Assignment #4: Smallville

Data Cowboys

ISM4212.001S19

April 26th, 2019

Team Leader:

Blake Rudder

Team Members:

Thomas McLaughlin

Dylan Stewart

Brandon Wolfram

Purpose:

This document is intended to be used as a reference for creating a database for Smallville company. This document details the entities and attributes that are to be tracked in the database. The document's intention is to illustrate user stories and then generate user reports and test queries from the database design described.

Narrative:

Smallville has a need to develop a database to move away from Excel spreadsheets and to a full-fledged database system. The business has three main entities that are of interest to them. These entities are customers, products, and orders. Customers need to be tracked by their customer id, first name, last name, address, and phone. Products are tracked by product id, description, list price, and inventory/product on hand. A customer usually places an order and this order is tracked by an order number, date of the order, and status. The order must have one or more products. Products can be in more than one order. Not every product is part of an order. Products are related to orders by quantity ordered, quantity supplied, and order price. For compensation purposes, employees are categorized as either sales reps, hourly workers, or salary workers. Employees are tracked by first name, last name, address, email, and phone number. Each customer is assigned a sales rep, and each sales rep is assigned at least one customer. For inventory purposes, suppliers are tracked by company name, address, and phone. Products may have more than one supplier and suppliers may offer more than one product.

Requirements (actors, roles):

Customers- Customers place orders on products. Customers are tracked by Customer ID, name, address, and phone number.

Orders - Orders of products are placed by customers. Orders are tracked by Order ID, date of the order, and order status.

Products- Products are ordered by customers. Products are tracked by Product ID, description, list price, and the quantity on hand.

Employees- Employees fulfill orders. Employees are tracked by Employee ID, name, email, phone, address, and salary.

Suppliers- Suppliers provide products. Suppliers are tracked by their Company Name, address, and phone.

Entities Identified to be Tracked:

Customer

Order

Product

Employee

Supplier

Entities with Attributes Nested:

Customer

- Customer ID
- Customer Name (First, Last)
- Customer Address
- Customer Phone

Order

- Order ID
- Date of Order
- Order Status

Product

- Product ID
- Product Description
- List Price
- Quantity on Hand

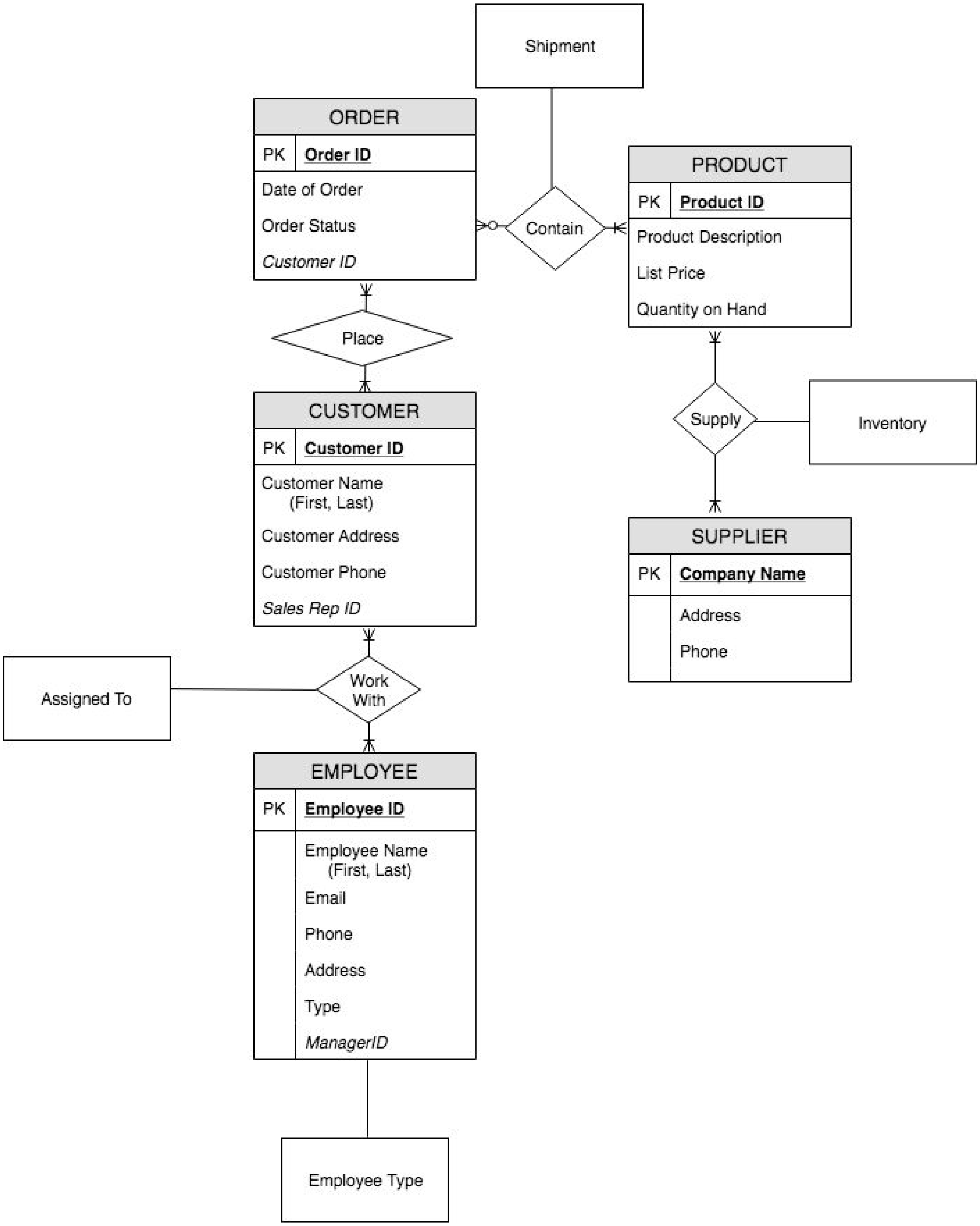
Employee

- Employee ID
- Employee Name (First, Last)
- Email
- Phone
- Address
- Salary

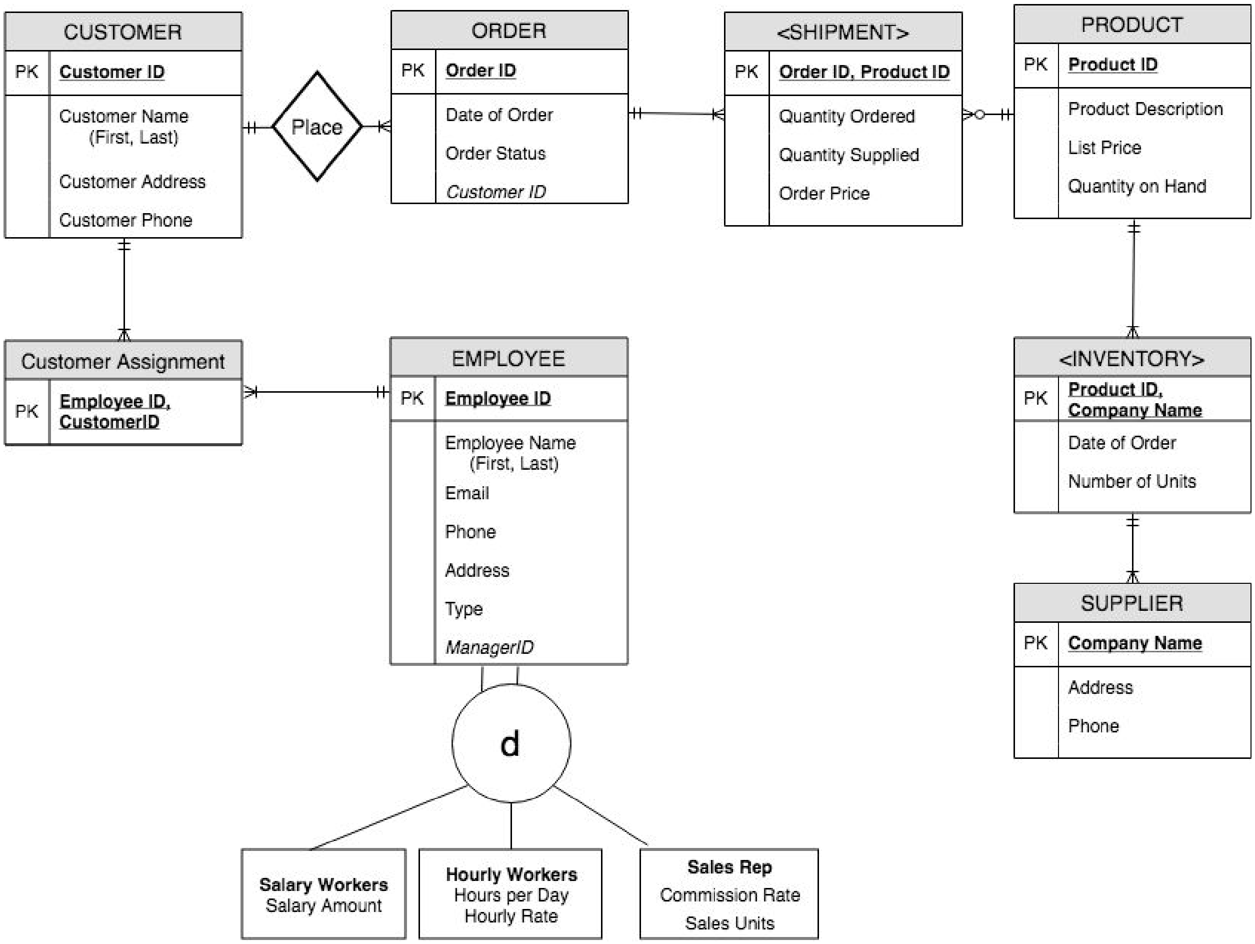
Supplier

- Company Name
- Address
- Phone

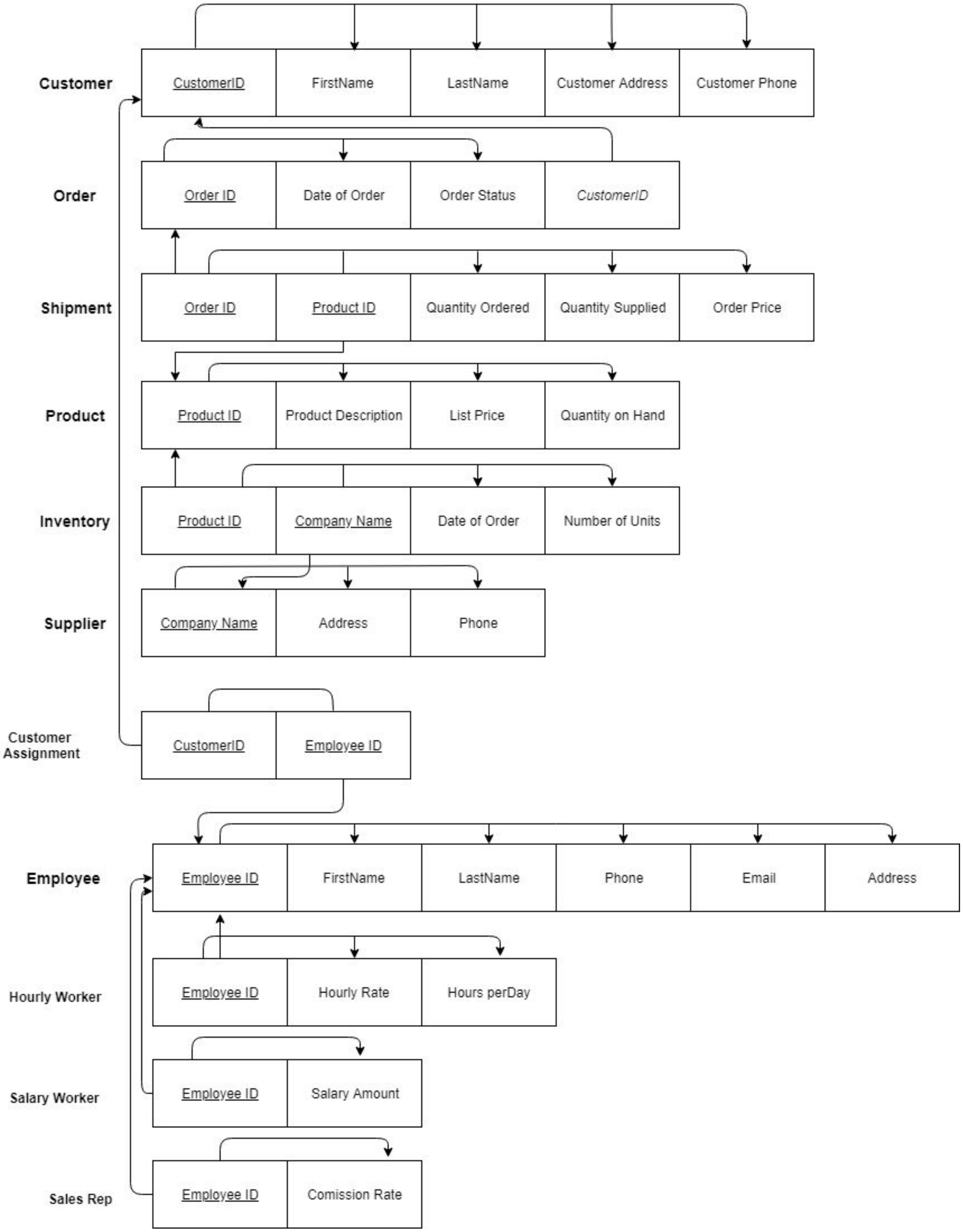
ERD:


Shipment
ORDER
PK
Order ID
Date of Order
Order Status
Customer ID
Contain
PRODUCT
PK
Product ID
Product Description
List Price
Quantity on Hand
Place
Supply
Inventory
CUSTOMER
PK
Customer ID
Customer Name
(First, Last)
Customer Address
Customer Phone
Sales Rep ID
SUPPLIER
PK
Company Name
Address
Phone
Assigned To
Work
With
EMPLOYEE
PK
Employee ID
Employee Name
(First, Last)
Email
Phone
Address
Type
ManagerID
Employee Type

EERD:


CUSTOMER
PK
Customer ID
Customer Name (First, Last)
Customer Address
Customer Phone
Place
ORDER
PK
Order ID
Date of Order
Order Status
Customer ID
<SHIPMENT>
PK
Order ID, Product ID
Quantity Ordered
Quantity Supplied
Order Price
PRODUCT
PK
Product ID
Product Description
List Price
Quantity on Hand
Customer Assignment
PK
Employee ID, CustomerID
EMPLOYEE
PK
Employee ID
Employee Name (First, Last)
Email
Phone
Address
Type
ManagerID
d
Salary Workers
Salary Amount
Hourly Workers
Hours per Day
Hourly Rate
Sales Rep
Commission Rate
Sales Units
<INVENTORY>
PK
Product ID, Company Name
Date of Order
Number of Units
SUPPLIER
PK
Company Name
Address
Phone

RS:


Customer
CustomerID
FirstName
LastName
Customer Address
Customer Phone
Order
Order ID
Date of Order
Order Status
CustomerID
Shipment
Order ID
Product ID
Quantity Ordered
Quantity Supplied
Order Price
Product
Product ID
Product Description
List Price
Quantity on Hand
Inventory
Product ID
Company Name
Date of Order
Number of Units
Supplier
Company Name
Address
Phone
Customer Assignment
CustomerID
Employee ID
Employee
Employee ID
FirstName
LastName
Phone
Email
Address
Hourly Worker
Employee ID
Hourly Rate
Hours perDay
Salary Worker
Employee ID
Salary Amount
Sales Rep
Employee ID
Comission Rate

DD:

Table: Customer

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
CustomerID	ID assigned to customers	smallint		Y	Y				Y
FirstName	First name of customer	varchar	20						
LastName	Last name of customer	varchar	30						
CustomerAddress	Street address of customer	varchar	50						
CustomerPhone	Phone number of customer	varchar	14				([0-9][0-9][0-9])-[0-9][0-9][0-9][0-9]		
EmployeeID	Sales rep to customer	smallint							

Table: Order

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
Order_ID	ID assigned to order	smallint		Y	Y				Y
Date	Date of order	Date					>= GETDATE()		
Status	Status of order	varchar							
CustomerID	ID assigned to customer	smallint							

Table: Shipment

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
Order_ID	ID assigned to order	smallint			Y				Y
ProductID	ID of product	smallin							
Quantity	Quantity ordered	smallint					>0		
QuantitySupplied	Amount supplied	smallint							
OrderPrice	Price of Order	money							

Table: Product

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
ProductID	ID assigned to product	smallint		Y	Y				Y
ProductDesc	Description of product	varchar	50						
ListPrice	Price of product	smallmoney						Y	
QuantityOnHand	Quantity of product on hand	smallint							

Table: Inventory

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
ProductID	ID assigned to product	smallint			Y				Y
CompanyName	Name of company product is from	varchar	20						
Date	Date of order	date					>= GETDATE()		
Units	Number of units from order	smallint							

Table: Supplier

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
SupplierID	Name of Company	varchar		Y	Y				
Address	Address of company	varchar	50						
CompanyPhone	Phone number to supplier	text	10				([0-9][0-9][0-9])-[0-9][0-9][0-9][0-9]		

Table: Employee

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
EmployeeID	ID assigned to employee	smallint		Y	Y				Y
FirstName	First name of employee	varchar	20						
LastName	Last name of employee	varchar	30						
EmployeeAddress	Street address of employee	varchar	50						
email	Email address of employee	varchar	50					Y	
EmployeePhone	Phone number of employee	varchar	14				([0-9][0-9][0-9])-[0-9][0-9][0-9][0-9]		
Salary	Salary class of employees	smallmoney							

Table: Hourly Worker

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
EmployeeID	ID of employee	smallint			Y				
HourlyRate	Hourly rate of employee	smallmoney							
HoursDay	Hours worked by employee	smallint							

Table: Salary Worker

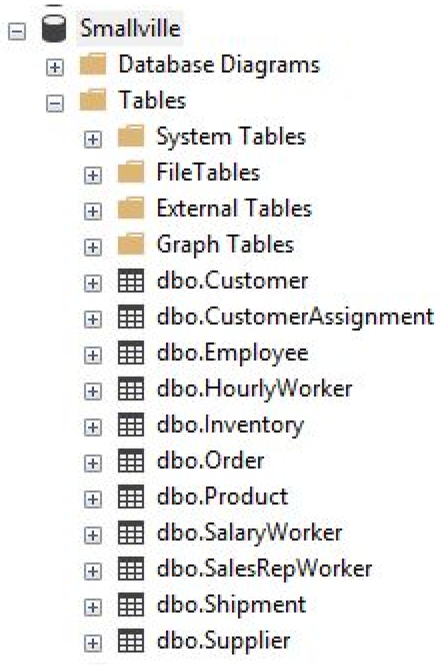
Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
EmployeeID	ID of employee	smallint			Y				
Salary	Salary of employee	money							

Table: Sales Rep

Column Name	Description	Data Type	Size	Identity	Unique	Default	Check	Allow Nulls	Index
EmployeeID	ID of employee	smallint			Y				
CommissionRate	Commission rate of employee	smallmoney							

Tables By Schema:

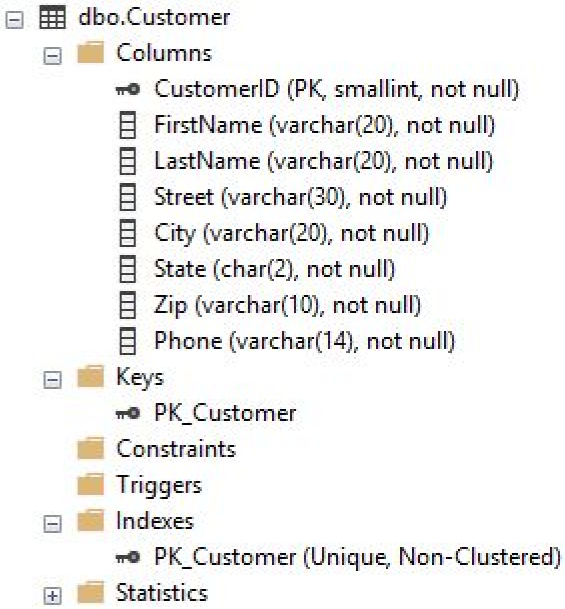
View of all tables created in the database:



- Smallville
  - Database Diagrams
  - Tables
    - System Tables
    - FileTables
    - External Tables
    - Graph Tables
    - dbo.Customer
    - dbo.CustomerAssignment
    - dbo.Employee
    - dbo.HourlyWorker
    - dbo.Inventory
    - dbo.Order
    - dbo.Product
    - dbo.SalaryWorker
    - dbo.SalesRepWorker
    - dbo.Shipment
    - dbo.Supplier

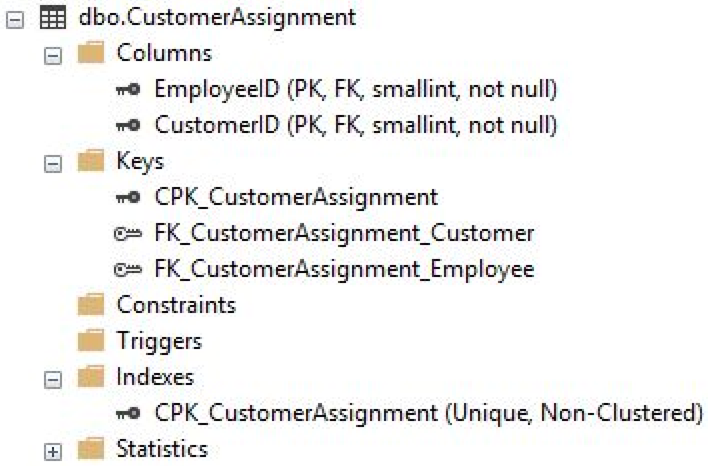
Tables Broken Down Individually:

Customer Table: Customers place orders on products. Customers are tracked by the CustomerID generated in the database. Customers are also tracked by their name, address, and phone number.



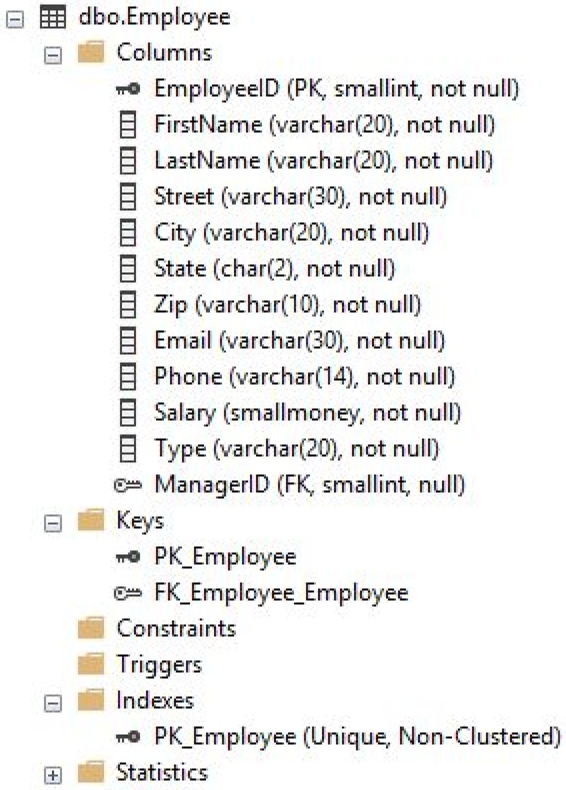
```
dbo.Customer
    Columns
        CustomerID (PK, smallint, not null)
        FirstName (varchar(20), not null)
        LastName (varchar(20), not null)
        Street (varchar(30), not null)
        City (varchar(20), not null)
        State (char(2), not null)
        Zip (varchar(10), not null)
        Phone (varchar(14), not null)
    Keys
        PK_Customer
    Constraints
    Triggers
    Indexes
        PK_Customer (Unique, Non-Clustered)
    Statistics
```

Customer Assignment Table: Since each customer is assigned to a sales representative (employee), this assignment table is used to associate an employee with a customer. The unique primary keys EmployeeID and CustomerID are used to record the assignment of a customer to a sales representative.



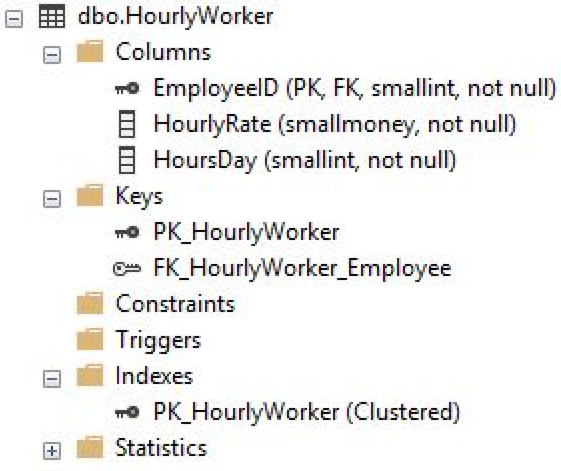
```
dbo.CustomerAssignment
    Columns
        EmployeeID (PK, FK, smallint, not null)
        CustomerID (PK, FK, smallint, not null)
    Keys
        CPK_CustomerAssignment
        FK_CustomerAssignment_Customer
        FK_CustomerAssignment_Employee
    Constraints
    Triggers
    Indexes
        CPK_CustomerAssignment (Unique, Non-Clustered)
    Statistics
```

Employee Table: Employees fulfill orders. Employees are tracked by the EmployeeID generated in the database. Personal information such as name, street address, email, and phone number are also recorded for each employee. Employees are also categorized by their type (sales rep, hourly worker, or salary worker), which is used to determine their salary. Each employee is associated with a manager (another employee) through the foreign key *ManagerID* within the Employee table.



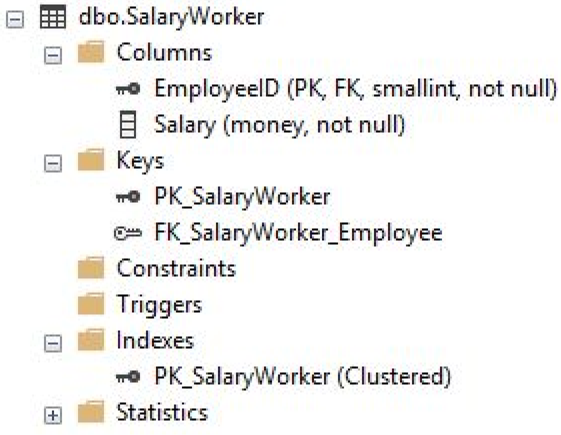
```
⊟ dbo.Employee
    ⊟ Columns
        EmployeeID (PK, smallint, not null)
        FirstName (varchar(20), not null)
        LastName (varchar(20), not null)
        Street (varchar(30), not null)
        City (varchar(20), not null)
        State (char(2), not null)
        Zip (varchar(10), not null)
        Email (varchar(30), not null)
        Phone (varchar(14), not null)
        Salary (smallmoney, not null)
        Type (varchar(20), not null)
        ManagerID (FK, smallint, null)
    ⊟ Keys
        PK_Employee
        FK_Employee_Employee
      Constraints
      Triggers
    ⊟ Indexes
        PK_Employee (Unique, Non-Clustered)
    ⊞ Statistics
```

Hourly Employee Table: For compensation purposes, this table records the hourly rate and daily hours worked for each employee who is considered an hourly worker.



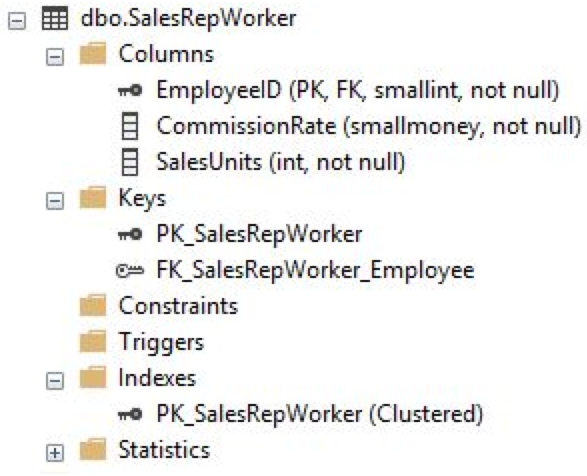
```
dbo.HourlyWorker
    Columns
        EmployeeID (PK, FK, smallint, not null)
        HourlyRate (smallmoney, not null)
        HoursDay (smallint, not null)
    Keys
        PK_HourlyWorker
        FK_HourlyWorker_Employee
    Constraints
    Triggers
    Indexes
        PK_HourlyWorker (Clustered)
    Statistics
```

Salary Employee Table: For compensation purposes, this table records the annual salary for each employee who is considered a salary worker.



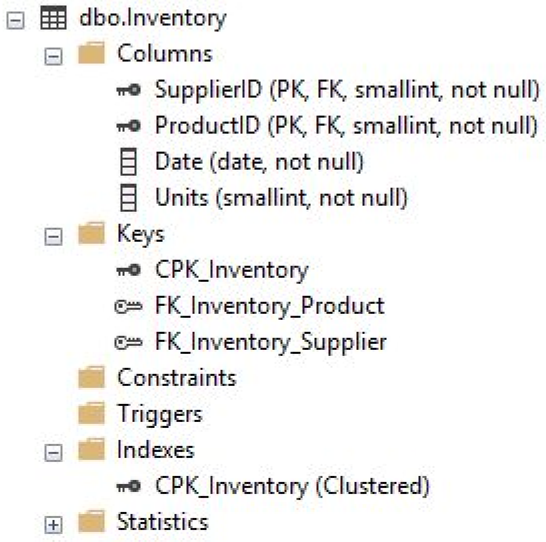
```
dbo.SalaryWorker
    Columns
        EmployeeID (PK, FK, smallint, not null)
        Salary (money, not null)
    Keys
        PK_SalaryWorker
        FK_SalaryWorker_Employee
    Constraints
    Triggers
    Indexes
        PK_SalaryWorker (Clustered)
    Statistics
```

Sales Rep Employee Table: For compensation purposes, this table records the commission rate and units sold for each employee who is considered a sales representative.



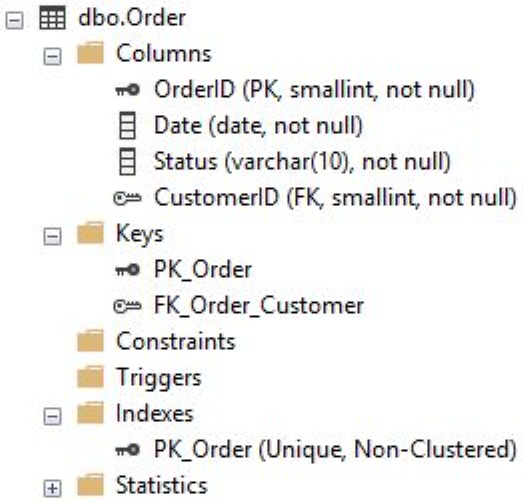
```
⊟ dbo.SalesRepWorker
    ⊟ Columns
          EmployeeID (PK, FK, smallint, not null)
          CommissionRate (smallmoney, not null)
          SalesUnits (int, not null)
    ⊟ Keys
          PK_SalesRepWorker
          FK_SalesRepWorker_Employee
      Constraints
      Triggers
    ⊟ Indexes
          PK_SalesRepWorker (Clustered)
    ⊞ Statistics
```

Inventory Table: Inventory is tracked by the SupplierID and ProductID primary keys generated in the Supplier and Product tables. The date and unit quantity of the product are also recorded for inventory purposes.



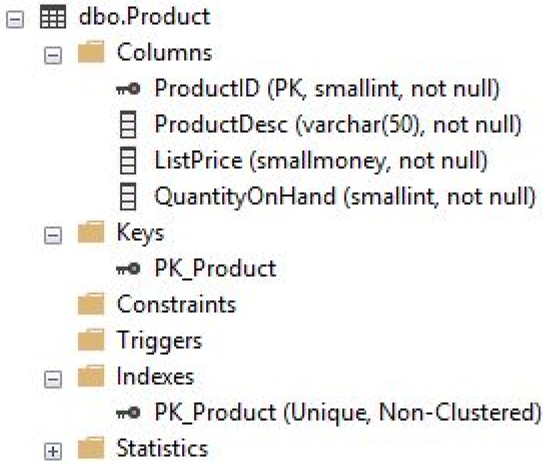
```
⊟ dbo.Inventory
    ⊟ Columns
          SupplierID (PK, FK, smallint, not null)
          ProductID (PK, FK, smallint, not null)
          Date (date, not null)
          Units (smallint, not null)
    ⊟ Keys
          CPK_Inventory
          FK_Inventory_Product
          FK_Inventory_Supplier
      Constraints
      Triggers
    ⊟ Indexes
          CPK_Inventory (Clustered)
    ⊞ Statistics
```

Order Table: Orders are tracked by the OrderID generated in the database. The date the order was placed and the fulfillment status are also recorded for each order. The foreign key of *CustomerID* is associated with the order table to identify which customer placed the order.



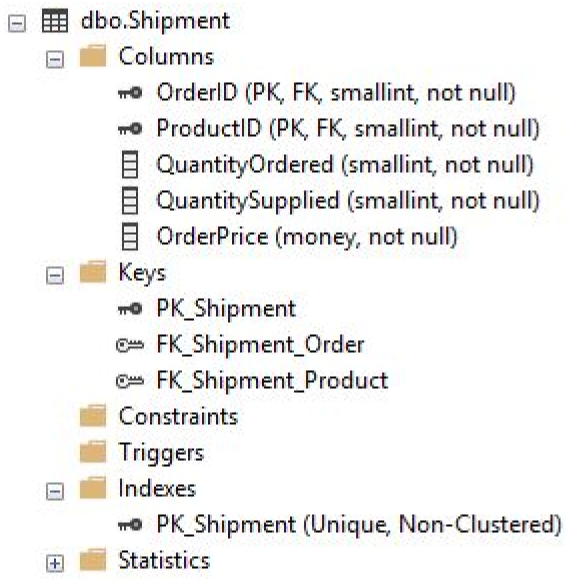
- dbo.Order
  - Columns
    - OrderID (PK, smallint, not null)
    - Date (date, not null)
    - Status (varchar(10), not null)
    - CustomerID (FK, smallint, not null)
  - Keys
    - PK_Order
    - FK_Order_Customer
  - Constraints
  - Triggers
  - Indexes
    - PK_Order (Unique, Non-Clustered)
  - Statistics

Product Table: Products are tracked by the ProductID generated in the database. Products may have more than one supplier. Product information such as the product description and list price are recorded for each product. The quantity on hand is also recorded for inventory purposes.



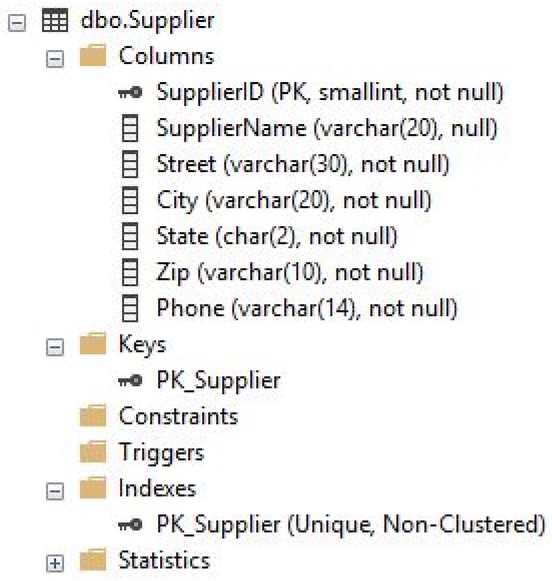
- dbo.Product
  - Columns
    - ProductID (PK, smallint, not null)
    - ProductDesc (varchar(50), not null)
    - ListPrice (smallmoney, not null)
    - QuantityOnHand (smallint, not null)
  - Keys
    - PK_Product
  - Constraints
  - Triggers
  - Indexes
    - PK_Product (Unique, Non-Clustered)
  - Statistics

Shipment Table: Shipments are tracked by the OrderID and ProductID generated by the Order and tables. The quantity ordered, quantity supplied, and total order price is recorded for each shipment.



```
⊟ dbo.Shipment
    ⊟ Columns
        OrderID (PK, FK, smallint, not null)
        ProductID (PK, FK, smallint, not null)
        QuantityOrdered (smallint, not null)
        QuantitySupplied (smallint, not null)
        OrderPrice (money, not null)
    ⊟ Keys
        PK_Shipment
        FK_Shipment_Order
        FK_Shipment_Product
      Constraints
      Triggers
    ⊟ Indexes
        PK_Shipment (Unique, Non-Clustered)
    ⊞ Statistics
```

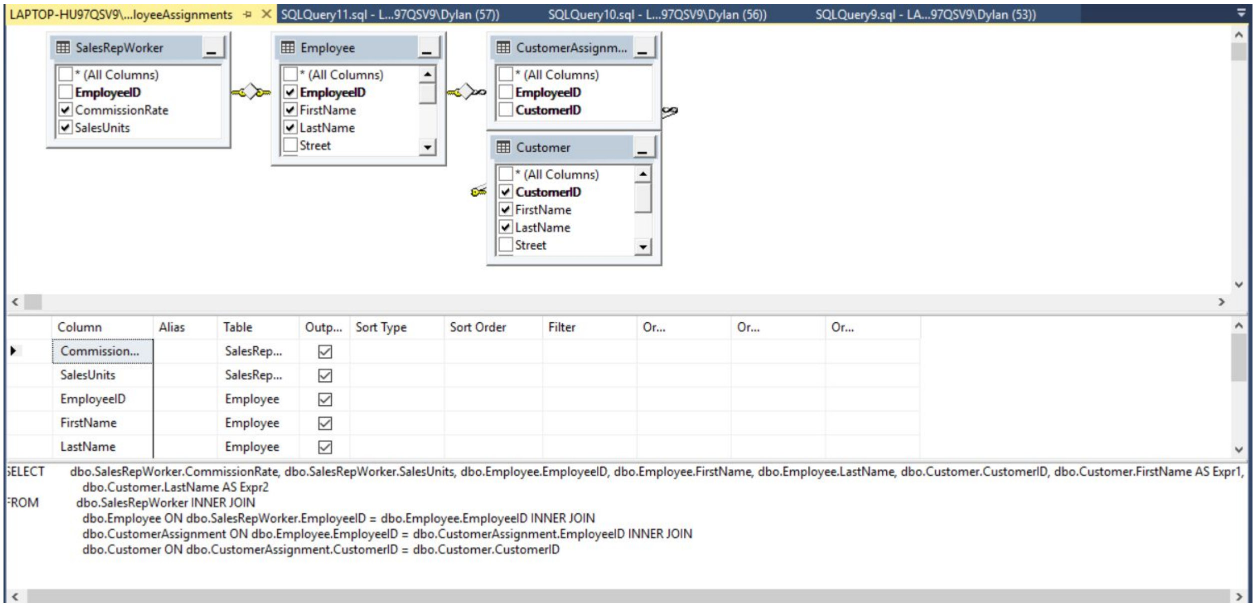
Supplier Table: Suppliers are tracked by the SupplierID generated in the database. Suppliers may offer more than one product. Business information such as name, address, and phone are recorded for each supplier.



```
⊟ dbo.Supplier
    ⊟ Columns
        SupplierID (PK, smallint, not null)
        SupplierName (varchar(20), null)
        Street (varchar(30), not null)
        City (varchar(20), not null)
        State (char(2), not null)
        Zip (varchar(10), not null)
        Phone (varchar(14), not null)
    ⊟ Keys
        PK_Supplier
      Constraints
      Triggers
    ⊟ Indexes
        PK_Supplier (Unique, Non-Clustered)
    ⊞ Statistics
```

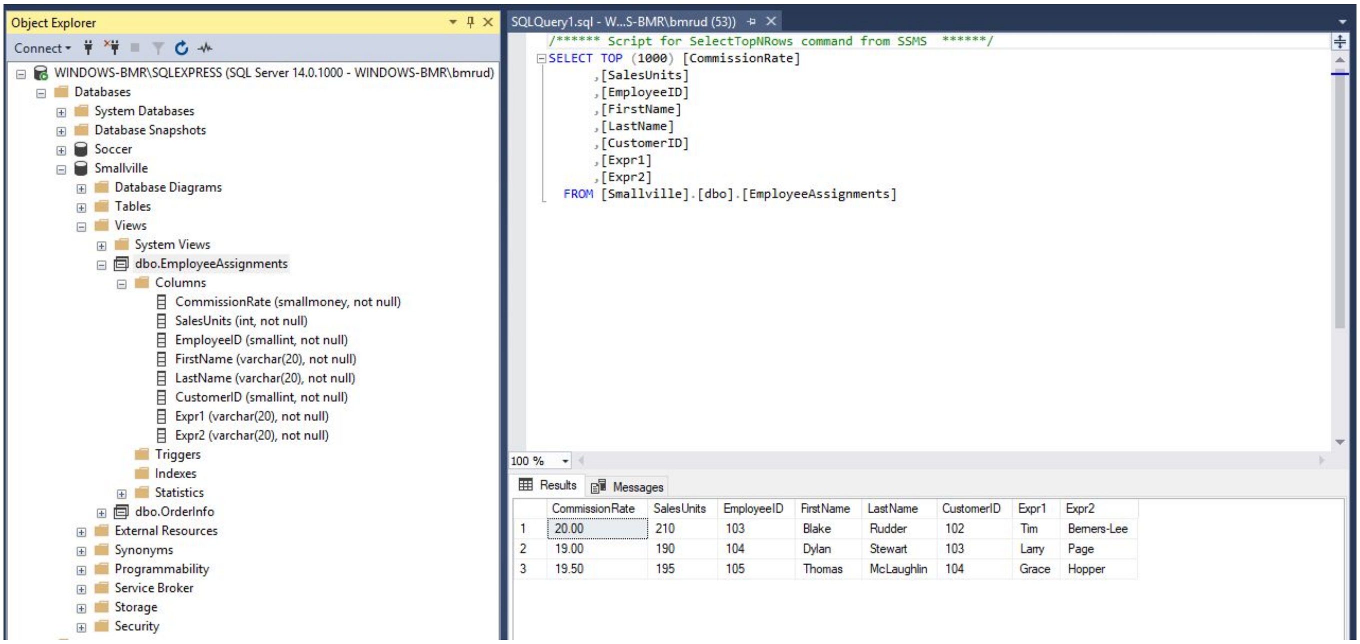
Table Views:

#1- This view shows all the employees, how many units they've sold, how much they make in commission, and the customer they are assigned to:



	Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
▶	Commission...		SalesRep...	☑						
	SalesUnits		SalesRep...	☑						
	EmployeeID		Employee	☑						
	FirstName		Employee	☑						
	LastName		Employee	☑						

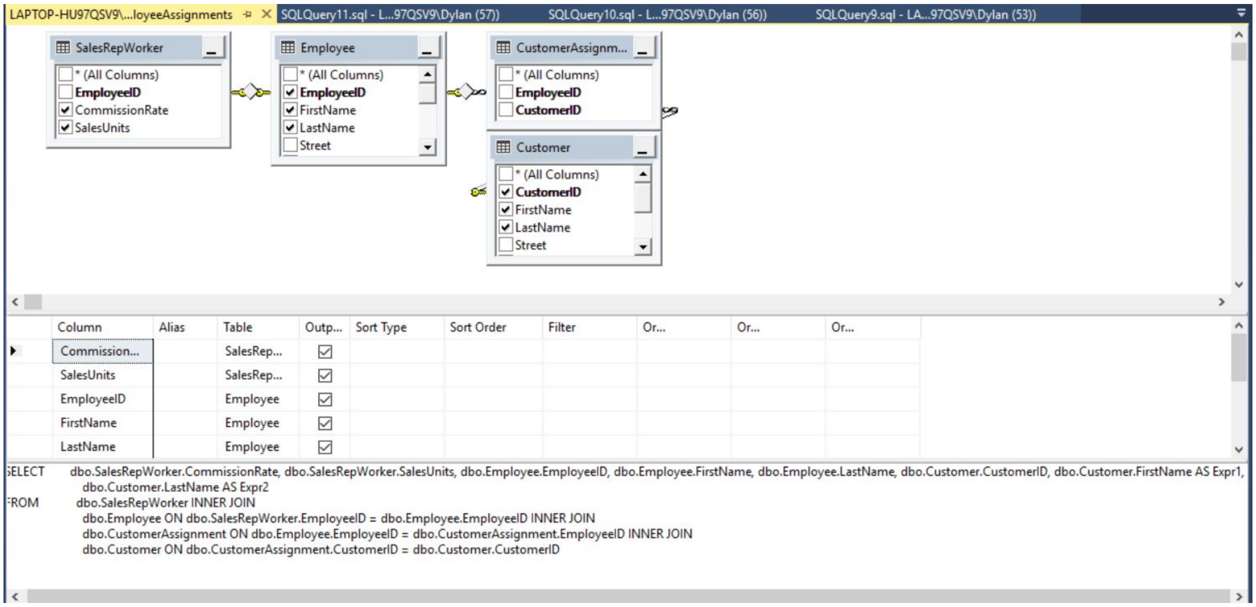
```
SELECT     dbo.SalesRepWorker.CommissionRate, dbo.SalesRepWorker.SalesUnits, dbo.Employee.EmployeeID, dbo.Employee.FirstName, dbo.Employee.LastName, dbo.Customer.CustomerID, dbo.Customer.FirstName AS Expr1,
             dbo.Customer.LastName AS Expr2
FROM        dbo.SalesRepWorker INNER JOIN
             dbo.Employee ON dbo.SalesRepWorker.EmployeeID = dbo.Employee.EmployeeID INNER JOIN
             dbo.CustomerAssignment ON dbo.Employee.EmployeeID = dbo.CustomerAssignment.EmployeeID INNER JOIN
             dbo.Customer ON dbo.CustomerAssignment.CustomerID = dbo.Customer.CustomerID
```


Object Explorer
Connect ▾
WINDOWS-BMR\SQLEXPRESS (SQL Server 14.0.1000 - WINDOWS-BMR\bmrud)
Databases
System Databases
Database Snapshots
Soccer
Smallville
Database Diagrams
Tables
Views
System Views
dbo.EmployeeAssignments
Columns
CommissionRate (smallmoney, not null)
SalesUnits (int, not null)
EmployeeID (smallint, not null)
FirstName (varchar(20), not null)
LastName (varchar(20), not null)
CustomerID (smallint, not null)
Expr1 (varchar(20), not null)
Expr2 (varchar(20), not null)
Triggers
Indexes
Statistics
dbo.OrderInfo
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
SQLQuery1.sql - W...S-BMR\bmrud (53))
100 %
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [CommissionRate]
      ,[SalesUnits]
      ,[EmployeeID]
      ,[FirstName]
      ,[LastName]
      ,[CustomerID]
      ,[Expr1]
      ,[Expr2]
  FROM [Smallville].[dbo].[EmployeeAssignments]
```

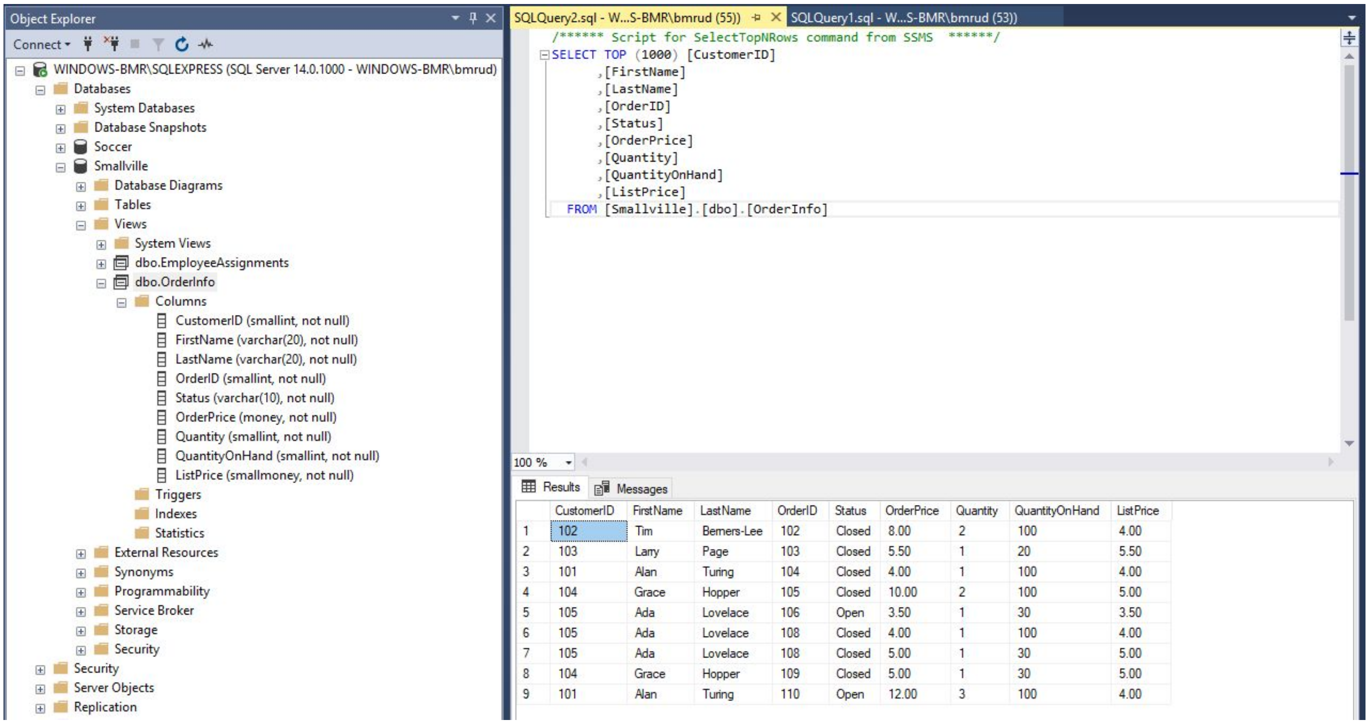
	CommissionRate	SalesUnits	EmployeeID	FirstName	LastName	CustomerID	Expr1	Expr2
1	20.00	210	103	Blake	Rudder	102	Tim	Berners-Lee
2	19.00	190	104	Dylan	Stewart	103	Larry	Page
3	19.50	195	105	Thomas	McLaughlin	104	Grace	Hopper

#2- This view shows every customer order, what the order contains, how much of that item is in stock, and the status and price of the order:



Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
Commission...		SalesRep...	☑						
SalesUnits		SalesRep...	☑						
EmployeeID		Employee	☑						
FirstName		Employee	☑						
LastName		Employee	☑						

```
SELECT    dbo.SalesRepWorker.CommissionRate, dbo.SalesRepWorker.SalesUnits, dbo.Employee.EmployeeID, dbo.Employee.FirstName, dbo.Employee.LastName, dbo.Customer.CustomerID, dbo.Customer.FirstName AS Expr1,
            dbo.Customer.LastName AS Expr2
FROM      dbo.SalesRepWorker INNER JOIN
            dbo.Employee ON dbo.SalesRepWorker.EmployeeID = dbo.Employee.EmployeeID INNER JOIN
            dbo.CustomerAssignment ON dbo.Employee.EmployeeID = dbo.CustomerAssignment.EmployeeID INNER JOIN
            dbo.Customer ON dbo.CustomerAssignment.CustomerID = dbo.Customer.CustomerID
```


Object Explorer
Connect ▾
SQLQuery2.sql - W...S-BMR\bmrud (55))
SQLQuery1.sql - W...S-BMR\bmrud (53))
100 %
Results
Messages

WINDOWS-BMR\SQLEXPRESS (SQL Server 14.0.1000 - WINDOWS-BMR\bmrud)
Databases
System Databases
Database Snapshots
Soccer
Smallville
Database Diagrams
Tables
Views
System Views
dbo.EmployeeAssignments
dbo.OrderInfo
Columns
CustomerID (smallint, not null)
FirstName (varchar(20), not null)
LastName (varchar(20), not null)
OrderID (smallint, not null)
Status (varchar(10), not null)
OrderPrice (money, not null)
Quantity (smallint, not null)
QuantityOnHand (smallint, not null)
ListPrice (smallmoney, not null)
Triggers
Indexes
Statistics
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication

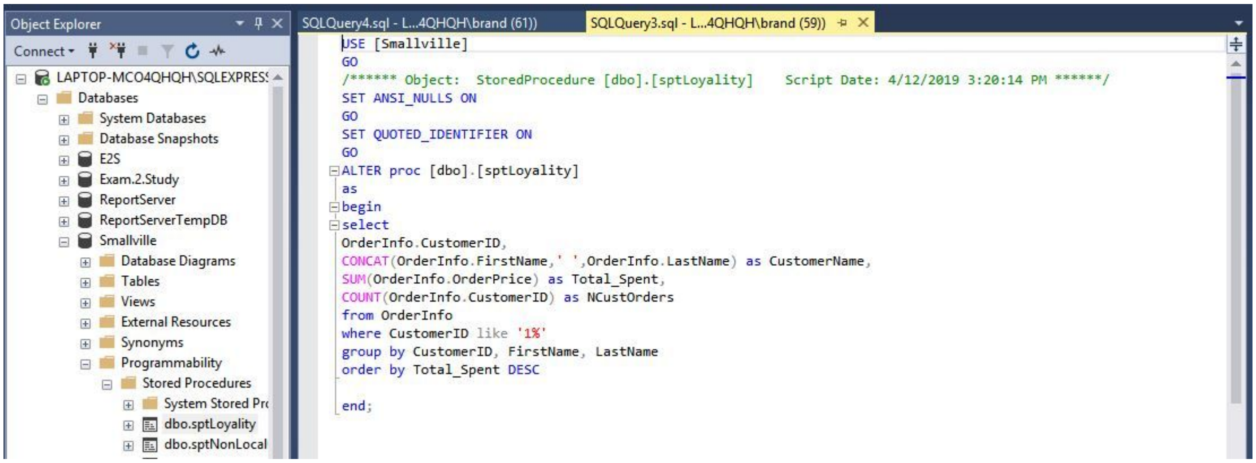
```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [CustomerID]
      ,[FirstName]
      ,[LastName]
      ,[OrderID]
      ,[Status]
      ,[OrderPrice]
      ,[Quantity]
      ,[QuantityOnHand]
      ,[ListPrice]
  FROM [Smallville].[dbo].[OrderInfo]
```

	CustomerID	FirstName	LastName	OrderID	Status	OrderPrice	Quantity	QuantityOnHand	ListPrice
1	102	Tim	Berners-Lee	102	Closed	8.00	2	100	4.00
2	103	Larry	Page	103	Closed	5.50	1	20	5.50
3	101	Alan	Turing	104	Closed	4.00	1	100	4.00
4	104	Grace	Hopper	105	Closed	10.00	2	100	5.00
5	105	Ada	Lovelace	106	Open	3.50	1	30	3.50
6	105	Ada	Lovelace	108	Closed	4.00	1	100	4.00
7	105	Ada	Lovelace	108	Closed	5.00	1	30	5.00
8	104	Grace	Hopper	109	Closed	5.00	1	30	5.00
9	101	Alan	Turing	110	Open	12.00	3	100	4.00

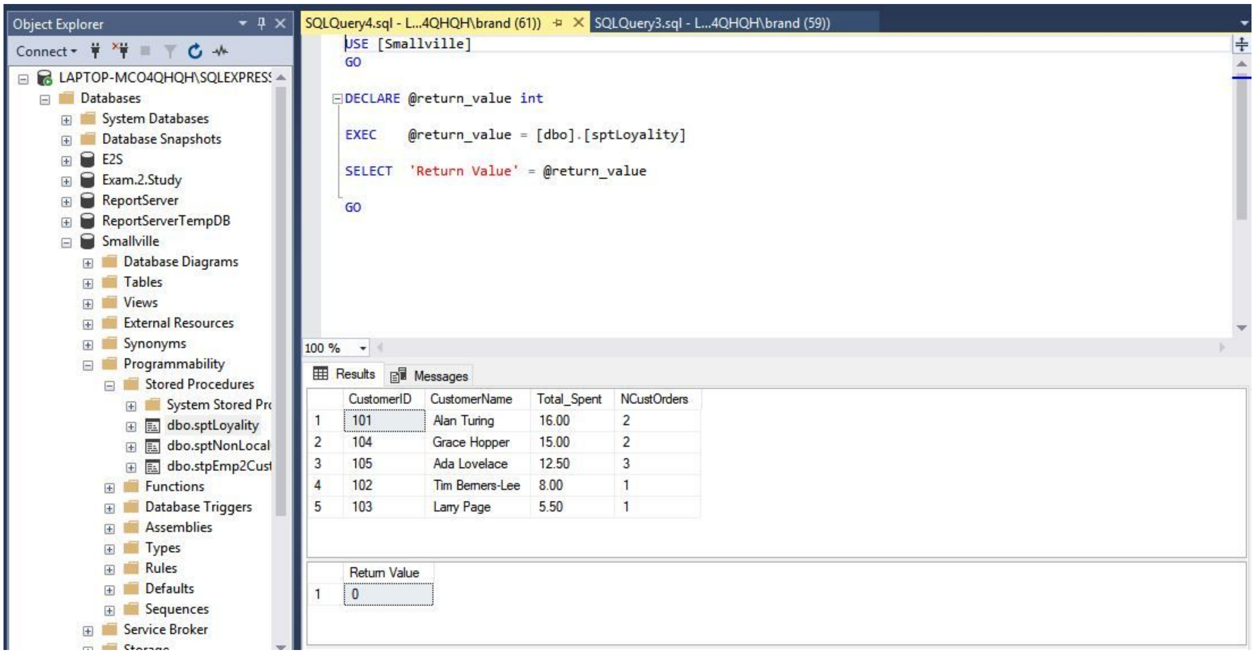
SPROC:

#1- This SPROC shows the amount spent by each customer with a customer ID starting with 1.

It also shows the number of orders each customer placed.



```
USE [Smallville]
GO
/****** Object:  StoredProcedure [dbo].[sptLoyality]    Script Date: 4/12/2019 3:20:14 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER proc [dbo].[sptLoyality]
as
begin
select
OrderInfo.CustomerID,
CONCAT(OrderInfo.FirstName,' ',OrderInfo.LastName) as CustomerName,
SUM(OrderInfo.OrderPrice) as Total_Spent,
COUNT(OrderInfo.CustomerID) as NCustOrders
from OrderInfo
where CustomerID like '1%'
group by CustomerID, FirstName, LastName
order by Total_Spent DESC

end;
```

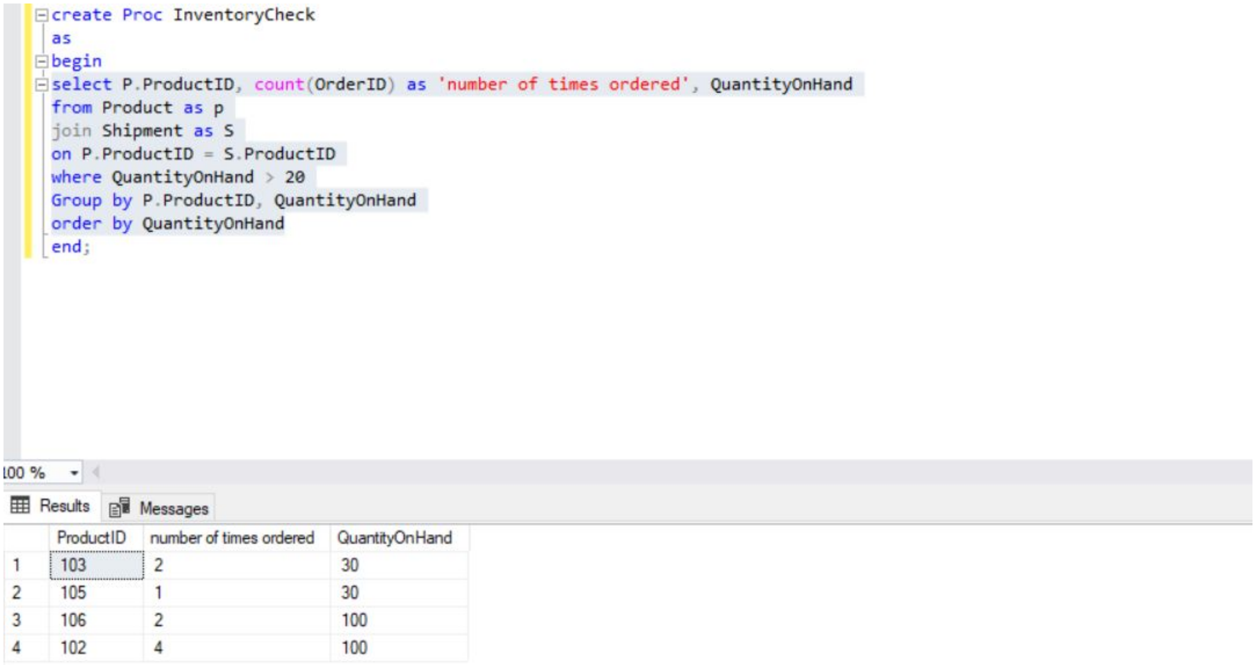


```
USE [Smallville]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[sptLoyality]

SELECT  'Return Value' = @return_value

GO
```

	CustomerID	CustomerName	Total_Spent	NCustOrders
1	101	Alan Turing	16.00	2
2	104	Grace Hopper	15.00	2
3	105	Ada Lovelace	12.50	3
4	102	Tim Berners-Lee	8.00	1
5	103	Larry Page	5.50	1

	Return Value
1	0

#2- This SPROC shows the number of orders each product is in, and how many of that product is on hand

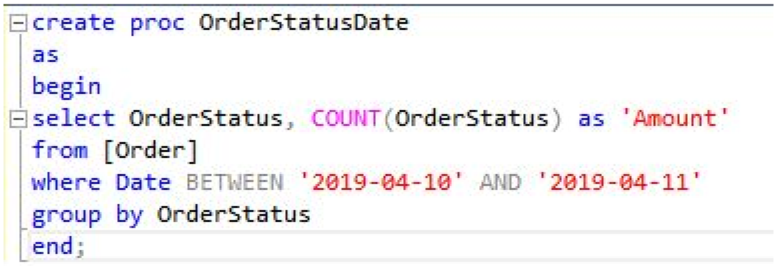


```
create Proc InventoryCheck
as
begin
select P.ProductID, count(OrderID) as 'number of times ordered', QuantityOnHand
from Product as p
join Shipment as S
on P.ProductID = S.ProductID
where QuantityOnHand > 20
Group by P.ProductID, QuantityOnHand
order by QuantityOnHand
end;
```

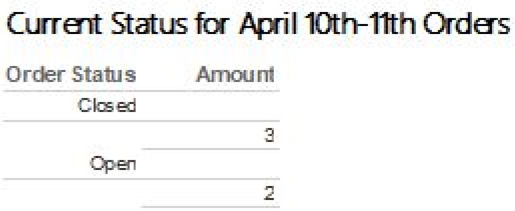
	ProductID	number of times ordered	QuantityOnHand
1	103	2	30
2	105	1	30
3	106	2	100
4	102	4	100

Reports & UAT:

This user acceptance test checks for the current status of the orders placed during the 2-day sale:



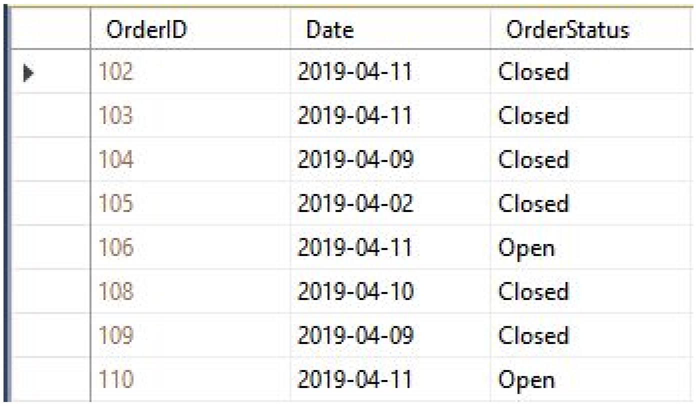
```
create proc OrderStatusDate
as
begin
select OrderStatus, COUNT(OrderStatus) as 'Amount'
from [Order]
where Date BETWEEN '2019-04-10' AND '2019-04-11'
group by OrderStatus
end;
```



Current Status for April 10th-11th Orders

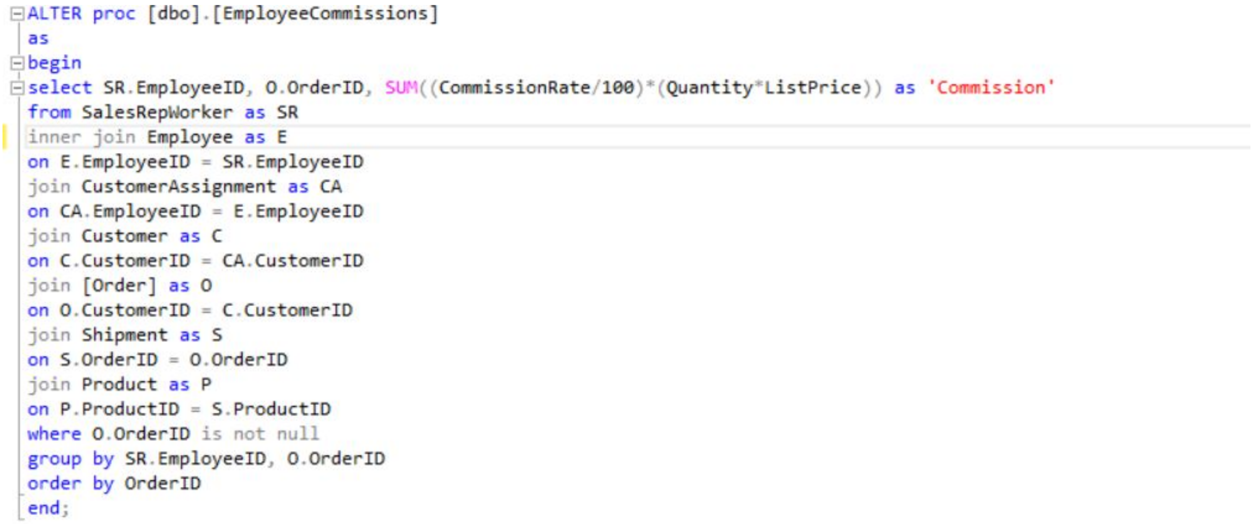
Order Status	Amount
Closed	3
Open	2

Logic Check

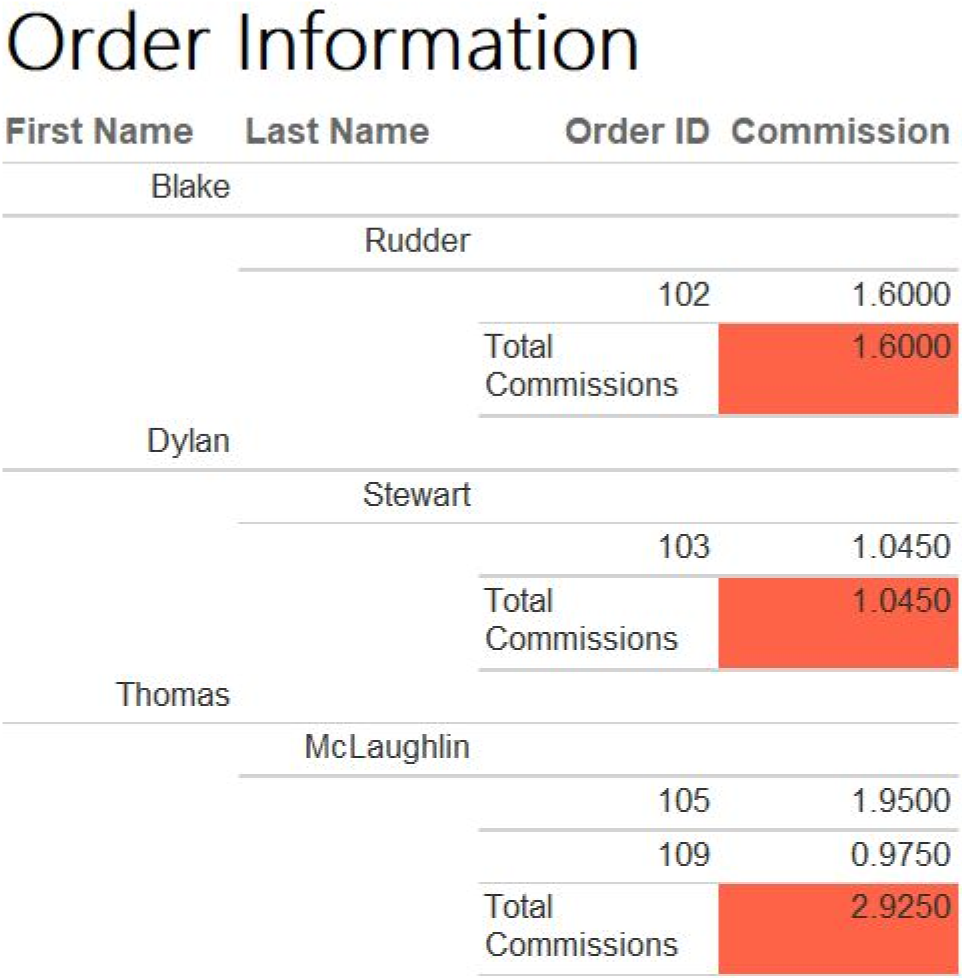


	OrderID	Date	OrderStatus
▶	102	2019-04-11	Closed
	103	2019-04-11	Closed
	104	2019-04-09	Closed
	105	2019-04-02	Closed
	106	2019-04-11	Open
	108	2019-04-10	Closed
	109	2019-04-09	Closed
	110	2019-04-11	Open

This user acceptance test shows the overall employee commissions for the sales representatives:



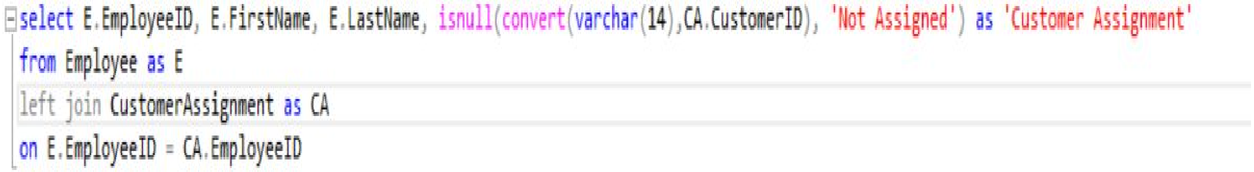
```
ALTER proc [dbo].[EmployeeCommissions]
as
begin
select SR.EmployeeID, O.OrderID, SUM((CommissionRate/100)*(Quantity*ListPrice)) as 'Commission'
from SalesRepWorker as SR
inner join Employee as E
on E.EmployeeID = SR.EmployeeID
join CustomerAssignment as CA
on CA.EmployeeID = E.EmployeeID
join Customer as C
on C.CustomerID = CA.CustomerID
join [Order] as O
on O.CustomerID = C.CustomerID
join Shipment as S
on S.OrderID = O.OrderID
join Product as P
on P.ProductID = S.ProductID
where O.OrderID is not null
group by SR.EmployeeID, O.OrderID
order by OrderID
end;
```



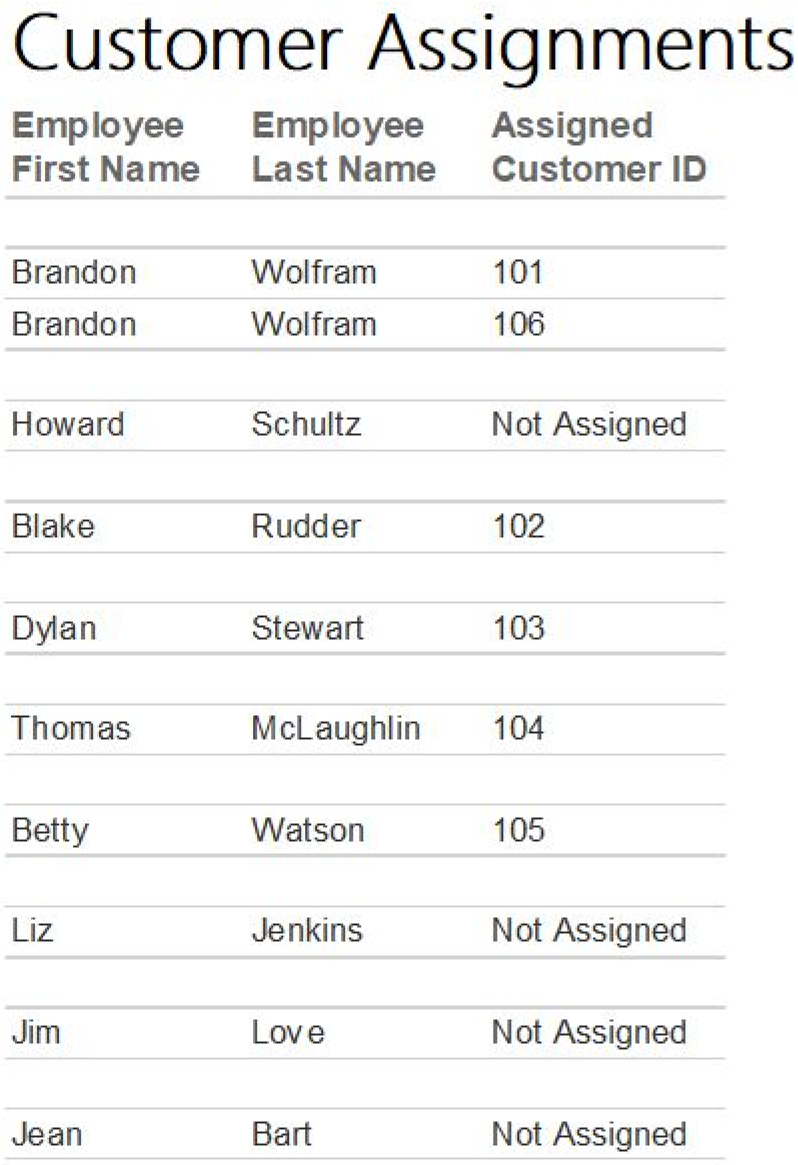
Order Information

First Name	Last Name	Order ID	Commission
Blake			
	Rudder		
		102	1.6000
		Total Commissions	1.6000
Dylan			
	Stewart		
		103	1.0450
		Total Commissions	1.0450
Thomas			
	McLaughlin		
		105	1.9500
		109	0.9750
		Total Commissions	2.9250

This user acceptance test checks for any employees that are not assigned to a customer:



```
select E.EmployeeID, E.FirstName, E.LastName, isnull(convert(varchar(14),CA.CustomerID), 'Not Assigned') as 'Customer Assignment'
from Employee as E
left join CustomerAssignment as CA
on E.EmployeeID = CA.EmployeeID
```



Customer Assignments

Employee First Name	Employee Last Name	Assigned Customer ID
Brandon	Wolfram	101
Brandon	Wolfram	106
Howard	Schultz	Not Assigned
Blake	Rudder	102
Dylan	Stewart	103
Thomas	McLaughlin	104
Betty	Watson	105
Liz	Jenkins	Not Assigned
Jim	Love	Not Assigned
Jean	Bart	Not Assigned