COMP.6212

Small Business Database (SQL Server 2019)

Codie Shannon
30013375

Table of Contents

Database Scenario

Jodie Glossop is the current circular newspaper supervisor of Whakatane, Bay of Plenty, New Zealand. She has requested my assistance in developing a way to store and access data more efficiently.

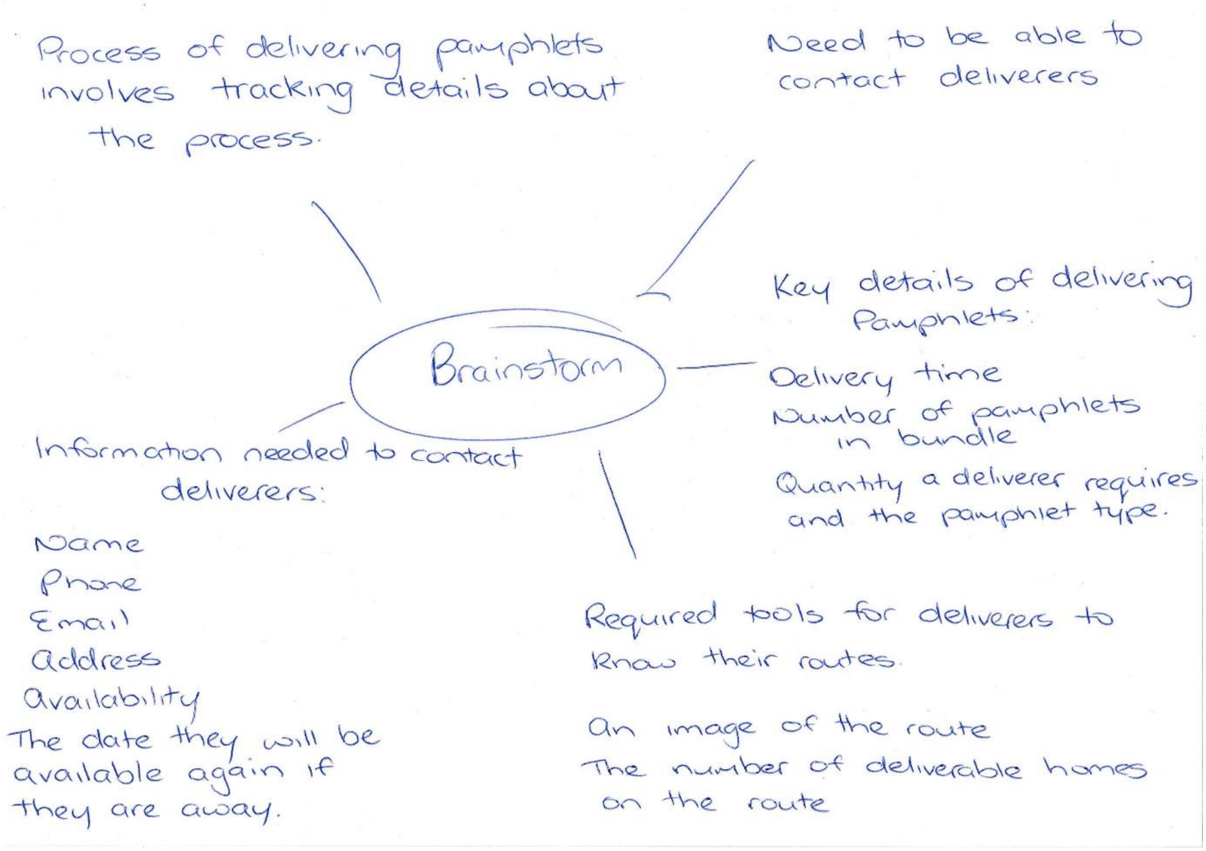
Jodie is employed by Ovato Distribution, a circular-newspaper enterprise that operates throughout the entirety of New Zealand. Ovato oversees the production and delivery nationwide to supervisors who then distribute onto deliverers thus ensuring coverage to the whole of New Zealand.

Supervisors are self-employed and therefore they are required to store data relating to deliverers and delivery addresses such as names, addresses, contact details and delivery territories which can be difficult to keep organised as the deliverers are constantly changing and Ovato has provided no storage solution to this problem.

Supervisors are presently utilizing an unreliable paper-based record system to keep track of their data. The paper-based record system consists of many problems such as data loss and damage, alongside time consumption attempting to find specific segments of data. The database I aim to develop will provide a simplistic system for supervisors, solving many of these issues, alongside providing a schematic place to organise and delete data as needed.

Research Thinking

In order to understand the business, I had a meeting with Jodie for an hour. We discussed the processes for her business, alongside current issues surrounding the management of data for supervisors within Ovato. Following this, we discussed what important data should be included in the database. We then created a brainstorm which reflects this discussion.



Brainstorm

Process of delivering pamphlets involves tracking details about the process.

Need to be able to contact deliverers

Key details of delivering Pamphlets:
Delivery time
Number of pamphlets in bundle
Quantity a deliverer requires and the pamphlet type.

Information needed to contact deliverers:
Name
Phone
Email
Address
Availability
The date they will be available again if they are away.

Required tools for deliverers to know their routes.

An image of the route
The number of deliverable homes on the route

Following the meeting with Jodie and the creation of the brainstorm, I did some research for a suitable database template. However, I was unable to locate an adequate template for the processes and data laid out in the brainstorm. Ensuing this, I designed a database based upon my knowledge attained in class.

Database Specifications

Ovato Distribution supervisors experience storage issues as they are reliant on an unreliable paper-based record system causing issues such as data loss and damage, alongside time consumption attempting to find specific segments of data.

The database will seek to solve these issues by providing a structured environment for supervisors to effortlessly create, read, update, and delete data pertaining to maps, deliveries, and contact details of deliverers. Supervisors should be able to simply query the database for the available deliverers, unfulfilled deliveries, and the contact information for deliverers.

The database will be comprised of five tables, Maps, Contacts, Deliverers, PamphletType and Deliveries.

The Maps table will contain three fields, MapsID, MapImg, and Deliverables. The MapID will be an integer, auto-incremented, primary key that is a unique identifier for each Map record within the table. MapImg will be of NVARCHAR data type and contain maps to delivery territories. Deliverables will be an int data type and contain the number of available homes for delivery within the map territories.

Table of Contacts will consist of nine fields, ContactsID, isAvailable, DateAvail, FirstName, LastName, Email, Phone, Street, and City. ContactsID will be an int, auto-incremented, primary key used to uniquely identify each record in the Contacts table. isAvailable will be of boolean data type and contain the value of true or false, representing the contacts availability i.e., present/away. DateAvail is the accompanying data value for Available, which accommodates the date in which the contact will be next available. The fields of FirstName, LastName, and Email will be of NVARCHAR data type and contain the correlated data about deliverers. Alongside this, Phone, Street, and City will be of VARCHAR data type and contain the corresponding data.

The Deliverers table will be comprised of the following fields, DeliverersID, MapsID, and ContactsID. The DeliverersID will be an int, auto-incremented, primary key that uniquely identifies each record within the table. MapsID will be of int data type and act as a foreign key for the MapsID field within the Maps table functioning as a link connecting the Maps table's records to the corresponding records in Deliverers. ContactsID will be an int data type, behaving as a foreign key for the ContactsID field within the Contacts table acting as a link attaching the Contacts table's records to the corresponding records in Deliverers.

Table of PamphletTypes will consist of two fields, PamType, and BundleQuantity. PamType will be of varchar, primary key data type, and act as a foreign key for the PamType field within the PamphletTypes table functioning as a link attaching the PamphletTypes table's records to the corresponding records in Deliveries. Alongside this, the field will also act as a validation tool for the PamType in Deliveries. The tool will draw a comparison between any value entered in to PamType within Deliveries, to check if it is of valid value (contained within the PamType field within the PampheletTypes table). BundleQuantity will be of data type int and specify the number of papers in each bundle of pamphlets.

The Deliveries table will contain ten fields, DeliveriesID, DeliverersID, isDelivered, DelDate, and six PamType fields. DeliveriesID will be an int, auto-incremented, primary key utilized to individually identify each record within the Deliveries table. The DeliverersID field will be of int data type, and function as a foreign key for the DeliverersID field within the Deliverers table acting as a link attaching the Deliverers table's records to the corresponding records in Deliveries. isDelivered will be of boolean data type and contain a of true or false value, representing whether the pamphlets have been delivered to a certain deliverer. DelDate will be the counterpart to Delivered and contain the date in which the pamphlets were delivered. The PamType fields will be of varchar datatype and specify the names of the pamphlets, alongside attaching the pamphlet's BundleQuantity from PamphletType.

The creation of this database will hopefully give supervisors peace of mind knowing that their data is securely stored away, alongside providing them flexibility, and faster and further efficiency while accessing data.

Data Structure

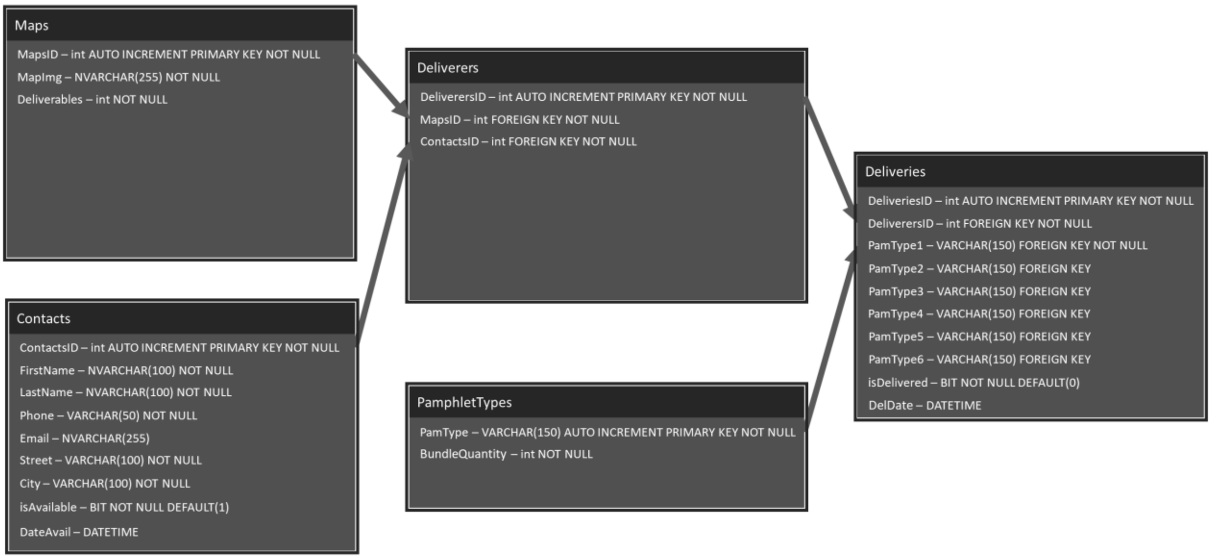
The database structure design contains a few noteworthy datatypes and relationships. The noteworthy datatypes and relationships of the design are the MapImg field of the Maps table, the Phone and isAvailable fields of the Contacts table, and the isDelivered and PamType fields of the Deliveries table, alongside the relationship between the Deliveries table and PamphletType table.

The MapImg field of the Maps table is of datatype nvarchar as it was a special condition where no adequate datatypes were available for the storing of images. The MapImg field will store image URLs for delivery territories.

The Phone field of the Contacts table is of varchar datatype. The field will store the phone number of deliverers. The datatype is of type varchar as it may start with the character zero. Number datatypes do not allow for the beginning character of cell values to start with zero.

The isAvailable field of the Contacts table and isDelivered field of the Deliveries table are of datatype BIT. These fields can contain the values 0 or 1 representing false or true. The fields are created with this datatype as there is no direct boolean datatype within SQL.

The PamType fields of the PamphletType, and Deliveries tables are a specific case where a data restriction was put in place for the validation of entered values within the PamType fields of the Deliveries table. When data is stored within the PamType fields of the Deliveries table, the entered value will first be compared to the values in PamType of the PamphletType table. If a match is not located, the database will reject the entered data and not store it within the database.



Maps

MapsID – int AUTO INCREMENT PRIMARY KEY NOT NULL
MapImg – NVARCHAR(255) NOT NULL
Deliverables – int NOT NULL

Deliverers

DeliverersID – int AUTO INCREMENT PRIMARY KEY NOT NULL
MapsID – int FOREIGN KEY NOT NULL
ContactsID – int FOREIGN KEY NOT NULL

Deliveries

DeliveriesID – int AUTO INCREMENT PRIMARY KEY NOT NULL
DeliverersID – int FOREIGN KEY NOT NULL
PamType1 – VARCHAR(150) FOREIGN KEY NOT NULL
PamType2 – VARCHAR(150) FOREIGN KEY
PamType3 – VARCHAR(150) FOREIGN KEY
PamType4 – VARCHAR(150) FOREIGN KEY
PamType5 – VARCHAR(150) FOREIGN KEY
PamType6 – VARCHAR(150) FOREIGN KEY
isDelivered – BIT NOT NULL DEFAULT(0)
DelDate – DATETIME

Contacts

ContactsID – int AUTO INCREMENT PRIMARY KEY NOT NULL
FirstName – NVARCHAR(100) NOT NULL
LastName – NVARCHAR(100) NOT NULL
Phone – VARCHAR(50) NOT NULL
Email – NVARCHAR(255)
Street – VARCHAR(100) NOT NULL
City – VARCHAR(100) NOT NULL
isAvailable – BIT NOT NULL DEFAULT(1)
DateAvail – DATETIME

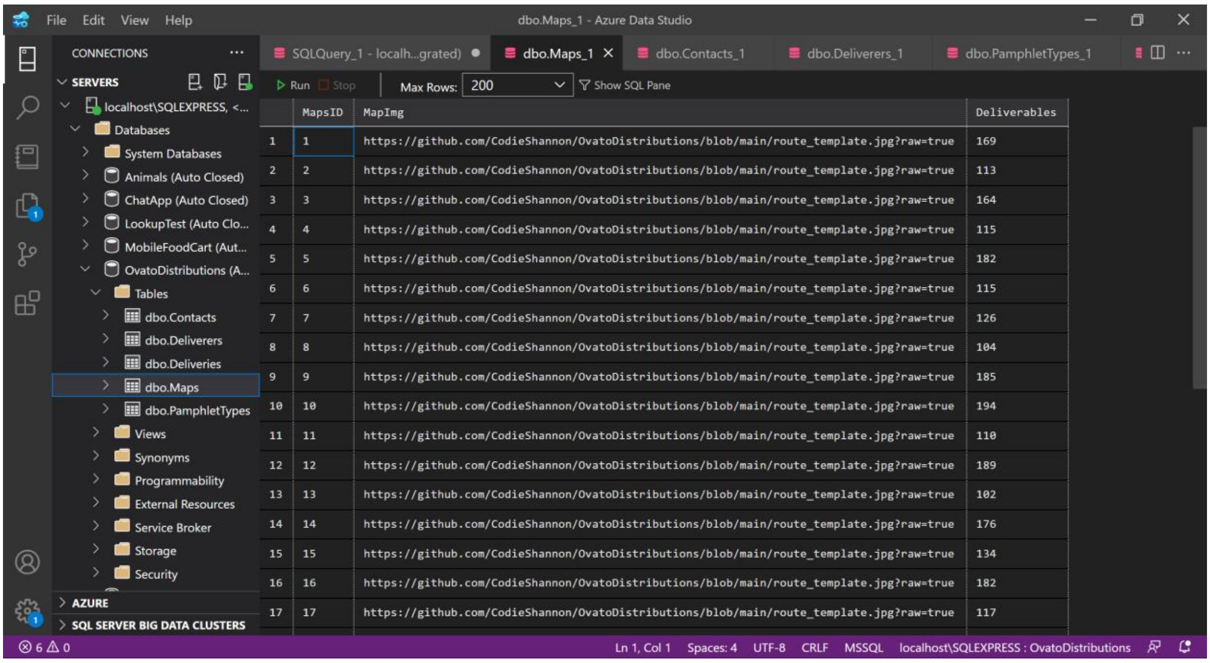
PamphletTypes

PamType – VARCHAR(150) AUTO INCREMENT PRIMARY KEY NOT NULL
BundleQuantity – int NOT NULL

Build Evidence

Tables

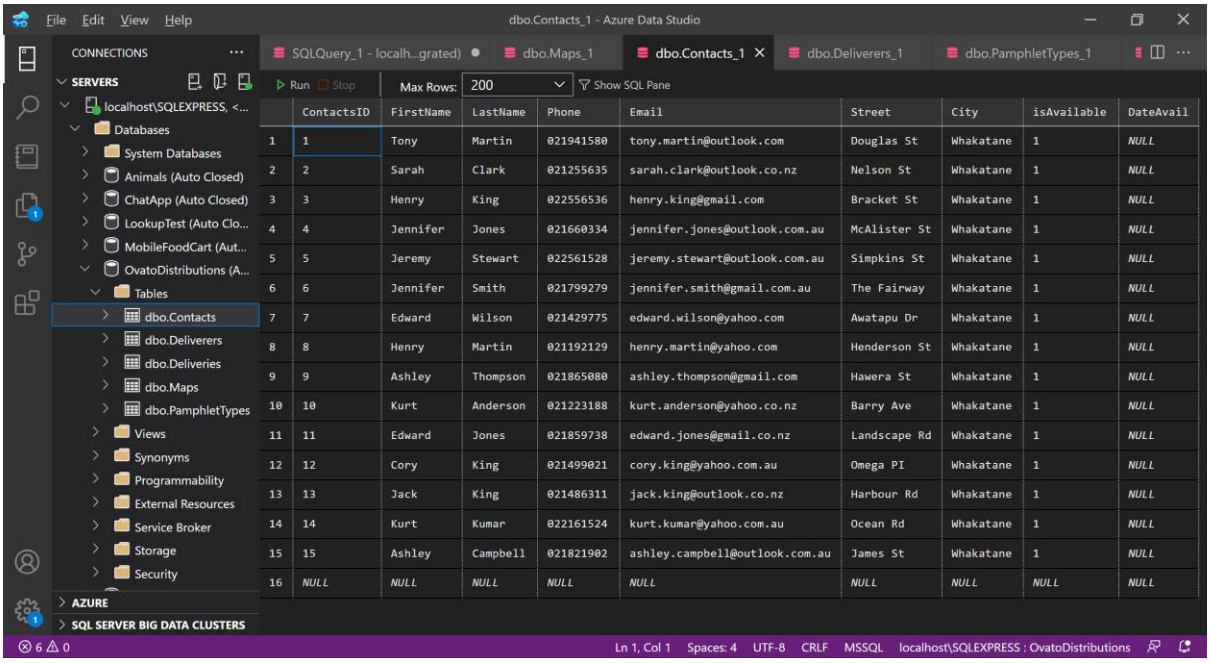
Maps Table



Max Rows: 200

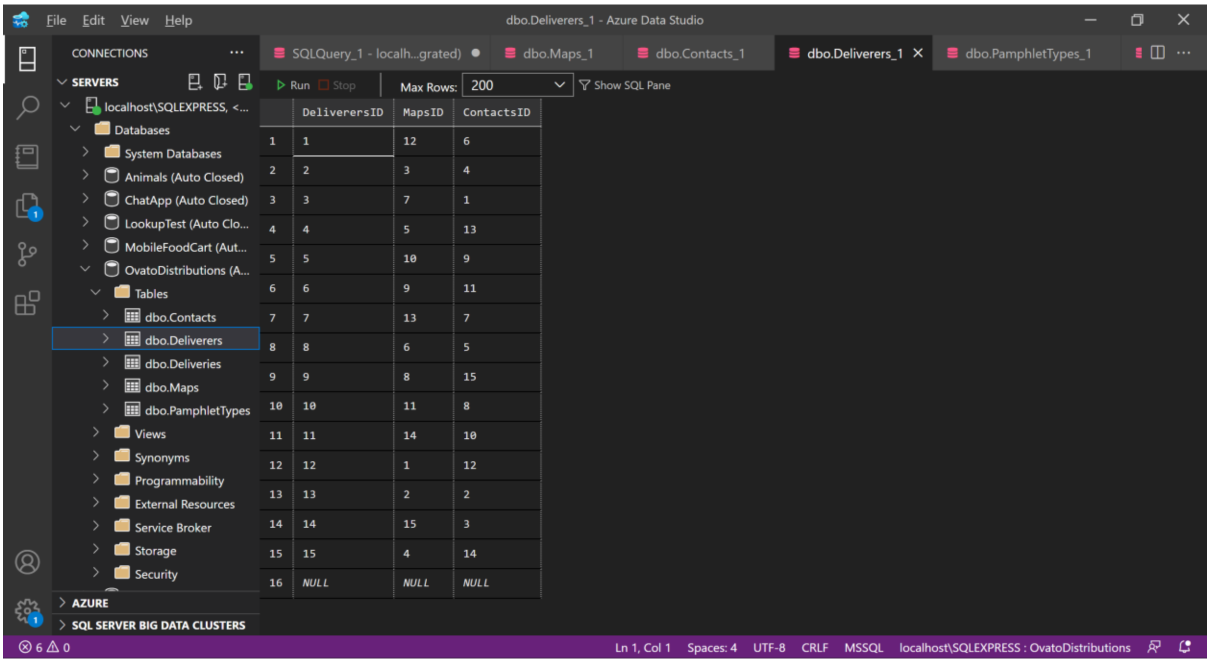
	MapsID	MapImg	Deliverables
1	1	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	169
2	2	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	113
3	3	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	164
4	4	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	115
5	5	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	182
6	6	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	115
7	7	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	126
8	8	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	104
9	9	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	185
10	10	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	194
11	11	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	110
12	12	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	189
13	13	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	102
14	14	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	176
15	15	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	134
16	16	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	182
17	17	https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template.jpg?raw=true	117

Contacts Table



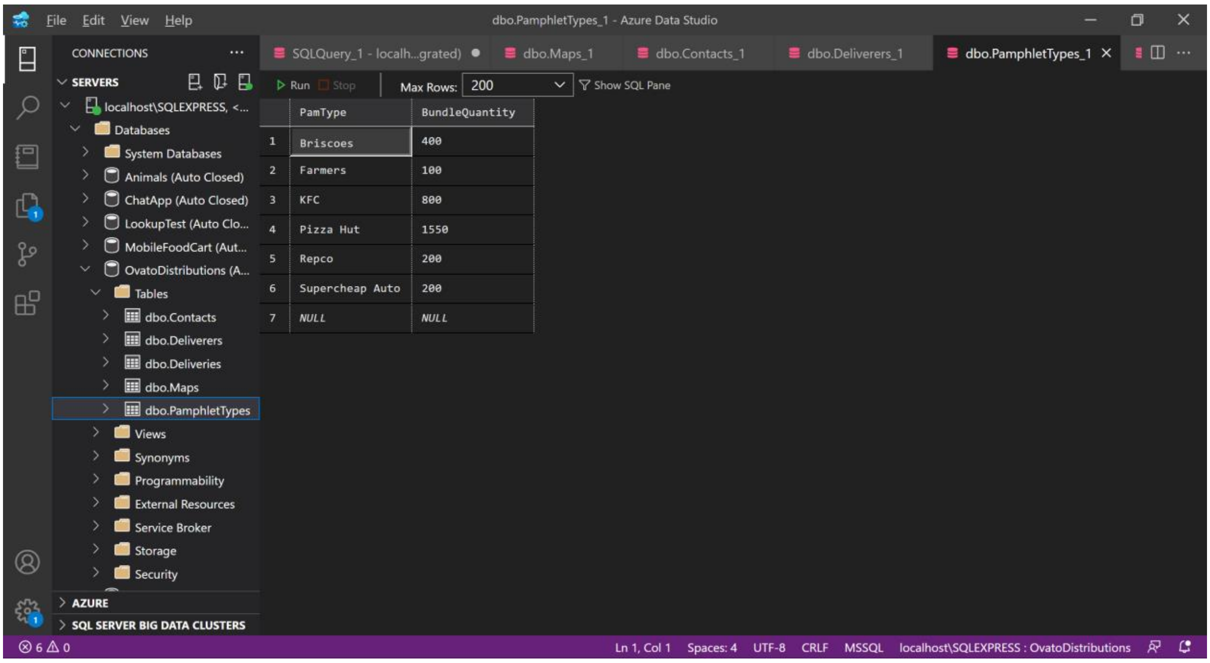
	ContactsID	FirstName	LastName	Phone	Email	Street	City	isAvailable	DateAvail
1	1	Tony	Martin	021941580	tony.martin@outlook.com	Douglas St	Whakatane	1	NULL
2	2	Sarah	Clark	021255635	sarah.clark@outlook.co.nz	Nelson St	Whakatane	1	NULL
3	3	Henry	King	022556536	henry.king@gmail.com	Bracket St	Whakatane	1	NULL
4	4	Jennifer	Jones	021660334	jennifer.jones@outlook.com.au	McAlister St	Whakatane	1	NULL
5	5	Jeremy	Stewart	022561528	jeremy.stewart@outlook.com.au	Simpkins St	Whakatane	1	NULL
6	6	Jennifer	Smith	021799279	jennifer.smith@gmail.com.au	The Fairway	Whakatane	1	NULL
7	7	Edward	Wilson	021429775	edward.wilson@yahoo.com	Awatapu Dr	Whakatane	1	NULL
8	8	Henry	Martin	021192129	henry.martin@yahoo.com	Henderson St	Whakatane	1	NULL
9	9	Ashley	Thompson	021865080	ashley.thompson@gmail.com	Hawera St	Whakatane	1	NULL
10	10	Kurt	Anderson	021223188	kurt.anderson@yahoo.co.nz	Barry Ave	Whakatane	1	NULL
11	11	Edward	Jones	021859738	edward.jones@gmail.co.nz	Landscape Rd	Whakatane	1	NULL
12	12	Cory	King	021499021	cory.king@yahoo.com.au	Omega PI	Whakatane	1	NULL
13	13	Jack	King	021486311	jack.king@outlook.co.nz	Harbour Rd	Whakatane	1	NULL
14	14	Kurt	Kumar	022161524	kurt.kumar@yahoo.com.au	Ocean Rd	Whakatane	1	NULL
15	15	Ashley	Campbell	021821902	ashley.campbell@outlook.com.au	James St	Whakatane	1	NULL
16	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Deliverers Table



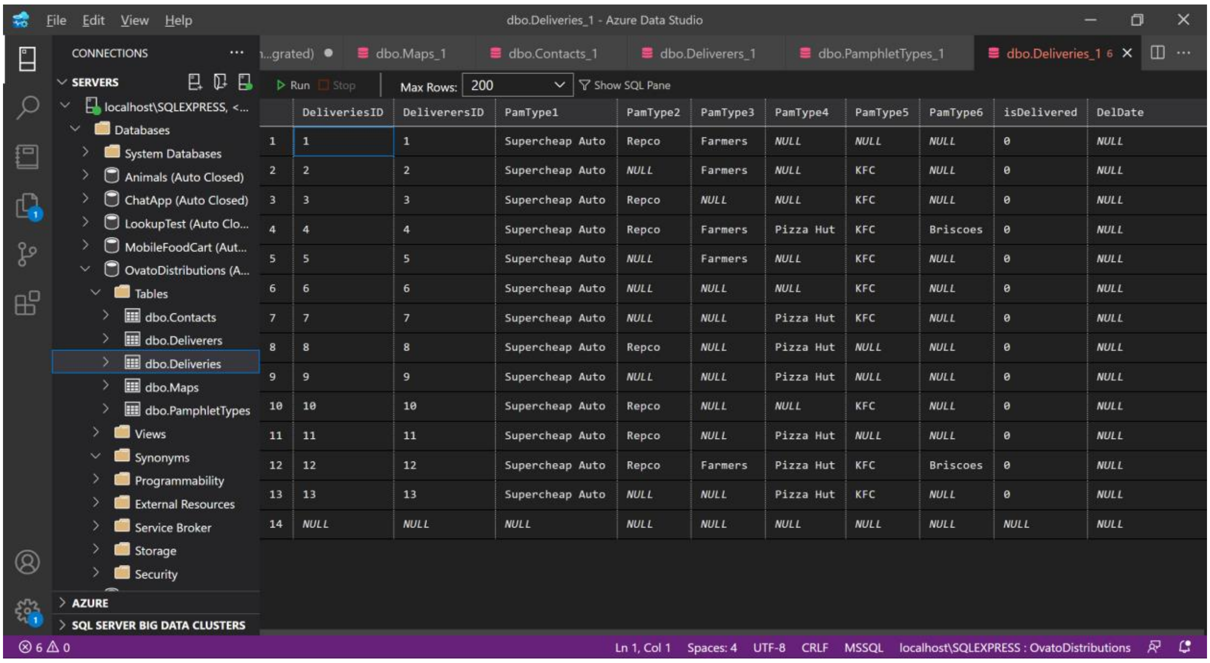
	DeliverersID	MapsID	ContactsID
1	1	12	6
2	2	3	4
3	3	7	1
4	4	5	13
5	5	10	9
6	6	9	11
7	7	13	7
8	8	6	5
9	9	8	15
10	10	11	8
11	11	14	10
12	12	1	12
13	13	2	2
14	14	15	3
15	15	4	14
16	NULL	NULL	NULL

PamphletTypes Table



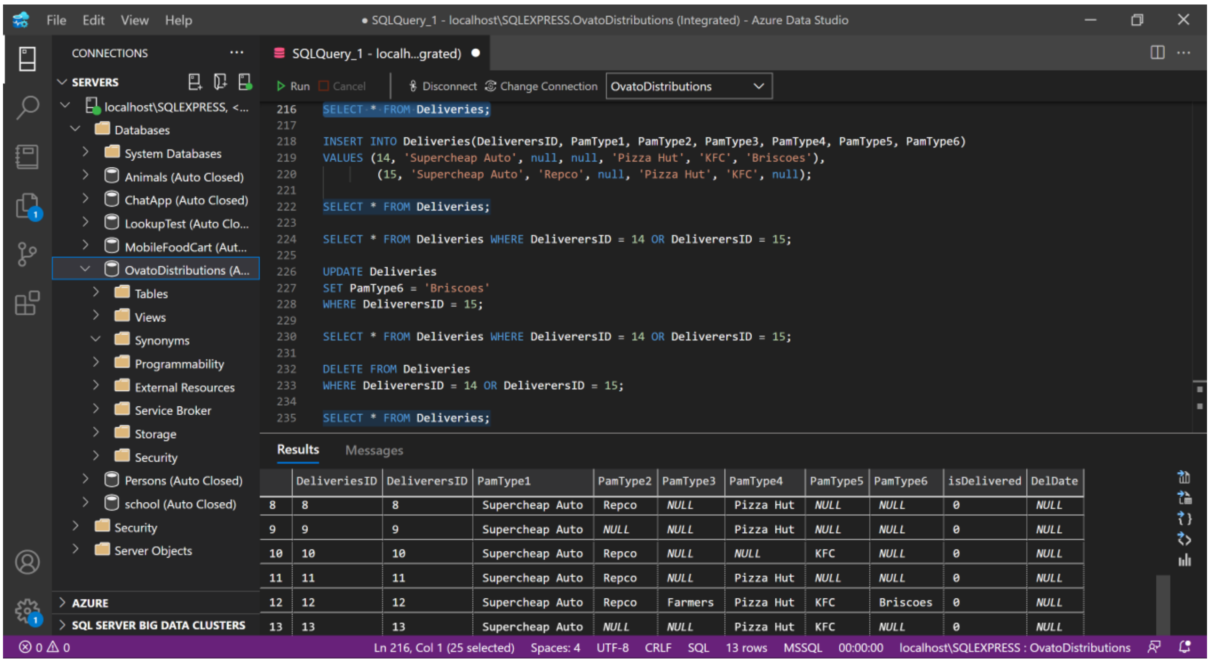
	PamType	BundleQuantity
1	Briscoes	400
2	Farmers	100
3	KFC	800
4	Pizza Hut	1550
5	Repco	200
6	Supercheap Auto	200
7	*NULL*	*NULL*

Deliveries Table



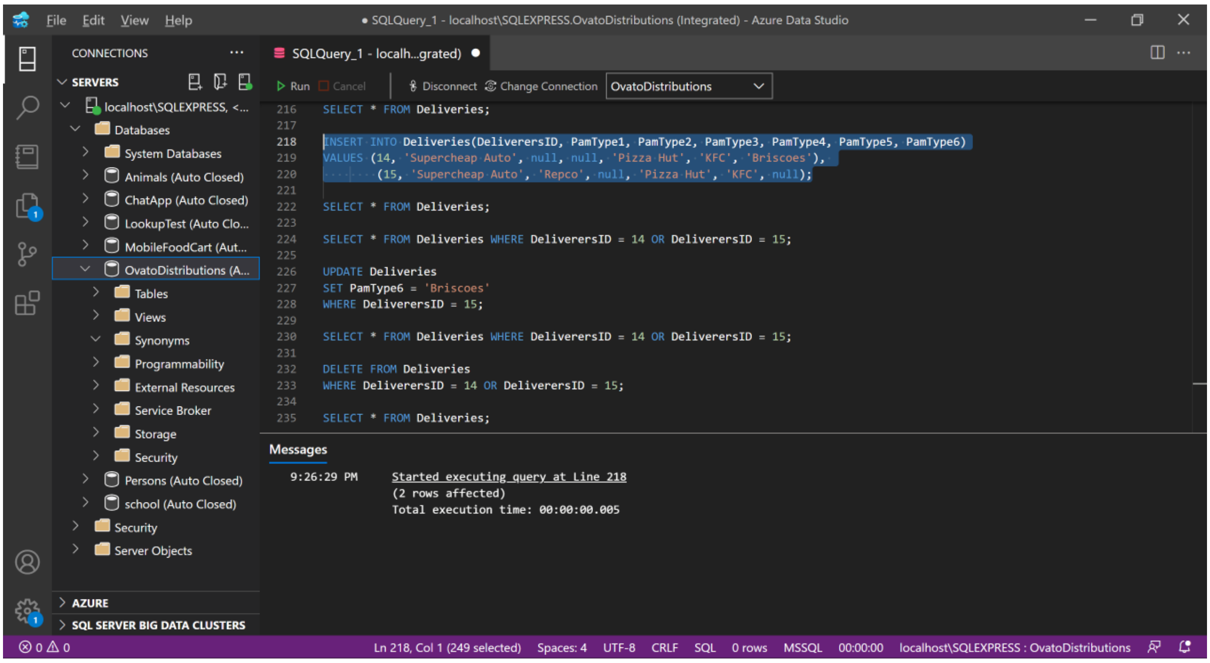
	DeliveriesID	DeliverersID	PamType1	PamType2	PamType3	PamType4	PamType5	PamType6	isDelivered	DelDate
1	1	1	Supercheap Auto	Repco	Farmers	*NULL*	*NULL*	*NULL*	0	*NULL*
2	2	2	Supercheap Auto	*NULL*	Farmers	*NULL*	KFC	*NULL*	0	*NULL*
3	3	3	Supercheap Auto	Repco	*NULL*	*NULL*	KFC	*NULL*	0	*NULL*
4	4	4	Supercheap Auto	Repco	Farmers	Pizza Hut	KFC	Briscoes	0	*NULL*
5	5	5	Supercheap Auto	*NULL*	Farmers	*NULL*	KFC	*NULL*	0	*NULL*
6	6	6	Supercheap Auto	*NULL*	*NULL*	*NULL*	KFC	*NULL*	0	*NULL*
7	7	7	Supercheap Auto	*NULL*	*NULL*	Pizza Hut	KFC	*NULL*	0	*NULL*
8	8	8	Supercheap Auto	Repco	*NULL*	Pizza Hut	*NULL*	*NULL*	0	*NULL*
9	9	9	Supercheap Auto	*NULL*	*NULL*	Pizza Hut	*NULL*	*NULL*	0	*NULL*
10	10	10	Supercheap Auto	Repco	*NULL*	*NULL*	KFC	*NULL*	0	*NULL*
11	11	11	Supercheap Auto	Repco	*NULL*	Pizza Hut	*NULL*	*NULL*	0	*NULL*
12	12	12	Supercheap Auto	Repco	Farmers	Pizza Hut	KFC	Briscoes	0	*NULL*
13	13	13	Supercheap Auto	*NULL*	*NULL*	Pizza Hut	KFC	*NULL*	0	*NULL*
14	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

CRUD Functionality



```sql
SELECT * FROM Deliveries;

INSERT INTO Deliveries(DeliverersID, PamType1, PamType2, PamType3, PamType4, PamType5, PamType6)
VALUES (14, 'Supercheap Auto', null, null, 'Pizza Hut', 'KFC', 'Briscoes'),
        (15, 'Supercheap Auto', 'Repco', null, 'Pizza Hut', 'KFC', null);

SELECT * FROM Deliveries;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

UPDATE Deliveries
SET PamType6 = 'Briscoes'
WHERE DeliverersID = 15;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

DELETE FROM Deliveries
WHERE DeliverersID = 14 OR DeliverersID = 15;

SELECT * FROM Deliveries;
```

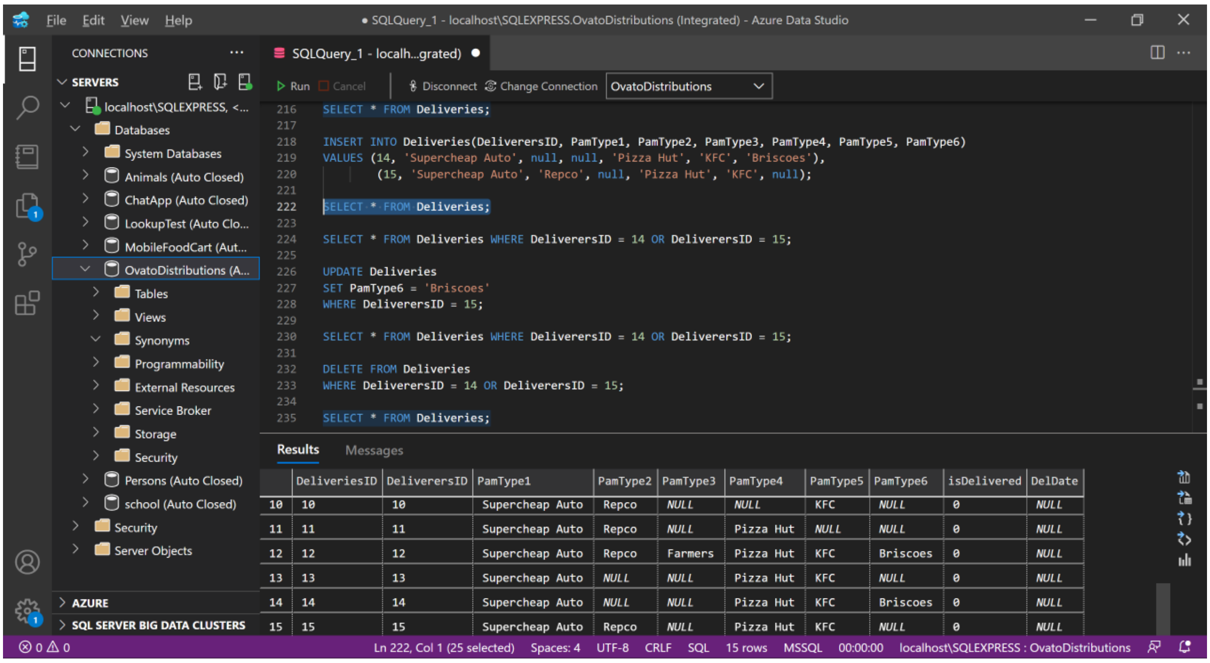
	DeliveriesID	DeliverersID	PamType1	PamType2	PamType3	PamType4	PamType5	PamType6	isDelivered	DelDate
8	8	8	Supercheap Auto	Repco	NULL	Pizza Hut	NULL	NULL	0	NULL
9	9	9	Supercheap Auto	NULL	NULL	Pizza Hut	NULL	NULL	0	NULL
10	10	10	Supercheap Auto	Repco	NULL	NULL	KFC	NULL	0	NULL
11	11	11	Supercheap Auto	Repco	NULL	Pizza Hut	NULL	NULL	0	NULL
12	12	12	Supercheap Auto	Repco	Farmers	Pizza Hut	KFC	Briscoes	0	NULL
13	13	13	Supercheap Auto	NULL	NULL	Pizza Hut	KFC	NULL	0	NULL



```sql
SELECT * FROM Deliveries;

INSERT INTO Deliveries(DeliverersID, PamType1, PamType2, PamType3, PamType4, PamType5, PamType6)
VALUES (14, 'Supercheap Auto', null, null, 'Pizza Hut', 'KFC', 'Briscoes'),
        (15, 'Supercheap Auto', 'Repco', null, 'Pizza Hut', 'KFC', null);

SELECT * FROM Deliveries;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

UPDATE Deliveries
SET PamType6 = 'Briscoes'
WHERE DeliverersID = 15;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

DELETE FROM Deliveries
WHERE DeliverersID = 14 OR DeliverersID = 15;

SELECT * FROM Deliveries;
```

Messages

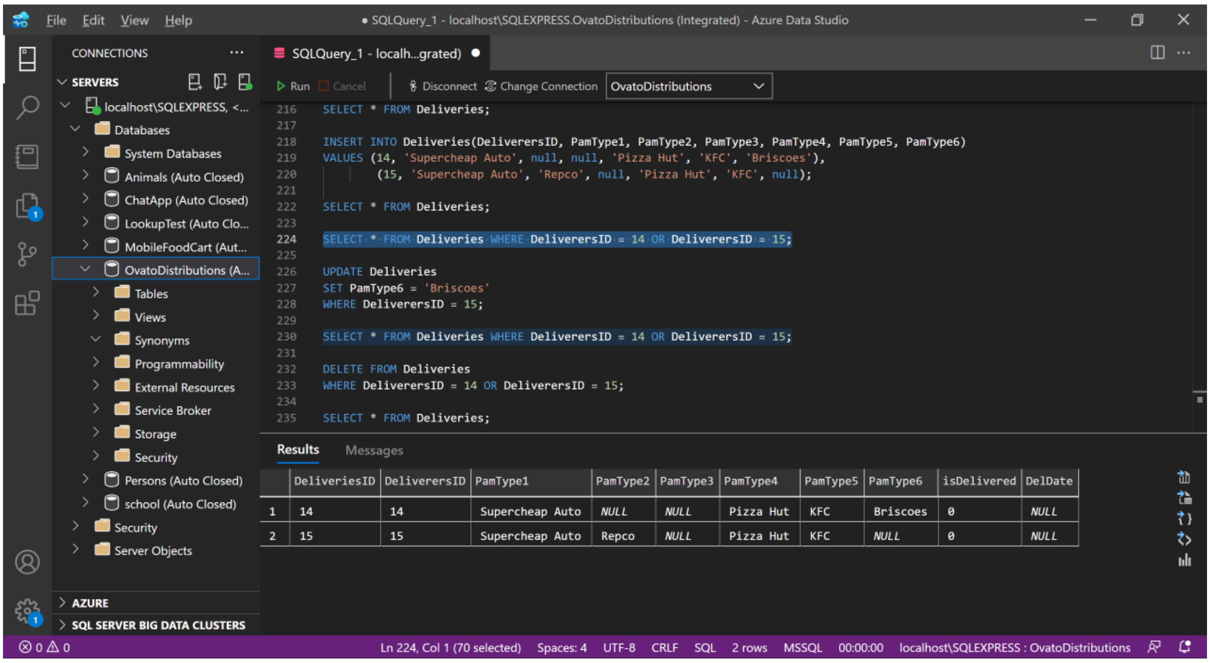
```
9:26:29 PM    Started executing query at Line 218
              (2 rows affected)
              Total execution time: 00:00:00.005
```



```sql
SELECT * FROM Deliveries;

INSERT INTO Deliveries(DeliverersID, PamType1, PamType2, PamType3, PamType4, PamType5, PamType6)
VALUES (14, 'Supercheap Auto', null, null, 'Pizza Hut', 'KFC', 'Briscoes'),
        (15, 'Supercheap Auto', 'Repco', null, 'Pizza Hut', 'KFC', null);

SELECT * FROM Deliveries;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

UPDATE Deliveries
SET PamType6 = 'Briscoes'
WHERE DeliverersID = 15;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

DELETE FROM Deliveries
WHERE DeliverersID = 14 OR DeliverersID = 15;

SELECT * FROM Deliveries;
```

	DeliveriesID	DeliverersID	PamType1	PamType2	PamType3	PamType4	PamType5	PamType6	isDelivered	DelDate
10	10	10	Supercheap Auto	Repco	NULL	NULL	KFC	NULL	0	NULL
11	11	11	Supercheap Auto	Repco	NULL	Pizza Hut	NULL	NULL	0	NULL
12	12	12	Supercheap Auto	Repco	Farmers	Pizza Hut	KFC	Briscoes	0	NULL
13	13	13	Supercheap Auto	NULL	NULL	Pizza Hut	KFC	NULL	0	NULL
14	14	14	Supercheap Auto	NULL	NULL	Pizza Hut	KFC	Briscoes	0	NULL
15	15	15	Supercheap Auto	Repco	NULL	Pizza Hut	KFC	NULL	0	NULL

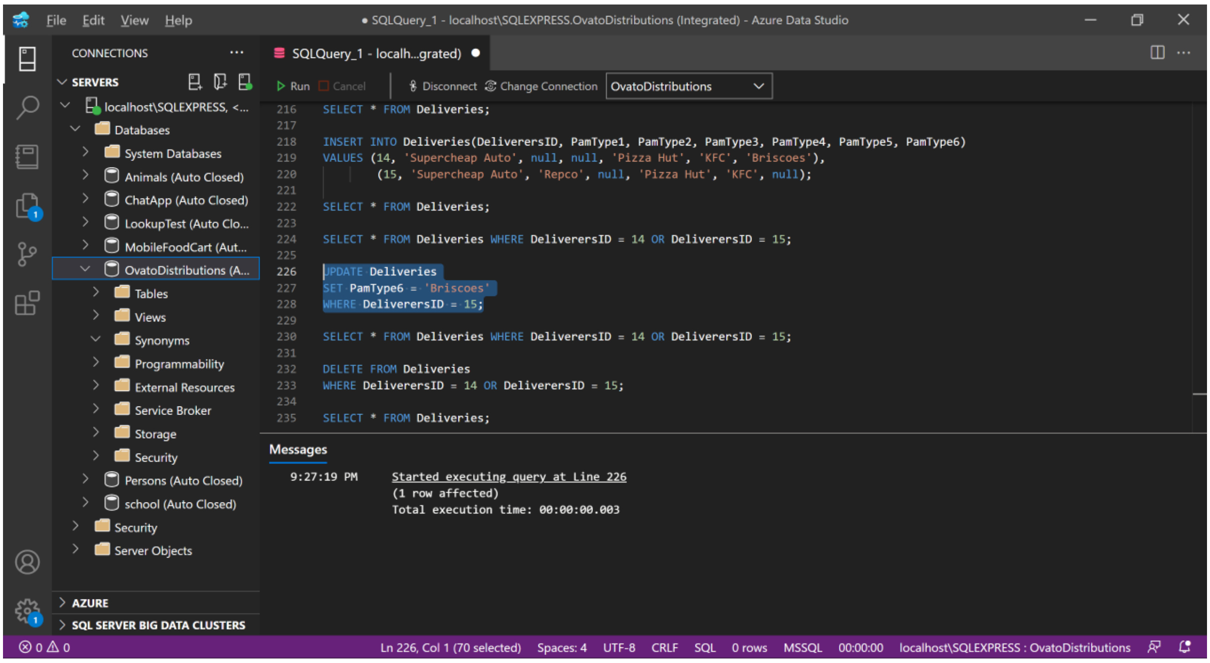
Ln 222, Col 1 (25 selected) Spaces: 4 UTF-8 CRLF SQL 15 rows MSSQL 00:00:00 localhost\SQLEXPRESS : OvatoDistributions



```sql
SELECT * FROM Deliveries;

INSERT INTO Deliveries(DeliverersID, PamType1, PamType2, PamType3, PamType4, PamType5, PamType6)
VALUES (14, 'Supercheap Auto', null, null, 'Pizza Hut', 'KFC', 'Briscoes'),
        (15, 'Supercheap Auto', 'Repco', null, 'Pizza Hut', 'KFC', null);

SELECT * FROM Deliveries;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

UPDATE Deliveries
SET PamType6 = 'Briscoes'
WHERE DeliverersID = 15;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

DELETE FROM Deliveries
WHERE DeliverersID = 14 OR DeliverersID = 15;

SELECT * FROM Deliveries;
```

	DeliveriesID	DeliverersID	PamType1	PamType2	PamType3	PamType4	PamType5	PamType6	isDelivered	DelDate
1	14	14	Supercheap Auto	NULL	NULL	Pizza Hut	KFC	Briscoes	0	NULL
2	15	15	Supercheap Auto	Repco	NULL	Pizza Hut	KFC	NULL	0	NULL

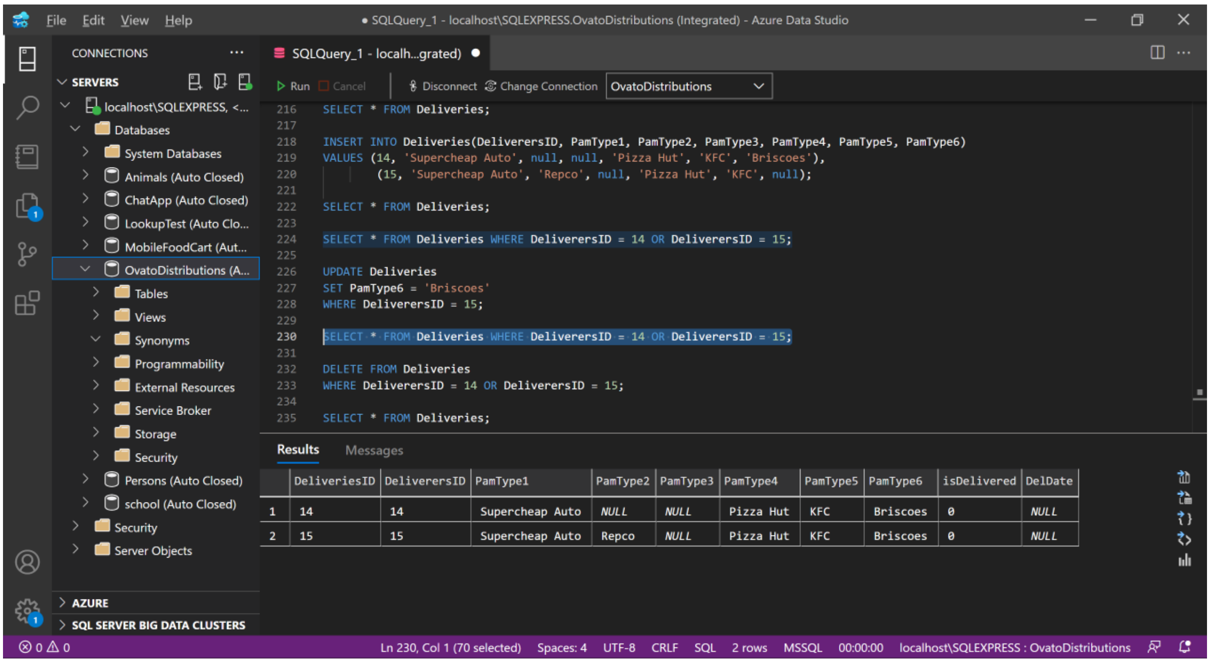
Ln 224, Col 1 (70 selected) Spaces: 4 UTF-8 CRLF SQL 2 rows MSSQL 00:00:00 localhost\SQLEXPRESS : OvatoDistributions



```
SELECT * FROM Deliveries;

INSERT INTO Deliveries(DeliverersID, PamType1, PamType2, PamType3, PamType4, PamType5, PamType6)
VALUES (14, 'Supercheap Auto', null, null, 'Pizza Hut', 'KFC', 'Briscoes'),
        (15, 'Supercheap Auto', 'Repco', null, 'Pizza Hut', 'KFC', null);

SELECT * FROM Deliveries;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

UPDATE Deliveries
SET PamType6 = 'Briscoes'
WHERE DeliverersID = 15;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

DELETE FROM Deliveries
WHERE DeliverersID = 14 OR DeliverersID = 15;

SELECT * FROM Deliveries;
```

Messages

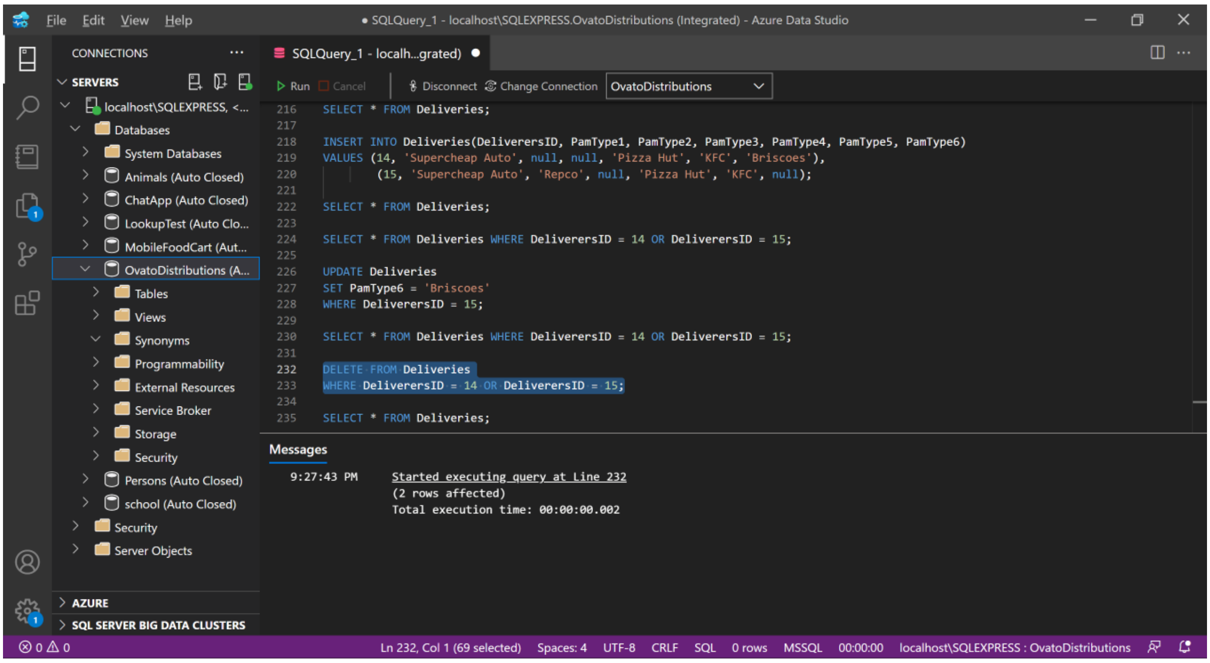
```
9:27:19 PM     Started executing query at Line 226
               (1 row affected)
               Total execution time: 00:00:00.003
```



```
SELECT * FROM Deliveries;

INSERT INTO Deliveries(DeliverersID, PamType1, PamType2, PamType3, PamType4, PamType5, PamType6)
VALUES (14, 'Supercheap Auto', null, null, 'Pizza Hut', 'KFC', 'Briscoes'),
        (15, 'Supercheap Auto', 'Repco', null, 'Pizza Hut', 'KFC', null);

SELECT * FROM Deliveries;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

UPDATE Deliveries
SET PamType6 = 'Briscoes'
WHERE DeliverersID = 15;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

DELETE FROM Deliveries
WHERE DeliverersID = 14 OR DeliverersID = 15;

SELECT * FROM Deliveries;
```

Results Messages

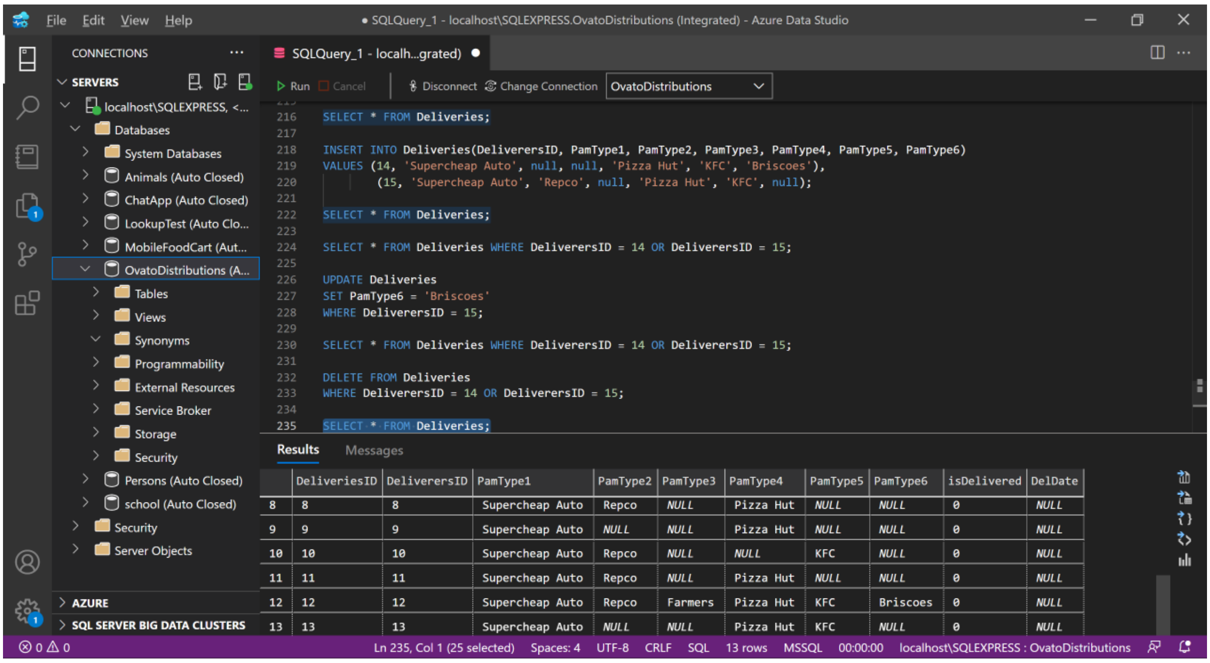
	DeliveriesID	DeliverersID	PamType1	PamType2	PamType3	PamType4	PamType5	PamType6	isDelivered	DelDate
1	14	14	Supercheap Auto	*NULL*	*NULL*	Pizza Hut	KFC	Briscoes	0	*NULL*
2	15	15	Supercheap Auto	Repco	*NULL*	Pizza Hut	KFC	Briscoes	0	*NULL*



```sql
SELECT * FROM Deliveries;

INSERT INTO Deliveries(DeliverersID, PamType1, PamType2, PamType3, PamType4, PamType5, PamType6)
VALUES (14, 'Supercheap Auto', null, null, 'Pizza Hut', 'KFC', 'Briscoes'),
       (15, 'Supercheap Auto', 'Repco', null, 'Pizza Hut', 'KFC', null);

SELECT * FROM Deliveries;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

UPDATE Deliveries
SET PamType6 = 'Briscoes'
WHERE DeliverersID = 15;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

DELETE FROM Deliveries
WHERE DeliverersID = 14 OR DeliverersID = 15;

SELECT * FROM Deliveries;
```

Messages

```
9:27:43 PM    Started executing query at Line 232
              (2 rows affected)
              Total execution time: 00:00:00.002
```



```sql
SELECT * FROM Deliveries;

INSERT INTO Deliveries(DeliverersID, PamType1, PamType2, PamType3, PamType4, PamType5, PamType6)
VALUES (14, 'Supercheap Auto', null, null, 'Pizza Hut', 'KFC', 'Briscoes'),
       (15, 'Supercheap Auto', 'Repco', null, 'Pizza Hut', 'KFC', null);

SELECT * FROM Deliveries;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

UPDATE Deliveries
SET PamType6 = 'Briscoes'
WHERE DeliverersID = 15;

SELECT * FROM Deliveries WHERE DeliverersID = 14 OR DeliverersID = 15;

DELETE FROM Deliveries
WHERE DeliverersID = 14 OR DeliverersID = 15;

SELECT * FROM Deliveries;
```

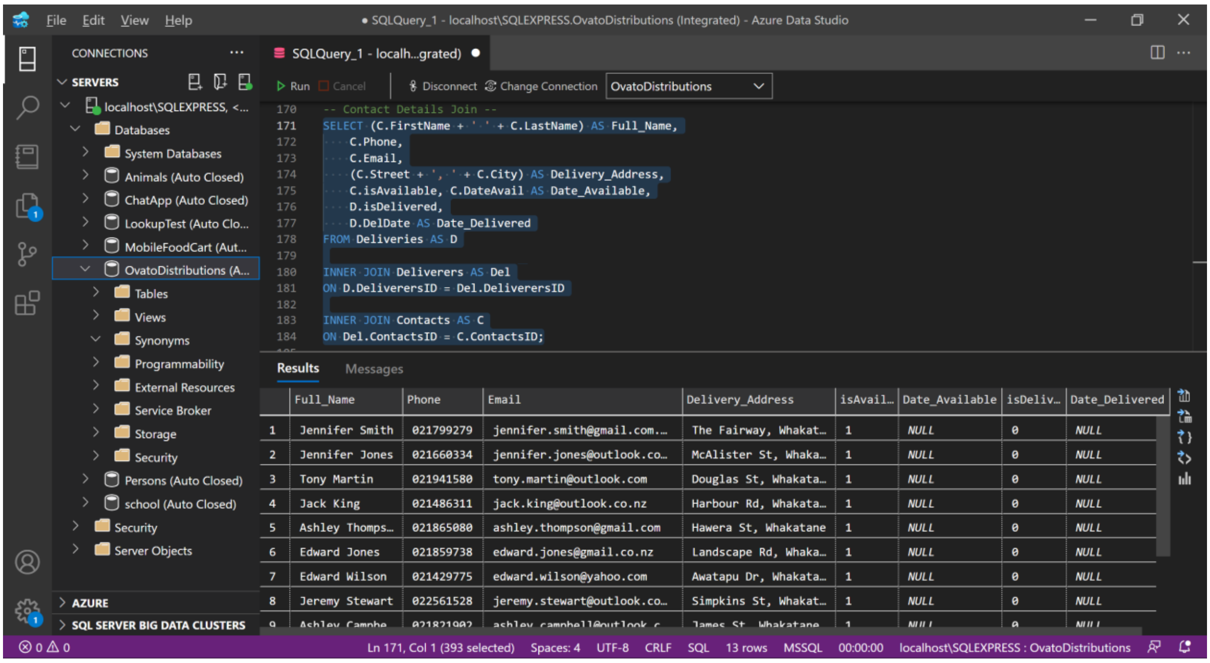
Results

	DeliveriesID	DeliverersID	PamType1	PamType2	PamType3	PamType4	PamType5	PamType6	isDelivered	DelDate
8	8	8	Supercheap Auto	Repco	NULL	Pizza Hut	NULL	NULL	0	NULL
9	9	9	Supercheap Auto	NULL	NULL	Pizza Hut	NULL	NULL	0	NULL
10	10	10	Supercheap Auto	Repco	NULL	NULL	KFC	NULL	0	NULL
11	11	11	Supercheap Auto	Repco	NULL	Pizza Hut	NULL	NULL	0	NULL
12	12	12	Supercheap Auto	Repco	Farmers	Pizza Hut	KFC	Briscoes	0	NULL
13	13	13	Supercheap Auto	NULL	NULL	Pizza Hut	KFC	NULL	0	NULL

File Edit View Help
• SQLQuery_1 - localhost\SQLEXPRESS.OvatoDistributions (Integrated) - Azure Data Studio
CONNECTIONS
SERVERS
SQLQuery_1 - localh...grated)
Run
Cancel
Disconnect
Change Connection
OvatoDistributions
Results
Messages
Ln 235, Col 1 (25 selected) Spaces: 4 UTF-8 CRLF SQL 13 rows MSSQL 00:00:00 localhost\SQLEXPRESS : OvatoDistributions

Join Queries

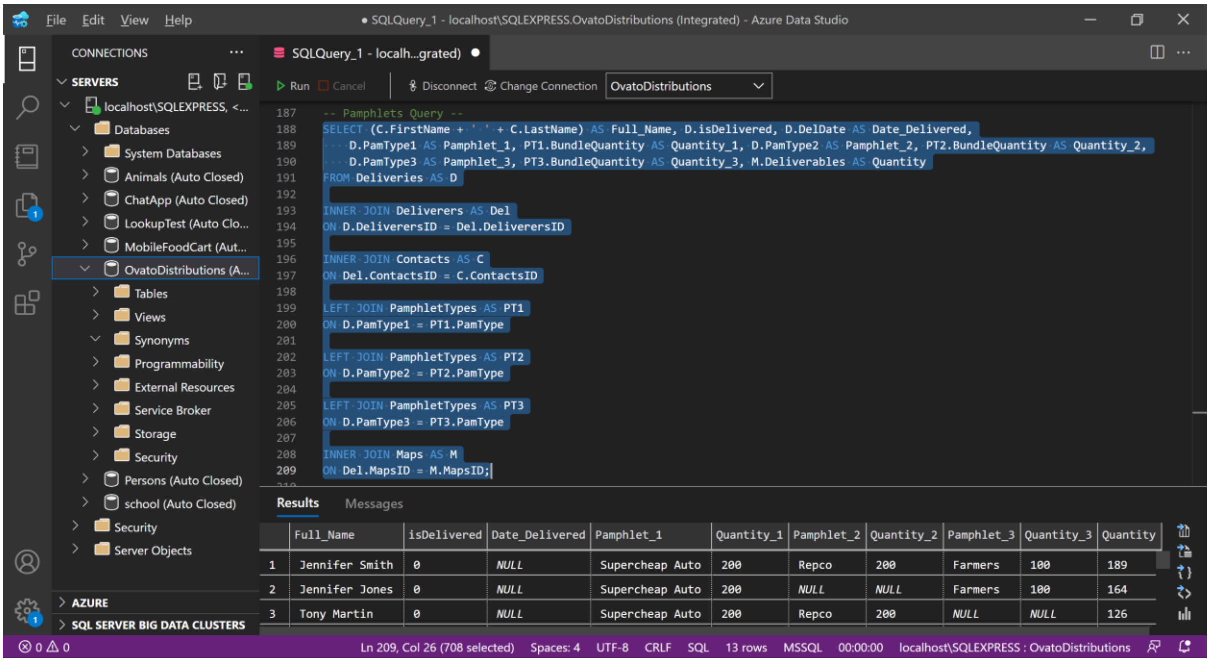
Contacts Join Query



```sql
-- Contact Details Join --
SELECT (C.FirstName + ' ' + C.LastName) AS Full_Name,
    C.Phone,
    C.Email,
    (C.Street + ', ' + C.City) AS Delivery_Address,
    C.isAvailable, C.DateAvail AS Date_Available,
    D.isDelivered,
    D.DelDate AS Date_Delivered
FROM Deliveries AS D

INNER JOIN Deliverers AS Del
ON D.DeliverersID = Del.DeliverersID

INNER JOIN Contacts AS C
ON Del.ContactsID = C.ContactsID;
```

	Full_Name	Phone	Email	Delivery_Address	isAvail...	Date_Available	isDeliv...	Date_Delivered
1	Jennifer Smith	021799279	jennifer.smith@gmail.com....	The Fairway, Whakat...	1	NULL	0	NULL
2	Jennifer Jones	021660334	jennifer.jones@outlook.co...	McAlister St, Whaka...	1	NULL	0	NULL
3	Tony Martin	021941580	tony.martin@outlook.com	Douglas St, Whakata...	1	NULL	0	NULL
4	Jack King	021486311	jack.king@outlook.co.nz	Harbour Rd, Whakata...	1	NULL	0	NULL
5	Ashley Thomps...	021865080	ashley.thompson@gmail.com	Hawera St, Whakatane	1	NULL	0	NULL
6	Edward Jones	021859738	edward.jones@gmail.co.nz	Landscape Rd, Whaka...	1	NULL	0	NULL
7	Edward Wilson	021429775	edward.wilson@yahoo.com	Awatapu Dr, Whakata...	1	NULL	0	NULL
8	Jeremy Stewart	022561528	jeremy.stewart@outlook.co...	Simpkins St, Whakat...	1	NULL	0	NULL
9	Ashley Campbe	021821902	ashley.campbell@outlook.c	James St, Whakatane	1	NULL	0	NULL

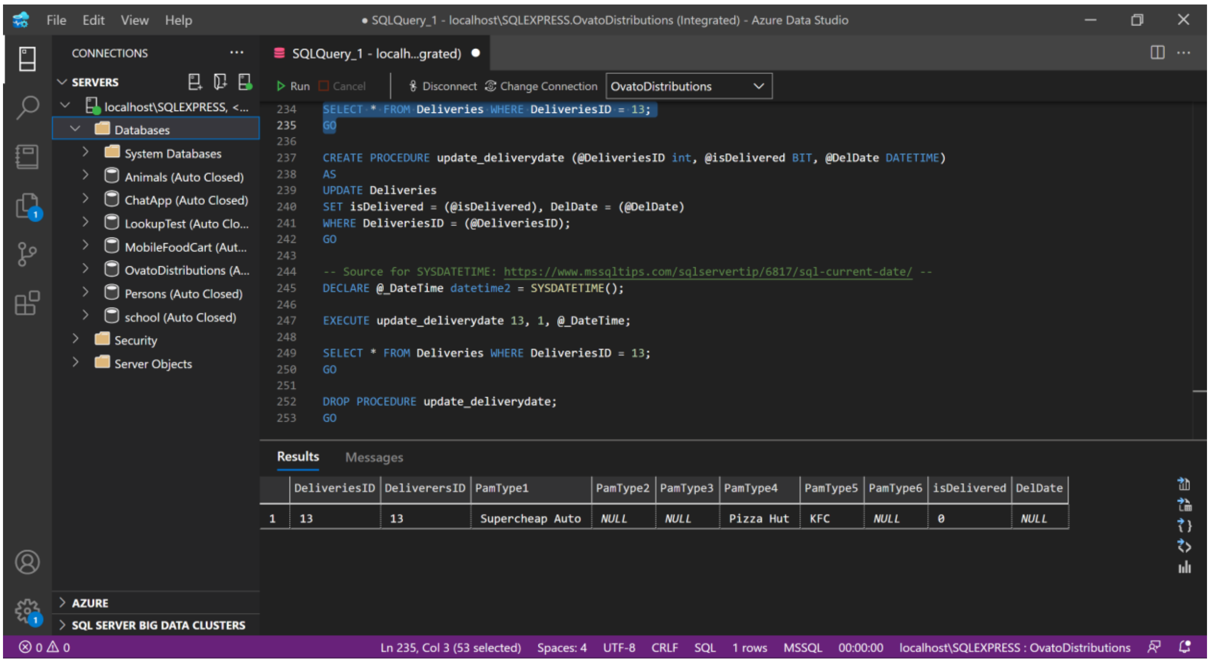
Pamphlet Join Query



```sql
-- Pamphlets Query --
SELECT (C.FirstName + ' ' + C.LastName) AS Full_Name, D.isDelivered, D.DelDate AS Date_Delivered,
    D.PamType1 AS Pamphlet_1, PT1.BundleQuantity AS Quantity_1, D.PamType2 AS Pamphlet_2, PT2.BundleQuantity AS Quantity_2,
    D.PamType3 AS Pamphlet_3, PT3.BundleQuantity AS Quantity_3, M.Deliverables AS Quantity
FROM Deliveries AS D

INNER JOIN Deliverers AS Del
ON D.DeliverersID = Del.DeliverersID

INNER JOIN Contacts AS C
ON Del.ContactsID = C.ContactsID

LEFT JOIN PamphletTypes AS PT1
ON D.PamType1 = PT1.PamType

LEFT JOIN PamphletTypes AS PT2
ON D.PamType2 = PT2.PamType

LEFT JOIN PamphletTypes AS PT3
ON D.PamType3 = PT3.PamType

INNER JOIN Maps AS M
ON Del.MapsID = M.MapsID;
```

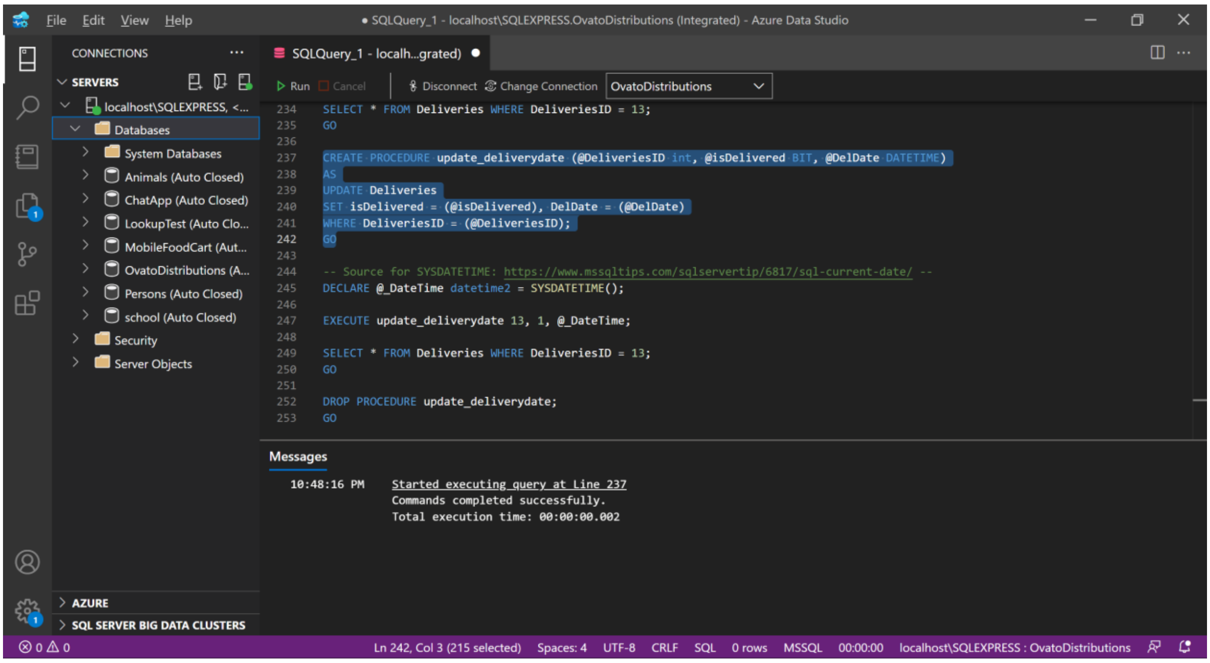
	Full_Name	isDelivered	Date_Delivered	Pamphlet_1	Quantity_1	Pamphlet_2	Quantity_2	Pamphlet_3	Quantity_3	Quantity
1	Jennifer Smith	0	NULL	Supercheap Auto	200	Repco	200	Farmers	100	189
2	Jennifer Jones	0	NULL	Supercheap Auto	200	NULL	NULL	Farmers	100	164
3	Tony Martin	0	NULL	Supercheap Auto	200	Repco	200	NULL	NULL	126

Stored Procedure



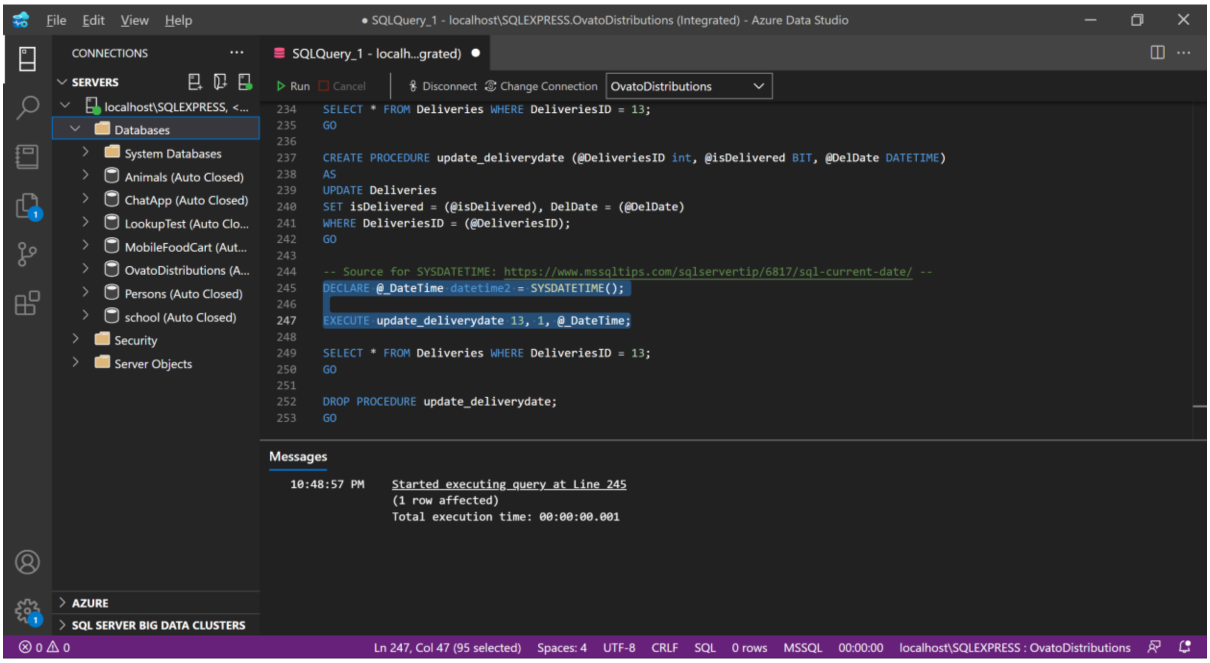
```sql
SELECT * FROM Deliveries WHERE DeliveriesID = 13;
GO

CREATE PROCEDURE update_deliverydate (@DeliveriesID int, @isDelivered BIT, @DelDate DATETIME)
AS
UPDATE Deliveries
SET isDelivered = (@isDelivered), DelDate = (@DelDate)
WHERE DeliveriesID = (@DeliveriesID);
GO

-- Source for SYSDATETIME: https://www.mssqltips.com/sqlservertip/6817/sql-current-date/ --
DECLARE @_DateTime datetime2 = SYSDATETIME();

EXECUTE update_deliverydate 13, 1, @_DateTime;

SELECT * FROM Deliveries WHERE DeliveriesID = 13;
GO

DROP PROCEDURE update_deliverydate;
GO
```

	DeliveriesID	DeliverersID	PamType1	PamType2	PamType3	PamType4	PamType5	PamType6	isDelivered	DelDate
1	13	13	Supercheap Auto	NULL	NULL	Pizza Hut	KFC	NULL	0	NULL



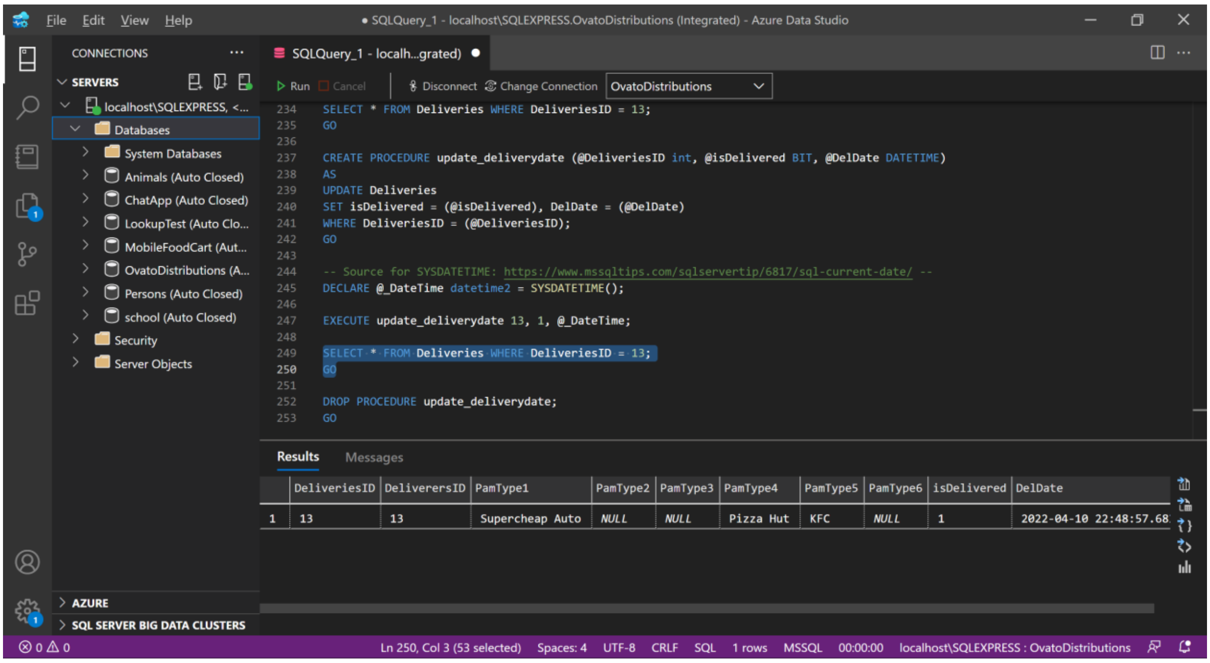
```sql
SELECT * FROM Deliveries WHERE DeliveriesID = 13;
GO

CREATE PROCEDURE update_deliverydate (@DeliveriesID int, @isDelivered BIT, @DelDate DATETIME)
AS
UPDATE Deliveries
SET isDelivered = (@isDelivered), DelDate = (@DelDate)
WHERE DeliveriesID = (@DeliveriesID);
GO

-- Source for SYSDATETIME: https://www.mssqltips.com/sqlservertip/6817/sql-current-date/ --
DECLARE @_DateTime datetime2 = SYSDATETIME();

EXECUTE update_deliverydate 13, 1, @_DateTime;

SELECT * FROM Deliveries WHERE DeliveriesID = 13;
GO

DROP PROCEDURE update_deliverydate;
GO
```

```
10:48:16 PM    Started executing query at Line 237
               Commands completed successfully.
               Total execution time: 00:00:00.002
```



```sql
SELECT * FROM Deliveries WHERE DeliveriesID = 13;
GO

CREATE PROCEDURE update_deliverydate (@DeliveriesID int, @isDelivered BIT, @DelDate DATETIME)
AS
UPDATE Deliveries
SET isDelivered = (@isDelivered), DelDate = (@DelDate)
WHERE DeliveriesID = (@DeliveriesID);
GO

-- Source for SYSDATETIME: https://www.mssqltips.com/sqlservertip/6817/sql-current-date/ --
DECLARE @_DateTime datetime2 = SYSDATETIME();

EXECUTE update_deliverydate 13, 1, @_DateTime;

SELECT * FROM Deliveries WHERE DeliveriesID = 13;
GO

DROP PROCEDURE update_deliverydate;
GO
```

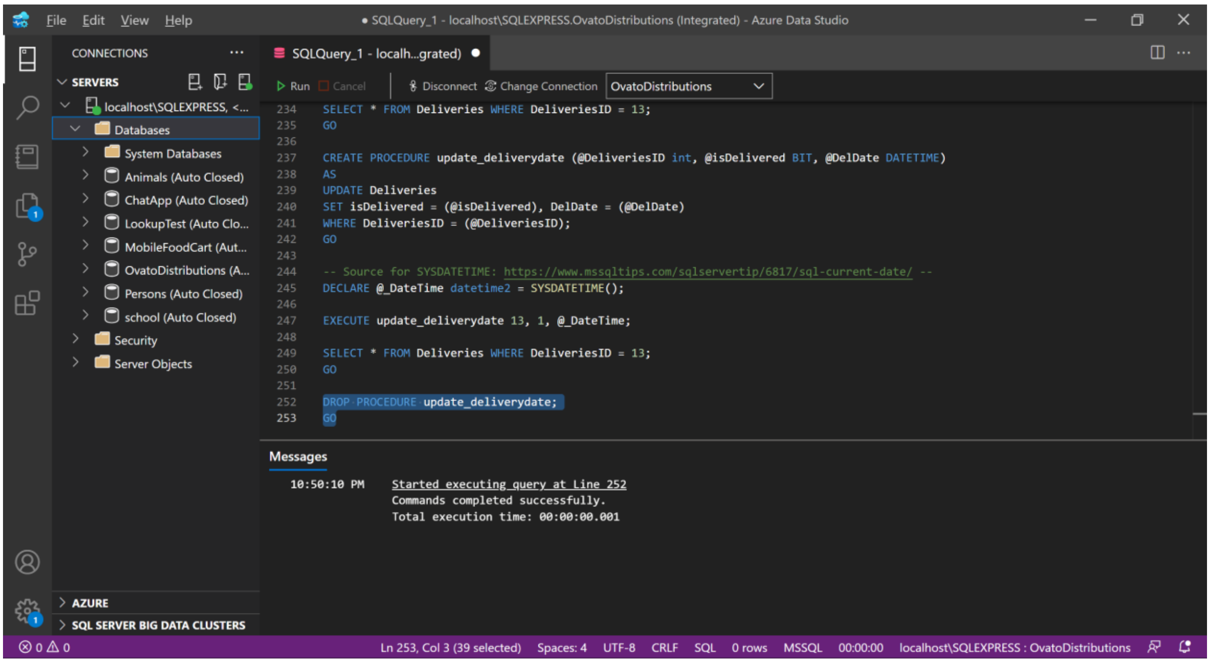
```
10:48:57 PM    Started executing query at Line 245
               (1 row affected)
               Total execution time: 00:00:00.001
```



```sql
SELECT * FROM Deliveries WHERE DeliveriesID = 13;
GO

CREATE PROCEDURE update_deliverydate (@DeliveriesID int, @isDelivered BIT, @DelDate DATETIME)
AS
UPDATE Deliveries
SET isDelivered = (@isDelivered), DelDate = (@DelDate)
WHERE DeliveriesID = (@DeliveriesID);
GO

-- Source for SYSDATETIME: https://www.mssqltips.com/sqlservertip/6817/sql-current-date/ --
DECLARE @_DateTime datetime2 = SYSDATETIME();

EXECUTE update_deliverydate 13, 1, @_DateTime;

SELECT * FROM Deliveries WHERE DeliveriesID = 13;
GO

DROP PROCEDURE update_deliverydate;
GO
```

	DeliveriesID	DeliverersID	PamType1	PamType2	PamType3	PamType4	PamType5	PamType6	isDelivered	DelDate
1	13	13	Supercheap Auto	NULL	NULL	Pizza Hut	KFC	NULL	1	2022-04-10 22:48:57.68

Codie Shannon
30013375



```sql
SELECT * FROM Deliveries WHERE DeliveriesID = 13;
GO

CREATE PROCEDURE update_deliverydate (@DeliveriesID int, @isDelivered BIT, @DelDate DATETIME)
AS
UPDATE Deliveries
SET isDelivered = (@isDelivered), DelDate = (@DelDate)
WHERE DeliveriesID = (@DeliveriesID);
GO

-- Source for SYSDATETIME: https://www.mssqltips.com/sqlservertip/6817/sql-current-date/ --
DECLARE @_DateTime datetime2 = SYSDATETIME();

EXECUTE update_deliverydate 13, 1, @_DateTime;

SELECT * FROM Deliveries WHERE DeliveriesID = 13;
GO

DROP PROCEDURE update_deliverydate;
GO
```

Messages

```
10:50:10 PM    Started executing query at Line 252
               Commands completed successfully.
               Total execution time: 00:00:00.001
```

Testing

Maps Table

Test	Execution	Expected Result	Result (Pass/Fail)	Comment

Add record to table	INSERT INTO Maps (MapImg, Deliverables)
	VALUES ('https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template_2.jpg?raw=true', 195);
		Record is added to table	Pass	

Read one record from table based on MapsID	SELECT * FROM Maps WHERE MapsID = 9;	The record with the MapsID of 9 is displayed	Pass	
Read records from table where the deliverable values are above 175	SELECT * FROM Maps WHERE Deliverables > 175;	All records with a deliverables value above 175 are displayed	Pass	
Read all records from table	SELECT * FROM Maps;	All records within the table are displayed	Pass	

Update record in table	UPDATE Maps
	SET Deliverables = 182
	WHERE MapsID = 9;
		Record with the MapsID of 9 is updated to have the deliverables value of 182	Pass	

Delete record in table	DELETE FROM Maps
	WHERE MapsID = 9;
		Record with the MapsID of 9 is removed from the table	Pass	Function does work however the correlated records in deliveries and deliverers must be modified to remove the reference to the MapsID or the records must be deleted before hand

Procedurally add record to table	CREATE PROCEDURE insert_map (@MapImg NVARCHAR(255), @Deliverables int)
	AS
	INSERT INTO Maps
	VALUES (@MapImg, @Deliverables);
	GO
	EXECUTE insert_map 'https://github.com/CodieShannon/OvatoDistributions/blob/main/route_template_2.jpg?raw=true', 168;
		Record is added to table	Pass	

	SELECT * FROM Maps;
	GO
	DROP PROCEDURE insert_map;
	GO			

Contacts Table

Test	Execution	Expected Result	Result (Pass/Fail)	Comment
Add record to table	INSERT INTO Contacts (FirstName, LastName, Phone, Email, Street, City) VALUES ('Billy', 'Bob', '0220218954', 'billy.bob100@outlook.com', '297 Bridge St', 'Whakatane');	Record is added to table	Pass	
Read one record from table based on FirstName and LastName	SELECT * FROM Contacts WHERE FirstName = 'Billy' AND LastName = 'Bob';	The record with the FirstName of Billy and LastName of Bob is displayed	Pass	
Read records from table where the FirstName values are Edward	SELECT * FROM Contacts WHERE FirstName = 'Edward';	All records with the FirstName Edward are displayed	Pass	
Read all records from table	SELECT * FROM Contacts;	All records in the table are displayed	Pass	
Update record in table	UPDATE Contacts SET Street = '101 Arawa Rd' WHERE FirstName = 'Billy' AND LastName = 'Bob';	Record with the FirstName of Billy and the LastName of Bob is updated to have the street	Pass	

		value of 101 Arawa Rd		

Delete record from table	DELETE FROM Contacts
	WHERE FirstName = 'Billy' AND LastName = 'Bob';
		Record with the FirstName of Billy and the LastName of Bob is deleted from the table	Pass	Function does work however if the record is referenced in the Deliveries or Deliverers table, the correlated records in the other tables must be modified or deleted to remove the reference to the ContactsID for the specified record in Contacts

Procedurally add record to table	CREATE PROCEDURE insert_contact (@FirstName NVARCHAR(100), @LastName NVARCHAR(100), @Phone VARCHAR(50), @Email NVARCHAR(255), @Street VARCHAR(100), @City VARCHAR(100), @isAvailable BIT, @DateAvail DATETIME)
	AS
	INSERT INTO Contacts
	VALUES (@FirstName, @LastName, @Phone, @Email, @Street, @City, @isAvailable, @DateAvail);
		Record is added to the table	Pass	

	GO
	EXECUTE insert_contact 'Billy', 'Bob', '0220218954', 'billy.bob100@outlook.com', '297 Bridge St', 'Whakatane', 1, null;
	SELECT * FROM Contacts;
	GO
	DROP PROCEDURE insert_contact;
	GO			

Deliverers Table

Test	Execution	Expected Result	Result (Pass/Fail)	Comment
Add record to table	INSERT INTO Deliverers (MapsID, ContactsID) VALUES (4, 5);	Adds record to table if the Maps table contains a primary key of 4, alongside the Contacts table containing a primary key of 5	Pass	
Read one record from table based on MapsID and ContactsID	SELECT * FROM Deliverers WHERE MapsID = 4 AND ContactsID = 5;	The record with the MapsID of 4 and the ContactsID of 5 is displayed	Pass	
Read records from table where the Deliverers values are above 5	SELECT * FROM Deliverers WHERE MapsID > 5;	All records with a MapsID above 5 are displayed	Pass	
Read all records from table	SELECT * FROM Deliverers;	All records within the Deliverers table are displayed	Pass	
Update record in table	UPDATE Deliverers	The record with the MapsID of 4	Pass	

	SET MapsID = 7 WHERE MapsID = 4 AND ContactsID = 5;	and the ContactsID of 5 is updated to have a MapsID of 7		
Delete record from table	DELETE FROM Deliverers WHERE MapsID = 7 AND ContactsID = 5;	Record is deleted from the table where the MapsID is 7 and the ContactsID is 5	Pass	

Procedurally add record to table	CREATE PROCEDURE insert_deliverer (@MapsID int, @ContactsID int)
	AS
	INSERT INTO Deliverers
	VALUES (@MapsID, @ContactsID);
	GO
	
	EXECUTE insert_deliverer 4, 5;
	
	SELECT * FROM Deliverers;
	GO
	
	DROP PROCEDURE insert_deliverer;
	GO
		Adds record to table if the Maps table contains a primary key of 4, alongside the Contacts table containing a primary key of 5	Pass	

PamphletTypes Table

Test	Execution	Expected Result	Result (Pass/Fail)	Comment
Add record to table	INSERT INTO PamphletTypes (PamType, BundleQuantity) VALUES ('Masport', 400);	Adds record to table with the PamType of Masport and the BundleQuantity of 400	Pass	
Read all records in table	SELECT * FROM PamphletTypes;	All the records in the table are displayed	Pass	

Update record in table	UPDATE PamphletTypes
	SET BundleQuantity = 800
	WHERE PamType = 'Masport';
		Record with the PamType of Masport is updated to have a BundleQuantity of 800	Pass	

Delete record in table	DELETE FROM PamphletTypes
	WHERE PamType = 'Masport';
		The record with the PamType of Masport is deleted from the table	Pass	

Procedurally add record to table	CREATE PROCEDURE insert_pamphlet (@PamType VARCHAR(150), @BundleQuantity int)
	AS
	INSERT INTO PamphletTypes
	VALUES (@PamType, @BundleQuantity);
	GO
	
	EXECUTE insert_pamphlet 'Masport', 400;
	
	SELECT * FROM PamphletTypes;
	GO
	
	DROP PROCEDURE insert_pamphlet;
	GO
		Adds record to table with the PamType of Masport and the BundleQuantity of 400	Pass	

Deliveries Table

Test	Execution	Expected Result	Result (Pass/Fail)	Comment
Add record to table	INSERT INTO Deliveries(DeliverersID, PamType1, PamType2,	Adds record to table under two conditions. if the values entered	Pass	

	PamType3, PamType4, PamType5, PamType6) VALUES (14, 'Supercheap Auto', 'Repco', null, null, null, null);	into the PamType fields are null or contained within the PamphletTypes table, and If the Deliverers table contains a primary key of 14		
Read one record from table based on DeliverersID	SELECT * FROM Deliveries WHERE DeliverersID = 14;	The record with the DeliverersID of 14 is displayed	Pass	
Read records from table where the PamType2 values are Repco	SELECT * FROM Deliveries WHERE PamType2 = 'Repco';	All records with the PamType2 value of Repco are displayed	Pass	
Read all records from table	SELECT * FROM Deliveries;	Displays all records within table	Pass	
Update record in table	UPDATE Deliveries SET PamType3 = 'Farmers' WHERE DeliverersID = 14;	Updates record with the DeliverersID of 14 to have a PamType3 value of Farmers	Pass	
Delete record from table	DELETE FROM Deliveries WHERE DeliverersID = 14;	The record with the DeliverersID of 14 is deleted from the table	Pass	
Procedurally add record to table	CREATE PROCEDURE insert_delivery (@DeliverersID int, @PamType1 VARCHAR(150), @PamType2 VARCHAR(150), @PamType3 VARCHAR(150), @PamType4 VARCHAR(150), @PamType5 VARCHAR(150), @PamType6 VARCHAR(150), @isDelivered BIT, @DelDate DateTime)	Adds record to table under two conditions. if the values entered into the PamType fields are null or contained within the PamphletTypes table, and If the Deliverers table contains a primary key of 14	Pass	

	AS INSERT INTO Deliveries VALUES (@DeliverersID, @PamType1, @PamType2, @PamType3, @PamType4, @PamType5, @PamType6, @isDelivered, @DelDate); GO EXECUTE insert_delivery 14, 'Supercheap Auto', 'Repco', null, null, null, null, 0, null; SELECT * FROM Deliveries; GO DROP PROCEDURE insert_delivery; GO			

Additional Documentation

Microsoft Planner for Assignment

Link: https://tasks.office.com/edunetnz.onmicrosoft.com/Home/PlanViews/ewUSfDM0YEChCJKB5CykqcgAFR0l?Type=PlanLink&Channel=Link&CreatedTime=637848335435790000

Video Build Evidence

Link: Video Evidence.mp4

SQL Server File

Link: SQL Files.zip