a

University of Houston - Downtown

Automobile Database Management System

Dylan Britain, Edgar Meza, Tai Quach

CS 4318, Database Management Systems

Professor Shengli Yuan

12/06/2021

Table of Contents

- Abstract -

We will be creating an Automobile Database Management System that will store information on vehicles that are being sold and rented at different dealership locations owned by a single company. The database will also have the ability to enter, update, and delete information pertaining to each dealership's receiving, service center, and parts warehouse. The database will be accessible by the employees of the company, such as the Director, Dealership Managers, Finance Managers, and individual Salespersons.

- Mission Statement -

The purpose of the Automobile Database Management System is to maintain the data pertaining to automobiles that are being sold at different dealership branches and the employees working for each dealership branch owned by John Doe Automotive.

- Mission Objectives -

To maintain (enter, update, and delete) data on dealership branches.

To maintain (enter, update, and delete) data on staff.

To maintain (enter, update, and delete) data on clients.

To maintain (enter, update, and delete) data on vehicles for sale at the dealerships.

To maintain (enter, update, and delete) data on vehicles leased at the dealerships.

To maintain (enter, update, and delete) data on rental vehicles.

To maintain (enter, update, and delete) data on purchased vehicles.

To maintain (enter, update, and delete) data on service centers.

To maintain (enter, update, and delete) data on parts warehouses.

To perform queries on branches.

To perform queries on staff.

To perform queries on clients.

To perform queries on vehicles for sale.

To perform queries on leases.

To perform queries on rentals.

To perform queries on purchased vehicles.

To perform queries on service centers.

To perform queries on parts warehouses.

To report on branches.

To report on staff.

To report on clients.

To report on vehicles for sale.

To report on leases.

To report on rentals.

To report on purchased vehicles.

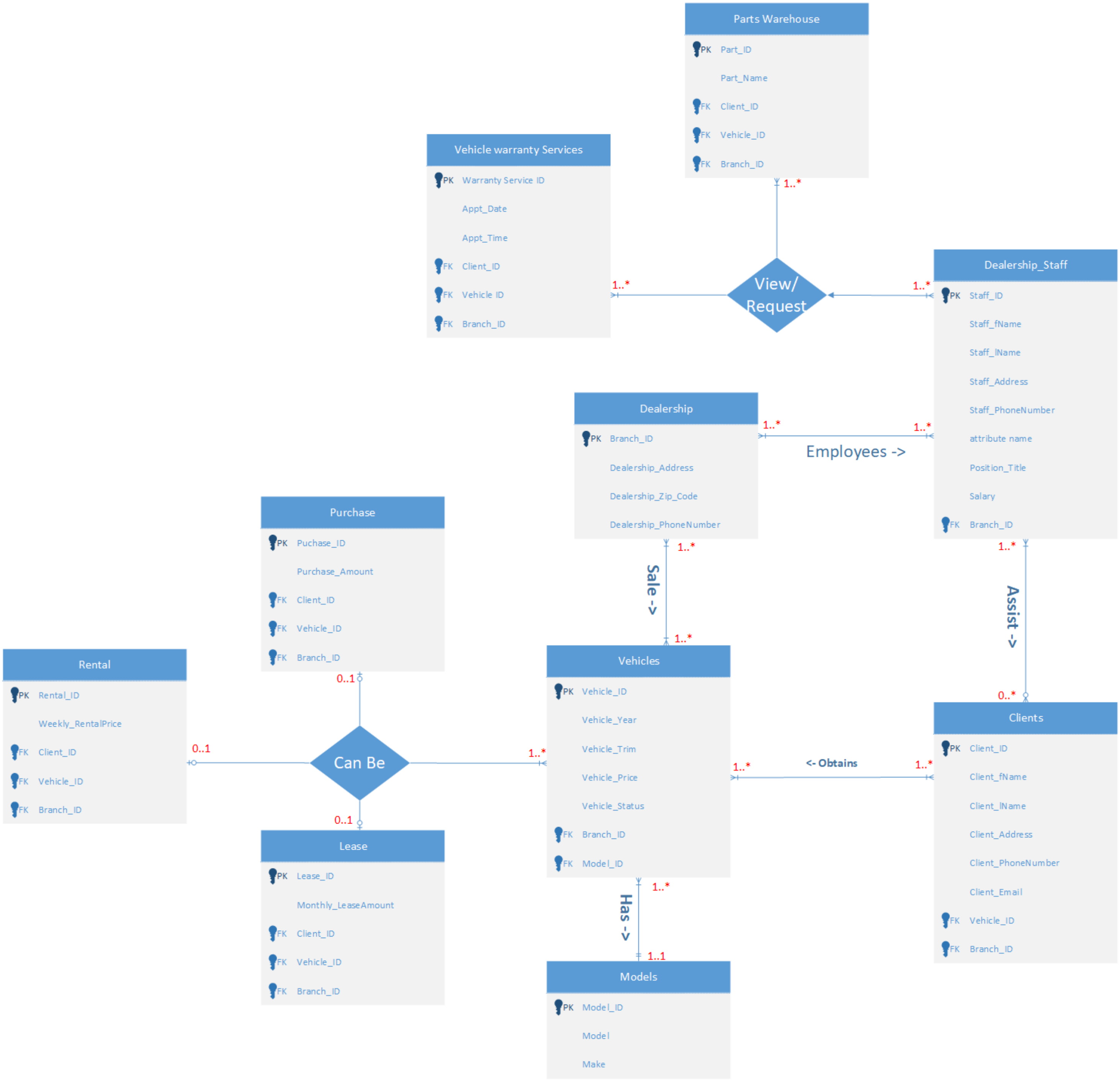
To report on service centers.

To report on parts warehouses.

- Major User Views -

DATA	ACCESS TYPE	DIRECTOR	DEALERSHIP MANAGER	FINANCE MANAGER	SALESPERSON
Dealership	Maintain	X	X		
	Query		X		
	Report	X	X		
Staff	Maintain		X		
	Query		X		
	Report	X	X		
Clients	Maintain		X	X	X
	Query		X	X	X
	Report	X	X	X	X
Available Vehicles	Maintain		X	X	
	Query		X	X	X
	Report	X	X	X	X
Vehicle Leases	Maintain		X	X	
	Query		X	X	X
	Report	X	X	X	X
Vehicle Rentals	Maintain		X		X
	Query		X		X
	Report	X	X		X
Vehicles Purchased	Maintain		X	X	
	Query		X		X
	Report	X	X		
Vehicle Warranty Services	Maintain		X		X
	Query		X		X
	Report	X	X		X
Parts Warehouse	Maintain		X		
	Query		X		X
	Report	X	X		X
Vehicle Models	Maintain		X		
	Query		X		X
	Report	X	X		X

- Entity-Relationship Diagram –



- Relational Models and BCNF Verification -

- Dealership Table

```
CREATE TABLE Dealership (
Branch_ID              VARCHAR (6)  NOT NULL,
Dealership_Address     VARCHAR (45) NOT NULL,
Dealership_PhoneNumber VARCHAR (10)  NOT NULL,
PRIMARY KEY (Branch_ID),
);
```

Dependencies:

Branch_ID -> Dealership_Address

Branch_ID -> Dealership_PhoneNumber

Normalization Justifications:

1NF – There is only one data entry for each cell. All values in each column have the same domain and unique names.

2NF – Is 1NF and there are no partial dependencies.

3NF - Is 2NF and does not contain any transitive dependencies.

BCNF – Is 3NF and Primary Key is not dependent on any non-Primary Key attributes.

- Dealership Staff Table

```
CREATE TABLE Dealership_Staff(
Staff_ID            VARCHAR (6)  NOT NULL,
Staff_fName         VARCHAR (45) NOT NULL,
Staff_lName         VARCHAR (45) NOT NULL,
Staff_Address       VARCHAR (45) NOT NULL,
Staff_PhoneNumber   VARCHAR (10) NOT NULL,
Position_Title      VARCHAR (45) NOT NULL,
Salary              FLOAT   (25) NOT NULL,
Branch_ID           VARCHAR (6)  NOT NULL,
PRIMARY KEY (Staff_ID),
FOREIGN KEY (Branch_ID) REFERENCES Dealership(Branch_ID),
);
```

Dependencies:

Staff_ID -> Staff_fName

Staff_ID -> lName

Staff_ID -> Staff_Address

Staff_ID -> Staff_PhoneNumber

Staff_ID -> Position_Title

Staff_ID -> Salary

Staff_ID -> Branch_ID

Normalization Justifications:

1NF – There is only one data entry for each cell. All values in each column have the same domain and unique names.

2NF – Is 1NF and there are no partial dependencies.

3NF - Is 2NF and does not contain any transitive dependencies.

BCNF – Is 3NF and Primary Key is not dependent on any non-Primary Key attributes.

- Vehicles Table

```
CREATE TABLE Vehicles(
Vehicle_ID     VARCHAR (8)  NOT NULL,
Model_ID       VARCHAR (6) NOT NULL,
Vehicle_Year   VARCHAR (4)  NOT NULL,
Vehicle_Trim   VARCHAR (45) NOT NULL,
Vehicle_Price  FLOAT   (25) NOT NULL,
Vehicle_Status VARCHAR (25) NOT NULL,
Branch_ID      VARCHAR (6)  NOT NULL,
PRIMARY KEY (Vehicle_ID),
FOREIGN KEY (Branch_ID) REFERENCES Dealership(Branch_ID),
FOREIGN KEY (Model_ID) REFERENCES Models(Model_ID)
);
```

Dependencies:

Vehicle_ID -> Model_ID

Vehicle_ID -> Vehicle_Year

Vehicle_ID -> Vehicle_Trim

Vehicle_ID -> Vehicle_Price

Vehicle_ID -> Vehicle_Status

Vehicle_ID -> Branch_ID

Normalization Justifications:

1NF – There is only one data entry for each cell. All values in each column have the same domain and unique names.

2NF – Is 1NF and there are no partial dependencies.

3NF - Is 2NF and does not contain any transitive dependencies.

BCNF – Is 3NF and Primary Key is not dependent on any non-Primary Key attributes.

- Lease Table

```
CREATE TABLE Lease(
Lease_ID               VARCHAR (6) NOT NULL,
Monthly_LeaseAmount    FLOAT   (25)NOT NULL,
Client_ID              VARCHAR (6) NOT NULL,
Vehicle_ID             VARCHAR (8) NOT NULL,
Branch_ID              VARCHAR (6) NOT NULL,
PRIMARY KEY (Lease_ID),
FOREIGN KEY (Client_ID) REFERENCES Clients(Client_ID),
FOREIGN KEY (Vehicle_ID)REFERENCES Vehicles(Vehicle_ID),
FOREIGN KEY (Branch_ID) REFERENCES Dealership(Branch_ID),
);
```

Dependencies:

Lease_ID -> Monthly_LeaseAmount

Lease_ID -> Client_ID

Lease_ID -> Vehicle_ID

Lease_ID -> Branch_ID

Normalization Justifications:

1NF – There is only one data entry for each cell. All values in each column have the same domain and unique names.

2NF – Is 1NF and there are no partial dependencies.

3NF - Is 2NF and does not contain any transitive dependencies.

BCNF – Is 3NF and Primary Key is not dependent on any non-Primary Key attributes.

- Rental Table

```
CREATE TABLE Rental(
Rental_ID             VARCHAR (6) NOT NULL,
Weekly_RentalPrice    FLOAT   (25)NOT NULL,
Client_ID             VARCHAR (6) NOT NULL,
Vehicle_ID            VARCHAR (8) NOT NULL,
Branch_ID             VARCHAR (6) NOT NULL,
PRIMARY KEY (Rental_ID),
FOREIGN KEY (Client_ID) REFERENCES Clients(Client_ID),
FOREIGN KEY (Vehicle_ID)REFERENCES Vehicles(Vehicle_ID),
FOREIGN KEY (Branch_ID) REFERENCES Dealership(Branch_ID),
);
```

Dependencies:

Rental_ID -> Weekly_RentalPrice

Rental_ID -> Client_ID

Rental_ID -> Vehicle_ID

Rental_ID -> Branch_ID

Normalization Justifications:

1NF – There is only one data entry for each cell. All values in each column have the same domain and unique names.

2NF – Is 1NF and there are no partial dependencies.

3NF - Is 2NF and does not contain any transitive dependencies.

BCNF – Is 3NF and Primary Key is not dependent on any non-Primary Key attributes.

- Parts Warehouse Table

```
CREATE TABLE Parts_Warehouse(
Part_ID              VARCHAR (6) NOT NULL,
Part_Name            VARCHAR (45)NOT NULL,
Client_ID            VARCHAR (6),
Vehicle_ID           VARCHAR (8),
Branch_ID            VARCHAR (6) NOT NULL,
PRIMARY KEY (Part_ID),
FOREIGN KEY (Client_ID) REFERENCES Clients(Client_ID),
FOREIGN KEY (Vehicle_ID)REFERENCES Vehicles(Vehicle_ID),
FOREIGN KEY (Branch_ID) REFERENCES Dealership(Branch_ID),
);
```

Dependencies:

Part_ID -> Part_Name

Part_ID -> Client_ID

Part_ID -> Vehicle_ID

Part_ID -> Branch_ID

Normalization Justifications:

1NF – There is only one data entry for each cell. All values in each column have the same domain and unique names.

2NF – Is 1NF and there are no partial dependencies.

3NF - Is 2NF and does not contain any transitive dependencies.

BCNF – Is 3NF and Primary Key is not dependent on any non-Primary Key attributes.

- Purchase Table

```
CREATE TABLE Purchase(
Purchase_ID          VARCHAR (6) NOT NULL,
Purchased_Amount     FLOAT   (25)NOT NULL,
Client_ID            VARCHAR (6) NOT NULL,
Vehicle_ID           VARCHAR (8) NOT NULL,
Branch_ID            VARCHAR (6) NOT NULL,
PRIMARY KEY (Purchase_ID),
FOREIGN KEY (Client_ID) REFERENCES Clients(Client_ID),
FOREIGN KEY (Vehicle_ID)REFERENCES Vehicles(Vehicle_ID),
FOREIGN KEY (Branch_ID) REFERENCES Dealership(Branch_ID),
);
```

Dependencies:

Purchase_ID -> Purchase_Amount

Purchase _ID -> Client_ID

Purchase _ID -> Vehicle_ID

Purchase _ID -> Branch_ID

Normalization Justifications:

1NF – There is only one data entry for each cell. All values in each column have the same domain and unique names.

2NF – Is 1NF and there are no partial dependencies.

3NF - Is 2NF and does not contain any transitive dependencies.

BCNF – Is 3NF and Primary Key is not dependent on any non-Primary Key attributes.

- Vehicle Warranty Services Table

```
CREATE TABLE Vehicle_WarrantyServices(
WarrantyService_ID    VARCHAR (6) NOT NULL,
Appt_Date             DATE     NOT NULL,
Appt_Time             TIME     NOT NULL,
Client_ID             VARCHAR (6) NOT NULL,
Vehicle_ID            VARCHAR (8) NOT NULL,
Branch_ID             VARCHAR (6) NOT NULL,
PRIMARY KEY (WarrantyService_ID),
FOREIGN KEY (Client_ID) REFERENCES Clients(Client_ID),
FOREIGN KEY (Vehicle_ID)REFERENCES Vehicles(Vehicle_ID),
FOREIGN KEY (Branch_ID) REFERENCES Dealership(Branch_ID),
);
```

Dependencies:

WarrantyService_ID -> Appt_Date

WarrantyService _ID -> Appt_Time

WarrantyService _ID -> Client_ID

WarrantyService _ID -> Vehicle_ID

WarrantyService _ID -> Branch_ID

Normalization Justifications:

1NF – There is only one data entry for each cell. All values in each column have the same domain and unique names.

2NF – Is 1NF and there are no partial dependencies.

3NF - Is 2NF and does not contain any transitive dependencies.

BCNF – Is 3NF and Primary Key is not dependent on any non-Primary Key attributes.

- Clients Table

```
CREATE TABLE Clients(
Client_ID           VARCHAR (6)  NOT NULL,
Client_fName        VARCHAR (45) NOT NULL,
Client_lName        VARCHAR (45) NOT NULL,
Client_Address      VARCHAR (45) NOT NULL,
Client_PhoneNumber  VARCHAR (10) NOT NULL,
Client_Email        VARCHAR (50) NOT NULL,
Vehicle_ID          VARCHAR (8),
Branch_ID           VARCHAR (6)  NOT NULL,
PRIMARY KEY (Client_ID),
FOREIGN KEY (Vehicle_ID)REFERENCES Vehicles(Vehicle_ID),
FOREIGN KEY (Branch_ID) REFERENCES Dealership(Branch_ID),
);
```

Dependencies:

Client_ID-> Client_fName

Client_ID-> Client_lName

Client_ID-> Client_Address

Client_ID-> Client_PhoneNumber

Client_ID-> Client_Email

Client_ID-> Vehicle_ID

Client_ID-> Branch_ID

Normalization Justifications:

1NF – There is only one data entry for each cell. All values in each column have the same domain and unique names.

2NF – Is 1NF and there are no partial dependencies.

3NF - Is 2NF and does not contain any transitive dependencies.

BCNF – Is 3NF and Primary Key is not dependent on any non-Primary Key attributes.

- Models Table

```
CREATE TABLE Models(
Model_ID        VARCHAR (6) NOT NULL,
Model           VARCHAR (24) NOT NULL,
Make            VARCHAR (24) NOT NULL
PRIMARY KEY (Model_ID)
);
```

Dependencies:

Model_ID -> Model

Model_ID -> Make

Normalization Justifications:

1NF – There is only one data entry for each cell. All values in each column have the same domain and unique names.

2NF – Is 1NF and there are no partial dependencies.

3NF - Is 2NF and does not contain any transitive dependencies.

BCNF – Is 3NF and Primary Key is not dependent on any non-Primary Key attributes.

- Use Cases -

1. *Use case name: Insert Dealership's Information within Dealership*

Actor/User: Dealership Manager, Director

Steps:

1. User clicks on "**Dealership**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Insert**" button;
4. User is prompted to enter the **Branch_ID** for the new branch.
5. Dealership information fields (**Dealership_Address**, **Dealership_PhoneNumber**, **Branch_ID**) loads up. The user **inserts** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Director, Dealership Manager:

```
INSERT INTO    Dealership(Dealership_Address,
               Dealership_PhoneNumber,Branch_ID)

VALUES         ('1234 Valhalla', '77087',

               '9175118512', 'br0005');
```

*2. **Use case name: Update Dealership's Information within Dealership***

Actor/User: Dealership Manager, Director

Steps:

1. User clicks on "**Dealership**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Update**" button
4. User is prompted to enter the **Branch_ID**;
5. Dealership information fields (**Dealership_Address**, **Dealership_PhoneNumber**, **Branch_ID**) loads up. The user **update** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Director, Dealership Manager:

```
UPDATE       Dealership
SET          Dealership_PhoneNumber = '8325740233'
 WHERE       Branch_ID = 'br0001';
```

*3. **Use case name: Delete Dealership's Information within Dealership***

Actor/User: Dealership Manager, Director

Steps:

1. User clicks on "**Dealership**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Delete**" button;
4. User is prompted to enter the **Branch_ID**;
5. All information is displayed on screen (**Dealership_Address**, **Dealership_PhoneNumber**, **Branch_ID**).
6. Prompt user to confirm the information to delete
7. User confirms delete by clicking on the "**Delete**" button.

Director, Dealership Manager:

```
DELETE FROM  Dealership
WHERE        Branch_ID = 'br0005';
```

4. Use case name: Querying Dealership's Information within Dealership

Actor/User: Dealership Manager

Steps:

1. User clicks on "**Dealership**" button;
2. User clicks on "**Query**" button;
3. Prompt user to enter dealership information (**Dealership_Address**, **Dealership_PhoneNumber**, **Branch_ID**);
4. User clicks on the "**Request**" button;
5. All information is displayed on the screen

Dealership Manager:

```
SELECT       *
FROM         Dealership
WHERE        Dealership_Address = '1234 Valhalla';
```

5. Use case name: Reporting Dealership's Information within Dealership

Actor/User: Dealership Manager

Steps:

1. User clicks on "**Dealership**" button;
2. User clicks on "**Report Data**" button;
3. Prompt User to enter the dealership information (**Dealership_Address**, **Dealership_PhoneNumber**, **Branch_ID**);
4. User clicks on "**Run Report**";
5. Report containing selected local dealerships' information is displayed on screen;
6. User may click on "**Print Report**" if needed

Dealership Manager:

```
SELECT DISTINCT Dealership_Address, Dealership_PhoneNumber,

                Dealership.Branch_ID, Parts_Warehouse.Part_Name

FROM Dealership

INNER JOIN                   Parts_Warehouse

ON Dealership.Branch_ID =    Parts_Warehouse.Branch_ID

ORDER BY                     Branch_ID;
```

6. Use case name: Insert Staff's Information within Dealership

Actor/User: Dealership Manager

Steps:

1. User clicks on "**Dealership_Staff**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Insert**" button;
4. User is prompted to enter the **Staff_ID** for the new staff;
5. Staff information fields (**Staff_fName, Staff_lName, Staff_Address, Staff_PhoneNumber, Position_Title, Salary, Branch_ID**) loads up. The user **inserts** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "Confirm" button.

Dealership Manager:

```
INSERT INTO Dealership_Staff(Staff_ID, Staff_fName,
             Staff_lName,   Staff_Address, Staff_PhoneNumber,
             Position_Title, Salary, Branch_ID)
 VALUES      ('MA8651', 'Maria', 'Ander', '2131
             Concord','9175118512','Sales',
             '5000.00', 'br0006');
```

7. *Use case name: Update Staff's Information within Dealership*

Actor/User: Dealership Manager

Steps:

1. User clicks on "**Dealership_Staff**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Update**" button
4. User is prompted to enter the **Staff_ID**;
5. Staff information fields (**Staff_fName, Staff_lName, Staff_Address, Staff_PhoneNumber, Position_Title, Salary, Branch_ID**) loads up. The user **updates** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "Confirm" button.

Dealership Manager:

```
UPDATE       Dealership_Staff

SET          Staff_lName = 'Oda', Staff_fName = 'John'

WHERE        Branch_ID = 'br0001';
```

8. *Use case name: Delete Staff's Information within Dealership*

Actor/User: Dealership Manager

Steps:

1. User clicks on "**Dealership_Staff**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Delete**" button;
4. User is prompted to enter the **Staff_ID**;
5. All information is displayed on screen (**Staff_fName, Staff_lName, Staff_Address, Staff_PhoneNumber, Position_Title, Salary, Branch_ID**).
6. User confirms delete by clicking on the "**Delete**" button.

Dealership Manager:

```
DELETE FROM      Dealership_Staff

WHERE            Staff_lName = 'Oda'

AND              Staff_fName = 'John';
```

9. Use case name: Querying Staff's information within Dealership

Actor/User: Dealership Manager

Steps:

1. User clicks on "**Dealership_Staff**" button;
2. User clicks on "**Query**" button;
3. Prompt user to enter staff information (**Staff_ID, Staff_fName, Staff_lName, Staff_Address, Staff_PhoneNumber, Position_Title, Salary**, **Branch_ID**);
4. User clicks on the "**Request**" button;
5. All information is displayed on the screen

Dealership Manager:

```
SELECT       Staff_PhoneNumber

FROM         Dealership_Staff

WHERE        Position_Title = 'Finance Manager'

AND          Branch_ID = 'br0007';
```

10. Use case name: Reporting on Staff's information within Dealership

Actor/User: Dealership Manager

Steps:

1. Users click on "**Dealership_Staff**" button;
2. User clicks on "**Report Data**" button;
3. Prompt user to enter staff information (**Staff_ID, Staff_fName, Staff_lName, Staff_Address, Staff_PhoneNumber, Position_Title, Salary**, **Branch_ID**);
4. User clicks on "**Run Report**";
5. Report containing staff information within selected branches is displayed on screen;
6. User may click on "**Print Report**" if needed.

Dealership Manager:

```
SELECT Staff_fName, Staff_lName, Dealership_Staff.Branch_ID

FROM                                Dealership_Staff

INNER JOIN                          Dealership

ON Dealership_Staff.Branch_ID =     Dealership.Branch_ID

ORDER BY                            Branch_ID;
```

11. Use case name: Insert Client's Information within Dealership

Actor/User: Dealership Manager, Finance Manager, Salesperson

Steps:

1. User clicks on "**Clients**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Insert**" button;
4. User is prompted to enter the **Client_ID** for the new client;
5. Client information fields (**Client_fName, Client_lName, Client_Address, Client_PhoneNumber, Client_Email, Vehicle_ID, Branch_ID**) loads up. The user **inserts** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Dealership Manager, Finance Manager, Salesperson:

```
INSERT INTO Clients (Client_fName, Client_lName,
                    Client_Address, Client_PhoneNumber,
                    Client_Email, Vehicle_ID, Branch_ID)
VALUES              ('Mike', 'Turner', '1234 Dead-end Dr',
                    '7954843658', 'mt123@gmail.com',
                    'lxis1234', 'br0002');
```

12. Use case name: Update Client’s Information within Dealership

Actor/User: Dealership Manager, Finance Manager, Salesperson

Steps:

1. User clicks on “**Clients**” button;
2. User clicks on “**Maintain**” button;
3. User clicks on “**Update**” button
4. User is prompted to enter the **Client_ID**;
5. Client information fields (**Client_fName, Client_lName, Client_Address, Client_PhoneNumber, Client_Email, Vehicle_ID, Branch_ID**) loads up. The user **updates** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the “**Confirm**” button.

Dealership Manager, Finance Manager, Salesperson:

```
UPDATE      Clients

SET         Vehicle_ID = ‘totu1234’, Client_Email =

            ‘jo123@gmail.com’

WHERE       Branch_ID = ‘br0001’;
```

13. Use case name: Delete Client’s Information within Dealership

Actor/User: Dealership Manager, Finance Manager, Salesperson

Steps:

1. User clicks on “**Clients**” button;
2. User clicks on “**Maintain**” button;
3. User clicks on “**Delete**” button;
4. User is prompted to enter the **Client_ID**;
5. All information is displayed on screen (**Client_ID, Client_fName, Client_lName, Client_Address, Client_PhoneNumber, Client_Email, Vehicle_ID, Branch_ID**).
6. Prompt user to confirm the information to delete
7. User confirms delete by clicking on the “**Delete**” button.

Dealership Manager, Finance Manager, Salesperson:

```
DELETE FROM Clients

WHERE       Client_lName = ‘Turner’

AND         Client_ID = ‘Cmt123’;
```

14. Use case name: Querying Clients within Dealership

Actor/User: Dealership Manager, Finance Manager, Salesperson

Steps:

1. User clicks on "**Clients**" button;
2. User clicks on "**Query**" button;
3. Prompt user to enter clients' information (**Client_ID, Client_fName, Client_lName, Client_Address, Client_PhoneNumber, Client_Email, Vehicle_ID, Branch_ID**);
4. User clicks on the "**Request**" button;
5. All information is displayed on the screen

Dealership Manager, Finance Manager, Salesperson:

```
SELECT DISTINCT COUNT (*) AS Total_Clients

FROM              Dealership

WHERE             Branch='br0008'
```

15. Use case name: Reporting on Clients within Dealership

Actor/User: Dealership Manager, Financial Manager, Salesperson

Steps:

1. User click on “**Clients**” button;
2. User clicks on “**Report Data**” button;
3. Prompt user to enter client information (**Client_ID, Client_fName, Client_lName, Client_Address, Client_PhoneNumber, Client_Email, Vehicle_ID, Branch_ID**);
4. User clicks on “**Run Report**”;
5. Report containing all clients’ information within selected local dealerships are displayed on screen.
6. User may click on “**Print Report**” if needed.

Dealership Manager, Finance Manager, Salesperson:

```
SELECT DISTINCT          Client_ID, Client_fName,

                         Client_PhoneNumber, Client.Branch_ID

FROM                     Client

INNER JOIN               Dealership

ON Client.Branch_ID =    Dealership.Branch_ID

ORDER BY                 Branch_ID;
```

16. Use case name: Insert Vehicle for Sale's Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on “**Vehicles for Sale**” button;
2. User clicks on “**Maintain**” button;
3. User clicks on “**Insert**” button;
4. User is prompted to enter the **Vehicle_ID** for the new vehicle;
5. Vehicle for Sale’s information fields (**Vehicle_Year, Model_ID, Vehicle_Trim, Vehicle_Price, Vehicle_Status, Branch_ID**) load up. The user **inserts** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the “**Confirm**” button.

Dealership Manager, Finance Manager:

```
INSERT INTO Vehicles_For_Sale (Vehicle_Year, Model_ID,

                              Vehicle_Trim, Vehicle_Price,

                              Vehicle_Status, Vehicle_Year,

                              Branch_ID)

VALUES                        (‘2010’, ‘toy4ru’, ‘SR5 Premium’,
                              ‘For sale’, ‘2022’, ‘br0002’);
```

17. Use case name: Update Vehicle for Sale's Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Vehicles for Sale**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Update**" button
4. User is prompted to enter the **Vehicle_ID**;
5. Vehicle for Sale's information fields (**Vehicle_Year, Model_ID, Vehicle_Trim, Vehicle_Price, Vehicle_Status, Branch_ID**) load up. The user **updates** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Dealership Manager, Finance Manager:

```
UPDATE    Vehicles_For_Sale

SET       Vehicle_Price = '35421.00', Branch_ID = 'br0004'

WHERE     Vehicle_ID = 'fofu7613';
```

18. Use case name: Delete Vehicle for Sale's Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Vehicles for Sale**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Delete**" button;
4. User is prompted to enter the **Vehicle_ID**;
5. All information is displayed on screen (**Vehicle_ID, Vehicle_Year, Model_ID, Vehicle_Trim, Vehicle_Price, Vehicle_Status, Branch_ID**).
6. User confirms delete by clicking on the "**Delete**" button.

Dealership Manager, Finance Manager:

```
DELETE FROM      Vehicles_For_Sale

WHERE            Vehicle_ID = 'caes2865';
```

19. Use case name: Querying Vehicles for Sale within Dealership

Actor/User: Dealership Manager, Finance Manager, Salesperson

Steps:

1. User clicks on **“Vehicles for Sale”** button;
2. User clicks on **“Query”** button;
3. Prompt user to enter vehicles information (**Vehicle_ID, Vehicle_Year, Model_ID, Vehicle_Trim, Vehicle_Price, Vehicle_Status, Branch_ID**);
4. User clicks on the **“Request”** button;
5. All information is displayed on the screen

Dealership Manager, Finance Manager, Salesperson:

```
SELECT    Model_ID, Vehicle_Price

FROM      Vehicles_For_Sale

WHERE     Vehicle_Price >= ‘29800.00’;

ORDER BY  Vehicle_Price DESC;
```

20. Use case name: Reporting on Available Vehicles within Dealership

Actor/User: Dealership Manager, Finance Manager, Salesperson

Steps:

1. User clicks on **"Available Vehicles"** button;
2. User clicks on **"Report Data"** button;
3. Prompt user to enter vehicle information (**Vehicle ID, Vehicle_Year, Model_ID, Vehicle Trim, Vehicle Price, Vehicle Status, Branch ID**);
4. User clicks on **"Run Report"**;
5. Report containing all information for Vehicles for Sale Within the Local Dealership is displayed on screen.
6. User may click on **"Print Report"** if needed.

Dealership Manager, Finance Manager, Salesperson:

```
SELECT DISTINCT     Vehicle_ID, Vehicle_Price,

                    Vehicle_Status, Vehicle.Branch_ID

FROM                Vehicles

INNER JOIN          Dealership

ON Vehicles.Branch_ID = Dealership.Branch_ID

ORDER BY            Branch_ID;
```

21. Use case name: Insert Vehicle for Lease's Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on “**Vehicles for Lease**” button;
2. User clicks on “**Maintain**” button;
3. User clicks on “**Insert**” button;
4. User is prompted to enter the **Lease_ID** for the new vehicle for lease;
5. Vehicle for Lease’s information fields (**Monthly_LeaseAmount, Client_ID, Vehicle_ID, Branch_ID**) loads up. The user **inserts** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the “**Confirm**” button.

Dealership Manager, Finance Manager:

```
INSERT INTO Leases  (Lease_ID, Monthly_LeaseAmount, Client_ID,
                    Vehicle_ID, Branch_ID)
VALUES              (‘chsi5637’, '500’, ‘cob678’, ‘fo3975’,
                    ‘br0006’);
```

22. Use case name: Update Vehicle for Lease’s Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on “**Vehicle for Lease**” button;
2. User clicks on “**Maintain**” button;
3. User clicks on “**Update**” button
4. User is prompted to enter the **Lease_ID**;
5. Vehicle for Lease’s information fields (**Monthly_LeaseAmount, Client_ID, Vehicle_ID, Branch_ID**) loads up. The user **updates** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the “**Confirm**” button.

Dealership Manager, Finance Manager:

```
UPDATE    Lease
SET       Monthly_LeaseAmount = ‘725.00’
WHERE     Lease_ID = ‘L17613’;
```

23. Use case name: Delete Vehicle for Lease's Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Clients**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Delete**" button;
4. User is prompted to enter the **Lease_ID**;
5. All information is displayed on screen (**Monthly_LeaseAmount, Client_ID, Vehicle_ID, Branch_ID**).
6. Prompt user to confirm the information to delete
7. User confirms delete by clicking on the "**Delete**" button.

Dealership Manager, Finance Manager:

```
DELETE FROM       Leases

WHERE             Lease_ID = 'L15637'
```

24. Use case name: Querying Vehicle Leases within Dealership

Actor/User: Dealership Manager, Finance Manager, Salesperson

Steps:

1. User clicks on **"Vehicles"** button;
2. User clicks on **"Query"** button;
3. Prompt user to enter the Vehicle for Lease's information (**Lease_ID, Monthly_LeaseAmount, Client_ID, Vehicle_ID, Branch_ID**);
4. User clicks on the **"Request"** button;
5. All information is displayed on the screen;

Dealership Manager, Finance Manager, Salesperson

```
SELECT    Lease_ID, Monthly_LeaseAmount,

          Client_ID, Vehicle_ID, Branch_ID

FROM      Leases

WHERE     Branch_ID = 'br0008'

ORDER BY  Client_ID;
```

25. Use case name: Reporting on Leases within Dealership

Actor/User: Dealership Manager, Finance Manager, Salesperson

Steps:

1. User clicks on **"Leases"** button;
2. User clicks on **"Report Data"** button;
3. Prompt user to enter the Vehicle for Lease's information (**Lease_ID, Monthly_LeaseAmount, Client_ID, Vehicle_ID, Branch_ID**);
4. User clicks on **"Run Report"**;
5. Report containing all lease information for Vehicles for Lease within the selected Branch are displayed on screen.
6. User may click on **"Print Report"** if needed.

Dealership Manager, Finance Manager, Salesperson:

```
SELECT DISTINCT     Lease_ID, Monthly_LeaseAmount, Vehicle_ID,

                    Lease.Branch_ID

FROM                Lease

INNER JOIN          Dealership

ON Lease.Branch_ID = Dealership.Branch_ID

ORDER BY            Lease_ID;
```

26. Use case name: Insert Rental Vehicle's Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Rentals**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Insert**" button;
4. User is prompted to enter the **Rental_ID** for the new vehicle for lease;
5. Rental's information fields (**Rental_ID, Weekly_RentalPrice, Client_ID, Vehicle_ID, Branch_ID**) loads up. The user **inserts** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Dealership Manager, Finance Manager:

```
INSERT INTO    Rental
VALUES         ('r78647', '325.84', 'crt879,
               'totu9857', 'br0005');
```

27. Use case name: Update Rental Vehicle's Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Rentals**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Update**" button
4. User is prompted to enter the **Rental_ID**;
5. Rental's information fields (**Rental_ID, Weekly_RentalPrice, Client_ID, Vehicle_ID, Branch_ID**) load up. The user **updates** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Dealership Manager, Finance Manager:

```
     UPDATE    Rentals
     SET       Vehicle_ID = 'kisp2059' AND
               Weekly_RentalPrice = '499.58'
     WHERE     Client_ID = 'cjh1964';
```

28. Use case name: Delete Rental Vehicle's Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Rentals**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Delete**" button;
4. User is prompted to enter the **Rental_ID**;
5. All information is displayed on screen (**Rental_ID, Weekly_RentalPrice, Client_ID, Vehicle_ID, Branch_ID**).
6. Prompt user to confirm the information to delete
7. User confirms delete by clicking on the "**Delete**" button.

Dealership Manager, Finance Manager:

```
DELETE FROM  Rentals

WHERE        Vehicle_ID = 'kisp2059'
```

29. Use case name: Querying Rental Vehicles within Dealership

Actor/User: Dealership Manager, Finance Manager, Salesperson

Steps:

1. User clicks on **"Rentals"** button;
2. User clicks on **"Query"** button;
3. Prompt user to enter Rental information (**Rental_ID, Weekly_RentalPrice, Client_ID, Vehicle_ID, Branch_ID**);
4. User clicks on the **"Request"** button;
5. All information is displayed on the screen;

Dealership Manager, Finance Manager, Salesperson:

```
SELECT DISTINCT COUNT (*) AS     Total_Rentals

FROM                             Rentals

WHERE                            Branch_ID='br0007'
```

30. Use case name: Reporting on Rental Vehicles within Dealership

Actor/User: Dealership Manager, Finance Manager, Salesperson

Steps:

1. User clicks on **"Rentals"** button;
2. User clicks on **"Report Data"** button;
3. Prompt user to enter Rental information (**Rental_ID, Weekly_RentalPrice, Client_ID, Vehicle_ID, Branch_ID**);
4. User clicks on **"Run Report"**;
5. Report containing all information for Rentals is displayed on screen.
6. User may click on **"Print Report"** if needed.

Dealership Manager, Finance Manager, Salesperson:

```
SELECT DISTINCT Rental_ID, Weekly_RentalPrice, Rental.Branch_ID

FROM            Rental

INNER JOIN      Dealership

ON Rental.Branch_ID = Dealership.Branch_ID

ORDER BY        Weekly_RentalPrice;
```

31. Use case name: Insert Purchased Vehicle's Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Purchased Vehicles**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Insert**" button;
4. User is prompted to enter the **Purchase_ID** for the new purchased vehicle;
5. Purchased Vehicle's information fields (**Client_ID, Vehicle_ID, Branch_ID**) loads up. The user **inserts** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Dealership Manager, Finance Manager:

```
INSERT INTO    Purchase

VALUES         ('ckr1423', 'nial1590', 'br0002');
```

32. Use case name: Update Purchased Vehicle's Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Purchased Vehicles**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Update**" button
4. User is prompted to enter the **Purchase_ID**;
5. Purchased Vehicle's information fields (**Client_ID, Vehicle_ID, Branch_ID**) load up. The user **updates** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Dealership Manager, Finance Manager:

```
UPDATE    Purchase

SET       Client_ID = 'cjh1964'

WHERE     Vehicle_ID = 'kifo2059' AND Branch_ID =

          'br0002';
```

33. Use case name: Delete Purchased Vehicle's Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Purchased Vehicles**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Delete**" button;
4. User is prompted to enter the **Purchase_ID**;
5. All information is displayed on screen (**Purchase_ID, Client_ID, Vehicle_ID, Branch_ID**).
6. Prompt user to confirm the information to delete;
7. User confirms delete by clicking on the "**Delete**" button.

Dealership Manager, Finance Manager:

```
DELETE FROM         Purchase

WHERE               Vehicle_ID = 'kifo2059'
```

34. Use case name: Querying Purchased Vehicles within Dealership

Actor/User: Dealership Manager, Salesperson

Steps:

1. User clicks on **“Purchased”** button;
2. User clicks on **“Query”** button;
3. Prompt user to enter Purchased information (**Purchase_ID, Client_ID, Vehicle_ID, Branch_ID**);
4. User clicks on the **“Request”** button;
5. All information is displayed on the screen;

Dealership Manager, Salesperson:

```
SELECT *
FROM   Purchase
WHERE  Vehicle_ID = ‘chsi2301’;
```

35. Use case name: Reporting on Purchased Vehicles within Dealership

Actor/User: Dealership Manager

Steps:

1. User clicks on **“Purchased”** button;
2. User clicks on **“Report Data”** button;
3. Prompt user to enter Purchased information (**Purchase_ID, Client_ID, Vehicle_ID, Branch_ID**);
4. User clicks on **“Run Report”**;
5. Report containing all information for Rentals is displayed on screen.
6. User may click on **“Print Report”** if needed.

Dealership Manager:

```
SELECT DISTINCT     Purchase_ID, Purchased_Amount, Client_ID,

                    Purchase.Branch_ID

FROM                Purchase

INNER JOIN          Dealership

ON Purchase.Branch_ID = Dealership.Branch_ID

ORDER BY Client_ID;
```

36. Use case name: Insert Vehicle Warranty Service *within Dealership*

Actor/User: Dealership Manager, Salesman

Steps:

1. User clicks on "**Service Centers**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Insert**" button;
4. User is prompted to enter the **WarrantyService_ID** for the new service center;
5. Warranty Service's information fields (**Appt_Date, Appt_Time, Client_ID, Vehicle_ID, Branch_ID**) loads up. The user **inserts** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Dealership Manager, Salesman:

```
INSERT INTO    Vehicle_WarrantyServices

VALUES         ('WS7345', '2021-12-12', '13:30', 'ctw623',

               'chsi0483', 'br004');
```

37. Use case name: Update Vehicle Warranty Service within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Service Centers**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Update**" button
4. User is prompted to enter the **WarrantyService_ID**;
5. Warranty Service's information fields (**Appt_Date, Appt_Time, Client_ID, Vehicle_ID, Branch_ID**) load up. The user **updates** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Dealership Manager, Finance Manager:

```
UPDATE    Vehicle_WarrantyServices

SET       Appt_Date = '2021-12-12' AND Appt_Time =

          '13:30:00'

WHERE     WarrantyService_ID = 'WS1379';
```

38. Use case name: Delete Vehicle_WarrantyServices Information within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Service Centers**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Delete**" button;
4. User is prompted to enter the **WarrantyService_ID**;
5. All information is displayed on screen (**WarrantyService_ID, Appt_Date, Appt_Time, Client_ID, Vehicle_ID, Branch_ID**).
6. Prompt user to confirm the information to delete;
7. User confirms delete by clicking on the "**Delete**" button.

Dealership Manager, Finance Manager:

```
DELETE FROM    Vehicle_WarrantyServices
WHERE          Vehicle_ID = 'kifo2059';
```

39. Use case name: Querying Vehicle Warranty Services within Dealership

Actor/User: Branch Manager, Salesman

Steps:

1. User clicks on **"Service Centers"** button;
2. User clicks on **"Query"** button;
3. Prompt user to enter Service Center information (**WarrantyService_ID, Appt_Date, Appt_Time, Client_ID, Vehicle_ID, Branch_ID**);
4. User clicks on the **"Request"** button;
5. All information is displayed on the screen;

Branch Manager, Salesman:

```
 SELECT DISTINCT COUNT (*) AS    Total_WarrantyServices
FROM                             WarrantyServices
WHERE                            Branch_ID='br0005'
```

40. Use case name: Reporting on Vehicle Warranty Services within Dealership

Actor/User: Branch Manager, Salesman

Steps:

1. User clicks on **“Service Centers”** button;
2. User clicks on **“Report Data”** button;
3. Prompt user to enter Service Center information (**WarrantyService_ID, Appt_Date, Appt_Time, Client_ID, Vehicle_ID, Branch_ID**);
4. User clicks on **“Run Report”**;
5. Report containing all information for the local dealership’s Service Center is displayed on screen.
6. User may click on **“Print Report”** if needed.

Branch Manager, Salesman:

```
SELECT DISTINCT     WarrantyService_ID, Client_ID, Appt_Date,

                    Appt_Time,

                    Vehicle_WarrantyServices.Branch_ID

FROM                Vehicle_WarrantyServices

INNER JOIN          Dealership

ON Vehicle_WarrantyServices.Branch_ID = Dealership.Branch_ID

ORDER BY            Client_ID;
```

41. Use case name: Insert Parts Warehouse within Dealership

Actor/User: Dealership Manager, Salesman

Steps:

1. User clicks on "**Parts Warehouses**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Insert**" button;
4. User is prompted to enter the **Part_ID** for the new part;
5. Parts Warehouse's information fields (**Part_ID, Part_Name, Vehicle_ID, Client_ID, Branch_ID**) load up. The user **inserts** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Dealership Manager, Salesman:

```
INSERT    INTO Parts_Warehouse

VALUES    ('pt3869', 'Rearview Mirror',

          'fofu0423', 'clj153', 'br002');
```

42. Use case name: Update Parts Warehouse within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Parts Warehouses**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Update**" button
4. User is prompted to enter the **Part_ID**;
5. Parts Warehouse's information fields (**Part_Name, Vehicle_ID, Client_ID, Branch_ID**) load up. The user **updates** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the "**Confirm**" button.

Dealership Manager, Finance Manager:

```
UPDATE    Parts_Warehouse

SET       Branch_ID = 'br0006'

WHERE     Part_ID BETWEEN 'pt0100' AND 'pt0199'
```

43. Use case name: Delete Parts Warehouse within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on "**Parts Warehouses**" button;
2. User clicks on "**Maintain**" button;
3. User clicks on "**Delete**" button;
4. User is prompted to enter the **Part_ID**;
5. All information is displayed on screen (**Part_ID, Part_Name, Vehicle_ID, Client_ID, Branch_ID**).
6. Prompt user to confirm the information to delete;
7. User confirms delete by clicking on the "**Delete**" button.

Dealership Manager, Finance Manager:

```
DELETE FROM    Parts_Warehouse
WHERE          Part_ID BETWEEN 'pt0200' AND
               'pt0249'
```

44. Use case name: Querying Parts Warehouse within Dealership

Actor/User: Branch Manager, Salesman

Steps:

1. User clicks on **"Parts Warehouse"** button;
2. User clicks on **"Query"** button;
3. Prompt user to enter Parts Warehouse information (**Part ID, Part Name, Client ID, Vehicle ID, Branch ID**);
4. User clicks on the **"Request"** button;
5. All information is displayed on the screen;

Branch Manager, Salesman:

```
SELECT *
FROM Parts Warehouse
WHERE Part_ID = 'pt0694';
```

45. Use case name: Reporting on Parts Warehouse within Dealership

Actor/User: Branch Manager, Salesman

Steps:

1. User clicks on **“Parts Warehouse”** button;
2. User clicks on **“Report”** button;
3. Prompt user to enter Part information (**Part ID, Part Name, Client ID, Vehicle ID, Branch ID**);
4. User clicks on **“Run Report”**;
5. Report containing all information for the Parts Warehouse at Local Dealership is displayed on screen.
6. User may click on **“Print Report”** if needed.

Branch Manager, Salesman:

```
SELECT DISTINCT     Part_ID, Part_Name, Vehicle_ID,

                    Parts_Warehouse.Branch_ID

FROM                Parts_Warehouse

INNER JOIN          Dealership

ON Parts_Warehouse.Branch_ID = Dealership.Branch_ID

ORDER BY            Part_Name;
```

46. Use case name: Insert Models within Dealership

Actor/User: Dealership Manager, Salesman

Steps:

1. User clicks on “**Models**” button;
2. User clicks on “**Maintain**” button;
3. User clicks on “**Insert**” button;
4. User is prompted to enter the **Model_ID** for the new Model;
5. Model’s information fields (**Model, Make**) load up. The user **inserts** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the “**Confirm**” button.

Dealership Manager, Salesman:

```
INSERT    INTO Models
VALUES    (‘chesil’, ‘Silverado’, ‘Chevrolet’);
```

47. Use case name: Update Models within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on “**Models**” button;
2. User clicks on “**Maintain**” button;
3. User clicks on “**Update**” button
4. User is prompted to enter the **Model_ID**;
5. Model’s information fields (**Model, Make**) load up. The user **updates** fields as needed;
6. Prompt user to confirm the information;
7. User confirms by clicking on the “**Confirm**” button.

Dealership Manager, Finance Manager:

```
UPDATE    Models
SET       Model = ‘Trax’
WHERE     Model = ‘Traz’
```

48. Use case name: Delete Models within Dealership

Actor/User: Dealership Manager, Finance Manager

Steps:

1. User clicks on **“Models”** button;
2. User clicks on “**Maintain**” button;
3. User clicks on “**Delete**” button;
4. User is prompted to enter the **Model_ID**;
5. All information is displayed on screen **(Model_ID, Model, Make).**
6. Prompt user to confirm the information to delete;
7. User confirms delete by clicking on the “**Delete**” button.

Dealership Manager, Finance Manager:

```
DELETE FROM    Models

WHERE          Model_ID BETWEEN ‘che000’ AND

               ‘chezzz’
```

49. Use case name: Querying Models within Dealership

Actor/User: Branch Manager, Salesman

Steps:

1. User clicks on **“Models”** button;
2. User clicks on **“Query”** button;
3. Prompt user to enter Model information **(Model_ID, Model, Make);**
4. User clicks on the **“Request”** button;
5. All information is displayed on the screen;

Branch Manager, Salesman:

```
SELECT    *

FROM      Models

WHERE     Make = ‘Chevrolet’;
```

50. Use case name: Reporting on Models within Dealership

Actor/User: Branch Manager, Salesman

Steps:

1. User clicks on **"Models"** button;
2. User clicks on **"Report"** button;
3. Prompt user to enter Model Information (**Model_ID, Model, Make**);
4. User clicks on **"Run Report"**;
5. Report containing all information for the Model is displayed on screen.
6. User may click on **"Print Report"** if needed.

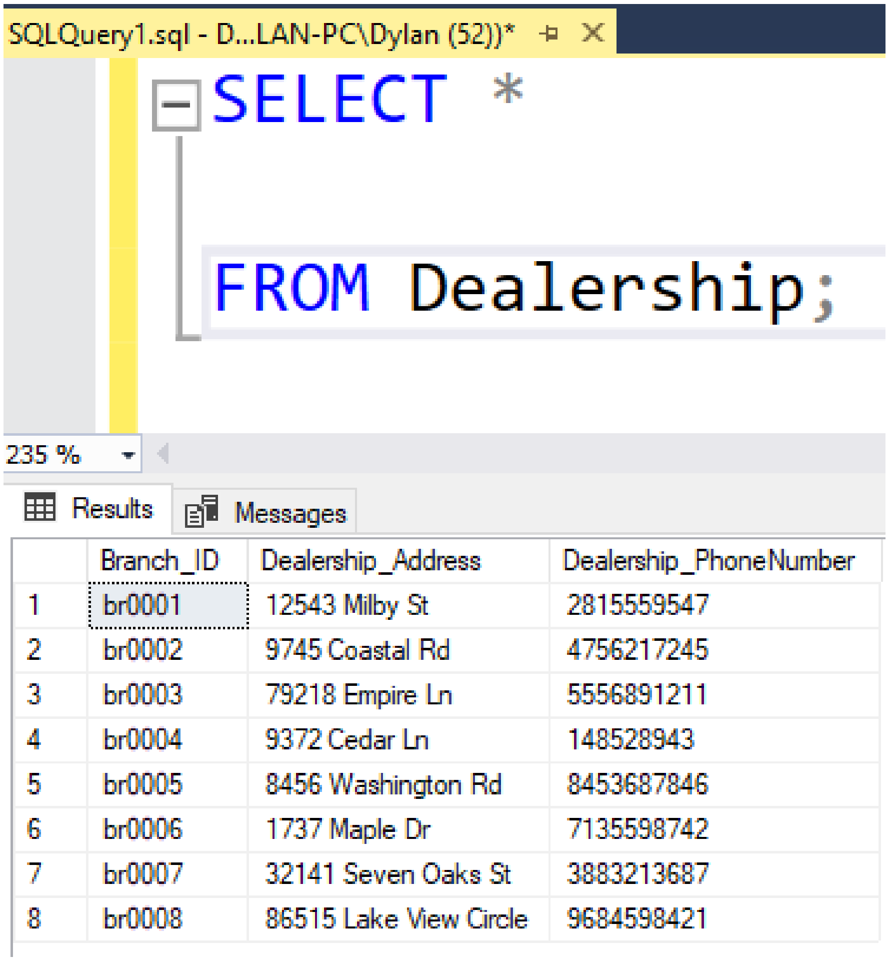
Branch Manager, Salesman:

```
SELECT DISTINCT     Model_ID, Model, Make, Vehicle.price

FROM                Models

INNER JOIN          Vehicles

ON Models.Model_ID = Vehicles.Model_ID

ORDER BY            Model_ID;
```

- Testing -

SHOW ALL DATA SQL STATEMENTS

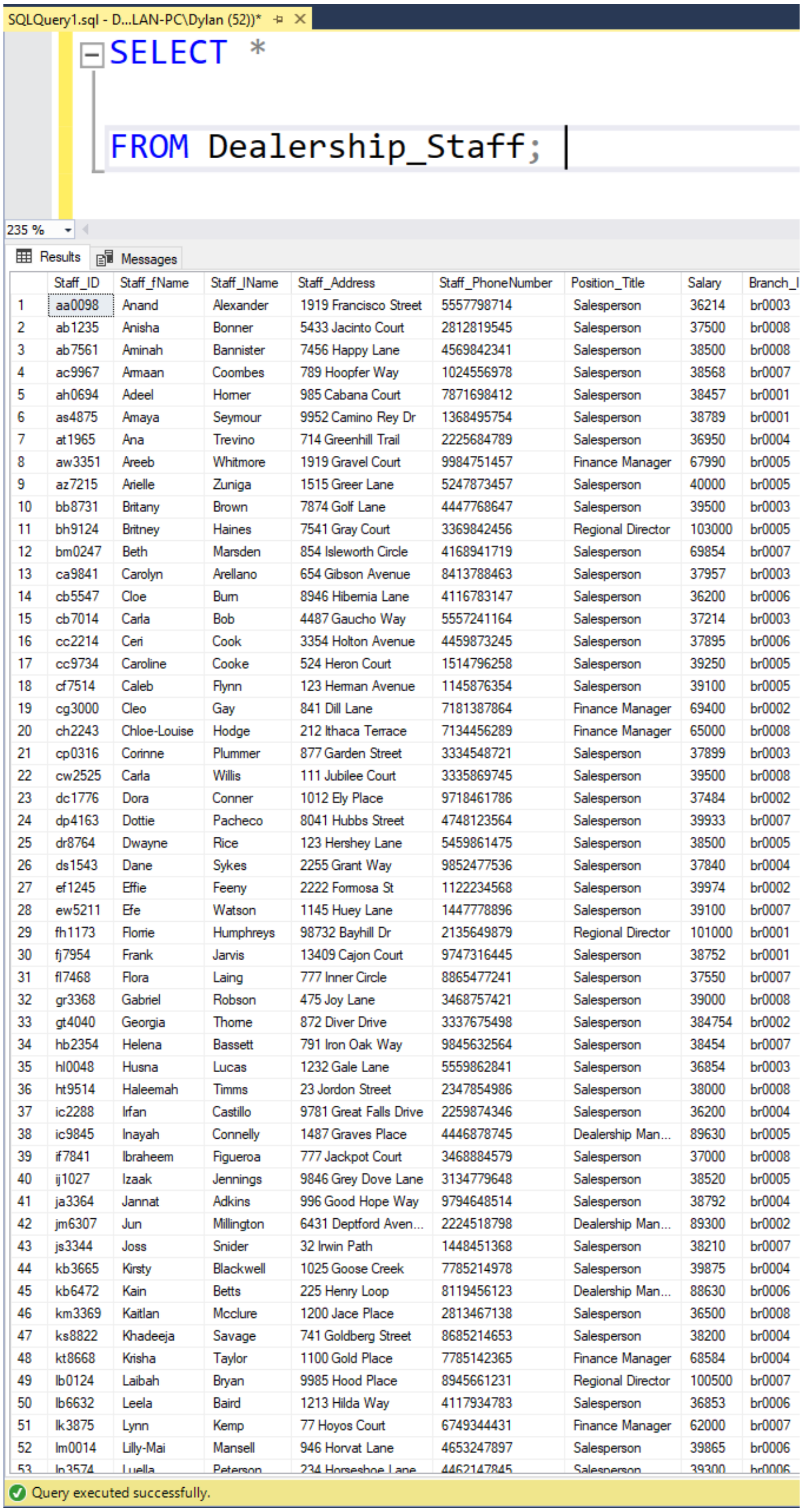
Query the Dealership Table



```
SELECT *

FROM Dealership;
```

	Branch_ID	Dealership_Address	Dealership_PhoneNumber
1	br0001	12543 Milby St	2815559547
2	br0002	9745 Coastal Rd	4756217245
3	br0003	79218 Empire Ln	5556891211
4	br0004	9372 Cedar Ln	148528943
5	br0005	8456 Washington Rd	8453687846
6	br0006	1737 Maple Dr	7135598742
7	br0007	32141 Seven Oaks St	3883213687
8	br0008	86515 Lake View Circle	9684598421

Query the Dealership_Staff Table

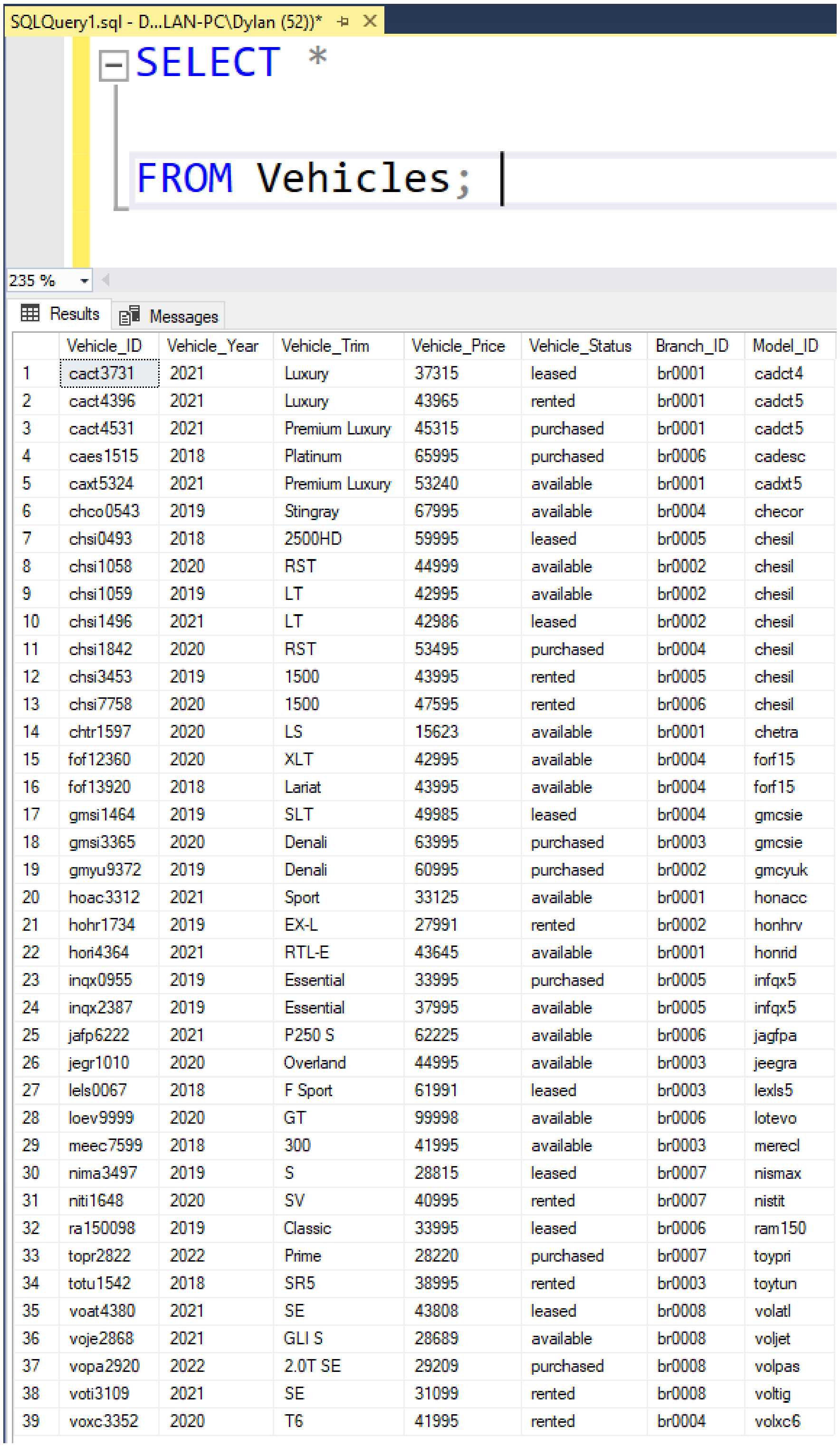


```
SELECT *

FROM Dealership_Staff;
```

	Staff_ID	Staff_fName	Staff_lName	Staff_Address	Staff_PhoneNumber	Position_Title	Salary	Branch_I
1	aa0098	Anand	Alexander	1919 Francisco Street	5557798714	Salesperson	36214	br0003
2	ab1235	Anisha	Bonner	5433 Jacinto Court	2812819545	Salesperson	37500	br0008
3	ab7561	Aminah	Bannister	7456 Happy Lane	4569842341	Salesperson	38500	br0008
4	ac9967	Armaan	Coombes	789 Hoopfer Way	1024556978	Salesperson	38568	br0007
5	ah0694	Adeel	Homer	985 Cabana Court	7871698412	Salesperson	38457	br0001
6	as4875	Amaya	Seymour	9952 Camino Rey Dr	1368495754	Salesperson	38789	br0001
7	at1965	Ana	Trevino	714 Greenhill Trail	2225684789	Salesperson	36950	br0004
8	aw3351	Areeb	Whitmore	1919 Gravel Court	9984751457	Finance Manager	67990	br0005
9	az7215	Arielle	Zuniga	1515 Greer Lane	5247873457	Salesperson	40000	br0005
10	bb8731	Britany	Brown	7874 Golf Lane	4447768647	Salesperson	39500	br0003
11	bh9124	Britney	Haines	7541 Gray Court	3369842456	Regional Director	103000	br0005
12	bm0247	Beth	Marsden	854 Isleworth Circle	4168941719	Salesperson	69854	br0007
13	ca9841	Carolyn	Arellano	654 Gibson Avenue	8413788463	Salesperson	37957	br0003
14	cb5547	Cloe	Burn	8946 Hibernia Lane	4116783147	Salesperson	36200	br0006
15	cb7014	Carla	Bob	4487 Gaucho Way	5557241164	Salesperson	37214	br0003
16	cc2214	Ceri	Cook	3354 Holton Avenue	4459873245	Salesperson	37895	br0006
17	cc9734	Caroline	Cooke	524 Heron Court	1514796258	Salesperson	39250	br0005
18	cf7514	Caleb	Flynn	123 Herman Avenue	1145876354	Salesperson	39100	br0005
19	cg3000	Cleo	Gay	841 Dill Lane	7181387864	Finance Manager	69400	br0002
20	ch2243	Chloe-Louise	Hodge	212 Ithaca Terrace	7134456289	Finance Manager	65000	br0008
21	cp0316	Corinne	Plummer	877 Garden Street	3334548721	Salesperson	37899	br0003
22	cw2525	Carla	Willis	111 Jubilee Court	3335869745	Salesperson	39500	br0008
23	dc1776	Dora	Conner	1012 Ely Place	9718461786	Salesperson	37484	br0002
24	dp4163	Dottie	Pacheco	8041 Hubbs Street	4748123564	Salesperson	39933	br0007
25	dr8764	Dwayne	Rice	123 Hershey Lane	5459861475	Salesperson	38500	br0005
26	ds1543	Dane	Sykes	2255 Grant Way	9852477536	Salesperson	37840	br0004
27	ef1245	Effie	Feeny	2222 Formosa St	1122234568	Salesperson	39974	br0002
28	ew5211	Efe	Watson	1145 Huey Lane	1447778896	Salesperson	39100	br0007
29	fh1173	Florrie	Humphreys	98732 Bayhill Dr	2135649879	Regional Director	101000	br0001
30	fj7954	Frank	Jarvis	13409 Cajon Court	9747316445	Salesperson	38752	br0001
31	fl7468	Flora	Laing	777 Inner Circle	8865477241	Salesperson	37550	br0007
32	gr3368	Gabriel	Robson	475 Joy Lane	3468757421	Salesperson	39000	br0008
33	gt4040	Georgia	Thorne	872 Diver Drive	3337675498	Salesperson	384754	br0002
34	hb2354	Helena	Bassett	791 Iron Oak Way	9845632564	Salesperson	38454	br0007
35	hl0048	Husna	Lucas	1232 Gale Lane	5559862841	Salesperson	36854	br0003
36	ht9514	Haleemah	Timms	23 Jordon Street	2347854986	Salesperson	38000	br0008
37	ic2288	Irfan	Castillo	9781 Great Falls Drive	2259874346	Salesperson	36200	br0004
38	ic9845	Inayah	Connelly	1487 Graves Place	4446878745	Dealership Man...	89630	br0005
39	if7841	Ibraheem	Figueroa	777 Jackpot Court	3468884579	Salesperson	37000	br0008
40	ij1027	Izaak	Jennings	9846 Grey Dove Lane	3134779648	Salesperson	38520	br0005
41	ja3364	Jannat	Adkins	996 Good Hope Way	9794648514	Salesperson	38792	br0004
42	jm6307	Jun	Millington	6431 Deptford Aven...	2224518798	Dealership Man...	89300	br0002
43	js3344	Joss	Snider	32 Irwin Path	1448451368	Salesperson	38210	br0007
44	kb3665	Kirsty	Blackwell	1025 Goose Creek	7785214978	Salesperson	39875	br0004
45	kb6472	Kain	Betts	225 Henry Loop	8119456123	Dealership Man...	88630	br0006
46	km3369	Kaitlan	Mcclure	1200 Jace Place	2813467138	Salesperson	36500	br0008
47	ks8822	Khadeeja	Savage	741 Goldberg Street	8685214653	Salesperson	38200	br0004
48	kt8668	Krisha	Taylor	1100 Gold Place	7785142365	Finance Manager	68584	br0004
49	lb0124	Laibah	Bryan	9985 Hood Place	8945661231	Regional Director	100500	br0007
50	lb6632	Leela	Baird	1213 Hilda Way	4117934783	Salesperson	36853	br0006
51	lk3875	Lynn	Kemp	77 Hoyos Court	6749344431	Finance Manager	62000	br0007
52	lm0014	Lilly-Mai	Mansell	946 Horvat Lane	4653247897	Salesperson	39865	br0006
53	ln3574	Luella	Peterson	234 Horseshoe Lane	4462147845	Salesperson	39300	br0006

Query executed successfully.

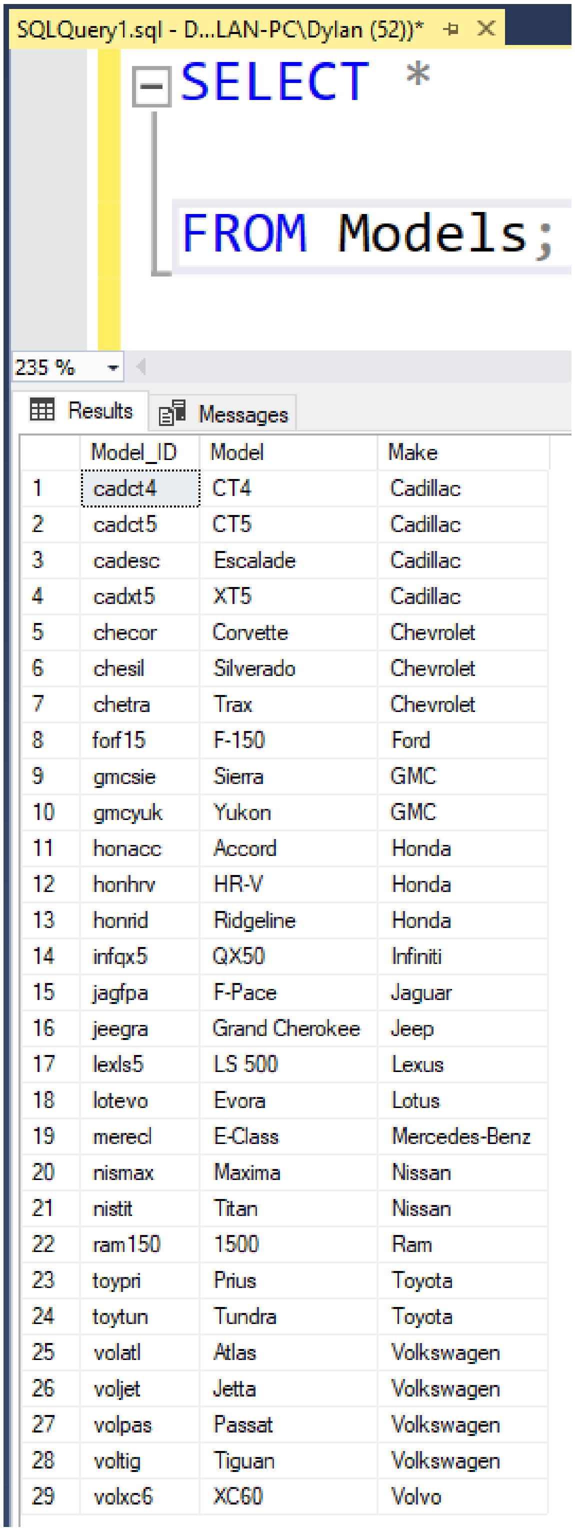
Query the Vehicles Table



```
SELECT *

FROM Vehicles;
```

	Vehicle_ID	Vehicle_Year	Vehicle_Trim	Vehicle_Price	Vehicle_Status	Branch_ID	Model_ID
1	cact3731	2021	Luxury	37315	leased	br0001	cadct4
2	cact4396	2021	Luxury	43965	rented	br0001	cadct5
3	cact4531	2021	Premium Luxury	45315	purchased	br0001	cadct5
4	caes1515	2018	Platinum	65995	purchased	br0006	cadesc
5	caxt5324	2021	Premium Luxury	53240	available	br0001	cadxt5
6	chco0543	2019	Stingray	67995	available	br0004	checor
7	chsi0493	2018	2500HD	59995	leased	br0005	chesil
8	chsi1058	2020	RST	44999	available	br0002	chesil
9	chsi1059	2019	LT	42995	available	br0002	chesil
10	chsi1496	2021	LT	42986	leased	br0002	chesil
11	chsi1842	2020	RST	53495	purchased	br0004	chesil
12	chsi3453	2019	1500	43995	rented	br0005	chesil
13	chsi7758	2020	1500	47595	rented	br0006	chesil
14	chtr1597	2020	LS	15623	available	br0001	chetra
15	fof12360	2020	XLT	42995	available	br0004	forf15
16	fof13920	2018	Lariat	43995	available	br0004	forf15
17	gmsi1464	2019	SLT	49985	leased	br0004	gmcsie
18	gmsi3365	2020	Denali	63995	purchased	br0003	gmcsie
19	gmyu9372	2019	Denali	60995	purchased	br0002	gmcyuk
20	hoac3312	2021	Sport	33125	available	br0001	honacc
21	hohr1734	2019	EX-L	27991	rented	br0002	honhrv
22	hori4364	2021	RTL-E	43645	available	br0001	honrid
23	inqx0955	2019	Essential	33995	purchased	br0005	infqx5
24	inqx2387	2019	Essential	37995	available	br0005	infqx5
25	jafp6222	2021	P250 S	62225	available	br0006	jagfpa
26	jegr1010	2020	Overland	44995	available	br0003	jeegra
27	lels0067	2018	F Sport	61991	leased	br0003	lexls5
28	loev9999	2020	GT	99998	available	br0006	lotevo
29	meec7599	2018	300	41995	available	br0003	merecl
30	nima3497	2019	S	28815	leased	br0007	nismax
31	niti1648	2020	SV	40995	rented	br0007	nistit
32	ra150098	2019	Classic	33995	leased	br0006	ram150
33	topr2822	2022	Prime	28220	purchased	br0007	toypri
34	totu1542	2018	SR5	38995	rented	br0003	toytun
35	voat4380	2021	SE	43808	leased	br0008	volatl
36	voje2868	2021	GLI S	28689	available	br0008	voljet
37	vopa2920	2022	2.0T SE	29209	purchased	br0008	volpas
38	voti3109	2021	SE	31099	rented	br0008	voltig
39	voxc3352	2020	T6	41995	rented	br0004	volxc6

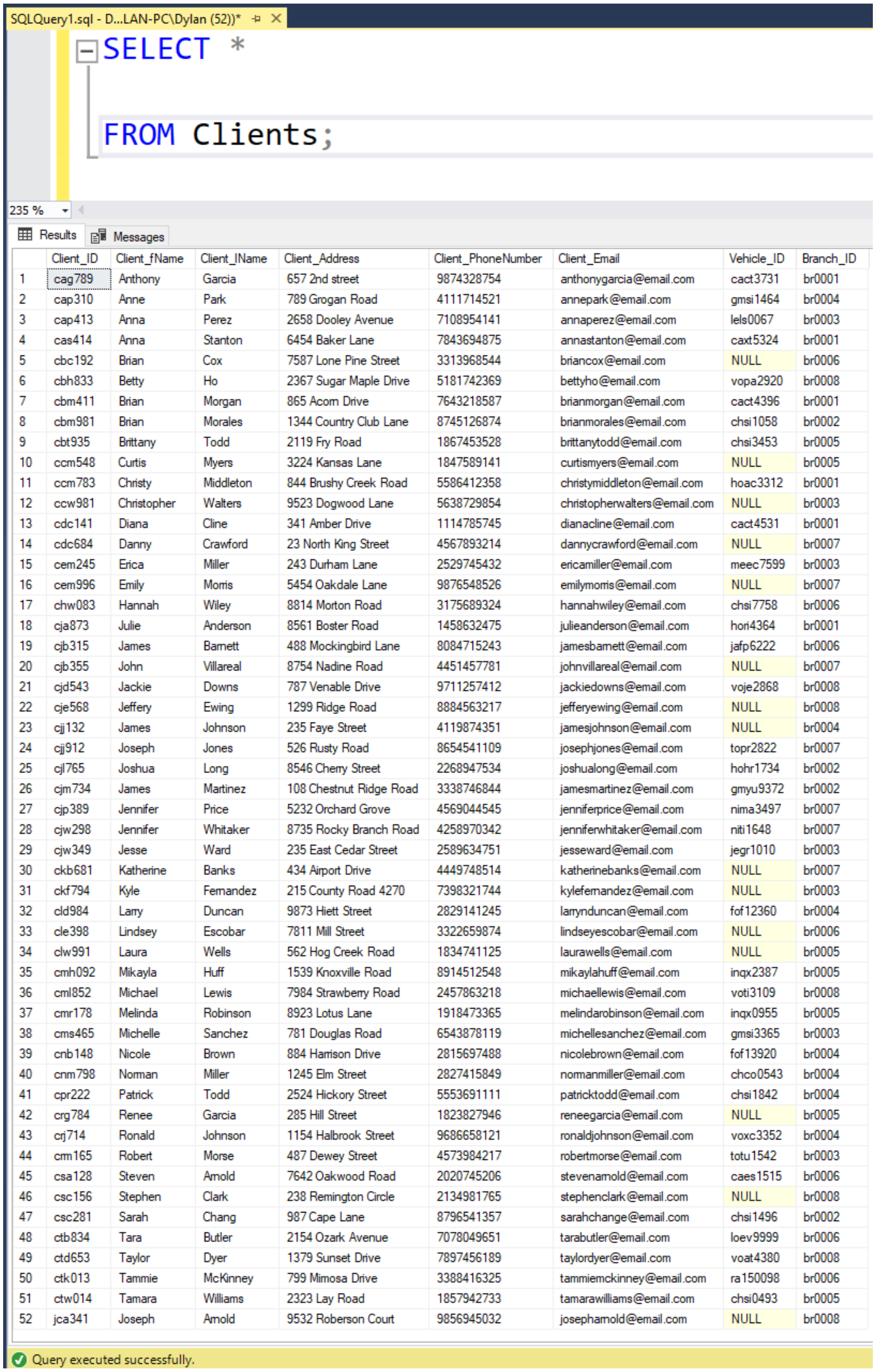
Query the Models Table



```
SELECT *

FROM Models;
```

	Model_ID	Model	Make
1	cadct4	CT4	Cadillac
2	cadct5	CT5	Cadillac
3	cadesc	Escalade	Cadillac
4	cadxt5	XT5	Cadillac
5	checor	Corvette	Chevrolet
6	chesil	Silverado	Chevrolet
7	chetra	Trax	Chevrolet
8	forf15	F-150	Ford
9	gmcsie	Sierra	GMC
10	gmcyuk	Yukon	GMC
11	honacc	Accord	Honda
12	honhrv	HR-V	Honda
13	honrid	Ridgeline	Honda
14	infqx5	QX50	Infiniti
15	jagfpa	F-Pace	Jaguar
16	jeegra	Grand Cherokee	Jeep
17	lexls5	LS 500	Lexus
18	lotevo	Evora	Lotus
19	merecl	E-Class	Mercedes-Benz
20	nismax	Maxima	Nissan
21	nistit	Titan	Nissan
22	ram150	1500	Ram
23	toypri	Prius	Toyota
24	toytun	Tundra	Toyota
25	volatl	Atlas	Volkswagen
26	voljet	Jetta	Volkswagen
27	volpas	Passat	Volkswagen
28	voltig	Tiguan	Volkswagen
29	volxc6	XC60	Volvo

Query the Clients Table

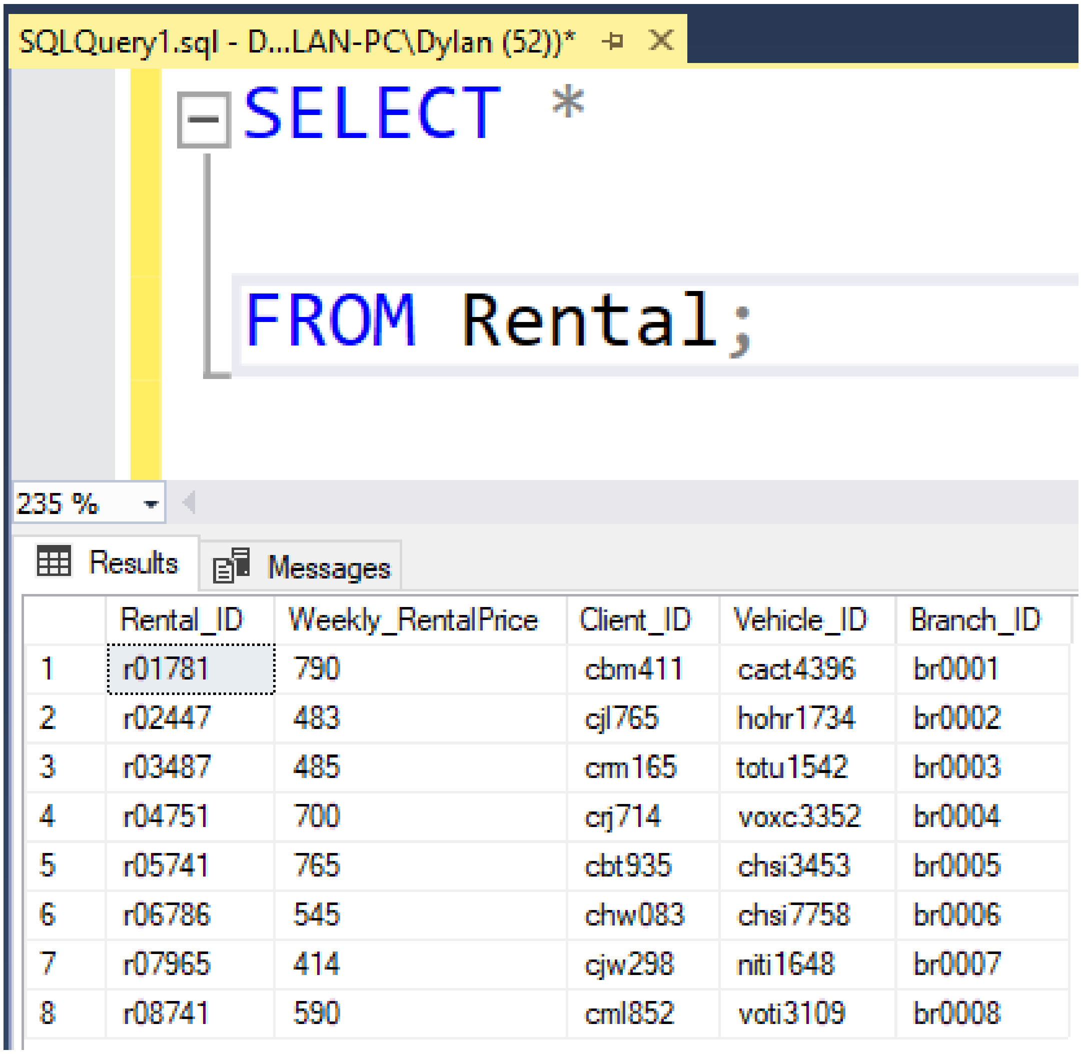


```
SELECT *

FROM Clients;
```

	Client_ID	Client_fName	Client_lName	Client_Address	Client_PhoneNumber	Client_Email	Vehicle_ID	Branch_ID
1	cag789	Anthony	Garcia	657 2nd street	9874328754	anthonygarcia@email.com	cact3731	br0001
2	cap310	Anne	Park	789 Grogan Road	4111714521	annepark@email.com	gmsi1464	br0004
3	cap413	Anna	Perez	2658 Dooley Avenue	7108954141	annaperez@email.com	lels0067	br0003
4	cas414	Anna	Stanton	6454 Baker Lane	7843694875	annastanton@email.com	caxt5324	br0001
5	cbc192	Brian	Cox	7587 Lone Pine Street	3313968544	briancox@email.com	NULL	br0006
6	cbh833	Betty	Ho	2367 Sugar Maple Drive	5181742369	bettyho@email.com	vopa2920	br0008
7	cbm411	Brian	Morgan	865 Acorn Drive	7643218587	brianmorgan@email.com	cact4396	br0001
8	cbm981	Brian	Morales	1344 Country Club Lane	8745126874	brianmorales@email.com	chsi1058	br0002
9	cbt935	Brittany	Todd	2119 Fry Road	1867453528	brittanytodd@email.com	chsi3453	br0005
10	ccm548	Curtis	Myers	3224 Kansas Lane	1847589141	curtismyers@email.com	NULL	br0005
11	ccm783	Christy	Middleton	844 Brushy Creek Road	5586412358	christymiddleton@email.com	hoac3312	br0001
12	ccw981	Christopher	Walters	9523 Dogwood Lane	5638729854	christopherwalters@email.com	NULL	br0003
13	cdc141	Diana	Cline	341 Amber Drive	1114785745	dianacline@email.com	cact4531	br0001
14	cdc684	Danny	Crawford	23 North King Street	4567893214	dannycrawford@email.com	NULL	br0007
15	cem245	Erica	Miller	243 Durham Lane	2529745432	ericamiller@email.com	meec7599	br0003
16	cem996	Emily	Morris	5454 Oakdale Lane	9876548526	emilymorris@email.com	NULL	br0007
17	chw083	Hannah	Wiley	8814 Morton Road	3175689324	hannahwiley@email.com	chsi7758	br0006
18	cja873	Julie	Anderson	8561 Boster Road	1458632475	julieanderson@email.com	hori4364	br0001
19	cjb315	James	Barnett	488 Mockingbird Lane	8084715243	jamesbarnett@email.com	jafp6222	br0006
20	cjb355	John	Villareal	8754 Nadine Road	4451457781	johnvillareal@email.com	NULL	br0007
21	cjd543	Jackie	Downs	787 Venable Drive	9711257412	jackiedowns@email.com	voje2868	br0008
22	cje568	Jeffery	Ewing	1299 Ridge Road	8884563217	jefferyewing@email.com	NULL	br0008
23	cjj132	James	Johnson	235 Faye Street	4119874351	jamesjohnson@email.com	NULL	br0004
24	cjj912	Joseph	Jones	526 Rusty Road	8654541109	josephjones@email.com	topr2822	br0007
25	cjl765	Joshua	Long	8546 Cherry Street	2268947534	joshualong@email.com	hohr1734	br0002
26	cjm734	James	Martinez	108 Chestnut Ridge Road	3338746844	jamesmartinez@email.com	gmyu9372	br0002
27	cjp389	Jennifer	Price	5232 Orchard Grove	4569044545	jenniferprice@email.com	nima3497	br0007
28	cjw298	Jennifer	Whitaker	8735 Rocky Branch Road	4258970342	jenniferwhitaker@email.com	niti1648	br0007
29	cjw349	Jesse	Ward	235 East Cedar Street	2589634751	jesseward@email.com	jegr1010	br0003
30	ckb681	Katherine	Banks	434 Airport Drive	4449748514	katherinebanks@email.com	NULL	br0007
31	ckf794	Kyle	Fernandez	215 County Road 4270	7398321744	kylefernandez@email.com	NULL	br0003
32	cld984	Larry	Duncan	9873 Hiett Street	2829141245	larryduncan@email.com	fof12360	br0004
33	cle398	Lindsey	Escobar	7811 Mill Street	3322659874	lindseyescobar@email.com	NULL	br0006
34	clw991	Laura	Wells	562 Hog Creek Road	1834741125	laurawells@email.com	NULL	br0005
35	cmh092	Mikayla	Huff	1539 Knoxville Road	8914512548	mikaylahuff@email.com	inqx2387	br0005
36	cml852	Michael	Lewis	7984 Strawberry Road	2457863218	michaellewis@email.com	voti3109	br0008
37	cmr178	Melinda	Robinson	8923 Lotus Lane	1918473365	melindarobinson@email.com	inqx0955	br0005
38	cms465	Michelle	Sanchez	781 Douglas Road	6543878119	michellesanchez@email.com	gmsi3365	br0003
39	cnb148	Nicole	Brown	884 Harrison Drive	2815697488	nicolebrown@email.com	fof13920	br0004
40	cnm798	Norman	Miller	1245 Elm Street	2827415849	normanmiller@email.com	chco0543	br0004
41	cpr222	Patrick	Todd	2524 Hickory Street	5553691111	patricktodd@email.com	chsi1842	br0004
42	crg784	Renee	Garcia	285 Hill Street	1823827946	reneegarcia@email.com	NULL	br0005
43	crj714	Ronald	Johnson	1154 Halbrook Street	9686658121	ronaldjohnson@email.com	voxc3352	br0004
44	crm165	Robert	Morse	487 Dewey Street	4573984217	robertmorse@email.com	totu1542	br0003
45	csa128	Steven	Arnold	7642 Oakwood Road	2020745206	stevenarnold@email.com	caes1515	br0006
46	csc156	Stephen	Clark	238 Remington Circle	2134981765	stephenclark@email.com	NULL	br0008
47	csc281	Sarah	Chang	987 Cape Lane	8796541357	sarahchange@email.com	chsi1496	br0002
48	ctb834	Tara	Butler	2154 Ozark Avenue	7078049651	tarabutler@email.com	loev9999	br0006
49	ctd653	Taylor	Dyer	1379 Sunset Drive	7897456189	taylordyer@email.com	voat4380	br0008
50	ctk013	Tammie	McKinney	799 Mimosa Drive	3388416325	tammiemckinney@email.com	ra150098	br0006
51	ctw014	Tamara	Williams	2323 Lay Road	1857942733	tamarawilliams@email.com	chsi0493	br0005
52	jca341	Joseph	Arnold	9532 Roberson Court	9856945032	josepharnold@email.com	NULL	br0008

✓ Query executed successfully.

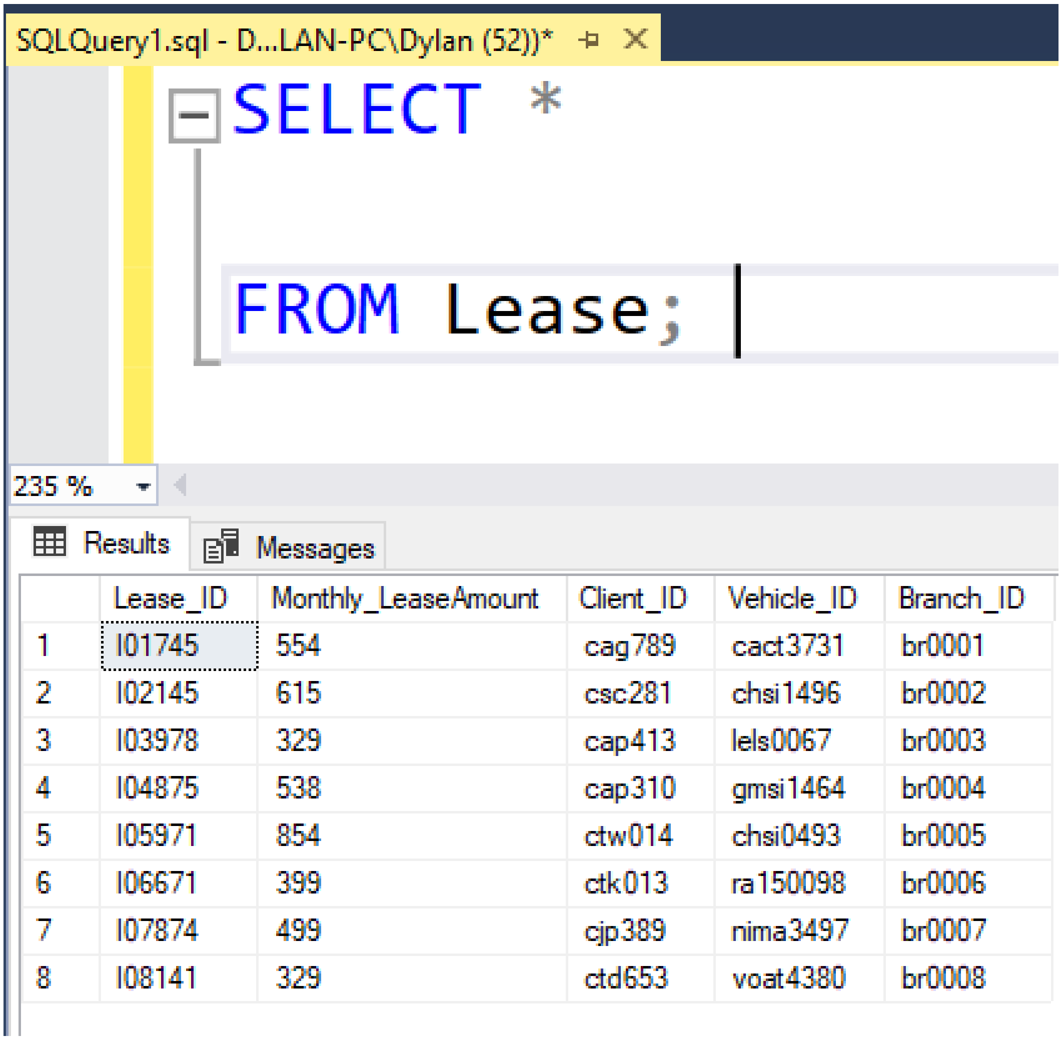
Query the Rental Table



```
SELECT *

FROM Rental;
```

	Rental_ID	Weekly_RentalPrice	Client_ID	Vehicle_ID	Branch_ID
1	r01781	790	cbm411	cact4396	br0001
2	r02447	483	cjl765	hohr1734	br0002
3	r03487	485	cm165	totu1542	br0003
4	r04751	700	crj714	voxc3352	br0004
5	r05741	765	cbt935	chsi3453	br0005
6	r06786	545	chw083	chsi7758	br0006
7	r07965	414	cjw298	niti1648	br0007
8	r08741	590	cml852	voti3109	br0008

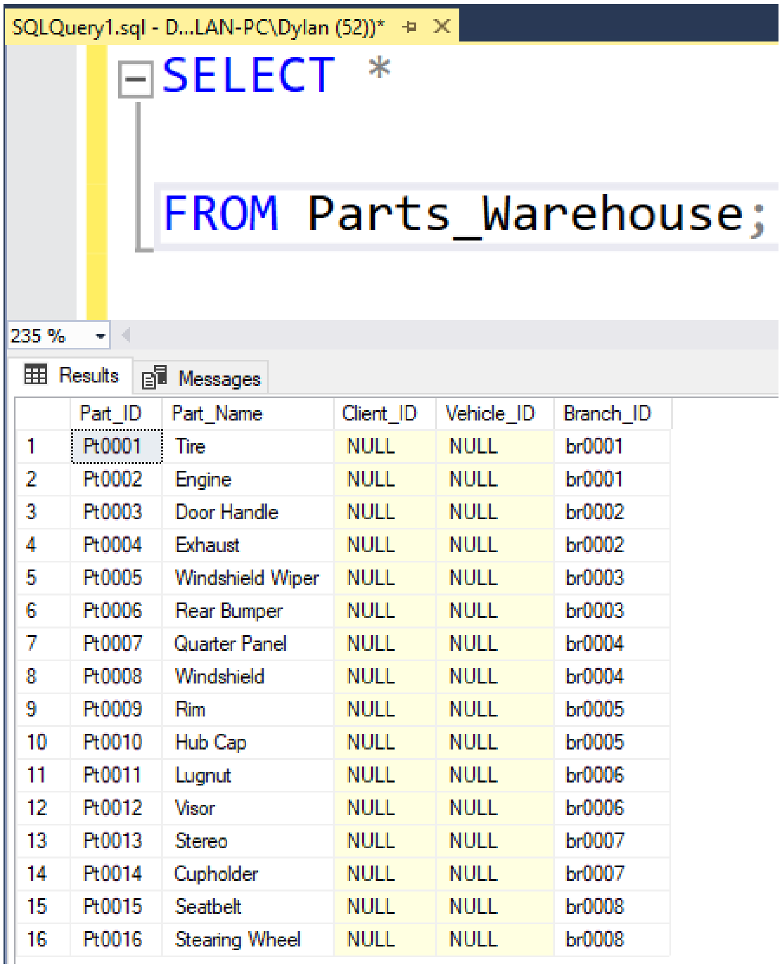
Query the Lease Table



```
SELECT *

FROM Lease;
```

	Lease_ID	Monthly_LeaseAmount	Client_ID	Vehicle_ID	Branch_ID
1	l01745	554	cag789	cact3731	br0001
2	l02145	615	csc281	chsi1496	br0002
3	l03978	329	cap413	lels0067	br0003
4	l04875	538	cap310	gmsi1464	br0004
5	l05971	854	ctw014	chsi0493	br0005
6	l06671	399	ctk013	ra150098	br0006
7	l07874	499	cjp389	nima3497	br0007
8	l08141	329	ctd653	voat4380	br0008

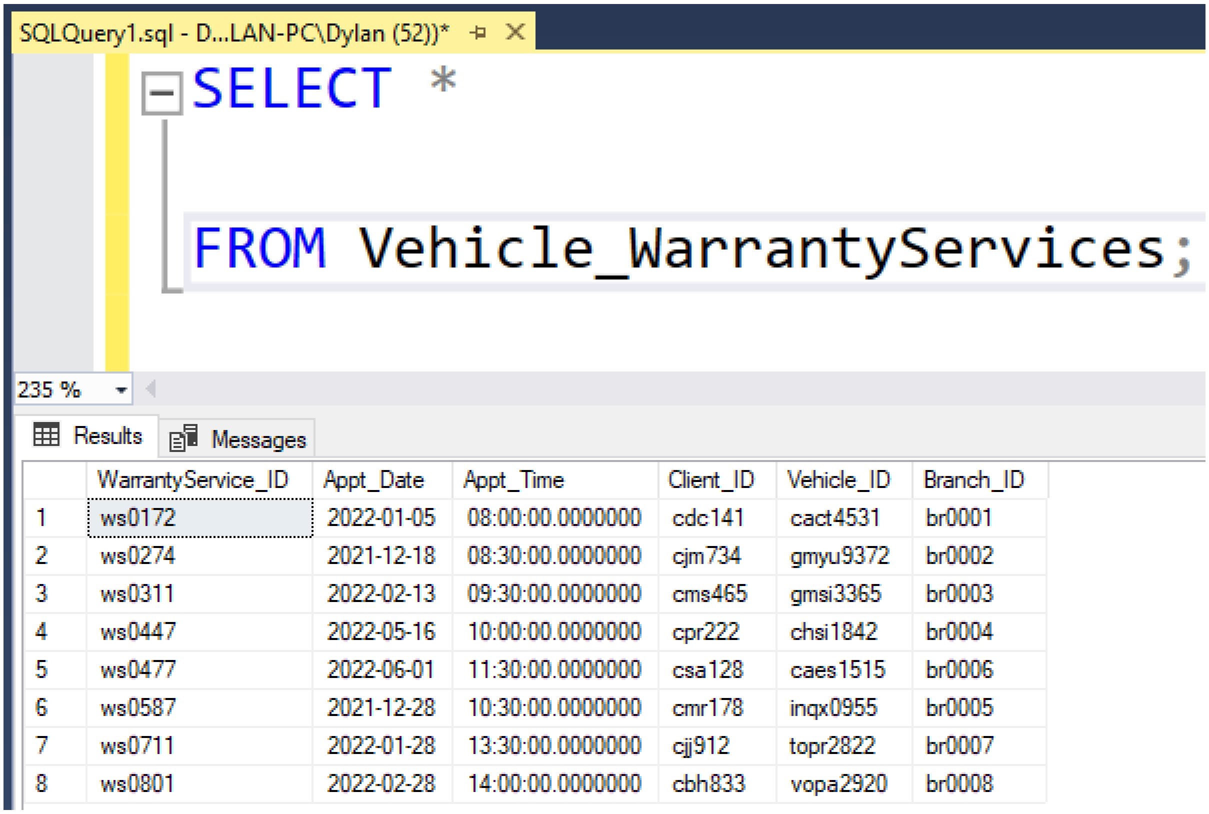
Query the Parts_Warehouse Table



```
SELECT *

FROM Parts_Warehouse;
```

	Part_ID	Part_Name	Client_ID	Vehicle_ID	Branch_ID
1	Pt0001	Tire	NULL	NULL	br0001
2	Pt0002	Engine	NULL	NULL	br0001
3	Pt0003	Door Handle	NULL	NULL	br0002
4	Pt0004	Exhaust	NULL	NULL	br0002
5	Pt0005	Windshield Wiper	NULL	NULL	br0003
6	Pt0006	Rear Bumper	NULL	NULL	br0003
7	Pt0007	Quarter Panel	NULL	NULL	br0004
8	Pt0008	Windshield	NULL	NULL	br0004
9	Pt0009	Rim	NULL	NULL	br0005
10	Pt0010	Hub Cap	NULL	NULL	br0005
11	Pt0011	Lugnut	NULL	NULL	br0006
12	Pt0012	Visor	NULL	NULL	br0006
13	Pt0013	Stereo	NULL	NULL	br0007
14	Pt0014	Cupholder	NULL	NULL	br0007
15	Pt0015	Seatbelt	NULL	NULL	br0008
16	Pt0016	Stearing Wheel	NULL	NULL	br0008

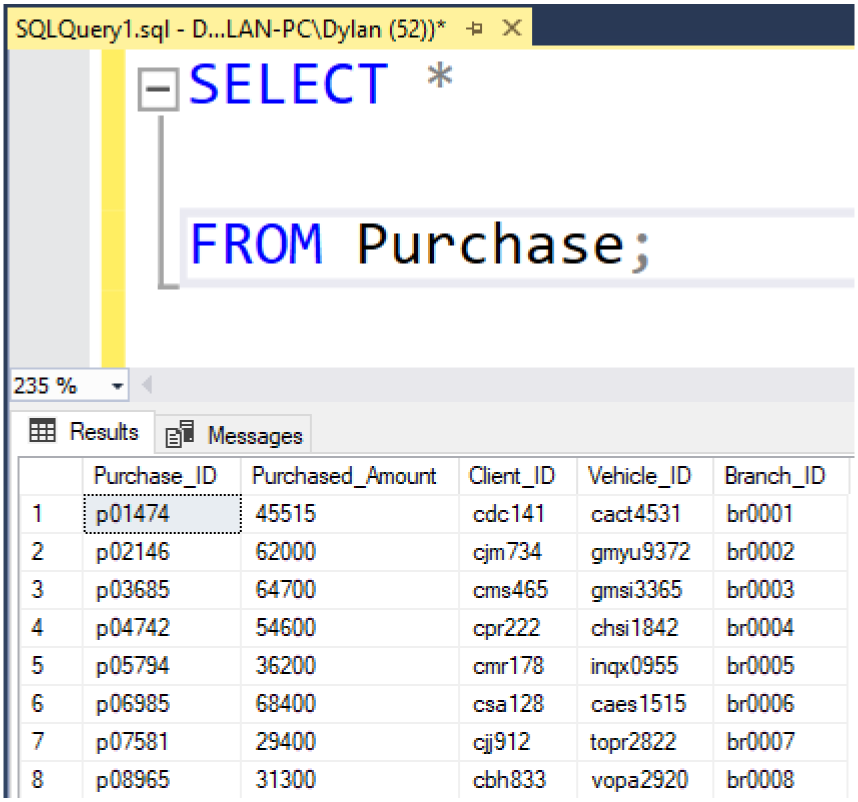
Query the Vehicle_WarrantyServices Table



```
SELECT *

FROM Vehicle_WarrantyServices;
```

	WarrantyService_ID	Appt_Date	Appt_Time	Client_ID	Vehicle_ID	Branch_ID
1	ws0172	2022-01-05	08:00:00.0000000	cdc141	cact4531	br0001
2	ws0274	2021-12-18	08:30:00.0000000	cjm734	gmyu9372	br0002
3	ws0311	2022-02-13	09:30:00.0000000	cms465	gmsi3365	br0003
4	ws0447	2022-05-16	10:00:00.0000000	cpr222	chsi1842	br0004
5	ws0477	2022-06-01	11:30:00.0000000	csa128	caes1515	br0006
6	ws0587	2021-12-28	10:30:00.0000000	cmr178	inqx0955	br0005
7	ws0711	2022-01-28	13:30:00.0000000	cjj912	topr2822	br0007
8	ws0801	2022-02-28	14:00:00.0000000	cbh833	vopa2920	br0008

Query the Purchase Table

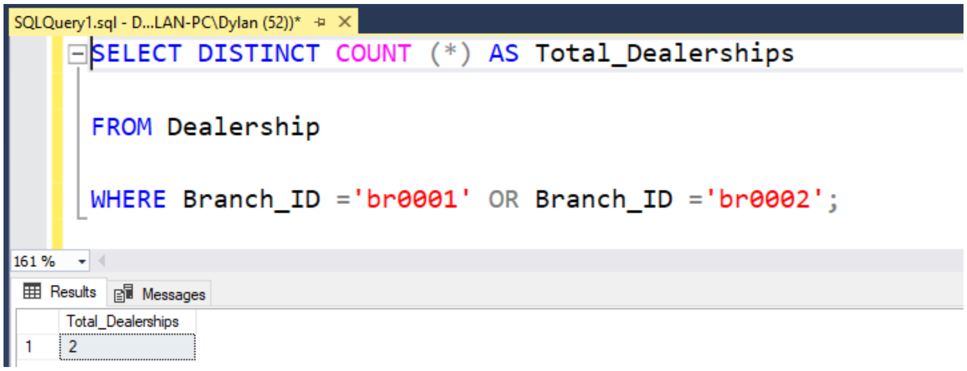


```
SELECT *

FROM Purchase;
```

	Purchase_ID	Purchased_Amount	Client_ID	Vehicle_ID	Branch_ID
1	p01474	45515	cdc141	cact4531	br0001
2	p02146	62000	cjm734	gmyu9372	br0002
3	p03685	64700	cms465	gmsi3365	br0003
4	p04742	54600	cpr222	chsi1842	br0004
5	p05794	36200	cmr178	inqx0955	br0005
6	p06985	68400	csa128	caes1515	br0006
7	p07581	29400	cjj912	topr2822	br0007
8	p08965	31300	cbh833	vopa2920	br0008

- AGGREGATE SQL STATEMENTS –

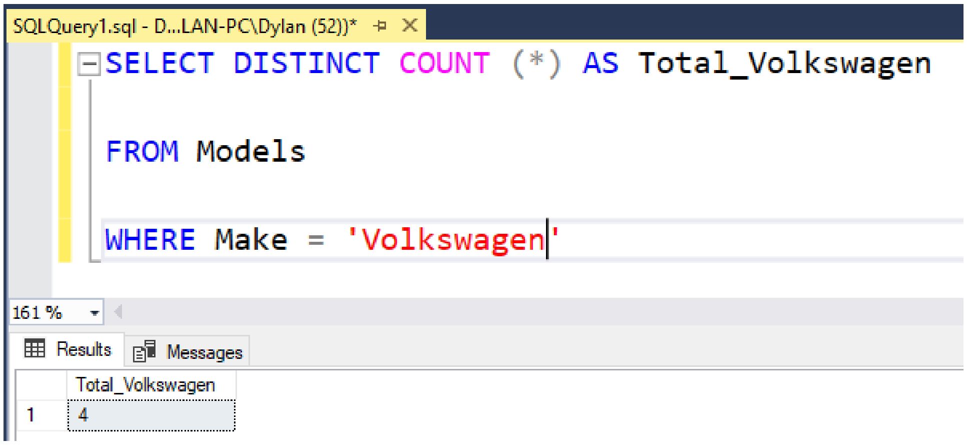
Aggregate count statement from Dealership Table



```sql
SELECT DISTINCT COUNT (*) AS Total_Dealerships

FROM Dealership

WHERE Branch_ID ='br0001' OR Branch_ID ='br0002';
```

	Total_Dealerships
1	2

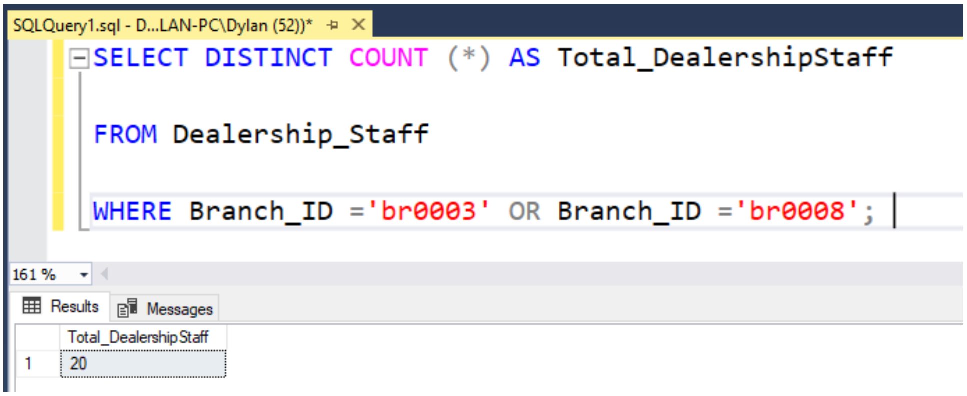
Aggregate count statement from Model Table



```sql
SELECT DISTINCT COUNT (*) AS Total_Volkswagen

FROM Models

WHERE Make = 'Volkswagen'
```

	Total_Volkswagen
1	4

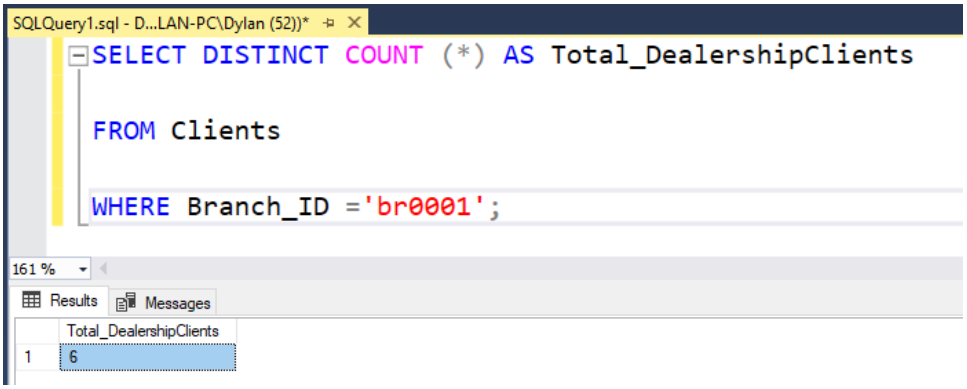
Aggregate count statement from Dealership_Staff Table



```
SELECT DISTINCT COUNT (*) AS Total_DealershipStaff

FROM Dealership_Staff

WHERE Branch_ID ='br0003' OR Branch_ID ='br0008';
```

	Total_DealershipStaff
1	20

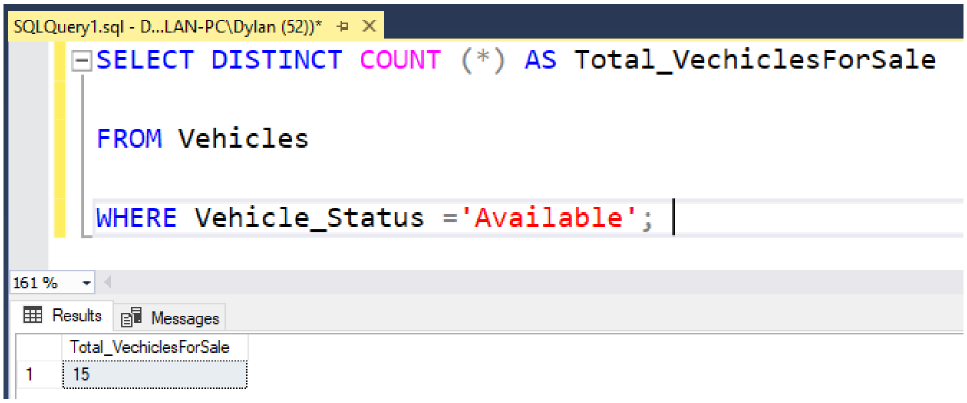
Aggregate count statement from Clients Table



```
SELECT DISTINCT COUNT (*) AS Total_DealershipClients

FROM Clients

WHERE Branch_ID ='br0001';
```

	Total_DealershipClients
1	6

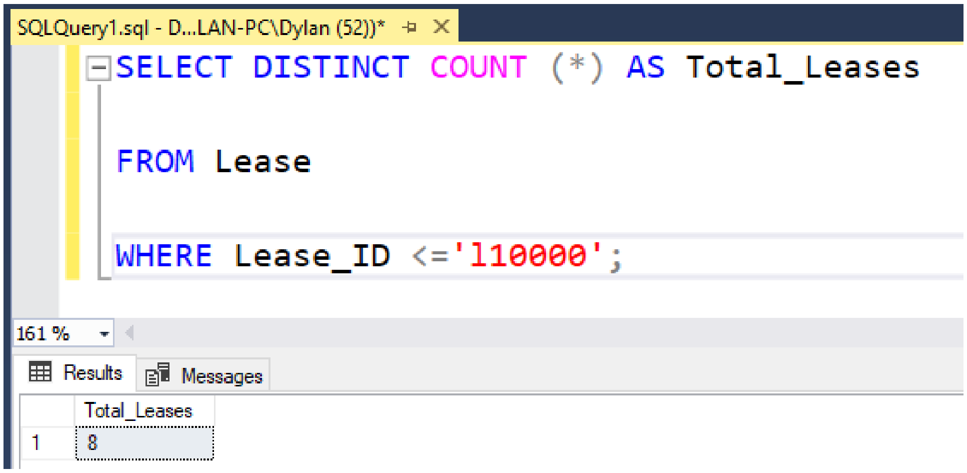
Aggregate count statement from Vehicles Table



```sql
SELECT DISTINCT COUNT (*) AS Total_VechiclesForSale

FROM Vehicles

WHERE Vehicle_Status ='Available';
```

	Total_VechiclesForSale
1	15

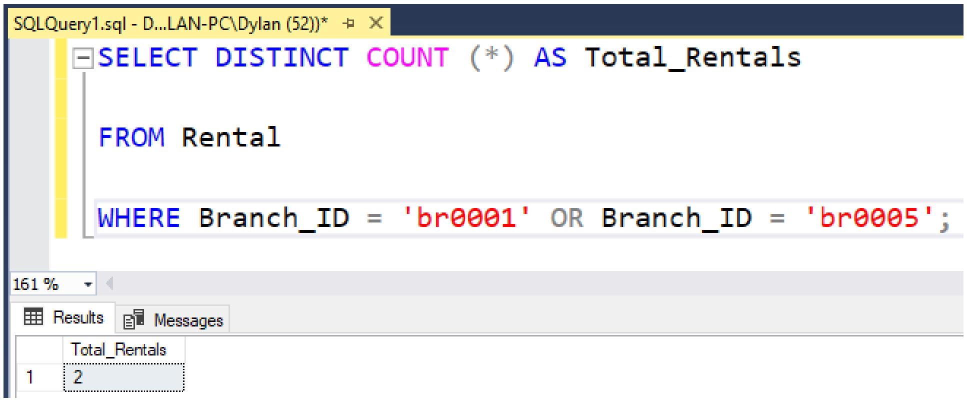
Aggregate count statement from Lease Table



```sql
SELECT DISTINCT COUNT (*) AS Total_Leases

FROM Lease

WHERE Lease_ID <='l10000';
```

	Total_Leases
1	8

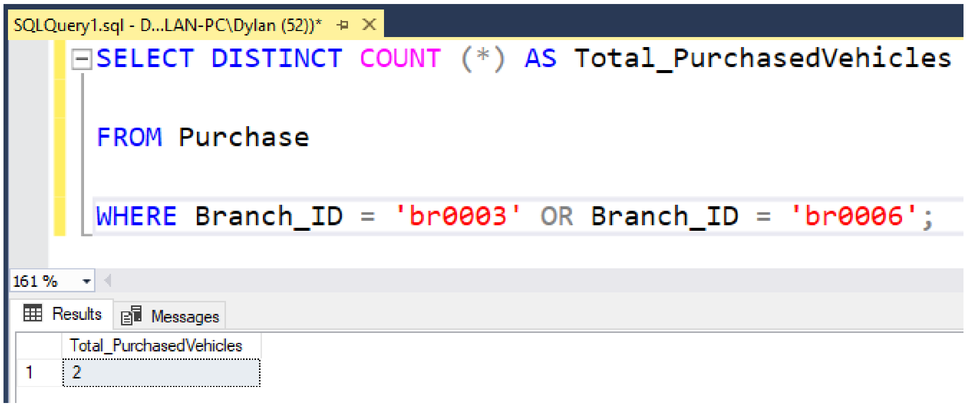
Aggregate count statement from Rental Table



```sql
SELECT DISTINCT COUNT (*) AS Total_Rentals

FROM Rental

WHERE Branch_ID = 'br0001' OR Branch_ID = 'br0005';
```

	Total_Rentals
1	2

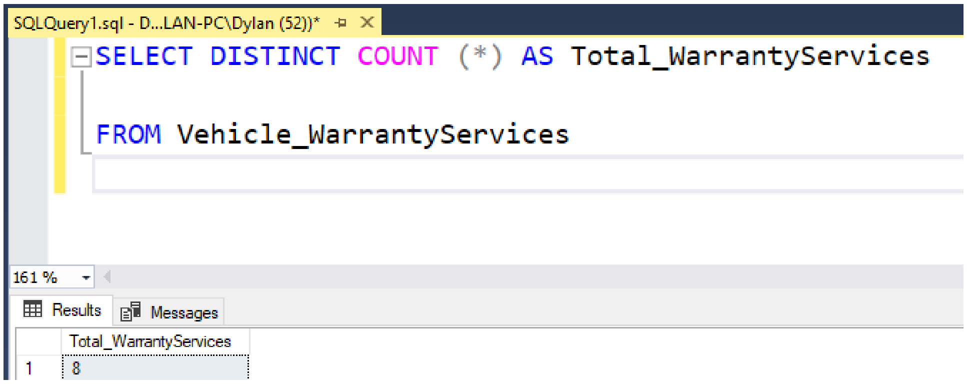
Aggregate count statement from Purchase Table



```sql
SELECT DISTINCT COUNT (*) AS Total_PurchasedVehicles

FROM Purchase

WHERE Branch_ID = 'br0003' OR Branch_ID = 'br0006';
```

	Total_PurchasedVehicles
1	2

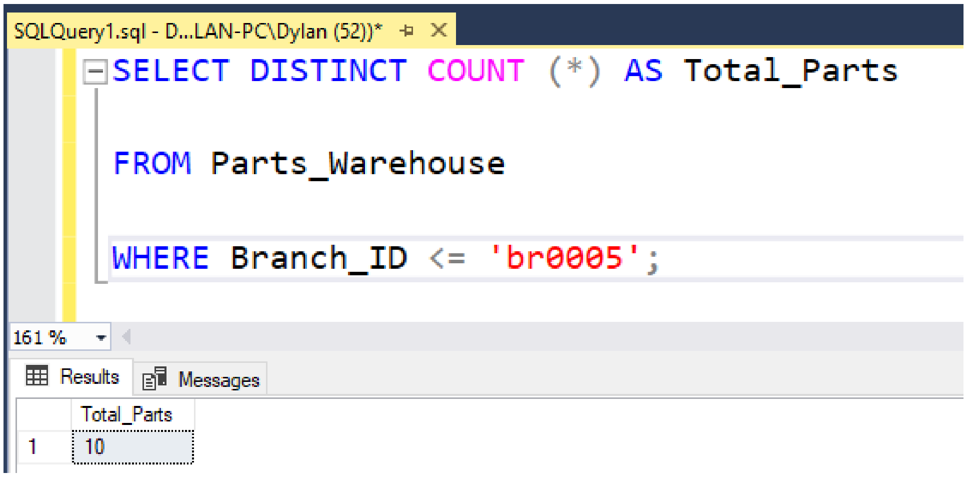
Aggregate count statement from Vehicle_WarrantyServices Table



```
SELECT DISTINCT COUNT (*) AS Total_WarrantyServices

FROM Vehicle_WarrantyServices
```

	Total_WarrantyServices
1	8

Aggregate count statement from Parts_Warehouse Table

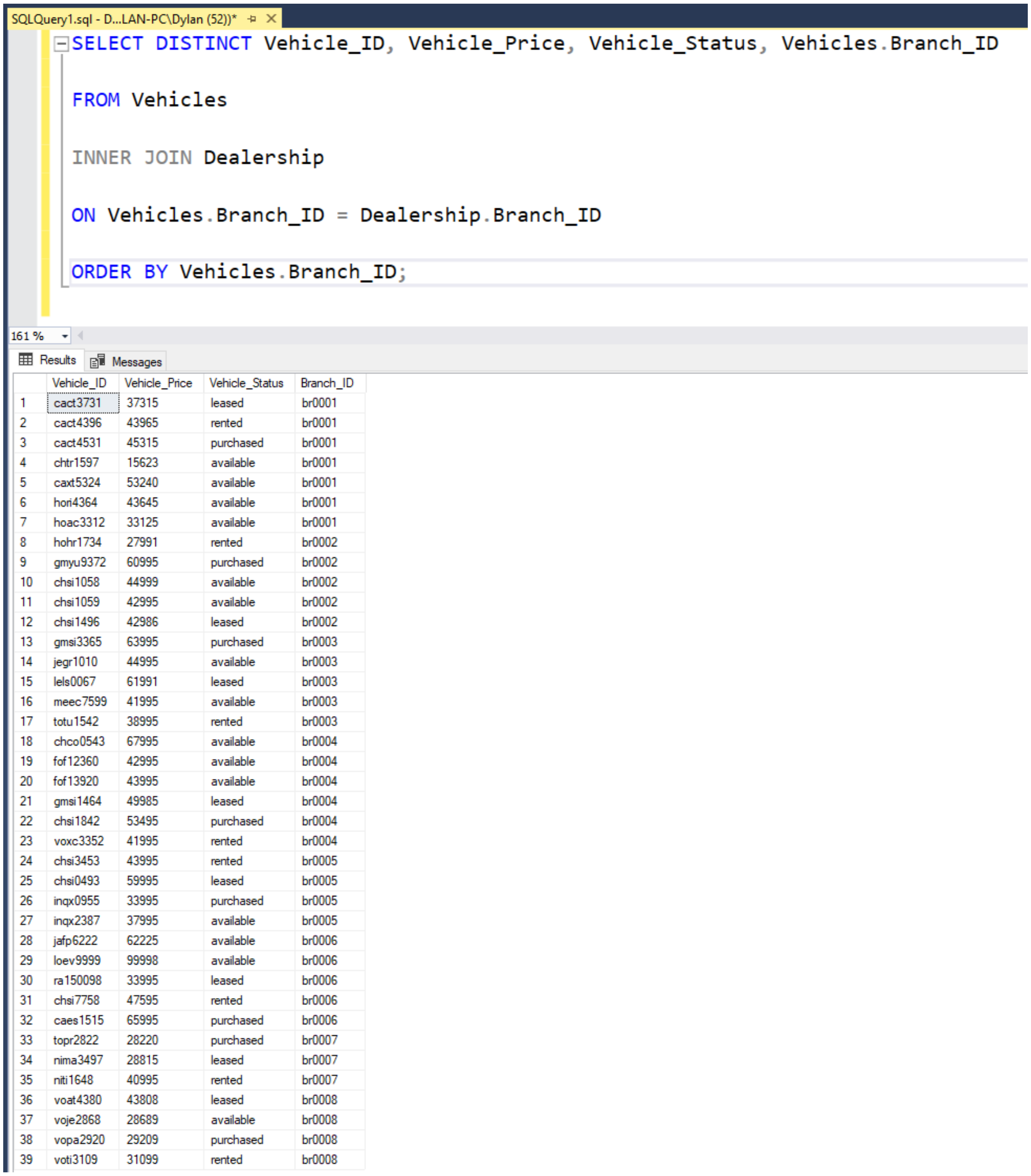


```
SELECT DISTINCT COUNT (*) AS Total_Parts

FROM Parts_Warehouse

WHERE Branch_ID <= 'br0005';
```

	Total_Parts
1	10

- JOINT SQL STATEMENTS –

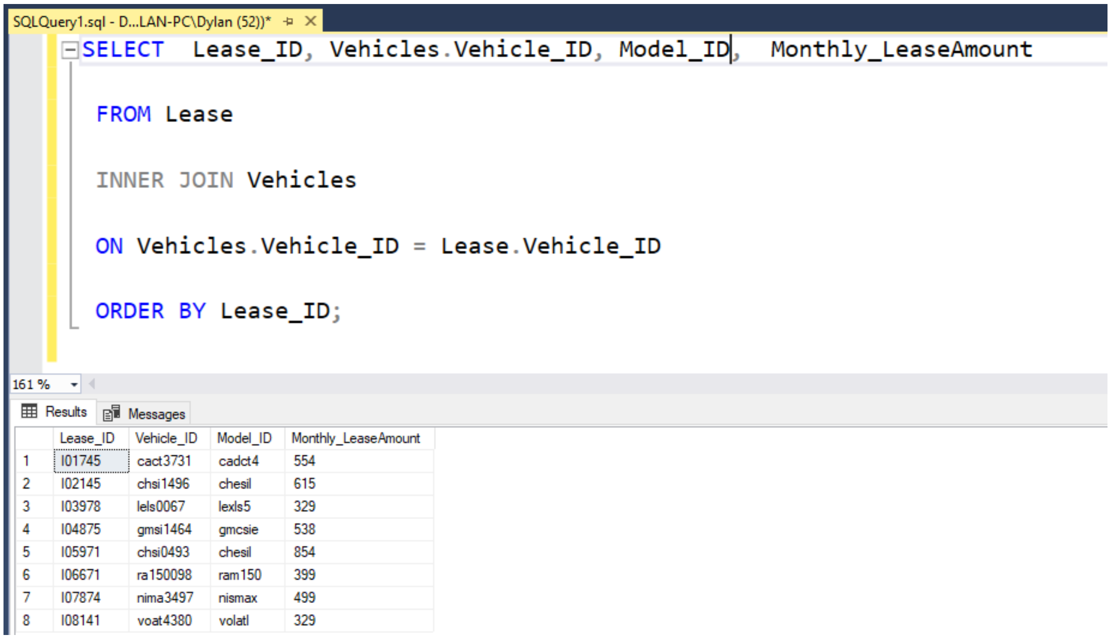
Joint query between Vehicles and Dealership Tables



```sql
SELECT DISTINCT Vehicle_ID, Vehicle_Price, Vehicle_Status, Vehicles.Branch_ID

FROM Vehicles

INNER JOIN Dealership

ON Vehicles.Branch_ID = Dealership.Branch_ID

ORDER BY Vehicles.Branch_ID;
```

	Vehicle_ID	Vehicle_Price	Vehicle_Status	Branch_ID
1	cact3731	37315	leased	br0001
2	cact4396	43965	rented	br0001
3	cact4531	45315	purchased	br0001
4	chtr1597	15623	available	br0001
5	caxt5324	53240	available	br0001
6	hori4364	43645	available	br0001
7	hoac3312	33125	available	br0001
8	hohr1734	27991	rented	br0002
9	gmyu9372	60995	purchased	br0002
10	chsi1058	44999	available	br0002
11	chsi1059	42995	available	br0002
12	chsi1496	42986	leased	br0002
13	gmsi3365	63995	purchased	br0003
14	jegr1010	44995	available	br0003
15	lels0067	61991	leased	br0003
16	meec7599	41995	available	br0003
17	totu1542	38995	rented	br0003
18	chco0543	67995	available	br0004
19	fof12360	42995	available	br0004
20	fof13920	43995	available	br0004
21	gmsi1464	49985	leased	br0004
22	chsi1842	53495	purchased	br0004
23	voxc3352	41995	rented	br0004
24	chsi3453	43995	rented	br0005
25	chsi0493	59995	leased	br0005
26	inqx0955	33995	purchased	br0005
27	inqx2387	37995	available	br0005
28	jafp6222	62225	available	br0006
29	loev9999	99998	available	br0006
30	ra150098	33995	leased	br0006
31	chsi7758	47595	rented	br0006
32	caes1515	65995	purchased	br0006
33	topr2822	28220	purchased	br0007
34	nima3497	28815	leased	br0007
35	niti1648	40995	rented	br0007
36	voat4380	43808	leased	br0008
37	voje2868	28689	available	br0008
38	vopa2920	29209	purchased	br0008
39	voti3109	31099	rented	br0008

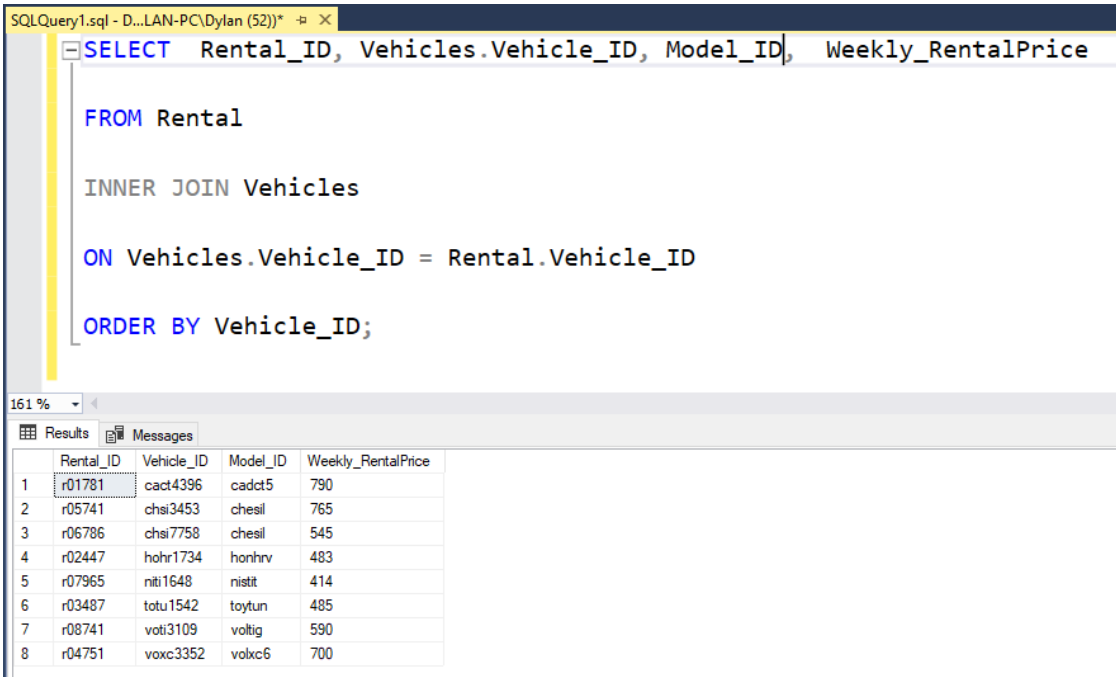
Joint query between Vehicles and Lease Tables



```
SELECT  Lease_ID, Vehicles.Vehicle_ID, Model_ID,  Monthly_LeaseAmount

 FROM Lease

 INNER JOIN Vehicles

 ON Vehicles.Vehicle_ID = Lease.Vehicle_ID

 ORDER BY Lease_ID;
```

	Lease_ID	Vehicle_ID	Model_ID	Monthly_LeaseAmount
1	l01745	cact3731	cadct4	554
2	l02145	chsi1496	chesil	615
3	l03978	lels0067	lexls5	329
4	l04875	gmsi1464	gmcsie	538
5	l05971	chsi0493	chesil	854
6	l06671	ra150098	ram150	399
7	l07874	nima3497	nismax	499
8	l08141	voat4380	volatl	329

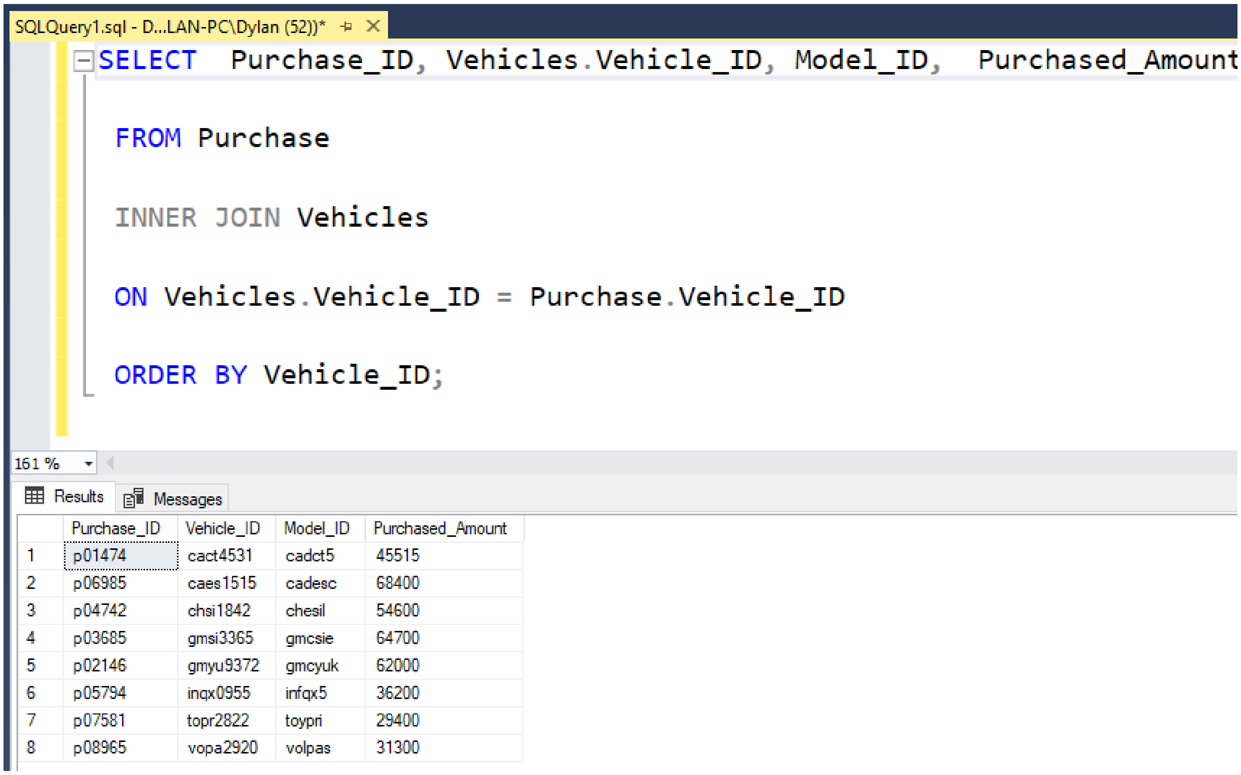
Joint query between Vehicles and Rental Tables



```
SELECT  Rental_ID, Vehicles.Vehicle_ID, Model_ID,  Weekly_RentalPrice

FROM Rental

INNER JOIN Vehicles

ON Vehicles.Vehicle_ID = Rental.Vehicle_ID

ORDER BY Vehicle_ID;
```

	Rental_ID	Vehicle_ID	Model_ID	Weekly_RentalPrice
1	r01781	cact4396	cadct5	790
2	r05741	chsi3453	chesil	765
3	r06786	chsi7758	chesil	545
4	r02447	hohr1734	honhrv	483
5	r07965	niti1648	nistit	414
6	r03487	totu1542	toytun	485
7	r08741	voti3109	voltig	590
8	r04751	voxc3352	volxc6	700

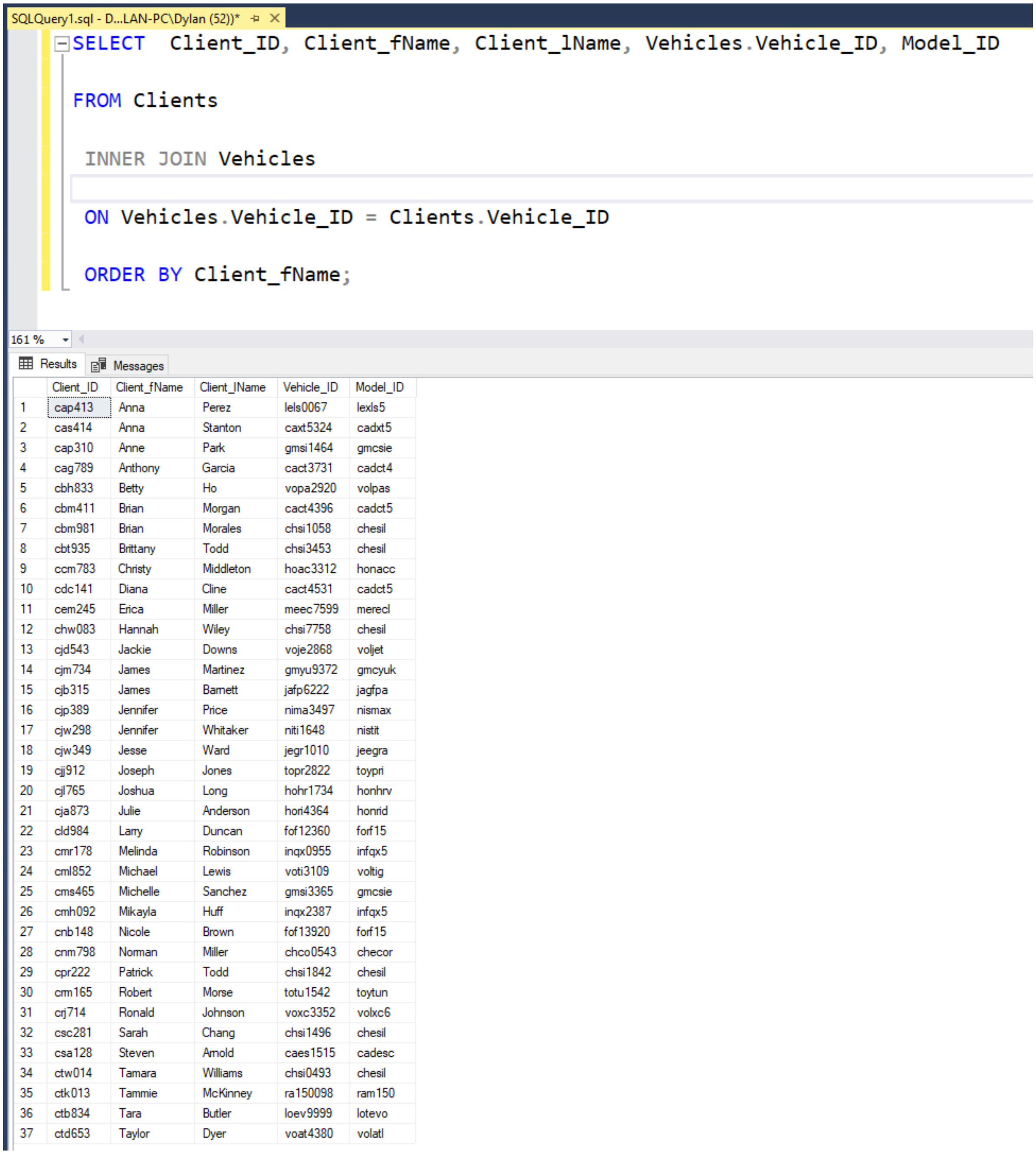
Joint query between Vehicles and Purchase Tables



```
SELECT  Purchase_ID, Vehicles.Vehicle_ID, Model_ID,  Purchased_Amount

 FROM Purchase

 INNER JOIN Vehicles

 ON Vehicles.Vehicle_ID = Purchase.Vehicle_ID

 ORDER BY Vehicle_ID;
```

	Purchase_ID	Vehicle_ID	Model_ID	Purchased_Amount
1	p01474	cact4531	cadct5	45515
2	p06985	caes1515	cadesc	68400
3	p04742	chsi1842	chesil	54600
4	p03685	gmsi3365	gmcsie	64700
5	p02146	gmyu9372	gmcyuk	62000
6	p05794	inqx0955	infqx5	36200
7	p07581	topr2822	toypri	29400
8	p08965	vopa2920	volpas	31300

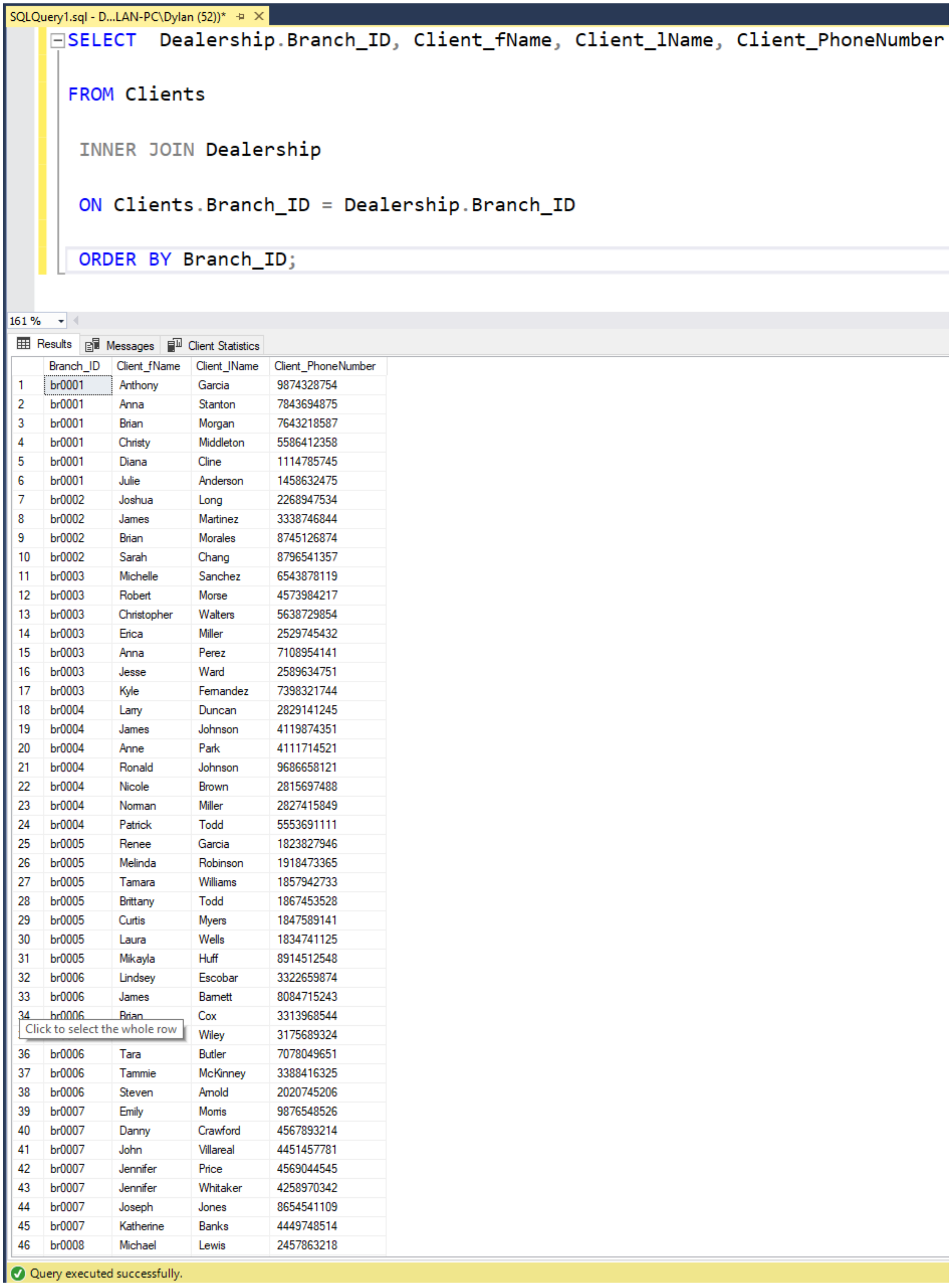
Joint query between Vehicles and Clients Tables



```sql
SELECT  Client_ID, Client_fName, Client_lName, Vehicles.Vehicle_ID, Model_ID

FROM Clients

 INNER JOIN Vehicles

 ON Vehicles.Vehicle_ID = Clients.Vehicle_ID

 ORDER BY Client_fName;
```

	Client_ID	Client_fName	Client_lName	Vehicle_ID	Model_ID
1	cap413	Anna	Perez	lels0067	lexls5
2	cas414	Anna	Stanton	caxt5324	cadxt5
3	cap310	Anne	Park	gmsi1464	gmcsie
4	cag789	Anthony	Garcia	cact3731	cadct4
5	cbh833	Betty	Ho	vopa2920	volpas
6	cbm411	Brian	Morgan	cact4396	cadct5
7	cbm981	Brian	Morales	chsi1058	chesil
8	cbt935	Brittany	Todd	chsi3453	chesil
9	ccm783	Christy	Middleton	hoac3312	honacc
10	cdc141	Diana	Cline	cact4531	cadct5
11	cem245	Erica	Miller	meec7599	merecl
12	chw083	Hannah	Wiley	chsi7758	chesil
13	cjd543	Jackie	Downs	voje2868	voljet
14	cjm734	James	Martinez	gmyu9372	gmcyuk
15	cjb315	James	Barnett	jafp6222	jagfpa
16	cjp389	Jennifer	Price	nima3497	nismax
17	cjw298	Jennifer	Whitaker	niti1648	nistit
18	cjw349	Jesse	Ward	jegr1010	jeegra
19	cjj912	Joseph	Jones	topr2822	toypri
20	cjl765	Joshua	Long	hohr1734	honhrv
21	cja873	Julie	Anderson	hori4364	honrid
22	cld984	Larry	Duncan	fof12360	forf15
23	cmr178	Melinda	Robinson	inqx0955	infqx5
24	cml852	Michael	Lewis	voti3109	voltig
25	cms465	Michelle	Sanchez	gmsi3365	gmcsie
26	cmh092	Mikayla	Huff	inqx2387	infqx5
27	cnb148	Nicole	Brown	fof13920	forf15
28	cnm798	Norman	Miller	chco0543	checor
29	cpr222	Patrick	Todd	chsi1842	chesil
30	crm165	Robert	Morse	totu1542	toytun
31	crj714	Ronald	Johnson	voxc3352	volxc6
32	csc281	Sarah	Chang	chsi1496	chesil
33	csa128	Steven	Arnold	caes1515	cadesc
34	ctw014	Tamara	Williams	chsi0493	chesil
35	ctk013	Tammie	McKinney	ra150098	ram150
36	ctb834	Tara	Butler	loev9999	lotevo
37	ctd653	Taylor	Dyer	voat4380	volatl

Joint query between Clients and Dealership Tables

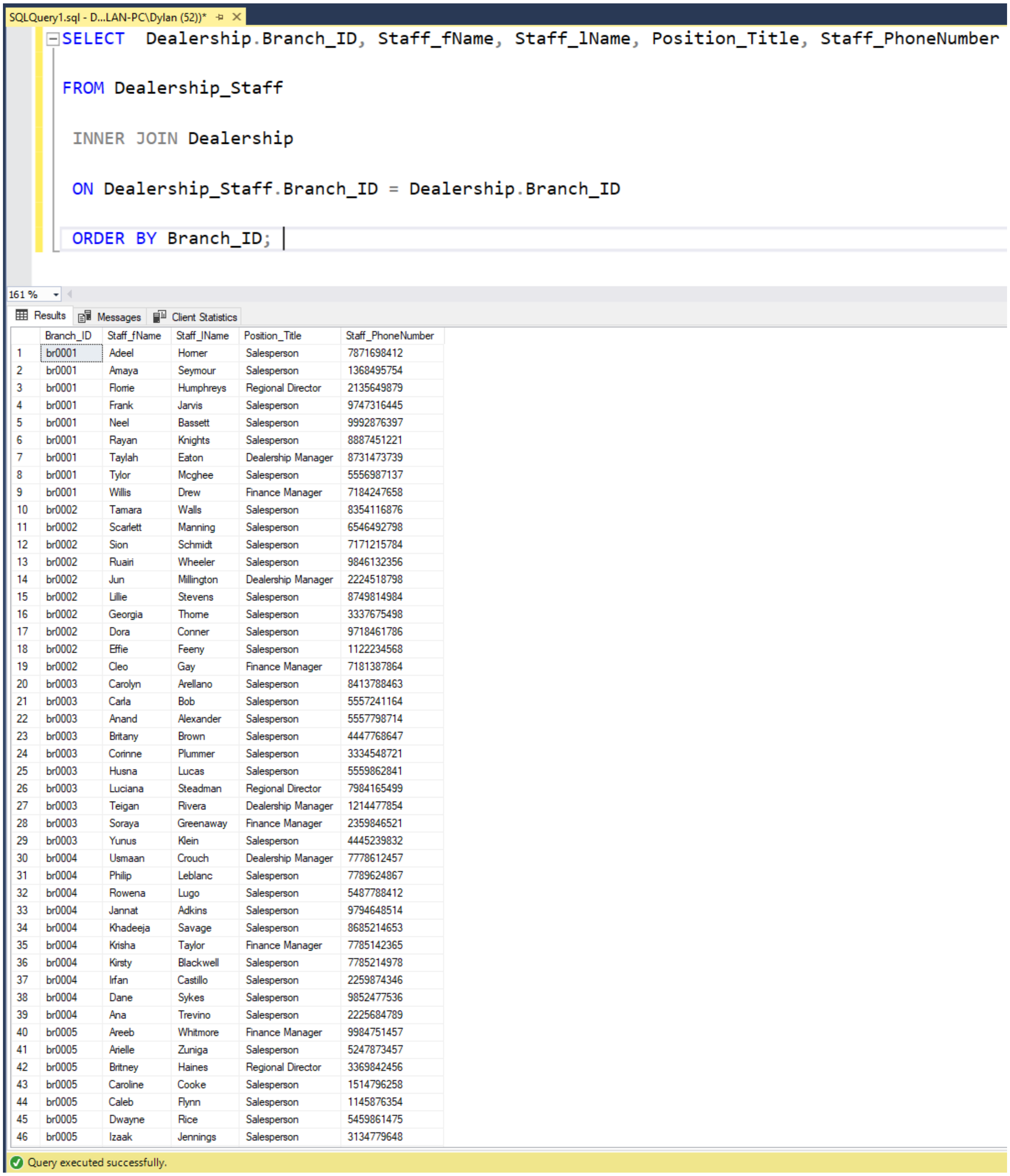

SQLQuery1.sql - D...LAN-PC\Dylan (52))*
161 %
Results
Messages
Client Statistics
Click to select the whole row

```
SELECT  Dealership.Branch_ID, Client_fName, Client_lName, Client_PhoneNumber

FROM Clients

 INNER JOIN Dealership

 ON Clients.Branch_ID = Dealership.Branch_ID

 ORDER BY Branch_ID;
```

	Branch_ID	Client_fName	Client_lName	Client_PhoneNumber
1	br0001	Anthony	Garcia	9874328754
2	br0001	Anna	Stanton	7843694875
3	br0001	Brian	Morgan	7643218587
4	br0001	Christy	Middleton	5586412358
5	br0001	Diana	Cline	1114785745
6	br0001	Julie	Anderson	1458632475
7	br0002	Joshua	Long	2268947534
8	br0002	James	Martinez	3338746844
9	br0002	Brian	Morales	8745126874
10	br0002	Sarah	Chang	8796541357
11	br0003	Michelle	Sanchez	6543878119
12	br0003	Robert	Morse	4573984217
13	br0003	Christopher	Walters	5638729854
14	br0003	Erica	Miller	2529745432
15	br0003	Anna	Perez	7108954141
16	br0003	Jesse	Ward	2589634751
17	br0003	Kyle	Fernandez	7398321744
18	br0004	Larry	Duncan	2829141245
19	br0004	James	Johnson	4119874351
20	br0004	Anne	Park	4111714521
21	br0004	Ronald	Johnson	9686658121
22	br0004	Nicole	Brown	2815697488
23	br0004	Norman	Miller	2827415849
24	br0004	Patrick	Todd	5553691111
25	br0005	Renee	Garcia	1823827946
26	br0005	Melinda	Robinson	1918473365
27	br0005	Tamara	Williams	1857942733
28	br0005	Brittany	Todd	1867453528
29	br0005	Curtis	Myers	1847589141
30	br0005	Laura	Wells	1834741125
31	br0005	Mikayla	Huff	8914512548
32	br0006	Lindsey	Escobar	3322659874
33	br0006	James	Barnett	8084715243
34	br0006	Brian	Cox	3313968544
[illegible]	[illegible]	[illegible]	Wiley	3175689324
36	br0006	Tara	Butler	7078049651
37	br0006	Tammie	McKinney	3388416325
38	br0006	Steven	Arnold	2020745206
39	br0007	Emily	Morris	9876548526
40	br0007	Danny	Crawford	4567893214
41	br0007	John	Villareal	4451457781
42	br0007	Jennifer	Price	4569044545
43	br0007	Jennifer	Whitaker	4258970342
44	br0007	Joseph	Jones	8654541109
45	br0007	Katherine	Banks	4449748514
46	br0008	Michael	Lewis	2457863218

✓ Query executed successfully.

Joint query between Dealership_Staff and Dealership Tables

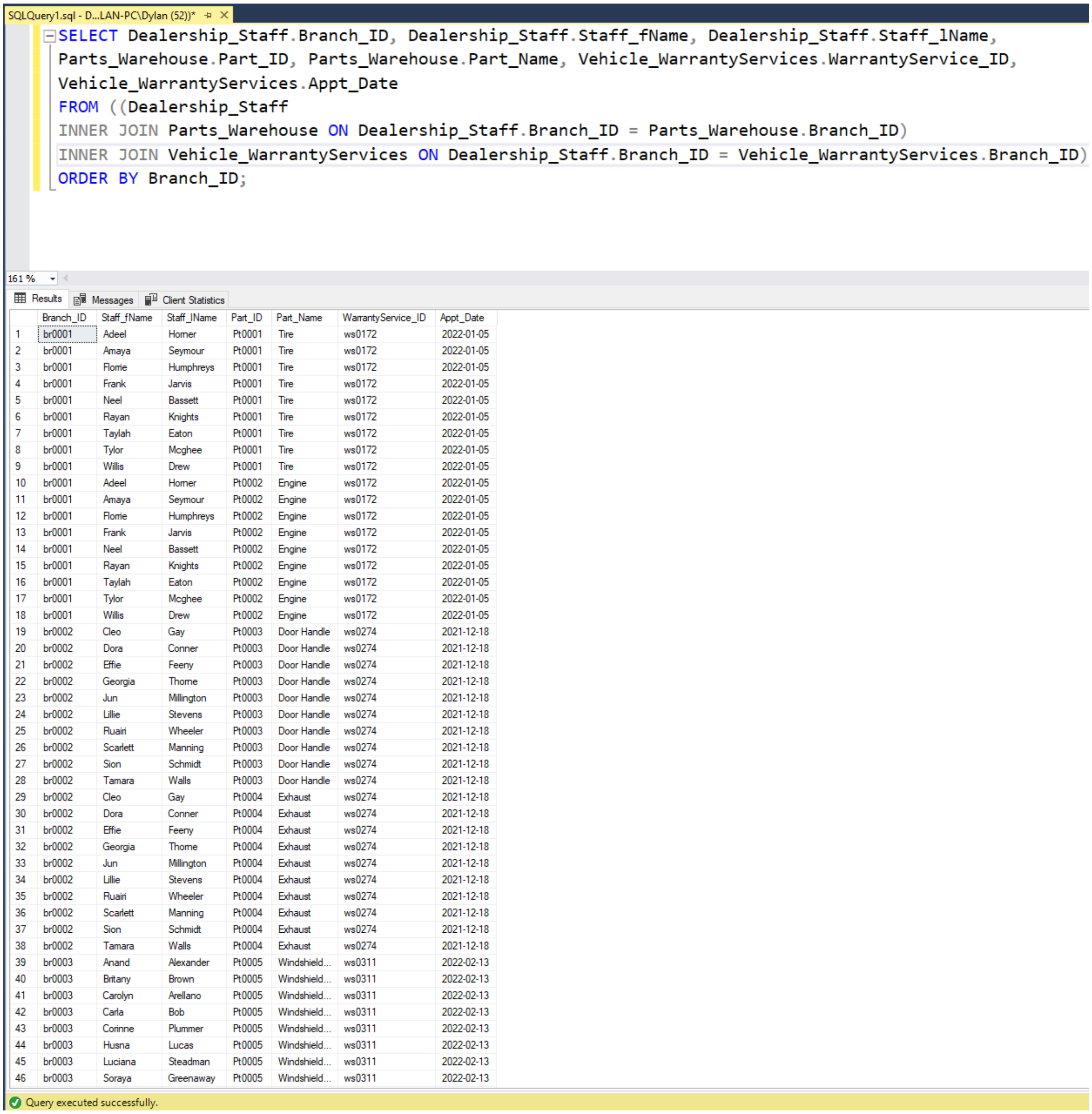


```sql
SELECT  Dealership.Branch_ID, Staff_fName, Staff_lName, Position_Title, Staff_PhoneNumber

FROM Dealership_Staff

 INNER JOIN Dealership

 ON Dealership_Staff.Branch_ID = Dealership.Branch_ID

 ORDER BY Branch_ID;
```

	Branch_ID	Staff_fName	Staff_lName	Position_Title	Staff_PhoneNumber
1	br0001	Adeel	Homer	Salesperson	7871698412
2	br0001	Amaya	Seymour	Salesperson	1368495754
3	br0001	Florrie	Humphreys	Regional Director	2135649879
4	br0001	Frank	Jarvis	Salesperson	9747316445
5	br0001	Neel	Bassett	Salesperson	9992876397
6	br0001	Rayan	Knights	Salesperson	8887451221
7	br0001	Taylah	Eaton	Dealership Manager	8731473739
8	br0001	Tylor	Mcghee	Salesperson	5556987137
9	br0001	Willis	Drew	Finance Manager	7184247658
10	br0002	Tamara	Walls	Salesperson	8354116876
11	br0002	Scarlett	Manning	Salesperson	6546492798
12	br0002	Sion	Schmidt	Salesperson	7171215784
13	br0002	Ruairi	Wheeler	Salesperson	9846132356
14	br0002	Jun	Millington	Dealership Manager	2224518798
15	br0002	Lillie	Stevens	Salesperson	8749814984
16	br0002	Georgia	Thorne	Salesperson	3337675498
17	br0002	Dora	Conner	Salesperson	9718461786
18	br0002	Effie	Feeny	Salesperson	1122234568
19	br0002	Cleo	Gay	Finance Manager	7181387864
20	br0003	Carolyn	Arellano	Salesperson	8413788463
21	br0003	Carla	Bob	Salesperson	5557241164
22	br0003	Anand	Alexander	Salesperson	5557798714
23	br0003	Britany	Brown	Salesperson	4447768647
24	br0003	Corinne	Plummer	Salesperson	3334548721
25	br0003	Husna	Lucas	Salesperson	5559862841
26	br0003	Luciana	Steadman	Regional Director	7984165499
27	br0003	Teigan	Rivera	Dealership Manager	1214477854
28	br0003	Soraya	Greenaway	Finance Manager	2359846521
29	br0003	Yunus	Klein	Salesperson	4445239832
30	br0004	Usmaan	Crouch	Dealership Manager	7778612457
31	br0004	Philip	Leblanc	Salesperson	7789624867
32	br0004	Rowena	Lugo	Salesperson	5487788412
33	br0004	Jannat	Adkins	Salesperson	9794648514
34	br0004	Khadeeja	Savage	Salesperson	8685214653
35	br0004	Krisha	Taylor	Finance Manager	7785142365
36	br0004	Kirsty	Blackwell	Salesperson	7785214978
37	br0004	Irfan	Castillo	Salesperson	2259874346
38	br0004	Dane	Sykes	Salesperson	9852477536
39	br0004	Ana	Trevino	Salesperson	2225684789
40	br0005	Areeb	Whitmore	Finance Manager	9984751457
41	br0005	Arielle	Zuniga	Salesperson	5247873457
42	br0005	Britney	Haines	Regional Director	3369842456
43	br0005	Caroline	Cooke	Salesperson	1514796258
44	br0005	Caleb	Flynn	Salesperson	1145876354
45	br0005	Dwayne	Rice	Salesperson	5459861475
46	br0005	Izaak	Jennings	Salesperson	3134779648

Query executed successfully.

Joint query between Dealership_Staff, Parts_Warehouse, and Vehicle_WarrantyServices Tables

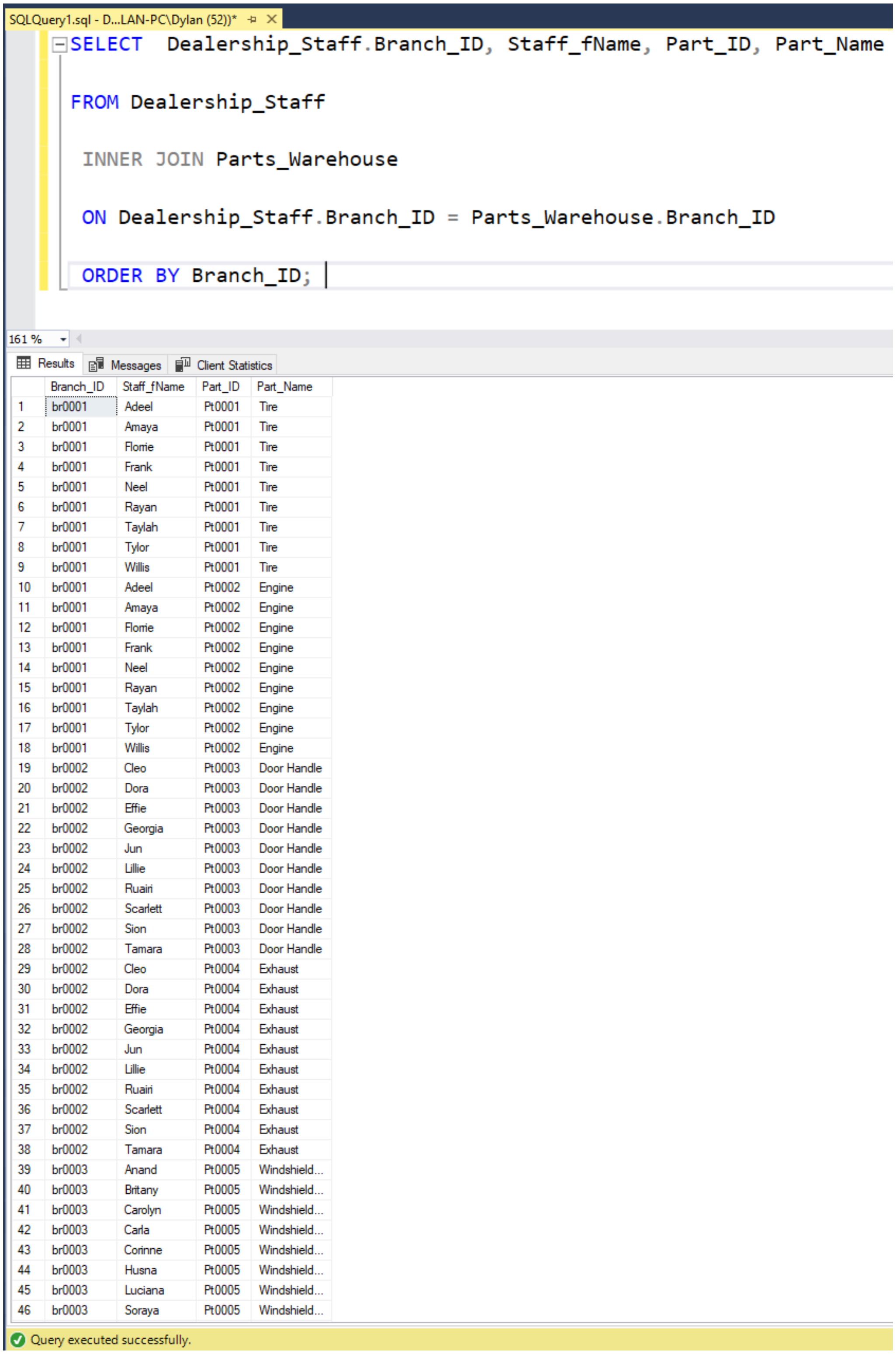


```
SELECT Dealership_Staff.Branch_ID, Dealership_Staff.Staff_fName, Dealership_Staff.Staff_lName,
Parts_Warehouse.Part_ID, Parts_Warehouse.Part_Name, Vehicle_WarrantyServices.WarrantyService_ID,
Vehicle_WarrantyServices.Appt_Date
FROM ((Dealership_Staff
INNER JOIN Parts_Warehouse ON Dealership_Staff.Branch_ID = Parts_Warehouse.Branch_ID)
INNER JOIN Vehicle_WarrantyServices ON Dealership_Staff.Branch_ID = Vehicle_WarrantyServices.Branch_ID)
ORDER BY Branch_ID;
```

	Branch_ID	Staff_fName	Staff_lName	Part_ID	Part_Name	WarrantyService_ID	Appt_Date
1	br0001	Adeel	Homer	Pt0001	Tire	ws0172	2022-01-05
2	br0001	Amaya	Seymour	Pt0001	Tire	ws0172	2022-01-05
3	br0001	Florrie	Humphreys	Pt0001	Tire	ws0172	2022-01-05
4	br0001	Frank	Jarvis	Pt0001	Tire	ws0172	2022-01-05
5	br0001	Neel	Bassett	Pt0001	Tire	ws0172	2022-01-05
6	br0001	Rayan	Knights	Pt0001	Tire	ws0172	2022-01-05
7	br0001	Taylah	Eaton	Pt0001	Tire	ws0172	2022-01-05
8	br0001	Tylor	Mcghee	Pt0001	Tire	ws0172	2022-01-05
9	br0001	Willis	Drew	Pt0001	Tire	ws0172	2022-01-05
10	br0001	Adeel	Homer	Pt0002	Engine	ws0172	2022-01-05
11	br0001	Amaya	Seymour	Pt0002	Engine	ws0172	2022-01-05
12	br0001	Florrie	Humphreys	Pt0002	Engine	ws0172	2022-01-05
13	br0001	Frank	Jarvis	Pt0002	Engine	ws0172	2022-01-05
14	br0001	Neel	Bassett	Pt0002	Engine	ws0172	2022-01-05
15	br0001	Rayan	Knights	Pt0002	Engine	ws0172	2022-01-05
16	br0001	Taylah	Eaton	Pt0002	Engine	ws0172	2022-01-05
17	br0001	Tylor	Mcghee	Pt0002	Engine	ws0172	2022-01-05
18	br0001	Willis	Drew	Pt0002	Engine	ws0172	2022-01-05
19	br0002	Cleo	Gay	Pt0003	Door Handle	ws0274	2021-12-18
20	br0002	Dora	Conner	Pt0003	Door Handle	ws0274	2021-12-18
21	br0002	Effie	Feeny	Pt0003	Door Handle	ws0274	2021-12-18
22	br0002	Georgia	Thorne	Pt0003	Door Handle	ws0274	2021-12-18
23	br0002	Jun	Millington	Pt0003	Door Handle	ws0274	2021-12-18
24	br0002	Lillie	Stevens	Pt0003	Door Handle	ws0274	2021-12-18
25	br0002	Ruairi	Wheeler	Pt0003	Door Handle	ws0274	2021-12-18
26	br0002	Scarlett	Manning	Pt0003	Door Handle	ws0274	2021-12-18
27	br0002	Sion	Schmidt	Pt0003	Door Handle	ws0274	2021-12-18
28	br0002	Tamara	Walls	Pt0003	Door Handle	ws0274	2021-12-18
29	br0002	Cleo	Gay	Pt0004	Exhaust	ws0274	2021-12-18
30	br0002	Dora	Conner	Pt0004	Exhaust	ws0274	2021-12-18
31	br0002	Effie	Feeny	Pt0004	Exhaust	ws0274	2021-12-18
32	br0002	Georgia	Thorne	Pt0004	Exhaust	ws0274	2021-12-18
33	br0002	Jun	Millington	Pt0004	Exhaust	ws0274	2021-12-18
34	br0002	Lillie	Stevens	Pt0004	Exhaust	ws0274	2021-12-18
35	br0002	Ruairi	Wheeler	Pt0004	Exhaust	ws0274	2021-12-18
36	br0002	Scarlett	Manning	Pt0004	Exhaust	ws0274	2021-12-18
37	br0002	Sion	Schmidt	Pt0004	Exhaust	ws0274	2021-12-18
38	br0002	Tamara	Walls	Pt0004	Exhaust	ws0274	2021-12-18
39	br0003	Anand	Alexander	Pt0005	Windshield...	ws0311	2022-02-13
40	br0003	Britany	Brown	Pt0005	Windshield...	ws0311	2022-02-13
41	br0003	Carolyn	Arellano	Pt0005	Windshield...	ws0311	2022-02-13
42	br0003	Carla	Bob	Pt0005	Windshield...	ws0311	2022-02-13
43	br0003	Corinne	Plummer	Pt0005	Windshield...	ws0311	2022-02-13
44	br0003	Husna	Lucas	Pt0005	Windshield...	ws0311	2022-02-13
45	br0003	Luciana	Steadman	Pt0005	Windshield...	ws0311	2022-02-13
46	br0003	Soraya	Greenaway	Pt0005	Windshield...	ws0311	2022-02-13

Query executed successfully.

Joint query between Dealership_Staff and Parts_Warehouse Tables

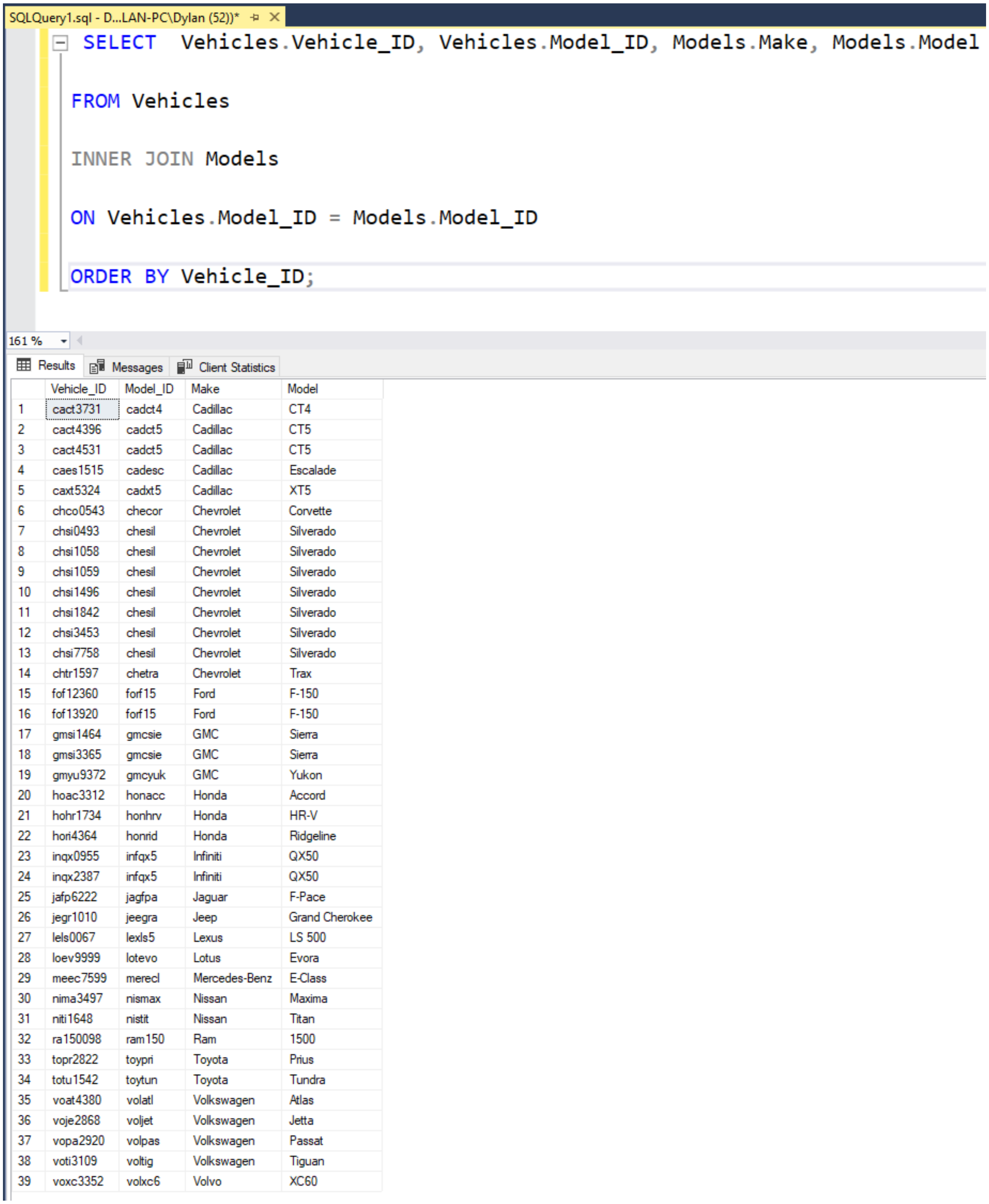


```
SELECT  Dealership_Staff.Branch_ID, Staff_fName, Part_ID, Part_Name

FROM Dealership_Staff

 INNER JOIN Parts_Warehouse

 ON Dealership_Staff.Branch_ID = Parts_Warehouse.Branch_ID

 ORDER BY Branch_ID;
```

	Branch_ID	Staff_fName	Part_ID	Part_Name
1	br0001	Adeel	Pt0001	Tire
2	br0001	Amaya	Pt0001	Tire
3	br0001	Florrie	Pt0001	Tire
4	br0001	Frank	Pt0001	Tire
5	br0001	Neel	Pt0001	Tire
6	br0001	Rayan	Pt0001	Tire
7	br0001	Taylah	Pt0001	Tire
8	br0001	Tylor	Pt0001	Tire
9	br0001	Willis	Pt0001	Tire
10	br0001	Adeel	Pt0002	Engine
11	br0001	Amaya	Pt0002	Engine
12	br0001	Florrie	Pt0002	Engine
13	br0001	Frank	Pt0002	Engine
14	br0001	Neel	Pt0002	Engine
15	br0001	Rayan	Pt0002	Engine
16	br0001	Taylah	Pt0002	Engine
17	br0001	Tylor	Pt0002	Engine
18	br0001	Willis	Pt0002	Engine
19	br0002	Cleo	Pt0003	Door Handle
20	br0002	Dora	Pt0003	Door Handle
21	br0002	Effie	Pt0003	Door Handle
22	br0002	Georgia	Pt0003	Door Handle
23	br0002	Jun	Pt0003	Door Handle
24	br0002	Lillie	Pt0003	Door Handle
25	br0002	Ruairi	Pt0003	Door Handle
26	br0002	Scarlett	Pt0003	Door Handle
27	br0002	Sion	Pt0003	Door Handle
28	br0002	Tamara	Pt0003	Door Handle
29	br0002	Cleo	Pt0004	Exhaust
30	br0002	Dora	Pt0004	Exhaust
31	br0002	Effie	Pt0004	Exhaust
32	br0002	Georgia	Pt0004	Exhaust
33	br0002	Jun	Pt0004	Exhaust
34	br0002	Lillie	Pt0004	Exhaust
35	br0002	Ruairi	Pt0004	Exhaust
36	br0002	Scarlett	Pt0004	Exhaust
37	br0002	Sion	Pt0004	Exhaust
38	br0002	Tamara	Pt0004	Exhaust
39	br0003	Anand	Pt0005	Windshield...
40	br0003	Britany	Pt0005	Windshield...
41	br0003	Carolyn	Pt0005	Windshield...
42	br0003	Carla	Pt0005	Windshield...
43	br0003	Corinne	Pt0005	Windshield...
44	br0003	Husna	Pt0005	Windshield...
45	br0003	Luciana	Pt0005	Windshield...
46	br0003	Soraya	Pt0005	Windshield...

✓ Query executed successfully.

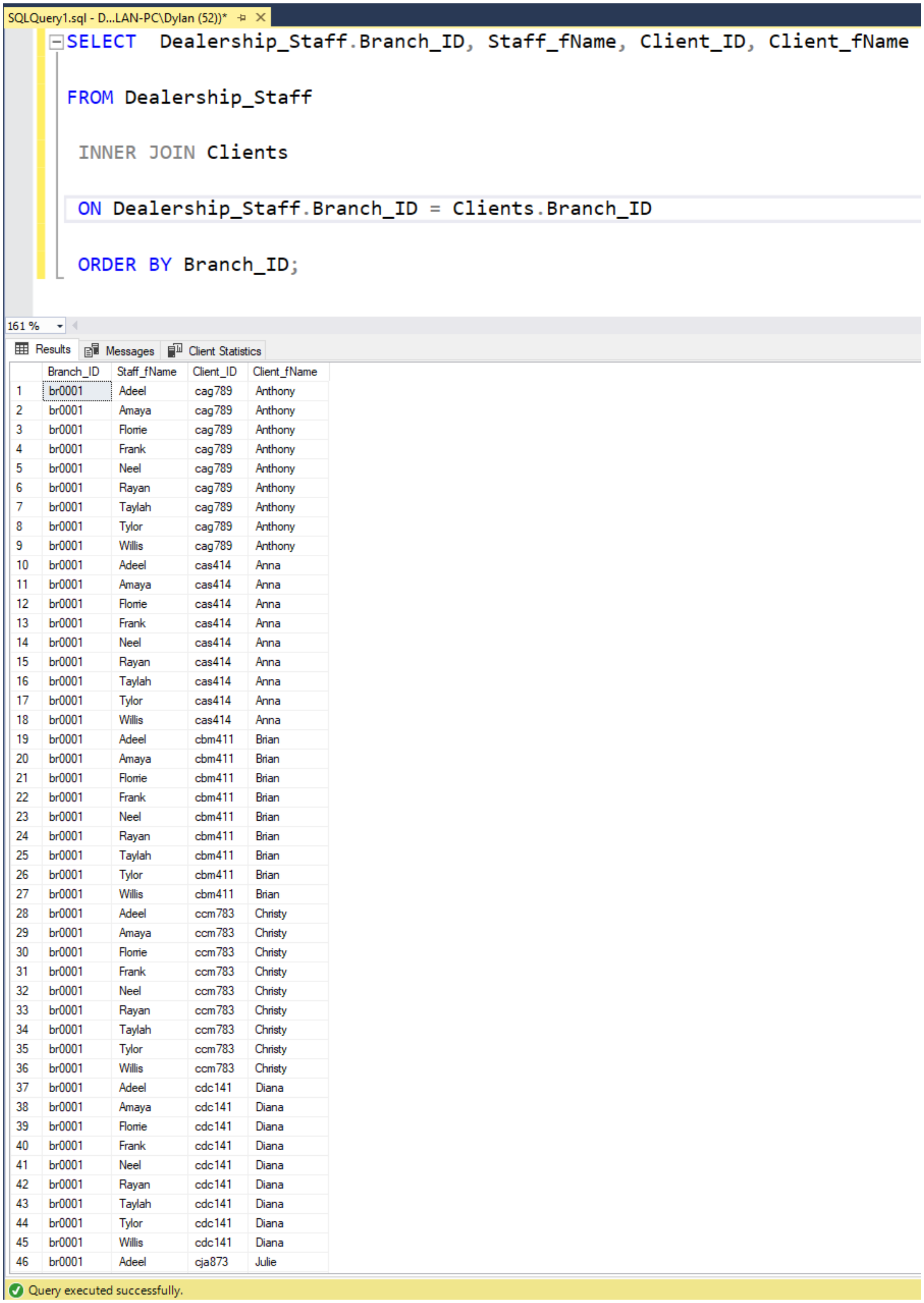
Joint query between Vehicles and Models Tables



```sql
SELECT  Vehicles.Vehicle_ID, Vehicles.Model_ID, Models.Make, Models.Model

FROM Vehicles

INNER JOIN Models

ON Vehicles.Model_ID = Models.Model_ID

ORDER BY Vehicle_ID;
```

	Vehicle_ID	Model_ID	Make	Model
1	cact3731	cadct4	Cadillac	CT4
2	cact4396	cadct5	Cadillac	CT5
3	cact4531	cadct5	Cadillac	CT5
4	caes1515	cadesc	Cadillac	Escalade
5	caxt5324	cadxt5	Cadillac	XT5
6	chco0543	checor	Chevrolet	Corvette
7	chsi0493	chesil	Chevrolet	Silverado
8	chsi1058	chesil	Chevrolet	Silverado
9	chsi1059	chesil	Chevrolet	Silverado
10	chsi1496	chesil	Chevrolet	Silverado
11	chsi1842	chesil	Chevrolet	Silverado
12	chsi3453	chesil	Chevrolet	Silverado
13	chsi7758	chesil	Chevrolet	Silverado
14	chtr1597	chetra	Chevrolet	Trax
15	fof12360	forf15	Ford	F-150
16	fof13920	forf15	Ford	F-150
17	gmsi1464	gmcsie	GMC	Sierra
18	gmsi3365	gmcsie	GMC	Sierra
19	gmyu9372	gmcyuk	GMC	Yukon
20	hoac3312	honacc	Honda	Accord
21	hohr1734	honhrv	Honda	HR-V
22	hori4364	honrid	Honda	Ridgeline
23	inqx0955	infqx5	Infiniti	QX50
24	inqx2387	infqx5	Infiniti	QX50
25	jafp6222	jagfpa	Jaguar	F-Pace
26	jegr1010	jeegra	Jeep	Grand Cherokee
27	lels0067	lexls5	Lexus	LS 500
28	loev9999	lotevo	Lotus	Evora
29	meec7599	merecl	Mercedes-Benz	E-Class
30	nima3497	nismax	Nissan	Maxima
31	niti1648	nistit	Nissan	Titan
32	ra150098	ram150	Ram	1500
33	topr2822	toypri	Toyota	Prius
34	totu1542	toytun	Toyota	Tundra
35	voat4380	volatl	Volkswagen	Atlas
36	voje2868	voljet	Volkswagen	Jetta
37	vopa2920	volpas	Volkswagen	Passat
38	voti3109	voltig	Volkswagen	Tiguan
39	voxc3352	volxc6	Volvo	XC60

Joint query between Dealership_Staff and Clients tables



```
SELECT  Dealership_Staff.Branch_ID, Staff_fName, Client_ID, Client_fName

FROM Dealership_Staff

 INNER JOIN Clients

 ON Dealership_Staff.Branch_ID = Clients.Branch_ID

 ORDER BY Branch_ID;
```

	Branch_ID	Staff_fName	Client_ID	Client_fName
1	br0001	Adeel	cag789	Anthony
2	br0001	Amaya	cag789	Anthony
3	br0001	Florrie	cag789	Anthony
4	br0001	Frank	cag789	Anthony
5	br0001	Neel	cag789	Anthony
6	br0001	Rayan	cag789	Anthony
7	br0001	Taylah	cag789	Anthony
8	br0001	Tylor	cag789	Anthony
9	br0001	Willis	cag789	Anthony
10	br0001	Adeel	cas414	Anna
11	br0001	Amaya	cas414	Anna
12	br0001	Florrie	cas414	Anna
13	br0001	Frank	cas414	Anna
14	br0001	Neel	cas414	Anna
15	br0001	Rayan	cas414	Anna
16	br0001	Taylah	cas414	Anna
17	br0001	Tylor	cas414	Anna
18	br0001	Willis	cas414	Anna
19	br0001	Adeel	cbm411	Brian
20	br0001	Amaya	cbm411	Brian
21	br0001	Florrie	cbm411	Brian
22	br0001	Frank	cbm411	Brian
23	br0001	Neel	cbm411	Brian
24	br0001	Rayan	cbm411	Brian
25	br0001	Taylah	cbm411	Brian
26	br0001	Tylor	cbm411	Brian
27	br0001	Willis	cbm411	Brian
28	br0001	Adeel	ccm783	Christy
29	br0001	Amaya	ccm783	Christy
30	br0001	Florrie	ccm783	Christy
31	br0001	Frank	ccm783	Christy
32	br0001	Neel	ccm783	Christy
33	br0001	Rayan	ccm783	Christy
34	br0001	Taylah	ccm783	Christy
35	br0001	Tylor	ccm783	Christy
36	br0001	Willis	ccm783	Christy
37	br0001	Adeel	cdc141	Diana
38	br0001	Amaya	cdc141	Diana
39	br0001	Florrie	cdc141	Diana
40	br0001	Frank	cdc141	Diana
41	br0001	Neel	cdc141	Diana
42	br0001	Rayan	cdc141	Diana
43	br0001	Taylah	cdc141	Diana
44	br0001	Tylor	cdc141	Diana
45	br0001	Willis	cdc141	Diana
46	br0001	Adeel	cja873	Julie

✓ Query executed successfully.

- Conclusion -

The DBMS project, automobile database management system is for the creation of a working database for several local dealerships. The DBMS allows for the collection of each dealership's data, which includes staff, clients, dealership information, vehicles, warranty services, and parts. The system allows users to enter, update and delete the information of each entity related to the dealerships. Furthermore, the users can extract reports, perform queries of the data, thus allowing users to view various information pertaining to each dealership. The database is very easy to use and can be adapted to new dealerships if needed. The DBMS project was very insightful and allowed our team to experience the process of creating a DBMS from scratch. The flow of the project and its requirements allowed us to learn while being able to create the DBMS. The additional modifications and requirements from phase to phase provided various opportunities to apply the knowledge and material received from instruction.

- References -

Connolly, Thomas M., and Carolyn E. Beg. Database Systems : a Practical Approach to Design, Implementation, and Management . Sixth edition., Pearson, 2015.

"SQL Tutorial." w*ww.W3schools.com*, www.w3schools.com/sql/default.asp. Accessed 4 Nov. 2021.

"SQL Tutorial" w*ww.Quackit.Com*, www.quackit.com/sql/tutorial. Accessed 13 Nov. 2021.