Team Name: Database Dimez

Team Motto: D’s are for Database

Team Lead:

Robert Lumadue

Team Members:

Comeskey, Matthew

Elliott, Kaydiann

Hartman, Justin

Contents

Purpose

The goal of this document is to outline how a database can be created, mapped out and implemented for a private business that mainly uses excel spreadsheets. Contained in this document are the narrative, requirements, entities, ERD/EERD, relational schema, data dictionary, table view, SPROCS, Reports, and user test queries.

Narrative

Smallville is a small privately owned business that wants to create a database for managing and tracking customers, orders, suppliers, products and employees rather than using multiple excel spreadsheets.

Smallville wants to be able to track the three different types of employees they have: Salary, Sales and Hourly. Each employee will have name, address, phone number and email. Sales employees will be assigned a customer, but can have more than one, and each customer can have more than one sales employee.

Customers place orders through smallville and those orders will contain products. Each customer will have a customer number, first name, last name, address and phone number. Once an order is placed, the order will populate in the system and will contain an order number, date of order, and status. Orders are made up of products that smallville keeps in their system. Each product will have a product number, description, “list price,” and quantity available to identify them. An order has to contain at least one product, but can include many products. Once a product is ordered it can be tracked by quantity ordered, quantity supplied, and
“Order price”.

Smallville has also decided to start tracking their suppliers and the products that they get from each supplier. Each supplier provides at least one product and different suppliers can carry the same product. Therefore Smallville has many suppliers that provide them with many products.

Requirements

(Actors and Roles)

CUSTOMER - A customer places one or more orders. A customer can be assigned to one or more Sales Reps.

ORDER - An order can only belong to one and only one customer. Orders are for one or more products.

PRODUCT- A product has multiple orders or it has no order associated. A product can have more than one supplier.

EMPLOYEE- There are three categories of employees: Sales Reps, Hourly Workers, and Salary Workers. Sales Reps have at least one or more customers.

SUPPLIER- A supplier can offer one or more products.

Entities Identified to be tracked

1. CUSTOMER
2. ORDER
3. PRODUCT
4. EMPLOYEE
5. SUPPLIER

Entities with Nested Attributes

CUSTOMER - CustomerNumber, First Name, Last Name, Street, City, State, Zip Code, Phone Number

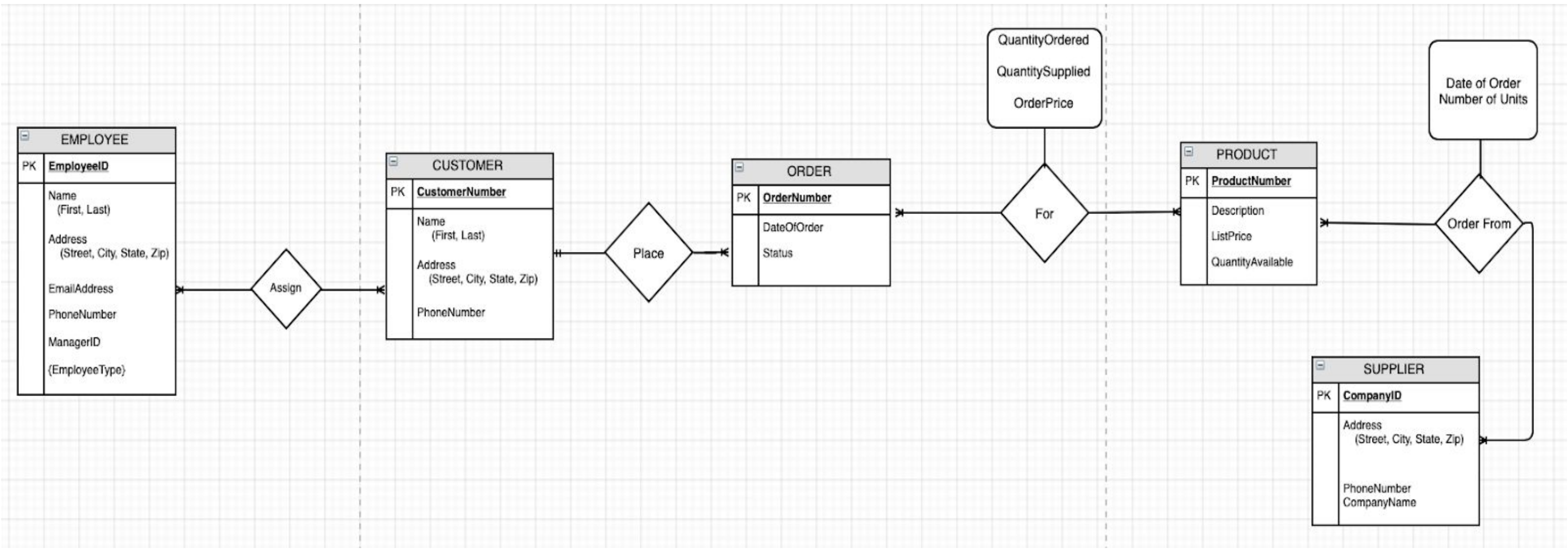
ORDER - OrderNumber, Date of Order, Status

PRODUCT - ProductNumber, Description, List Price, Quantity Available

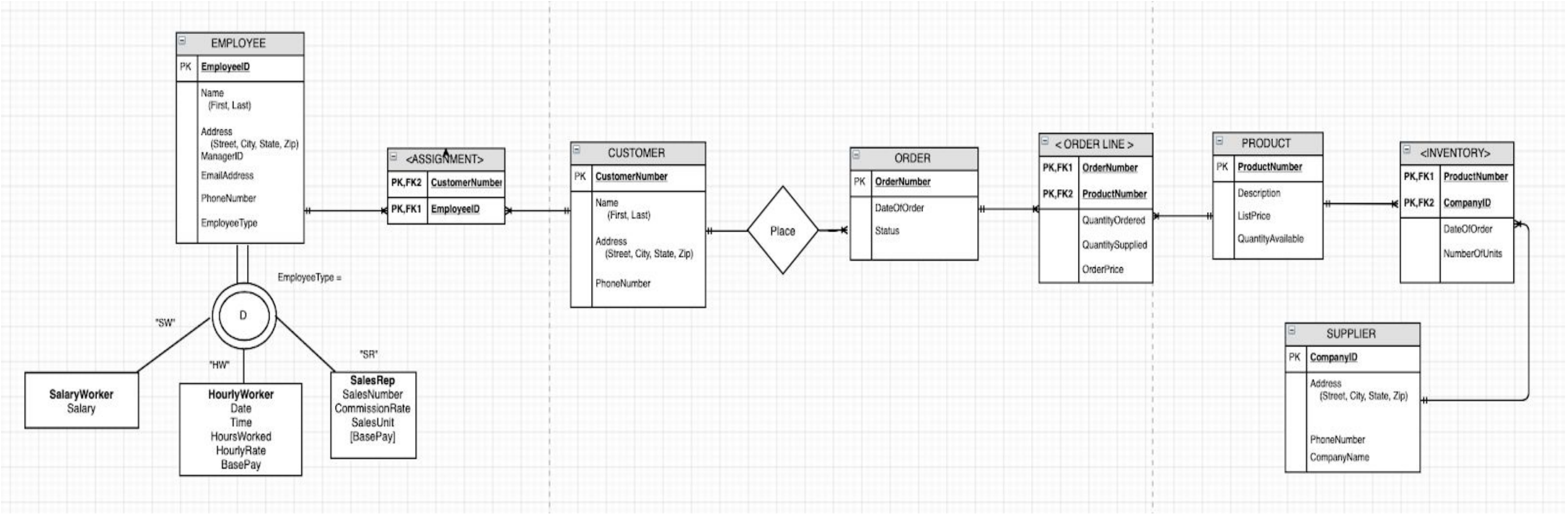
EMPLOYEE - EmployeeID, First Name, Last Name, Street, City, State, Zip Code, Email Address, Phone Number, Employee Type

SUPPLIER - CompanyID, CompanyName, Street, City, State, Zip Code, Phone Number

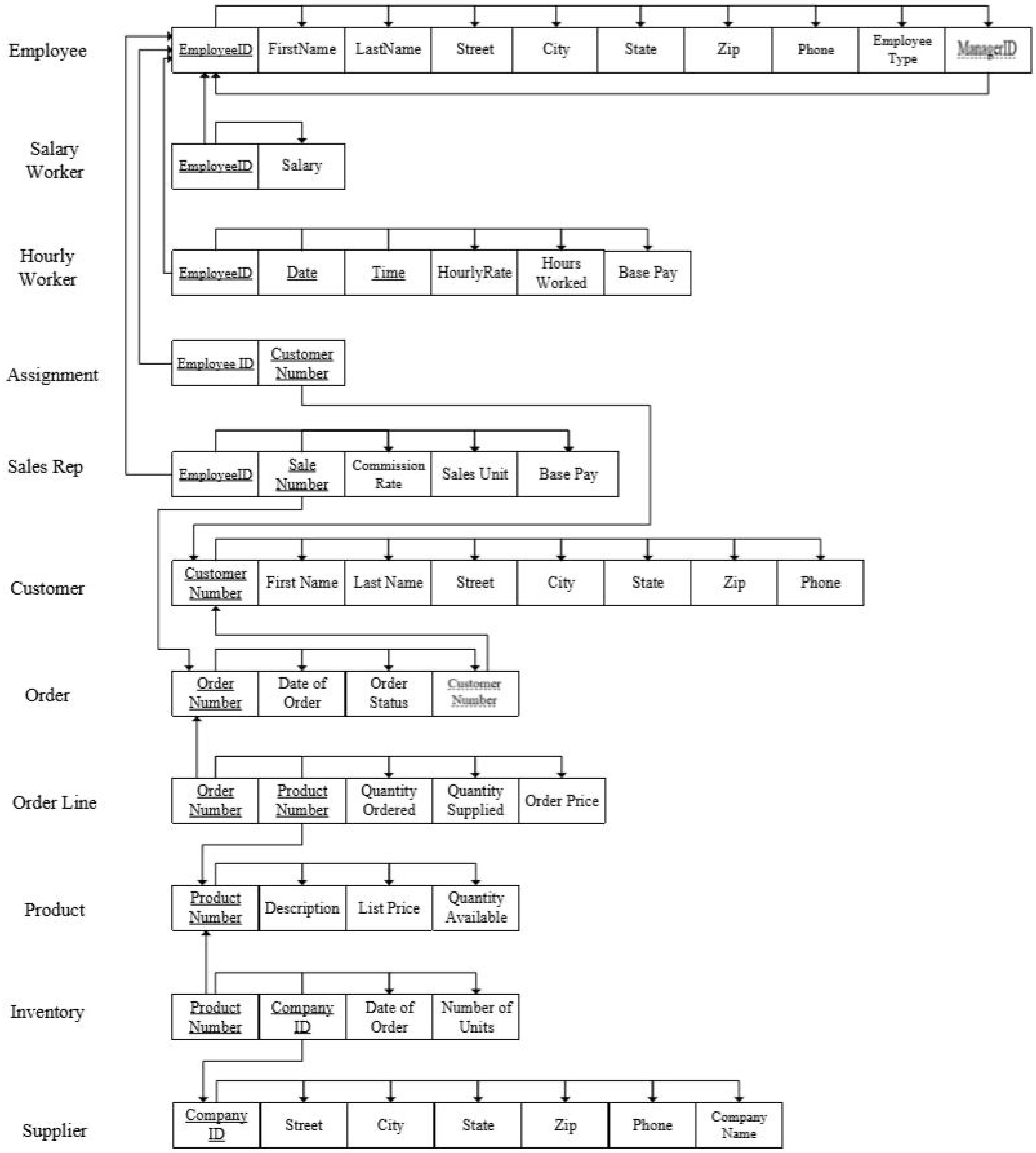
ERD


EMPLOYEE
PK
EmployeeID
Name
(First, Last)
Address
(Street, City, State, Zip)
EmailAddress
PhoneNumber
ManagerID
{EmployeeType}
Assign
CUSTOMER
PK
CustomerNumber
Name
(First, Last)
Address
(Street, City, State, Zip)
PhoneNumber
Place
ORDER
PK
OrderNumber
DateOfOrder
Status
QuantityOrdered
QuantitySupplied
OrderPrice
For
PRODUCT
PK
ProductNumber
Description
ListPrice
QuantityAvailable
Date of Order
Number of Units
Order From
SUPPLIER
PK
CompanyID
Address
(Street, City, State, Zip)
PhoneNumber
CompanyName

EERD


EMPLOYEE
PK
EmployeeID
Name
(First, Last)
Address
(Street, City, State, Zip)
ManagerID
EmailAddress
PhoneNumber
EmployeeType
EmployeeType =
D
"SW"
"HW"
"SR"
SalaryWorker
Salary
HourlyWorker
Date
Time
HoursWorked
HourlyRate
BasePay
SalesRep
SalesNumber
CommissionRate
SalesUnit
[BasePay]
<ASSIGNMENT>
PK,FK2
CustomerNumber
PK,FK1
EmployeeID
CUSTOMER
PK
CustomerNumber
Name
(First, Last)
Address
(Street, City, State, Zip)
PhoneNumber
Place
ORDER
PK
OrderNumber
DateOfOrder
Status
< ORDER LINE >
PK,FK1
OrderNumber
PK,FK2
ProductNumber
QuantityOrdered
QuantitySupplied
OrderPrice
PRODUCT
PK
ProductNumber
Description
ListPrice
QuantityAvailable
<INVENTORY>
PK,FK1
ProductNumber
PK,FK2
CompanyID
DateOfOrder
NumberOfUnits
SUPPLIER
PK
CompanyID
Address
(Street, City, State, Zip)
PhoneNumber
CompanyName

Relational Schema



Data Dictionary Summary

Employee (EmployeeID, FirstName, LastName, Street, City, Sate, Zip, Email, PhoneNumber, EmployeeType, ManagerID)

Salary Worker (EmployeeID, Salary)

Hourly Worker (EmployeeID, Date, Time, HoursWorked, HourlyRate, [BasePay],)

Sales Rep (EmployeeID, SaleNumber,Commission Rate, [BasePay], SalesUnit)

Assignment (EmployeeID, CustomerNumber)

Customer (CustomerNumber, FirstName, LastName, Street, City, Sate, Zip, PhoneNumber)

Order (OrderNumber, OrderPrice, OrderStatus, CustomerNumber)

OrderLine (OrderNumber, ProductNumber, QuantityOrdered, QuantitySupplied, OrderPrice)

Product (ProductNumber, Description, ListPrice, QuantityAvailable)

Invetory (ProductNumber, CompanyID, OrderDate, NumberOfUnits)

Supplier (CompanyID, CompanyName, Street, City, Sate, Zip, PhoneNumber)

Smallville - Data Dictionary

(Microsoft SQL server notation)

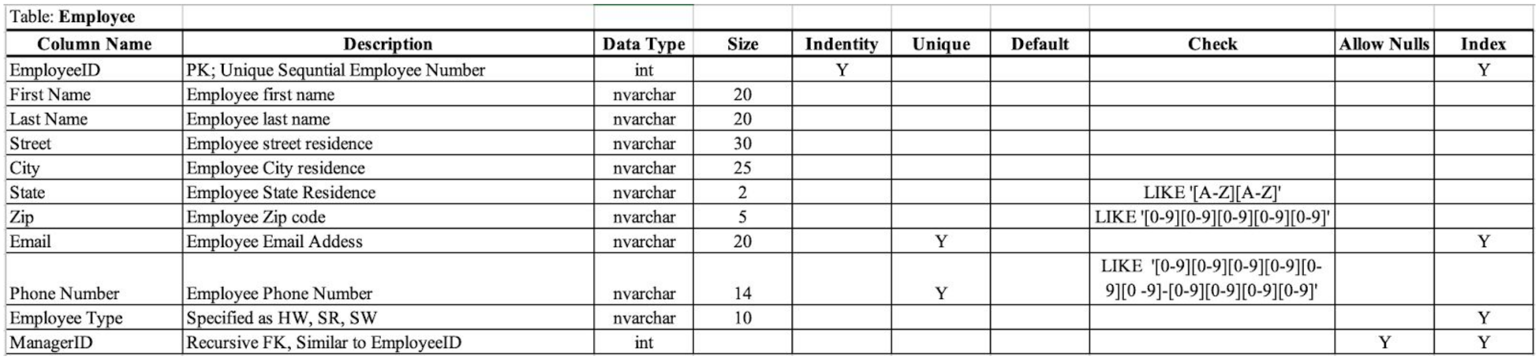


Table: **Employee**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
EmployeeID	PK; Unique Sequntial Employee Number	int		Y					Y
First Name	Employee first name	nvarchar	20						
Last Name	Employee last name	nvarchar	20						
Street	Employee street residence	nvarchar	30						
City	Employee City residence	nvarchar	25						
State	Employee State Residence	nvarchar	2				LIKE '[A-Z][A-Z]'		
Zip	Employee Zip code	nvarchar	5				LIKE '[0-9][0-9][0-9][0-9][0-9]'		
Email	Employee Email Addess	nvarchar	20		Y				Y
Phone Number	Employee Phone Number	nvarchar	14		Y		LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0 -9]-[0-9][0-9][0-9][0-9]'		
Employee Type	Specified as HW, SR, SW	nvarchar	10						Y
ManagerID	Recursive FK, Similar to EmployeeID	int						Y	Y

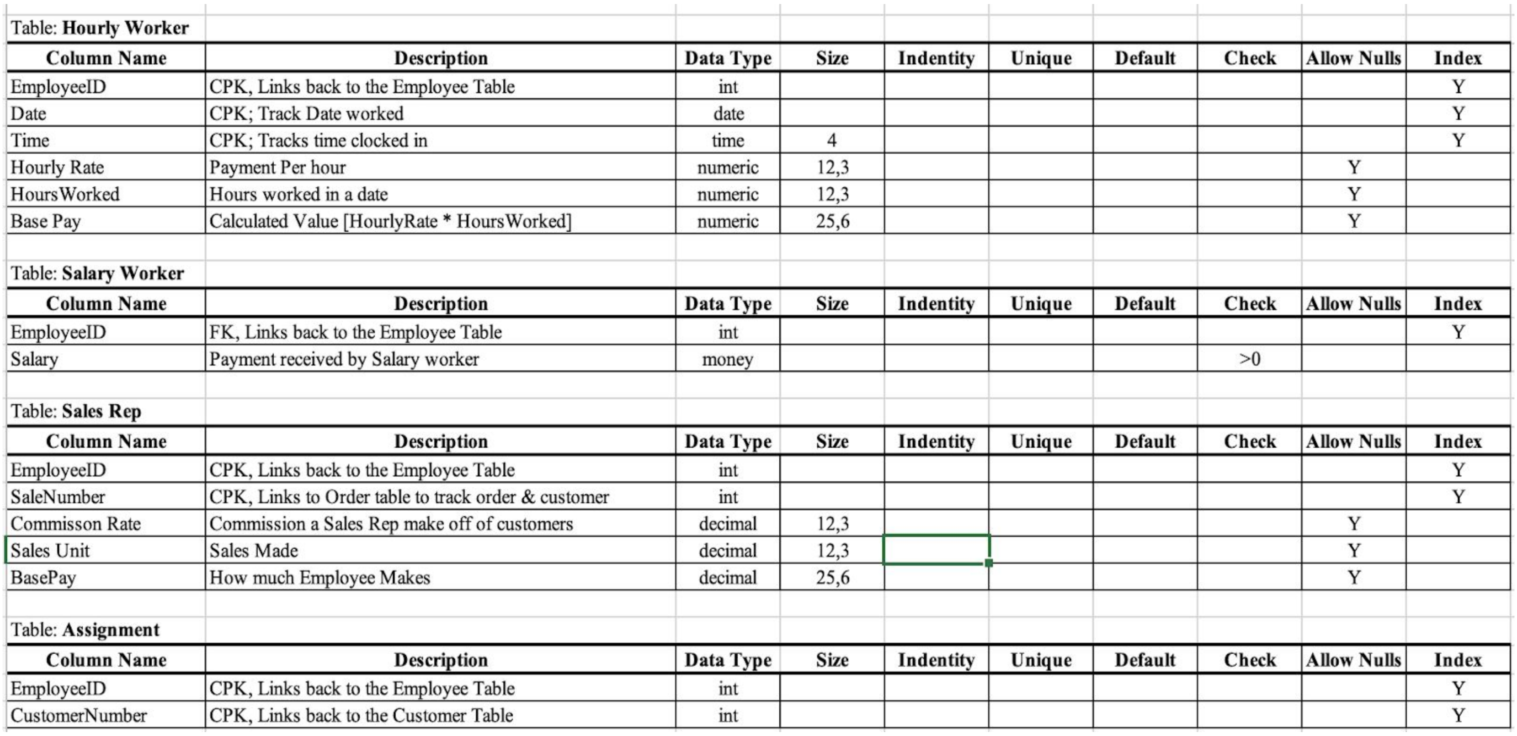


Table: **Hourly Worker**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
EmployeeID	CPK, Links back to the Employee Table	int							Y
Date	CPK; Track Date worked	date							Y
Time	CPK; Tracks time clocked in	time	4						Y
Hourly Rate	Payment Per hour	numeric	12,3					Y	
HoursWorked	Hours worked in a date	numeric	12,3					Y	
Base Pay	Calculated Value [HourlyRate * HoursWorked]	numeric	25,6					Y	

Table: **Salary Worker**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
EmployeeID	FK, Links back to the Employee Table	int							Y
Salary	Payment received by Salary worker	money					>0		

Table: **Sales Rep**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
EmployeeID	CPK, Links back to the Employee Table	int							Y
SaleNumber	CPK, Links to Order table to track order & customer	int							Y
Commisson Rate	Commission a Sales Rep make off of customers	decimal	12,3					Y	
Sales Unit	Sales Made	decimal	12,3					Y	
BasePay	How much Employee Makes	decimal	25,6					Y	

Table: **Assignment**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
EmployeeID	CPK, Links back to the Employee Table	int							Y
CustomerNumber	CPK, Links back to the Customer Table	int							Y

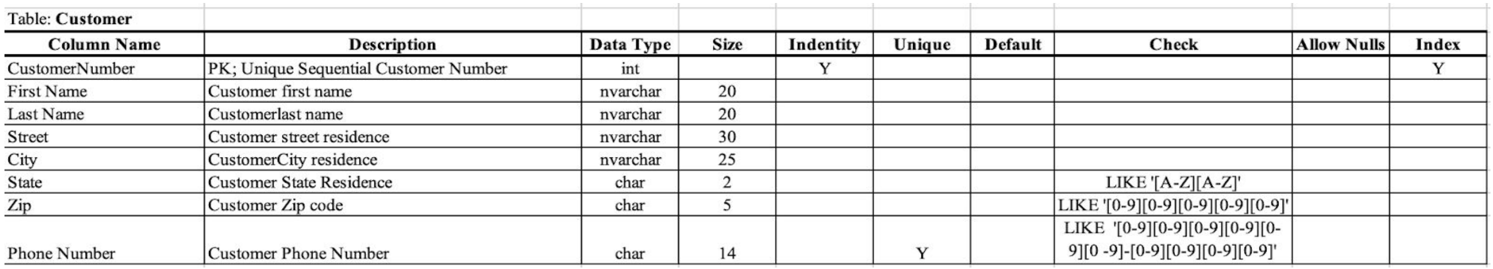


Table: **Customer**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
CustomerNumber	PK; Unique Sequential Customer Number	int		Y					Y
First Name	Customer first name	nvarchar	20						
Last Name	Customerlast name	nvarchar	20						
Street	Customer street residence	nvarchar	30						
City	CustomerCity residence	nvarchar	25						
State	Customer State Residence	char	2				LIKE '[A-Z][A-Z]'		
Zip	Customer Zip code	char	5				LIKE '[0-9][0-9][0-9][0-9][0-9]'		
Phone Number	Customer Phone Number	char	14		Y		LIKE '[0-9][0-9][0-9][0-9][0-9][0 -9]-[0-9][0-9][0-9][0-9]'		

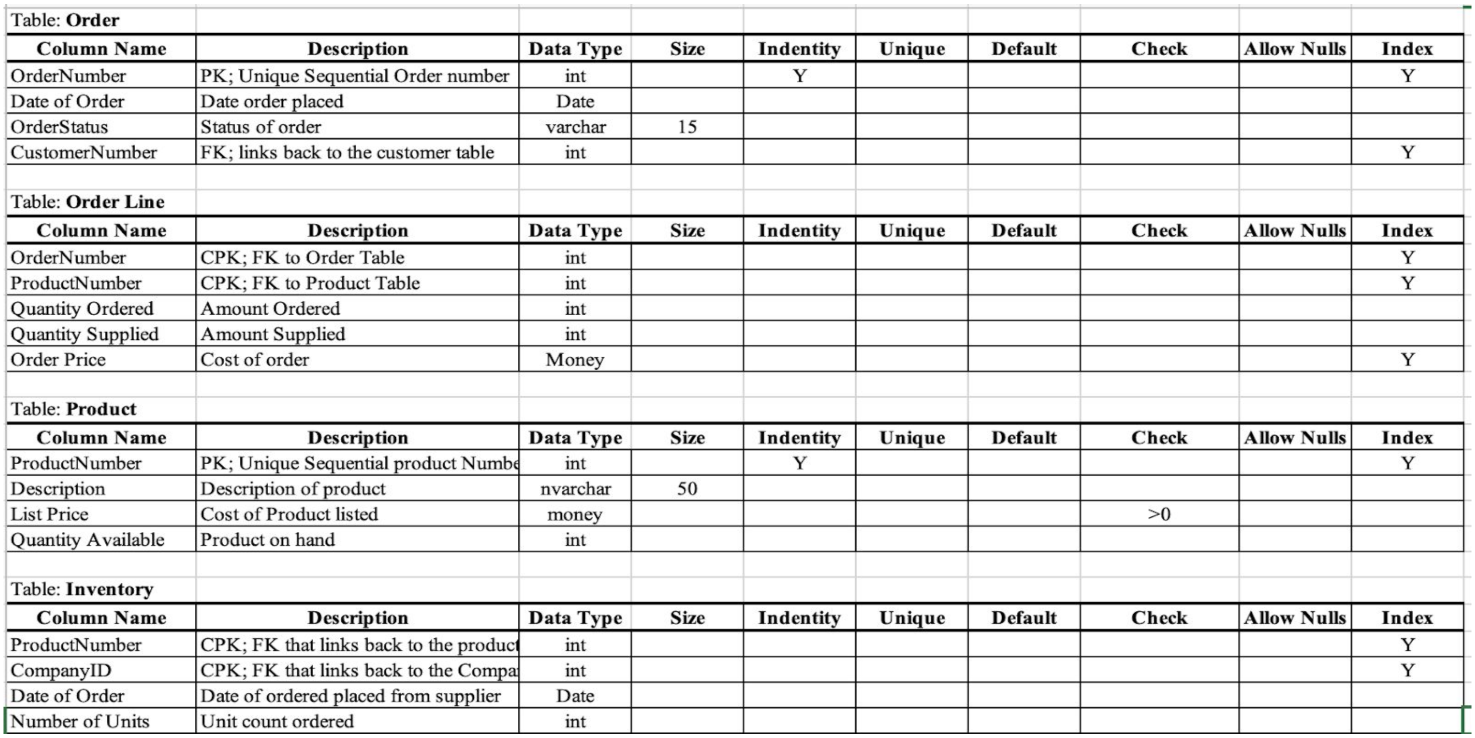


Table: **Order**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
OrderNumber	PK; Unique Sequential Order number	int		Y					Y
Date of Order	Date order placed	Date							
OrderStatus	Status of order	varchar	15						
CustomerNumber	FK; links back to the customer table	int							Y

Table: **Order Line**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
OrderNumber	CPK; FK to Order Table	int							Y
ProductNumber	CPK; FK to Product Table	int							Y
Quantity Ordered	Amount Ordered	int							
Quantity Supplied	Amount Supplied	int							
Order Price	Cost of order	Money							Y

Table: **Product**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
ProductNumber	PK; Unique Sequential product Numbe	int		Y					Y
Description	Description of product	nvarchar	50						
List Price	Cost of Product listed	money					>0		
Quantity Available	Product on hand	int							

Table: **Inventory**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
ProductNumber	CPK; FK that links back to the product	int							Y
CompanyID	CPK; FK that links back to the Compa	int							Y
Date of Order	Date of ordered placed from supplier	Date							
Number of Units	Unit count ordered	int							

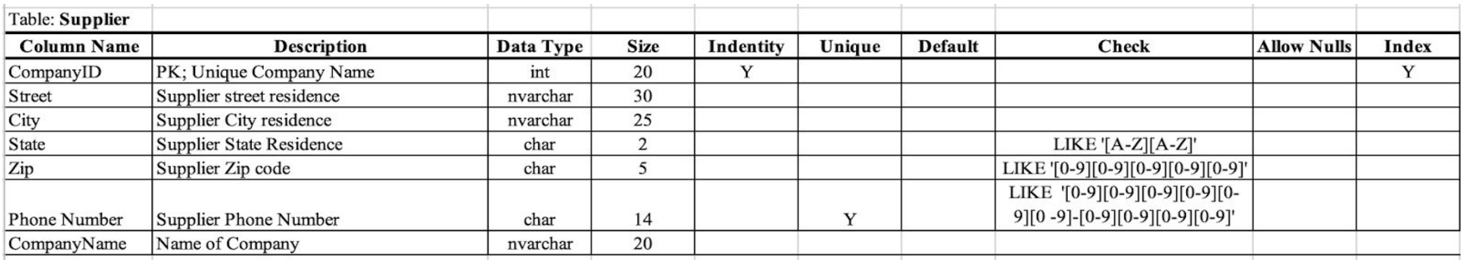


Table: **Supplier**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
CompanyID	PK; Unique Company Name	int	20	Y					Y
Street	Supplier street residence	nvarchar	30						
City	Supplier City residence	nvarchar	25						
State	Supplier State Residence	char	2				LIKE '[A-Z][A-Z]'		
Zip	Supplier Zip code	char	5				LIKE '[0-9][0-9][0-9][0-9][0-9]'		
Phone Number	Supplier Phone Number	char	14		Y		LIKE '[0-9][0-9][0-9][0-9][0-9][0 -9]-[0-9][0-9][0-9][0-9]'		
CompanyName	Name of Company	nvarchar	20						

Data Diagram

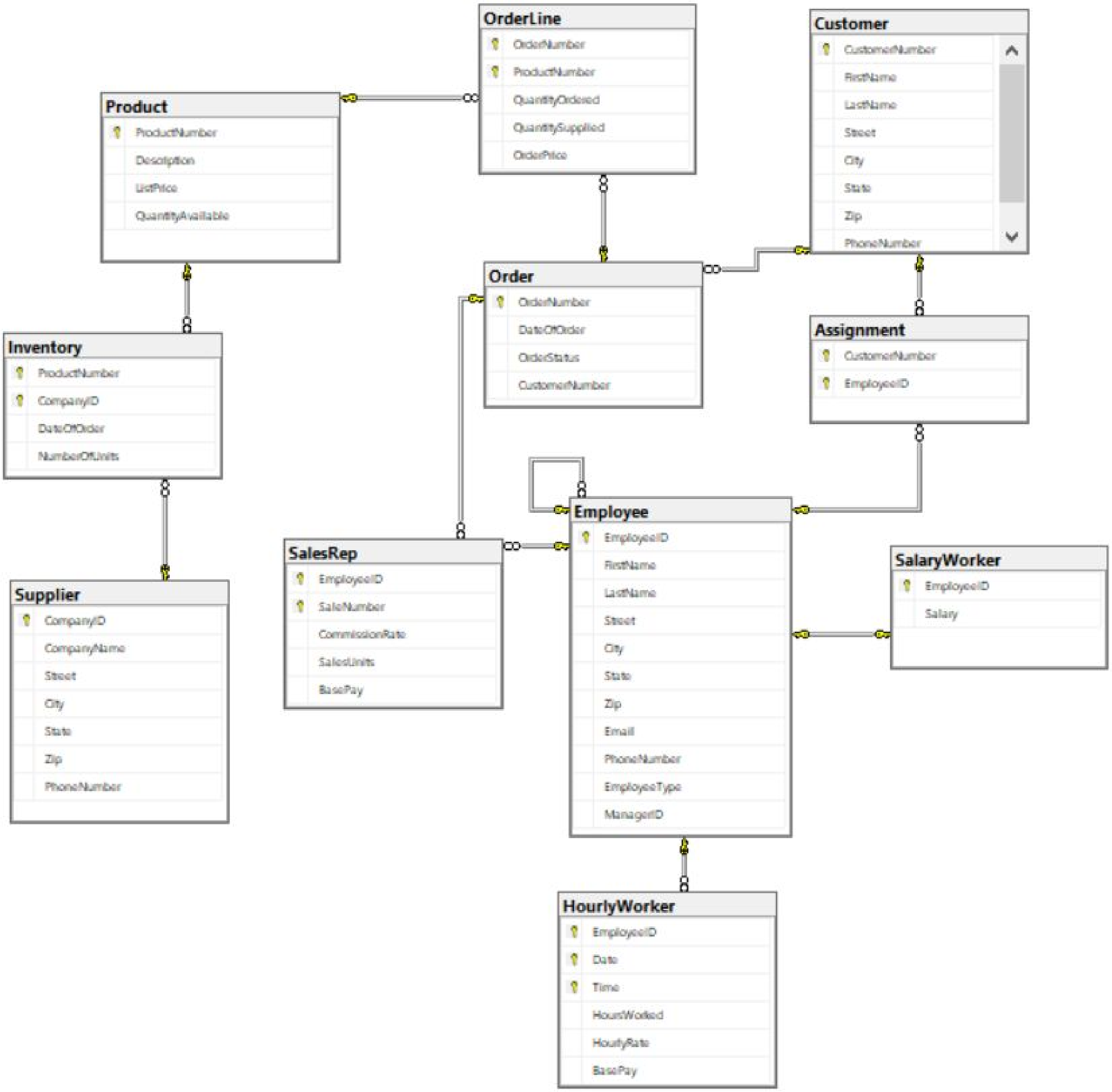
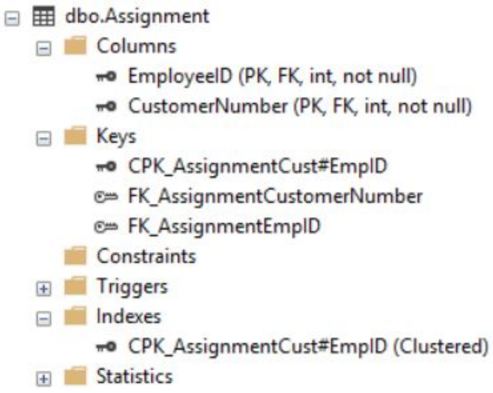


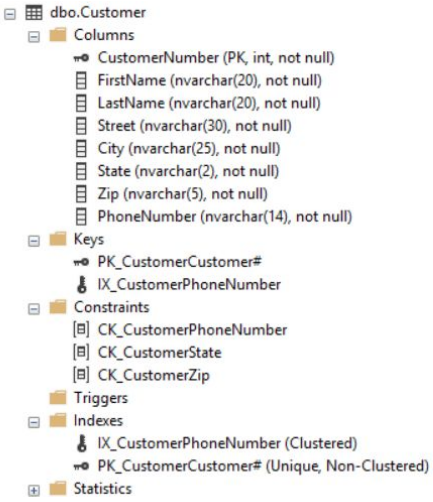
Table Views

The assignment table is used to track the Employee that is working with the customer. SalesID is the sales rep ID number.



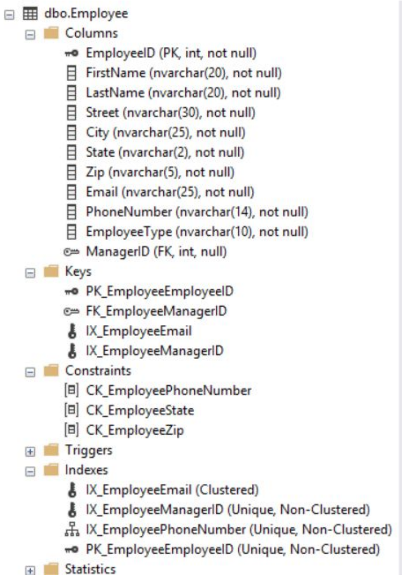
```
dbo.Assignment
  Columns
    EmployeeID (PK, FK, int, not null)
    CustomerNumber (PK, FK, int, not null)
  Keys
    CPK_AssignmentCust#EmpID
    FK_AssignmentCustomerNumber
    FK_AssignmentEmpID
  Constraints
  Triggers
  Indexes
    CPK_AssignmentCust#EmpID (Clustered)
  Statistics
```

The customer table is used to track information of the customer.



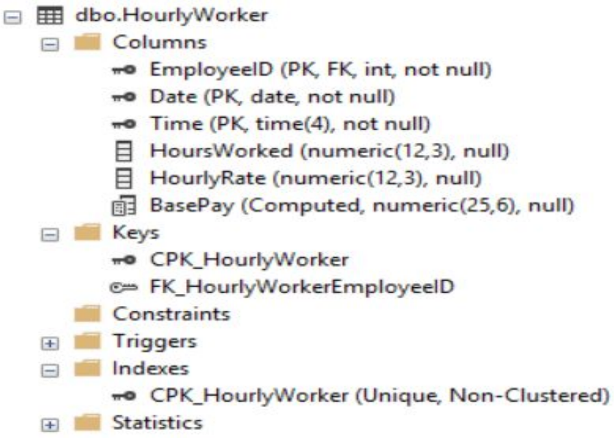
```
dbo.Customer
  Columns
    CustomerNumber (PK, int, not null)
    FirstName (nvarchar(20), not null)
    LastName (nvarchar(20), not null)
    Street (nvarchar(30), not null)
    City (nvarchar(25), not null)
    State (nvarchar(2), not null)
    Zip (nvarchar(5), not null)
    PhoneNumber (nvarchar(14), not null)
  Keys
    PK_CustomerCustomer#
    IX_CustomerPhoneNumber
  Constraints
    CK_CustomerPhoneNumber
    CK_CustomerState
    CK_CustomerZip
  Triggers
  Indexes
    IX_CustomerPhoneNumber (Clustered)
    PK_CustomerCustomer# (Unique, Non-Clustered)
  Statistics
```

The Employee table tracks Employee information and lists employee type and managerID.



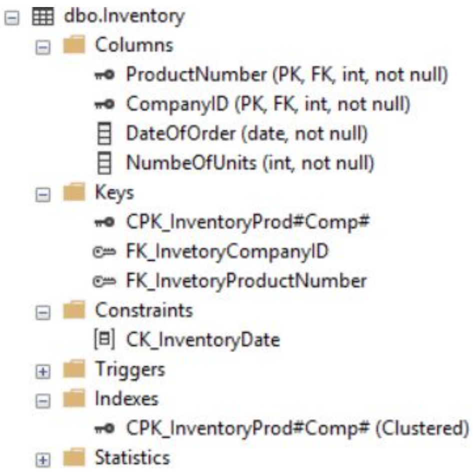
```
dbo.Employee
  Columns
    EmployeeID (PK, int, not null)
    FirstName (nvarchar(20), not null)
    LastName (nvarchar(20), not null)
    Street (nvarchar(30), not null)
    City (nvarchar(25), not null)
    State (nvarchar(2), not null)
    Zip (nvarchar(5), not null)
    Email (nvarchar(25), not null)
    PhoneNumber (nvarchar(14), not null)
    EmployeeType (nvarchar(10), not null)
    ManagerID (FK, int, null)
  Keys
    PK_EmployeeEmployeeID
    FK_EmployeeManagerID
    IX_EmployeeEmail
    IX_EmployeeManagerID
  Constraints
    CK_EmployeePhoneNumber
    CK_EmployeeState
    CK_EmployeeZip
  Triggers
  Indexes
    IX_EmployeeEmail (Clustered)
    IX_EmployeeManagerID (Unique, Non-Clustered)
    IX_EmployeePhoneNumber (Unique, Non-Clustered)
    PK_EmployeeEmployeeID (Unique, Non-Clustered)
  Statistics
```

Hourly worker is tracked by an FK EmployeeID and tracks payment.



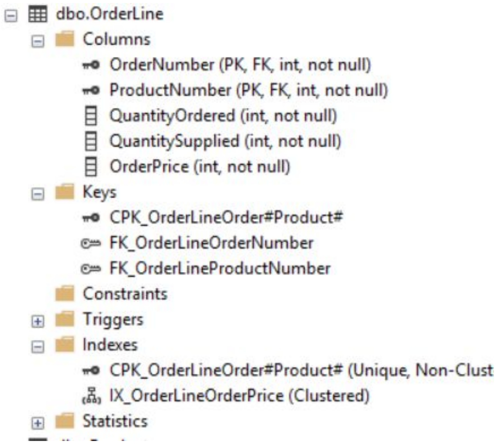
```
dbo.HourlyWorker
  Columns
    EmployeeID (PK, FK, int, not null)
    Date (PK, date, not null)
    Time (PK, time(4), not null)
    HoursWorked (numeric(12,3), null)
    HourlyRate (numeric(12,3), null)
    BasePay (Computed, numeric(25,6), null)
  Keys
    CPK_HourlyWorker
    FK_HourlyWorkerEmployeeID
  Constraints
  Triggers
  Indexes
    CPK_HourlyWorker (Unique, Non-Clustered)
  Statistics
```

The inventory table is tracked by product number and company ID .



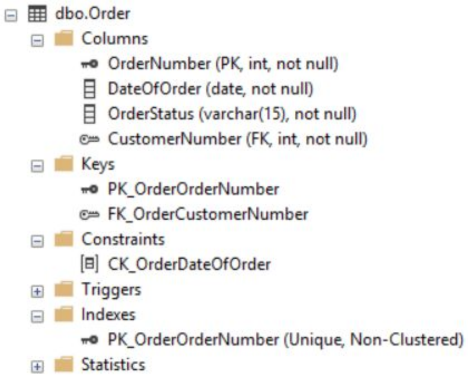
```
dbo.Inventory
  Columns
    ProductNumber (PK, FK, int, not null)
    CompanyID (PK, FK, int, not null)
    DateOfOrder (date, not null)
    NumbeOfUnits (int, not null)
  Keys
    CPK_InventoryProd#Comp#
    FK_InvetoryCompanyID
    FK_InvetoryProductNumber
  Constraints
    CK_InventoryDate
  Triggers
  Indexes
    CPK_InventoryProd#Comp# (Clustered)
  Statistics
```

The order Line table is used to tracked Order Number and product Number as well as other info in regards to the order.



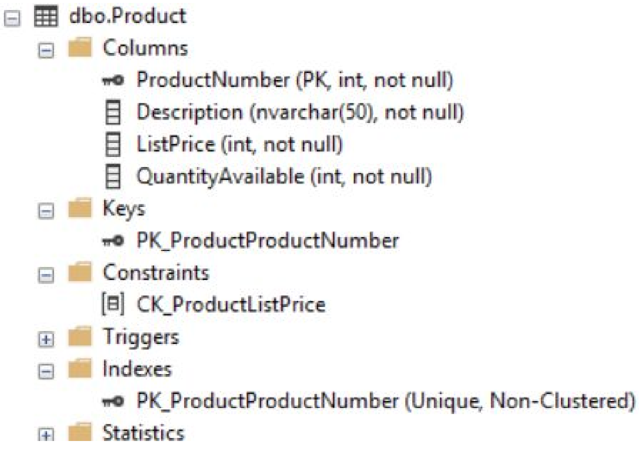
```
⊟ dbo.OrderLine
    ⊟ Columns
          OrderNumber (PK, FK, int, not null)
          ProductNumber (PK, FK, int, not null)
          QuantityOrdered (int, not null)
          QuantitySupplied (int, not null)
          OrderPrice (int, not null)
    ⊟ Keys
          CPK_OrderLineOrder#Product#
          FK_OrderLineOrderNumber
          FK_OrderLineProductNumber
      Constraints
    ⊞ Triggers
    ⊟ Indexes
          CPK_OrderLineOrder#Product# (Unique, Non-Clust
          IX_OrderLineOrderPrice (Clustered)
    ⊞ Statistics
```

Order tracks Order Number and links to the customer table to track which customer placed the order.



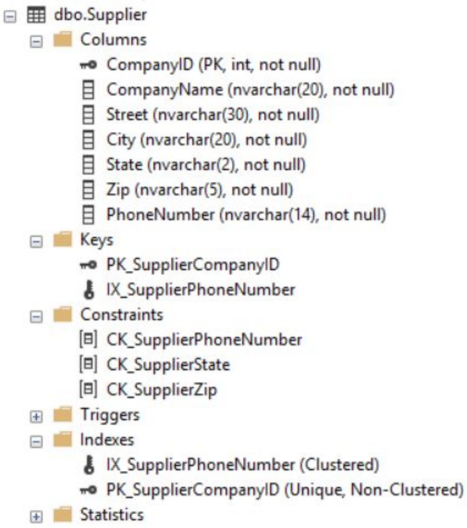
```
⊟ dbo.Order
    ⊟ Columns
          OrderNumber (PK, int, not null)
          DateOfOrder (date, not null)
          OrderStatus (varchar(15), not null)
          CustomerNumber (FK, int, not null)
    ⊟ Keys
          PK_OrderOrderNumber
          FK_OrderCustomerNumber
    ⊟ Constraints
          CK_OrderDateOfOrder
    ⊞ Triggers
    ⊟ Indexes
          PK_OrderOrderNumber (Unique, Non-Clustered)
    ⊞ Statistics
```

The product table tracks information regarding the product.



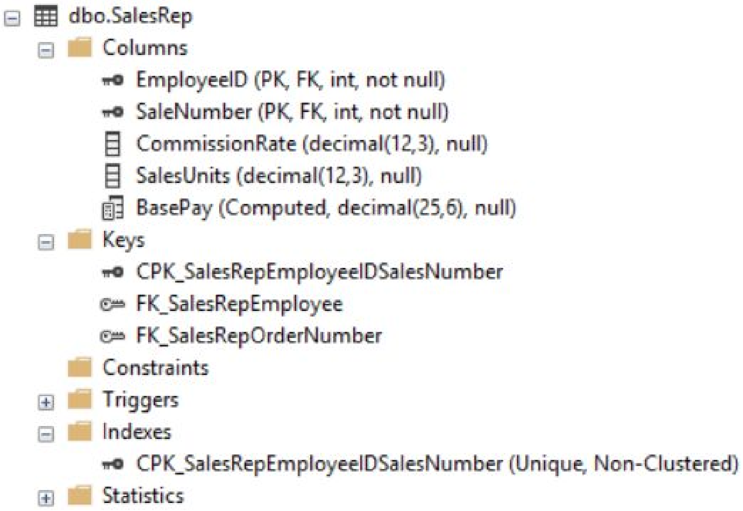
```
⊟ dbo.Product
    ⊟ Columns
          ProductNumber (PK, int, not null)
          Description (nvarchar(50), not null)
          ListPrice (int, not null)
          QuantityAvailable (int, not null)
    ⊟ Keys
          PK_ProductProductNumber
    ⊟ Constraints
          CK_ProductListPrice
    ⊞ Triggers
    ⊟ Indexes
          PK_ProductProductNumber (Unique, Non-Clustered)
    ⊞ Statistics
```

The supplier table is used to track the company that product is being ordered from.



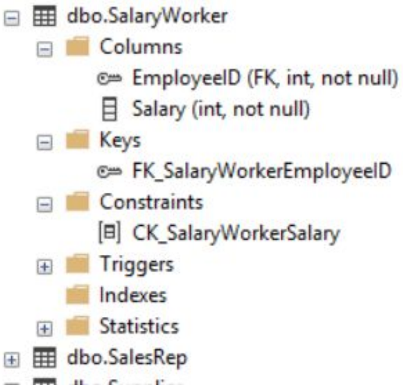
```
⊟ dbo.Supplier
    ⊟ Columns
          CompanyID (PK, int, not null)
          CompanyName (nvarchar(20), not null)
          Street (nvarchar(30), not null)
          City (nvarchar(20), not null)
          State (nvarchar(2), not null)
          Zip (nvarchar(5), not null)
          PhoneNumber (nvarchar(14), not null)
    ⊟ Keys
          PK_SupplierCompanyID
          IX_SupplierPhoneNumber
    ⊟ Constraints
          CK_SupplierPhoneNumber
          CK_SupplierState
          CK_SupplierZip
    ⊞ Triggers
    ⊟ Indexes
          IX_SupplierPhoneNumber (Clustered)
          PK_SupplierCompanyID (Unique, Non-Clustered)
    ⊞ Statistics
```

The Sales Rep table tracks sales rep information and pay. Linking back to the employee table



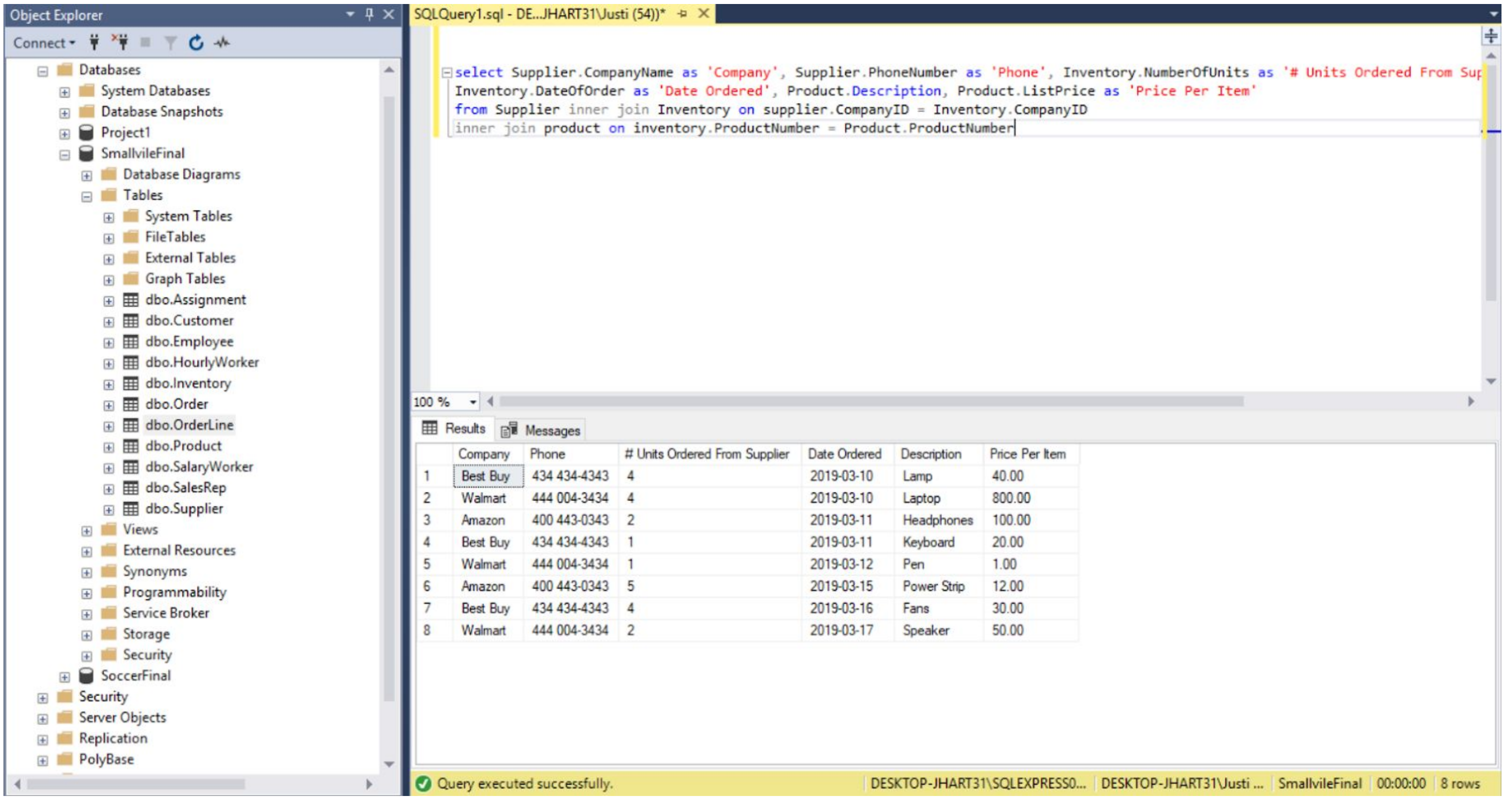
```
dbo.SalesRep
    Columns
        EmployeeID (PK, FK, int, not null)
        SaleNumber (PK, FK, int, not null)
        CommissionRate (decimal(12,3), null)
        SalesUnits (decimal(12,3), null)
        BasePay (Computed, decimal(25,6), null)
    Keys
        CPK_SalesRepEmployeeIDSalesNumber
        FK_SalesRepEmployee
        FK_SalesRepOrderNumber
    Constraints
    Triggers
    Indexes
        CPK_SalesRepEmployeeIDSalesNumber (Unique, Non-Clustered)
    Statistics
```

The Salary table tracks the amount made by a salaried employee and links back to the Employee table



```
dbo.SalaryWorker
    Columns
        EmployeeID (FK, int, not null)
        Salary (int, not null)
    Keys
        FK_SalaryWorkerEmployeeID
    Constraints
        CK_SalaryWorkerSalary
    Triggers
    Indexes
    Statistics
dbo.SalesRep
```

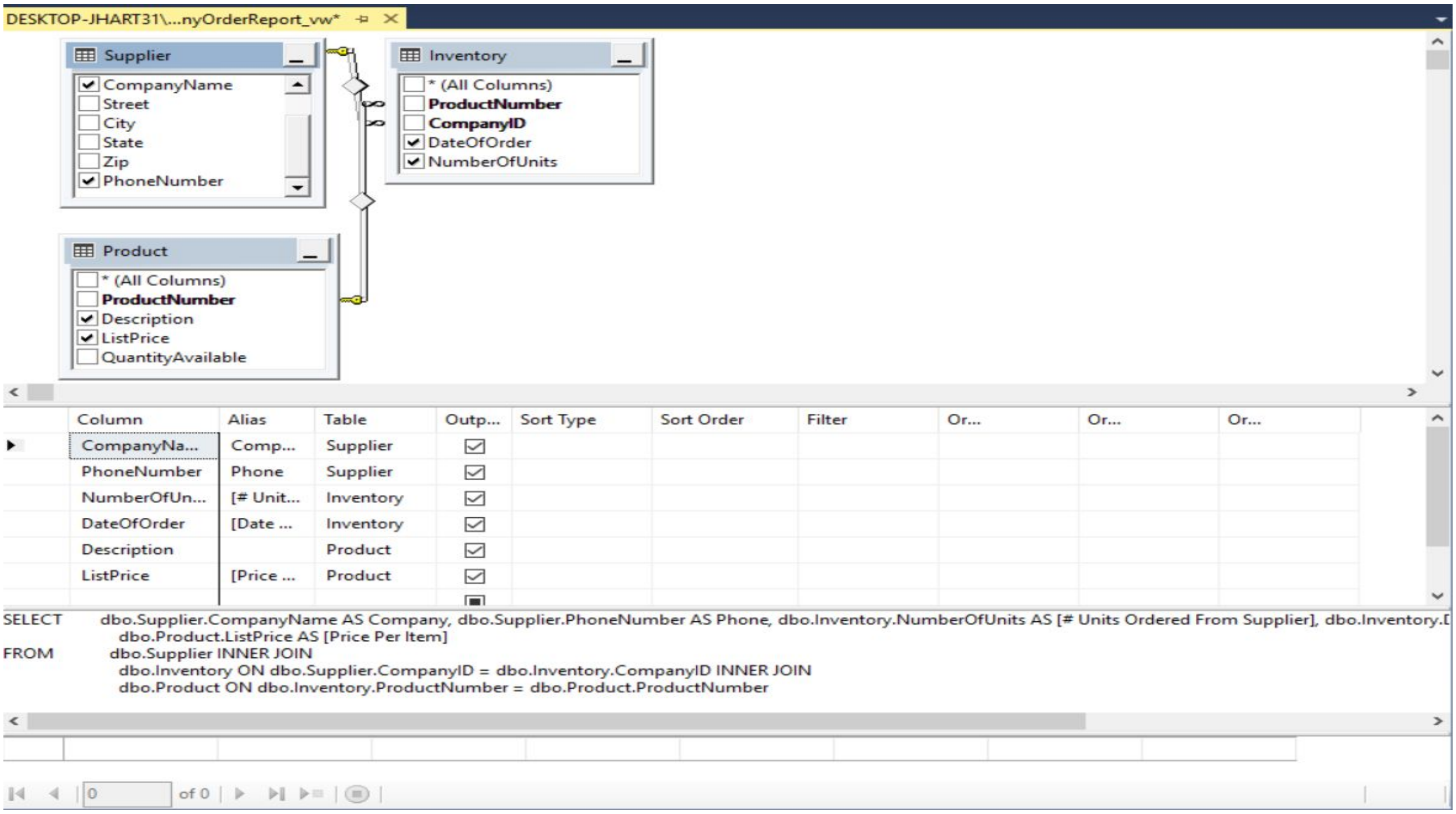
Table_VW



```sql
select Supplier.CompanyName as 'Company', Supplier.PhoneNumber as 'Phone', Inventory.NumberOfUnits as '# Units Ordered From Sup
Inventory.DateOfOrder as 'Date Ordered', Product.Description, Product.ListPrice as 'Price Per Item'
from Supplier inner join Inventory on supplier.CompanyID = Inventory.CompanyID
inner join product on inventory.ProductNumber = Product.ProductNumber
```

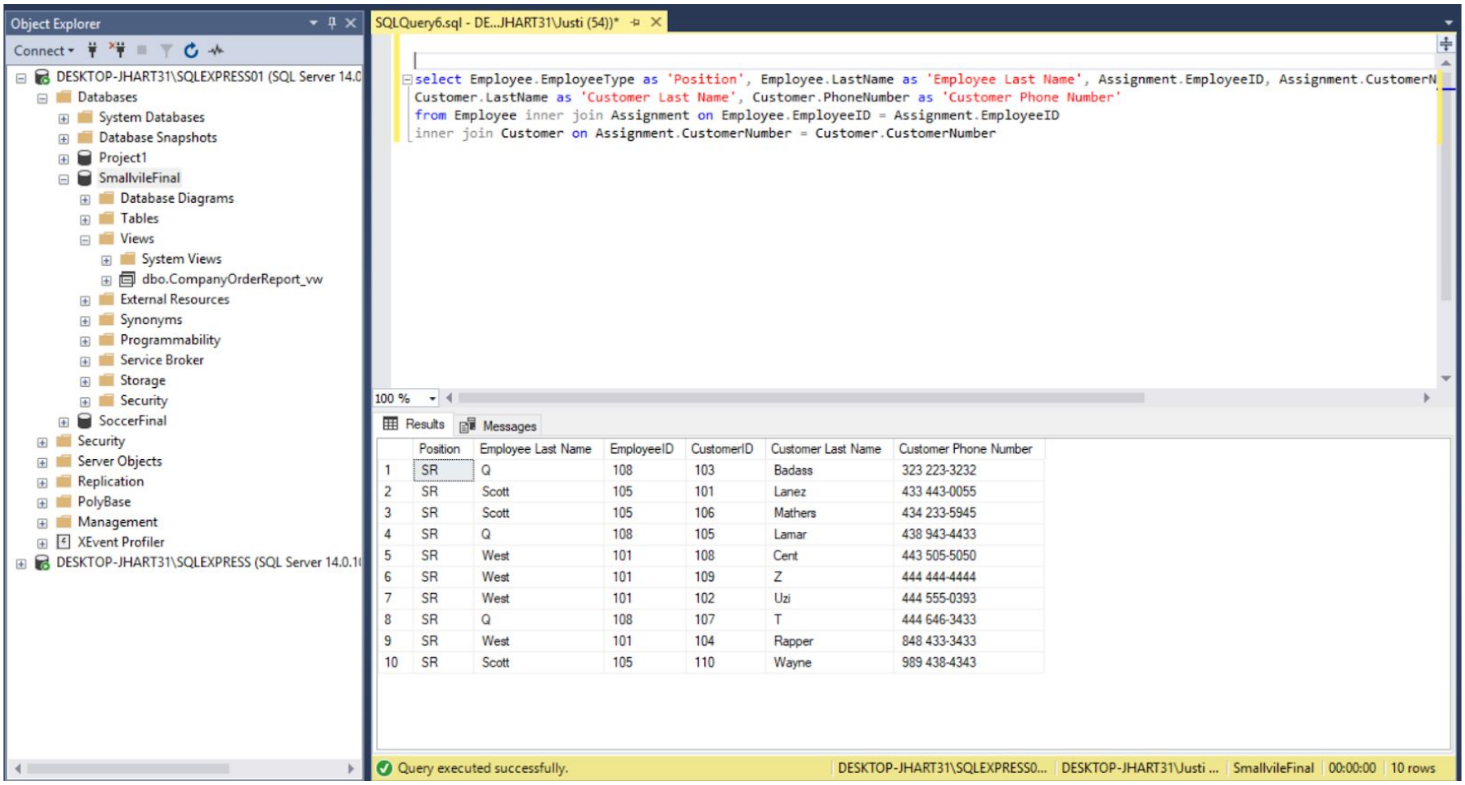
	Company	Phone	# Units Ordered From Supplier	Date Ordered	Description	Price Per Item
1	Best Buy	434 434-4343	4	2019-03-10	Lamp	40.00
2	Walmart	444 004-3434	4	2019-03-10	Laptop	800.00
3	Amazon	400 443-0343	2	2019-03-11	Headphones	100.00
4	Best Buy	434 434-4343	1	2019-03-11	Keyboard	20.00
5	Walmart	444 004-3434	1	2019-03-12	Pen	1.00
6	Amazon	400 443-0343	5	2019-03-15	Power Strip	12.00
7	Best Buy	434 434-4343	4	2019-03-16	Fans	30.00
8	Walmart	444 004-3434	2	2019-03-17	Speaker	50.00

Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS0... | DESKTOP-JHART31\Justi ... | SmallvileFinal | 00:00:00 | 8 rows



Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
CompanyNa...	Comp...	Supplier	☑						
PhoneNumber	Phone	Supplier	☑						
NumberOfUn...	[# Unit...	Inventory	☑						
DateOfOrder	[Date ...	Inventory	☑						
Description		Product	☑						
ListPrice	[Price ...	Product	☑						

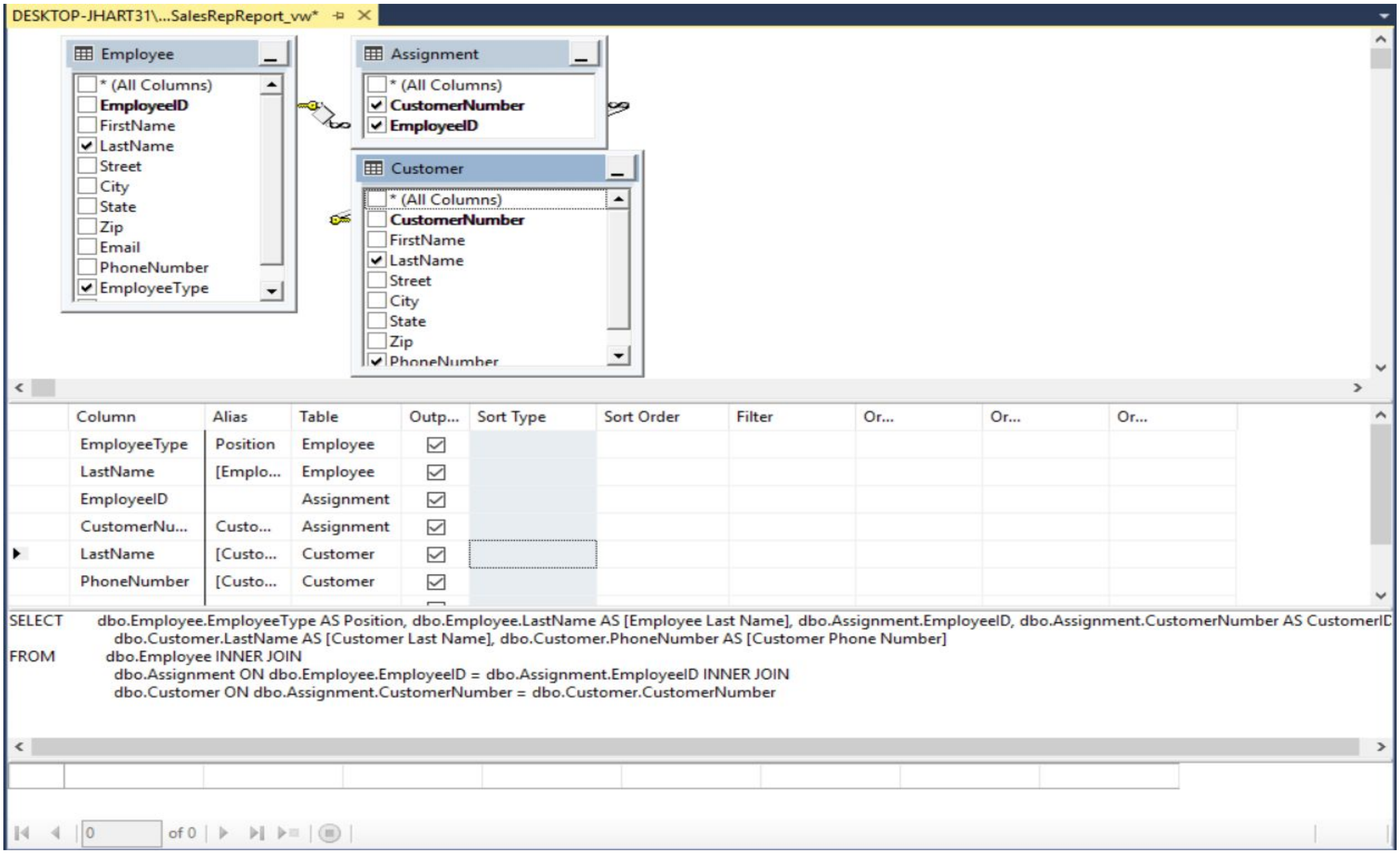
```sql
SELECT     dbo.Supplier.CompanyName AS Company, dbo.Supplier.PhoneNumber AS Phone, dbo.Inventory.NumberOfUnits AS [# Units Ordered From Supplier], dbo.Inventory.[
             dbo.Product.ListPrice AS [Price Per Item]
FROM       dbo.Supplier INNER JOIN
             dbo.Inventory ON dbo.Supplier.CompanyID = dbo.Inventory.CompanyID INNER JOIN
             dbo.Product ON dbo.Inventory.ProductNumber = dbo.Product.ProductNumber
```



```
select Employee.EmployeeType as 'Position', Employee.LastName as 'Employee Last Name', Assignment.EmployeeID, Assignment.CustomerN
Customer.LastName as 'Customer Last Name', Customer.PhoneNumber as 'Customer Phone Number'
from Employee inner join Assignment on Employee.EmployeeID = Assignment.EmployeeID
inner join Customer on Assignment.CustomerNumber = Customer.CustomerNumber
```

	Position	Employee Last Name	EmployeeID	CustomerID	Customer Last Name	Customer Phone Number
1	SR	Q	108	103	Badass	323 223-3232
2	SR	Scott	105	101	Lanez	433 443-0055
3	SR	Scott	105	106	Mathers	434 233-5945
4	SR	Q	108	105	Lamar	438 943-4433
5	SR	West	101	108	Cent	443 505-5050
6	SR	West	101	109	Z	444 444-4444
7	SR	West	101	102	Uzi	444 555-0393
8	SR	Q	108	107	T	444 646-3433
9	SR	West	101	104	Rapper	848 433-3433
10	SR	Scott	105	110	Wayne	989 438-4343

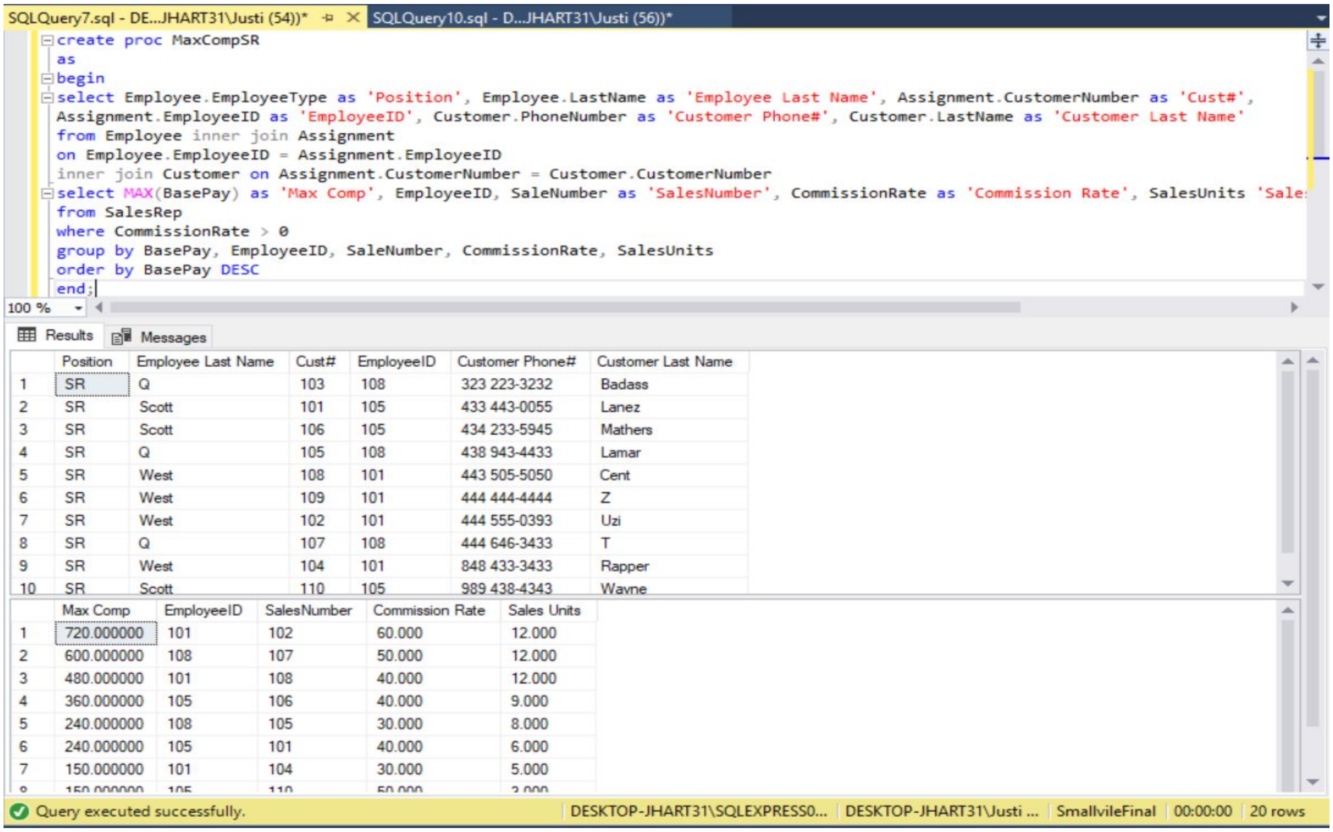
Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS0... | DESKTOP-JHART31\Justi ... | SmallvileFinal | 00:00:00 | 10 rows



Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
EmployeeType	Position	Employee	☑						
LastName	[Emplo...	Employee	☑						
EmployeeID		Assignment	☑						
CustomerNu...	Custo...	Assignment	☑						
LastName	[Custo...	Customer	☑						
PhoneNumber	[Custo...	Customer	☑						

```
SELECT     dbo.Employee.EmployeeType AS Position, dbo.Employee.LastName AS [Employee Last Name], dbo.Assignment.EmployeeID, dbo.Assignment.CustomerNumber AS CustomerID
             dbo.Customer.LastName AS [Customer Last Name], dbo.Customer.PhoneNumber AS [Customer Phone Number]
FROM        dbo.Employee INNER JOIN
             dbo.Assignment ON dbo.Employee.EmployeeID = dbo.Assignment.EmployeeID INNER JOIN
             dbo.Customer ON dbo.Assignment.CustomerNumber = dbo.Customer.CustomerNumber
```

SPROCS

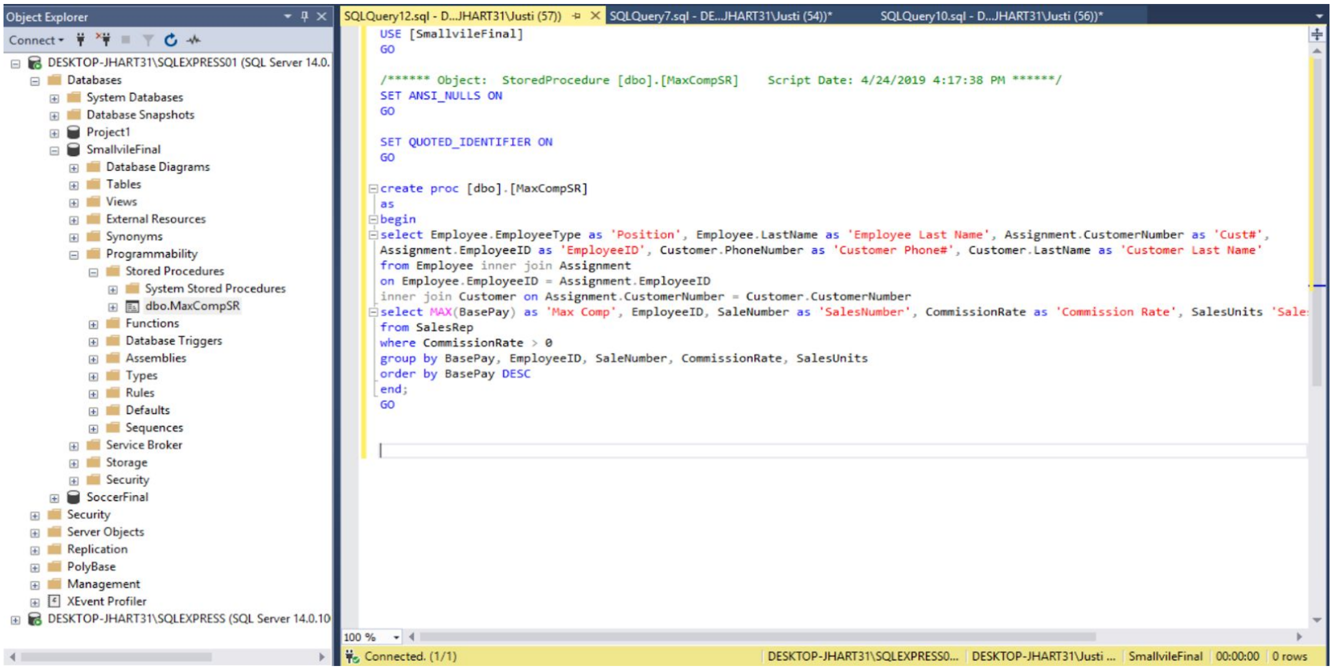

SQLQuery7.sql - DE...JHART31\Justi (54))*
SQLQuery10.sql - D...JHART31\Justi (56))*
100 %
Results
Messages

```sql
create proc MaxCompSR
as
begin
select Employee.EmployeeType as 'Position', Employee.LastName as 'Employee Last Name', Assignment.CustomerNumber as 'Cust#',
Assignment.EmployeeID as 'EmployeeID', Customer.PhoneNumber as 'Customer Phone#', Customer.LastName as 'Customer Last Name'
from Employee inner join Assignment
on Employee.EmployeeID = Assignment.EmployeeID
inner join Customer on Assignment.CustomerNumber = Customer.CustomerNumber
select MAX(BasePay) as 'Max Comp', EmployeeID, SaleNumber as 'SalesNumber', CommissionRate as 'Commission Rate', SalesUnits 'Sale
from SalesRep
where CommissionRate > 0
group by BasePay, EmployeeID, SaleNumber, CommissionRate, SalesUnits
order by BasePay DESC
end;
```

	Position	Employee Last Name	Cust#	EmployeeID	Customer Phone#	Customer Last Name
1	SR	Q	103	108	323 223-3232	Badass
2	SR	Scott	101	105	433 443-0055	Lanez
3	SR	Scott	106	105	434 233-5945	Mathers
4	SR	Q	105	108	438 943-4433	Lamar
5	SR	West	108	101	443 505-5050	Cent
6	SR	West	109	101	444 444-4444	Z
7	SR	West	102	101	444 555-0393	Uzi
8	SR	Q	107	108	444 646-3433	T
9	SR	West	104	101	848 433-3433	Rapper
10	SR	Scott	110	105	989 438-4343	Wayne

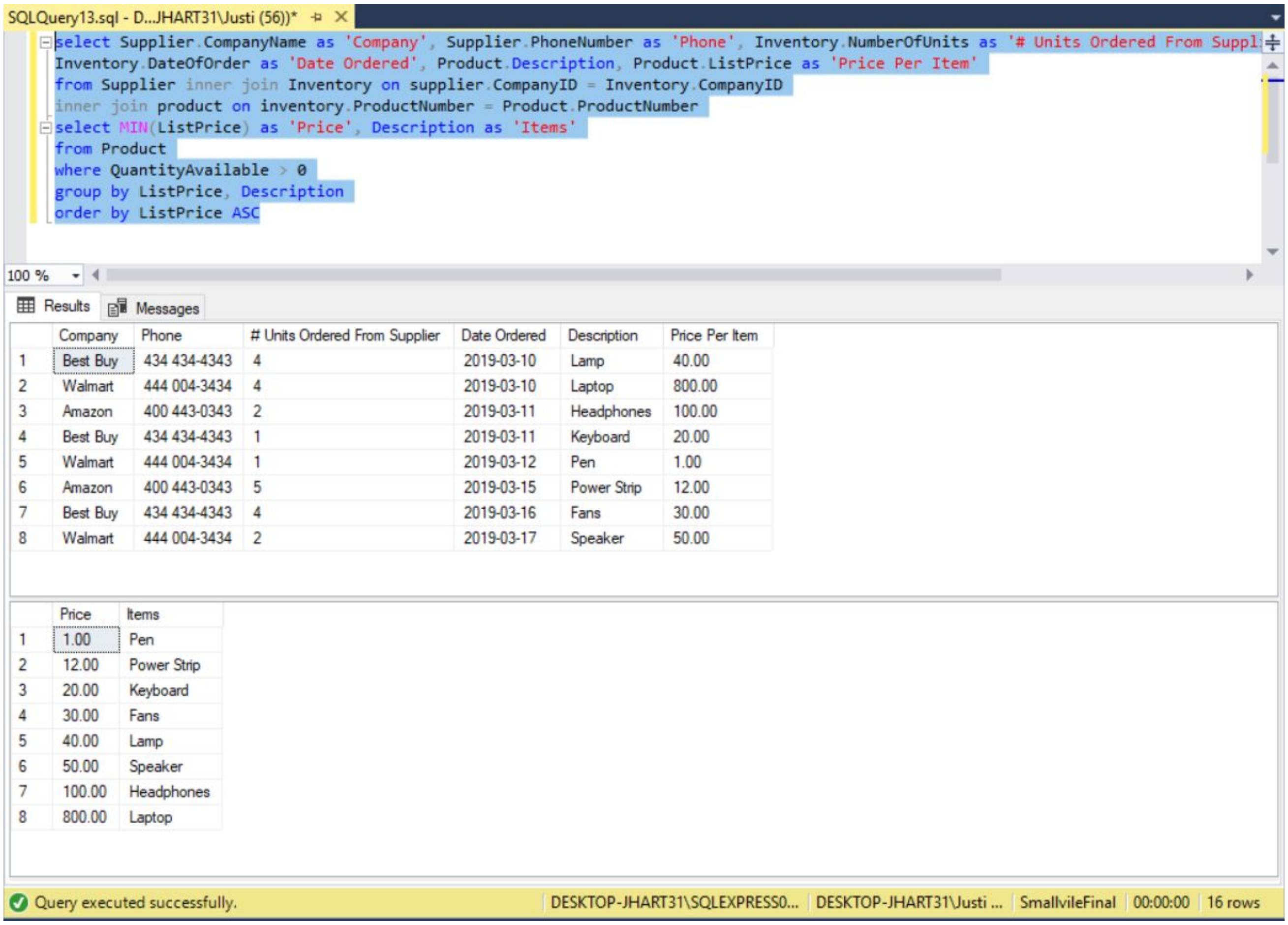
	Max Comp	EmployeeID	SalesNumber	Commission Rate	Sales Units
1	720.000000	101	102	60.000	12.000
2	600.000000	108	107	50.000	12.000
3	480.000000	101	108	40.000	12.000
4	360.000000	105	106	40.000	9.000
5	240.000000	108	105	30.000	8.000
6	240.000000	105	101	40.000	6.000
7	150.000000	101	104	30.000	5.000
8	150.000000	105	110	50.000	2.000

Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS0... | DESKTOP-JHART31\Justi ... | SmallvileFinal | 00:00:00 | 20 rows



```sql
USE [SmallvileFinal]
GO

/****** Object:  StoredProcedure [dbo].[MaxCompSR]    Script Date: 4/24/2019 4:17:38 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

create proc [dbo].[MaxCompSR]
as
begin
select Employee.EmployeeType as 'Position', Employee.LastName as 'Employee Last Name', Assignment.CustomerNumber as 'Cust#',
Assignment.EmployeeID as 'EmployeeID', Customer.PhoneNumber as 'Customer Phone#', Customer.LastName as 'Customer Last Name'
from Employee inner join Assignment
on Employee.EmployeeID = Assignment.EmployeeID
inner join Customer on Assignment.CustomerNumber = Customer.CustomerNumber
select MAX(BasePay) as 'Max Comp', EmployeeID, SaleNumber as 'SalesNumber', CommissionRate as 'Commission Rate', SalesUnits 'Sale
from SalesRep
where CommissionRate > 0
group by BasePay, EmployeeID, SaleNumber, CommissionRate, SalesUnits
order by BasePay DESC
end;
GO
```

Connected. (1/1) | DESKTOP-JHART31\SQLEXPRESS0... | DESKTOP-JHART31\Justi ... | SmallvileFinal | 00:00:00 | 0 rows

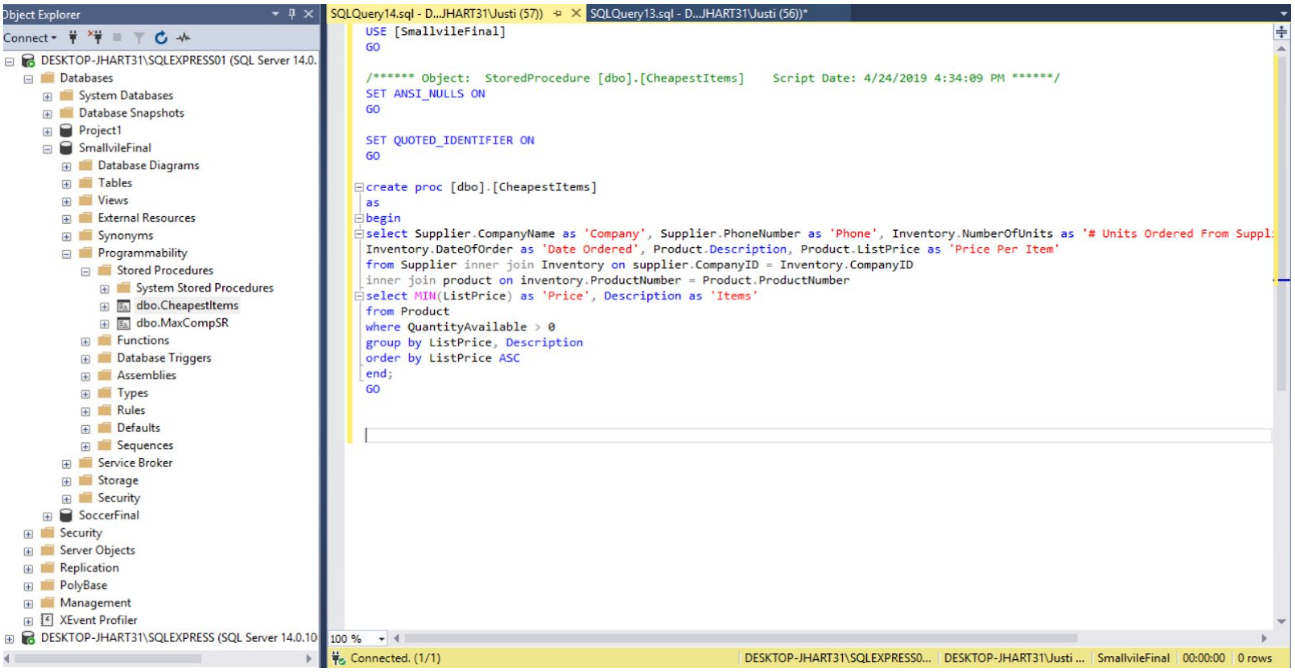


```
select Supplier.CompanyName as 'Company', Supplier.PhoneNumber as 'Phone', Inventory.NumberOfUnits as '# Units Ordered From Suppl
Inventory.DateOfOrder as 'Date Ordered', Product.Description, Product.ListPrice as 'Price Per Item'
from Supplier inner join Inventory on supplier.CompanyID = Inventory.CompanyID
inner join product on inventory.ProductNumber = Product.ProductNumber
select MIN(ListPrice) as 'Price', Description as 'Items'
from Product
where QuantityAvailable > 0
group by ListPrice, Description
order by ListPrice ASC
```

	Company	Phone	# Units Ordered From Supplier	Date Ordered	Description	Price Per Item
1	Best Buy	434 434-4343	4	2019-03-10	Lamp	40.00
2	Walmart	444 004-3434	4	2019-03-10	Laptop	800.00
3	Amazon	400 443-0343	2	2019-03-11	Headphones	100.00
4	Best Buy	434 434-4343	1	2019-03-11	Keyboard	20.00
5	Walmart	444 004-3434	1	2019-03-12	Pen	1.00
6	Amazon	400 443-0343	5	2019-03-15	Power Strip	12.00
7	Best Buy	434 434-4343	4	2019-03-16	Fans	30.00
8	Walmart	444 004-3434	2	2019-03-17	Speaker	50.00

	Price	Items
1	1.00	Pen
2	12.00	Power Strip
3	20.00	Keyboard
4	30.00	Fans
5	40.00	Lamp
6	50.00	Speaker
7	100.00	Headphones
8	800.00	Laptop

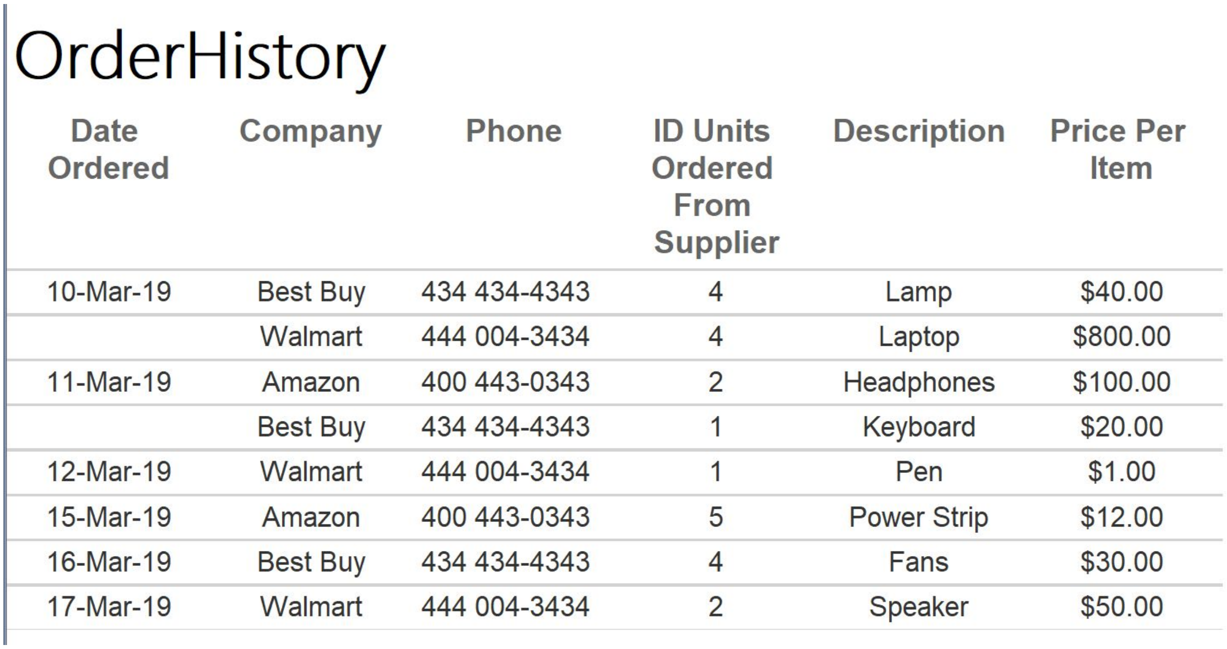
Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS0... | DESKTOP-JHART31\Justi ... | SmallvileFinal | 00:00:00 | 16 rows



```
USE [SmallvileFinal]
GO

/****** Object:  StoredProcedure [dbo].[CheapestItems]    Script Date: 4/24/2019 4:34:09 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

create proc [dbo].[CheapestItems]
as
begin
select Supplier.CompanyName as 'Company', Supplier.PhoneNumber as 'Phone', Inventory.NumberOfUnits as '# Units Ordered From Suppl
Inventory.DateOfOrder as 'Date Ordered', Product.Description, Product.ListPrice as 'Price Per Item'
from Supplier inner join Inventory on supplier.CompanyID = Inventory.CompanyID
inner join product on inventory.ProductNumber = Product.ProductNumber
select MIN(ListPrice) as 'Price', Description as 'Items'
from Product
where QuantityAvailable > 0
group by ListPrice, Description
order by ListPrice ASC
end;
GO
```

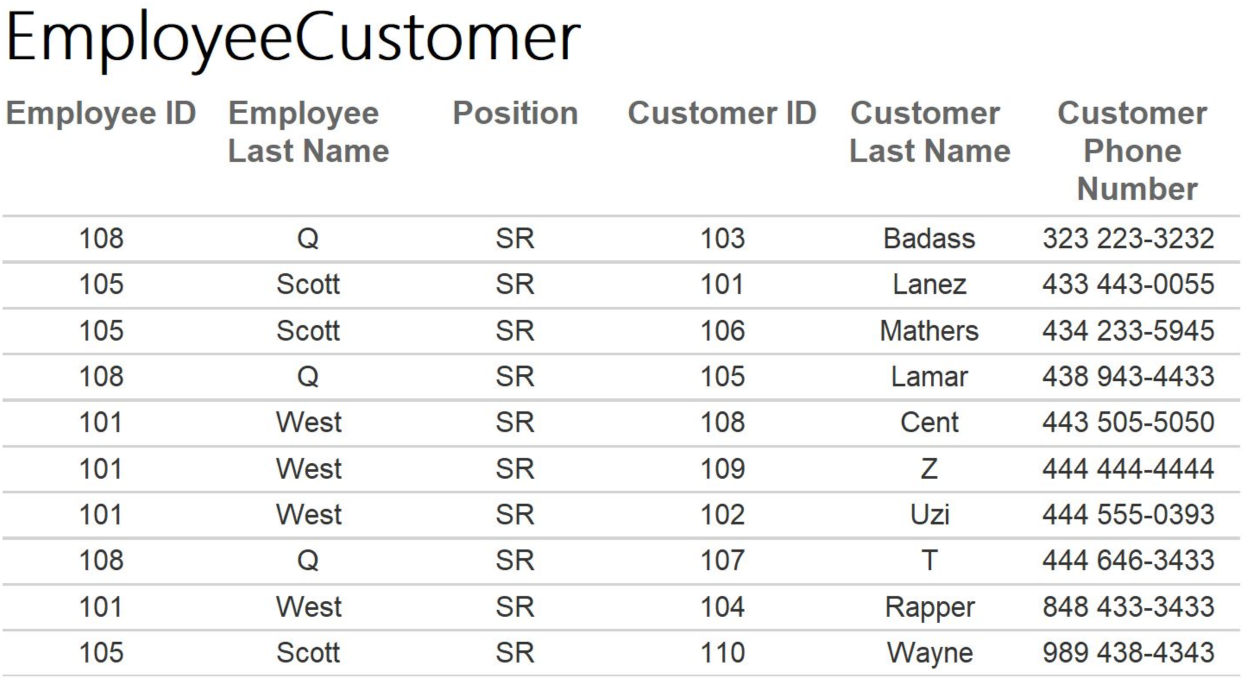
Connected. (1/1) | DESKTOP-JHART31\SQLEXPRESS0... | DESKTOP-JHART31\Justi ... | SmallvileFinal | 00:00:00 | 0 rows

SSRS Reports



OrderHistory

Date Ordered	Company	Phone	ID Units Ordered From Supplier	Description	Price Per Item
10-Mar-19	Best Buy	434 434-4343	4	Lamp	$40.00
	Walmart	444 004-3434	4	Laptop	$800.00
11-Mar-19	Amazon	400 443-0343	2	Headphones	$100.00
	Best Buy	434 434-4343	1	Keyboard	$20.00
12-Mar-19	Walmart	444 004-3434	1	Pen	$1.00
15-Mar-19	Amazon	400 443-0343	5	Power Strip	$12.00
16-Mar-19	Best Buy	434 434-4343	4	Fans	$30.00
17-Mar-19	Walmart	444 004-3434	2	Speaker	$50.00



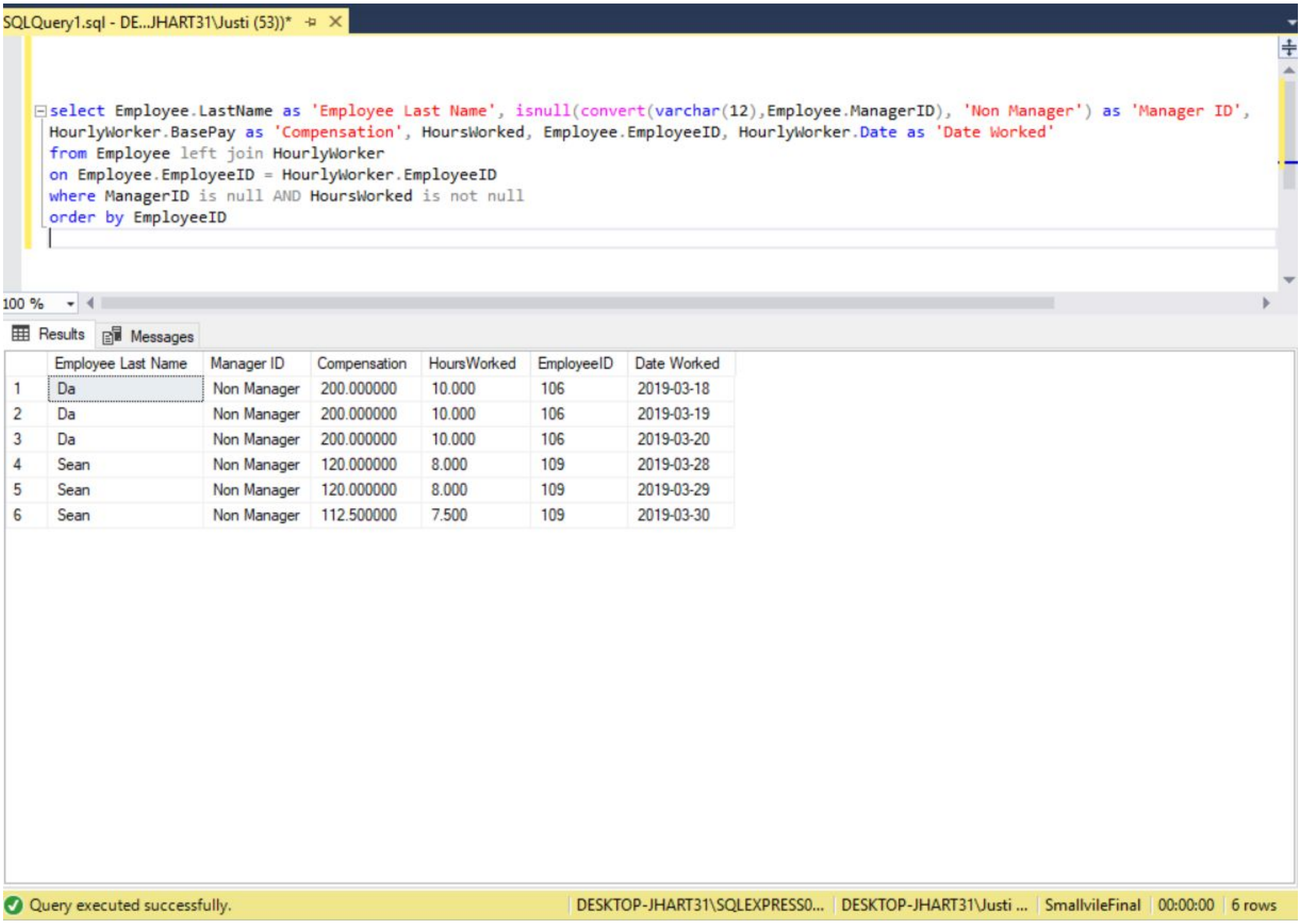
EmployeeCustomer

Employee ID	Employee Last Name	Position	Customer ID	Customer Last Name	Customer Phone Number
108	Q	SR	103	Badass	323 223-3232
105	Scott	SR	101	Lanez	433 443-0055
105	Scott	SR	106	Mathers	434 233-5945
108	Q	SR	105	Lamar	438 943-4433
101	West	SR	108	Cent	443 505-5050
101	West	SR	109	Z	444 444-4444
101	West	SR	102	Uzi	444 555-0393
108	Q	SR	107	T	444 646-3433
101	West	SR	104	Rapper	848 433-3433
105	Scott	SR	110	Wayne	989 438-4343

User Acceptance Query Tests

Question: Show me all hourly workers who are not managers and sort by EmployeeID. I also want to know their compensation, hours worked, and dates worked. Replace ManagerID with ‘non manager’.

Answer:



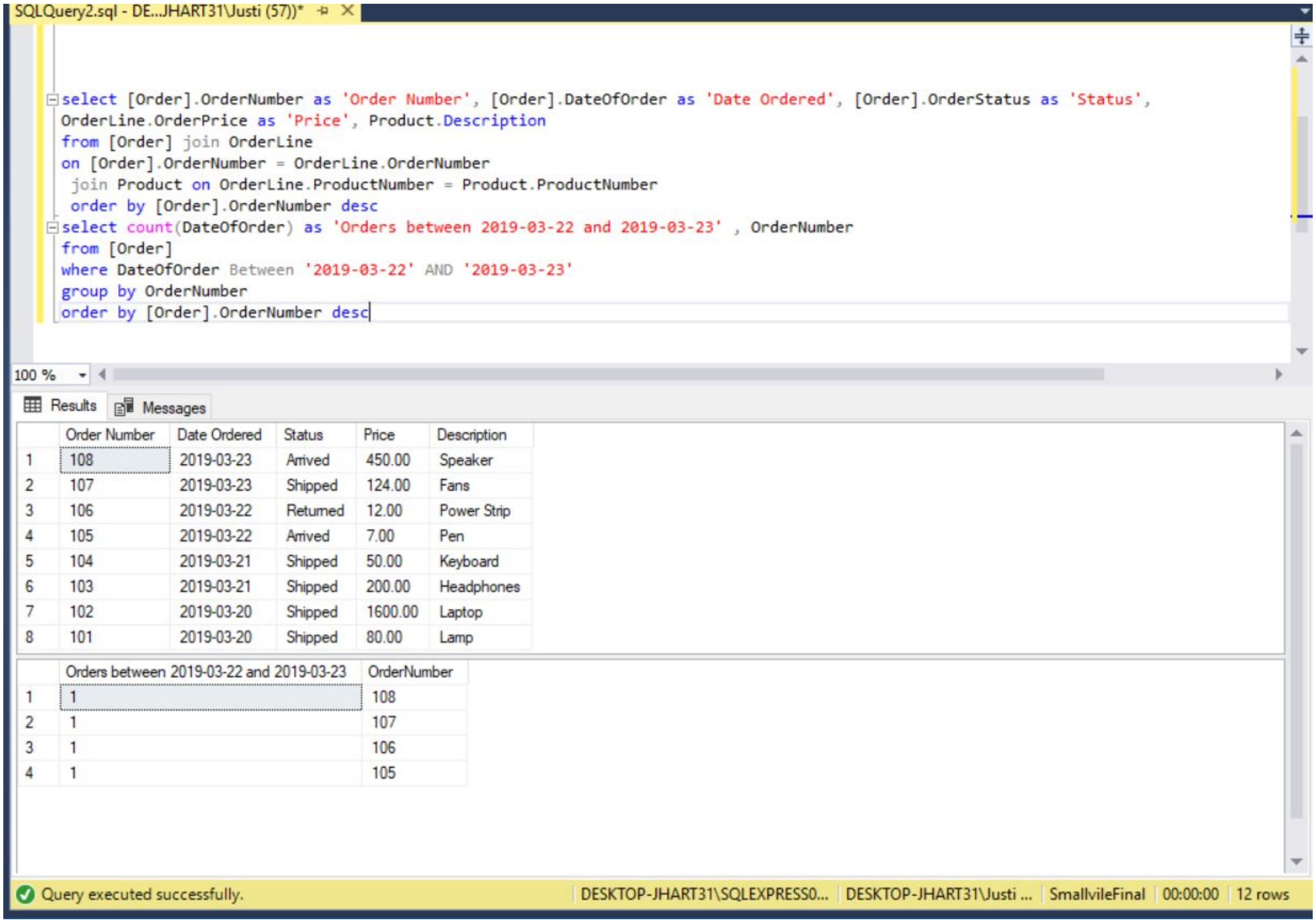
```
select Employee.LastName as 'Employee Last Name', isnull(convert(varchar(12),Employee.ManagerID), 'Non Manager') as 'Manager ID',
HourlyWorker.BasePay as 'Compensation', HoursWorked, Employee.EmployeeID, HourlyWorker.Date as 'Date Worked'
from Employee left join HourlyWorker
on Employee.EmployeeID = HourlyWorker.EmployeeID
where ManagerID is null AND HoursWorked is not null
order by EmployeeID
```

	Employee Last Name	Manager ID	Compensation	HoursWorked	EmployeeID	Date Worked
1	Da	Non Manager	200.000000	10.000	106	2019-03-18
2	Da	Non Manager	200.000000	10.000	106	2019-03-19
3	Da	Non Manager	200.000000	10.000	106	2019-03-20
4	Sean	Non Manager	120.000000	8.000	109	2019-03-28
5	Sean	Non Manager	120.000000	8.000	109	2019-03-29
6	Sean	Non Manager	112.500000	7.500	109	2019-03-30

Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS0... | DESKTOP-JHART31\Justi ... | SmallvileFinal | 00:00:00 | 6 rows

Question: Show me all order numbers, date of orders, status, price and description. I then want a count of how many orders have been placed through 2019-03-22 and 2019-03-23, with the associated order number.

Answer:



```sql
select [Order].OrderNumber as 'Order Number', [Order].DateOfOrder as 'Date Ordered', [Order].OrderStatus as 'Status',
OrderLine.OrderPrice as 'Price', Product.Description
from [Order] join OrderLine
on [Order].OrderNumber = OrderLine.OrderNumber
 join Product on OrderLine.ProductNumber = Product.ProductNumber
 order by [Order].OrderNumber desc
select count(DateOfOrder) as 'Orders between 2019-03-22 and 2019-03-23' , OrderNumber
from [Order]
where DateOfOrder Between '2019-03-22' AND '2019-03-23'
group by OrderNumber
order by [Order].OrderNumber desc
```

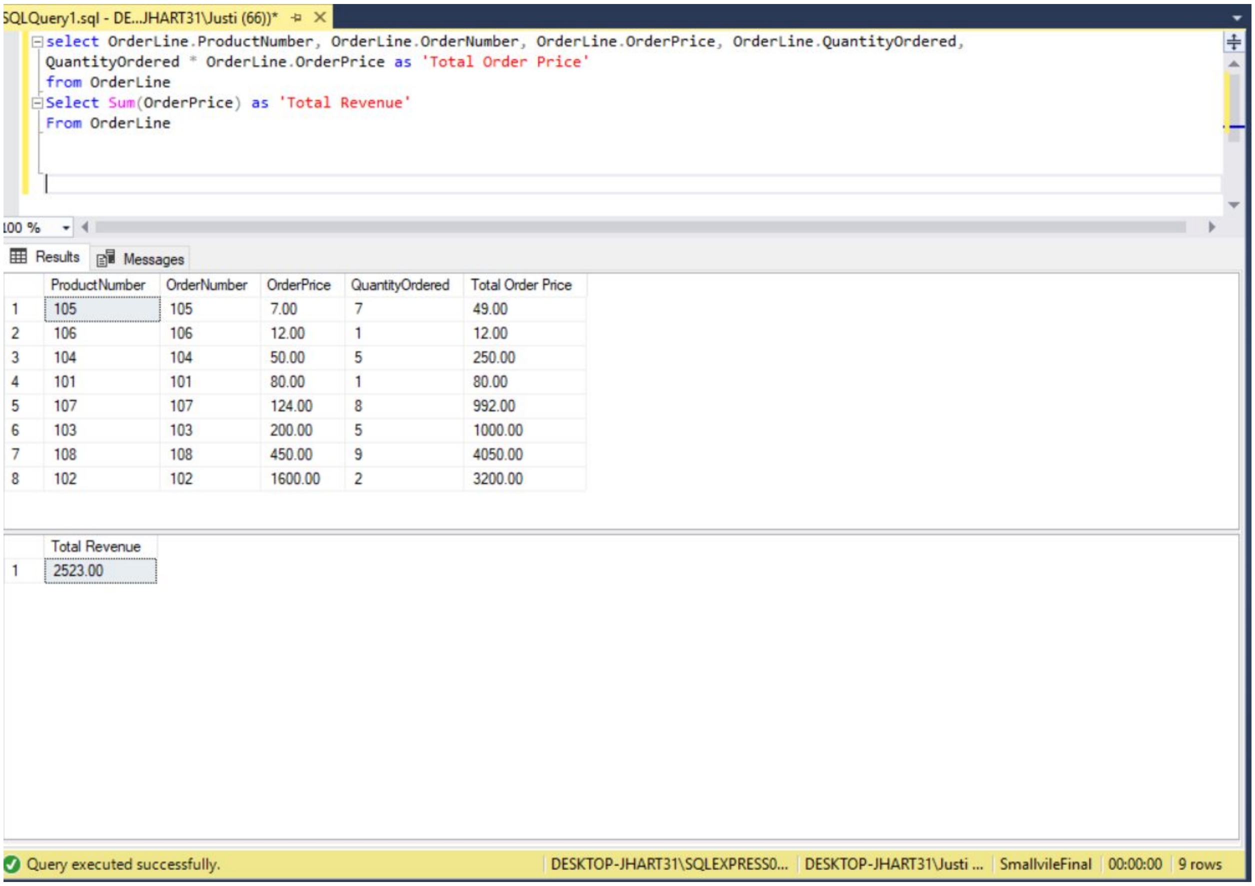
	Order Number	Date Ordered	Status	Price	Description
1	108	2019-03-23	Arrived	450.00	Speaker
2	107	2019-03-23	Shipped	124.00	Fans
3	106	2019-03-22	Returned	12.00	Power Strip
4	105	2019-03-22	Arrived	7.00	Pen
5	104	2019-03-21	Shipped	50.00	Keyboard
6	103	2019-03-21	Shipped	200.00	Headphones
7	102	2019-03-20	Shipped	1600.00	Laptop
8	101	2019-03-20	Shipped	80.00	Lamp

	Orders between 2019-03-22 and 2019-03-23	OrderNumber
1	1	108
2	1	107
3	1	106
4	1	105

Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS0... | DESKTOP-JHART31\Justi ... | SmallvileFinal | 00:00:00 | 12 rows

Question: Show me all products numbers, order numbers, order price, and the quantity ordered. From there I was to know the total order price. Lastly, show me the total revenue of all orders.

Answer:



```
select OrderLine.ProductNumber, OrderLine.OrderNumber, OrderLine.OrderPrice, OrderLine.QuantityOrdered,
QuantityOrdered * OrderLine.OrderPrice as 'Total Order Price'
from OrderLine
Select Sum(OrderPrice) as 'Total Revenue'
From OrderLine
```

	ProductNumber	OrderNumber	OrderPrice	QuantityOrdered	Total Order Price
1	105	105	7.00	7	49.00
2	106	106	12.00	1	12.00
3	104	104	50.00	5	250.00
4	101	101	80.00	1	80.00
5	107	107	124.00	8	992.00
6	103	103	200.00	5	1000.00
7	108	108	450.00	9	4050.00
8	102	102	1600.00	2	3200.00

	Total Revenue
1	2523.00

Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS0... | DESKTOP-JHART31\Justi ... | SmallvileFinal | 00:00:00 | 9 rows