POWERSHELL RANGERS | 23 April 2019 | ISM 4212 | Dr. Hyman

**Motto:**

It’s Database Time!

**Members:**

Brennan Harrison (Lead)

Huong Le

Brian Pueschel

Mike Sangrene

Smallville Database Design Document

**Version History**

|  |  |
| --- | --- |
| **Version** | **Description** |
| 1.0 | First released draft |
| 2.0 | Summary of changes:   1. Additional Requirements, Entities, Relations. 2. Updated ERD 3. Updated EERD 4. Smallville Relational Schema 5. Smallville Data Dictionary Header 6. Smallville Data Dictionary |
| 3.0 | Summary of changes:   1. Updated ERD 2. Changed name of Category attribute to Type. 3. Corrected optionality connecting Employee to Customers. 4. Updated EERD 5. Corrected optionality connecting Employee to Sales-Rep Assignment. 6. Updated Smallville Data Dictionary    1. Dropped all check constraints that were not on State columns or Type column.    2. Converted all Primary Key and Foreign Key data types to int.    3. Converted all Phone Number column data types to varchar (20).    4. Converted all Zip column data types to varchar (10).    5. Converted all Quantity column data types to int.    6. Converted CommissionRate column data type to varchar (10).    7. Converted HourlyRate column data type to decimal (4,2).    8. Converted Compensation data type to decimal (8,2).    9. Converted AnnualCompensation column data type to varchar (20).    10. Modified all Foreign Key columns to allow nulls.    11. Modified all Phone Number columns to allow nulls.    12. Modified Description column to allow nulls.    13. Modified all Price columns to allow nulls.    14. Modified all Quantity columns to allow nulls.    15. Modified all Date columns to allow nulls.    16. Modified all Email columns to allow nulls.    17. Modified Commission Rate column to allow nulls.    18. Modified AnnualCompensation column to allow nulls.    19. Removed the identity seed and increment for all Primary Keys in associate tables. 7. Table Views 8. SPROCS |
| 4.0 | Summary of changes:   1. Requirements    1. Updated to be more readable 2. Entities    1. Updated to be more readable 3. EERD    1. Updated with new columns for Hourly table 4. Relational Schema    1. Updated with new columns for Hourly table 5. Data Dictionary Summary Header    1. Updated with new columns for Hourly table 6. Data Dictionary    1. Updated with new columns for Hourly table 7. Table View Reports 8. User-Acceptance Test Queries |

**Purpose**

The purpose of this document is to list the major objects of interest to Smallville, the characteristics of those objects and how the objects logically relate to each other using a conceptual data model.

**Narrative**

Smallville’s employees are either sales reps, salary or hourly workers. Smallville’s sales reps are assigned customers. The date and description of each sales rep assignment to each customer is documented. Customers place orders for products they wish to purchase. Each order is placed by one customer. Orders request products that a customer desires. Each product may or may not be tied to any orders. The quantity ordered, quantity supplied, and order price of each product tied to each order is documented. Smallville’s products are offered by suppliers. The number of products and date they were ordered from each supplier is documented.

**Requirements (Actors and Roles)**

EMPLOYEE – An employee may be assigned customers.

CUSTOMER – A customer places orders.

ORDER – An order requests products.

PRODUCT – A product is offered by suppliers.

**Entities Identified**

EMPLOYEE

CUSTOMER

ORDER

PRODUCT

SUPPLIER

**Entities with Nested Attributes**

CUSTOMER

Customer ID

First Name

Last Name

Street

City

State

Zip

Phone Number

EMPLOYEE

Employee ID

First Name

Last Name

Street

City

State

Zip

Email

Phone Number

Type

SUPPLIER

Supplier ID

Company Name

Street

City

State

Zip

Phone Number

ORDER

Order ID

Order Date

Status

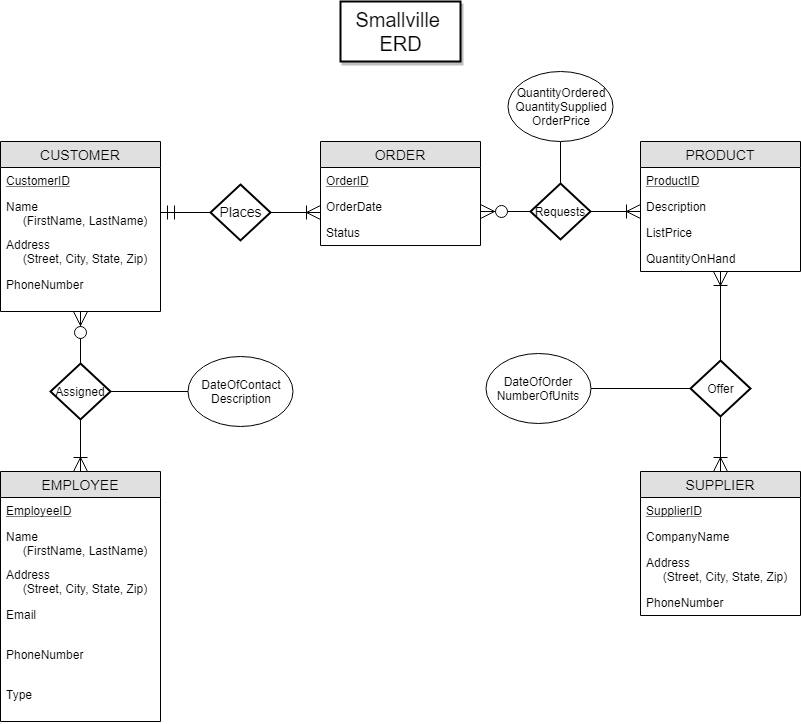
PRODUCT

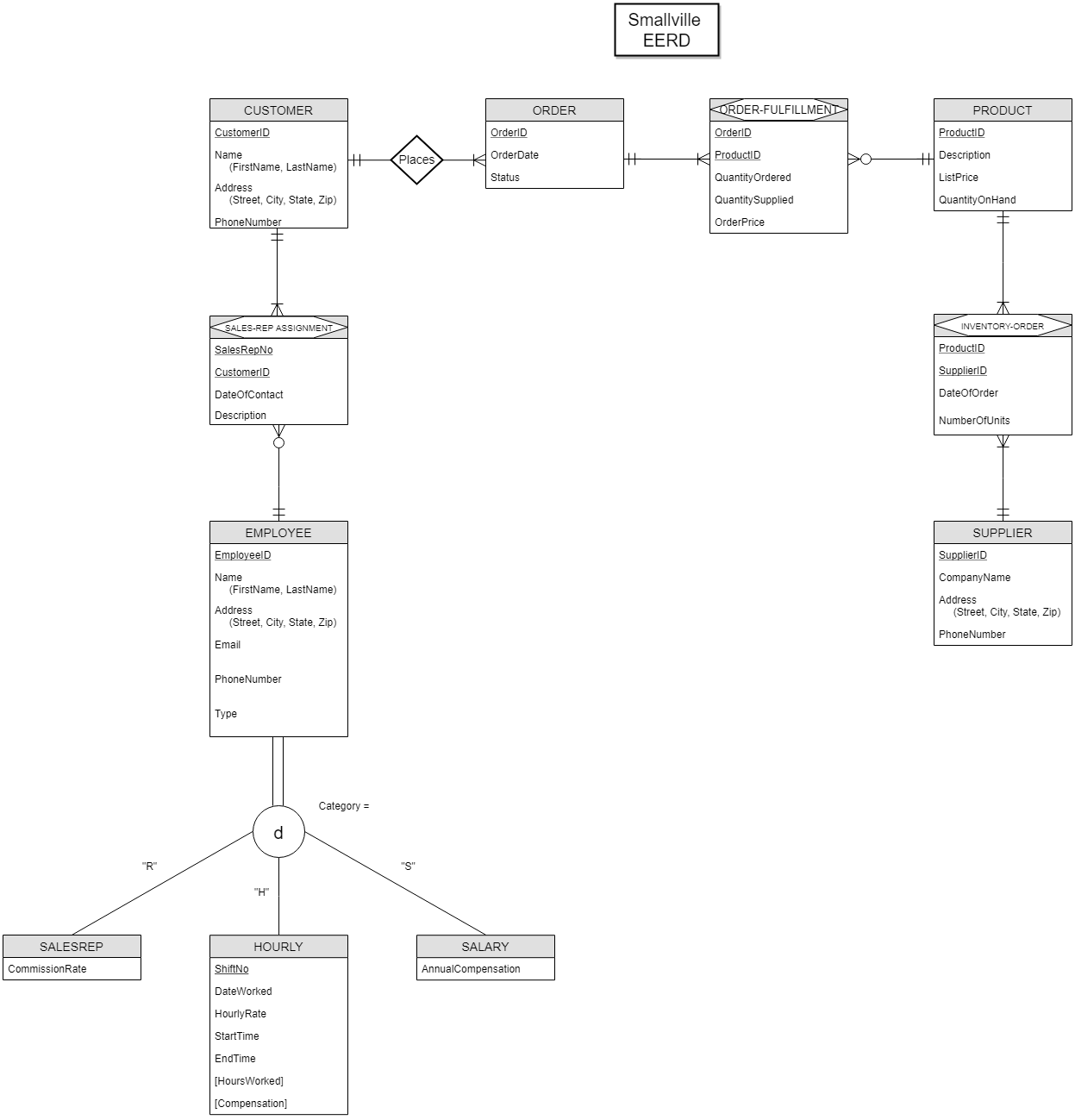
Product ID

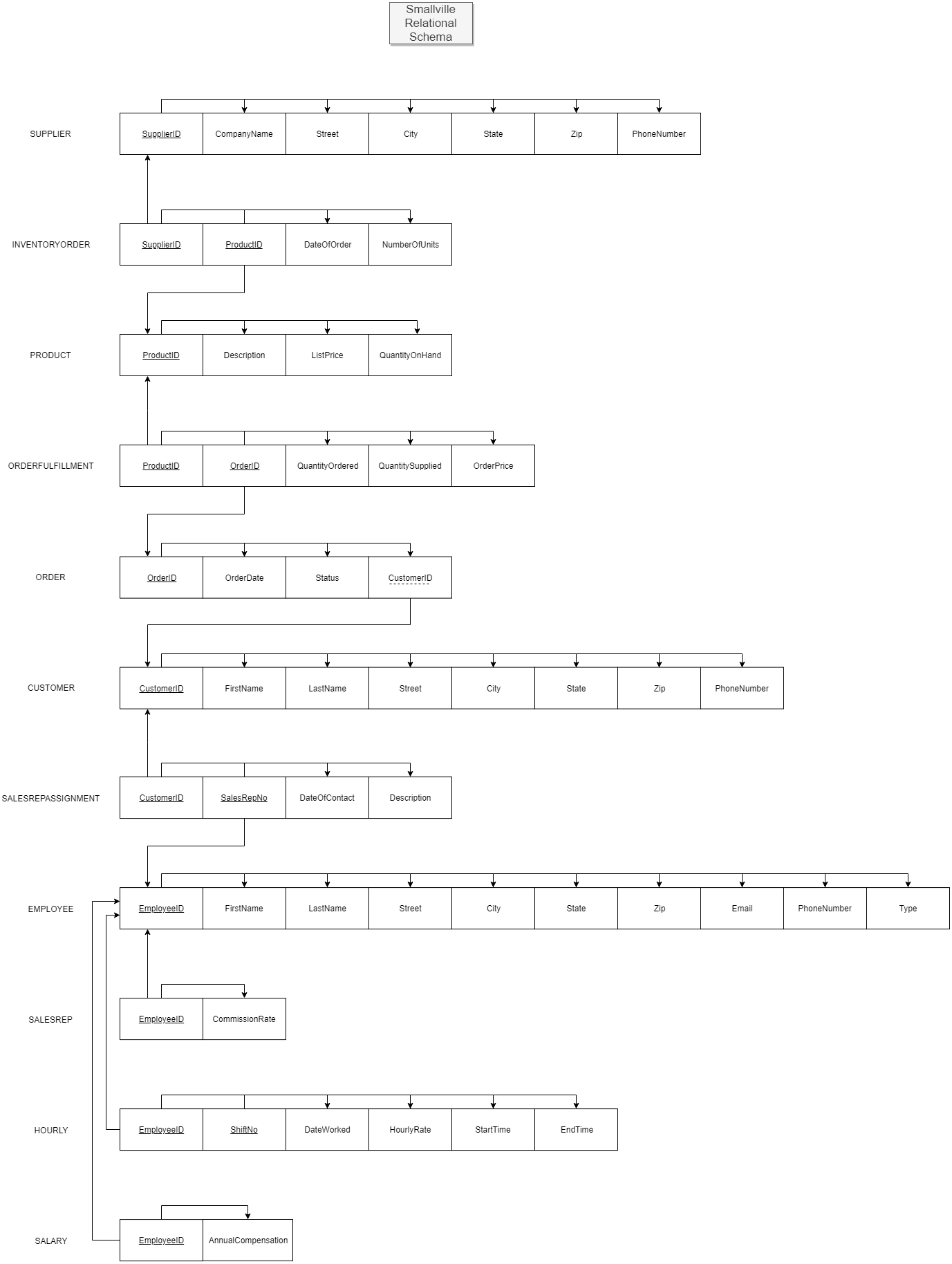
Description

List Price

Quantity On Hand







**Data Dictionary Summary Header**

**Smallville Database Tables and Attributes**

Supplier (SupplierID, CompanyName, Street, City, State Zip, PhoneNumber)

InventoryOrder (SupplierID, ProductID, DateOfOrder, NumberOfUnits)

Product (ProductID, Description, ListPrice, QuantityOnHand)

OrderFulfillment (ProductID, OrderID, QuantityOrdered, QuantitySupplied, OrderPrice)

Order (OrderID, OrderDate, Satus, CustomerID)

Customer (CustomerID, FirstName, LastName, Street, City, State, Zip, PhoneNumber)

SalesRepAssigment (CustomerID, SalesRepNo, DateOfContact, Description)

Employee (EmployeID, FirstName, LastName, Street, City, State, Zip, Email, PhoneNumber, Type)

SalesRep (EmployeeID, CommisionRate)

Hourly (EmployeeID, ShiftNo, DateWorked, HourlyRate, StartTime, EndTime, HoursWorked, Compensation)

Salary (EmployeeID, AnnualCompensation)

SMALLVILLE – DATA DICTIONARY

**(Microsoft SQL Server Notation)**

Table: **Supplier**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| SupplierID | **PK**; Unique sequential supplier ID number | int |  | Y |  |  |  |  | Y |
| CompanyName | Name of the company | varchar | 30 |  |  |  |  |  |  |
| Street | Street of the company | varchar | 20 |  |  |  |  |  |  |
| City | City of the company | varchar | 15 |  |  |  |  |  |  |
| State | State of the company | char | 2 |  |  |  | LIKE ‘[A-Z][A-Z]’ |  |  |
| Zip | Zip code of the company | varchar | 10 |  |  |  |  |  |  |
| PhoneNumber | Phone number of the company | varchar | 20 |  |  |  |  | Y |  |

Table: **InventoryOrder**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| SupplierID | **CPK;** **FK** to Supplier table | int |  |  |  |  |  |  | Y |
| ProductID | **CPK; FK** to Product table | int |  |  |  |  |  |  | Y |
| DateOfOrder | Date of product order to supplier | date |  |  |  |  |  | Y |  |
| NumberOfUnits | Number of product shipped from supplier | int |  |  |  |  |  | Y |  |

Table: **Product**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| ProductID | **PK**; Unique sequential product ID number | int |  | Y |  |  |  |  | Y |
| Description | Description of product | varchar | 75 |  |  |  |  | Y |  |
| ListPrice | List price of product | decimal | (5,2) |  |  |  |  | Y |  |
| QuantityOnHand | Number of product in stock | smallint |  |  |  |  |  | Y |  |

Table: **OrderFulfillment**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| ProductID | **CPK**; FK to Product table | int |  |  |  |  |  |  | Y |
| OrderID | **CPK**; **FK** to Order table | int |  |  |  |  |  |  | Y |
| QuantityOrdered | Quantity of product ordered | int |  |  |  |  |  | Y |  |
| QuantitySupplied | Quantity of product supplied | int |  |  |  |  |  | Y |  |
| OrderPrice | Order price of product | varchar | 20 |  |  |  |  | Y |  |

Table: **Order**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| OrderID | **PK**; Unique sequential order ID number | int |  | Y |  |  |  |  | Y |
| OrderDate | Date of customer order | date |  |  |  |  |  | Y |  |
| Status | Status of customer order | varchar | 15 |  |  |  |  | Y |  |
| CustomerID | **FK** to Customer table; customer that placed order | int |  |  |  |  |  | Y |  |

Table: **Customer**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| CustomerID | **PK**; Unique sequential customer ID number | int |  | Y |  |  |  |  | Y |
| FirstName | Customer first name | varchar | 15 |  |  |  |  |  |  |
| LastName | Customer last name | varchar | 20 |  |  |  |  |  |  |
| Street | Street of the customer | varchar | 20 |  |  |  |  |  |  |
| City | City of the customer | varchar | 15 |  |  |  |  |  |  |
| State | State of the customer | char | 2 |  |  |  | LIKE ‘[A-Z][A-Z]’ |  |  |
| Zip | Zip code of the customer | varchar | 10 |  |  |  |  |  |  |
| PhoneNumber | Phone number of the customer | varchar | 20 |  |  |  |  | Y |  |

Table: **SalesRepAssignment**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| CustomerID | **CPK; FK** to Customer table | int |  |  |  |  |  |  | Y |
| SalesRepNo | **CPK; FK** to Employee table; synonym for EmployeeID; customer sales representative | int |  |  |  |  |  |  | Y |
| DateOfContact | Date of customer contact | date |  |  |  |  |  | Y |  |
| Description | Description of sales rep contact | varchar | 100 |  |  |  |  | Y |  |

Table: **Employee**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| EmployeeID | **PK**; Unique sequential employee ID number | int |  | Y |  |  |  |  | Y |
| FirstName | Employee first name | varchar | 15 |  |  |  |  |  |  |
| LastName | Employee last name | varchar | 20 |  |  |  |  |  |  |
| Street | Street of the employee | varchar | 20 |  |  |  |  |  |  |
| City | City of the employee | varchar | 15 |  |  |  |  |  |  |
| State | State of the employee | char | 2 |  |  |  | LIKE ‘[A-Z][A-Z]’ |  |  |
| Zip | Zip code of the employee | varchar | 10 |  |  |  |  |  |  |
| Email | Email of the employee | varchar | 40 |  |  |  |  | Y |  |
| PhoneNumber | Phone number of the employee | varchar | 20 |  |  |  |  | Y |  |
| Type | Type of the employee | char | 1 |  |  |  | ([Type] = (‘R’)  OR  [Type] = (‘H’) OR [Type] = (‘S’)) |  |  |

Table: **SalesRep**

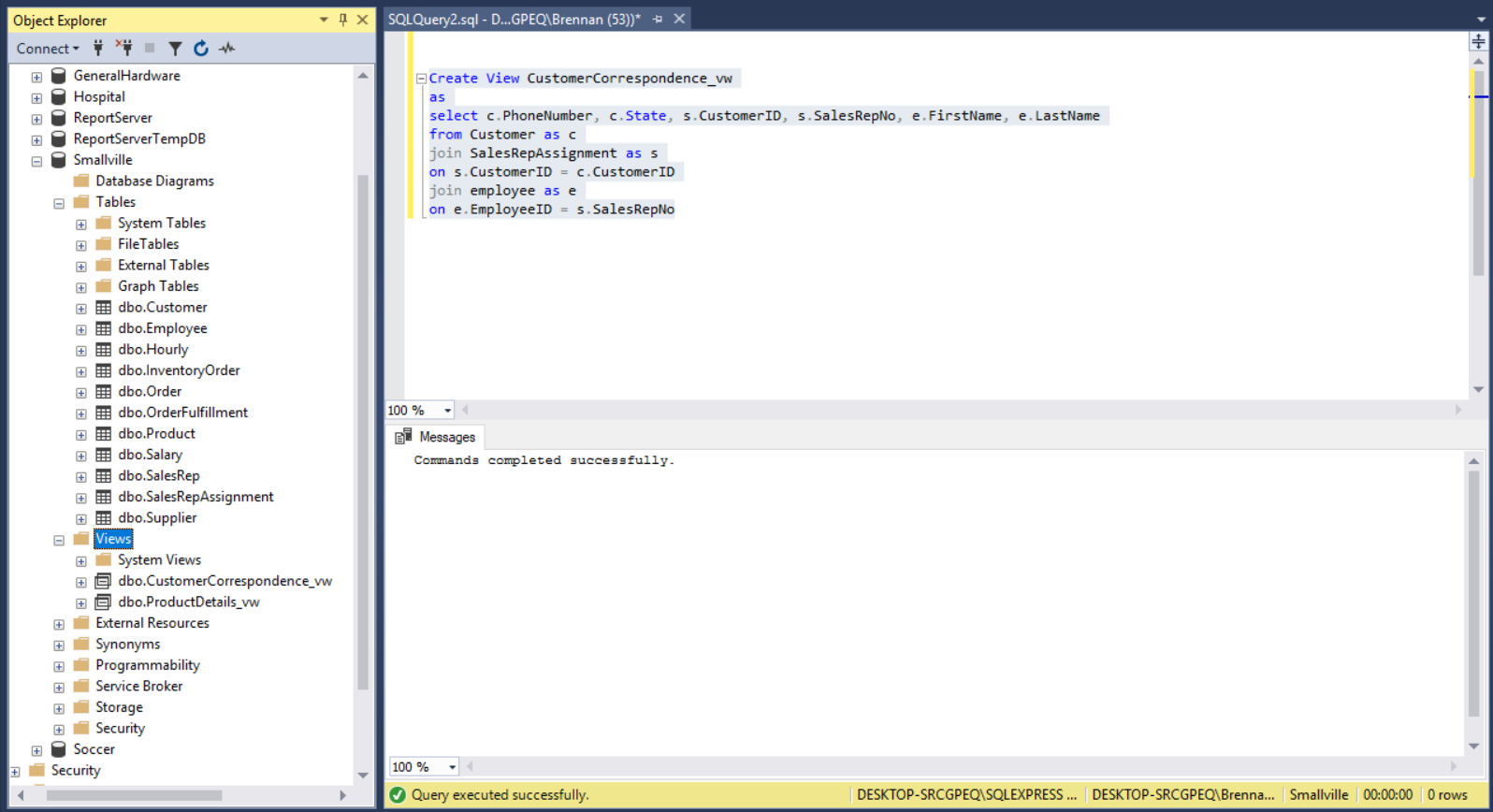
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| EmployeeID | **PK**; Unique sequential employee ID number; **FK** to Employee table | int |  |  |  |  |  |  | Y |
| CommisionRate | Commission earned by sales representative | varchar | 10 |  |  |  |  | Y |  |

Table: **Hourly**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| EmployeeID | **CPK**; Unique sequential employee ID number; **FK** to Employee table | int |  |  |  |  |  |  | Y |
| ShiftNo | **CPK**; Unique sequential Shift number | int |  |  |  |  |  |  | Y |
| DateWorked | Date worked by hourly employee | date |  |  |  |  |  | Y |  |
| HourlyRate | Rate of hourly employee | decimal | (4,2) |  |  |  |  | Y |  |
| StartTime | Start Time of employee | time | 0 |  |  |  |  | Y |  |
| EndTime | End Time of employee | time | 0 |  |  |  |  | Y |  |
| HoursWorked | Hours worked by hourly employee | decimal | (5,2) |  |  |  |  | Y |  |
| Compensation | Compensation earned by hourly employee | decimal | (22,9) |  |  |  |  | Y |  |

Table: **Salary**

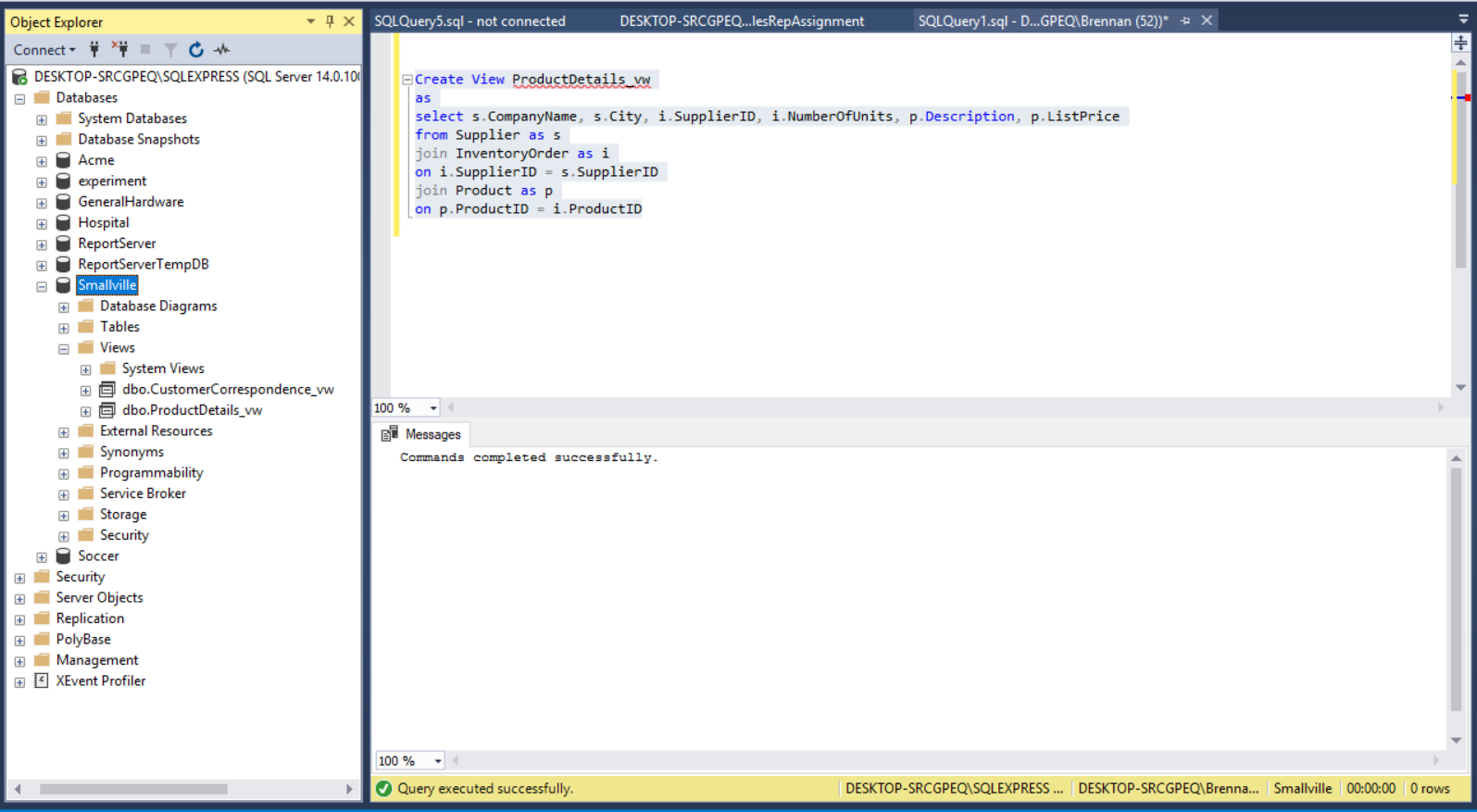
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Description** | **Data Type** | **Size** | **Identity** | **Unique** | **Default** | **Check** | **Allow Nulls** | **Index** |
| EmployeeID | **PK**; Unique sequential employee ID number; **FK** to Employee table | int |  |  |  |  |  |  | Y |
| AnnualCompensation | Annual compensation earned by salary employee | varchar | 20 |  |  |  |  | Y |  |

**Customer Correspondence Table View**

**Product Details Table View**

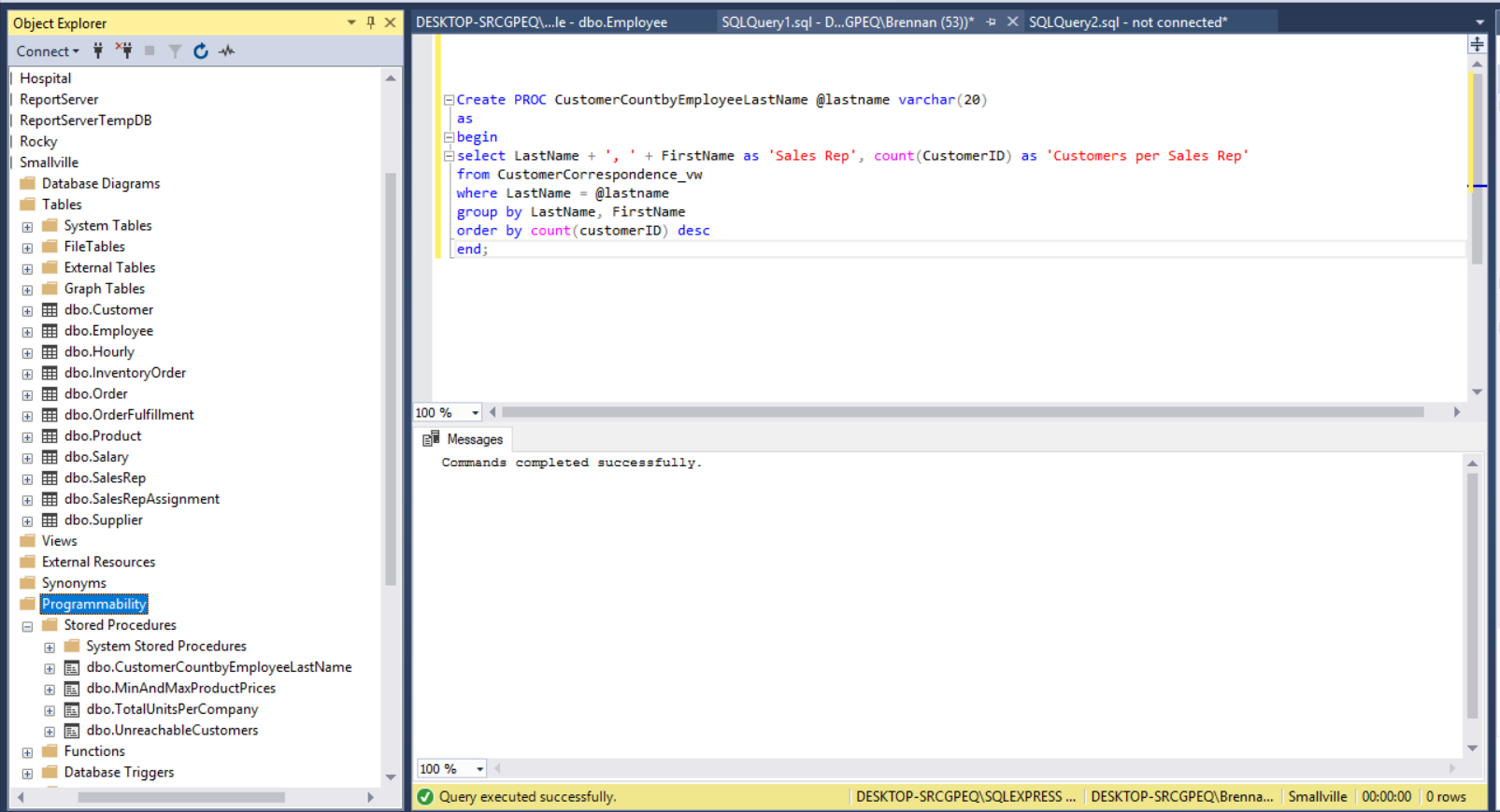
**Customer Correspondence Table View Description**

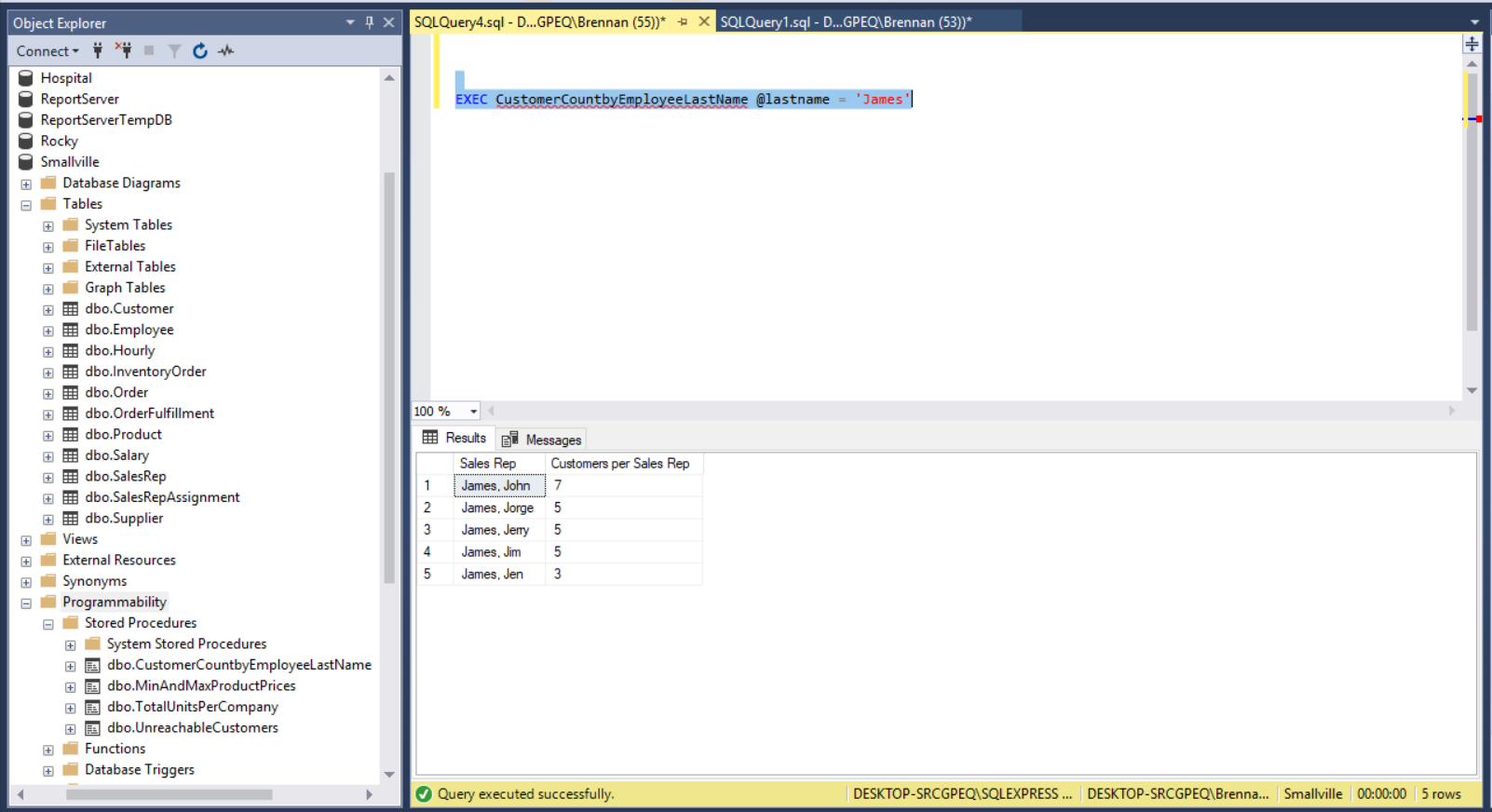
The Customer Correspondence table view joins the Customer, Sales Rep Assignment, and Employee tables on their primary and foreign key connections. Each table is given an alias for reader comprehension. The columns that are matched in this table view are Phone Number, State, Customer ID, Sales Rep No, First Name, and Last Name. The PowerShellRangers chose these columns to answer questions such as how many customers are serviced by specific sales reps and how many from each state do not have a phone number.

**Product Details Table View**

**Product Details Table View Description**

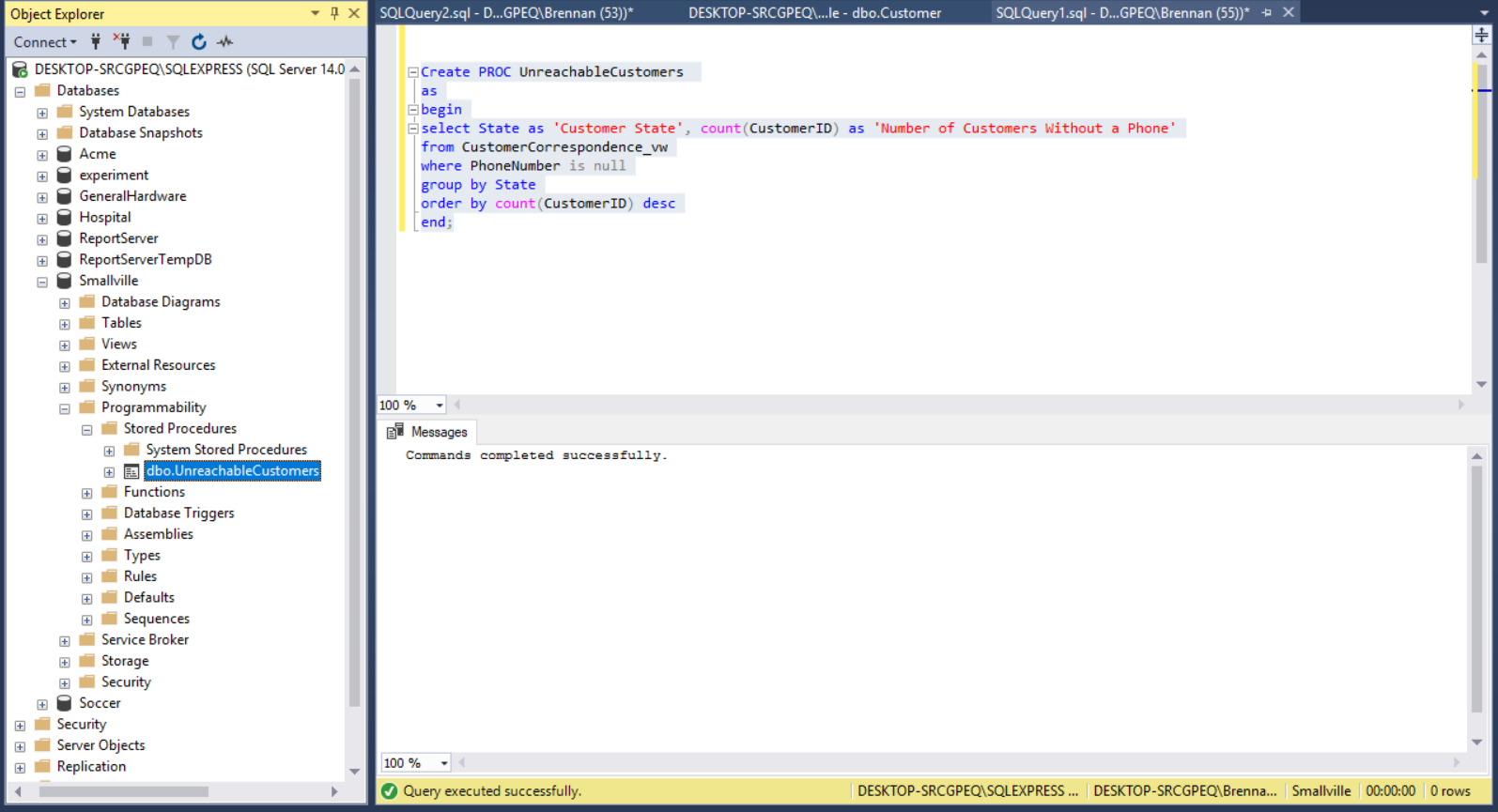
The Product Details table view joins the Supplier, Product and Inventory Order tables on their primary and foreign key connections. Each table is given an alias for reader comprehension. The columns that are matched in this table view are Company Name, City, Supplier ID, Number of Units, Description, and List Price. The PowerShellRangers chose these columns to answer questions such as what products have the highest and lowest prices, and how many products are provided by specific suppliers.

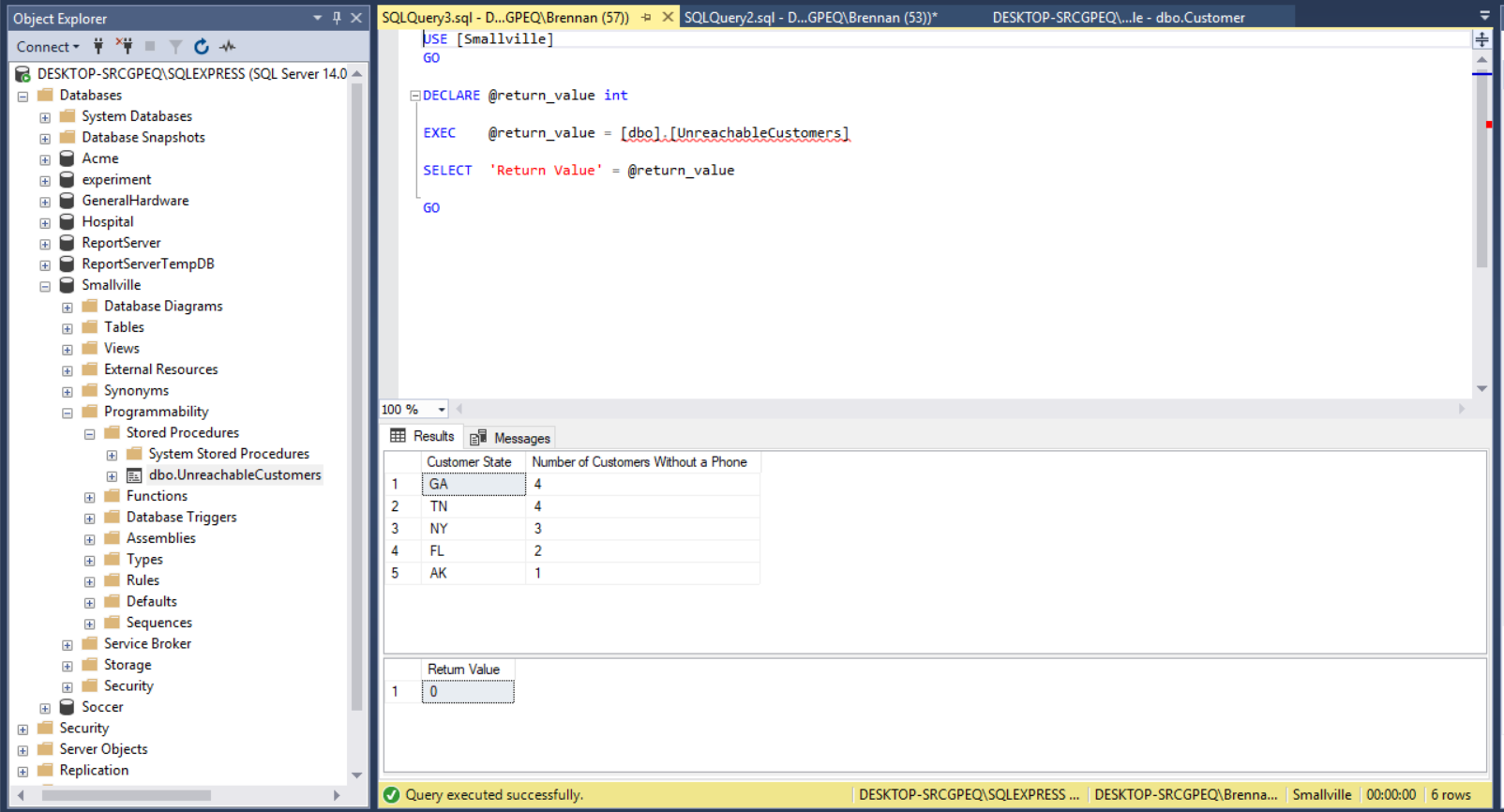
**Customer Count by Employee Last Name SPROC**



**Customer Count by Employee Last Name SPROC Description**

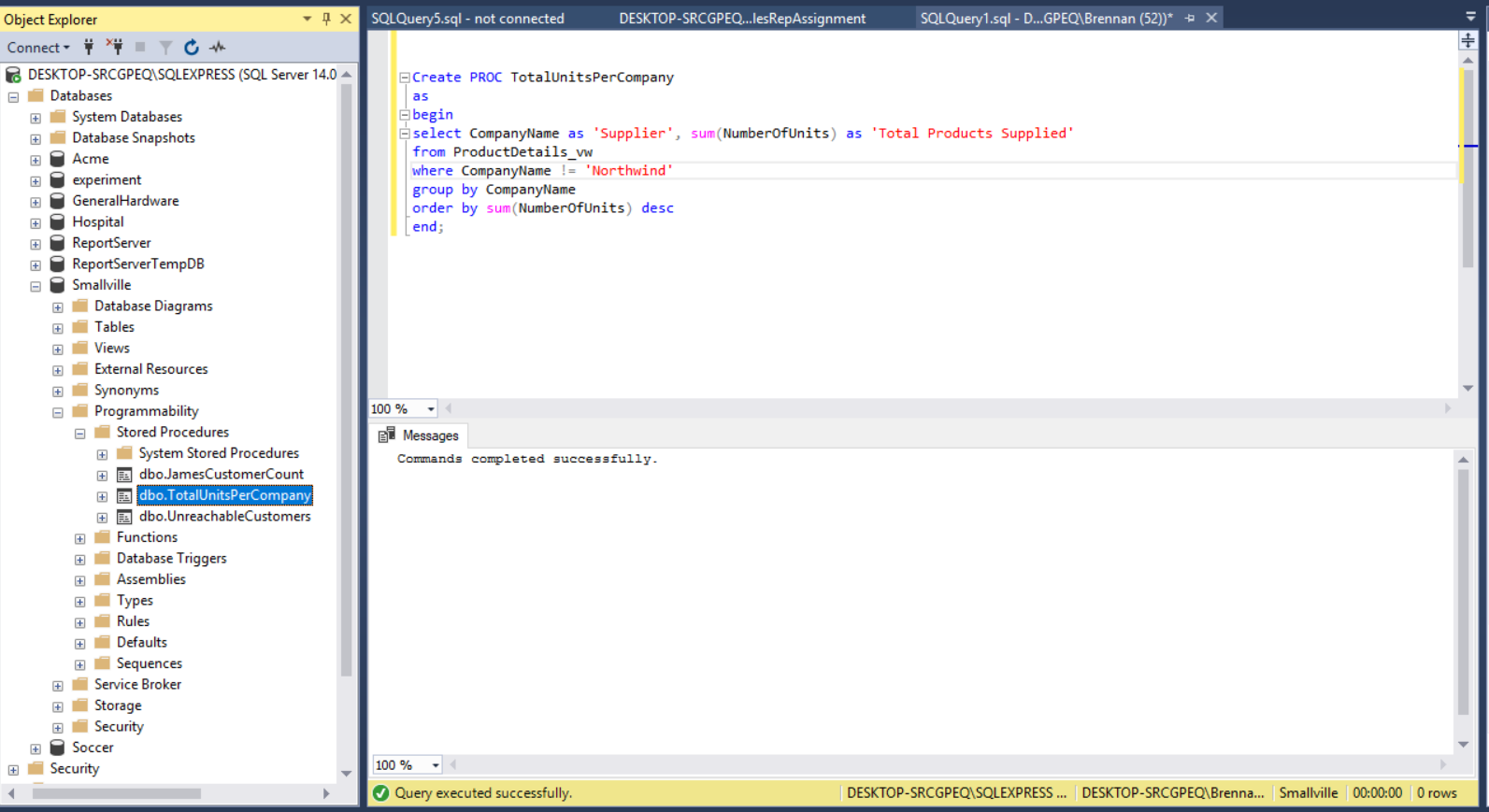
The SPROC named CustomerCountbyEmployeeLastName was created using the Customer Correspondence table view and is searching for the number of customers assigned to sales reps who share the same last name. Each column uses a header for their output. The where clause sets the Last Name field to the @lastname parameter for searching on a specific value. The count aggregator was used, and as a result, the group by clause was used to group the Last Name and First Name fields. Finally, the results were sorted by the count of Customer ID in descending order.

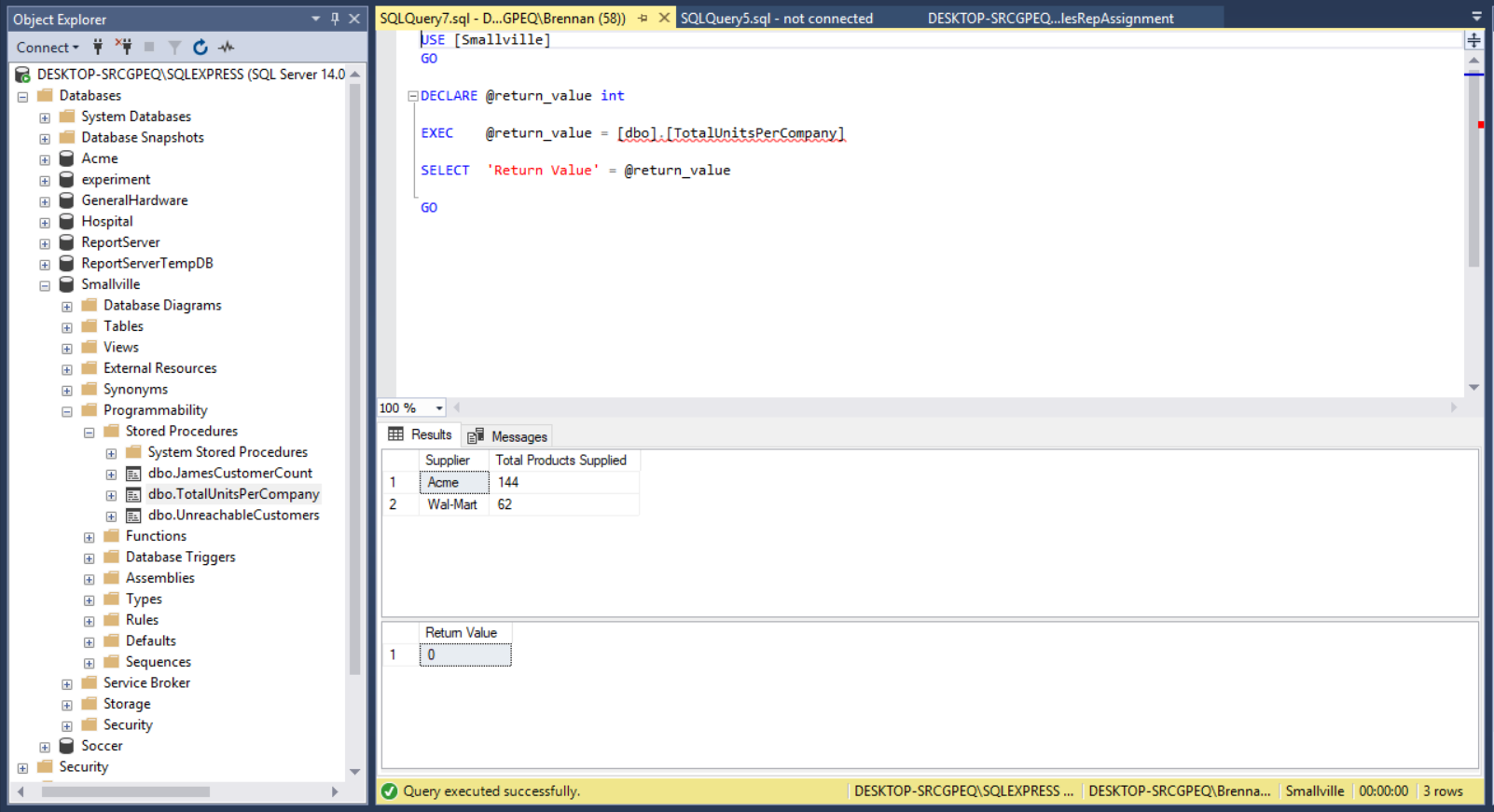
**Unreachable Customers SPROC**



**Unreachable Customers SPROC Description**

The SPROC named UnreachableCustomers was created using the Customer Correspondence table view and is searching for the number of customers in each state that do not have a phone number. Each column uses a header for their output. The where clause forces the SPROC to search only for records where phone number is null. Since an aggregator was used, all returned values were grouped by their recorded state. Finally, the results were sorted by the count of the Customer ID in descending order.

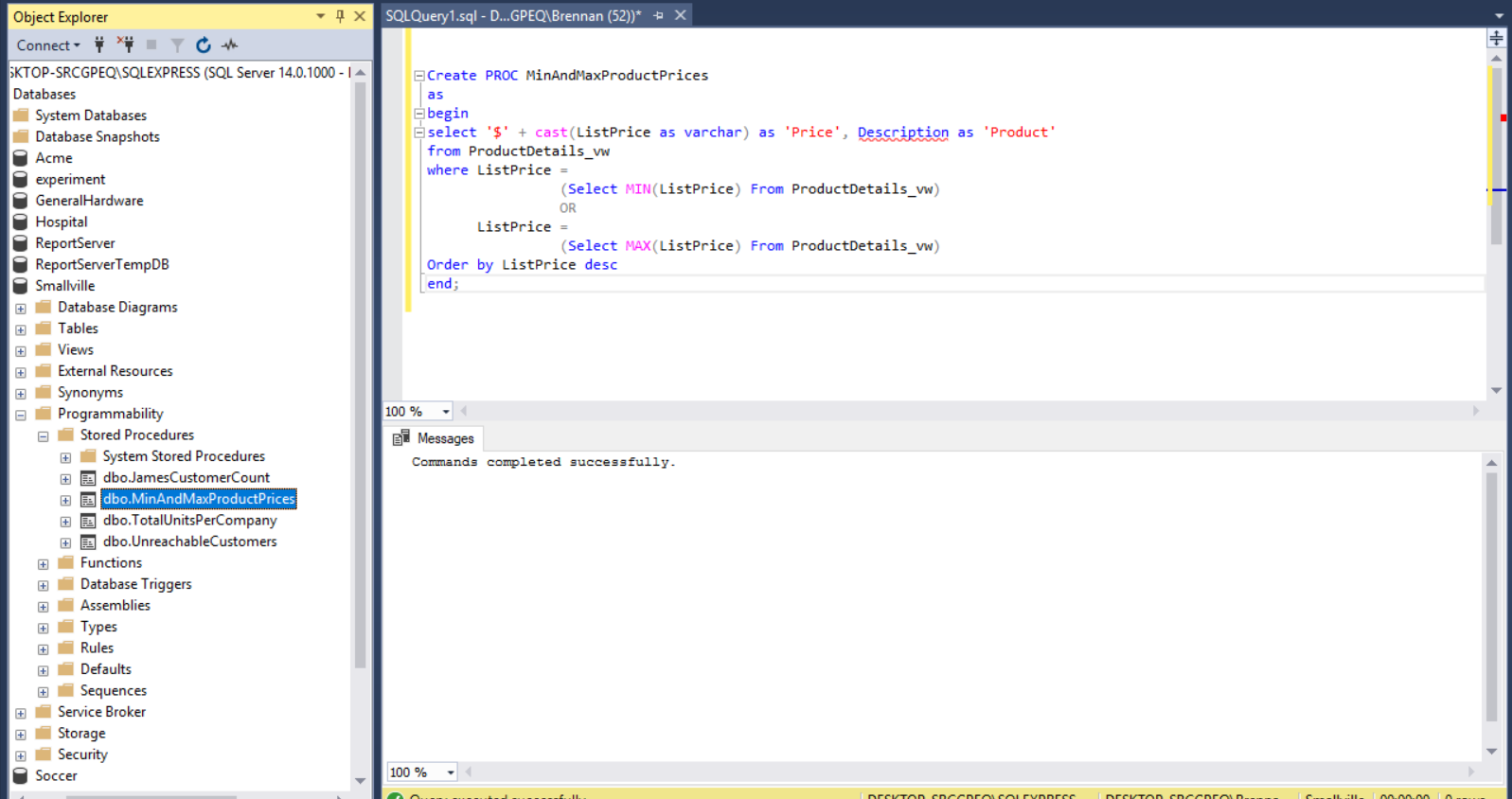
**Total Units per Company SPROC**

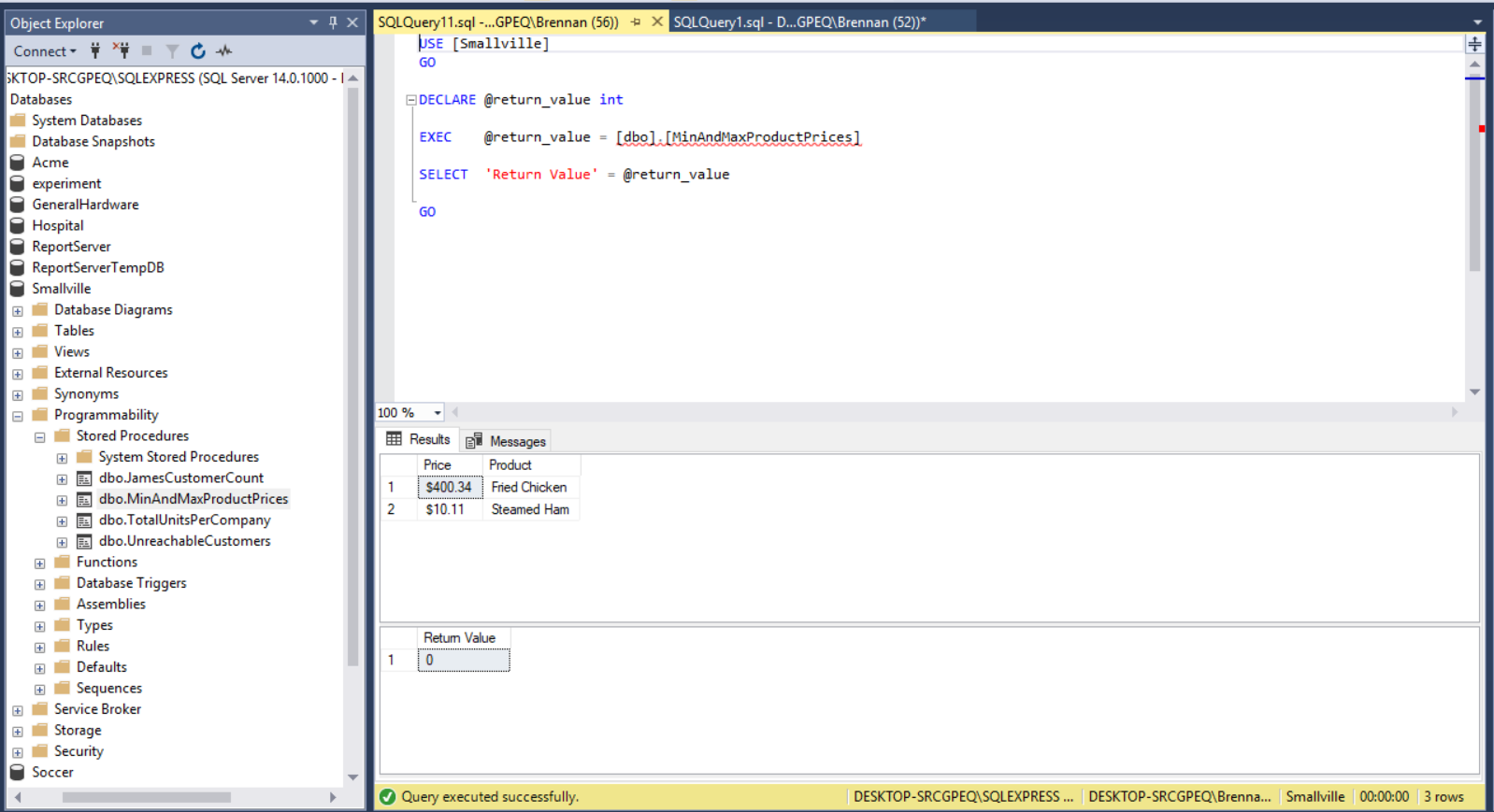


**Total Units per Company SPROC Description**

The Total Units Per Company SPROC was created from the Product Details table view. This SPROC searches for the amount of products provided from all suppliers besides Northwind. This stored procedure creates the column headers “Suppliers” and “Total Products Supplied.” A sum aggregtate function is used for the NumberofUnits column. The aggregator must be used in conjunction with a group by clause which grouped the result set by the CompanyName column. Finally, the results were also sorted by the total number of units per supplier in descending order.

**Min and Max Product Prices SPROC**

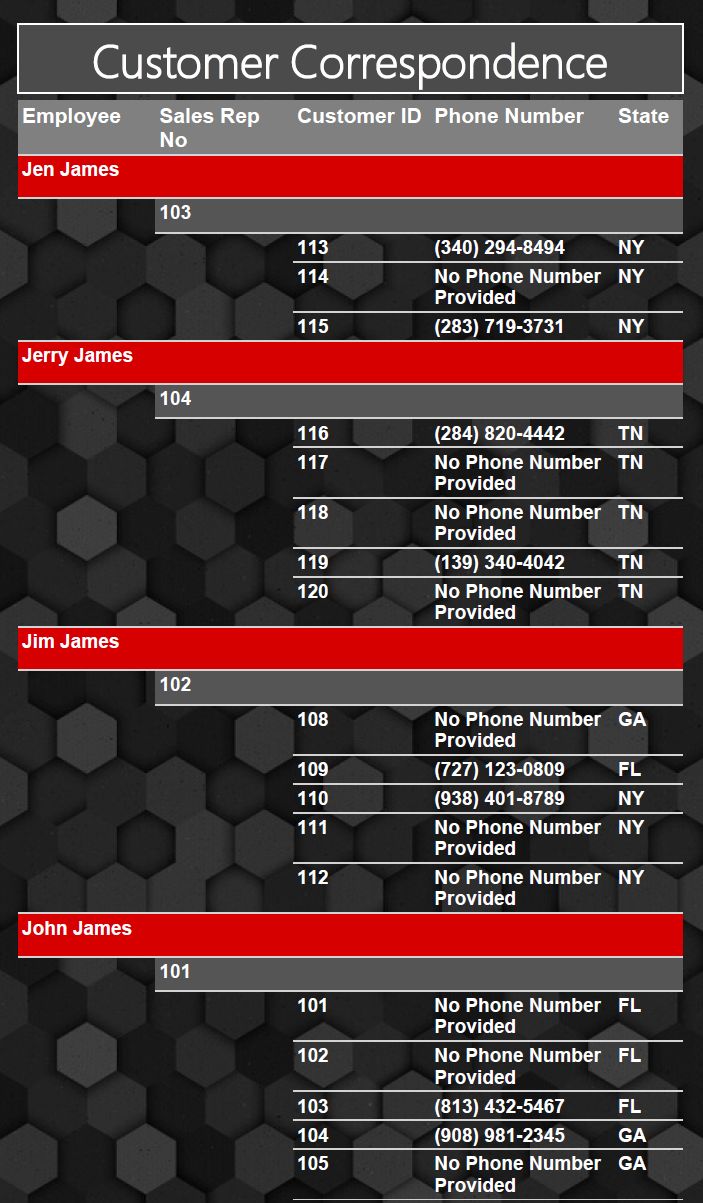


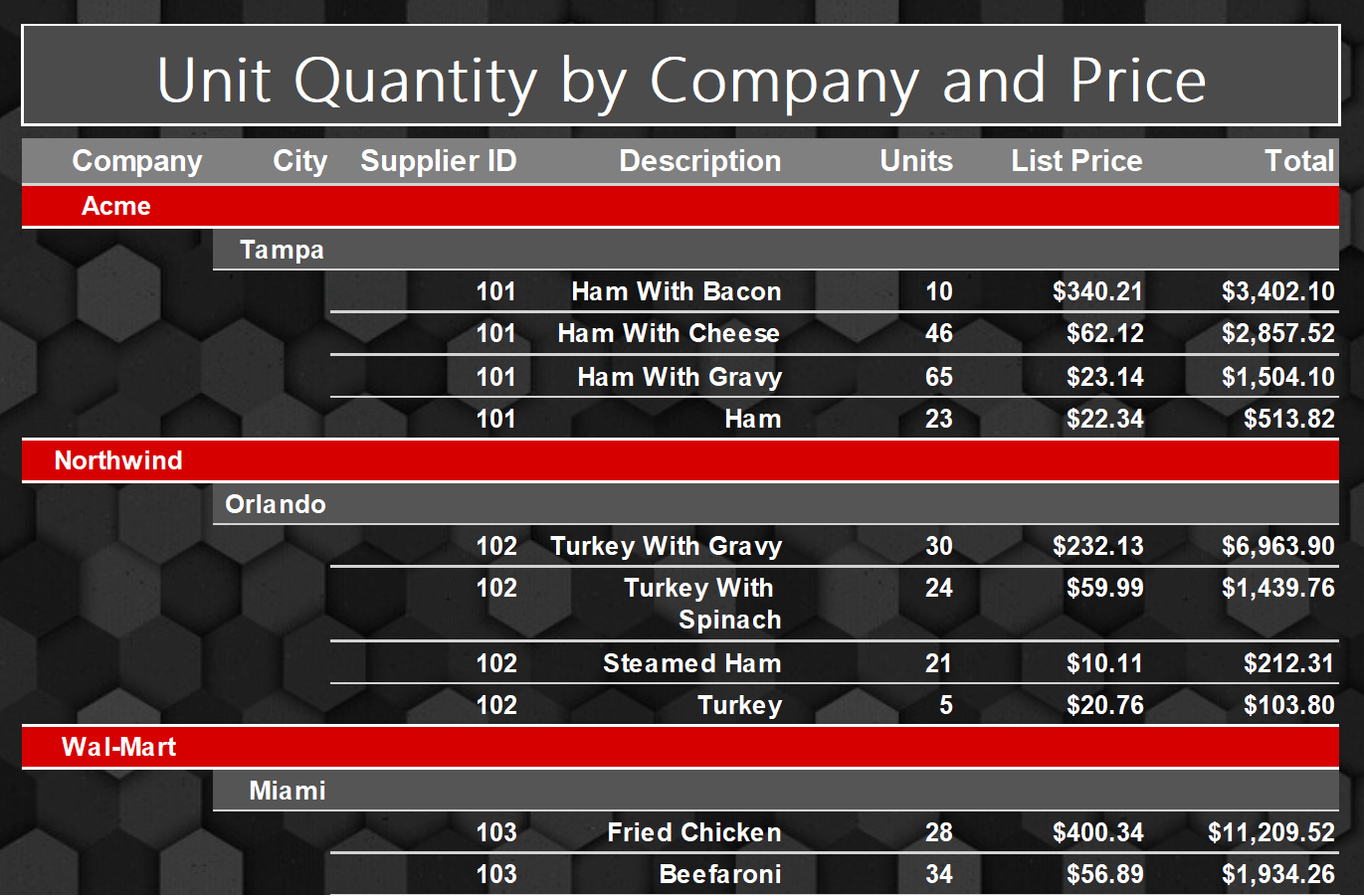


**Min and Max Product Prices SPROC Description**

The SPROC named MinandMaxProductPrices was created using the Product Details table view. This SPROC searches for the products that have the highest or lowest prices. A concatenation is used in the select statement in conjunction with the Price column in order to add a “$” character in front of the List Price field. To find the minimum and maximum list prices, the MIN and MAX functions are applied in the where clause. The headers used by the SPROC include the “Price” and “Product” columns. The SPROC results are then sorted by the ListPrice field in descending order.

**Customer Correspondence Table View Report**



**Product Details Table View Report**

**User Acceptance Test Queries**

What is the total compensation earned for hourly employees on each date worked, starting from the most recent date?

select '$' + cast(convert(decimal(22,2),sum([compensation])) as varchar) as 'Total Compensation',

concat([FirstName], ' ', [LastName]) as 'Hourly Employee', [DateWorked] as 'Date'

from [Hourly]

join [Employee]

on [Employee].[EmployeeID] = [Hourly].[EmployeeID]

where [DateWorked] is not null and [Compensation] is not null

group by [FirstName], [LastName], [DateWorked]

order by [DateWorked] desc

How many orders have all customers placed, starting at the most orders placed?

select count([OrderID]) as 'Number of Orders', concat([FirstName], ' ', [LastName]) as 'Customer'

from [Order]

right join [customer]

on [Order].[CustomerID] = [Customer].[CustomerID]

group by [FirstName], [LastName]

order by 'Number of Orders' desc

What are the emails for all employees?

select isnull(convert(varchar(15), [Email]), 'No Email') as 'Email', concat([FirstName], ' ', [LastName]) as 'Employee'

from [Employee]