

Exercise 9 – Joins for MyGuitarShop

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

October 30, 2018

Contents

1	Select CategoryName, ProductName, ListPrice	1
2	Get Allan Sherwood’s information	3
3	Get shipping addresses	4
4	Summarize orders with aliases	5
5	Names of products with the same price	7
6	Return categories which have never been used	8
7	Union of Shipped and Not Shipped orders sorted by OrderDate	10

List of Figures

1	Per 1	2
2	Per 2	3
3	Per 3	5
4	Per 4	6
5	Per 5	8
6	Per 6	9
7	Per 7	11

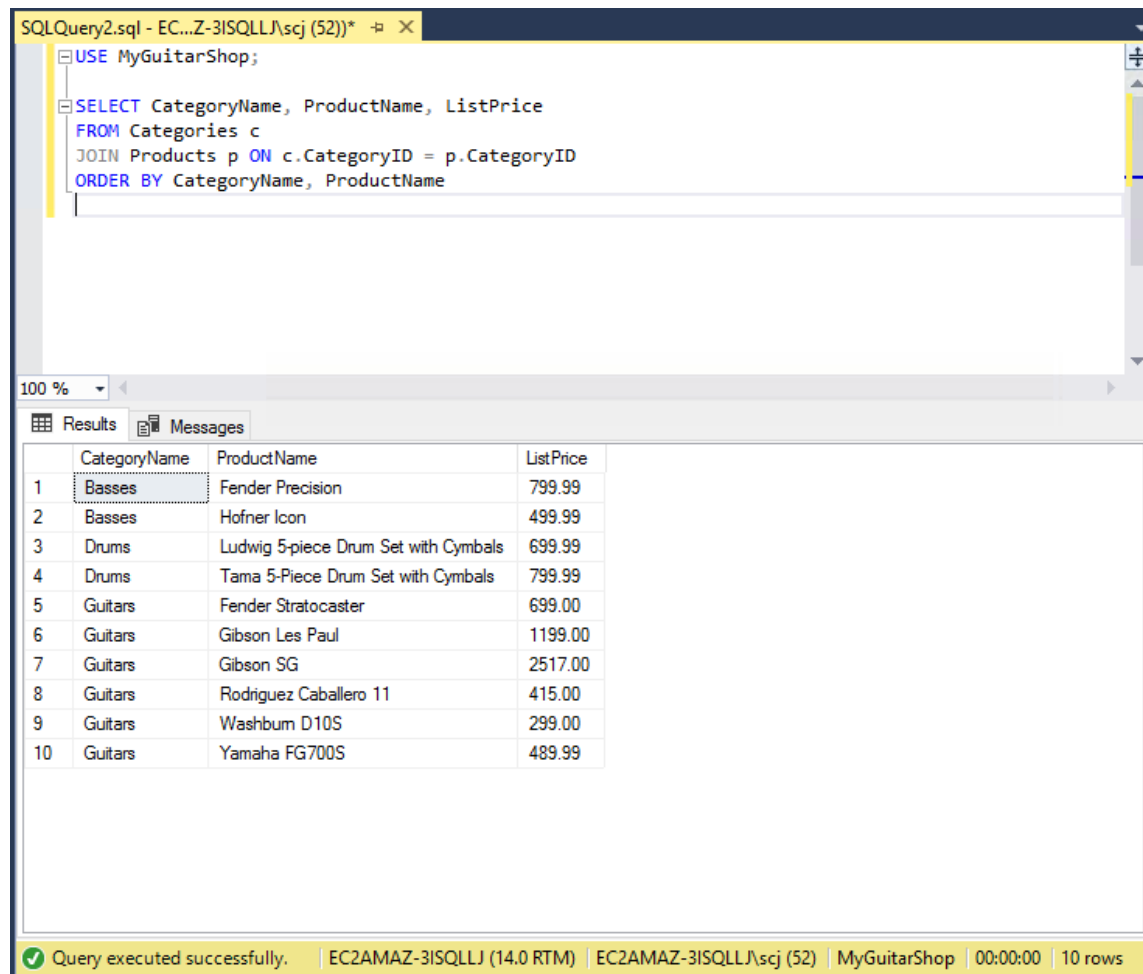
1 Select CategoryName, ProductName, ListPrice

See figure 1

```
USE MyGuitarShop;
```

```
SELECT CategoryName, ProductName, ListPrice  
FROM Categories c  
JOIN Products p ON c.CategoryID = p.CategoryID  
ORDER BY CategoryName, ProductName
```

Figure 1: Per 1



The screenshot shows a SQL Server Enterprise Manager window with a query titled 'SQLQuery2.sql - EC2AMAZ-3ISQLLJ (52)'. The query is as follows:

```
USE MyGuitarShop;  
  
SELECT CategoryName, ProductName, ListPrice  
FROM Categories c  
JOIN Products p ON c.CategoryID = p.CategoryID  
ORDER BY CategoryName, ProductName
```

The 'Results' tab is selected, displaying a table with 10 rows. The columns are 'CategoryName', 'ProductName', and 'ListPrice'. The data is as follows:

	CategoryName	ProductName	ListPrice
1	Basses	Fender Precision	799.99
2	Basses	Hofner Icon	499.99
3	Drums	Ludwig 5-piece Drum Set with Cymbals	699.99
4	Drums	Tama 5-Piece Drum Set with Cymbals	799.99
5	Guitars	Fender Stratocaster	699.00
6	Guitars	Gibson Les Paul	1199.00
7	Guitars	Gibson SG	2517.00
8	Guitars	Rodriguez Caballero 11	415.00
9	Guitars	Washburn D10S	299.00
10	Guitars	Yamaha FG700S	489.99

The status bar at the bottom indicates: 'Query executed successfully. | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLLJ (52) | MyGuitarShop | 00:00:00 | 10 rows'.

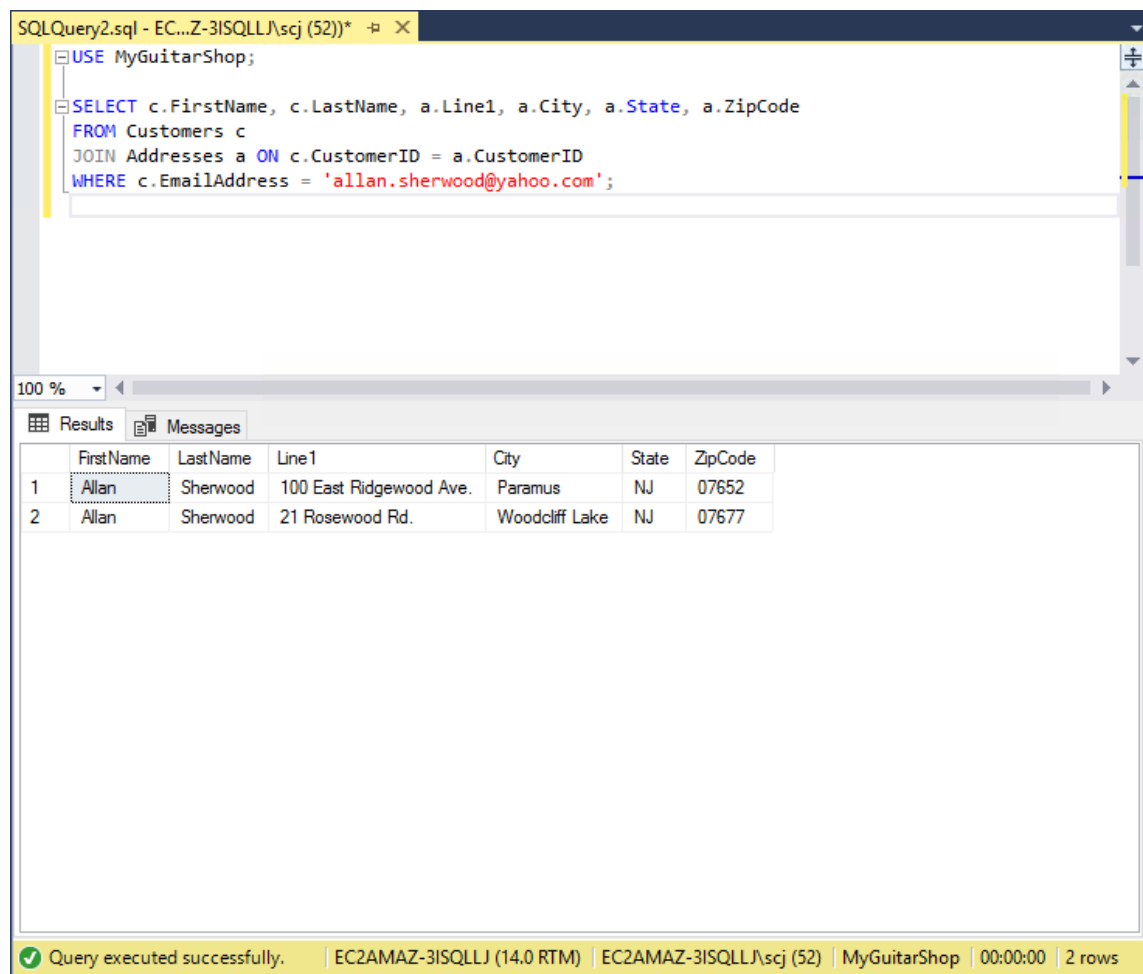
2 Get Allan Sherwood's information

See figure 2

```
USE MyGuitarShop;
```

```
SELECT c.FirstName, c.LastName, a.Line1, a.City, a.State, a.ZipCode  
FROM Customers c  
JOIN Addresses a ON c.CustomerID = a.CustomerID  
WHERE c.EmailAddress = 'allan.sherwood@yahoo.com';
```

Figure 2: Per 2



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the following SQL query:

```
USE MyGuitarShop;  
  
SELECT c.FirstName, c.LastName, a.Line1, a.City, a.State, a.ZipCode  
FROM Customers c  
JOIN Addresses a ON c.CustomerID = a.CustomerID  
WHERE c.EmailAddress = 'allan.sherwood@yahoo.com';
```

The bottom pane shows the results of the query, which returned two rows of data:

	FirstName	LastName	Line1	City	State	ZipCode
1	Allan	Sherwood	100 East Ridgewood Ave.	Paramus	NJ	07652
2	Allan	Sherwood	21 Rosewood Rd.	Woodcliff Lake	NJ	07677

The status bar at the bottom indicates: Query executed successfully. | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLLJ\scj (52) | MyGuitarShop | 00:00:00 | 2 rows

3 Get shipping addresses

See figure 3

```
USE MyGuitarShop;

SELECT c.FirstName, c.LastName, a.Line1, a.City, a.State, a.ZipCode
FROM Customers c
JOIN Addresses a ON c.CustomerID = a.CustomerID
WHERE c.ShippingAddressID = a.AddressID;
```

Figure 3: 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:

```
USE MyGuitarShop;  
  
SELECT c.FirstName, c.LastName, a.Line1, a.City, a.State, a.ZipCode  
FROM Customers c  
JOIN Addresses a ON c.CustomerID = a.CustomerID  
WHERE c.ShippingAddressID = a.AddressID;
```

The results pane displays a table with 7 columns: FirstName, LastName, Line1, City, State, and ZipCode. The table contains 16 rows of data, sorted by LastName. The first row is highlighted.

	FirstName	LastName	Line1	City	State	ZipCode
1	Allan	Sherwood	100 East Ridgewood Ave.	Paramus	NJ	07652
2	Bary	Zimmer	16285 Wendell St.	Omaha	NE	68135
3	Christine	Brown	19270 NW Cornell Rd.	Beaverton	OR	97006
4	David	Goldstein	186 Vermont St.	San Francisco	CA	94110
5	Erin	Valentino	6982 Palm Ave.	Fresno	CA	93711
6	Frank Lee	Wilson	23 Mountain View St.	Denver	CO	80208
7	Gary	Hernandez	7361 N. 41st St.	New York	NY	10012
8	Heather	Esway	2381 Buena Vista St.	Los Angeles	CA	90023
9	James	Butt	96950 Hidden Ln	Aberdeen	MD	21001
10	Josephine	Darakjy	6980 Dorsett Rd	Abilene	KS	67410
11	Art	Venere	33 State St	Abilene	TX	79601
12	Lenna	Paprocki	8284 Hart St	Abilene	KS	67410
13	Donette	Foller	9387 Charcot Ave	Absecon	NJ	8201
14	Simona	Morasca	3777 E Richmond St #900	Akron	OH	44302
15	Mitsue	Tollner	74 S Westgate St	Albany	NY	12204
16	Leta	Dillard	89992 E 15th St	Alliance	NE	68301

The status bar at the bottom indicates: Query exec... | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLLJ\scj (52) | MyGuitarShop | 00:00:00 | 483 rows

4 Summarize orders with aliases

Sorted by LastName, OrderDate, and ProductName See figure 4

```

USE MyGuitarShop;

SELECT c.LastName,
c.FirstName,
o.OrderDate,
p.ProductName,
oi.ItemPrice,
oi.DiscountAmount,
oi.Quantity
FROM Customers AS c
JOIN Orders AS o ON c.CustomerID = o.CustomerID
JOIN OrderItems AS oi ON o.OrderID = oi.OrderID
JOIN Products AS p ON oi.ProductID = p.ProductID
ORDER BY c.LastName, o.OrderDate, p.ProductName;

```

Figure 4: Per 4

The screenshot shows a SQL Server Enterprise Manager window with a query executed. The query is displayed in the top pane, and the results are shown in the bottom pane. The results table has 7 columns: LastName, FirstName, OrderDate, ProductName, ItemPrice, DiscountAmount, and Quantity. The results are sorted by LastName, then OrderDate, and then ProductName.

	LastName	FirstName	OrderDate	ProductName	ItemPrice	DiscountAmount	Quantity
1	Albares	Cammy	2016-04-20 08:14:45.000	Rodriguez Caballero 11	699.00	209.70	1
2	Amigon	Minna	2016-04-11 08:21:32.000	Gibson SG	799.99	240.00	1
3	Brown	Christine	2016-03-30 15:22:31.000	Gibson Les Paul	1199.00	359.70	1
4	Butt	James	2016-04-04 06:24:44.000	Rodriguez Caballero 11	699.00	209.70	1
5	Caldarera	Kiley	2016-04-17 17:40:22.000	Rodriguez Caballero 11	699.00	209.70	1
6	Caudy	Chanel	2016-05-09 07:52:55.000	Hofner Icon	489.99	186.20	1
7	Caudy	Chanel	2016-05-09 07:52:55.000	Rodriguez Caballero 11	699.00	209.70	1
8	Darakjy	Josephine	2016-04-04 08:15:12.000	Fender Stratocaster	2517.00	1308.84	1
9	Dilliard	Leota	2016-04-06 18:41:53.000	Rodriguez Caballero 11	699.00	209.70	1
10	Esway	Heather	2016-04-03 14:59:20.000	Fender Precision	499.99	125.00	1
11	Esway	Heather	2016-04-12 12:26:52.000	Fender Stratocaster	2517.00	1308.84	1
12	Flosi	Fletcher	2016-05-01 09:11:51.000	Tama 5-Piece Drum Set with Cymbals	299.00	0.00	1
13	Foller	Donette	2016-04-05 14:52:17.000	Gibson SG	799.99	240.00	1
14	Garufi	Meaghan	2016-04-21 17:52:24.000	Washburn D10S	699.99	210.00	1
15	Goldstein	David	2016-03-31 05:43:11.000	Tama 5-Piece Drum Set with Cymbals	299.00	0.00	1

Query execut... | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLLJ\scj (52) | MyGuitarShop | 00:00:00 | 47 rows

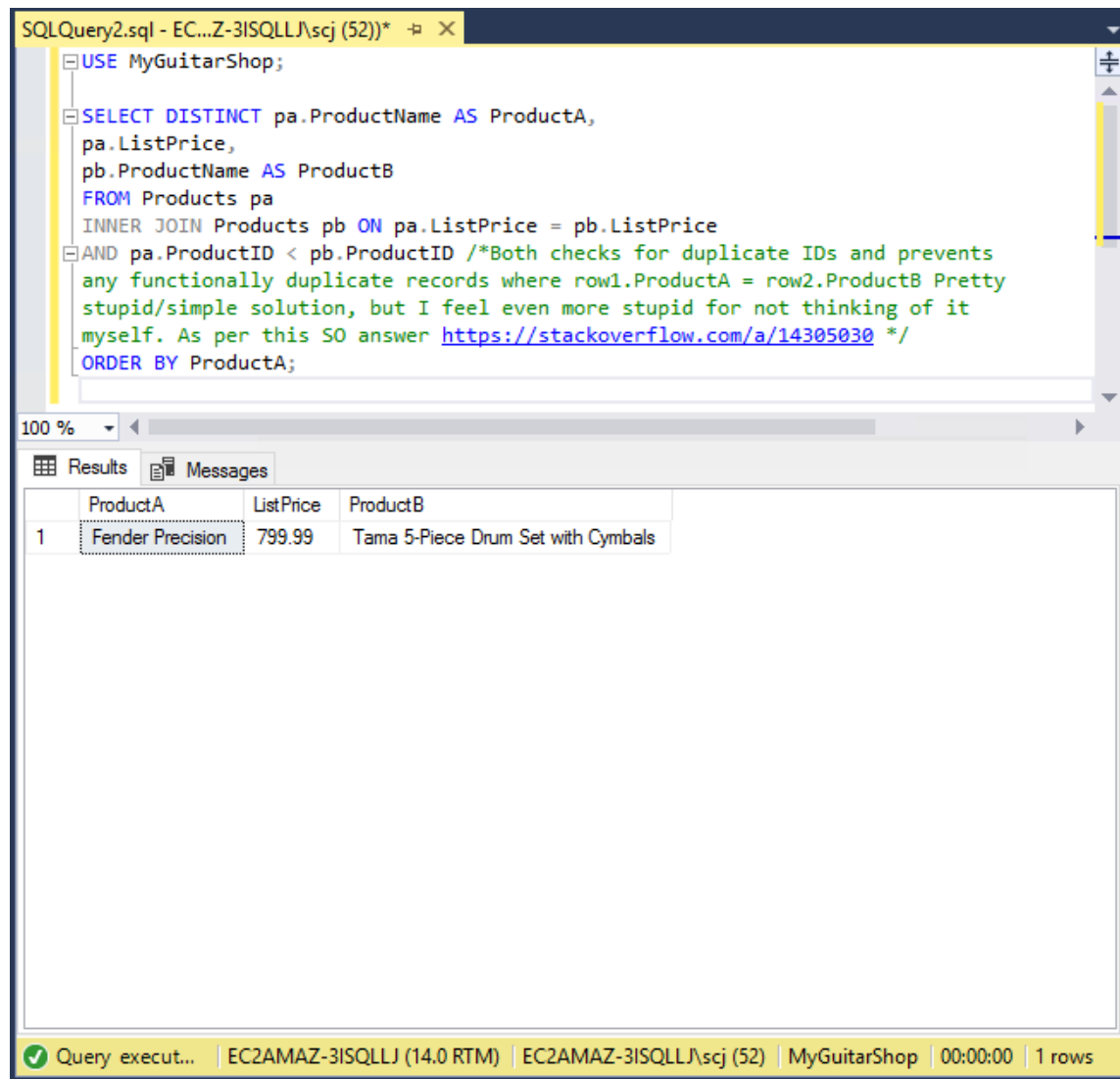
5 Names of products with the same price

See figure 5

```
USE MyGuitarShop;
```

```
SELECT DISTINCT pa.ProductName AS ProductA,  
pa.ListPrice,  
pb.ProductName AS ProductB  
FROM Products pa  
INNER JOIN Products pb ON pa.ListPrice = pb.ListPrice  
AND pa.ProductID < pb.ProductID /*Both checks for duplicate IDs and prevents  
any functionally duplicate records where row1.ProductA = row2.ProductB Pretty  
stupid/simple solution, but I feel even more stupid for not thinking of it  
myself. As per this SO answer https://stackoverflow.com/a/14305030 */  
ORDER BY ProductA;
```

Figure 5: Per 5



The screenshot shows a SQL Server Enterprise Manager window with a query titled "SQLQuery2.sql - EC...Z-3ISQLLJ\scj (52)". The query is as follows:

```
USE MyGuitarShop;  
  
SELECT DISTINCT pa.ProductName AS ProductA,  
pa.ListPrice,  
pb.ProductName AS ProductB  
FROM Products pa  
INNER JOIN Products pb ON pa.ListPrice = pb.ListPrice  
AND pa.ProductID < pb.ProductID /*Both checks for duplicate IDs and prevents  
any functionally duplicate records where row1.ProductA = row2.ProductB Pretty  
stupid/simple solution, but I feel even more stupid for not thinking of it  
myself. As per this SO answer https://stackoverflow.com/a/14305030 */  
ORDER BY ProductA;
```

The query results are displayed in a table with the following data:

	ProductA	ListPrice	ProductB
1	Fender Precision	799.99	Tama 5-Piece Drum Set with Cymbals

The status bar at the bottom indicates: "Query execut... | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLLJ\scj (52) | MyGuitarShop | 00:00:00 | 1 rows".

6 Return categories which have never been used

See figure 6


```

USE MyGuitarShop;

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

```

Figure 6: Per 6

The screenshot shows a SQL Server Enterprise Manager window with a query titled 'SQLQuery2.sql - EC...Z-3ISQLL\scj (52)'. The query is as follows:

```

USE MyGuitarShop;

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

```

Below the query editor, the 'Results' tab is active, displaying a table with the following data:

	CategoryName	ProductID
1	Keyboards	NULL

The status bar at the bottom indicates: 'Query execut... | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLL\scj (52) | MyGuitarShop | 00:00:00 | 1 rows'.

7 Union of Shipped and Not Shipped orders sorted by OrderDate

See figure 7

```
USE MyGuitarShop;

SELECT 'NOT SHIPPED' AS ShipStatus,
       OrderID,
       OrderDate
FROM Orders
WHERE ShipDate IS NULL
UNION
SELECT 'SHIPPED' AS ShipStatus,
       OrderID,
       OrderDate
FROM Orders
WHERE ShipDate IS NOT NULL
ORDER BY OrderDate;
```

Figure 7: Per 7

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a SQL query in the 'SQLQuery2.sql' file. The query is a UNION of two SELECT statements, both from the 'Orders' table. The first SELECT statement filters for 'NOT SHIPPED' orders, and the second filters for 'SHIPPED' orders. The results are ordered by 'OrderDate'. The bottom pane shows the 'Results' tab with a table containing 41 rows. The table has four columns: 'ShipStatus', 'OrderID', and 'OrderDate'. The 'ShipStatus' column contains values 'SHIPPED' and 'NOT SHIPPED'. The 'OrderID' column contains values from 27 to 41. The 'OrderDate' column contains timestamps from 2016-04-20 to 2016-05-09. The status bar at the bottom indicates that the query was executed successfully, returning 41 rows in 00:00:00 seconds.

```
SELECT 'NOT SHIPPED' AS ShipStatus,
OrderID,
OrderDate
FROM Orders
WHERE ShipDate IS NULL
UNION
SELECT 'SHIPPED' AS ShipStatus,
OrderID,
OrderDate
FROM Orders
WHERE ShipDate IS NOT NULL
ORDER BY OrderDate;
```

	ShipStatus	OrderID	OrderDate
27	SHIPPED	27	2016-04-20 09:17:52.000
28	SHIPPED	28	2016-04-21 17:52:24.000
29	SHIPPED	29	2016-04-25 23:36:41.000
30	SHIPPED	30	2016-04-27 16:21:31.000
31	SHIPPED	31	2016-04-29 06:47:14.000
32	NOT SHI...	32	2016-05-01 01:23:23.000
33	SHIPPED	33	2016-05-01 09:11:51.000
34	SHIPPED	34	2016-05-02 11:36:12.000
35	SHIPPED	35	2016-05-04 03:52:23.000
36	SHIPPED	36	2016-05-04 12:31:33.000
37	SHIPPED	37	2016-05-06 14:15:21.000
38	NOT SHI...	38	2016-05-08 11:41:24.000
39	NOT SHI...	40	2016-05-08 21:41:29.000
40	NOT SHI...	39	2016-05-08 22:22:26.000
41	NOT SHI...	41	2016-05-09 07:52:55.000

Query execut... | EC2AMAZ-3ISQLLJ (14.0 RTM) | EC2AMAZ-3ISQLLJ\scj (52) | MyGuitarShop | 00:00:00 | 41 rows