CTU training solutions

0861 100 395 | www.ctutraining.co.za | enquiry@ctutraining.co.za

Gabriella Rakgotsoka

20232605

PRG522_Formative 3 **21/05/2024**

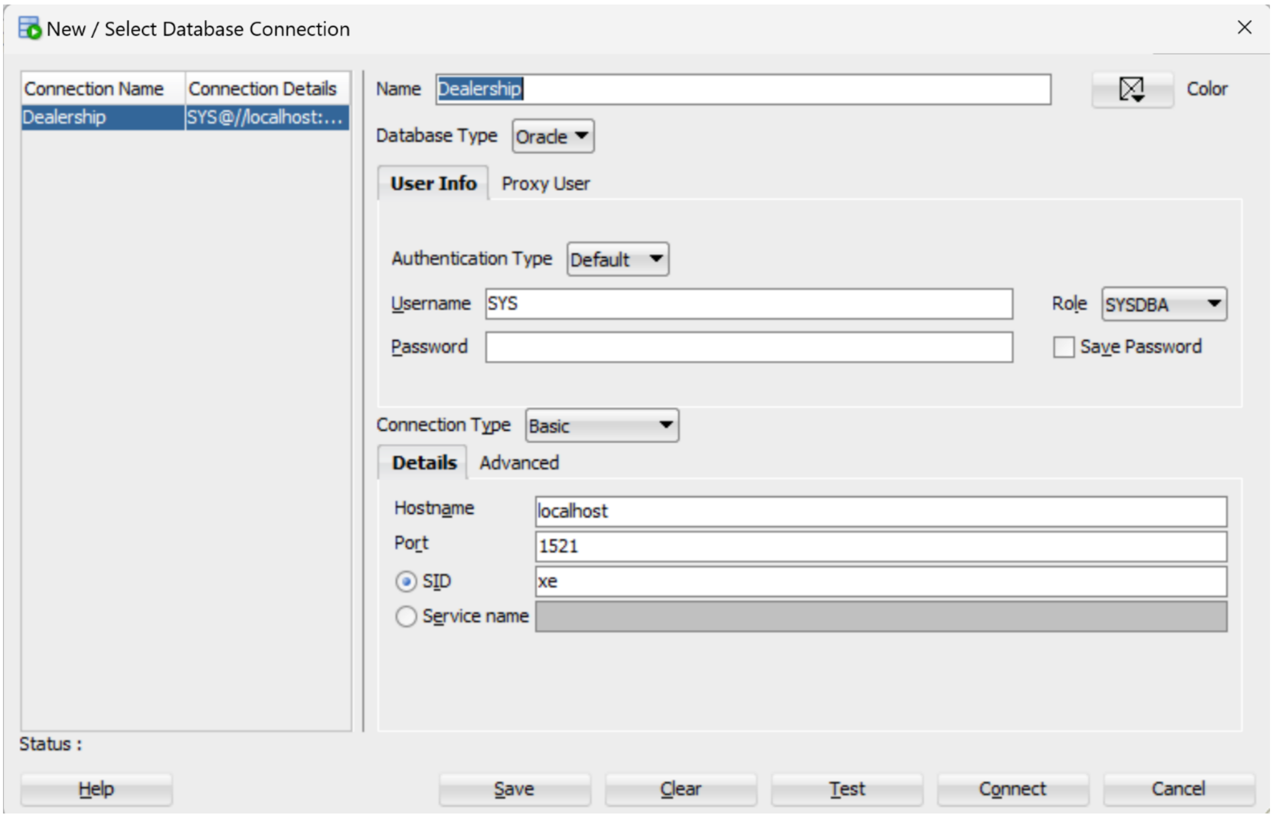
Table of Contents

Scenario Question(s)

Question 1

This task includes preparing the data

1.1 On Oracle, write SQL statements to create a database for Nicky Motors called “dealership” (2 Marks)



New / Select Database Connection

Connection Name	Connection Details
Dealership	SYS@//localhost:...

Name: Dealership

Database Type: Oracle

User Info

Authentication Type: Default

Username: SYS

Password:

Role: SYSDBA

□ Save Password

Connection Type: Basic

Details

Hostname: localhost

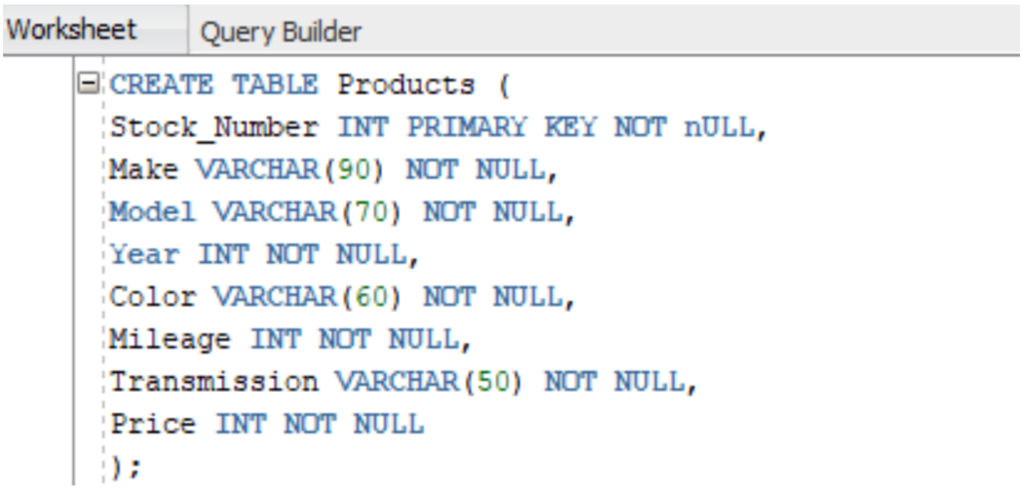
Port: 1521

● SID: xe

○ Service name

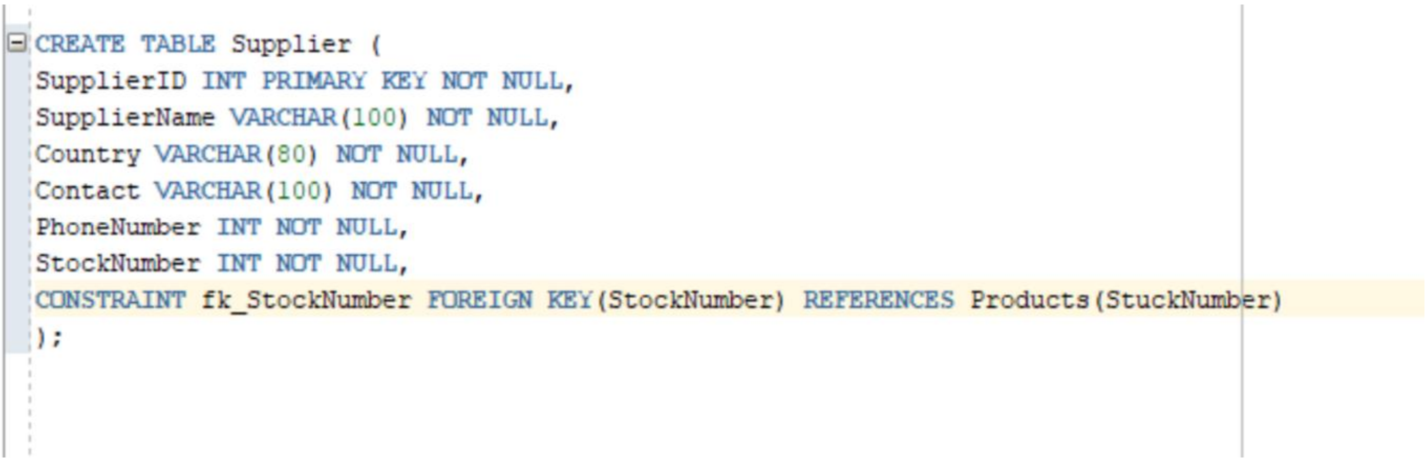
Status :

1.2 Write SQL statements that will recreate the product table for Nicky Motors, this table should include an additional column called “Supplier ID” which has a foreign key and the data can not be null. (9 Marks)



```
CREATE TABLE Products (
Stock_Number INT PRIMARY KEY NOT nULL,
Make VARCHAR(90) NOT NULL,
Model VARCHAR(70) NOT NULL,
Year INT NOT NULL,
Color VARCHAR(60) NOT NULL,
Mileage INT NOT NULL,
Transmission VARCHAR(50) NOT NULL,
Price INT NOT NULL
);
```

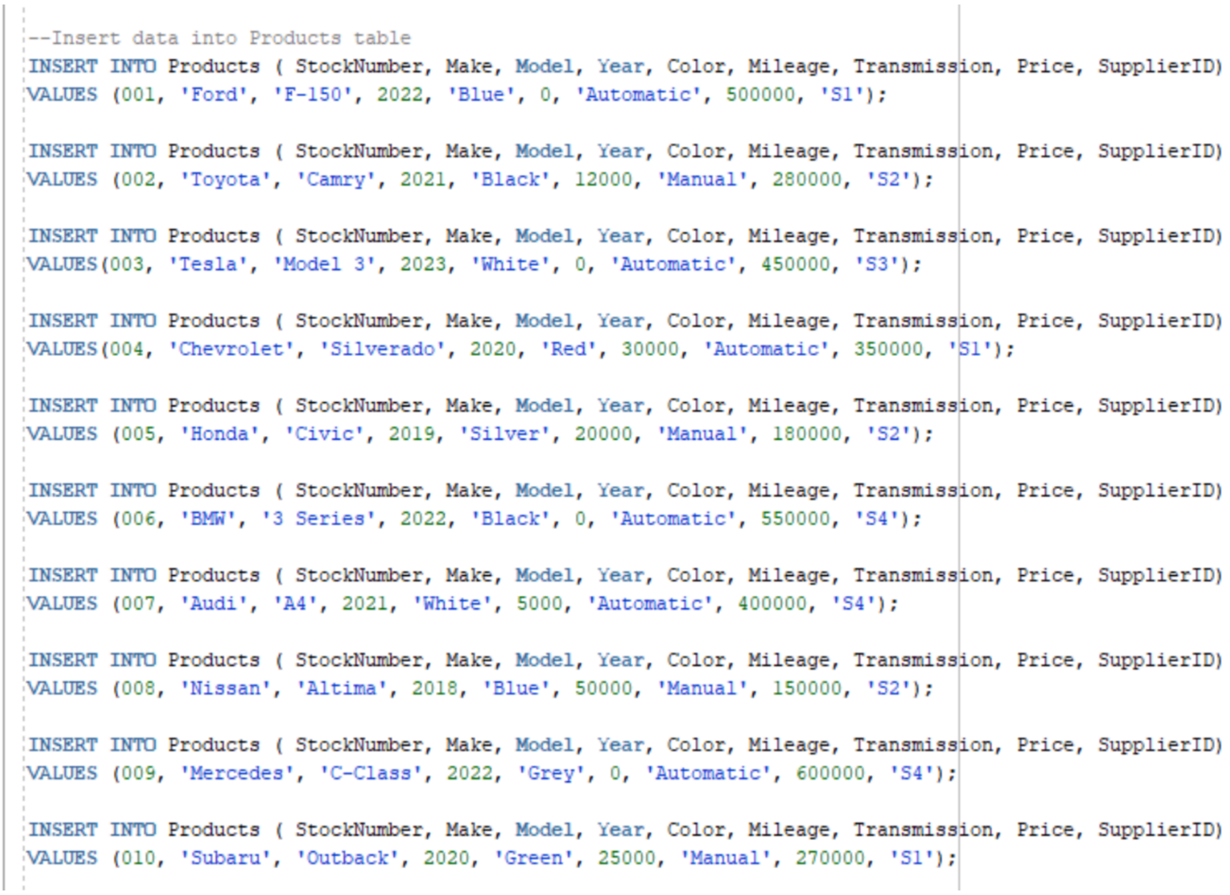
1.3 Write SQL statements that will recreate the supplier table for Nicky Motors on Oracle. (6 Marks)



```
CREATE TABLE Supplier (
SupplierID INT PRIMARY KEY NOT NULL,
SupplierName VARCHAR(100) NOT NULL,
Country VARCHAR(80) NOT NULL,
Contact VARCHAR(100) NOT NULL,
PhoneNumber INT NOT NULL,
StockNumber INT NOT NULL,
CONSTRAINT fk_StockNumber FOREIGN KEY(StockNumber) REFERENCES Products(StuckNumber)
);
```

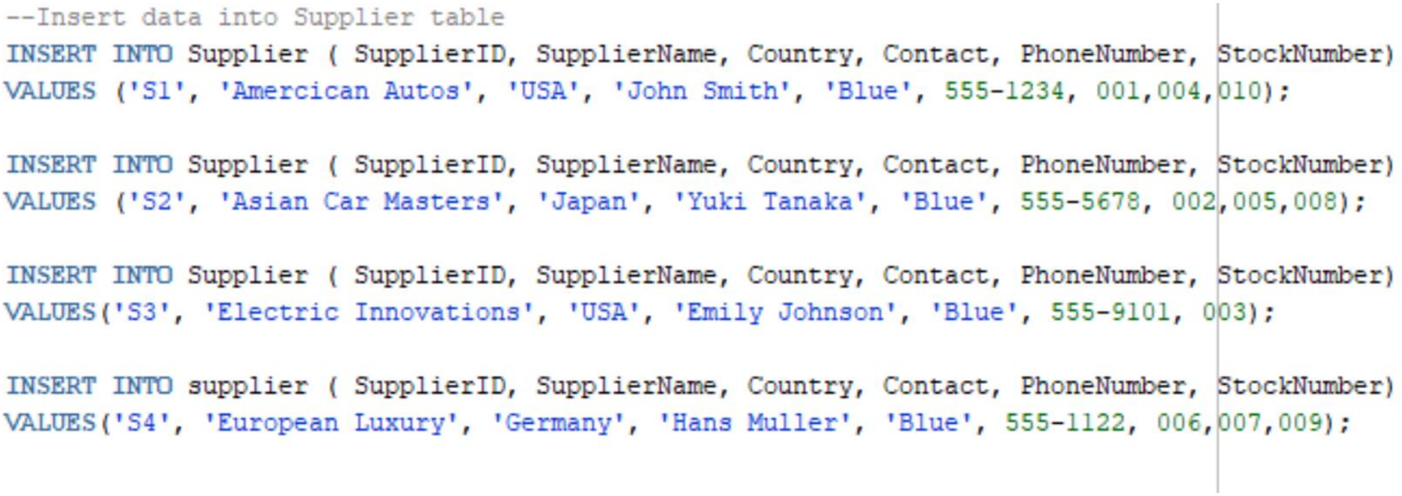
1.4 Write SQL statements that will flood both tables with the data as shown above. (8 Marks)

Inserting data into the Products table:



```
--Insert data into Products table
INSERT INTO Products ( StockNumber, Make, Model, Year, Color, Mileage, Transmission, Price, SupplierID)
VALUES (001, 'Ford', 'F-150', 2022, 'Blue', 0, 'Automatic', 500000, 'S1');

INSERT INTO Products ( StockNumber, Make, Model, Year, Color, Mileage, Transmission, Price, SupplierID)
VALUES (002, 'Toyota', 'Camry', 2021, 'Black', 12000, 'Manual', 280000, 'S2');

INSERT INTO Products ( StockNumber, Make, Model, Year, Color, Mileage, Transmission, Price, SupplierID)
VALUES(003, 'Tesla', 'Model 3', 2023, 'White', 0, 'Automatic', 450000, 'S3');

INSERT INTO Products ( StockNumber, Make, Model, Year, Color, Mileage, Transmission, Price, SupplierID)
VALUES(004, 'Chevrolet', 'Silverado', 2020, 'Red', 30000, 'Automatic', 350000, 'S1');

INSERT INTO Products ( StockNumber, Make, Model, Year, Color, Mileage, Transmission, Price, SupplierID)
VALUES (005, 'Honda', 'Civic', 2019, 'Silver', 20000, 'Manual', 180000, 'S2');

INSERT INTO Products ( StockNumber, Make, Model, Year, Color, Mileage, Transmission, Price, SupplierID)
VALUES (006, 'BMW', '3 Series', 2022, 'Black', 0, 'Automatic', 550000, 'S4');

INSERT INTO Products ( StockNumber, Make, Model, Year, Color, Mileage, Transmission, Price, SupplierID)
VALUES (007, 'Audi', 'A4', 2021, 'White', 5000, 'Automatic', 400000, 'S4');

INSERT INTO Products ( StockNumber, Make, Model, Year, Color, Mileage, Transmission, Price, SupplierID)
VALUES (008, 'Nissan', 'Altima', 2018, 'Blue', 50000, 'Manual', 150000, 'S2');

INSERT INTO Products ( StockNumber, Make, Model, Year, Color, Mileage, Transmission, Price, SupplierID)
VALUES (009, 'Mercedes', 'C-Class', 2022, 'Grey', 0, 'Automatic', 600000, 'S4');

INSERT INTO Products ( StockNumber, Make, Model, Year, Color, Mileage, Transmission, Price, SupplierID)
VALUES (010, 'Subaru', 'Outback', 2020, 'Green', 25000, 'Manual', 270000, 'S1');
```

Inserting data into the Supplier table:



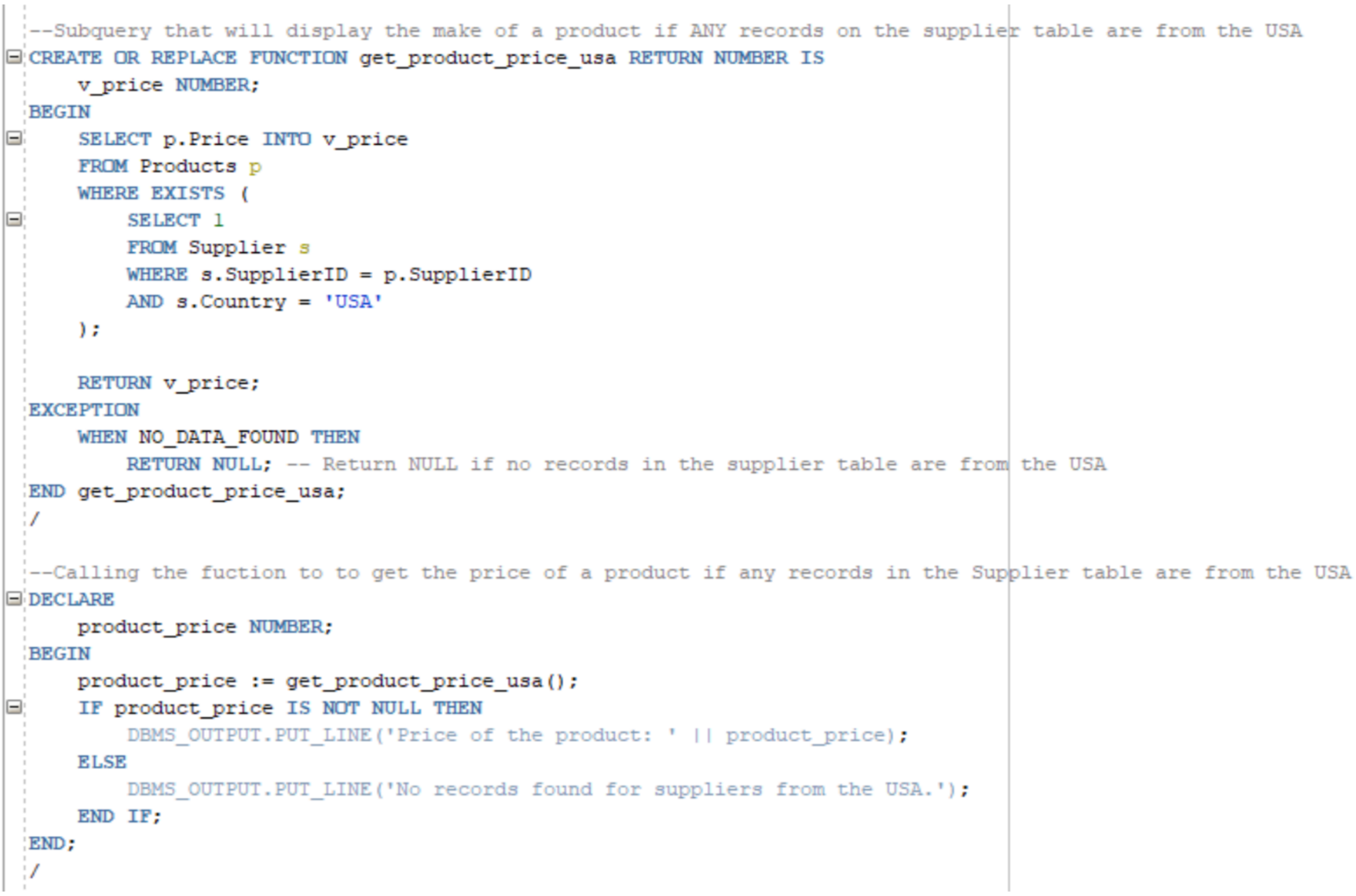
```
--Insert data into Supplier table
INSERT INTO Supplier ( SupplierID, SupplierName, Country, Contact, PhoneNumber, StockNumber)
VALUES ('S1', 'Amercican Autos', 'USA', 'John Smith', 'Blue', 555-1234, 001,004,010);

INSERT INTO Supplier ( SupplierID, SupplierName, Country, Contact, PhoneNumber, StockNumber)
VALUES ('S2', 'Asian Car Masters', 'Japan', 'Yuki Tanaka', 'Blue', 555-5678, 002,005,008);

INSERT INTO Supplier ( SupplierID, SupplierName, Country, Contact, PhoneNumber, StockNumber)
VALUES('S3', 'Electric Innovations', 'USA', 'Emily Johnson', 'Blue', 555-9101, 003);

INSERT INTO supplier ( SupplierID, SupplierName, Country, Contact, PhoneNumber, StockNumber)
VALUES('S4', 'European Luxury', 'Germany', 'Hans Muller', 'Blue', 555-1122, 006,007,009);
```

[Total = 25 Marks]

Question 2

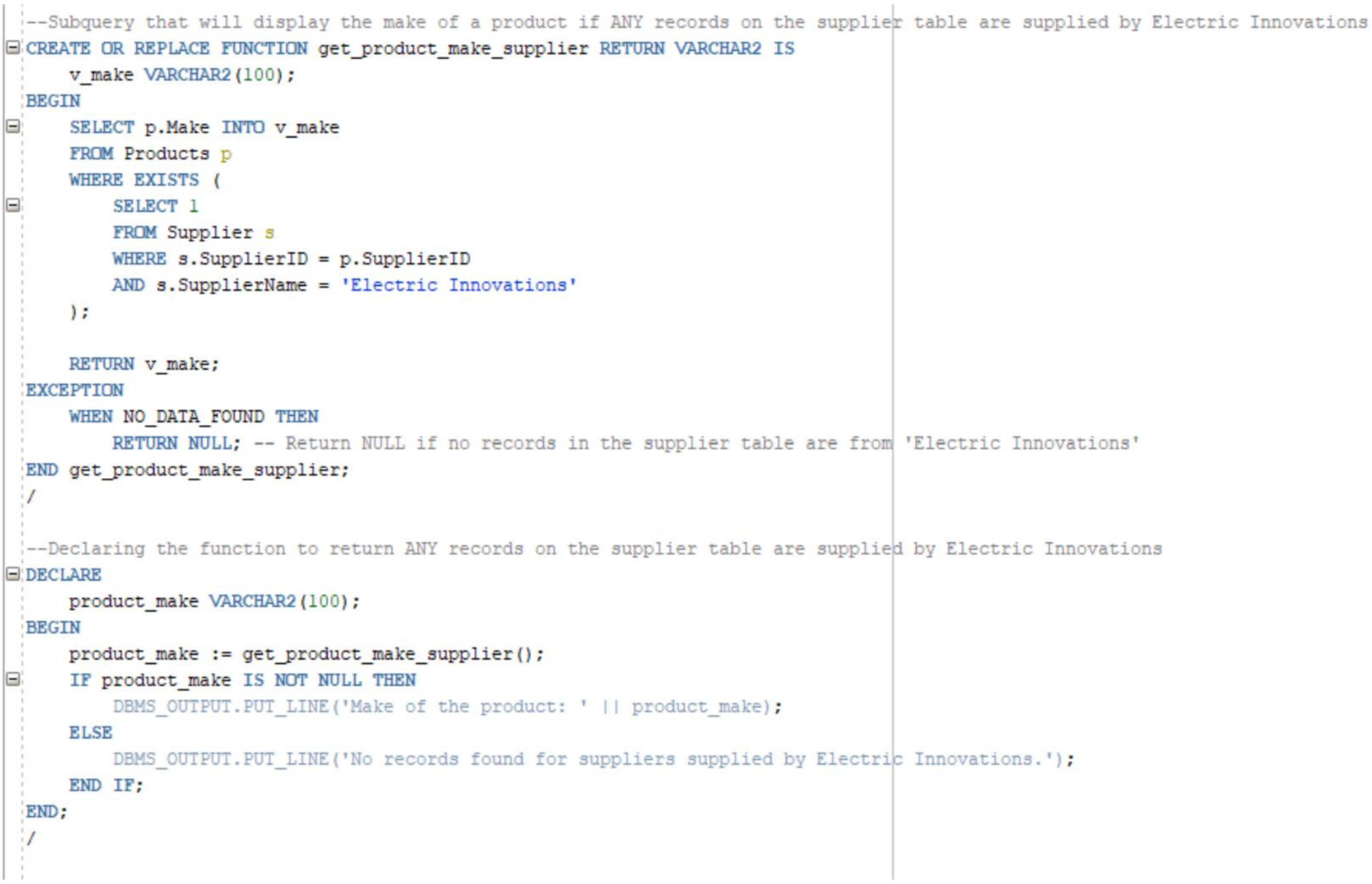
This task includes learning unit 9: Using subqueries to solve queries. Write SQL statements to perform the following subqueries:

2.1 Display the price of a product if ANY records on the supplier table are from the USA   (7 Marks)



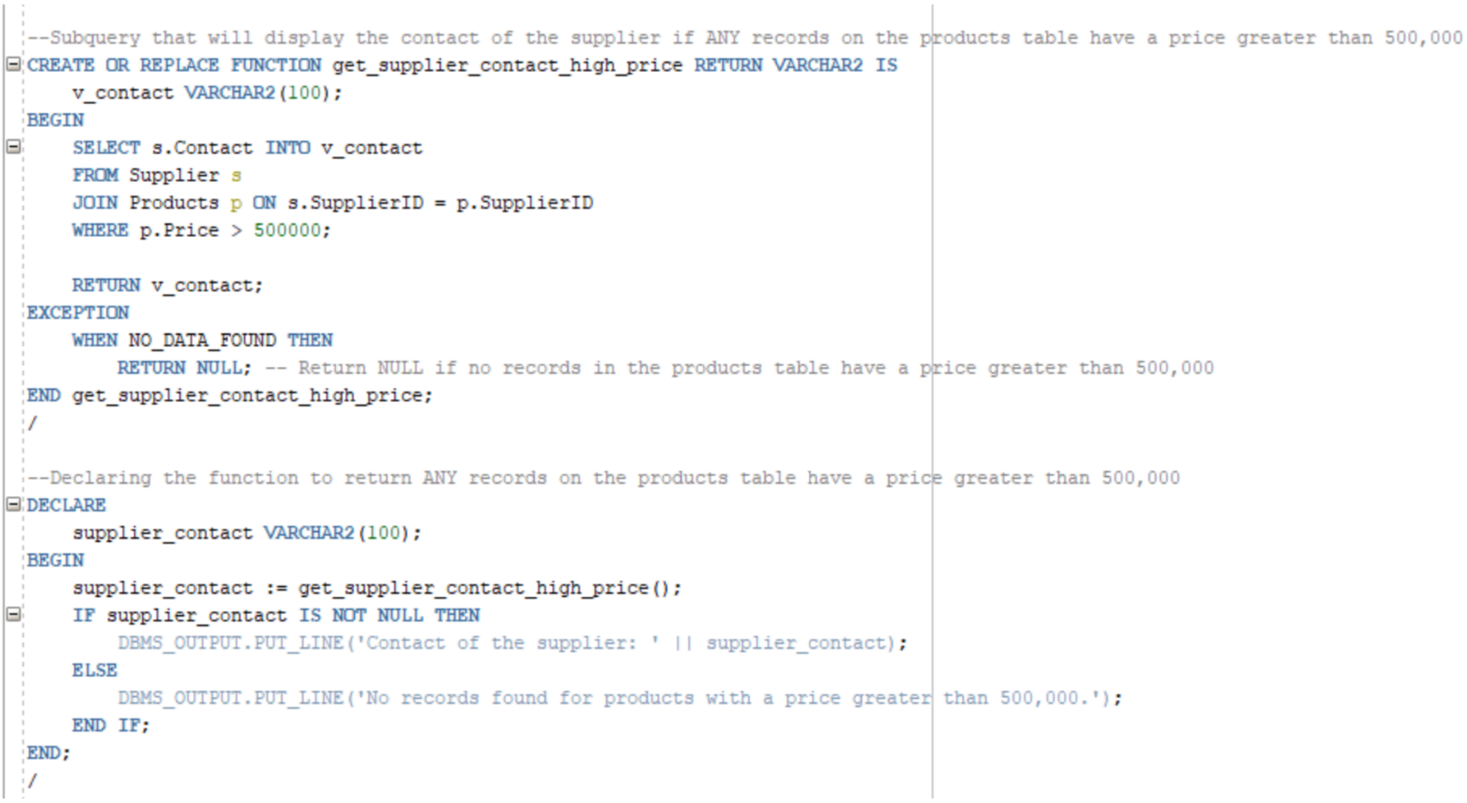
```
--Subquery that will display the make of a product if ANY records on the supplier table are from the USA
CREATE OR REPLACE FUNCTION get_product_price_usa RETURN NUMBER IS
    v_price NUMBER;
BEGIN
    SELECT p.Price INTO v_price
    FROM Products p
    WHERE EXISTS (
        SELECT 1
        FROM Supplier s
        WHERE s.SupplierID = p.SupplierID
        AND s.Country = 'USA'
    );

    RETURN v_price;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL; -- Return NULL if no records in the supplier table are from the USA
END get_product_price_usa;
/

--Calling the fuction to to get the price of a product if any records in the Supplier table are from the USA
DECLARE
    product_price NUMBER;
BEGIN
    product_price := get_product_price_usa();
    IF product_price IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Price of the product: ' || product_price);
    ELSE
        DBMS_OUTPUT.PUT_LINE('No records found for suppliers from the USA.');
    END IF;
END;
/
```

2.2 Display the make of a product if ANY records on the supplier table are supplied by Electric Innovations (7 Marks)



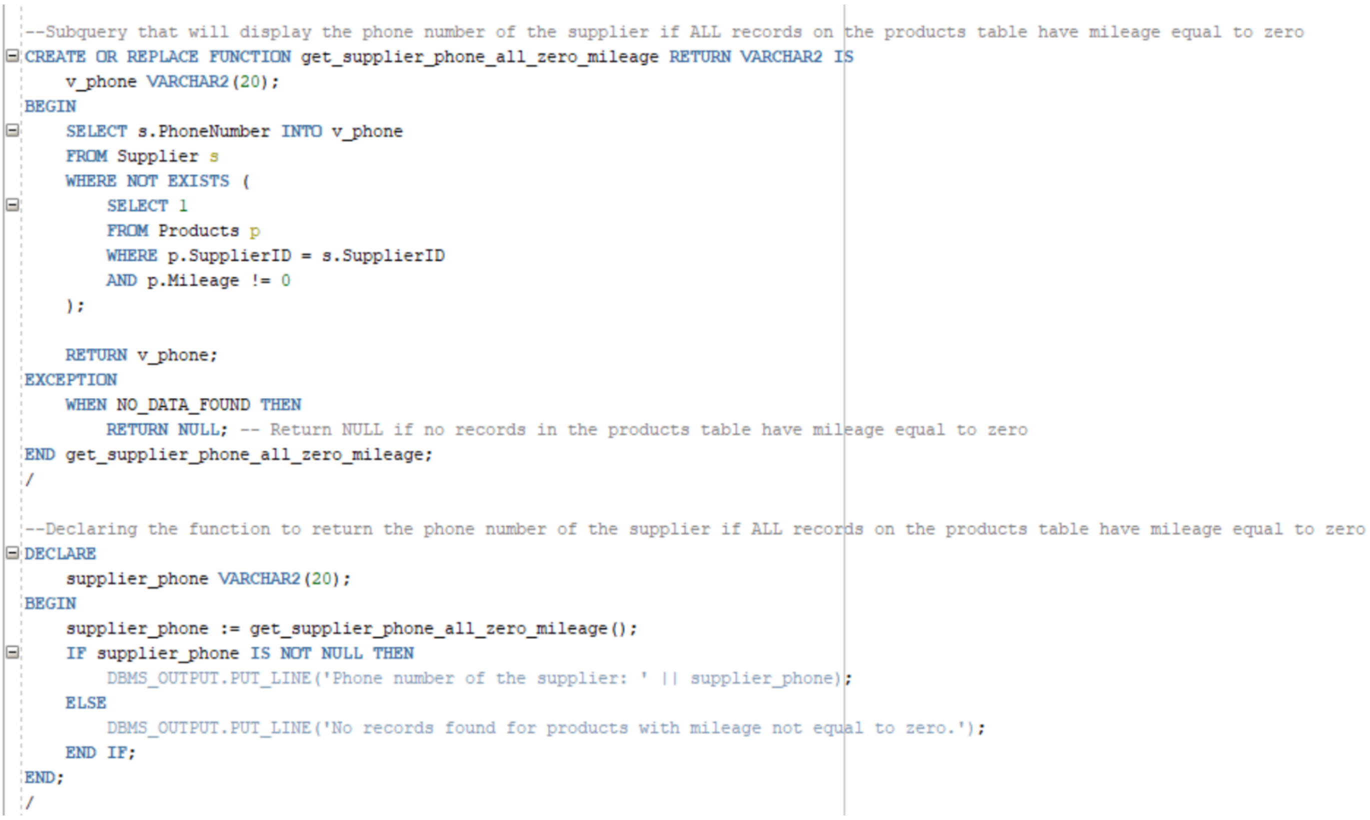
```
--Subquery that will display the make of a product if ANY records on the supplier table are supplied by Electric Innovations
CREATE OR REPLACE FUNCTION get_product_make_supplier RETURN VARCHAR2 IS
    v_make VARCHAR2(100);
BEGIN
    SELECT p.Make INTO v_make
    FROM Products p
    WHERE EXISTS (
        SELECT 1
        FROM Supplier s
        WHERE s.SupplierID = p.SupplierID
        AND s.SupplierName = 'Electric Innovations'
    );

    RETURN v_make;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL; -- Return NULL if no records in the supplier table are from 'Electric Innovations'
END get_product_make_supplier;
/

--Declaring the function to return ANY records on the supplier table are supplied by Electric Innovations
DECLARE
    product_make VARCHAR2(100);
BEGIN
    product_make := get_product_make_supplier();
    IF product_make IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Make of the product: ' || product_make);
    ELSE
        DBMS_OUTPUT.PUT_LINE('No records found for suppliers supplied by Electric Innovations.');
    END IF;
END;
/
```

2.3 Display the contact of the supplier if ANY records on the products table have a price greater than 500,000 (7 Marks)



```
--Subquery that will display the contact of the supplier if ANY records on the products table have a price greater than 500,000
CREATE OR REPLACE FUNCTION get_supplier_contact_high_price RETURN VARCHAR2 IS
    v_contact VARCHAR2(100);
BEGIN
    SELECT s.Contact INTO v_contact
    FROM Supplier s
    JOIN Products p ON s.SupplierID = p.SupplierID
    WHERE p.Price > 500000;

    RETURN v_contact;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL; -- Return NULL if no records in the products table have a price greater than 500,000
END get_supplier_contact_high_price;
/

--Declaring the function to return ANY records on the products table have a price greater than 500,000
DECLARE
    supplier_contact VARCHAR2(100);
BEGIN
    supplier_contact := get_supplier_contact_high_price();
    IF supplier_contact IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Contact of the supplier: ' || supplier_contact);
    ELSE
        DBMS_OUTPUT.PUT_LINE('No records found for products with a price greater than 500,000.');
    END IF;
END;
/
```

2.4 Display the phone number of the supplier if ALL records on the products table have mileage equal to zero (4 Marks)



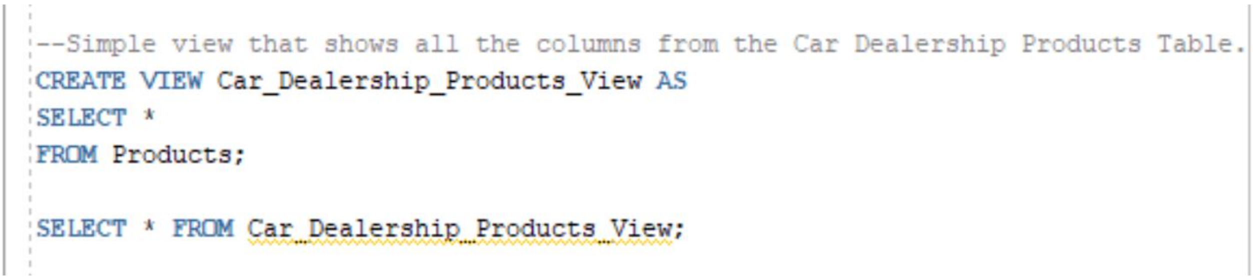
```
--Subquery that will display the phone number of the supplier if ALL records on the products table have mileage equal to zero
CREATE OR REPLACE FUNCTION get_supplier_phone_all_zero_mileage RETURN VARCHAR2 IS
    v_phone VARCHAR2(20);
BEGIN
    SELECT s.PhoneNumber INTO v_phone
    FROM Supplier s
    WHERE NOT EXISTS (
        SELECT 1
        FROM Products p
        WHERE p.SupplierID = s.SupplierID
        AND p.Mileage != 0
    );

    RETURN v_phone;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL; -- Return NULL if no records in the products table have mileage equal to zero
END get_supplier_phone_all_zero_mileage;
/

--Declaring the function to return the phone number of the supplier if ALL records on the products table have mileage equal to zero
DECLARE
    supplier_phone VARCHAR2(20);
BEGIN
    supplier_phone := get_supplier_phone_all_zero_mileage();
    IF supplier_phone IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Phone number of the supplier: ' || supplier_phone);
    ELSE
        DBMS_OUTPUT.PUT_LINE('No records found for products with mileage not equal to zero.');
    END IF;
END;
/
```

[Total = 25 Marks]

Question 3

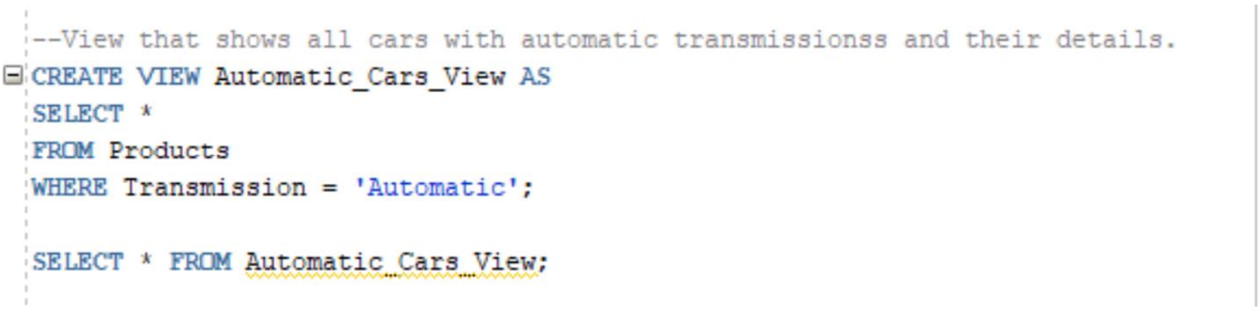
This task includes learning unit 10: Managing Schema Objects. Write SQL statements to perform the following Views:

3.1 Create a simple view that shows all the columns from the Car Dealership Products Table. (3 Marks)



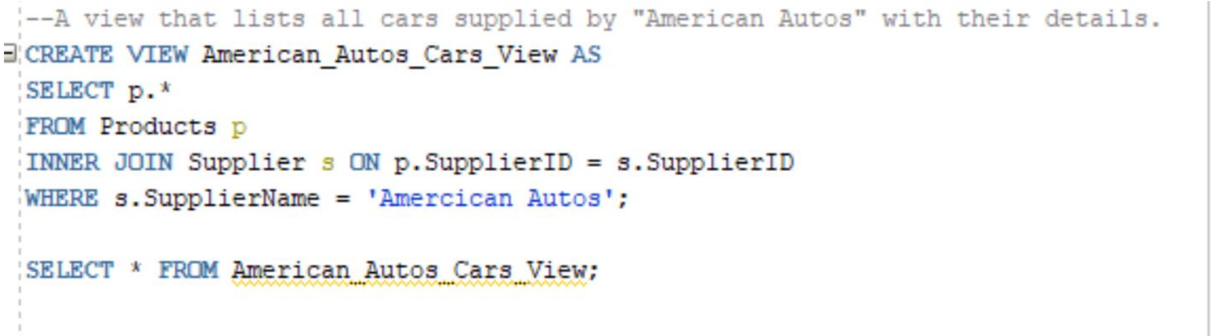
```
--Simple view that shows all the columns from the Car Dealership Products Table.
CREATE VIEW Car_Dealership_Products_View AS
SELECT *
FROM Products;

SELECT * FROM Car_Dealership_Products_View;
```

3.2 Create a view that shows all cars with automatic transmission and their details. (4 Marks)



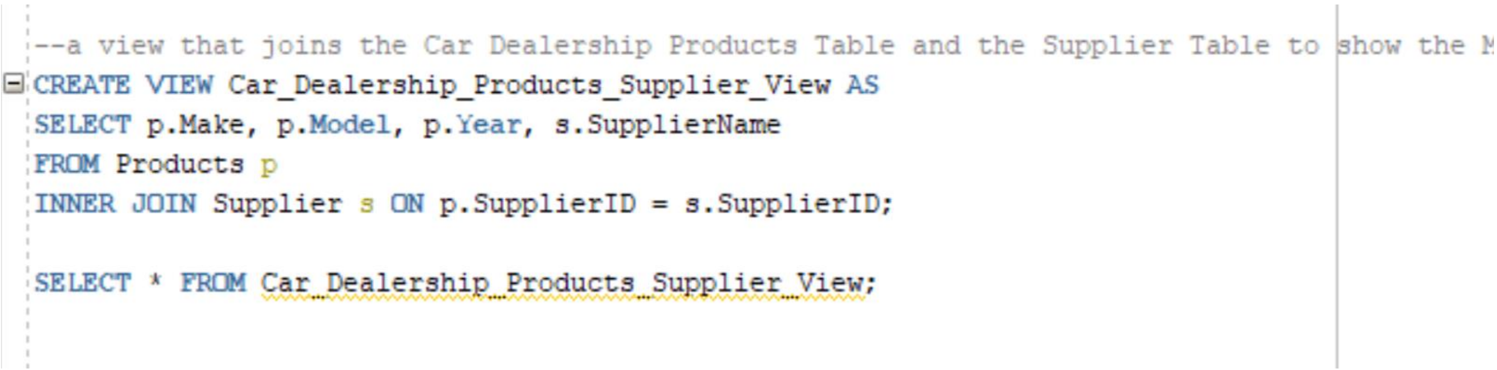
```
--View that shows all cars with automatic transmissionss and their details.
CREATE VIEW Automatic_Cars_View AS
SELECT *
FROM Products
WHERE Transmission = 'Automatic';

SELECT * FROM Automatic_Cars_View;
```

3.3 Create a view that lists all cars supplied by "American Autos" with their details. (6 Marks)



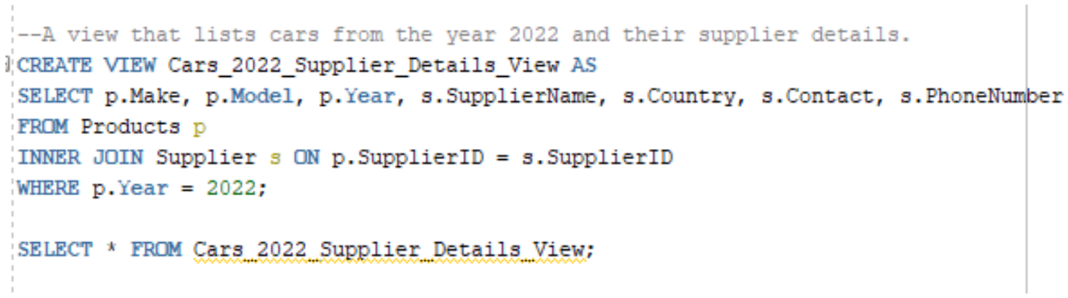
```
--A view that lists all cars supplied by "American Autos" with their details.
CREATE VIEW American_Autos_Cars_View AS
SELECT p.*
FROM Products p
INNER JOIN Supplier s ON p.SupplierID = s.SupplierID
WHERE s.SupplierName = 'Amercican Autos';

SELECT * FROM American_Autos_Cars_View;
```

3.4 Create a complex view that joins the Car Dealership Products Table and the Supplier Table to show the Make, Model, Year, and Supplier Name. (6 Marks)



```
--a view that joins the Car Dealership Products Table and the Supplier Table to show the M
CREATE VIEW Car_Dealership_Products_Supplier_View AS
SELECT p.Make, p.Model, p.Year, s.SupplierName
FROM Products p
INNER JOIN Supplier s ON p.SupplierID = s.SupplierID;

SELECT * FROM Car_Dealership_Products_Supplier_View;
```

3.5 Create a view that lists cars from the year 2022 and their supplier details. (6 Marks)



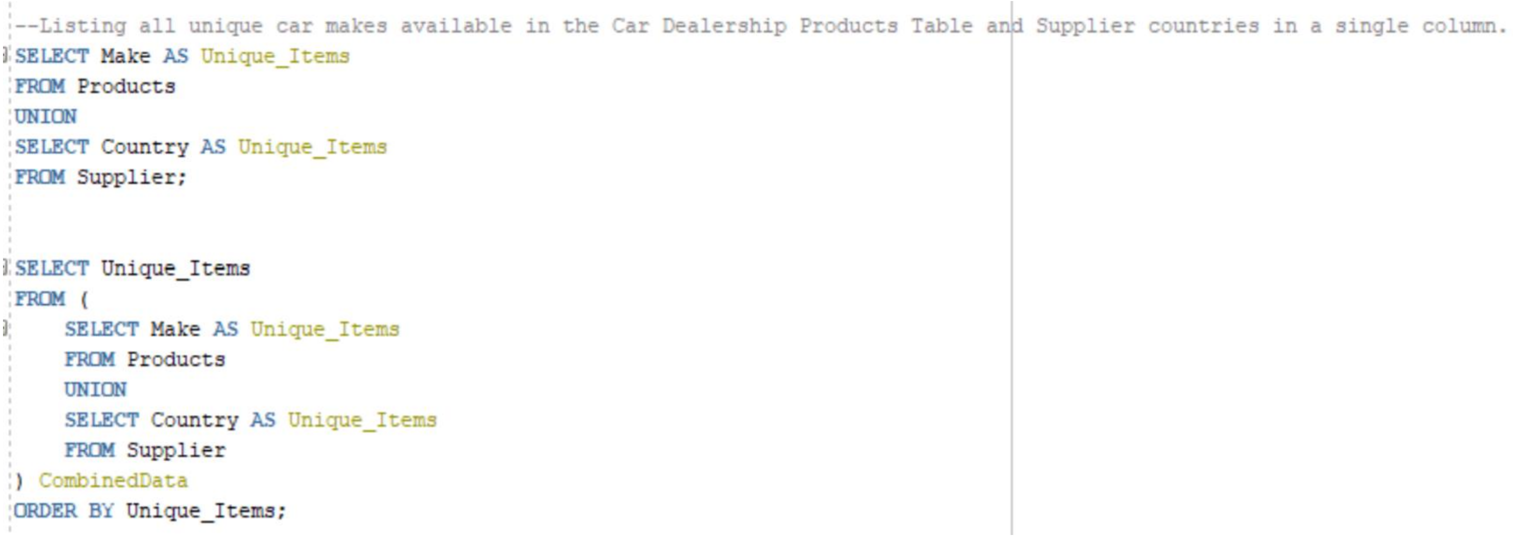
```
--A view that lists cars from the year 2022 and their supplier details.
CREATE VIEW Cars_2022_Supplier_Details_View AS
SELECT p.Make, p.Model, p.Year, s.SupplierName, s.Country, s.Contact, s.PhoneNumber
FROM Products p
INNER JOIN Supplier s ON p.SupplierID = s.SupplierID
WHERE p.Year = 2022;

SELECT * FROM Cars_2022_Supplier_Details_View;
```

[Total = 25 Marks]

Question 4

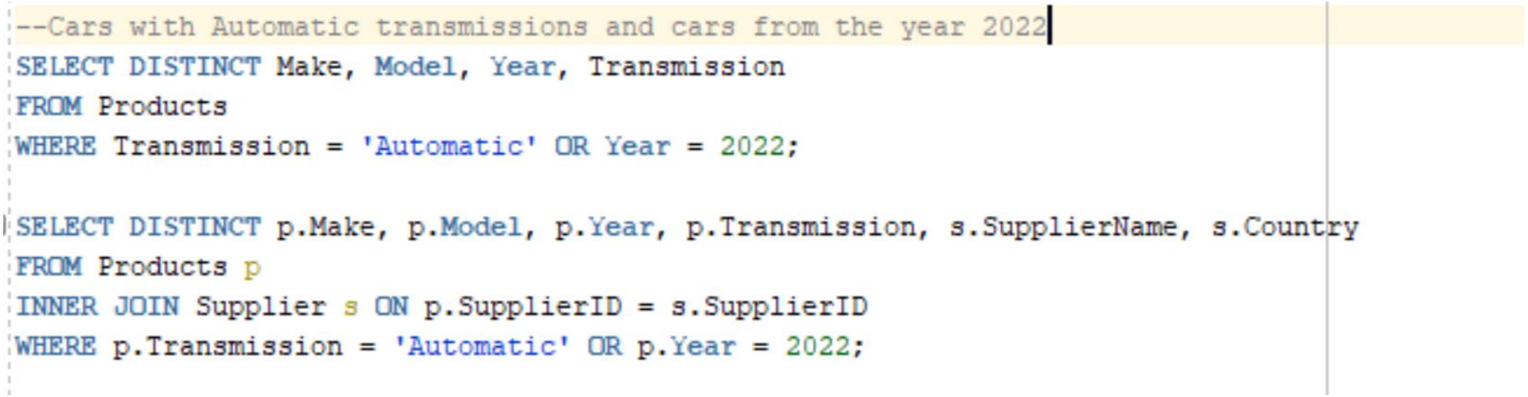
This task includes learning unit 11: Using the Set Operators. Write SQL statements to use Set Operator to Combine Multiple Queries into a Single Query:

4.1 List all unique car makes available in the Car Dealership Products Table and Supplier countries in a single column. (4 Marks)



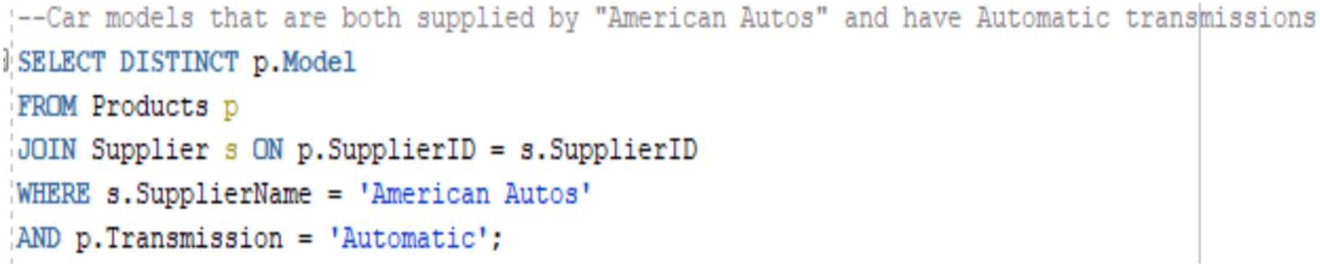
```
--Listing all unique car makes available in the Car Dealership Products Table and Supplier countries in a single column.
SELECT Make AS Unique_Items
FROM Products
UNION
SELECT Country AS Unique_Items
FROM Supplier;


SELECT Unique_Items
FROM (
    SELECT Make AS Unique_Items
    FROM Products
    UNION
    SELECT Country AS Unique_Items
    FROM Supplier
) CombinedData
ORDER BY Unique_Items;
```

4.2 List cars with Automatic transmissions and cars from the year 2022. Eliminate duplicate rows. (5 Marks)



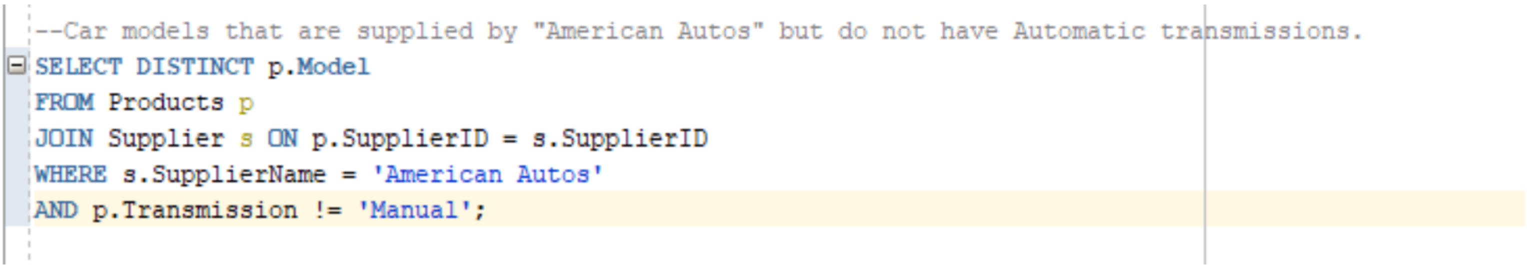
```
--Cars with Automatic transmissions and cars from the year 2022
SELECT DISTINCT Make, Model, Year, Transmission
FROM Products
WHERE Transmission = 'Automatic' OR Year = 2022;

SELECT DISTINCT p.Make, p.Model, p.Year, p.Transmission, s.SupplierName, s.Country
FROM Products p
INNER JOIN Supplier s ON p.SupplierID = s.SupplierID
WHERE p.Transmission = 'Automatic' OR p.Year = 2022;
```

4.3 Find car models that are both supplied by "American Autos" and have Automatic transmissions. (8 Marks)



```
--Car models that are both supplied by "American Autos" and have Automatic transmissions
SELECT DISTINCT p.Model
FROM Products p
JOIN Supplier s ON p.SupplierID = s.SupplierID
WHERE s.SupplierName = 'American Autos'
AND p.Transmission = 'Automatic';
```

4.4 List car models that are supplied by "American Autos" but do not have Automatic transmissions.(8 Marks)



```sql
--Car models that are supplied by "American Autos" but do not have Automatic transmissions.
SELECT DISTINCT p.Model
FROM Products p
JOIN Supplier s ON p.SupplierID = s.SupplierID
WHERE s.SupplierName = 'American Autos'
AND p.Transmission != 'Manual';
```

[Total = 25 Marks]

Completed Declaration of Authenticity

I Gabriella Rakgotsoka _ hereby
(FULL NAME)
declare that the contents of this assignment PRG522_FA3 is entirely my own work except for the following documents: (List the documents and page numbers of work in this portfolio that were generated in a group)

Activity	Date

Signature: Date: 2024/05/21