

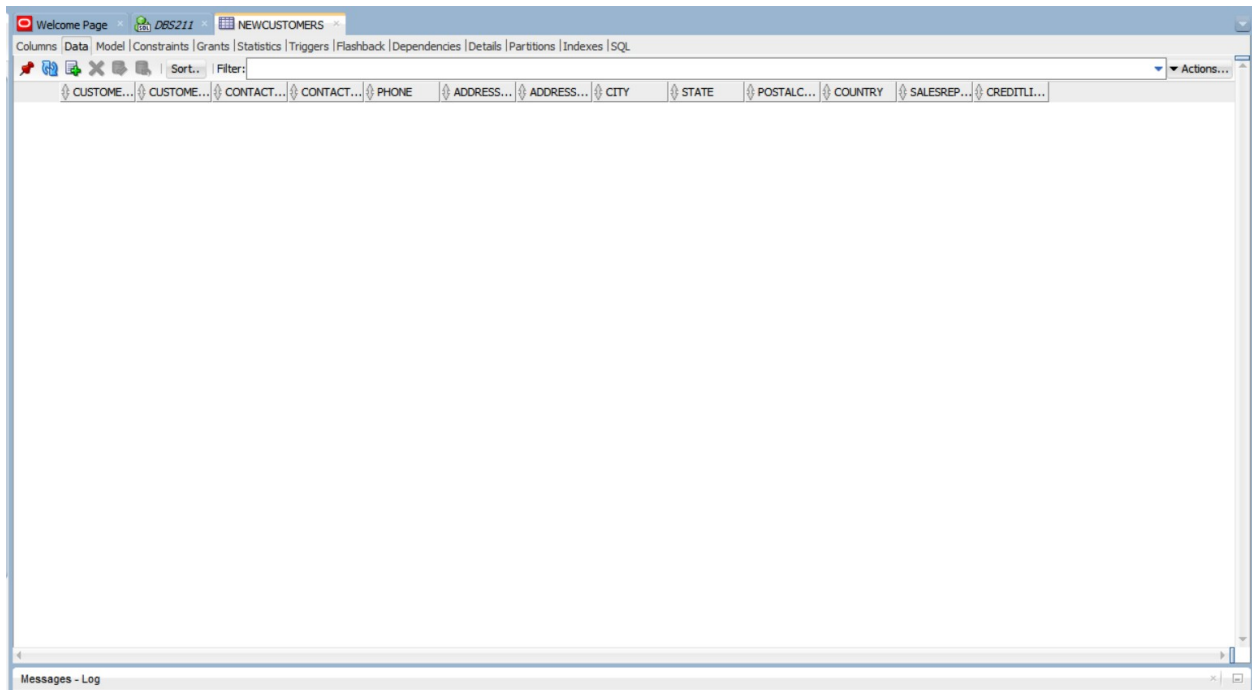
Q1

```
CREATE TABLE newCustomerS AS
```

```
SELECT *
```

```
FROM RETAILCUSTOMERS
```

```
WHERE 1 = 2;
```



Q2

```
INSERT ALL
```

```
INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city,  
country, customerName) VALUES (100, 'Patel', 'Ralph', '2233355555', '10 Seneca Way', 'Paris', 'France',  
'Ralph Patel')
```

```
INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city,  
country, customerName) VALUES (101, 'Denis', 'Betty', '3344455555', '110 Seneca Way', 'Chicago', 'USA',  
'Betty Denis')
```

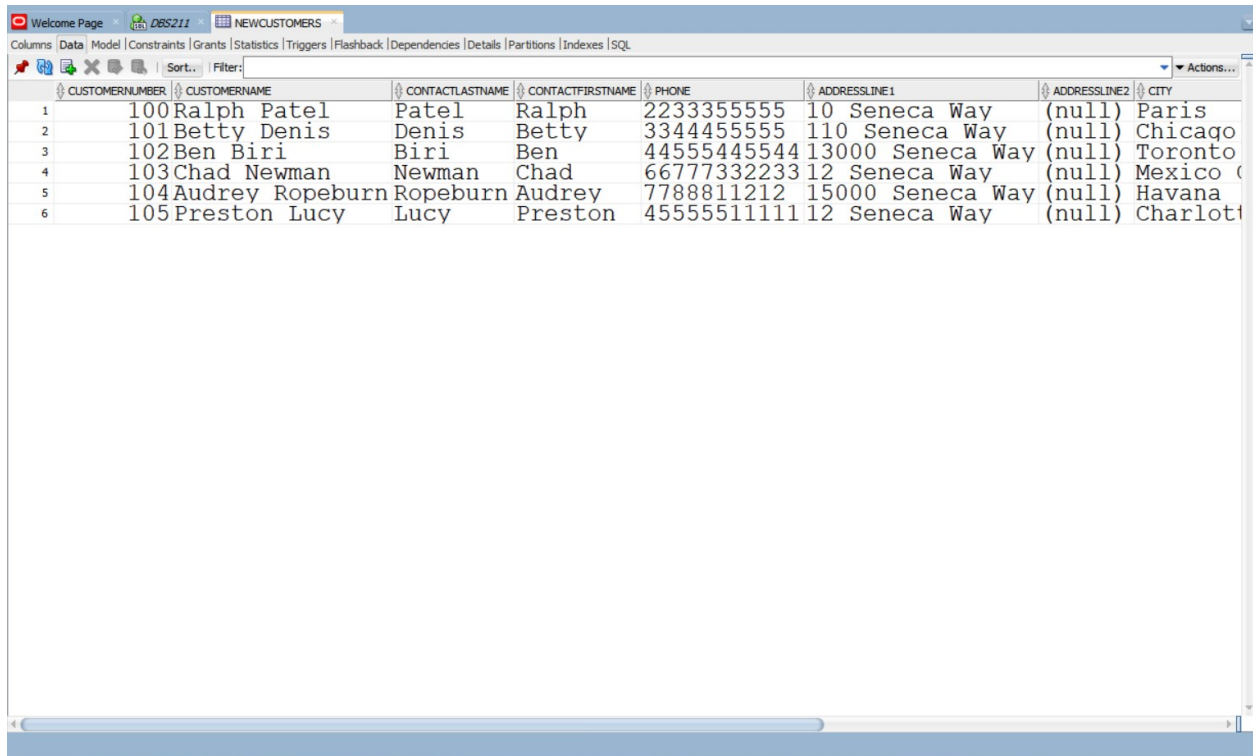
INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (102, 'Biri', 'Ben', '44555445544', '13000 Seneca Way', 'Toronto', 'Canada', 'Ben Biri')

INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (103, 'Newman', 'Chad', '66777332233', '12 Seneca Way', 'Mexico City', 'Mexico', 'Chad Newman')

INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (104, 'Ropeburn', 'Audrey', '7788811212', '15000 Seneca Way', 'Havana', 'Cuba', 'Audrey Ropeburn')

INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (105, 'Lucy', 'Preston', '45555511111', '12 Seneca Way', 'Charlotte', 'USA', 'Preston Lucy')

SELECT * FROM DUAL;

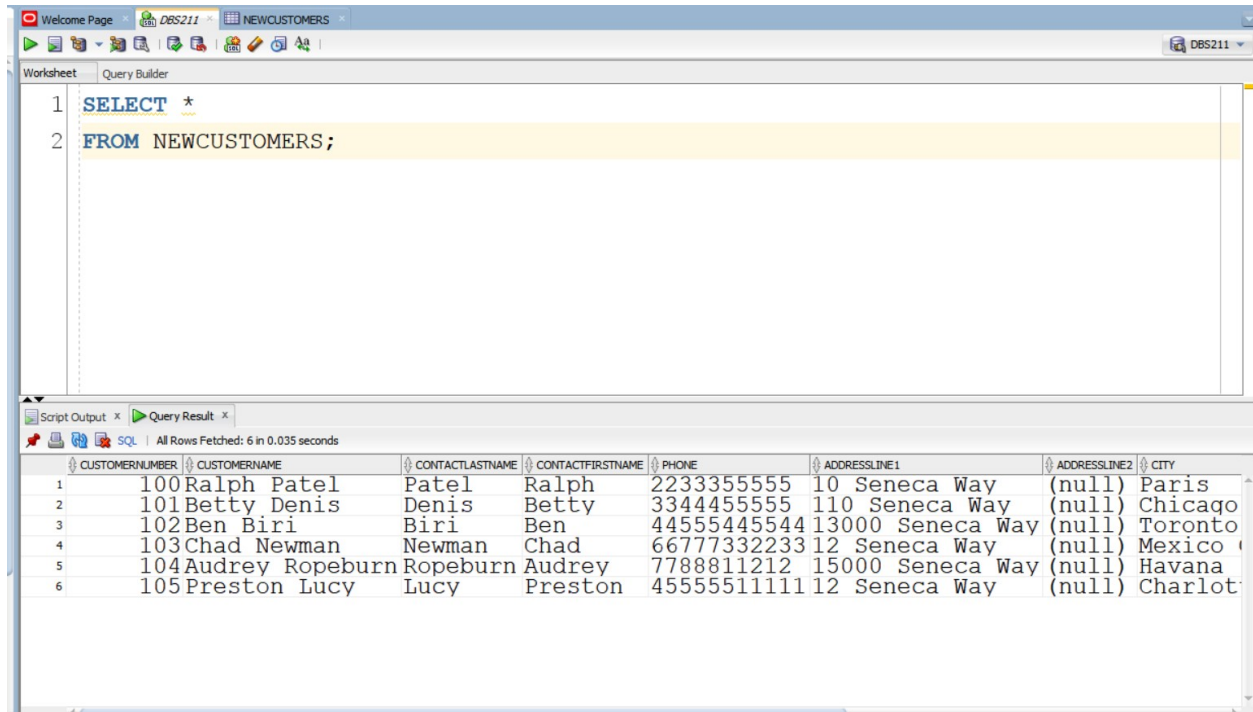


The screenshot shows a database management tool interface with a table named 'NEWCUSTOMERS'. The table has 7 columns: CUSTOMERNUMBER, CUSTOMERNAME, CONTACTLASTNAME, CONTACTFIRSTNAME, PHONE, ADDRESSLINE1, ADDRESSLINE2, and CITY. There are 6 rows of data displayed.

	CUSTOMERNUMBER	CUSTOMERNAME	CONTACTLASTNAME	CONTACTFIRSTNAME	PHONE	ADDRESSLINE1	ADDRESSLINE2	CITY
1	100	Ralph Patel	Patel	Ralph	2233355555	10 Seneca Way	(null)	Paris
2	101	Betty Denis	Denis	Betty	3344455555	110 Seneca Way	(null)	Chicago
3	102	Ben Biri	Biri	Ben	44555445544	13000 Seneca Way	(null)	Toronto
4	103	Chad Newman	Newman	Chad	66777332233	12 Seneca Way	(null)	Mexico City
5	104	Audrey Ropeburn	Ropeburn	Audrey	7788811212	15000 Seneca Way	(null)	Havana
6	105	Preston Lucy	Lucy	Preston	45555511111	12 Seneca Way	(null)	Charlotte

SELECT *

FROM NEWCUSTOMERS;



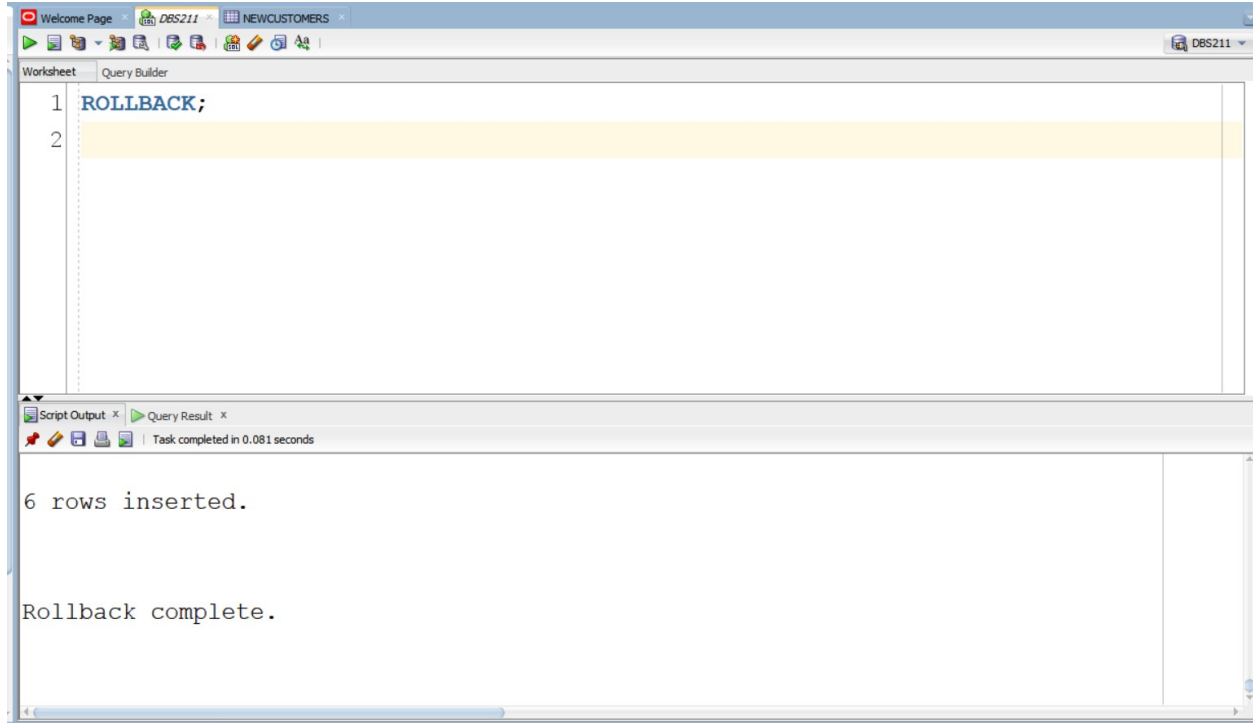
The screenshot shows a database query interface. The top part is the Query Builder, where the SQL statement `SELECT * FROM NEWCUSTOMERS;` is entered. Below this is the Query Result tab, which displays the results of the query. The results are shown in a table with 8 columns: CUSTOMERNUMBER, CUSTOMERNAME, CONTACTLASTNAME, CONTACTFIRSTNAME, PHONE, ADDRESSLINE1, ADDRESSLINE2, and CITY. There are 6 rows of data.

	CUSTOMERNUMBER	CUSTOMERNAME	CONTACTLASTNAME	CONTACTFIRSTNAME	PHONE	ADDRESSLINE1	ADDRESSLINE2	CITY
1	100	Ralph Patel	Patel	Ralph	2233355555	10 Seneca Way	(null)	Paris
2	101	Betty Denis	Denis	Betty	3344455555	110 Seneca Way	(null)	Chicago
3	102	Ben Biri	Biri	Ben	44555445544	13000 Seneca Way	(null)	Toronto
4	103	Chad Newman	Newman	Chad	66777332233	12 Seneca Way	(null)	Mexico
5	104	Audrey Ropeburn	Ropeburn	Audrey	7788811212	15000 Seneca Way	(null)	Havana
6	105	Preston Lucy	Lucy	Preston	45555511111	12 Seneca Way	(null)	Charlot

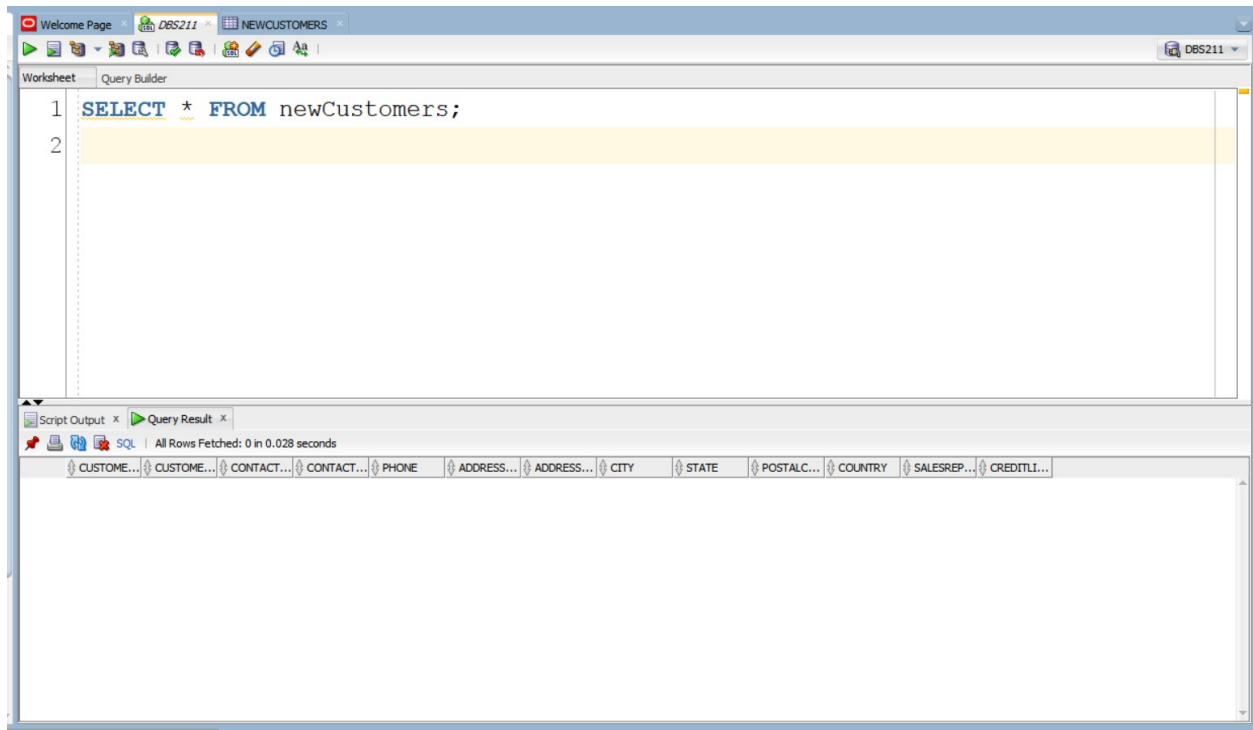
TOTAL 6 ROWS

Q4

ROLLBACK;



SELECT * FROM newCustomers;



Q5

INSERT ALL

INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (100, 'Patel', 'Ralph', '2233355555', '10 Seneca Way', 'Paris', 'France', 'Ralph Patel')

INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (101, 'Denis', 'Betty', '3344455555', '110 Seneca Way', 'Chicago', 'USA', 'Betty Denis')

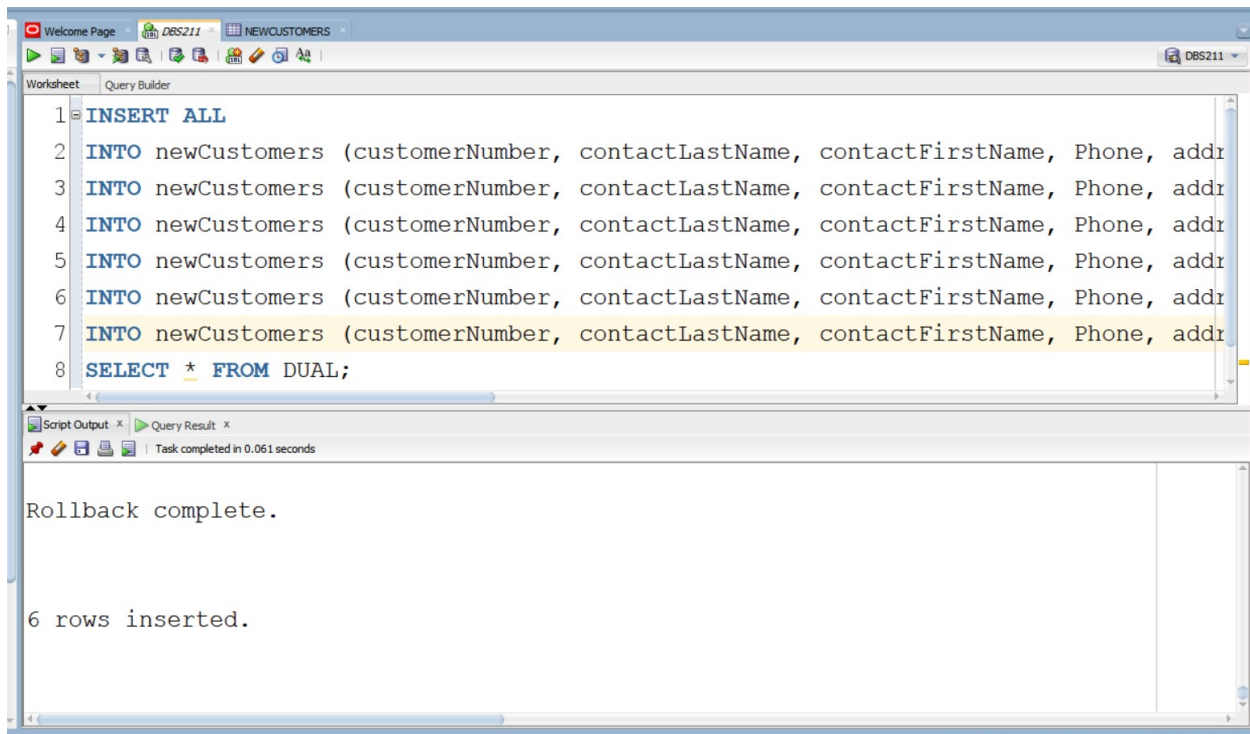
INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (102, 'Biri', 'Ben', '44555445544', '13000 Seneca Way', 'Toronto', 'Canada', 'Ben Biri')

INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (103, 'Newman', 'Chad', '66777332233', '12 Seneca Way', 'Mexico City', 'Mexico', 'Chad Newman')

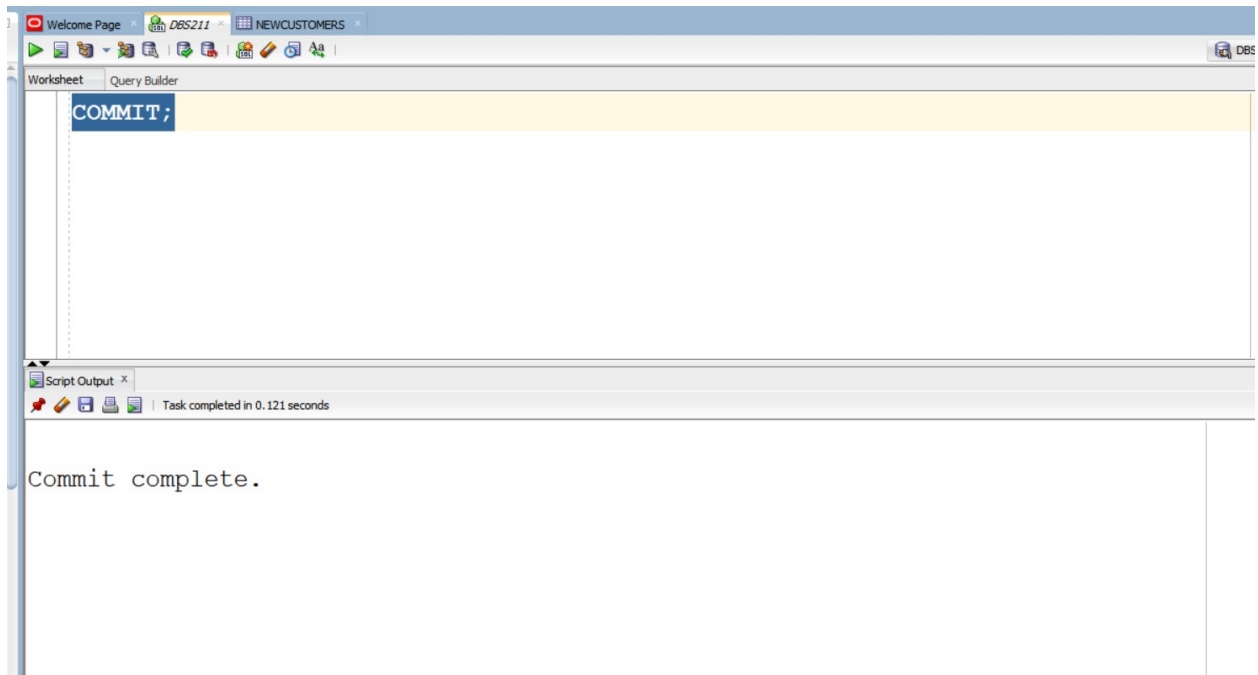
INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (104, 'Ropeburn', 'Audrey', '7788811212', '15000 Seneca Way', 'Havana', 'Cuba', 'Audrey Ropeburn')

INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (105, 'Lucy', 'Preston', '45555511111', '12 Seneca Way', 'Charlotte', 'USA', 'Preston Lucy')

SELECT * FROM DUAL;



COMMIT;



SELECT * FROM newCustomers;

The screenshot shows a database query tool interface. At the top, there are tabs for 'Welcome Page', 'DBS211', and 'NEWCUSTOMERS'. Below the tabs is a toolbar with various icons. The main area is divided into 'Worksheet' and 'Query Builder' tabs. The 'Query Builder' tab is active, showing a SQL query: `SELECT * FROM newCustomers;`. Below the query, there is a 'Query Result' tab showing the results of the query. The results are displayed in a table with 8 columns: CUSTOMERNUMBER, CUSTOMERNAME, CONTACTLASTNAME, CONTACTFIRSTNAME, PHONE, ADDRESSLINE1, ADDRESSLINE2, and CITY. There are 6 rows of data.

CUSTOMERNUMBER	CUSTOMERNAME	CONTACTLASTNAME	CONTACTFIRSTNAME	PHONE	ADDRESSLINE1	ADDRESSLINE2	CITY
100	Ralph Patel	Patel	Ralph	2233355555	10 Seneca Way	(null)	Paris
101	Betty Denis	Denis	Betty	3344455555	110 Seneca Way	(null)	Chicago
102	Ben Biri	Biri	Ben	44555445544	13000 Seneca Way	(null)	Toronto
103	Chad Newman	Newman	Chad	66777332233	12 Seneca Way	(null)	Mexico
104	Audrey Ropeburn	Ropeburn	Audrey	7788811212	15000 Seneca Way	(null)	Havana
105	Preston Lucy	Lucy	Preston	45555511111	12 Seneca Way	(null)	Charlotte

6 ROWS

Q6

UPDATE newCustomers

SET addressLine1 = 'unknown';

DBS211 NEWCUSTOMERS

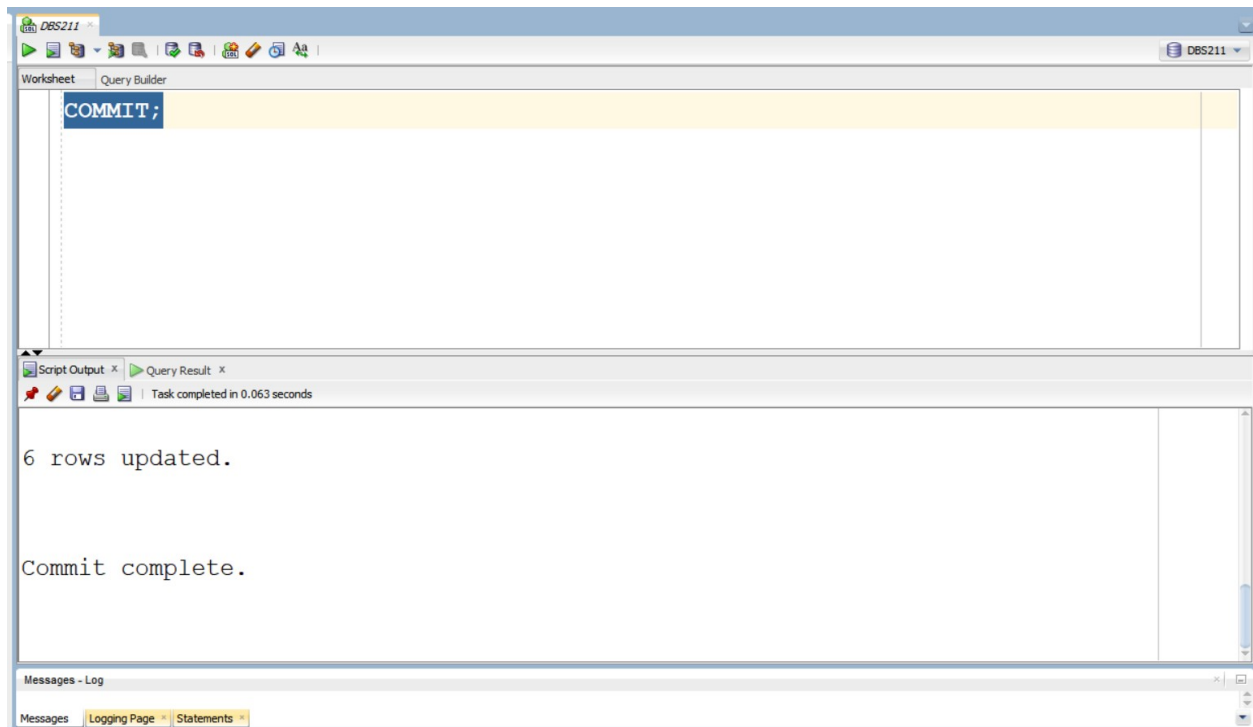
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort... Filter...

	CONTACTLASTNAME	CONTACTFIRSTNAME	PHONE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATE	POSTALCODE	COUNTRY	SALESREPEMPOLEENUN
1	Patel	Ralph	2233355555	unknown	(null)	Paris	(null)	(null)	France	(nul
2	Denis	Betty	3344455555	unknown	(null)	Chicago	(null)	(null)	USA	(nul
3	Biri	Ben	44555445544	unknown	(null)	Toronto	(null)	(null)	Canada	(nul
4	Newman	Chad	66777332233	unknown	(null)	Mexico City	(null)	(null)	Mexico	(nul
5	Ropeburn	Audrey	7788811212	unknown	(null)	Havana	(null)	(null)	Cuba	(nul
6	Lucy	Preston	45555511111	unknown	(null)	Charlotte	(null)	(null)	USA	(nul

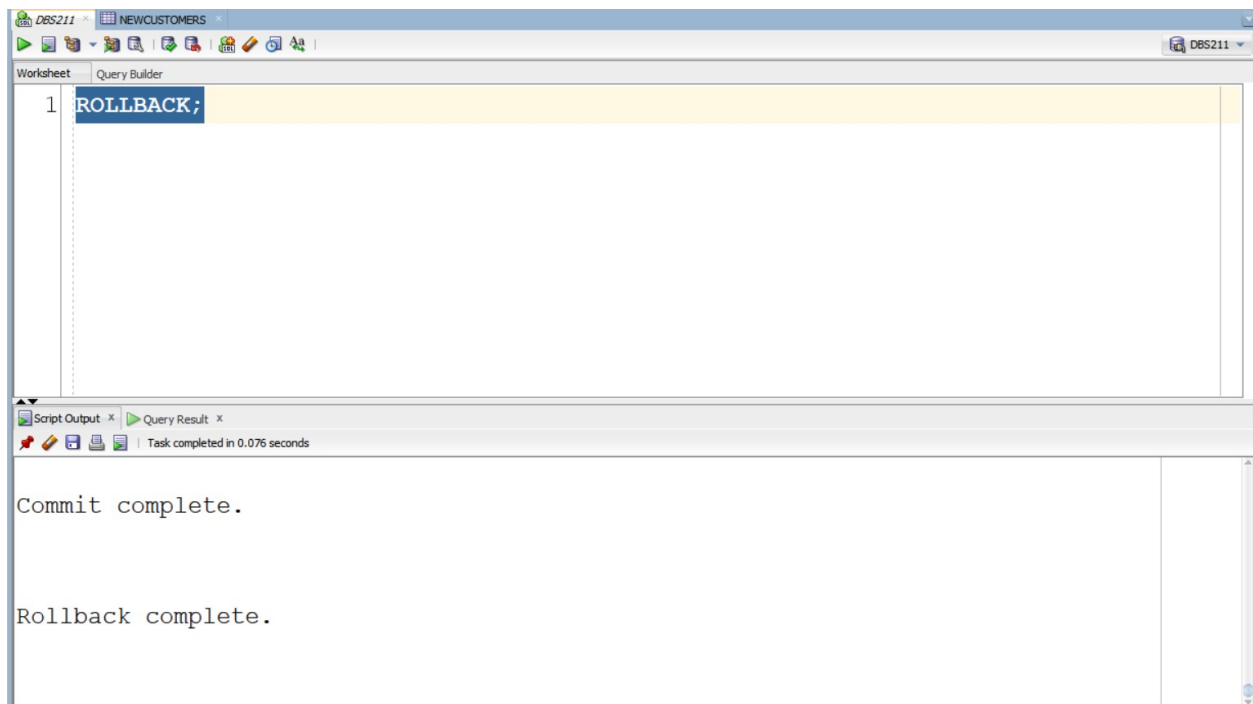
Q7

COMMIT;



Q8

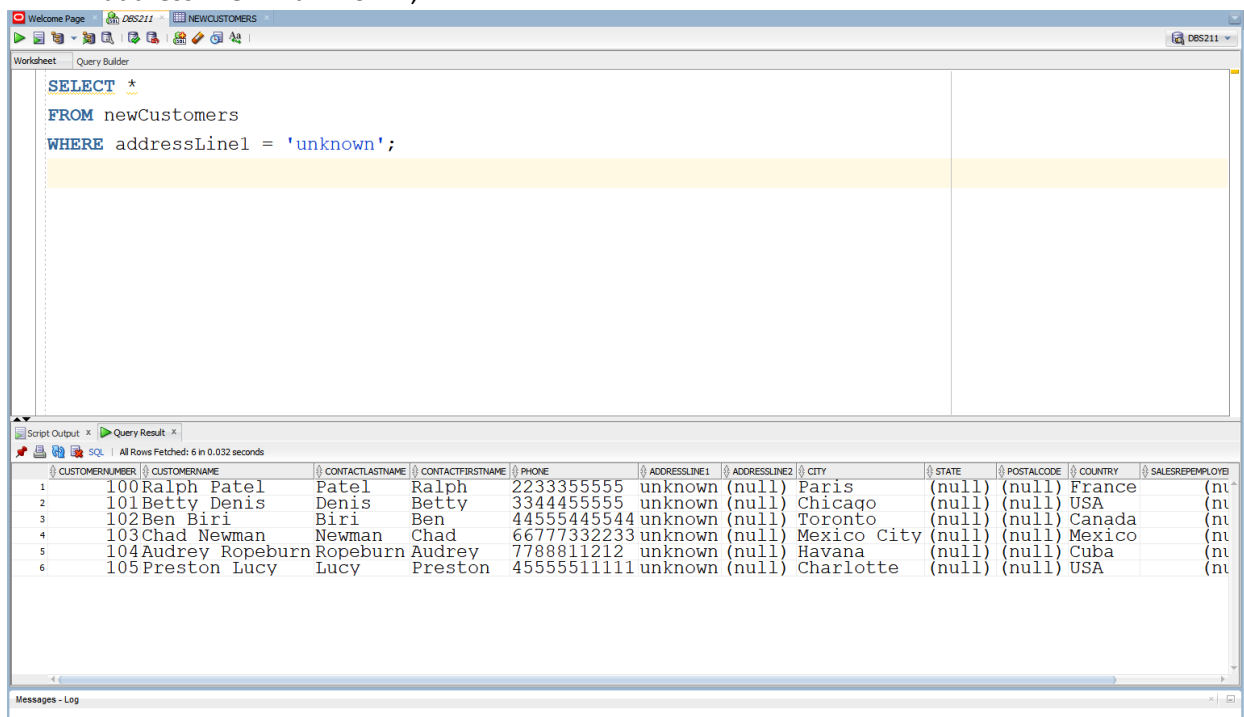
ROLLBACK;



SELECT *

FROM newCustomers

WHERE addressLine1 = 'unknown';



a. Yes, stil updated

- b. It can be effective starting from committing
- c. In the case of question 6, when you roll back, you return to the starting point of the last commit, so no tables are visible. On the other hand, this question is useless because the data is permanently saved because you have already committed by rolling back immediately after committing.

Q9

BEGIN

DELETE FROM newCustomers;

END;

The screenshot displays the Oracle SQL Developer interface. The top window, titled 'NEWCUSTOMERS', shows a SQL script in the 'Query Builder' tab. The script consists of three lines: `BEGIN`, `DELETE FROM newCustomers;`, and `END;`. The `END;` line is highlighted in yellow. Below the script editor, the 'Script Output' and 'Query Result' tabs are visible. The 'Script Output' tab shows the message 'Task completed in 0.095 seconds'. The 'Query Result' tab displays the execution results: '6 rows updated.' and 'PL/SQL procedure successfully completed.'.

The bottom window shows the 'Messages - Log' pane, which is currently empty. The top status bar indicates the database is 'D6S211' and the schema is 'NEWCUSTOMERS'. The bottom status bar shows the 'Messages' tab is active, with 'Logging Page' and 'Statements' tabs also visible.

Q10

BEGIN

DELETE FROM newCustomers;

ROLLBACK;

END;

The screenshot displays the Oracle SQL Developer environment. The main window is titled 'NEWCUSTOMERS' and shows a PL/SQL script in the 'Query Builder' tab. The script is as follows:

```
END;  
  
BEGIN  
  
    DELETE FROM newCustomers;  
    ROLLBACK;  
  
END;
```

Below the script editor, the 'Script Output' window shows the execution results. It indicates that the task was completed in 0.1 seconds and provides the following messages:

```
PL/SQL procedure successfully completed.  
  
6 rows updated.  
  
PL/SQL procedure successfully completed.
```

At the bottom of the interface, the 'Messages - Log' window is visible, showing tabs for 'Messages', 'Logging Page', and 'Statements'.

285211 NEWCUSTOMERS

Columns: Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort: Filter

	CUSTOMERNUMBER	CUSTOMERNAME	CONTACTLASTNAME	CONTACTFIRSTNAME	PHONE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATE	POSTALCODE	COUNTRY	SALESREPEMLOYEENUMBER	CR
1	100	Ralph Patel	Patel	Ralph	2233355555	unknown	(null)	Paris	(null)	(null)	France	(null)	(n)
2	101	Betty Denis	Denis	Betty	3344455555	unknown	(null)	Chicago	(null)	(null)	USA	(null)	(n)
3	102	Ben Biri	Biri	Ben	44555445544	unknown	(null)	Toronto	(null)	(null)	Canada	(null)	(n)
4	103	Chad Newman	Newman	Chad	66777332233	unknown	(null)	Mexico City	(null)	(null)	Mexico	(null)	(n)
5	104	Audrey Ropeburn	Ropeburn	Audrey	7788811212	unknown	(null)	Havana	(null)	(null)	Cuba	(null)	(n)
6	105	Preston Lucy	Lucy	Preston	4555511111	unknown	(null)	Charlotte	(null)	(null)	USA	(null)	(n)

Messages - Log

Messages | Function Data | Statements

Q11

BEGIN

INSERT ALL

INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (100, 'Patel', 'Ralph', '2233355555', '10 Seneca Way', 'Paris', 'France', 'Ralph Patel')

INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (101, 'Denis', 'Betty', '3344455555', '110 Seneca Way', 'Chicago', 'USA', 'Betty Denis')

INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (102, 'Biri', 'Ben', '44555445544', '13000 Seneca Way', 'Toronto', 'Canada', 'Ben Biri')

INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1, city, country, customerName) VALUES (103, 'Newman', 'Chad', '66777332233', '12 Seneca Way', 'Mexico City', 'Mexico', 'Chad Newman')

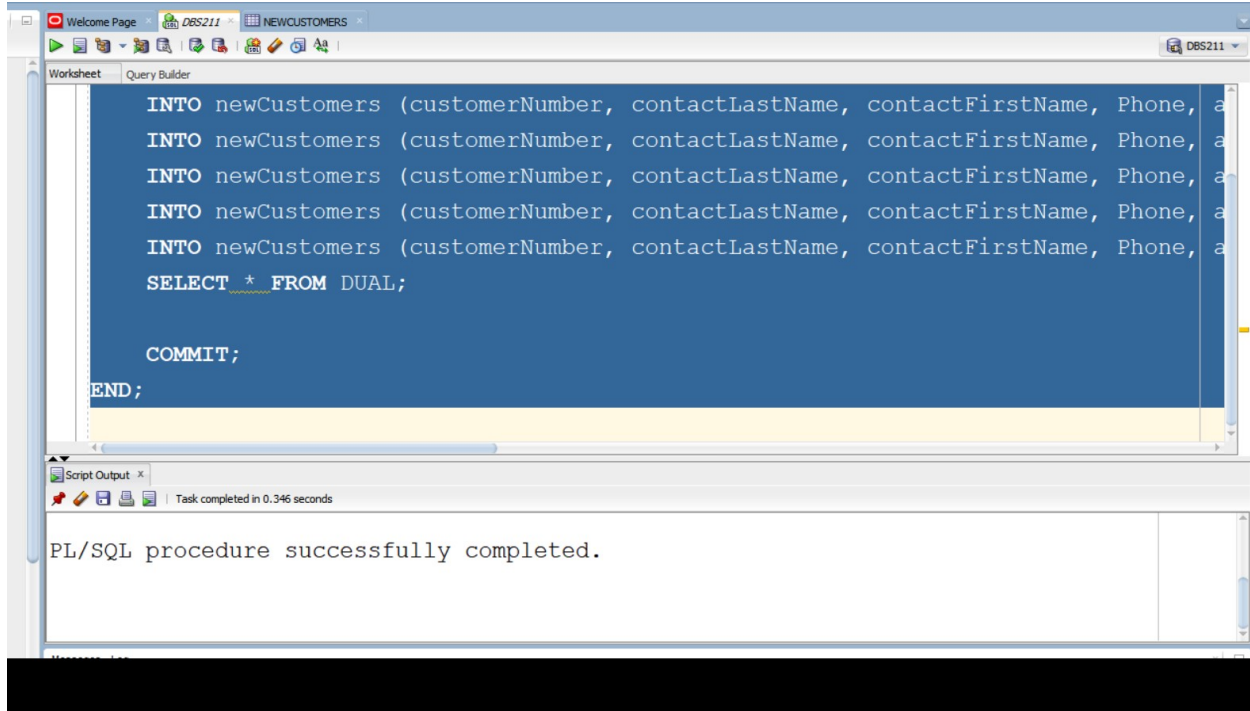
```
    INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1,  
city, country, customerName) VALUES (104, 'Ropeburn', 'Audrey', '7788811212', '15000 Seneca Way',  
'Havana', 'Cuba', 'Audrey Ropeburn')
```

```
    INTO newCustomers (customerNumber, contactLastName, contactFirstName, Phone, addressLine1,  
city, country, customerName) VALUES (105, 'Lucy', 'Preston', '45555511111', '12 Seneca Way', 'Charlotte',  
'USA', 'Preston Lucy')
```

```
SELECT * FROM DUAL;
```

```
COMMIT;
```

END;

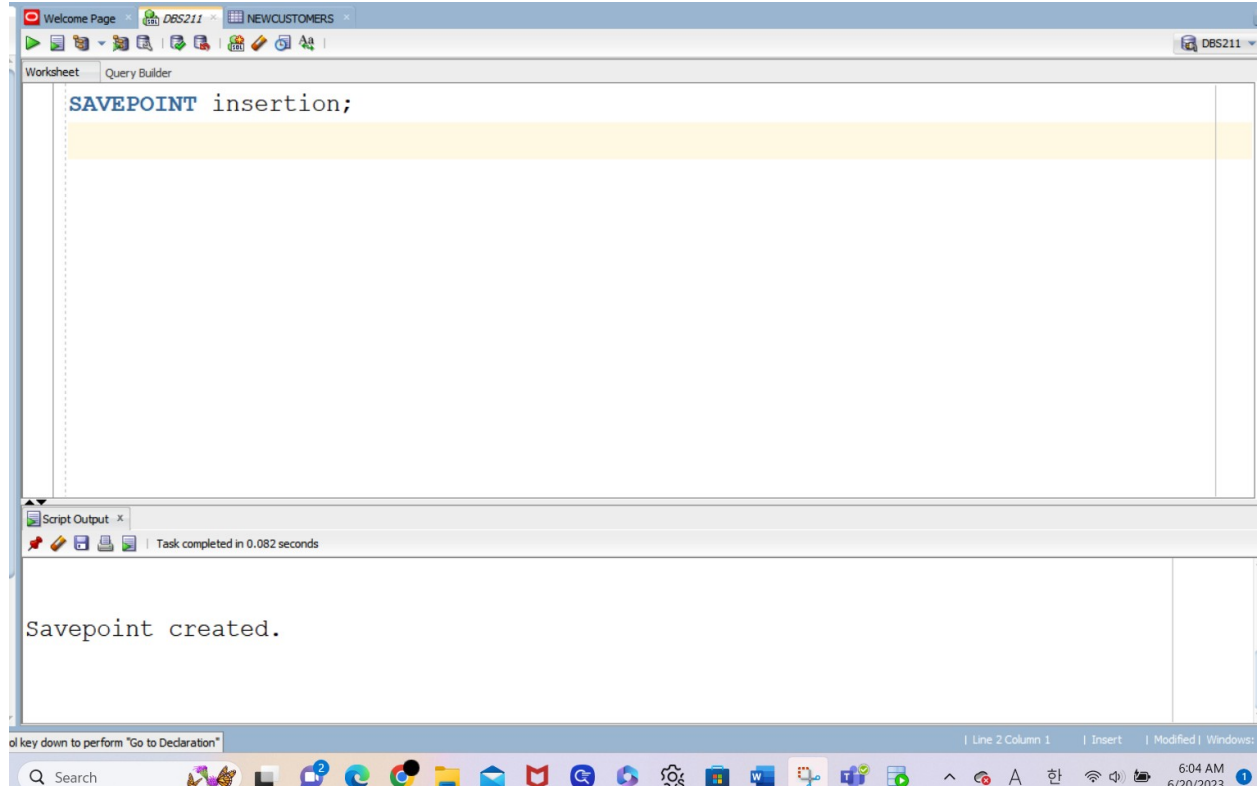


	CUSTOMERNUMBER	CUSTOMERNAME	CONTACTLASTNAME	CONTACTFIRSTNAME	PHONE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATE
1	100	Ralph Patel	Patel	Ralph	2233355555	10 S...	(null)	Paris	(null)
2	101	Betty Denis	Denis	Betty	3344455555	110 ...	(null)	Chicago	(null)
3	102	Ben Biri	Biri	Ben	44555445544	1300...	(null)	Toronto	(null)
4	103	Chad Newman	Newman	Chad	66777332233	12 S...	(null)	Mexico City	(null)
5	104	Audrey Ropeburn	Ropeburn	Audrey	7788811212	1500...	(null)	Havana	(null)
6	105	Preston Lucy	Lucy	Preston	45555511111	12 S...	(null)	Charlotte	(null)

Messages - Log

Messages Logging Page Statements

SAVEPOINT insertion;



Q13

UPDATE newCustomers

SET addressLine1 = 'unknown';

SELECT * FROM newCustomers;

Worksheet Query Builder

```

UPDATE newCustomers
SET addressLine1 = 'unknown';

SELECT * FROM newCustomers;

```

Script Output x Query Result x

SQL | All Rows Fetched: 6 in 0.033 seconds

	CUSTOMERNUMBER	CUSTOMERNAME	CONTACTLASTNAME	CONTACTFIRSTNAME	PHONE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATE
1	100	Ralph Patel	Patel	Ralph	2233355555	unknown	(null)	Paris	(null)
2	101	Betty Denis	Denis	Betty	3344455555	unknown	(null)	Chicago	(null)
3	102	Ben Biri	Biri	Ben	44555445544	unknown	(null)	Toronto	(null)
4	103	Chad Newman	Newman	Chad	66777332233	unknown	(null)	Mexico City	(null)
5	104	Audrey Ropeburn	Ropeburn	Audrey	7788811212	unknown	(null)	Havana	(null)
6	105	Preston Lucy	Lucy	Preston	45555511111	unknown	(null)	Charlotte	(null)

pl key down to perform "Go to Declaration" | Line 5 Column 1 | Insert | Modified | Windows: C | 6:07 AM

Q14

ROLLBACK TO SAVEPOINT insertion;

SELECT * FROM newCustomers;

The screenshot shows a database query tool interface. The top pane displays a SQL script with two statements: `ROLLBACK TO SAVEPOINT insertion;` and `SELECT * FROM newCustomers;`. The second statement is highlighted in yellow. The bottom pane shows the query results for the `SELECT` statement, displaying 6 rows of customer data. The status bar at the bottom indicates 'All Rows Fetched: 6 in 0.031 seconds'.

	CUSTOMERNUMBER	CUSTOMERNAME	CONTACTLASTNAME	CONTACTFIRSTNAME	PHONE	ADDRESSLINE1	ADDRESSLINE2	CITY
1	100	Ralph Patel	Patel	Ralph	2233355555	10 Seneca Way	(null)	Paris
2	101	Betty Denis	Denis	Betty	3344455555	110 Seneca Way	(null)	Chicago
3	102	Ben Biri	Biri	Ben	44555445544	13000 Seneca Way	(null)	Toronto
4	103	Chad Newman	Newman	Chad	66777332233	12 Seneca Way	(null)	Mexico
5	104	Audrey Ropeburn	Ropeburn	Audrey	7788811212	15000 Seneca Way	(null)	Havana
6	105	Preston Lucy	Lucy	Preston	45555511111	12 Seneca Way	(null)	Charlot

After executing the `ROLLBACK TO SAVEPOINT insertion;` statement, the transaction is rolled back to the savepoint named "insertion." This means that all changes made after the savepoint, including the update statement from Question 6, are undone, and the data is reverted to its state at the savepoint.

As a result, the `addressLine1` column of the `newCustomers` table is restored to its previous values, which were present when the savepoint was set. The update that set all `addressLine1` values to 'unknown' is rolled back.

Q15

ROLLBACK;

The screenshot shows a database management tool interface. The top bar includes tabs for 'Welcome Page', 'DBS211', and 'NEWCUSTOMERS'. Below the top bar, there's a 'Worksheet' tab and a 'Query Builder' tab. The main area displays a blue header with the text 'ROLLBACK;' and a yellow background. Below this, a 'Script Output' tab is active, showing the message 'Rollback complete.' and 'Task completed in 0.056 seconds'. The bottom status bar indicates 'Line 2 Column 1' and 'Insert' mode.

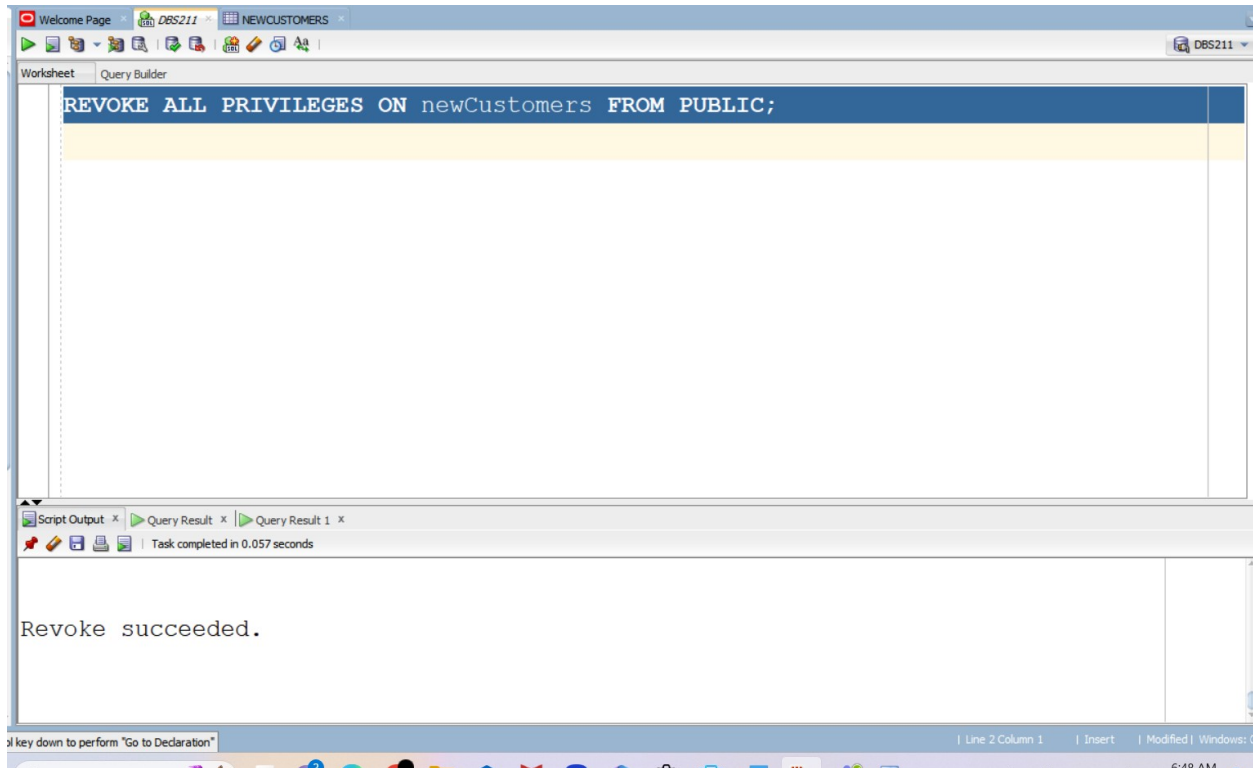
The screenshot shows a database management tool interface. The top bar includes tabs for 'Welcome Page', 'DBS211', and 'NEWCUSTOMERS'. Below the top bar, there's a 'Columns' tab and a 'Data' tab. The main area displays a table with the following columns: CUSTOMERNUMBER, CUSTOMERNAME, CONTACTLASTNAME, CONTACTFIRSTNAME, PHONE, ADDRESSLINE1, ADDRESSLINE2, CITY, and STATE. The table contains 6 rows of data:

CUSTOMERNUMBER	CUSTOMERNAME	CONTACTLASTNAME	CONTACTFIRSTNAME	PHONE	ADDRESSLINE1	ADDRESSLINE2	CITY	STATE
100	Ralph Patel	Patel	Ralph	2233355555	10 S...	(null)	Paris	(null)
101	Betty Denis	Denis	Betty	3344455555	110 ...	(null)	Chicago	(null)
102	Ben Biri	Biri	Ben	44555445544	1300...	(null)	Toronto	(null)
103	Chad Newman	Newman	Chad	66777332233	12 S...	(null)	Mexico City	(null)
104	Audrey Ropeburn	Ropeburn	Audrey	7788811212	1500...	(null)	Havana	(null)
105	Preston Lucy	Lucy	Preston	45555511111	12 S...	(null)	Charlotte	(null)

If we execute the ROLLBACK statement without specifying a savepoint, it will roll back the entire transaction to its starting point, discarding all changes made during the transaction. This includes any insertions, updates, or deletions performed within the transaction. After executing the ROLLBACK statement, if you run the SELECT * FROM newCustomers query again, you will see the data in its original state before the transaction began. All modifications made within the transaction, including the initial data insertion and any subsequent updates, will be discarded. The results will show the original data that existed in the newCustomers table before the transaction started. Any changes made during the transaction will no longer be reflected in the table.

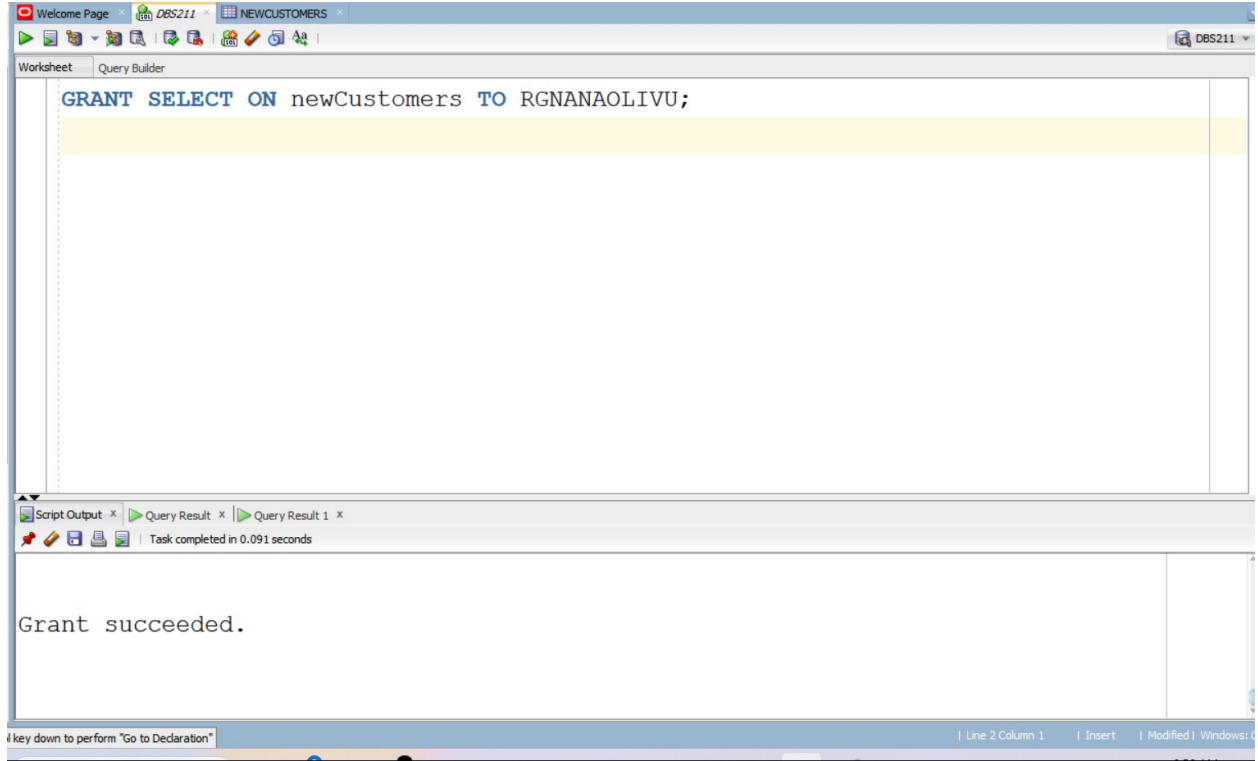
Q16

REVOKE ALL PRIVILEGES ON newCustomers FROM PUBLIC;



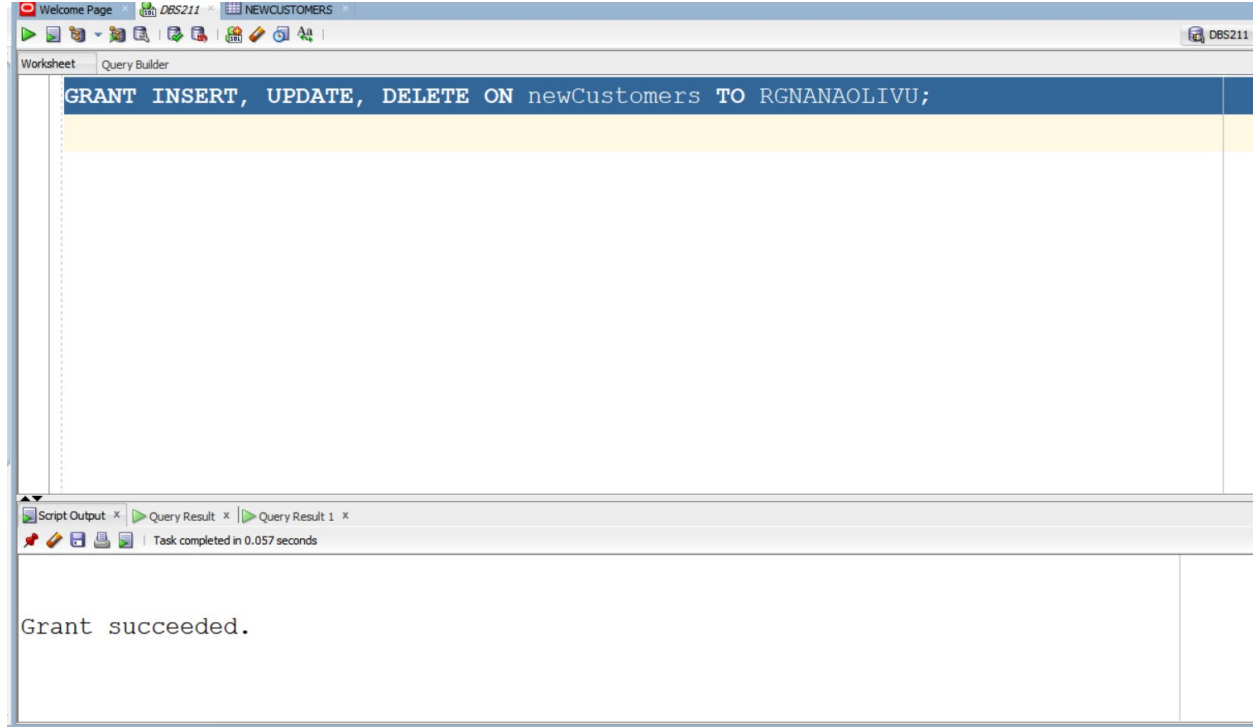
Q17

GRANT SELECT ON newCustomers TO RGNANAOLIVU;



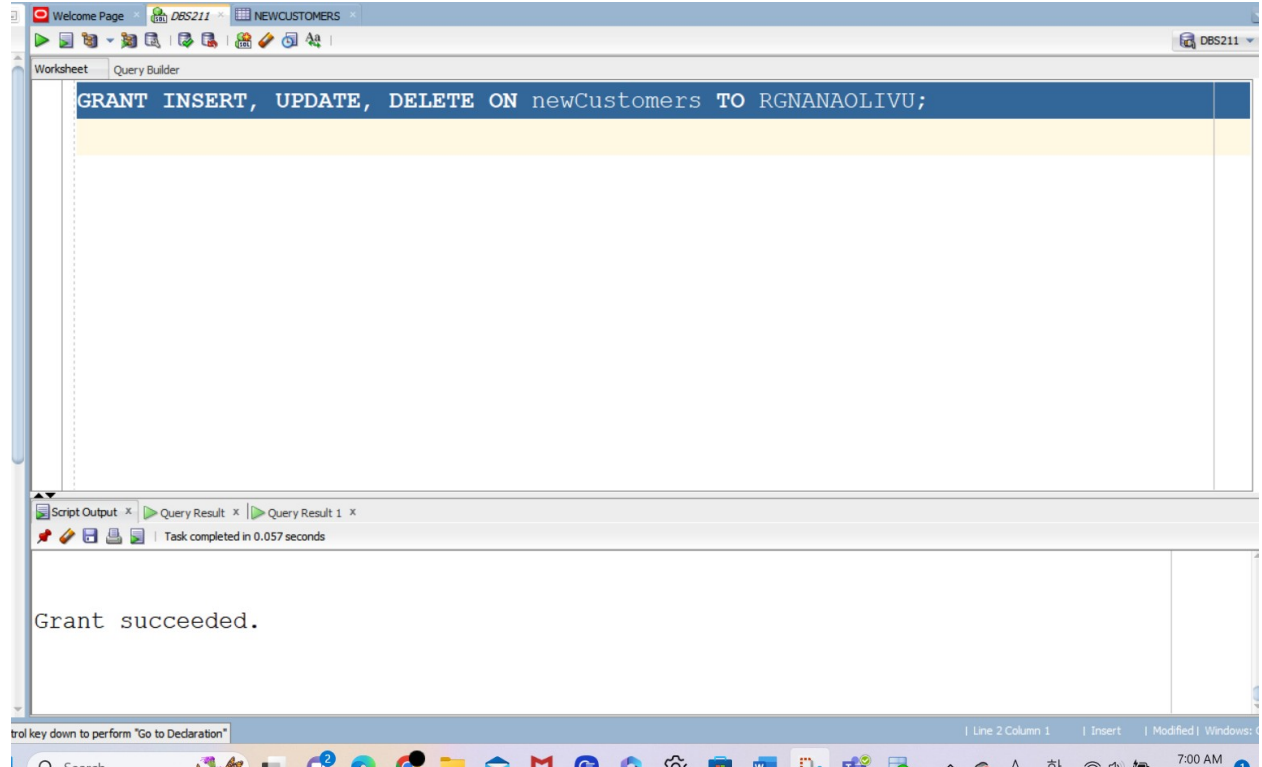
Q18

GRANT INSERT, UPDATE, DELETE ON newCustomers TO RGNANAOLIVU;



Q19

GRANT INSERT, UPDATE, DELETE ON newCustomers TO RGNANAOLIVU;



Q20

DROP TABLE NEWCUSTOMERS;

COMMIT;

The screenshot shows a database management tool interface. At the top, there are tabs for 'Welcome Page', 'DBS211', and 'NEWCUSTOMERS'. Below the tabs is a toolbar with various icons. The main area is divided into two sections: 'Worksheet' and 'Query Builder'. The 'Worksheet' section contains the following SQL code:

```
DROP TABLE NEWCUSTOMERS;  
COMMIT;
```

The 'Query Builder' section is empty. Below the main area is a 'Script Output' section. It shows the following output:

```
Table NEWCUSTOMERS dropped.  
  
Commit complete.
```

At the bottom of the interface, there is a 'Messages - Log' section.