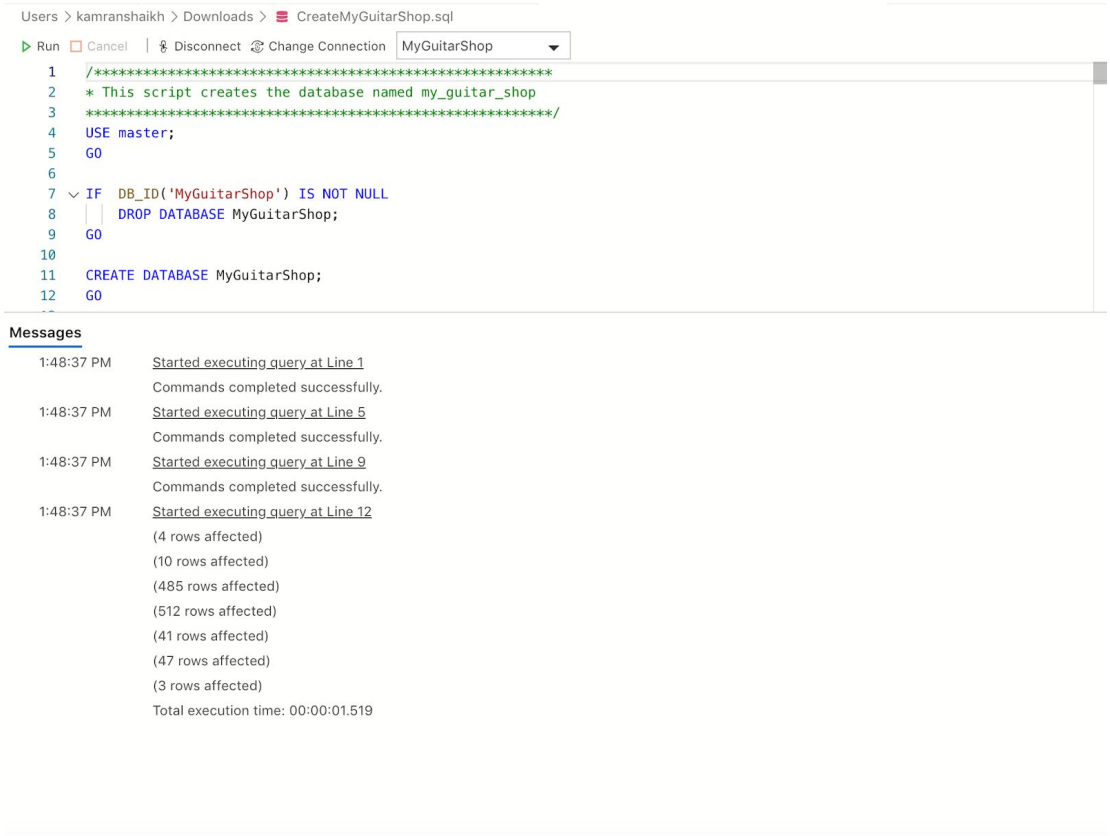
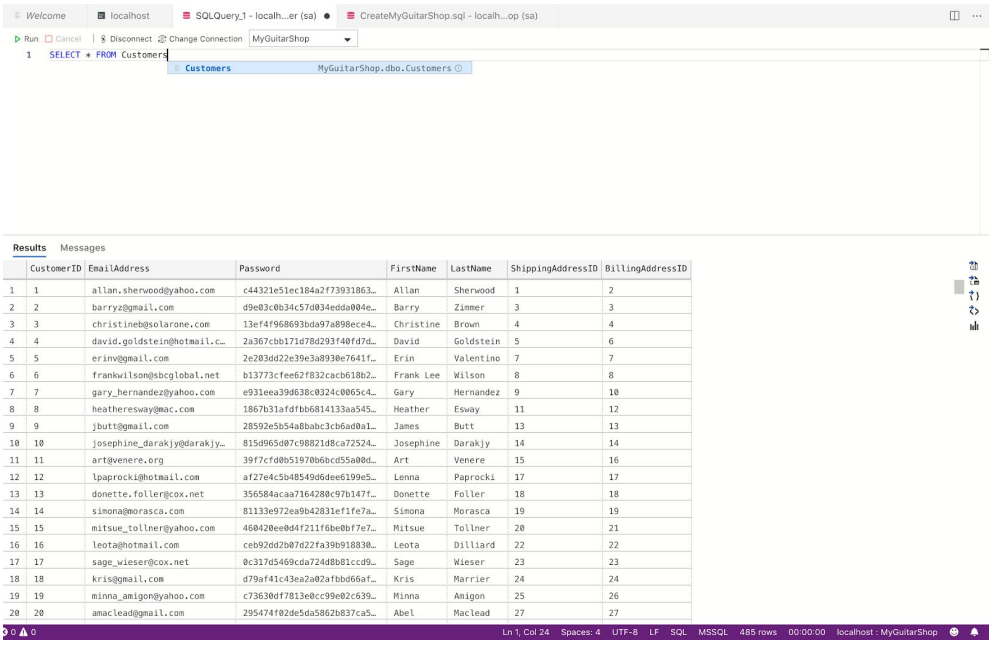
Kamran Shaikh
Project 1

A) Database Set Up

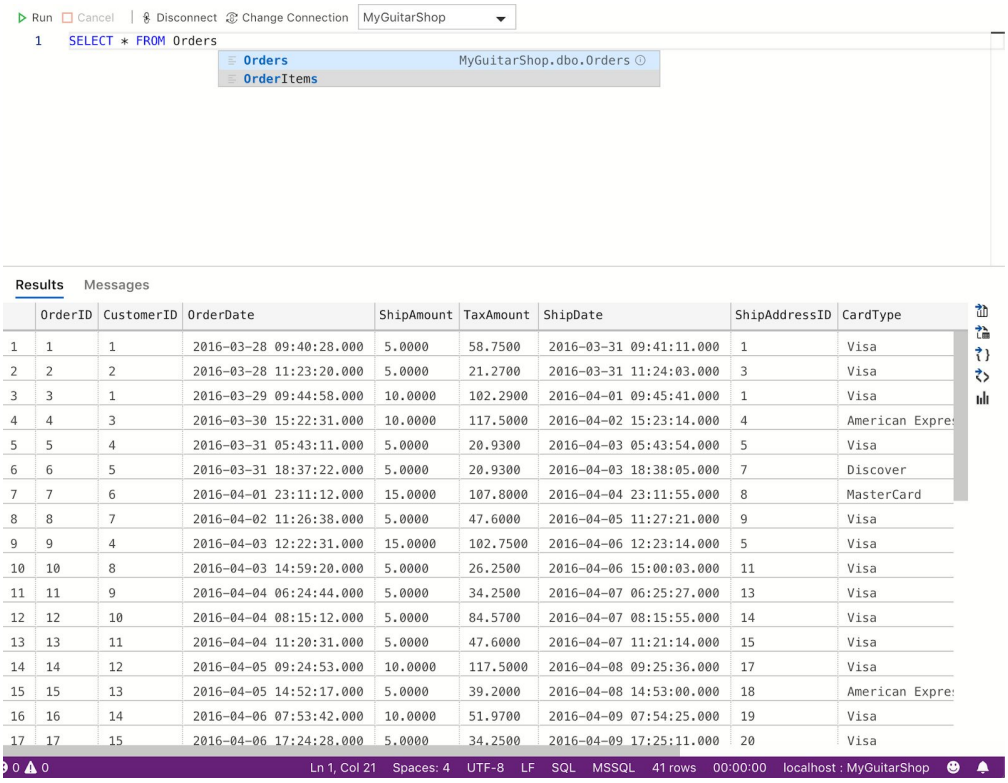
1)



2) Customers Table



Orders Table



B) An Introduction to SQL

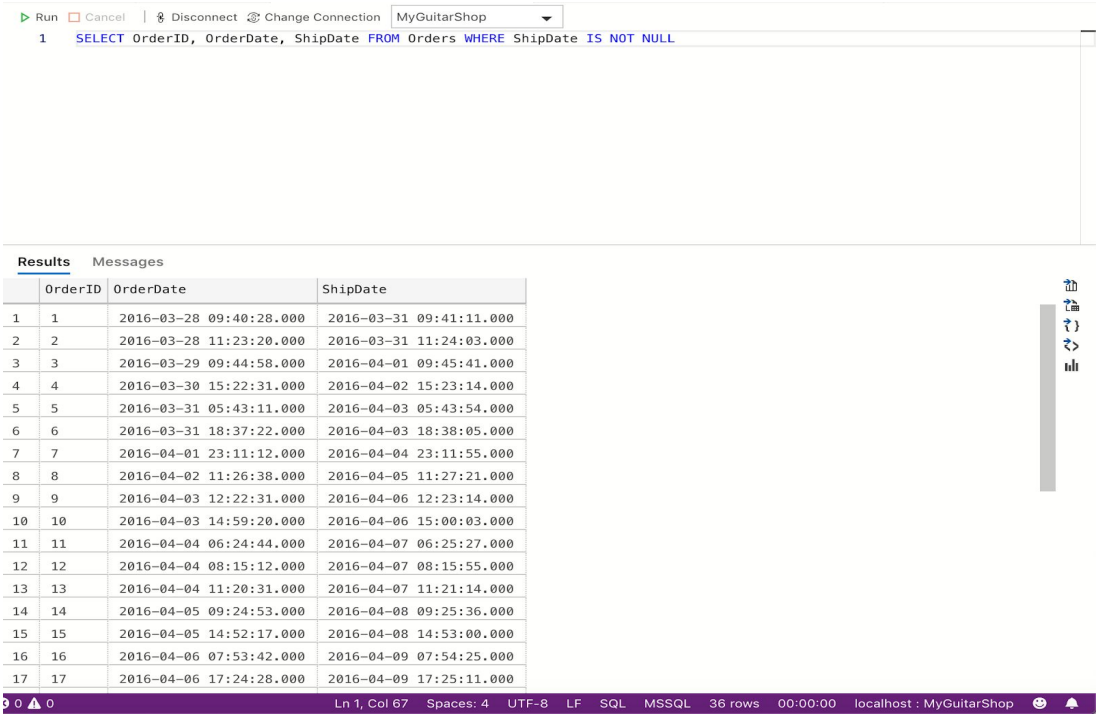
1)

```
1  SELECT FirstName + ', ' + LastName AS FullName FROM Customers WHERE LastName LIKE '[ABC]%' ORDER BY FirstName
```

Results | Messages

	FullName
1	Ahmed, Angalich
2	Alaine, Bergesen
3	Alease, Buemi
4	Alecia, Bubash
5	Aliza, Baltimore
6	Alyce, Arias
7	Ammie, Corrio
8	Angella, Cetta
9	Annabelle, Boord
10	Annmarie, Castros
11	Barbra, Adkin
12	Beatriz, Corrington
13	Brandon, Callaro
14	Brock, Bolognia
15	Buddy, Cloney
16	Cammy, Albares
17	Candida, Corbley

2)

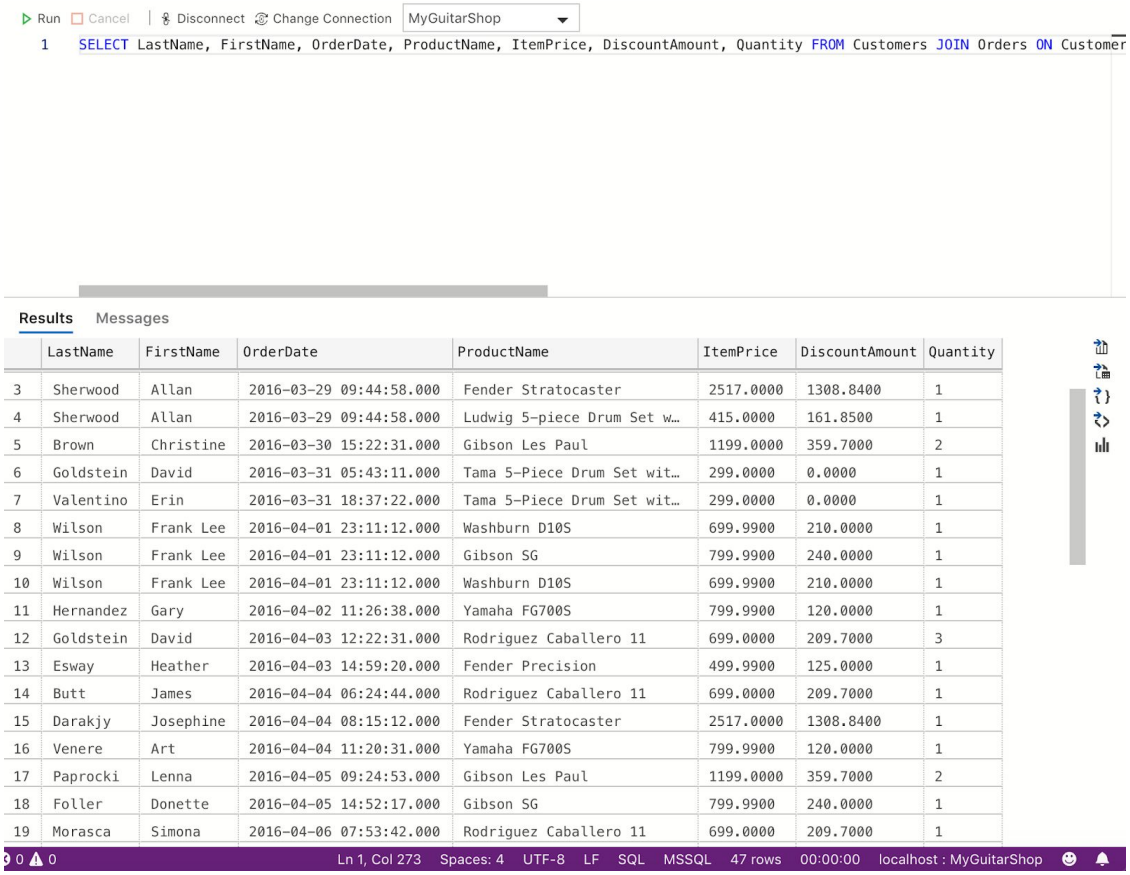


	OrderID	OrderDate	ShipDate
1	1	2016-03-28 09:40:28.000	2016-03-31 09:41:11.000
2	2	2016-03-28 11:23:20.000	2016-03-31 11:24:03.000
3	3	2016-03-29 09:44:58.000	2016-04-01 09:45:41.000
4	4	2016-03-30 15:22:31.000	2016-04-02 15:23:14.000
5	5	2016-03-31 05:43:11.000	2016-04-03 05:43:54.000
6	6	2016-03-31 18:37:22.000	2016-04-03 18:38:05.000
7	7	2016-04-01 23:11:12.000	2016-04-04 23:11:55.000
8	8	2016-04-02 11:26:38.000	2016-04-05 11:27:21.000
9	9	2016-04-03 12:22:31.000	2016-04-06 12:23:14.000
10	10	2016-04-03 14:59:20.000	2016-04-06 15:00:03.000
11	11	2016-04-04 06:24:44.000	2016-04-07 06:25:27.000
12	12	2016-04-04 08:15:12.000	2016-04-07 08:15:55.000
13	13	2016-04-04 11:20:31.000	2016-04-07 11:21:14.000
14	14	2016-04-05 09:24:53.000	2016-04-08 09:25:36.000
15	15	2016-04-05 14:52:17.000	2016-04-08 14:53:00.000
16	16	2016-04-06 07:53:42.000	2016-04-09 07:54:25.000
17	17	2016-04-06 17:24:28.000	2016-04-09 17:25:11.000

C) Essential SQL Skills

1)

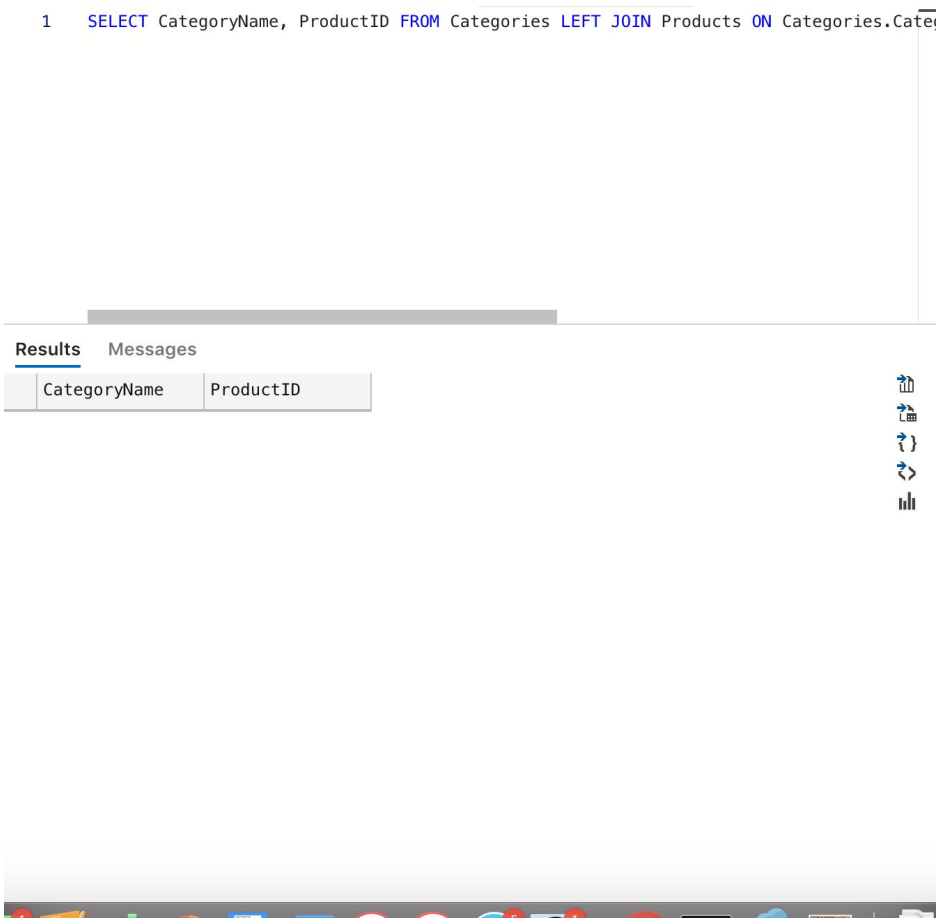
```
SELECT LastName, FirstName, OrderDate, ProductName, ItemPrice, DiscountAmount,
Quantity FROM Customers JOIN Orders ON Customers.CustomerID = Orders.CustomerID JOIN
OrderItems ON Orders.OrderID = OrderItems.OrderID JOIN Products ON
OrderItems.ProductID = Products.ProductID ORDER BY LastName, OrderDate, ProductName
```



	LastName	FirstName	OrderDate	ProductName	ItemPrice	DiscountAmount	Quantity
3	Sherwood	Allan	2016-03-29 09:44:58.000	Fender Stratocaster	2517.0000	1308.8400	1
4	Sherwood	Allan	2016-03-29 09:44:58.000	Ludwig 5-piece Drum Set w…	415.0000	161.8500	1
5	Brown	Christine	2016-03-30 15:22:31.000	Gibson Les Paul	1199.0000	359.7000	2
6	Goldstein	David	2016-03-31 05:43:11.000	Tama 5-Piece Drum Set wit…	299.0000	0.0000	1
7	Valentino	Erin	2016-03-31 18:37:22.000	Tama 5-Piece Drum Set wit…	299.0000	0.0000	1
8	Wilson	Frank Lee	2016-04-01 23:11:12.000	Washburn D10S	699.9900	210.0000	1
9	Wilson	Frank Lee	2016-04-01 23:11:12.000	Gibson SG	799.9900	240.0000	1
10	Wilson	Frank Lee	2016-04-01 23:11:12.000	Washburn D10S	699.9900	210.0000	1
11	Hernandez	Gary	2016-04-02 11:26:38.000	Yamaha FG700S	799.9900	120.0000	1
12	Goldstein	David	2016-04-03 12:22:31.000	Rodriguez Caballero 11	699.0000	209.7000	3
13	Esway	Heather	2016-04-03 14:59:20.000	Fender Precision	499.9900	125.0000	1
14	Butt	James	2016-04-04 06:24:44.000	Rodriguez Caballero 11	699.0000	209.7000	1
15	Darakjy	Josephine	2016-04-04 08:15:12.000	Fender Stratocaster	2517.0000	1308.8400	1
16	Venere	Art	2016-04-04 11:20:31.000	Yamaha FG700S	799.9900	120.0000	1
17	Paprocki	Lenna	2016-04-05 09:24:53.000	Gibson Les Paul	1199.0000	359.7000	2
18	Foller	Donette	2016-04-05 14:52:17.000	Gibson SG	799.9900	240.0000	1
19	Morasca	Simona	2016-04-06 07:53:42.000	Rodriguez Caballero 11	699.0000	209.7000	1

2)

```
SELECT CategoryName, ProductID FROM Categories LEFT JOIN Products ON
Categories.CategoryID = Products.ProductID WHERE Products.ProductID IS NULL
```



3)

Run | Cancel | Disconnect | Change Connection | MyGuitarShop

```
1  SELECT EmailAddress, SUM(ItemPrice * Quantity) AS ItemPriceTotal,
2  SUM(DiscountAmount * Quantity) AS DiscountAmountTotal
3  FROM Customers
4  JOIN Orders ON Customers.CustomerID = Orders.CustomerID
5  JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID
6  GROUP BY EmailAddress
7  ORDER BY ItemPriceTotal ASC
```

Results | Messages

	EmailAddress	ItemPriceTotal	DiscountAmountTotal
1	erinv@gmail.com	299.0000	0.0000
2	barryz@gmail.com	489.9900	186.2000
3	willard@hotmail.com	489.9900	186.2000
4	gladys.rim@rim.org	499.9900	125.0000
5	fletcher.flosi@yahoo.com	598.0000	0.0000
6	allene_iturbide@cox.net	699.0000	209.7000
7	amaclead@gmail.com	699.0000	209.7000
8	calbares@gmail.com	699.0000	209.7000
9	mitsue_tollner@yahoo.com	699.0000	209.7000
10	jbutt@gmail.com	699.0000	209.7000
11	kiley.caldarera@aol.com	699.0000	209.7000
12	leota@hotmail.com	699.0000	209.7000
13	meaghan@hotmail.com	699.9900	210.0000
14	minna_amigon@yahoo.com	799.9900	240.0000
15	vinouye@aol.com	799.9900	240.0000
16	art@venere.org	799.9900	120.0000
17	donette.foller@cox.net	799.9900	240.0000

Ln 7, Col 28 Spaces: 4 UTF-8 LF SQL MSSQL 35 rows 00:00:00 localhost : MyGuitarShop

4)

```
SELECT EmailAddress, COUNT(Orders.OrderID) AS NumberOrders,
SUM((ItemPrice-DiscountAmount)*Quantity) AS TotalAmount FROM Customers JOIN Orders ON
Customers.CustomerID = Orders.CustomerID JOIN OrderItems ON Orders.OrderID =
OrderItems.OrderID WHERE ItemPrice > 600 GROUP BY EmailAddress ORDER BY TotalAmount
DESC
```

Results Messages

	EmailAddress	NumberOrders	TotalAmount
1	david.goldstein@hotmail.c…	2	4267.8500
2	allan.sherwood@yahoo.com	2	2047.4600
3	kris@gmail.com	2	1888.1500
4	yuki_whobrey@aol.com	2	1697.4600
5	lpaprocki@hotmail.com	1	1678.6000
6	christineb@solarone.com	1	1678.6000
7	sage_wieser@cox.net	1	1678.6000
8	gruta@cox.net	1	1678.6000
9	alisha@slusarski.com	1	1678.6000
10	frankwilson@sbcglobal.net	3	1539.9700
11	bette_nicka@cox.net	2	1328.6000
12	heatheresway@mac.com	1	1208.1600
13	josephine_darakjy@darakjy…	1	1208.1600
14	mroyster@royster.com	1	1208.1600
15	mattie@aol.com	1	1208.1600
16	gary_hernandez@yahoo.com	1	679.9900
17	art@venere.org	1	679.9900

1 selected) Spaces: 4 UTF-8 LF SQL MSSQL 30 rows 00:00:00 localhost : MyGuitarShop

5)A

```
SELECT EmailAddress, Orders.OrderID, SUM((ItemPrice-DiscountAmount)*Quantity)
AS TotalAmount FROM Customers JOIN Orders ON Customers.CustomerID = Orders.CustomerID
JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID GROUP BY EmailAddress,
Orders.OrderID
```

```
1   SELECT EmailAddress, Orders.OrderID, SUM((ItemPrice-DiscountAmount)*Quantity)
```

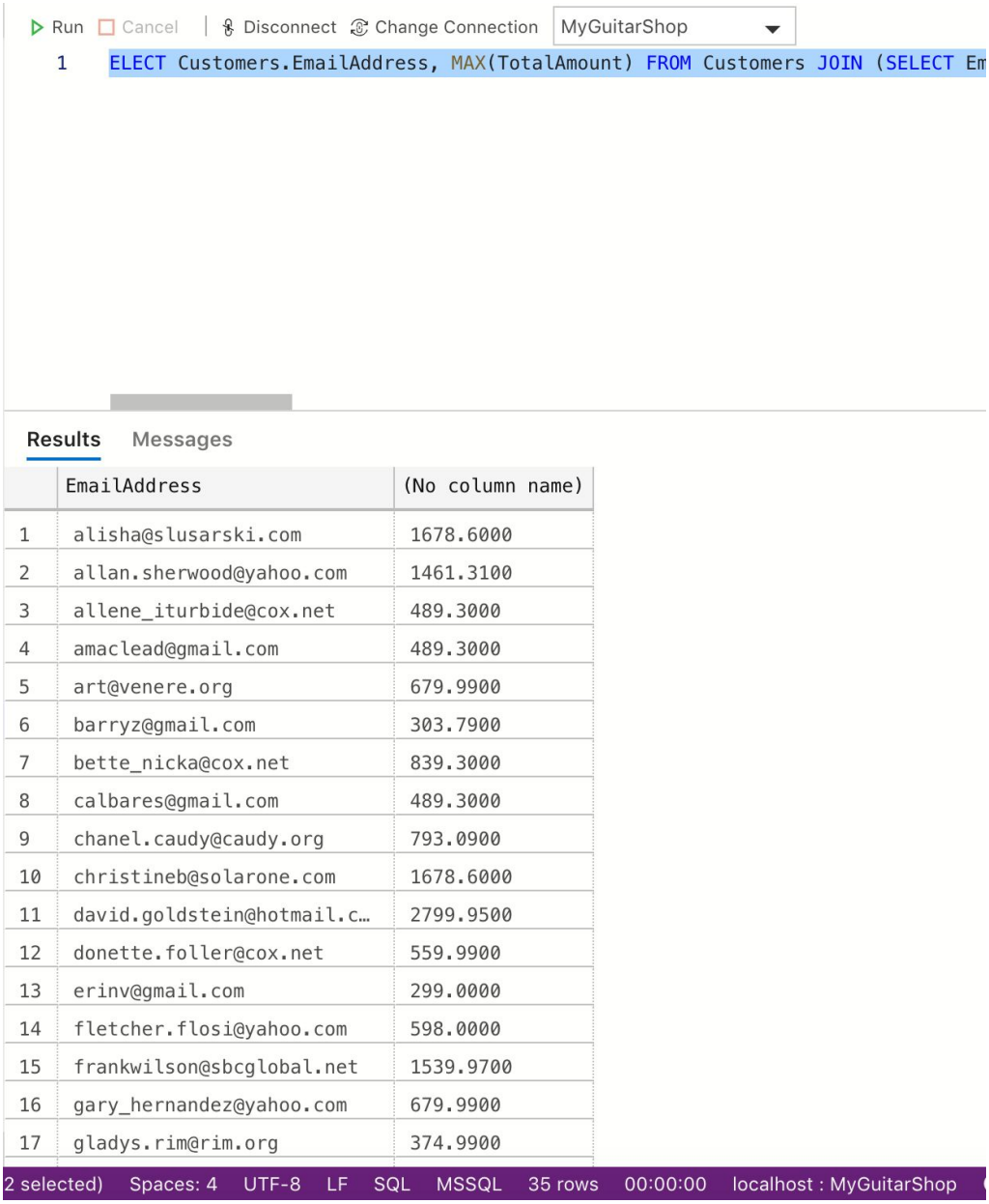
Results | Messages

	EmailAddress	OrderID	TotalAmount
1	allan.sherwood@yahoo.com	1	839.3000
2	barryz@gmail.com	2	303.7900
3	allan.sherwood@yahoo.com	3	1461.3100
4	christineb@solarone.com	4	1678.6000
5	david.goldstein@hotmail.c…	5	299.0000
6	erinv@gmail.com	6	299.0000
7	frankwilson@sbcglobal.net	7	1539.9700
8	gary_hernandez@yahoo.com	8	679.9900
9	david.goldstein@hotmail.c…	9	1467.9000
10	heatheresway@mac.com	10	374.9900
11	jbutt@gmail.com	11	489.3000
12	josephine_darakjy@darakjy…	12	1208.1600
13	art@venere.org	13	679.9900
14	lpaprocki@hotmail.com	14	1678.6000
15	donette.foller@cox.net	15	559.9900
16	simona@morasca.com	16	742.4500
17	mitsue_tollner@yahoo.com	17	489.3000

6 selected) Spaces: 4 UTF-8 LF SQL MSSQL 41 rows 00:00:00 localhost : MyGuitarShop

5B)

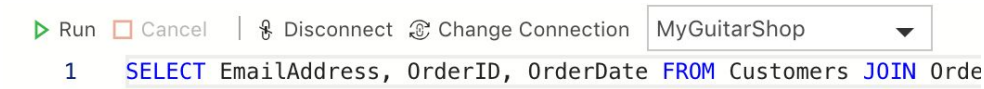
```
SELECT Customers.EmailAddress, MAX(TotalAmount) FROM Customers JOIN (SELECT
EmailAddress, Orders.OrderID, SUM((ItemPrice-DiscountAmount)*Quantity) AS TotalAmount
FROM Customers JOIN Orders ON Customers.CustomerID = Orders.CustomerID JOIN OrderItems
ON Orders.OrderID = OrderItems.OrderID GROUP BY EmailAddress, Orders.OrderID) AS
CustomerTotals ON Customers.EmailAddress = CustomerTotals.EmailAddress GROUP BY
Customers.EmailAddress
```



	EmailAddress	(No column name)
1	alisha@slusarski.com	1678.6000
2	allan.sherwood@yahoo.com	1461.3100
3	allene_iturbide@cox.net	489.3000
4	amaclead@gmail.com	489.3000
5	art@venere.org	679.9900
6	barryz@gmail.com	303.7900
7	bette_nicka@cox.net	839.3000
8	calbares@gmail.com	489.3000
9	chanel.caudy@caudy.org	793.0900
10	christineb@solarone.com	1678.6000
11	david.goldstein@hotmail.c…	2799.9500
12	donette.foller@cox.net	559.9900
13	erinv@gmail.com	299.0000
14	fletcher.flosi@yahoo.com	598.0000
15	frankwilson@sbcglobal.net	1539.9700
16	gary_hernandez@yahoo.com	679.9900
17	gladys.rim@rim.org	374.9900

6)

```
SELECT EmailAddress, OrderID, OrderDate FROM Customers JOIN Orders ON
Customers.CustomerID = Orders.CustomerID WHERE Orders.Orderdate = (SELECT
MAX(OrderDate) FROM Orders WHERE Customers.CustomerID = Orders.CustomerID)
```

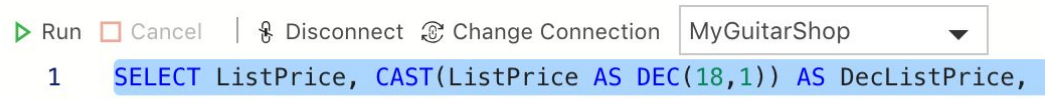


Results Messages

	EmailAddress	OrderID	OrderDate
1	chanel.caudy@caudy.org	41	2016-05-09 07:52:55.000
2	allene_iturbide@cox.net	40	2016-05-08 21:41:29.000
3	alisha@slusarski.com	38	2016-05-08 11:41:24.000
4	mroyster@royster.com	37	2016-05-06 14:15:21.000
5	willard@hotmail.com	36	2016-05-04 12:31:33.000
6	vinouye@aol.com	35	2016-05-04 03:52:23.000
7	bette_nicka@cox.net	39	2016-05-08 22:22:26.000
8	fletcher.flosi@yahoo.com	33	2016-05-01 09:11:51.000
9	yuki_whobrey@aol.com	31	2016-04-29 06:47:14.000
10	gladys.rim@rim.org	30	2016-04-27 16:21:31.000
11	meaghan@hotmail.com	28	2016-04-21 17:52:24.000
12	mattie@aol.com	27	2016-04-20 09:17:52.000
13	calbares@gmail.com	26	2016-04-20 08:14:45.000
14	gruta@cox.net	25	2016-04-20 08:23:32.000
15	kiley.caldarera@aol.com	24	2016-04-17 17:40:22.000
16	amaclead@gmail.com	23	2016-04-14 07:59:31.000
17	minna_amigon@yahoo.com	21	2016-04-11 08:21:32.000

7)

```
SELECT ListPrice, CAST(ListPrice AS DEC(18,1)) AS DecListPrice, CONVERT(INT,
ListPrice) AS IntListPrice, CAST(ListPrice AS INT) AS IntConListPrice FROM Products
```

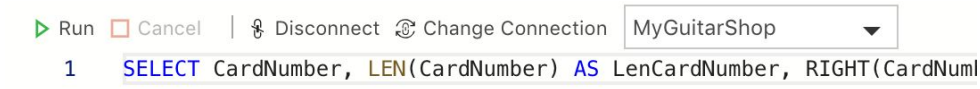


Results Messages

	ListPrice	DecListPrice	IntListPrice	IntConListPrice
1	699.0000	699.0	699	699
2	1199.0000	1199.0	1199	1199
3	2517.0000	2517.0	2517	2517
4	489.9900	490.0	490	490
5	299.0000	299.0	299	299
6	415.0000	415.0	415	415
7	799.9900	800.0	800	800
8	499.9900	500.0	500	500
9	699.9900	700.0	700	700
10	799.9900	800.0	800	800

8)

```
SELECT CardNumber, LEN(CardNumber) AS LenCardNumber, RIGHT(CardNumber,4) AS Last4Digits, 'XXXX-XXXX-XXXX-' + RIGHT(CardNumber,4) AS SecureLast4 FROM Orders
```



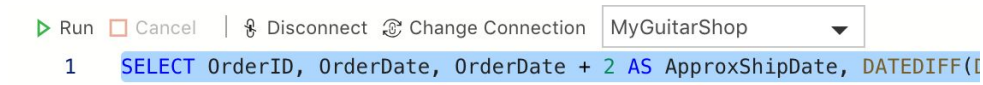
Results Messages

	CardNumber	LenCardNumber	Last4Digits	SecureLast4
1	4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
2	4012888888881881	16	1881	XXXX-XXXX-XXXX-1881
3	4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
4	3782822463100005	16	0005	XXXX-XXXX-XXXX-0005
5	4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
6	6011111111111117	16	1117	XXXX-XXXX-XXXX-1117
7	5555555555554444	16	4444	XXXX-XXXX-XXXX-4444
8	4012888888881881	16	1881	XXXX-XXXX-XXXX-1881
9	4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
10	4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
11	4012888888881881	16	1881	XXXX-XXXX-XXXX-1881
12	4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
13	4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
14	4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
15	3782822463100005	16	0005	XXXX-XXXX-XXXX-0005
16	4111111111111111	16	1111	XXXX-XXXX-XXXX-1111
17	4111111111111111	16	1111	XXXX-XXXX-XXXX-1111

n 1, Col 156 Spaces: 4 UTF-8 LF SQL MSSQL 41 rows 00:00:00 localhost : MyG

9A)

```
SELECT OrderID, OrderDate, OrderDate + 2 AS ApproxShipDate,
DATEDIFF(DAY,OrderDate,ShipDate) AS Difference FROM Orders
```



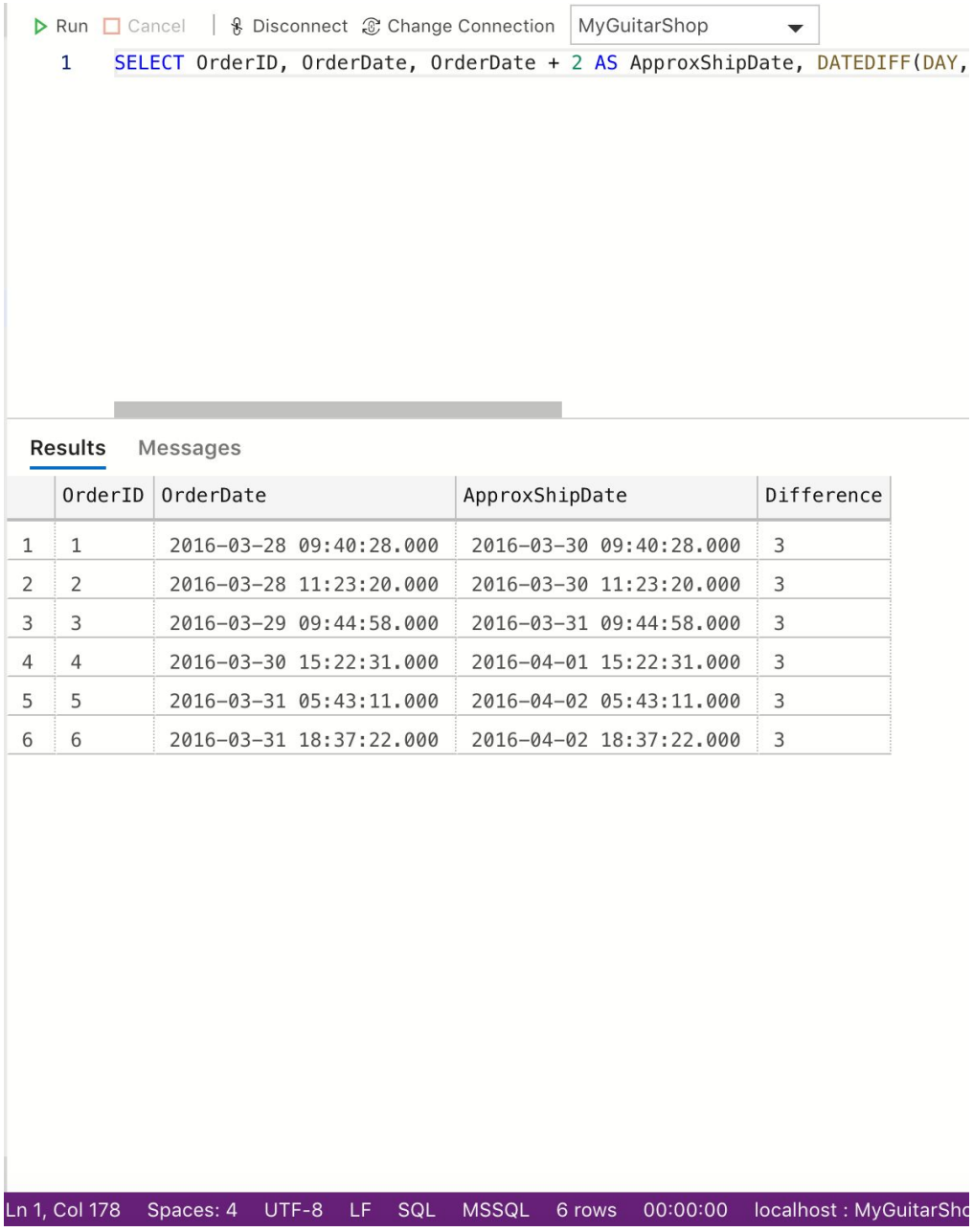
Results | Messages

	OrderID	OrderDate	ApproxShipDate	Difference
1	1	2016-03-28 09:40:28.000	2016-03-30 09:40:28.000	3
2	2	2016-03-28 11:23:20.000	2016-03-30 11:23:20.000	3
3	3	2016-03-29 09:44:58.000	2016-03-31 09:44:58.000	3
4	4	2016-03-30 15:22:31.000	2016-04-01 15:22:31.000	3
5	5	2016-03-31 05:43:11.000	2016-04-02 05:43:11.000	3
6	6	2016-03-31 18:37:22.000	2016-04-02 18:37:22.000	3
7	7	2016-04-01 23:11:12.000	2016-04-03 23:11:12.000	3
8	8	2016-04-02 11:26:38.000	2016-04-04 11:26:38.000	3
9	9	2016-04-03 12:22:31.000	2016-04-05 12:22:31.000	3
10	10	2016-04-03 14:59:20.000	2016-04-05 14:59:20.000	3
11	11	2016-04-04 06:24:44.000	2016-04-06 06:24:44.000	3
12	12	2016-04-04 08:15:12.000	2016-04-06 08:15:12.000	3
13	13	2016-04-04 11:20:31.000	2016-04-06 11:20:31.000	3
14	14	2016-04-05 09:24:53.000	2016-04-07 09:24:53.000	3
15	15	2016-04-05 14:52:17.000	2016-04-07 14:52:17.000	3
16	16	2016-04-06 07:53:42.000	2016-04-08 07:53:42.000	3
17	17	2016-04-06 17:24:28.000	2016-04-08 17:24:28.000	3

8 selected) Spaces: 4 UTF-8 LF SQL MSSQL 41 rows 00:00:00 localhost : MyGuita

9B)

```
SELECT OrderID, OrderDate, OrderDate + 2 AS ApproxShipDate,
DATEDIFF(DAY,OrderDate,ShipDate) AS Difference FROM Orders WHERE MONTH(OrderDate) =
'03' AND YEAR(OrderDate) = '2016'
```



10)

```
INSERT INTO Customers (EmailAddress,Password,FirstName,LastName) VALUES
('ellie@Krieger.com','','Ellie','Kreiger')
```

```
1   INSERT INTO Customers (EmailAddress,Password,FirstName,LastName) VALUES ('el
```

Messages

6:26:39 PM Started executing query at Line 1

(1 row affected)

Total execution time: 00:00:00.025

Ln 1, Col 115 Spaces: 4 UTF-8 LF SQL MSSQL 0 rows 00:00:00 localhost : MyGuitarShop

Results)

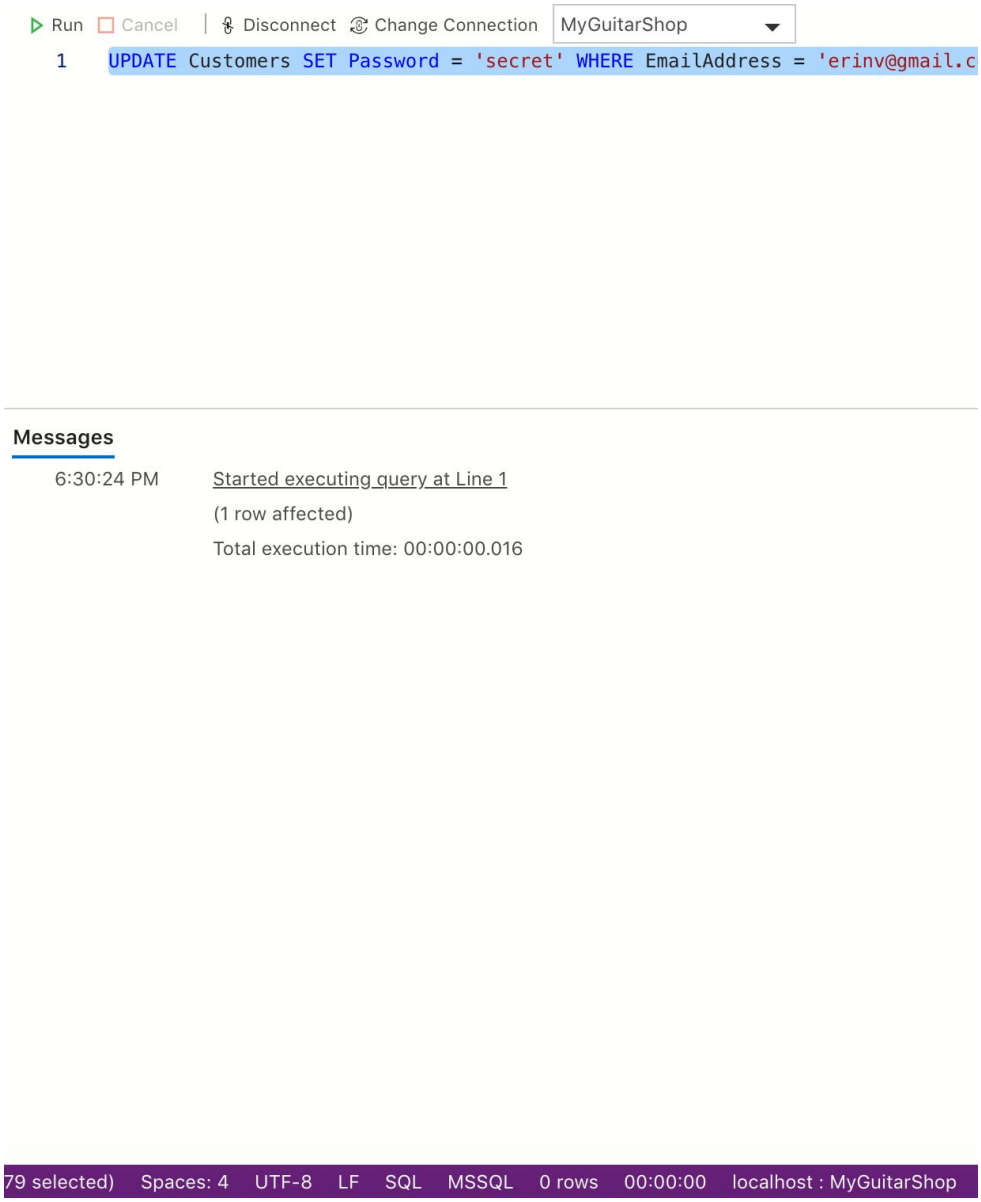
```
SELECT EmailAddress,Password,FirstName,LastName FROM Customers WHERE FirstName =
'Ellie'
```

Results Messages

	EmailAddress	Password	FirstName	LastName
1	ellie@Krieger.com		Ellie	Kreiger

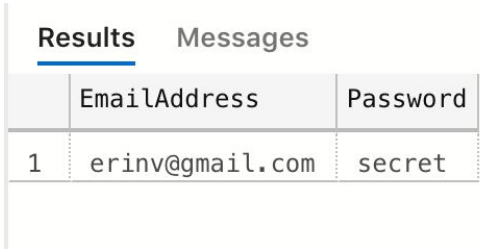
11)

```
UPDATE Customers SET Password = 'secret' WHERE EmailAddress = 'erinv@gmail.com'
```



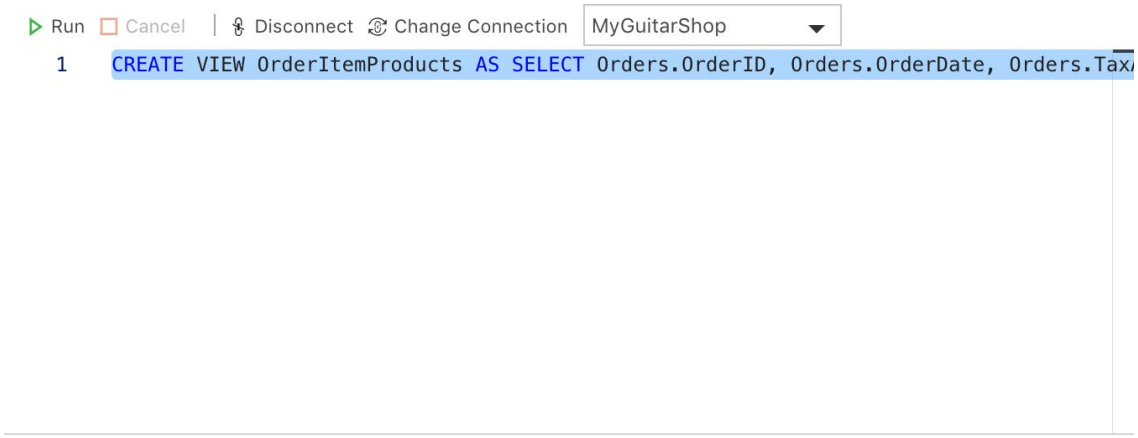
Results)

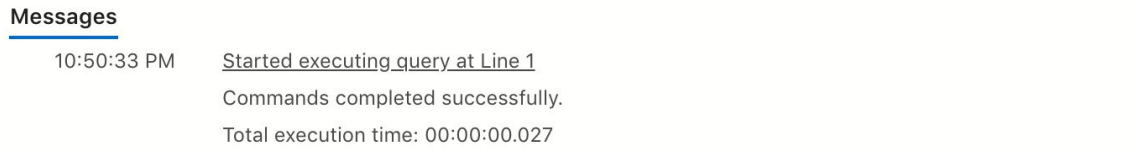
```
SELECT EmailAddress, Password FROM Customers WHERE EmailAddress = 'erinv@gmail.com'
```

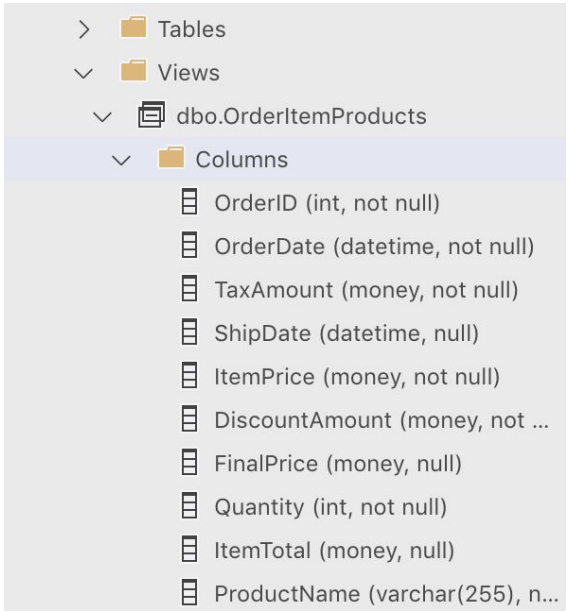


1)

```
CREATE VIEW OrderItemProducts AS SELECT Orders.OrderID, Orders.OrderDate,
Orders.TaxAmount, Orders.ShipDate, OrderItems.ItemPrice, OrderItems.DiscountAmount,
(OrderItems.ItemPrice-OrderItems.DiscountAmount) AS FinalPrice,
OrderItems.Quantity,(OrderItems.Quantity *
(OrderItems.ItemPrice-OrderItems.DiscountAmount)) AS ItemTotal, Products.ProductName
FROM Orders JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID JOIN Products ON
OrderItems.ProductID = Products.ProductID
```







2)

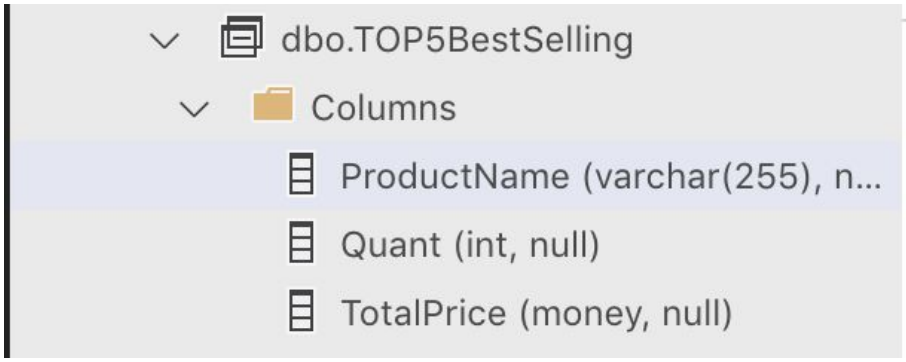
```
CREATE VIEW [TOP5BestSelling] AS SELECT TOP(5) ProductName, COUNT(Products.ProductID)
AS Quant, SUM(Quantity*ListPrice) AS TotalPrice FROM Products JOIN OrderItems ON
Products.ProductID = OrderItems.ProductID GROUP BY ProductName
```

Messages

12:26:25 PM Started executing query at Line 1

Commands completed successfully.

Total execution time: 00:00:00.017



Results Messages

	ProductName	Quant	TotalPrice
1	Fender Precision	2	1599.9800
2	Fender Stratocaster	7	4893.0000
3	Gibson Les Paul	7	14388.0000
4	Gibson SG	5	22653.0000
5	Hofner Icon	3	1499.9700

3)

```
CREATE PROC spUpdateProductDiscount
   @Product_ID INT,
   @Discount_Percent INT OUTPUT
AS
IF @Discount_Percent < 0
   THROW 50001, 'Cannot be a negative value!', 1;
UPDATE PRODUCTS
SET DiscountPercent = @Discount_Percent
WHERE ProductID = @Product_ID
```

Before Procedure)

Results Messages

	ProductID	CategoryID	ProductCode	ProductName	Description	ListPrice	DiscountPercent	DateAdded
1	1	1	strat	Fender Stratocaster	The Fender Stratocaster i…	699.0000	30.0000	2015-10-30 09:32:40

1st Exec Statement)

```
EXEC spUpdateProductDiscount @Product_ID = 3, @Discount_Percent = 15
```

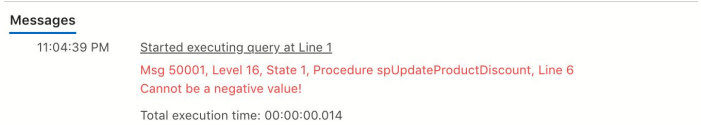
After Procedure)

Results Messages

	ProductID	CategoryID	ProductCode	ProductName	Description	ListPrice	DiscountPercent	DateAdded
1	1	1	strat	Fender Stratocaster	The Fender Stratocaster i…	699.0000	15.0000	2015-10-30 09:32:40

2nd Exec Statement)

```
EXEC spUpdateProductDiscount @Product_ID = 2, @Discount_Percent = -2
```



4)

```
CREATE PROC CommonFactors AS
BEGIN
  DECLARE @counter INT
  DECLARE @fact10 INT
  DECLARE @fact20 INT
  DECLARE @msg VARCHAR(100)
  SET @fact10 = 10;
  SET @fact20 = 20;
  SET @counter = 1;
  PRINT 'Factors of 10 and 20: '
  WHILE(@counter <= 10)
      BEGIN
          IF(@fact10 % @counter = 0 AND @fact20 % @counter = 0)
              PRINT @counter
      SET @counter = @counter + 1;
      END
END;
```

Exec Statement)

```
EXEC CommonsFactors
```

Messages

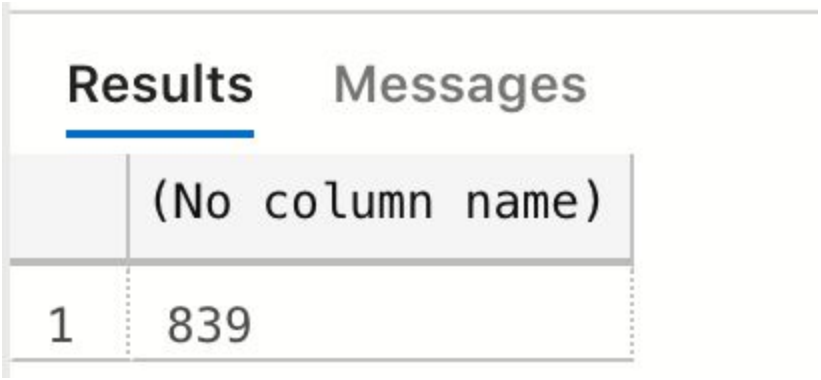
```
11:24:18 PM    Started executing query at Line 1
               Factors of 10 and 20:
               1
               2
               5
               10
               Total execution time: 00:00:00.003
```

5)A

```
CREATE FUNCTION fnDiscountPrice(@Item_ID INT)
  RETURNS INT
BEGIN
  RETURN (SELECT (ItemPrice-DiscountAmount) FROM OrderItems WHERE ItemID = @Item_ID);
END;
```

Select Statement)

```
SELECT dbo.fnDiscountPrice(1)
```



5)B

```
CREATE FUNCTION fnItemTotal
   (@Item_ID INT)
RETURNS INT
BEGIN
RETURN
(SELECT dbo.fnDiscountPrice(@Item_ID) * (SELECT Quantity FROM OrderItems WHERE ItemID
= @Item_ID));
END
```

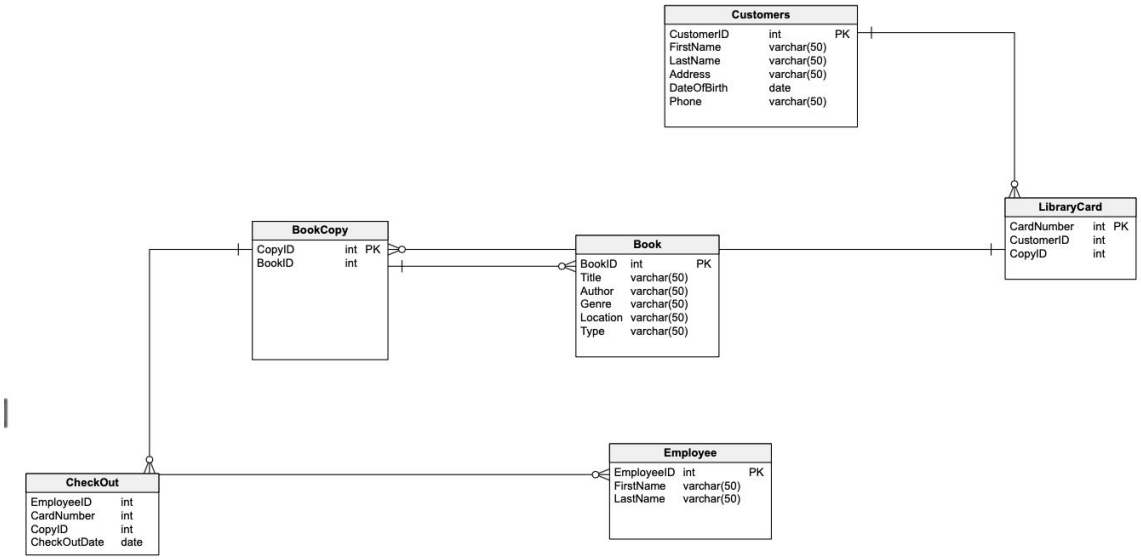
Select Statement)

```
SELECT dbo.fnItemTotals(5)
```

Results Messages

	(No column name)
1	1678

D) Database Design)



The tables for the database above were already in third normalization form since they were all unique tables with their own unique elements.

Below are the following relationships
Books are allowed to have many copies, but copies can only be made of one book
Customers can have one Library Card, but many library cards can be given to many customers
A library card can have many different book copies, and book copies can be assigned to many different library cards
Employees can check out many customers, and customers can be checked out by many employees

These relationships are portrayed above
Books will have a BookId, Title, Author, Genre, Location, and type to know the details about the book
BookCopy will have the CopyID and BookID of that copy.
Customers will have a CustomerID, FirstName, LastName, Address, DateofBirth and Phone.
The LibraryCard references the Customer Table where it is linked to the CustomerID
The LibraryCard will have a CardNumber, CustomerID, and CopyID. This allows us to know which CopyID the Customer has
The Employee will have an EmployeeID, FirstName and LastName

The CheckOut will have an EmployeeID, CardNumber,CopyID, and CheckOutDate. This allows us to know which Employee completed the checkout to which library card of which book, and know when the copy was checked out.

Remarks)

Overall I loved this project. It was a great way to implement everything we've learned up to this point. I breezed by the Part A and B, but had some trouble with the scripts in part C. It took me about three hours to complete that part. Before this project, I wasn't really sure on how to implement views, procedures, and functions, but after struggling for multiple hours I was able to figure out how they work. Now I can use knowledge to complete the next lab. The database design was interesting as we get to actually design a database that is implemented in libraries. I'm still not 100% positive about designing them, but feel like I did a pretty good job designing the tables and explaining the relationships. I actually had fun working on this project and plan on putting it on my resume. I am looking forward to working on Project 2