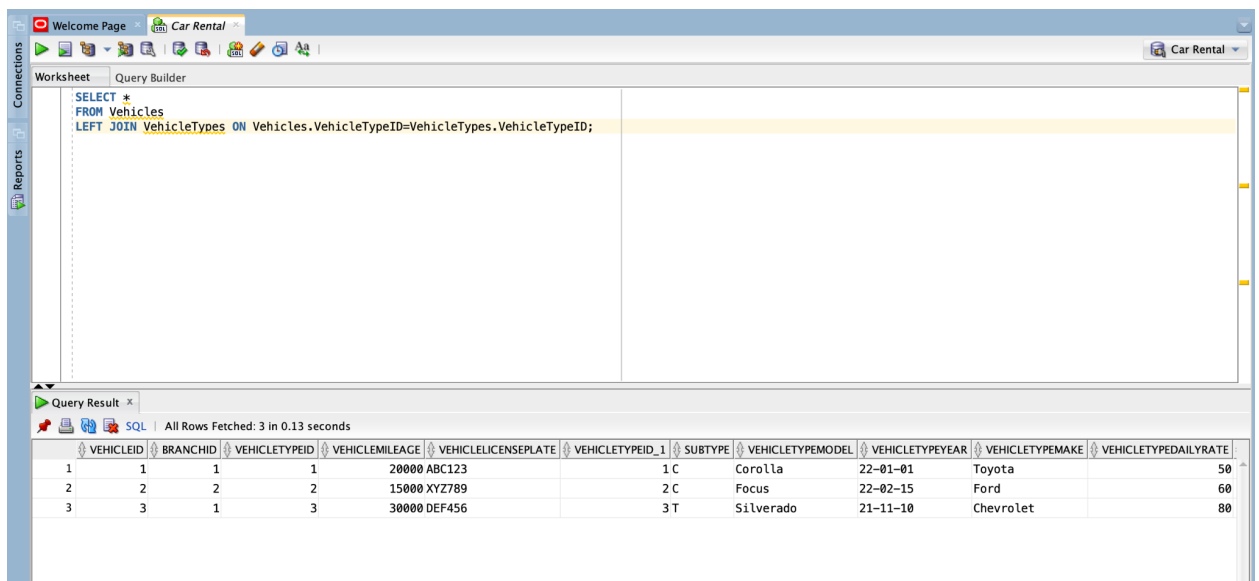


JOIN QUERIES

1. Retrieve all the Vehicle type “Car” in the database.

```
SELECT *  
FROM Vehicles  
LEFT JOIN VehicleTypes ON  
Vehicles.VehicleTypeID=VehicleTypes.VehicleTypeID;
```

Performing the query and the result:



The screenshot shows a database query tool interface. The top pane displays the SQL query: `SELECT * FROM Vehicles LEFT JOIN VehicleTypes ON Vehicles.VehicleTypeID=VehicleTypes.VehicleTypeID;`. The bottom pane shows the query results in a table format. The table has 11 columns: VEHICLEID, BRANCHID, VEHICLETYPEID, VEHICLEMILEAGE, VEHICLELICENSEPLATE, VEHICLEYPEID_1, SUBTYPE, VEHICLEYPEMODEL, VEHICLEYPEYEAR, VEHICLEYPEMAKE, and VEHICLEYPEDAILYRATE. There are 3 rows of data.

	VEHICLEID	BRANCHID	VEHICLETYPEID	VEHICLEMILEAGE	VEHICLELICENSEPLATE	VEHICLEYPEID_1	SUBTYPE	VEHICLEYPEMODEL	VEHICLEYPEYEAR	VEHICLEYPEMAKE	VEHICLEYPEDAILYRATE
1	1	1	1	20000	ABC123	1	C	Corolla	22-01-01	Toyota	50
2	2	2	2	15000	XYZ789	2	C	Focus	22-02-15	Ford	60
3	3	3	1	30000	DEF456	3	T	Silverado	21-11-10	Chevrolet	80

2. Retrieve all the users that are Customers.

```

SELECT *
FROM Users
INNER JOIN Customers ON Users.UserID=Customers.UserID;

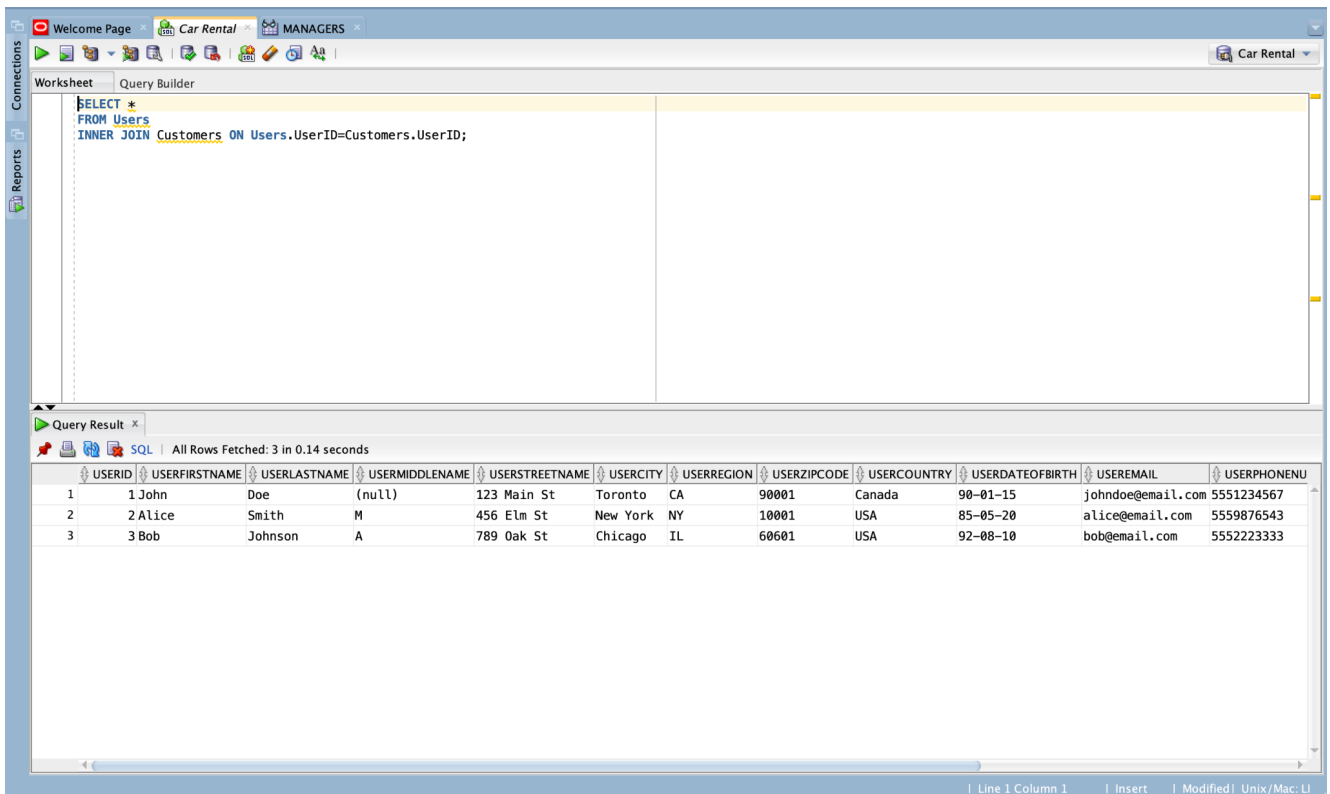
```

```

SELECT *
FROM Users s
JOIN PaymentDetails p ON s.UserID = p.CustomerID
JOIN Transactions t ON p.PaymentDetailsID
= t.PaymentDetailsID;

```

Performing the query and the result:



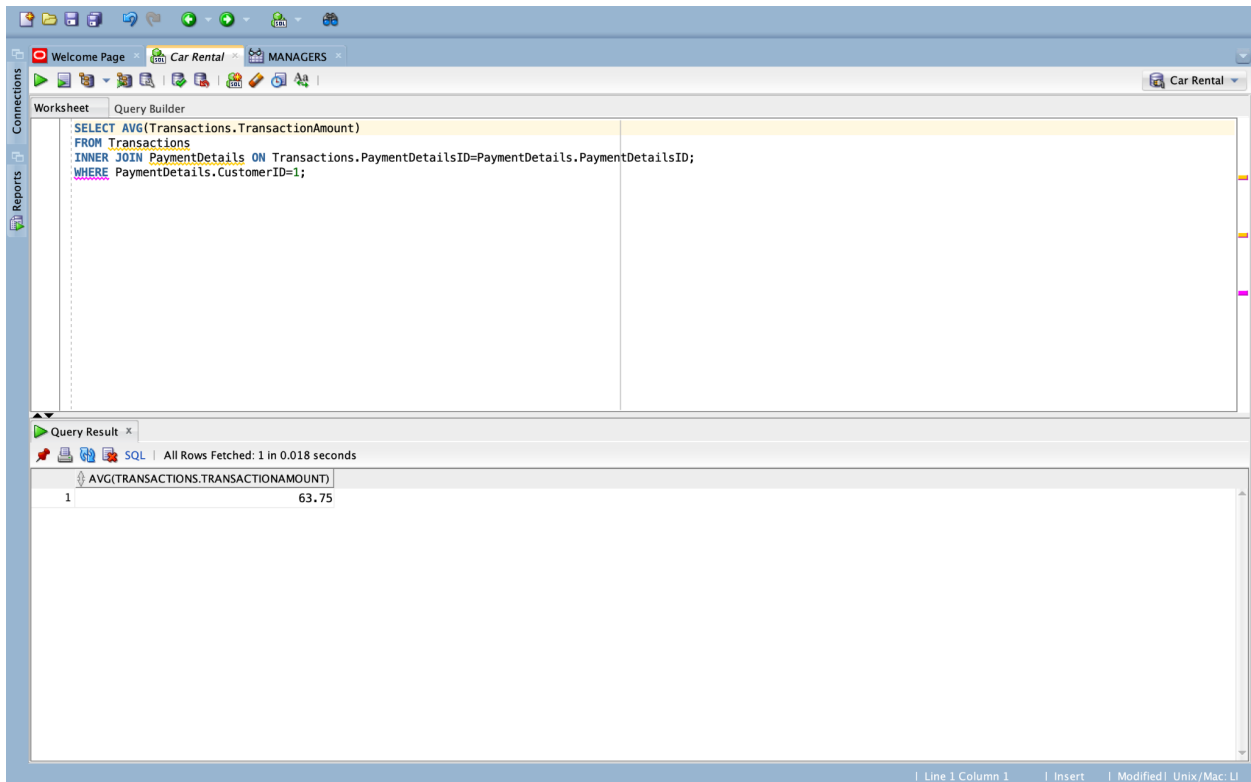
The screenshot shows a database management tool interface. The top pane displays a SQL query: `SELECT * FROM Users INNER JOIN Customers ON Users.UserID=Customers.UserID;`. The bottom pane shows the query results, which are 3 rows of user data. The status bar indicates 'All Rows Fetched: 3 in 0.14 seconds'.

USERID	USERFIRSTNAME	USERLASTNAME	USERMIDDLENAME	USERSTREETNAME	USERCITY	USERREGION	USERZIPCODE	USERCOUNTRY	USERDATEOFBIRTH	USEREMAIL	USERPHONENU
1	John	Doe	(null)	123 Main St	Toronto	CA	90001	Canada	90-01-15	john.doe@email.com	5551234567
2	Alice	Smith	M	456 Elm St	New York	NY	10001	USA	85-05-20	alice@email.com	5559876543
3	Bob	Johnson	A	789 Oak St	Chicago	IL	60601	USA	92-08-10	bob@email.com	5552223333

3. Returns the average of all the TransactionAmounts associated with CustomerID = 1.

```
SELECT AVG(Transactions.TransactionAmount)
FROM Transactions
INNER JOIN PaymentDetails ON
Transactions.PaymentDetailsID=PaymentDetails.PaymentDetailsID;
WHERE PaymentDetails.CustomerID=1;
```

Performing the query and the result:



The screenshot shows a database management tool interface. The top toolbar includes icons for file operations, database connections, and query execution. The main window is divided into two panes: 'Query Builder' and 'Query Result'. The 'Query Builder' pane contains the following SQL query:

```
SELECT AVG(Transactions.TransactionAmount)
FROM Transactions
INNER JOIN PaymentDetails ON Transactions.PaymentDetailsID=PaymentDetails.PaymentDetailsID;
WHERE PaymentDetails.CustomerID=1;
```

The 'Query Result' pane displays the results of the query. It shows a single row with the following data:

AVG(TRANSACTIONS.TRANSACTIONAMOUNT)
63.75

The status bar at the bottom indicates 'Line 1 Column 1', 'Insert', and 'Modified | Unix/Mac: Lf'.

4. Listing all transactions with their respective payment details and user information.

```

SELECT *
FROM Users s
JOIN PaymentDetails p ON s.UserID = p.CustomerID
JOIN Transactions t ON p.PaymentDetailsID
= t.PaymentDetailsID;

```

Performing the query and the result:

The screenshot shows a database management tool interface. The top pane displays the following SQL query:

```

SELECT *
FROM Users s
JOIN PaymentDetails p ON s.UserID = p.CustomerID
JOIN Transactions t ON p.PaymentDetailsID
= t.PaymentDetailsID;

```

The bottom pane shows the results of the query, indicating that 4 rows were fetched in 0.283 seconds. The results are displayed in a table with the following columns: USERID, USERFIRSTNAME, USERLASTNAME, USERMIDDLENAME, USERSTREETNAME, USERCITY, USERREGION, USERZIPCODE, USERCOUNTRY, USERDATEOFBIRTH, USEREMAIL, and USERPHONENUM.

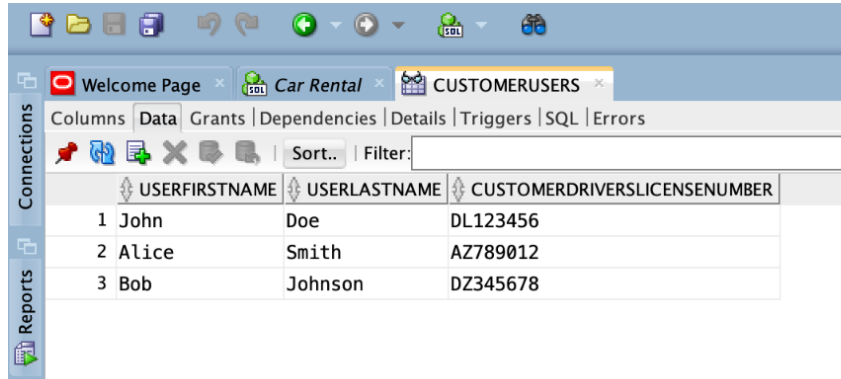
USERID	USERFIRSTNAME	USERLASTNAME	USERMIDDLENAME	USERSTREETNAME	USERCITY	USERREGION	USERZIPCODE	USERCOUNTRY	USERDATEOFBIRTH	USEREMAIL	USERPHONENUM
1	2 Alice	Smith	M	456 Elm St	New York	NY	10001	USA	85-05-20	alice@email.com	5559876543
2	1 John	Doe	(null)	123 Main St	Toronto	CA	90001	Canada	90-01-15	johndoe@email.com	5551234567
3	2 Alice	Smith	M	456 Elm St	New York	NY	10001	USA	85-05-20	alice@email.com	5559876543
4	3 Bob	Johnson	A	789 Oak St	Chicago	IL	60601	USA	92-08-10	bob@email.com	5552223333

View Queries

1. Create a view for Users that are Customers.

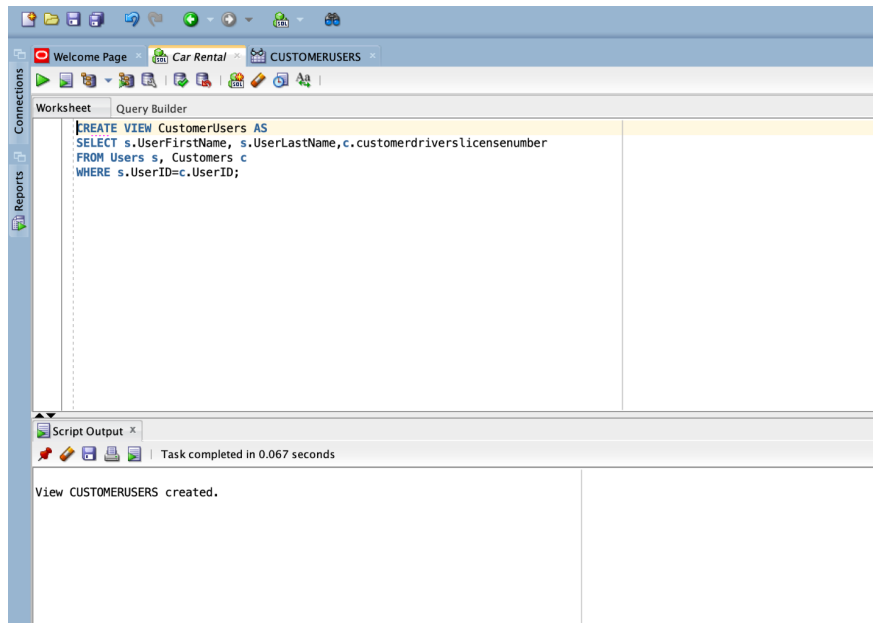
```
CREATE VIEW CustomerUsers AS
SELECT *
FROM Users s, Customers c
WHERE s.UserID=c.UserID;
```

The View:



	USERFIRSTNAME	USERLASTNAME	CUSTOMERDRIVERSLICENSENUMBER
1	John	Doe	DL123456
2	Alice	Smith	AZ789012
3	Bob	Johnson	DZ345678

Performing the query and the result:



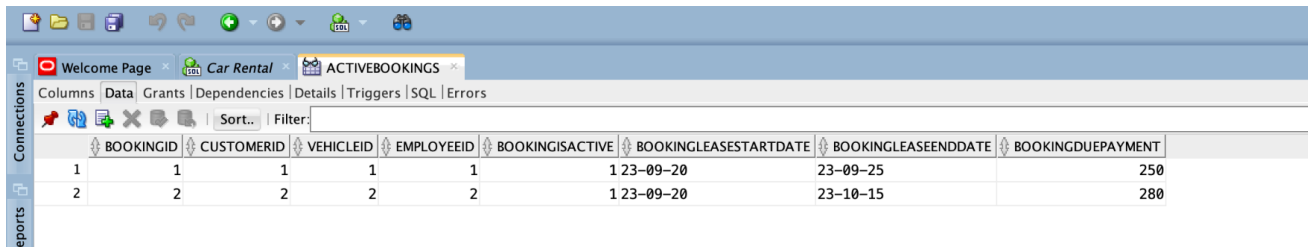
```
CREATE VIEW CustomerUsers AS
SELECT s.UserFirstName, s.UserLastName, c.customerdriverslicensenumbr
FROM Users s, Customers c
WHERE s.UserID=c.UserID;
```

Script Output x
Task completed in 0.067 seconds
View CUSTOMERUSERS created.

2. Create a view for Active Bookings.

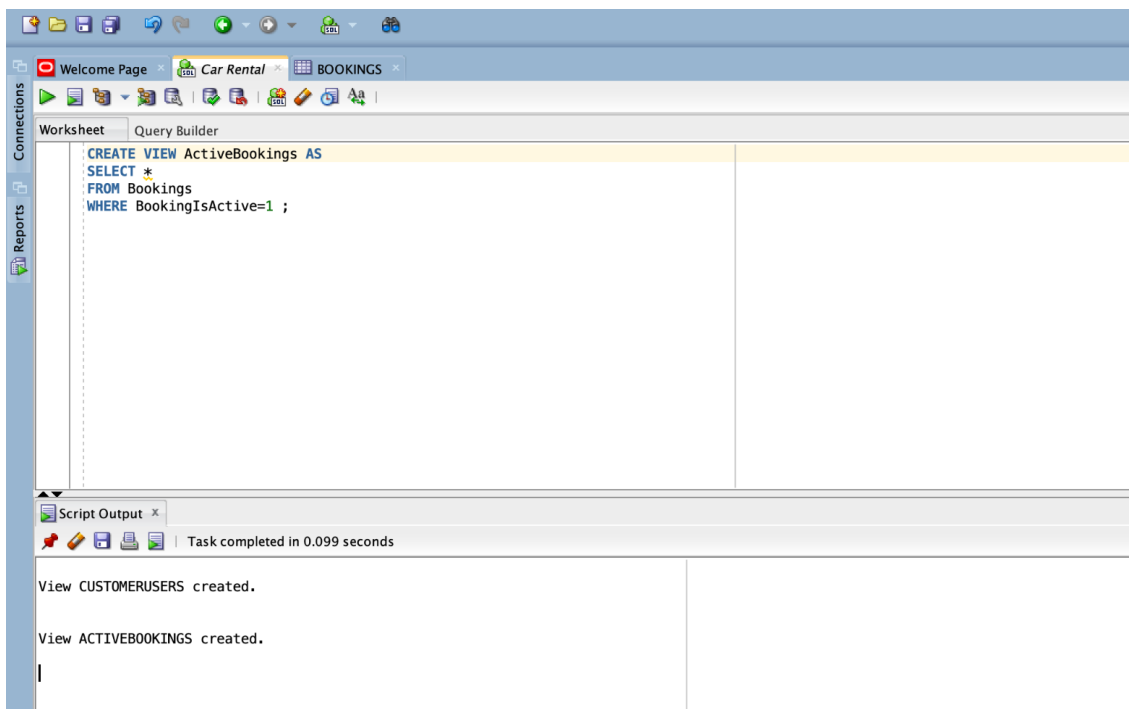
```
CREATE VIEW ActiveBookings AS
SELECT *
FROM Bookings
WHERE BookingIsActive=1 ;
```

The View:



BOOKINGID	CUSTOMERID	VEHICLEID	EMPLOYEEID	BOOKINGISACTIVE	BOOKINGLEASESTARTDATE	BOOKINGLEASEENDDATE	BOOKINGDUEPAYMENT
1	1	1	1	1	23-09-20	23-09-25	250
2	2	2	2	1	23-09-20	23-10-15	280

Performing the query and the result:



```
CREATE VIEW ActiveBookings AS
SELECT *
FROM Bookings
WHERE BookingIsActive=1 ;
```

Script Output

Task completed in 0.099 seconds

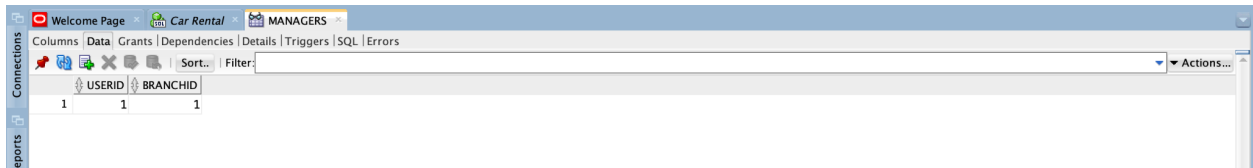
View CUSTOMERUSERS created.

View ACTIVEBOOKINGS created.

3. Create a view for Managers.

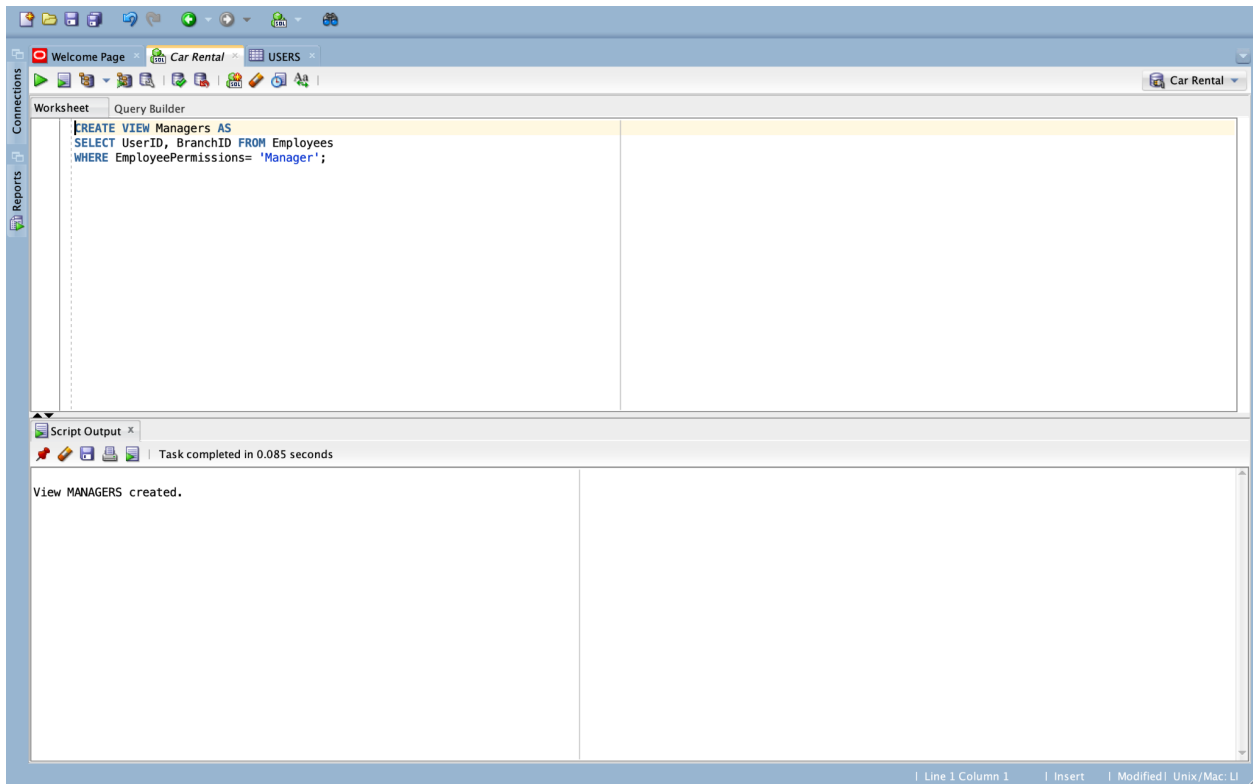
```
CREATE VIEW Managers AS
SELECT UserID, BranchID FROM Employees
WHERE EmployeePermissions='Manager';
```

The View:



USERID	BRANCHID	
1	1	1

Performing the query and the result:



```
CREATE VIEW Managers AS
SELECT UserID, BranchID FROM Employees
WHERE EmployeePermissions= 'Manager';
```

Script Output

Task completed in 0.085 seconds

View MANAGERS created.