_ID) REFERENCES Employee(EMPLOYEE_ID) ON DELETE SET NULL,
        RENTAL_START_DATE DATE,
        RENTAL_END_DATE DATE,
        RENTAL_DURATION_IN_DAYS AS (TRUNC(RENTAL_END_DATE - RENTAL_START_DATE)),
        RENTAL_COST DECIMAL(10, 2),
        LATE_FEE_RATE DECIMAL(5, 2) DEFAULT 100.00,
        RENTAL_STATUS VARCHAR(15) DEFAULT 'Rented'
    );


    -- Determines the rental cost with a late fee
    UPDATE Rental_transaction
    SET RENTAL_COST = (SELECT VEHICLE_RENTAL_PRICE + (TRUNC(SYSDATE - RENTAL_END_DATE) * LATE_FEE_RATE)
            FROM Vehicle
            WHERE Vehicle.VEHICLE_ID = Rental_transaction.VEHICLE_ID AND RENTAL_STATUS = 'Returned');


    -- Determines the rental cost without a late fee
    UPDATE Rental_transaction
    SET RENTAL_COST = (SELECT VEHICLE_RENTAL_PRICE
                        FROM Vehicle
                        WHERE Vehicle.VEHICLE_ID = Rental_transaction.VEHICLE_ID AND
    RENTAL_STATUS = 'Returned');


    ALTER TABLE Rental_transaction
    ADD CHECK (RENTAL_COST >= 0);
```

Vehicle Maintenance Table

```
CREATE TABLE Vehicle_availability (
          AVAILABILITY_ID INT,
          VEHICLE_ID INT,
          PRIMARY KEY (AVAILABILITY_ID, VEHICLE_ID),
          FOREIGN KEY (VEHICLE_ID) REFERENCES Vehicle(VEHICLE_ID),
          VEHICLE_CONDITION VARCHAR(10),
          VEHICLE_AVAILABILITY VARCHAR(15),
          UNIQUE (VEHICLE_ID),
          UNIQUE (AVAILABILITY_ID)
);


-- Determines vehicle availability
UPDATE Vehicle_availability
SET VEHICLE_AVAILABILITY = 'Available'
WHERE VEHICLE_CONDITION = 'Good'
```

```
AND EXISTS (SELECT 1 FROM Rental_transaction
        WHERE Rental_transaction.RENTAL_STATUS = 'Returned');


UPDATE Vehicle_availability
SET VEHICLE_AVAILABILITY = 'Not available'
WHERE VEHICLE_CONDITION = 'Bad'
OR EXISTS (SELECT 1 FROM Rental_transaction
        WHERE Rental_transaction.RENTAL_STATUS = 'Not returned');
```

Vehicle Availability Table

```
 CREATE TABLE Vehicle_availability (
        AVAILABILITY_ID INT,
        VEHICLE_ID INT,
        PRIMARY KEY (AVAILABILITY_ID, VEHICLE_ID),
        FOREIGN KEY (VEHICLE_ID) REFERENCES Vehicle(VEHICLE_ID),
        VEHICLE_CONDITION VARCHAR(10),
        VEHICLE_AVAILABILITY VARCHAR(15),
        UNIQUE (VEHICLE_ID),
        UNIQUE (AVAILABILITY_ID)
);


-- Determines vehicle availability
UPDATE Vehicle_availability
```

```
SET VEHICLE_AVAILABILITY = 'Available'
WHERE VEHICLE_CONDITION = 'Good'
AND EXISTS (SELECT 1 FROM Rental_transaction
        WHERE Rental_transaction.RENTAL_STATUS = 'Returned');
UPDATE Vehicle_availability
SET VEHICLE_AVAILABILITY = 'Not available'
WHERE VEHICLE_CONDITION = 'Bad'
OR EXISTS (SELECT 1 FROM Rental_transaction
        WHERE Rental_transaction.RENTAL_STATUS = 'Not returned');
```

Address Table

```
 CREATE TABLE Address (
        ADDRESS_ID INT PRIMARY KEY,
        STREET_NAME VARCHAR(20),
        CITY VARCHAR(20),
   PROVINCE VARCHAR(20),
        POSTAL_CODE VARCHAR(4)
);
```

Banking Information Table

```
CREATE TABLE Banking_information (
        CUSTOMER_ID INT PRIMARY KEY,
        BANK_NAME VARCHAR(20),
        CARD_NUMBER VARCHAR(16),
        FOREIGN KEY (CARD_NUMBER) REFERENCES Card(CARD_NUMBER) ON DELETE SET NULL,
        ACCOUNT_NUMBER VARCHAR(10),
        ACCOUNT_TYPE VARCHAR(10)
);
```

Card Table

```
CREATE TABLE Card (
    CARD_NUMBER VARCHAR(16) PRIMARY KEY,
    EXPIRATION_DATE DATE
);
```

Views

Popular Vehicles View

```
CREATE VIEW Popular_Vehicles AS
SELECT VEHICLE_TYPE, COUNT(RENTAL_ID) AS Rental_Count
FROM Rental_transaction
JOIN Vehicle ON Rental_transaction.VEHICLE_ID = Vehicle.VEHICLE_ID
GROUP BY VEHICLE_TYPE
ORDER BY Rental_Count DESC;
```

Available Vehicles View

```
CREATE VIEW Available_vehicles AS
SELECT VEHICLE_ID, VEHICLE_TYPE, VEHICLE_COLOUR
FROM Vehicle
```

```
WHERE VEHICLE_ID IN (
        SELECT VEHICLE_ID
        FROM Vehicle_availability
        WHERE VEHICLE_AVAILABILITY = 'Available'
);
```

Overdue Rentals View

```
CREATE VIEW Overdue_Rentals AS
SELECT RENTAL_ID, RENTAL_END_DATE, VEHICLE_TYPE, VEHICLE_COLOUR, CUSTOMER_NAME, CUSTOMER_SURNAME
FROM Rental_transaction
JOIN Vehicle ON Rental_transaction.VEHICLE_ID = Vehicle.VEHICLE_ID
JOIN Customer ON Rental_transaction.CUSTOMER_ID = Customer.CUSTOMER_ID
WHERE RENTAL_END_DATE < SYSDATE;
```

Revenue Summary View

```
CREATE VIEW Revenue_Summary AS
SELECT SUM(RENTAL_COST) AS Total_Revenue,
        TRUNC(RENTAL_START_DATE) AS Rental_Date
FROM Rental_transaction
GROUP BY TRUNC(RENTAL_START_DATE);
```

Customer Rentals View

```
CREATE VIEW Customer_Rentals AS
```

```
SELECT RENTAL_ID, RENTAL_START_DATE, RENTAL_END_DATE, VEHICLE_TYPE, VEHICLE_COLOUR

FROM Rental_transaction

JOIN Vehicle ON Rental_transaction.VEHICLE_ID = Vehicle.VEHICLE_ID

JOIN Customer ON Rental_transaction.CUSTOMER_ID = Customer.CUSTOMER_ID;;
```

Total Revenue Per Day over R5000 View

```
CREATE VIEW Total_Revenue_Per_Day_over_R5000 AS

SELECT TRUNC(RENTAL_START_DATE) AS Rental_Date, SUM(RENTAL_COST) AS Total_Revenue

FROM Rental_transaction

GROUP BY TRUNC(RENTAL_START_DATE)

HAVING SUM(RENTAL_COST) > 5000;
```

Search Employees View

```
CREATE VIEW Search_Employees AS

SELECT EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_SURNAME, EMPLOYEE_PHONE, EMPLOYEE_EMAIL

FROM Employee

WHERE EMPLOYEE_POSITION LIKE '%Manager%';
```

Search Vehicles View

```
CREATE VIEW Search_Vehicles AS

SELECT VEHICLE_ID, VEHICLE_TYPE, VEHICLE_COLOUR, VEHICLE_AVAILABILITY

FROM Vehicle

JOIN Vehicle_availability ON Vehicle.VEHICLE_ID = Vehicle_availability.VEHICLE_ID
```

```
WHERE VEHICLE_TYPE LIKE '%sedan%' OR VEHICLE_COLOUR = 'Red';
```

Available Sedans View

```
CREATE VIEW Available_sedans AS

SELECT VEHICLE_ID, VEHICLE_TYPE, VEHICLE_COLOUR, VEHICLE_RENTAL_PRICE

FROM Vehicle

JOIN Vehicle_availability ON Vehicle.VEHICLE_ID = Vehicle_availability.VEHICLE_ID

WHERE VEHICLE_TYPE = 'Sedan' AND VEHICLE_AVAILABILITY = 'Available';
```

Indexes

Index on Customer Phone

```
CREATE INDEX idx_customer_phone ON Customer(CUSTOMER_PHONE);
```

Index on Employee Email

sql

Copy code

```
CREATE INDEX idx_employee_email ON Employee(EMPLOYEE_EMAIL);
```

Index on Rental Transaction Dates

```
CREATE INDEX idx_rental_dates ON Rental_transaction(RENTAL_START_DATE,
RENTAL_END_DATE);
```

Index on Vehicle Type

```
CREATE INDEX idx_vehicle_type ON Vehicle(VEHICLE_TYPE);
```

Sample Data Insertion

Inserting into Address Table

```
INSERT INTO Address (ADDRESS_ID, STREET_NAME, CITY, PROVINCE,
POSTAL_CODE) VALUES (1, 'Main St', 'Springfield', 'IL', '62704');

INSERT INTO Address (ADDRESS_ID, STREET_NAME, CITY, PROVINCE,
POSTAL_CODE) VALUES (2, '2nd St', 'Springfield', 'IL', '62704');
```

Inserting into Branch Table

```
INSERT INTO Branch (BRANCH_ID, ADDRESS_ID, BRANCH_PHONE, BRANCH_EMAIL,
MANAGER_ID, BRANCH_NAME) VALUES (1, 1, '2175551234', 'branch1@xyz.com', 1,
'Springfield North');

INSERT INTO Branch (BRANCH_ID, ADDRESS_ID, BRANCH_PHONE, BRANCH_EMAIL,
MANAGER_ID, BRANCH_NAME) VALUES (2, 2, '2175555678', 'branch2@xyz.com', 2,
'Springfield South');
```

Inserting into Employee Table

```
INSERT INTO Employee (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_SURNAME,
EMPLOYEE_PHONE, EMPLOYEE_EMAIL, EMPLOYEE_POSITION, BRANCH_ID)
VALUES (1, 'John', 'Doe', '2175551001', 'johndoe@xyz.com', 'Manager', 1);

INSERT INTO Employee (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_SURNAME,
EMPLOYEE_PHONE, EMPLOYEE_EMAIL, EMPLOYEE_POSITION, BRANCH_ID)
VALUES (2, 'Jane', 'Smith', '2175551002', 'janesmith@xyz.com', 'Manager', 2);
```

Inserting into Customer Table

```
INSERT INTO Customer (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_SURNAME,
CUSTOMER_PHONE, CUSTOMER_EMAIL, ADDRESS_ID,
CUSTOMER_DATE_OF_BIRTH, CUSTOMER_AGE, AGENT_ID) VALUES (1, 'Alice',
'Johnson', '2175552001', 'alicej@xyz.com', 1, TO_DATE('1990-01-01', 'YYYY-MM-DD'), 34,
1);

INSERT INTO Customer (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_SURNAME,
CUSTOMER_PHONE, CUSTOMER_EMAIL, ADDRESS_ID,
CUSTOMER_DATE_OF_BIRTH, CUSTOMER_AGE, AGENT_ID) VALUES (2, 'Bob',
'Williams', '2175552002', 'bobw@xyz.com', 2, TO_DATE('1985-05-15', 'YYYY-MM-DD'), 39, 2);
```

Inserting into Vehicle Table

```
INSERT INTO Vehicle (VEHICLE_ID, VEHICLE_COLOUR, VEHICLE_TYPE, AVAILABILITY_ID, VEHICLE_MILEAGE, VEHICLE_RENTAL_PRICE, MAINTENANCE_ID, VEHICLE_MAKE) VALUES (1, 'Red', 'Sedan', 1, 15000, 50.00, 1, 'Toyota');

INSERT INTO Vehicle (VEHICLE_ID, VEHICLE_COLOUR, VEHICLE_TYPE, AVAILABILITY_ID, VEHICLE_MILEAGE, VEHICLE_RENTAL_PRICE, MAINTENANCE_ID, VEHICLE_MAKE) VALUES (2, 'Blue', 'SUV', 2, 20000, 70.00, 2, 'Honda');
```

Inserting into Vehicle Availability Table

```
INSERT INTO Vehicle_availability (AVAILABILITY_ID, VEHICLE_ID, VEHICLE_CONDITION, VEHICLE_AVAILABILITY) VALUES (1, 1, 'Good', 'Available');

INSERT INTO Vehicle_availability (AVAILABILITY_ID, VEHICLE_ID, VEHICLE_CONDITION, VEHICLE_AVAILABILITY) VALUES (2, 2, 'Good', 'Available');
```

Inserting into Vehicle Maintenance Table

```
INSERT INTO Vehicle_maintenance (MAINTENANCE_ID, VEHICLE_ID, MAINTENANCE_TYPE, MECHANIC_ID, MAINTENANCE_DATE) VALUES (1, 1, 'Oil Change', 1, TO_DATE('2023-01-01', 'YYYY-MM-DD'));

INSERT INTO Vehicle_maintenance (MAINTENANCE_ID, VEHICLE_ID, MAINTENANCE_TYPE, MECHANIC_ID, MAINTENANCE_DATE) VALUES (2, 2, 'Brake Inspection', 2, TO_DATE('2023-02-01', 'YYYY-MM-DD'));
```

Inserting into Rental Transaction Table

```
INSERT INTO Rental_transaction (RENTAL_ID, CUSTOMER_ID, VEHICLE_ID, AGENT_ID, RENTAL_START_DATE, RENTAL_END_DATE, RENTAL_COST, RENTAL_STATUS) VALUES (1, 1, 1, 1, TO_DATE('2023-03-01', 'YYYY-MM-DD'), TO_DATE('2023-03-10', 'YYYY-MM-DD'), 500.00, 'Returned');
INSERT INTO Rental_transaction (RENTAL_ID, CUSTOMER_ID, VEHICLE_ID, AGENT_ID,
RENTAL_START_DATE, RENTAL_END_DATE, RENTAL_COST, RENTAL_STATUS)
VALUES (2, 2, 2, 2, TO_DATE('2023-04-01', 'YYYY-MM-DD'), TO_DATE('2023-04-05',
'YYYY-MM-DD'), 350.00, 'Returned');
```

Inserting into Banking Information Table

```
INSERT INTO Banking_information (CUSTOMER_ID, BANK_NAME, CARD_NUMBER,
ACCOUNT_NUMBER, ACCOUNT_TYPE) VALUES (1, 'Bank of America',
'1234567812345678', '100200300', 'Checking');
```

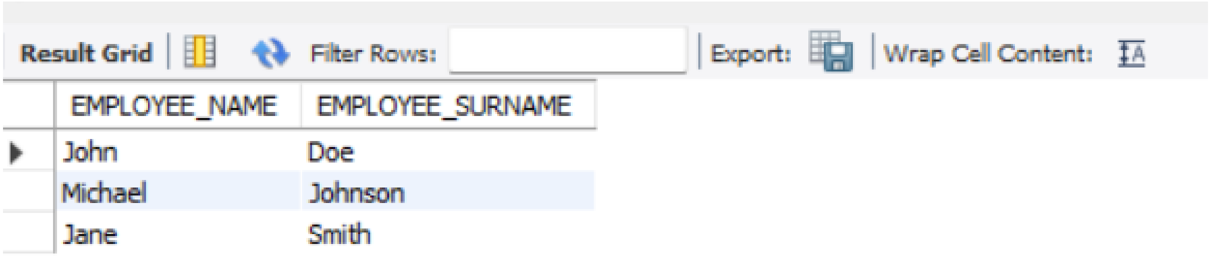
```
INSERT INTO Banking_information (CUSTOMER_ID, BANK_NAME, CARD_NUMBER,
ACCOUNT_NUMBER, ACCOUNT_TYPE) VALUES (2, 'Chase', '8765432187654321',
'400500600', 'Savings');
```

Inserting into Card Table

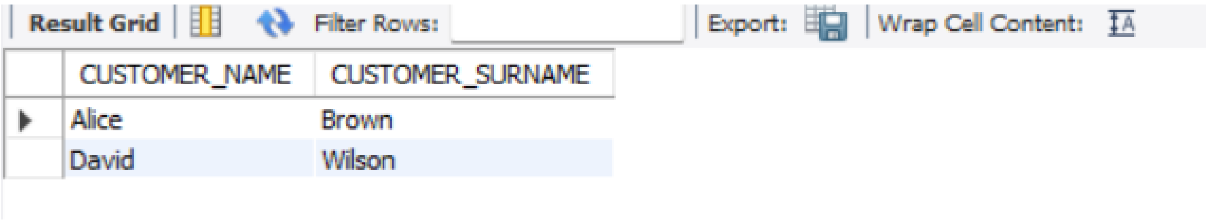
```
INSERT INTO Card (CARD_NUMBER, EXPIRATION_DATE) VALUES
('1234567812345678', TO_DATE('2025-12-31', 'YYYY-MM-DD'));
```

```
INSERT INTO Card (CARD_NUMBER, EXPIRATION_DATE) VALUES
('8765432187654321', TO_DATE('2026-06-30', 'YYYY-MM-DD'));
```

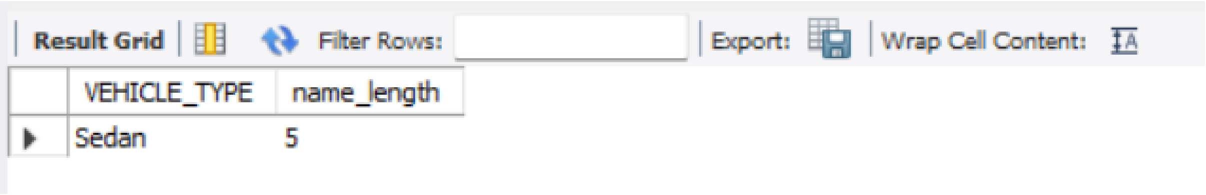
THE PICTURES OF THE DATABASE TABLE



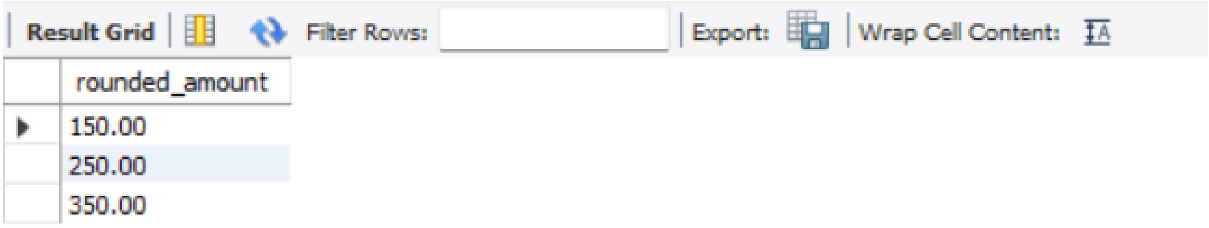
EMPLOYEE_NAME	EMPLOYEE_SURNAME
John	Doe
Michael	Johnson
Jane	Smith



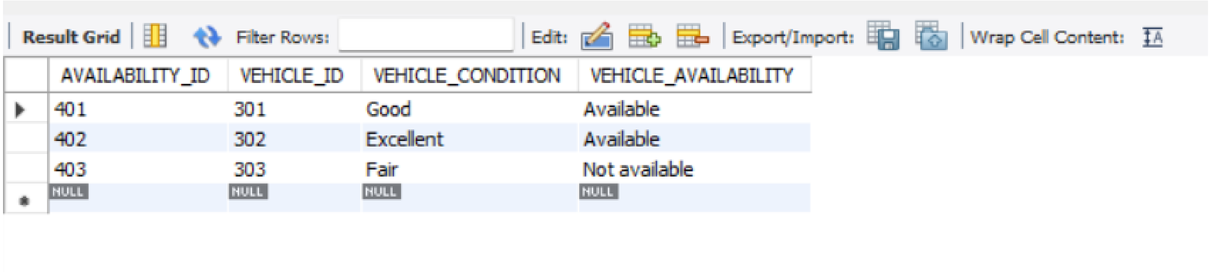
CUSTOMER_NAME	CUSTOMER_SURNAME
Alice	Brown
David	Wilson



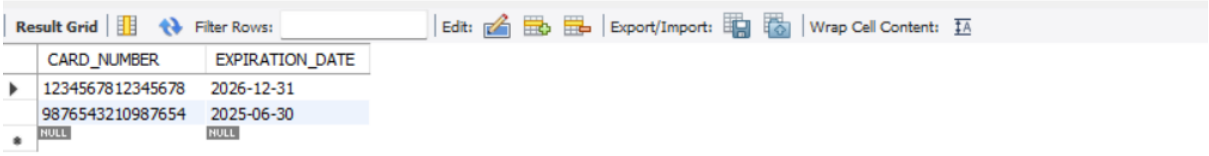
VEHICLE_TYPE	name_length
Sedan	5



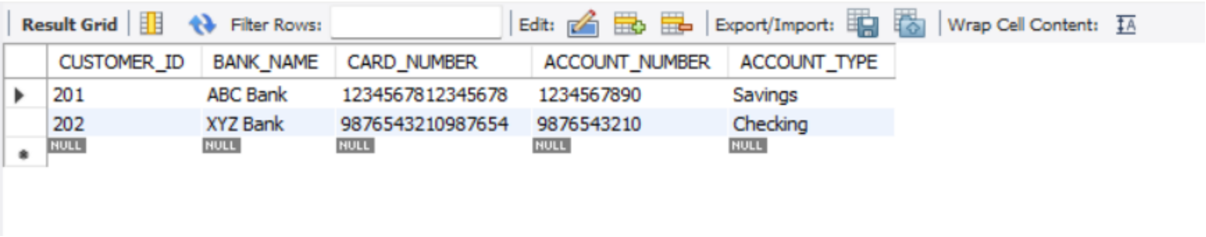
rounded_amount
150.00
250.00
350.00



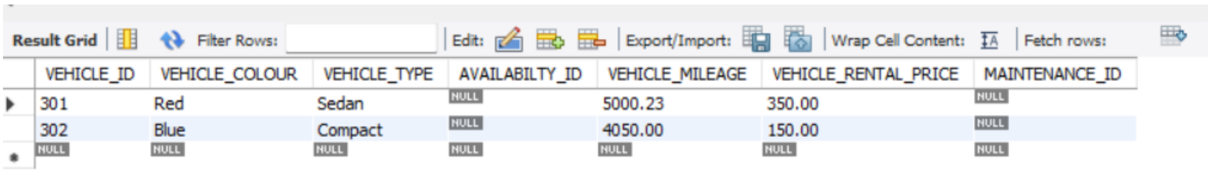
AVAILABILITY_ID	VEHICLE_ID	VEHICLE_CONDITION	VEHICLE_AVAILABILITY
401	301	Good	Available
402	302	Excellent	Available
403	303	Fair	Not available
NULL	NULL	NULL	NULL



CARD_NUMBER	EXPIRATION_DATE
1234567812345678	2026-12-31
9876543210987654	2025-06-30
NULL	NULL



CUSTOMER_ID	BANK_NAME	CARD_NUMBER	ACCOUNT_NUMBER	ACCOUNT_TYPE
201	ABC Bank	1234567812345678	1234567890	Savings
202	XYZ Bank	9876543210987654	9876543210	Checking
NULL	NULL	NULL	NULL	NULL



VEHICLE_ID	VEHICLE_COLOUR	VEHICLE_TYPE	AVAILABILTY_ID	VEHICLE_MILEAGE	VEHICLE_RENTAL_PRICE	MAINTENANCE_ID
301	Red	Sedan	NULL	5000.23	350.00	NULL
302	Blue	Compact	NULL	4050.00	150.00	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Group Member contribution

Student Number	Phase 3 Contribution
37320629	100%
42618282	100%
37574175	100%
40837335	100%
35111550	100%
34812938	100%
28844467	100%