

Exercise 7 – Select Statements

Stewart Johnston
CIS 150 – Intro to Database Administration
NCMC
johnstons1@student.ncmich.edu

October 20, 2018

Contents

1	SELECT four columns	1
2	Return FullName	3
3	Select name, listprice, dateadded	4
4	Return Name, Price, Discount, etc	5
5	Return Item ID, Price, Discount, etc from OrderItems	7
6	Return attributes where one is null	8
7	SELECT without FROM	9

List of Figures

1	Select four columns of the products table, per section 1	2
2	Summarized FullName, as per 2	3
3	Returns Name, Price, and Date, per 3	5
4	Per 4, return Name, Price, and Discount, ordered by DiscountPrice	6
5	Per 5, return ID, Price, etc from OrderItems	8
6	Per 6, return ID, Dates, from Orders, where ShipDate is null	9
7	Per 7, SELECT custom values into columns w/o a FROM clause	10

1 SELECT four columns

See figure 1

```

SELECT ProductCode
,ProductName
,ListPrice
,DiscountPercent
FROM Products
ORDER BY ListPrice DESC

```

Figure 1: Select four columns of the products table, per section 1

The screenshot shows a SQL Server Enterprise Manager window with a query executed successfully. The query selects four columns from the Products table, ordered by ListPrice in descending order. The results are displayed in a table with 10 rows.

	ProductCode	ProductName	ListPrice	DiscountPercent
1	sg	Gibson SG	2517.00	52.00
2	les_paul	Gibson Les Paul	1199.00	30.00
3	precision	Fender Precision	799.99	30.00
4	tama	Tama 5-Piece Drum Set with Cymbals	799.99	15.00
5	ludwig	Ludwig 5-piece Drum Set with Cymbals	699.99	30.00
6	strat	Fender Stratocaster	699.00	30.00
7	hofner	Hofner Icon	499.99	25.00
8	fg700s	Yamaha FG700S	489.99	38.00
9	rodriguez	Rodriguez Caballero 11	415.00	39.00
10	washburn	Washburn D10S	299.00	0.00

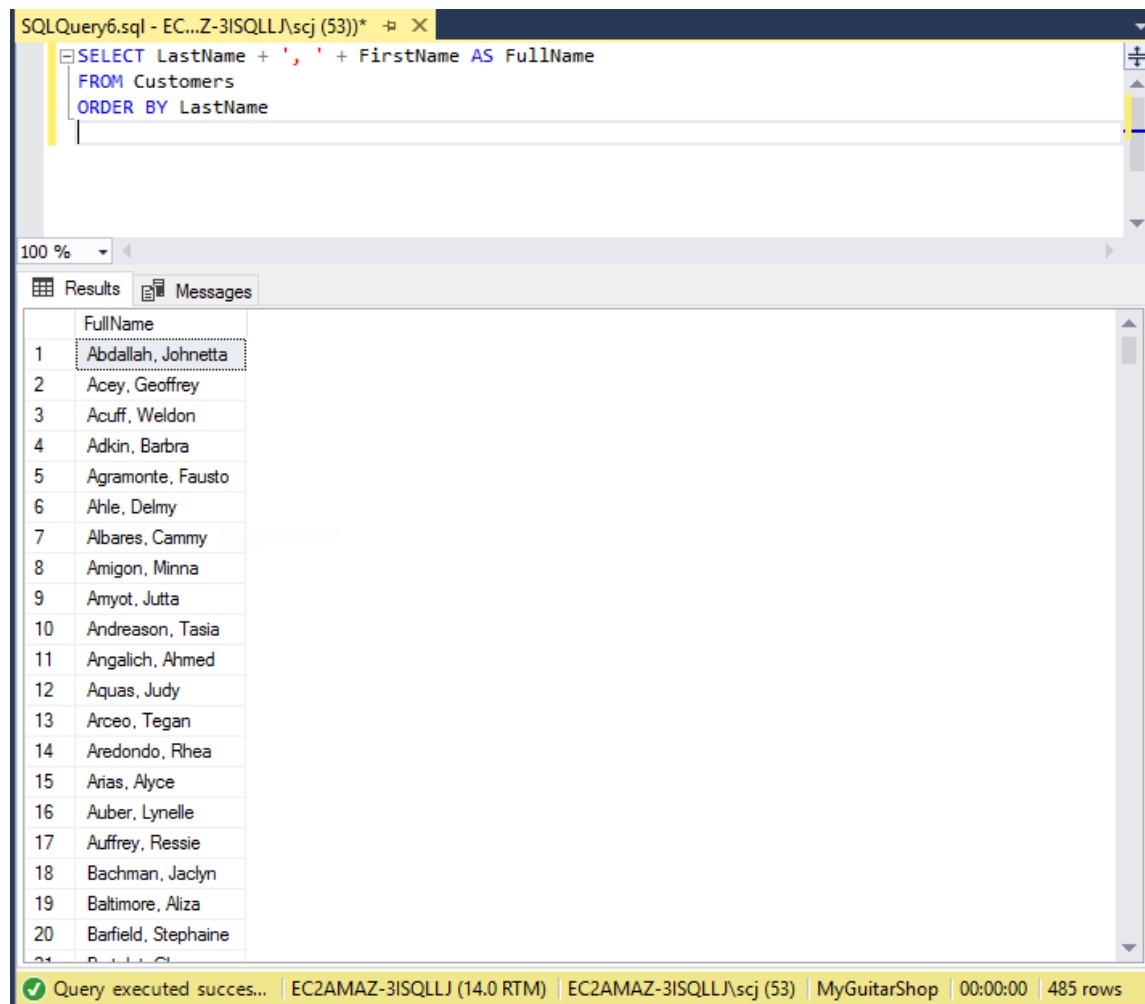
Query executed success... | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLLJ\scj (53) | MyGuitarShop | 00:00:00 | 10 rows

2 Return FullName

```
SELECT LastName + ', ' + FirstName AS FullName  
FROM Customers  
ORDER BY LastName
```

See figure 2

Figure 2: Summarized FullName, as per 2



The screenshot shows a SQL Server Enterprise Manager window with a query titled 'SQLQuery6.sql - EC...Z-3ISQLLJ\scj (53)*'. The query is:
`SELECT LastName + ', ' + FirstName AS FullName
FROM Customers
ORDER BY LastName`
The 'Results' tab is active, displaying a table with 485 rows. The first 20 rows are visible, showing the concatenated 'FullName' for each customer, ordered by last name. The status bar at the bottom indicates 'Query executed succes...' (sic), 'EC2AMAZ-3ISQLLJ (14.0 RTM)', 'EC2AMAZ-3ISQLLJ\scj (53)', 'MyGuitarShop', '00:00:00', and '485 rows'.

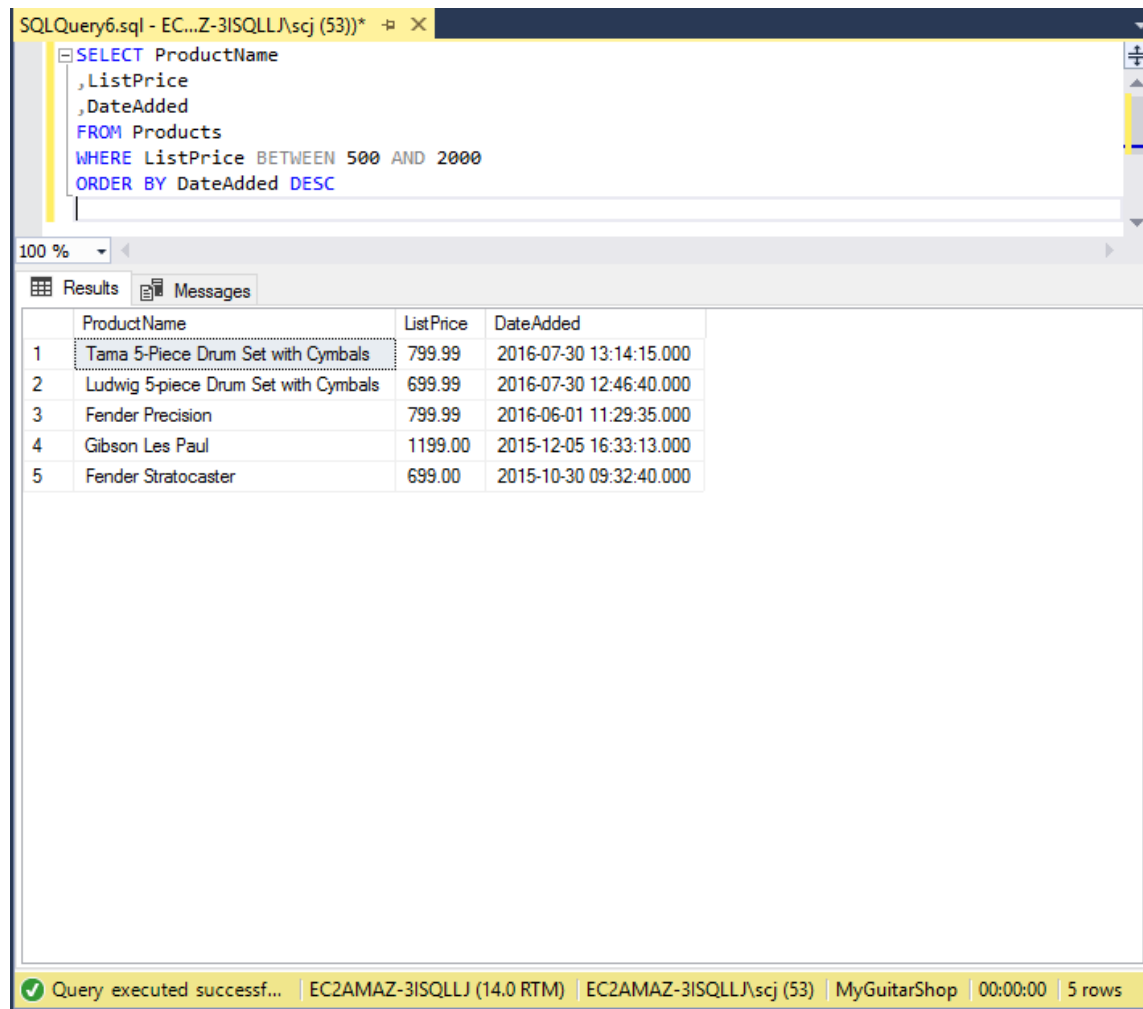
	FullName
1	Abdallah, Johnetta
2	Acey, Geoffrey
3	Acuff, Weldon
4	Adkin, Barbra
5	Agramonte, Fausto
6	Ahle, Delmy
7	Albares, Cammy
8	Amigon, Minna
9	Amyot, Jutta
10	Andreason, Tasia
11	Angalich, Ahmed
12	Aquas, Judy
13	Arceo, Tegan
14	Aredondo, Rhea
15	Arias, Alyce
16	Auber, Lynelle
17	Auffrey, Ressie
18	Bachman, Jaclyn
19	Baltimore, Aliza
20	Barfield, Stephaine

3 Select name, listprice, dateadded

```
SELECT ProductName  
  ,ListPrice  
  ,DateAdded  
FROM Products  
WHERE ListPrice BETWEEN 500 AND 2000  
ORDER BY DateAdded DESC
```

See figure 3

Figure 3: Returns Name, Price, and Date, per 3



The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results pane. The query editor contains the following SQL code:

```
SELECT ProductName
, ListPrice
, DateAdded
FROM Products
WHERE ListPrice BETWEEN 500 AND 2000
ORDER BY DateAdded DESC
```

The results pane displays a table with 5 rows and 4 columns: ProductName, ListPrice, and DateAdded. The data is as follows:

	ProductName	ListPrice	DateAdded
1	Tama 5-Piece Drum Set with Cymbals	799.99	2016-07-30 13:14:15.000
2	Ludwig 5-piece Drum Set with Cymbals	699.99	2016-07-30 12:46:40.000
3	Fender Precision	799.99	2016-06-01 11:29:35.000
4	Gibson Les Paul	1199.00	2015-12-05 16:33:13.000
5	Fender Stratocaster	699.00	2015-10-30 09:32:40.000

The status bar at the bottom indicates: Query executed successfully... | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLLJ\scj (53) | MyGuitarShop | 00:00:00 | 5 rows

4 Return Name, Price, Discount, etc

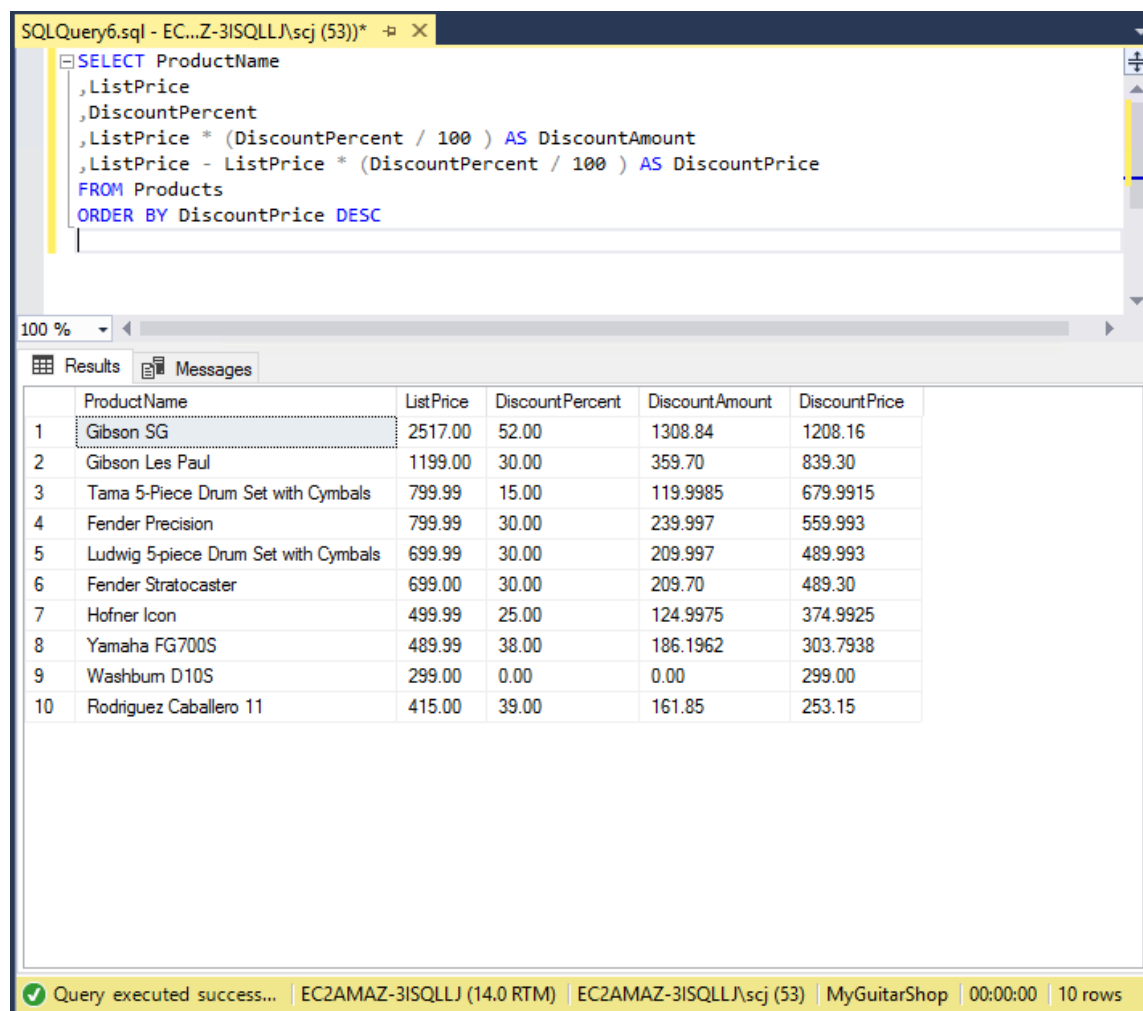
See figure 4

```

SELECT ProductName
,ListPrice
,DiscountPercent
,ListPrice * (DiscountPercent / 100 ) AS DiscountAmount
,ListPrice - ListPrice * (DiscountPercent / 100 ) AS DiscountPrice /*Because
aliases aren't reusable in the same scope, much to my annoyance, and we haven't
learned subqueries or CTEs yet.*/
FROM Products
ORDER BY DiscountPrice DESC

```

Figure 4: Per 4, return Name, Price, and Discount, ordered by DiscountPrice



SQLQuery6.sql - EC...Z-3ISQLLJ\scj (53)

```

SELECT ProductName
,ListPrice
,DiscountPercent
,ListPrice * (DiscountPercent / 100 ) AS DiscountAmount
,ListPrice - ListPrice * (DiscountPercent / 100 ) AS DiscountPrice
FROM Products
ORDER BY DiscountPrice DESC

```

100 %

Results Messages

	ProductName	ListPrice	DiscountPercent	DiscountAmount	DiscountPrice
1	Gibson SG	2517.00	52.00	1308.84	1208.16
2	Gibson Les Paul	1199.00	30.00	359.70	839.30
3	Tama 5-Piece Drum Set with Cymbals	799.99	15.00	119.9985	679.9915
4	Fender Precision	799.99	30.00	239.997	559.993
5	Ludwig 5-piece Drum Set with Cymbals	699.99	30.00	209.997	489.993
6	Fender Stratocaster	699.00	30.00	209.70	489.30
7	Hofner Icon	499.99	25.00	124.9975	374.9925
8	Yamaha FG700S	489.99	38.00	186.1962	303.7938
9	Washburn D10S	299.00	0.00	0.00	299.00
10	Rodriguez Caballero 11	415.00	39.00	161.85	253.15

Query executed success... | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLLJ\scj (53) | MyGuitarShop | 00:00:00 | 10 rows

5 Return Item ID, Price, Discount, etc from OrderItems

```
SELECT ItemID
,ItemPrice
,DiscountAmount
,Quantity
,ItemPrice * Quantity AS PriceTotal
,DiscountAmount * Quantity AS DiscountTotal
,(ItemPrice - DiscountAmount) * Quantity AS ItemTotal
FROM OrderItems
WHERE ((ItemPrice - DiscountAmount) * Quantity) > 500 /*Again, no subqueries
yet. Kind of dumb that this is how it works by default, because otherwise, I'd
happily use ItemTotal*/
ORDER BY ItemTotal DESC
```

See figure 5

Figure 5: Per 5, return ID, Price, etc from OrderItems

The screenshot shows a SQL query window with the following text:

```
SELECT ItemID
,ItemPrice
,DiscountAmount
,Quantity
,ItemPrice * Quantity AS PriceTotal
,DiscountAmount * Quantity AS DiscountTotal
,(ItemPrice - DiscountAmount) * Quantity AS ItemTotal
FROM OrderItems
WHERE ((ItemPrice - DiscountAmount) * Quantity) > 500 /*Again, no subqueries
yet. Kind of dumb that this is how it works by default, because otherwise, I'd
happily use ItemTotal*/
ORDER BY ItemTotal DESC
```

Below the query, the 'Results' tab is active, displaying a table with 8 columns: ItemID, ItemPrice, DiscountAmount, Quantity, PriceTotal, DiscountTotal, and ItemTotal. The table contains 16 rows of data. The status bar at the bottom indicates 'Query executed success...', 'EC2AMAZ-3ISQLLJ (14.0 RTM)', 'EC2AMAZ-3ISQLLJ\scj (53)', 'MyGuitarShop', '00:00:00', and '24 rows'.

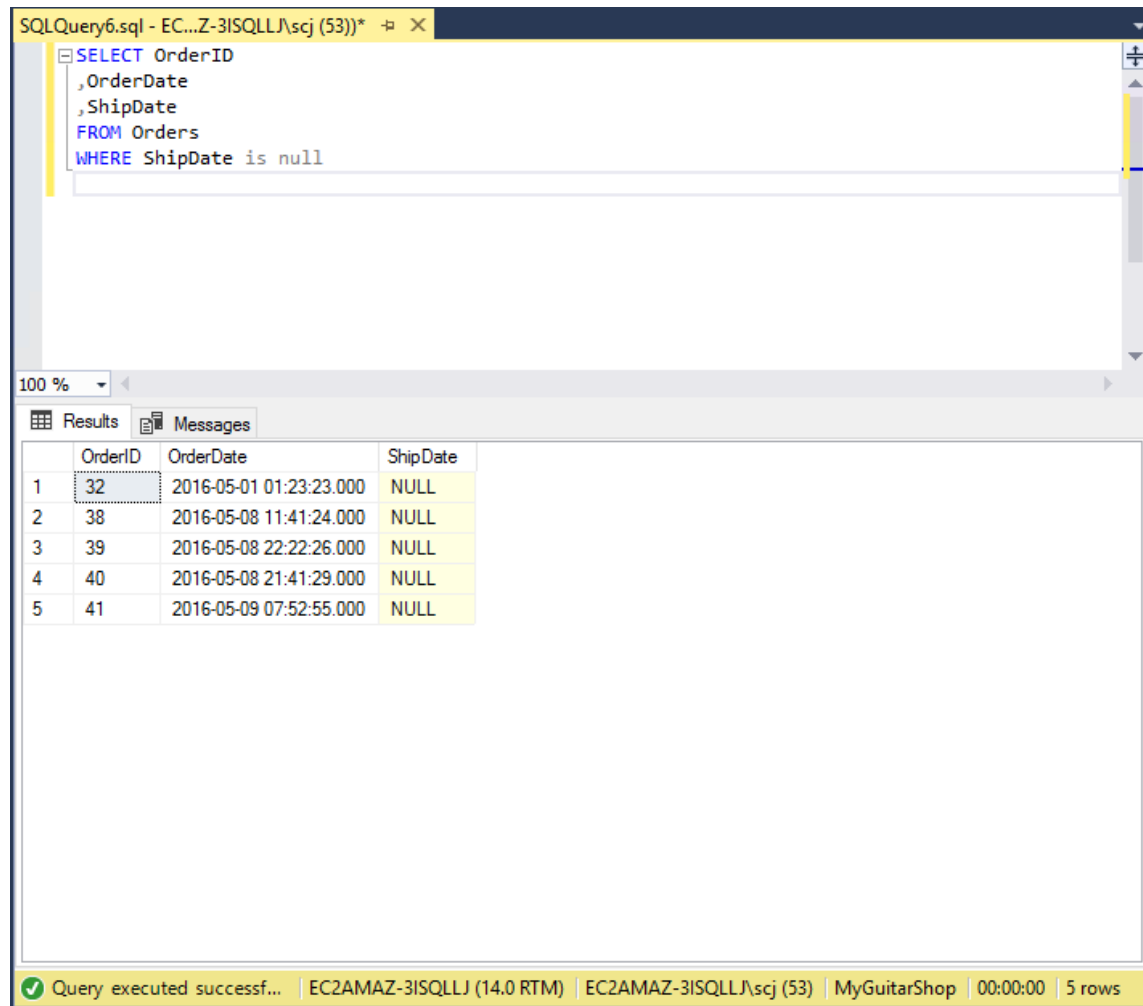
	ItemID	ItemPrice	DiscountAmount	Quantity	PriceTotal	DiscountTotal	ItemTotal
1	33	799.99	240.00	5	3999.95	1200.00	2799.95
2	23	1199.00	359.70	2	2398.00	719.40	1678.60
3	5	1199.00	359.70	2	2398.00	719.40	1678.60
4	17	1199.00	359.70	2	2398.00	719.40	1678.60
5	29	1199.00	359.70	2	2398.00	719.40	1678.60
6	43	1199.00	359.70	2	2398.00	719.40	1678.60
7	12	699.00	209.70	3	2097.00	629.10	1467.90
8	15	2517.00	1308.84	1	2517.00	1308.84	1208.16
9	3	2517.00	1308.84	1	2517.00	1308.84	1208.16
10	26	2517.00	1308.84	1	2517.00	1308.84	1208.16
11	35	2517.00	1308.84	1	2517.00	1308.84	1208.16
12	37	2517.00	1308.84	1	2517.00	1308.84	1208.16
13	42	2517.00	1308.84	1	2517.00	1308.84	1208.16
14	31	2517.00	1308.84	1	2517.00	1308.84	1208.16
15	39	1199.00	359.70	1	1199.00	359.70	839.30
16	1	1199.00	359.70	1	1199.00	359.70	839.30

6 Return attributes where one is null

```
SELECT OrderID
,OrderDate
,ShipDate
FROM Orders
WHERE ShipDate is null
```


See figure 6

Figure 6: Per 6, return ID, Dates, from Orders, where ShipDate is null



The screenshot shows a SQL Server Enterprise Manager window titled "SQLQuery6.sql - EC...Z-3ISQLL\scj (53))". The query editor contains the following SQL code:

```
SELECT OrderID
,OrderDate
,ShipDate
FROM Orders
WHERE ShipDate is null
```

Below the query editor, the "Results" tab is active, displaying a table with 5 rows and 3 columns: OrderID, OrderDate, and ShipDate. The data is as follows:

	OrderID	OrderDate	ShipDate
1	32	2016-05-01 01:23:23.000	NULL
2	38	2016-05-08 11:41:24.000	NULL
3	39	2016-05-08 22:22:26.000	NULL
4	40	2016-05-08 21:41:29.000	NULL
5	41	2016-05-09 07:52:55.000	NULL

The status bar at the bottom indicates: "Query executed successf... | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLL\scj (53) | MyGuitarShop | 00:00:00 | 5 rows".

7 SELECT without FROM

```
SELECT 100 AS Price
,0.07 AS TaxRate
,100 * 0.07 AS TaxAmount /*Eww, magic numbers*/
,100 + (100 * 0.07) AS Total /*Why is SQL gross?*/
```

See figure 7

Figure 7: Per 7, SELECT custom values into columns w/o a FROM clause

The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays a query window titled "SQLQuery6.sql - EC2AMAZ-3ISQLLJ\scj (53)". The query text is:

```
SELECT 100 AS Price
,0.07 AS TaxRate
,100 * 0.07 AS TaxAmount /*Eww, magic numbers*/
,100 + (100 * 0.07) AS Total /*Why is SQL gross?*/
```

The bottom pane shows the "Results" tab with a single row of data:

	Price	TaxRate	TaxAmount	Total
1	100	0.07	7.00	107.00

The status bar at the bottom indicates: "Query executed successf... | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLLJ\scj (53) | MyGuitarShop | 00:00:00 | 1 rows".