Team Name: Database Dimez

Team Motto: D’s are for Database

Team Lead:

Robert Lumadue

Team Members:

Comeskey, Matthew

Elliott, Kaydiann

Hartman, Justin

Contents

<u>Version</u>	<u>Description</u>
Version 1.0	First released draft
Version 1.1	Second released draft. Summary of changes: 1. Updated: a. RS b. ERD c. EERD d. Data Dictionary e. Table Views
Version 2.1	Third Released Draft Summary of Changes: 1. Updated: a. RS b. Table_vw c. SPROCS
Version 3.1	Fourth Released draft Summary of Changes: 1. Updated: a. SSRS Reports b. User acceptance test queries c. Changed one of the SPROCS

Purpose

The goal of this document is to lay down a template for how a database will be created and set in place for a mechanic shop that services customers cars and allows customers to order automotive products. Items that will be discussed in this document include Narrative, Requirements, Entities, ERD, EERD, Relational Schema, Data Dictionary, Microsoft SQL server tables, Table views, SPROCS, SSRS Reports, and User acceptance test queries.

Narrative

Dan’s Auto-Mechanic Shop is located in a small town in Indiana and they want to track daily services. The store has multiple employees and each is assigned a unique worker ID to clock in with. The employee’s first and last name, phone number, and speciality (mechanic, clerk, etc.) is also tracked. Some employees act as supervisors to other employees, some supervisors will have no employees or will supervise multiple employees. No employee is supervised by more than one supervisor, and an employee can be unsupervised.

An employee of Dan’s shop can be responsible for providing service to no cars or multiple cars, while the car can receive service from multiple employees or no employees. The shop assigns each car a unique Car ID and tracks the car make and model. They also want to track appointment date & time, and results of the service. Each car belongs to one customer, and the customer can bring in multiple cars to be serviced. A customer is granted a unique ID when an appointment is set up. The shop wants to track the name (first and last), address (street, city, state, and zip), and their contact phone number.

A customer has the ability to place no orders or multiple orders if they are advised to by the shop employee. If the customer decides to follow the employees recommendation it is the customers responsibility to place an order for the parts. Once this occurs a unique order number is created that will show the order date and status. An order can have one or more products attached to it’s order number. Products are tracked by a product number assigned by Dan’s mechanic shop and will include a product description, handling information, estimated installation time in hours (each product that is purchased comes with this information), and company. The relationship between order and product is marked by quantity ordered and price.

Requirements

EMPLOYEE - An employee will service none or multiple cars. An employee will also supervise none or multiple employees. An employee can have no more than one supervisor or can be unsupervised.

CUSTOMER - A customer will have one or more cars to be serviced. A customer will also create multiple or no orders.

CAR - A car can receive service from multiple employees or no employees. A car also belongs to only one customer.

ORDER - An order is associated with only one customer. An order can contain one or more products.

PRODUCT - A product can be associated with one or more orders.

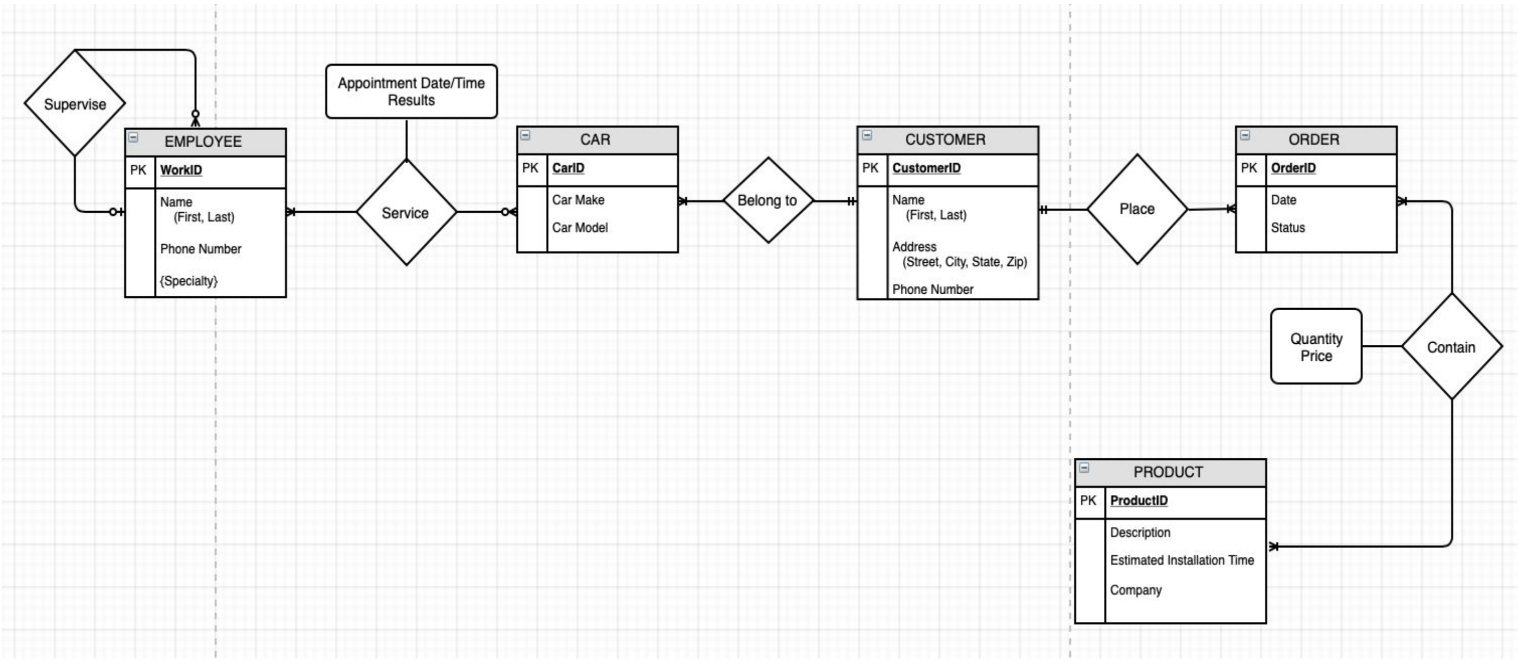
Entities to be Tracked

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

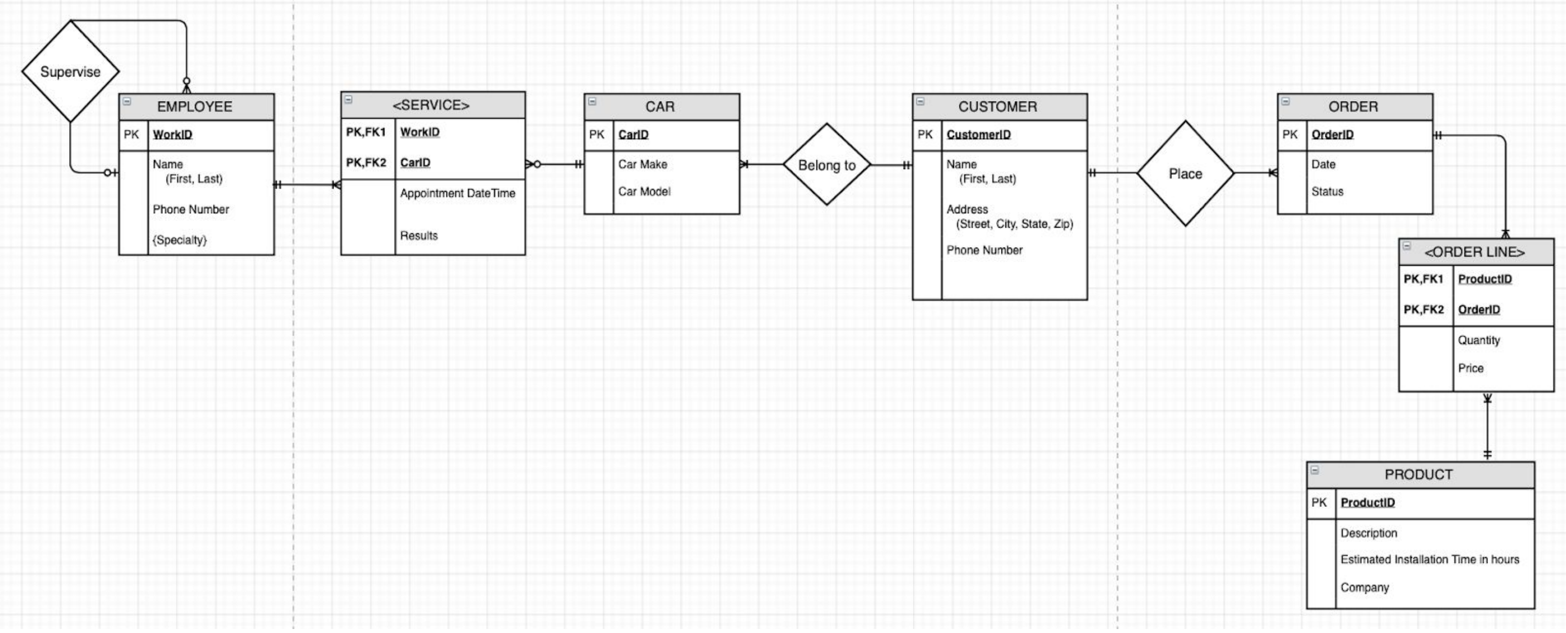
Entities with Nested Attributes

1. EMPLOYEE - WorkID, First Name, Last Name, Phone number, Specialty
2. CUSTOMER - CustomerID, First Name, Last Name, Address, Phone number
3. CAR - CarID, Car Make, Car Model
4. ORDER - OrderID, Date, Status
5. PRODUCT - ProductID, Description, Estimated Installation Time, Company

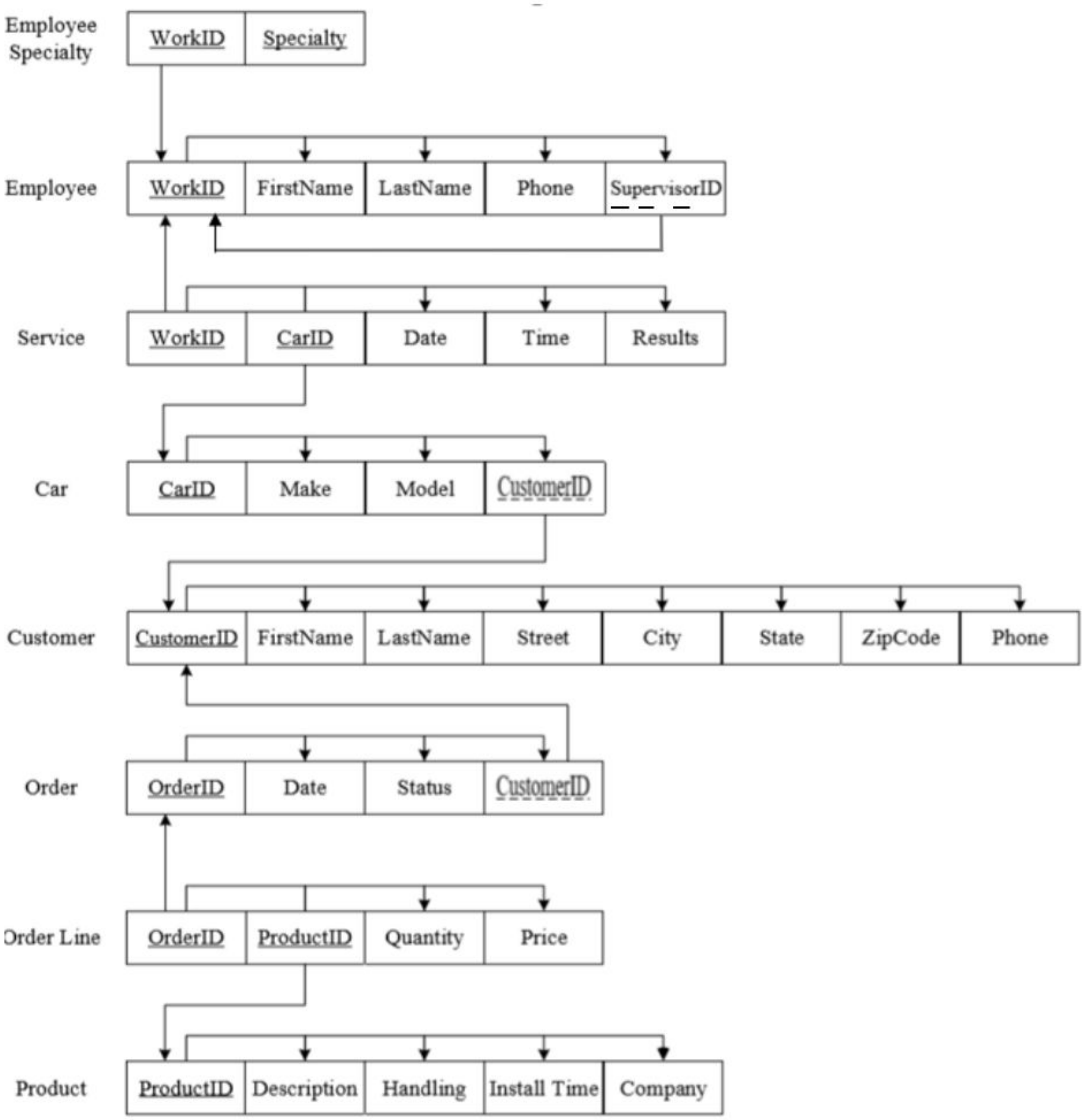
ERD


Supervise
EMPLOYEE
PK
WorkID
Name
(First, Last)
Phone Number
{Specialty}
Appointment Date/Time
Results
Service
CAR
PK
CarID
Car Make
Car Model
Belong to
CUSTOMER
PK
CustomerID
Name
(First, Last)
Address
(Street, City, State, Zip)
Phone Number
Place
ORDER
PK
OrderID
Date
Status
Quantity
Price
Contain
PRODUCT
PK
ProductID
Description
Estimated Installation Time
Company

EERD

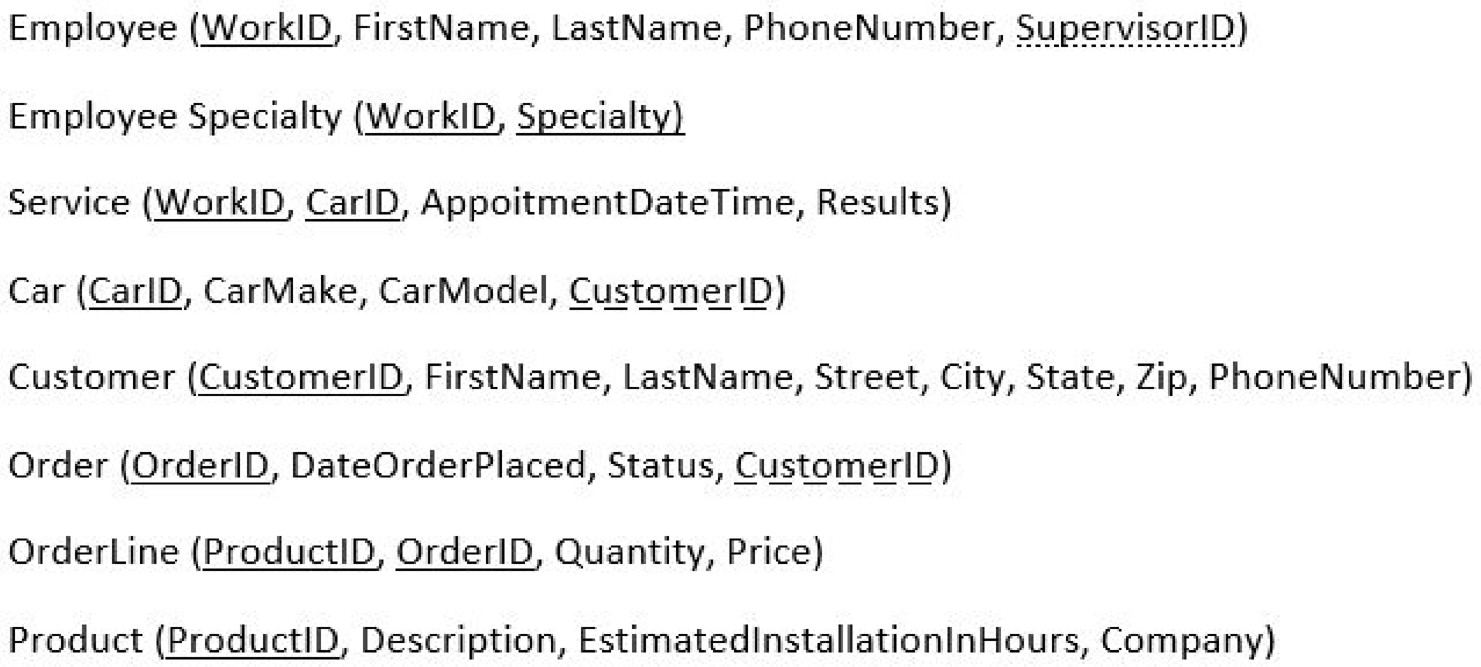

Supervise
EMPLOYEE
PK WorkID
Name
(First, Last)
Phone Number
{Specialty}
<SERVICE>
PK,FK1 WorkID
PK,FK2 CarID
Appointment DateTime
Results
CAR
PK CarID
Car Make
Car Model
Belong to
CUSTOMER
PK CustomerID
Name
(First, Last)
Address
(Street, City, State, Zip)
Phone Number
Place
ORDER
PK OrderID
Date
Status
<ORDER LINE>
PK,FK1 ProductID
PK,FK2 OrderID
Quantity
Price
PRODUCT
PK ProductID
Description
Estimated Installation Time in hours
Company

Relational Schema



Data Dictionary Summary Header

Dan’s Mechanic Shop Tables and Attributes



Employee (WorkID, FirstName, LastName, PhoneNumber, SupervisorID)

Employee Specialty (WorkID, Specialty)

Service (WorkID, CarID, AppoitmentDateTime, Results)

Car (CarID, CarMake, CarModel, CustomerID)

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

Order (OrderID, DateOrderPlaced, Status, CustomerID)

OrderLine (ProductID, OrderID, Quantity, Price)

Product (ProductID, Description, EstimatedInstallationInHours, Company)

Dan’s Mechanic Shop - Data Dictionary

(Microsoft SQL Server Notation)

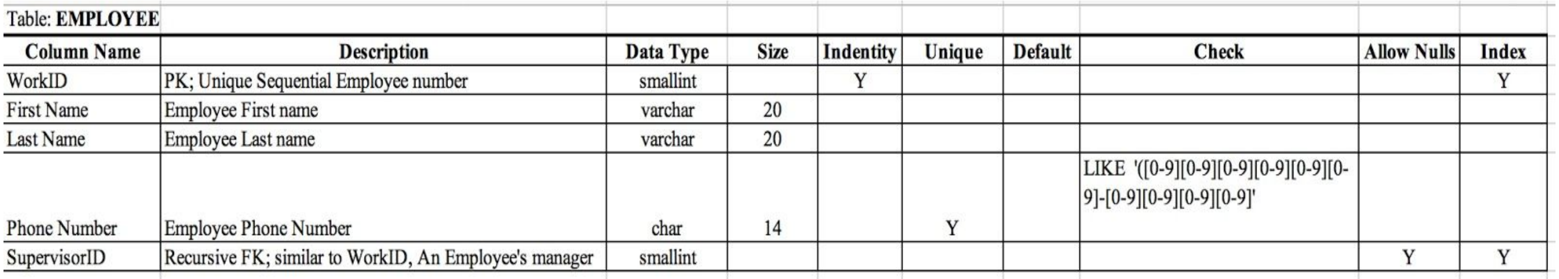


Table: **EMPLOYEE**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
WorkID	PK; Unique Sequential Employee number	smallint		Y					Y
First Name	Employee First name	varchar	20						
Last Name	Employee Last name	varchar	20						
Phone Number	Employee 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]'		
SupervisorID	Recursive FK; similar to WorkID, An Employee's manager	smallint						Y	Y

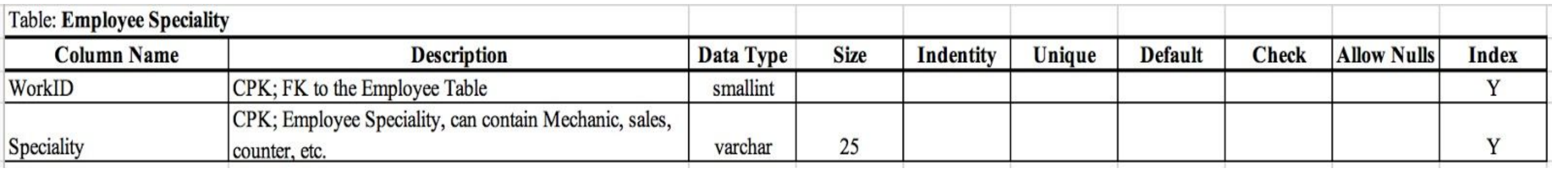


Table: **Employee Speciality**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
WorkID	CPK; FK to the Employee Table	smallint							Y
Speciality	CPK; Employee Speciality, can contain Mechanic, sales, counter, etc.	varchar	25						Y

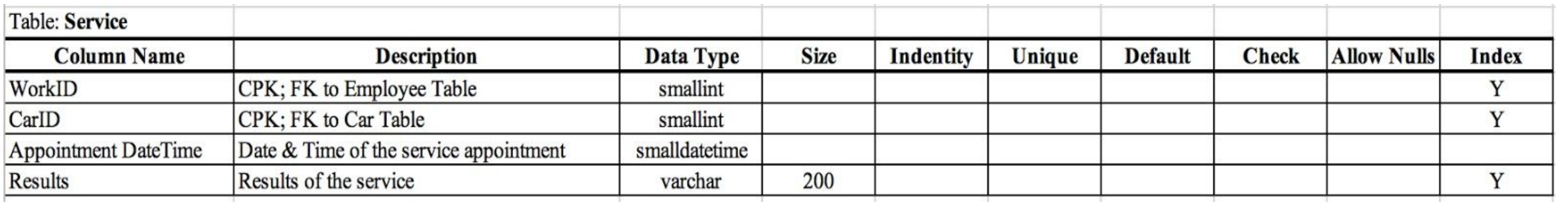


Table: **Service**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
WorkID	CPK; FK to Employee Table	smallint							Y
CarID	CPK; FK to Car Table	smallint							Y
Appointment DateTime	Date & Time of the service appointment	smalldatetime							
Results	Results of the service	varchar	200						Y

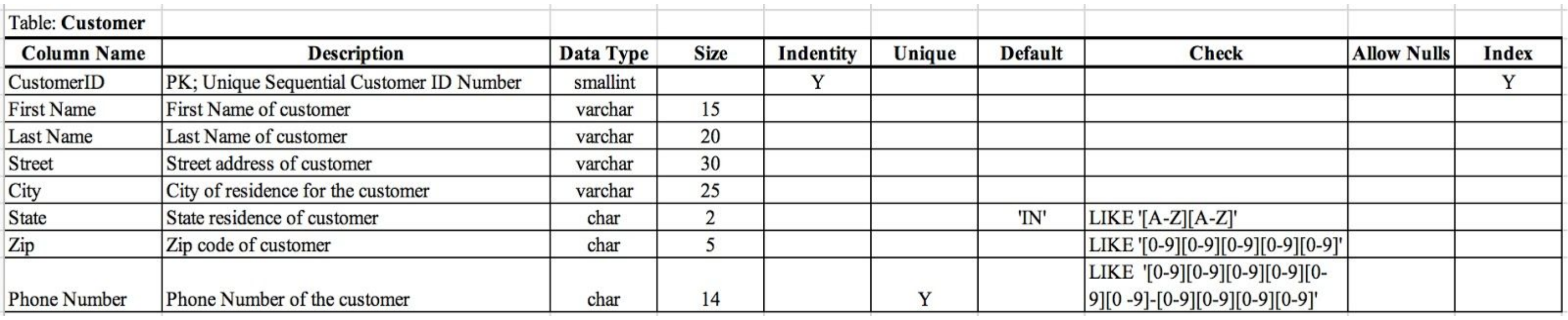


Table: **Customer**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
CustomerID	PK; Unique Sequential Customer ID Number	smallint		Y					Y
First Name	First Name of customer	varchar	15						
Last Name	Last Name of customer	varchar	20						
Street	Street address of customer	varchar	30						
City	City of residence for the customer	varchar	25						
State	State residence of customer	char	2			'IN'	LIKE '[A-Z][A-Z]'		
Zip	Zip code of customer	char	5				LIKE '[0-9][0-9][0-9][0-9][0-9]'		
Phone Number	Phone Number of the customer	char	14		Y		LIKE '[0-9][0-9][0-9][0-9][0-9][0 -9]-[0-9][0-9][0-9][0-9]'		

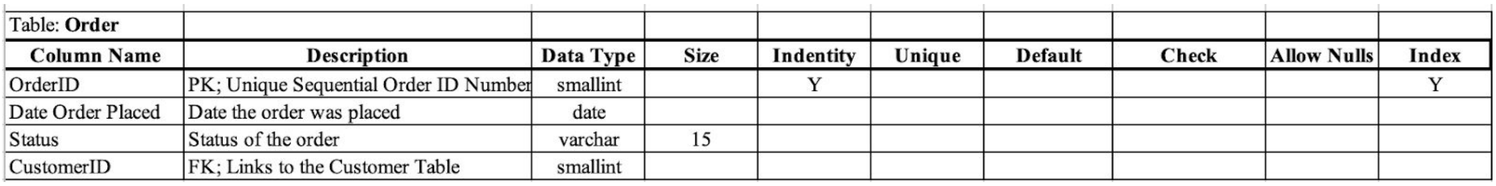


Table: **Order**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
OrderID	PK; Unique Sequential Order ID Number	smallint		Y					Y
Date Order Placed	Date the order was placed	date							
Status	Status of the order	varchar	15						
CustomerID	FK; Links to the Customer Table	smallint							

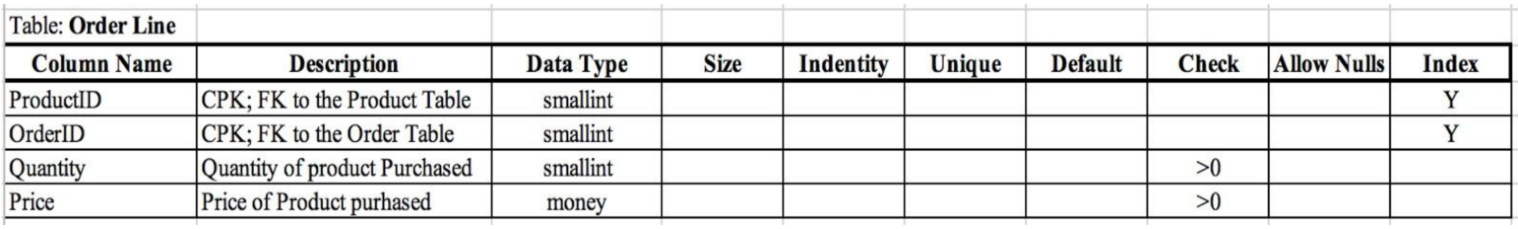


Table: **Order Line**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
ProductID	CPK; FK to the Product Table	smallint							Y
OrderID	CPK; FK to the Order Table	smallint							Y
Quantity	Quantity of product Purchased	smallint					>0		
Price	Price of Product purhased	money					>0		

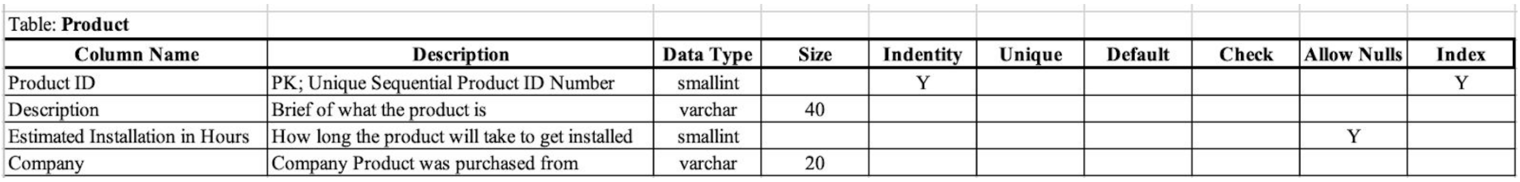


Table: **Product**

Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
Product ID	PK; Unique Sequential Product ID Number	smallint		Y					Y
Description	Brief of what the product is	varchar	40						
Estimated Installation in Hours	How long the product will take to get installed	smallint						Y	
Company	Company Product was purchased from	varchar	20						

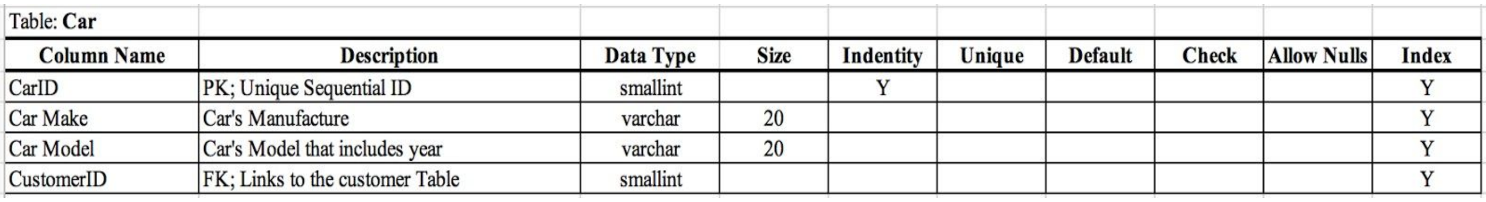
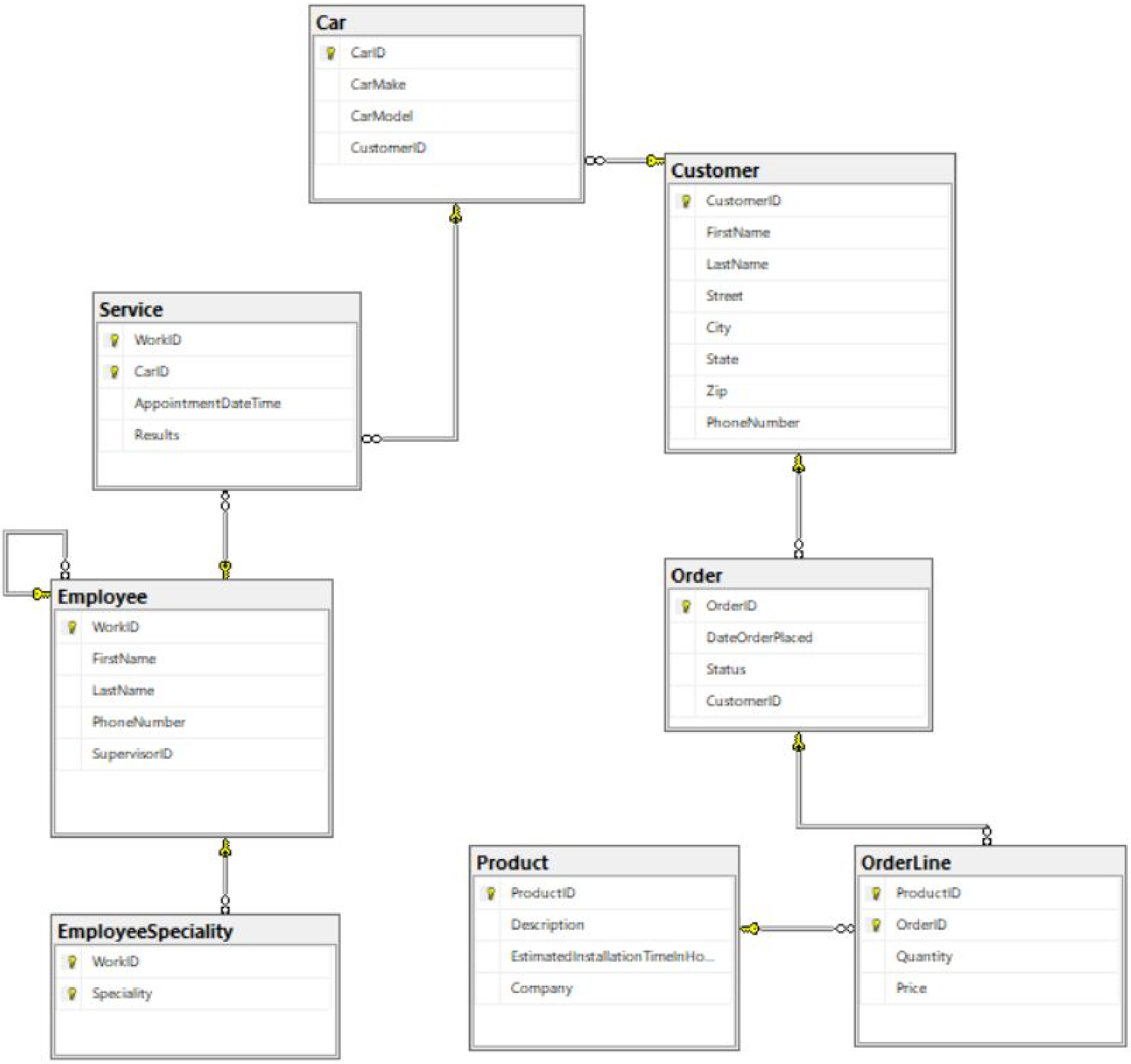


Table: **Car**

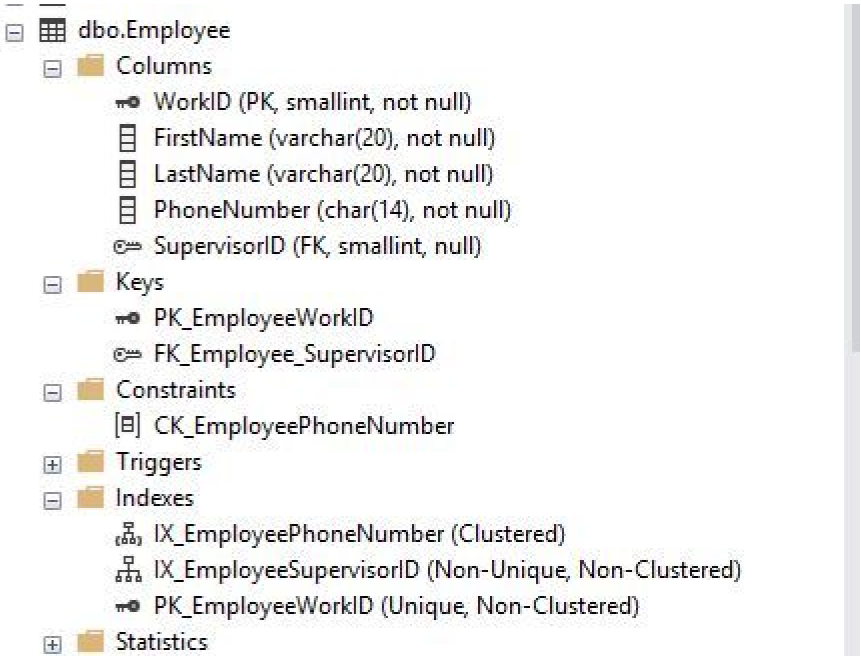
Column Name	Description	Data Type	Size	Indentity	Unique	Default	Check	Allow Nulls	Index
CarID	PK; Unique Sequential ID	smallint		Y					Y
Car Make	Car's Manufacture	varchar	20						Y
Car Model	Car's Model that includes year	varchar	20						Y
CustomerID	FK; Links to the customer Table	smallint							Y

Data Diagram



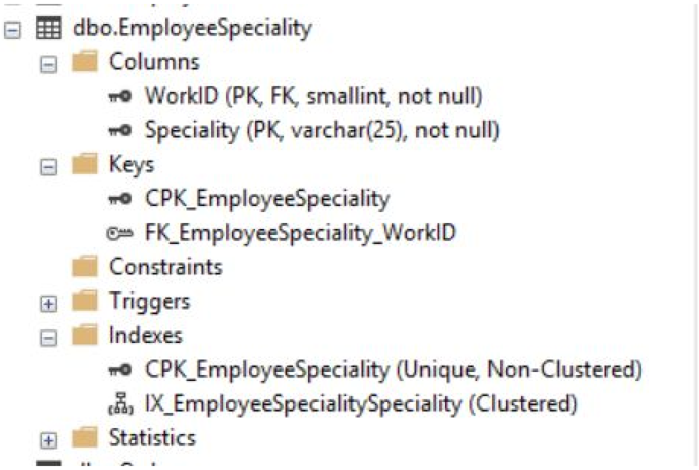
<u>Table Views</u>

Employee Table: Employees are tracked by a WorkID. Supervisor is a recursive FK for the table to track who are the managers. An employee phone number is a unique key & there is a constraint on the digits of a number that can be entered.



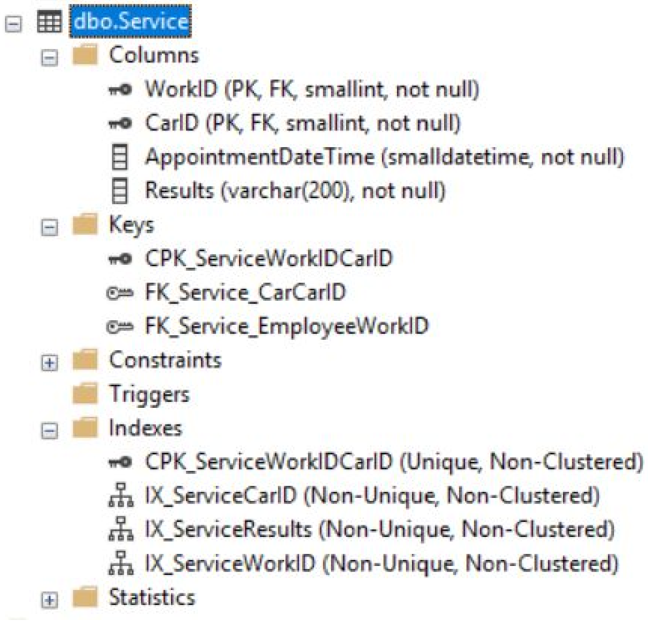
```
⊟ dbo.Employee
    ⊟ Columns
          WorkID (PK, smallint, not null)
          FirstName (varchar(20), not null)
          LastName (varchar(20), not null)
          PhoneNumber (char(14), not null)
          SupervisorID (FK, smallint, null)
    ⊟ Keys
          PK_EmployeeWorkID
          FK_Employee_SupervisorID
    ⊟ Constraints
          CK_EmployeePhoneNumber
    ⊞ Triggers
    ⊟ Indexes
          IX_EmployeePhoneNumber (Clustered)
          IX_EmployeeSupervisorID (Non-Unique, Non-Clustered)
          PK_EmployeeWorkID (Unique, Non-Clustered)
    ⊞ Statistics
```

Employee Speciality Table: This table tracks the specialty value of an employee and contains the WorkID foreign Key from the Employee Table.



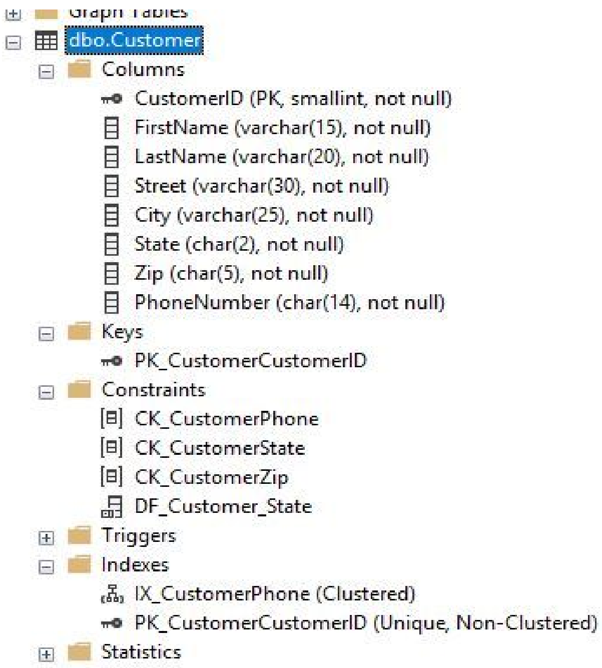
```
⊟ dbo.EmployeeSpeciality
    ⊟ Columns
          WorkID (PK, FK, smallint, not null)
          Speciality (PK, varchar(25), not null)
    ⊟ Keys
          CPK_EmployeeSpeciality
          FK_EmployeeSpeciality_WorkID
      Constraints
    ⊞ Triggers
    ⊟ Indexes
          CPK_EmployeeSpeciality (Unique, Non-Clustered)
          IX_EmployeeSpecialitySpeciality (Clustered)
    ⊞ Statistics
```

Service Table: This is the associative entity table that is tracked by two FK's that make up PK's from the Customer Table & Employee Table.



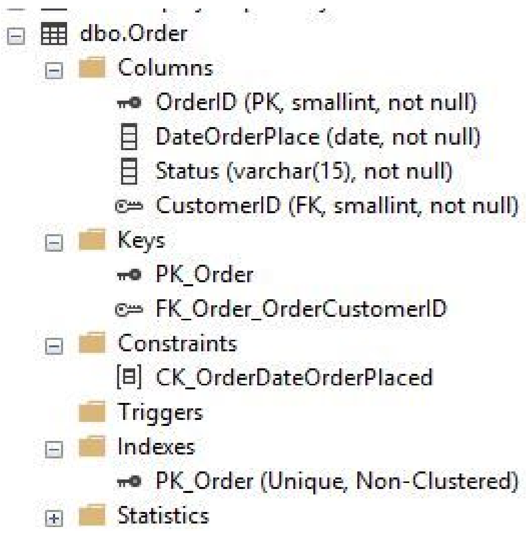
- dbo.Service
  - Columns
    - WorkID (PK, FK, smallint, not null)
    - CarID (PK, FK, smallint, not null)
    - AppointmentDateTime (smalldatetime, not null)
    - Results (varchar(200), not null)
  - Keys
    - CPK_ServiceWorkIDCarID
    - FK_Service_CarCarID
    - FK_Service_EmployeeWorkID
  - Constraints
  - Triggers
  - Indexes
    - CPK_ServiceWorkIDCarID (Unique, Non-Clustered)
    - IX_ServiceCarID (Non-Unique, Non-Clustered)
    - IX_ServiceResults (Non-Unique, Non-Clustered)
    - IX_ServiceWorkID (Non-Unique, Non-Clustered)
  - Statistics

Customer Table: CustomerID is the PK of this table. There are check constraints on phone, state, and zip. As well as a Default Value for Customer State.



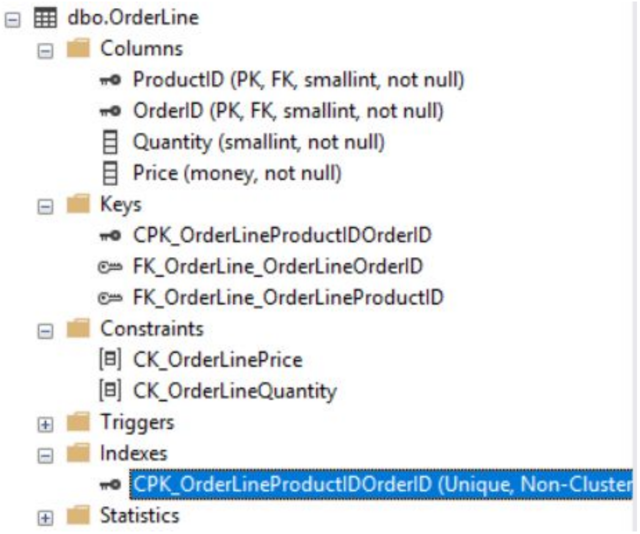
- Graph Tables
- dbo.Customer
  - Columns
    - CustomerID (PK, smallint, not null)
    - FirstName (varchar(15), not null)
    - LastName (varchar(20), not null)
    - Street (varchar(30), not null)
    - City (varchar(25), not null)
    - State (char(2), not null)
    - Zip (char(5), not null)
    - PhoneNumber (char(14), not null)
  - Keys
    - PK_CustomerCustomerID
  - Constraints
    - CK_CustomerPhone
    - CK_CustomerState
    - CK_CustomerZip
    - DF_Customer_State
  - Triggers
  - Indexes
    - IX_CustomerPhone (Clustered)
    - PK_CustomerCustomerID (Unique, Non-Clustered)
  - Statistics

Order Table: An OrderId is the primary Key for this table, with CustomerID set as a normal Foreign Key. There is a check constraint on a order placed.



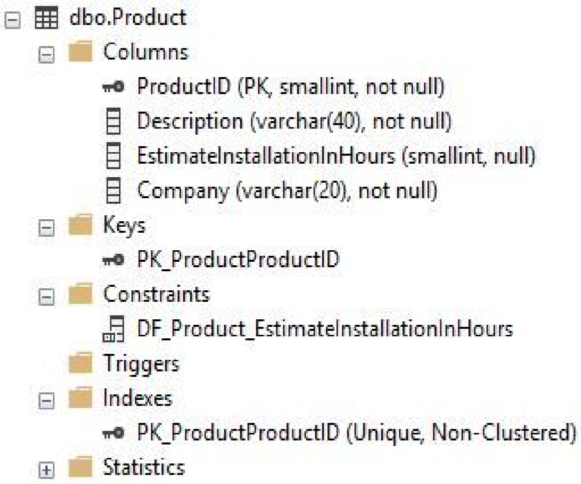
- dbo.Order
  - Columns
    - OrderID (PK, smallint, not null)
    - DateOrderPlace (date, not null)
    - Status (varchar(15), not null)
    - CustomerID (FK, smallint, not null)
  - Keys
    - PK_Order
    - FK_Order_OrderCustomerID
  - Constraints
    - CK_OrderDateOrderPlaced
  - Triggers
  - Indexes
    - PK_Order (Unique, Non-Clustered)
  - Statistics

Order Line: Product ID and OrderID make up Primary Key's but they link back to the order table & product table. There are constraints on Price & Quantity that make their values have to be greater than 0



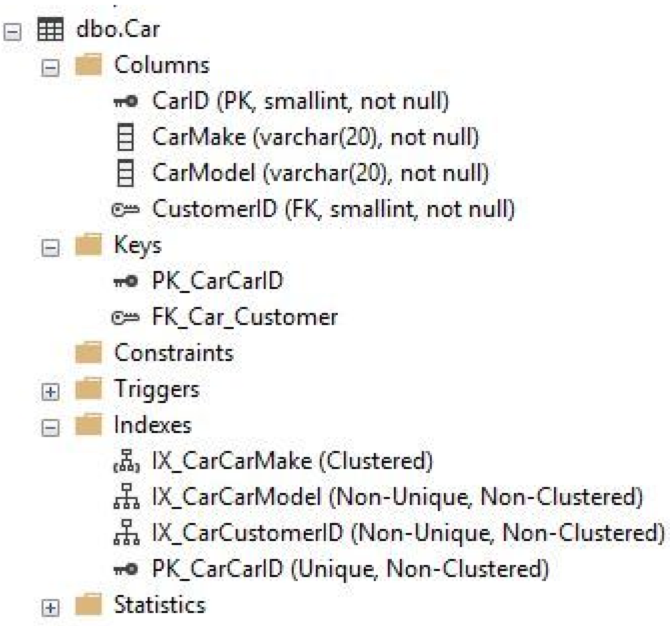
- dbo.OrderLine
  - Columns
    - ProductID (PK, FK, smallint, not null)
    - OrderID (PK, FK, smallint, not null)
    - Quantity (smallint, not null)
    - Price (money, not null)
  - Keys
    - CPK_OrderLineProductIDOrderID
    - FK_OrderLine_OrderLineOrderID
    - FK_OrderLine_OrderLineProductID
  - Constraints
    - CK_OrderLinePrice
    - CK_OrderLineQuantity
  - Triggers
  - Indexes
    - CPK_OrderLineProductIDOrderID (Unique, Non-Cluster
  - Statistics

Product Table: ProductID tracks this table. There is a constraint on estimated installation hours as well.



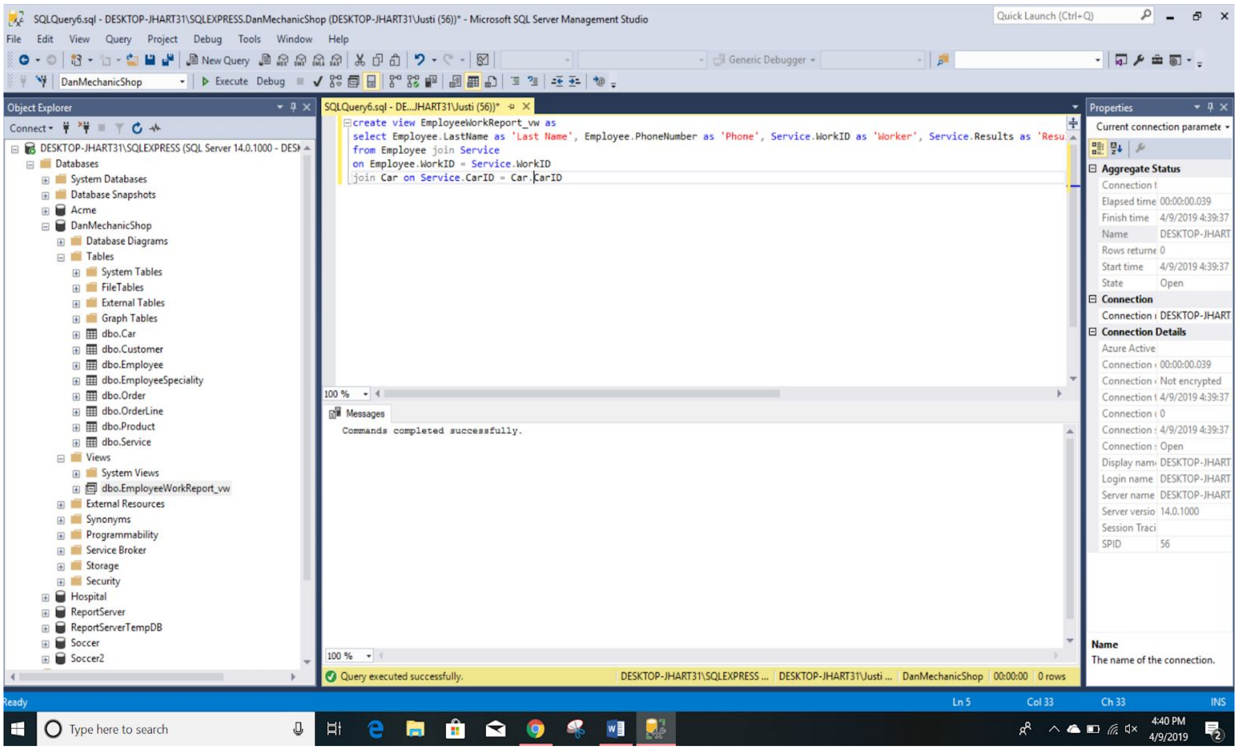
```
dbo.Product
  Columns
    ProductID (PK, smallint, not null)
    Description (varchar(40), not null)
    EstimateInstallationInHours (smallint, null)
    Company (varchar(20), not null)
  Keys
    PK_ProductProductID
  Constraints
    DF_Product_EstimateInstallationInHours
  Triggers
  Indexes
    PK_ProductProductID (Unique, Non-Clustered)
  Statistics
```

Car Table: The car table is comprised of PK CarID, with indexed columns CarMake and CarModel tracked for the shop. CustomerID is a FK for this table so the customer cars can be tracked.



```
dbo.Car
  Columns
    CarID (PK, smallint, not null)
    CarMake (varchar(20), not null)
    CarModel (varchar(20), not null)
    CustomerID (FK, smallint, not null)
  Keys
    PK_CarCarID
    FK_Car_Customer
  Constraints
  Triggers
  Indexes
    IX_CarCarMake (Clustered)
    IX_CarCarModel (Non-Unique, Non-Clustered)
    IX_CarCustomerID (Non-Unique, Non-Clustered)
    PK_CarCarID (Unique, Non-Clustered)
  Statistics
```

Table_vw



SQLQuery6.sql - DESKTOP-JHART31\SQLEXPRESS.DanMechanicShop (DESKTOP-JHART31\Justi (56))* - Microsoft SQL Server Management Studio

```
create view EmployeeWorkReport_vw as
select Employee.LastName as 'Last Name', Employee.PhoneNumber as 'Phone', Service.WorkID as 'Worker', Service.Results as 'Resu
from Employee join Service
on Employee.WorkID = Service.WorkID
join Car on Service.CarID = Car.CarID
```

```
Commands completed successfully.
```

Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS ... | DESKTOP-JHART31\Justi ... | DanMechanicShop | 00:00:00 | 0 rows

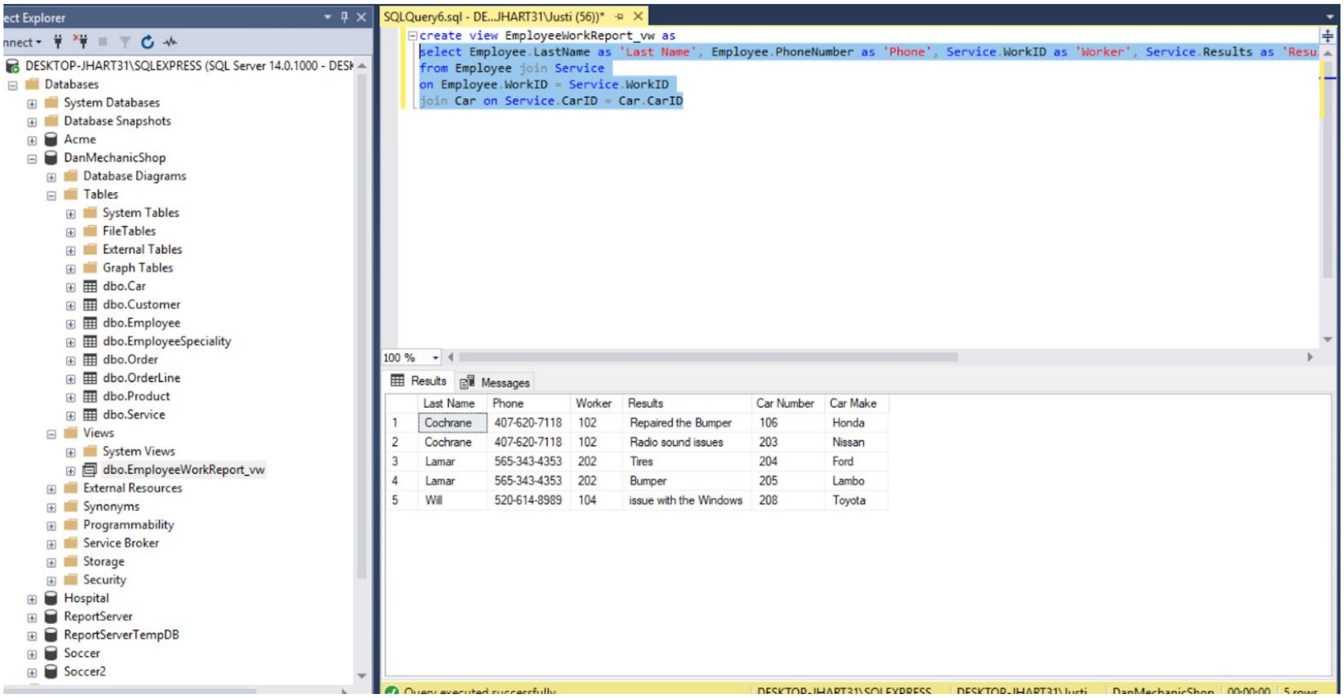
Properties: Current connection parameters

Aggregate Status
- Elapsed time: 00:00:00.039
- Finish time: 4/9/2019 4:39:37
- Name: DESKTOP-JHART
- Rows returne: 0
- Start time: 4/9/2019 4:39:37
- State: Open

Connection
- Connection: DESKTOP-JHART

Connection Details
- Connection: 00:00:00.039
- Connection: Not encrypted
- Connection: 4/9/2019 4:39:37
- Connection: 0
- Connection: 4/9/2019 4:39:37
- Connection: Open
- Display nam: DESKTOP-JHART
- Login name: DESKTOP-JHART
- Server name: DESKTOP-JHART
- Server versio: 14.0.1000
- SPID: 56

Name
The name of the connection.

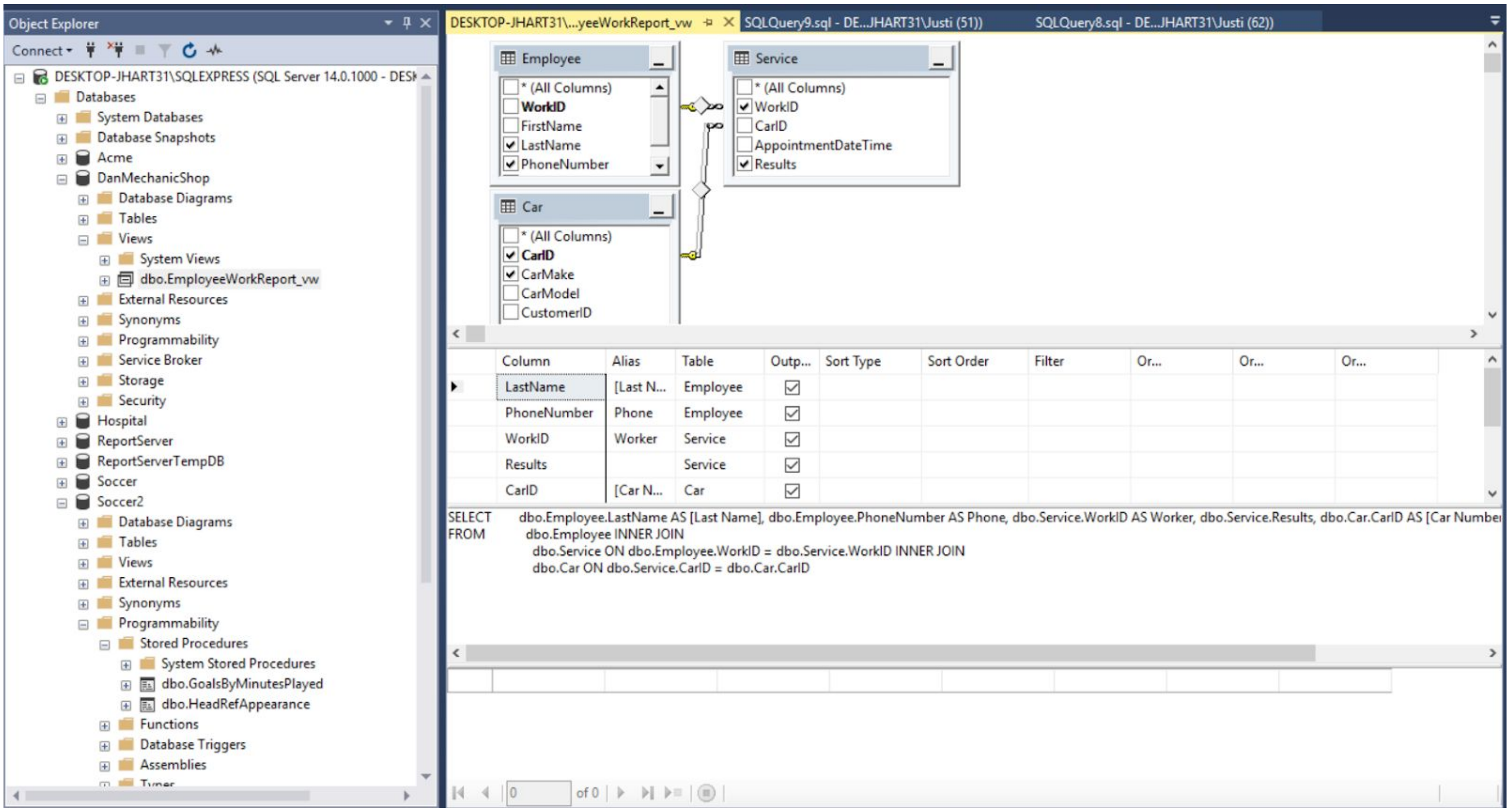


SQLQuery6.sql - DE...JHART31\Justi (56))*

```
create view EmployeeWorkReport_vw as
select Employee.LastName as 'Last Name', Employee.PhoneNumber as 'Phone', Service.WorkID as 'Worker', Service.Results as 'Resu
from Employee join Service
on Employee.WorkID = Service.WorkID
join Car on Service.CarID = Car.CarID
```

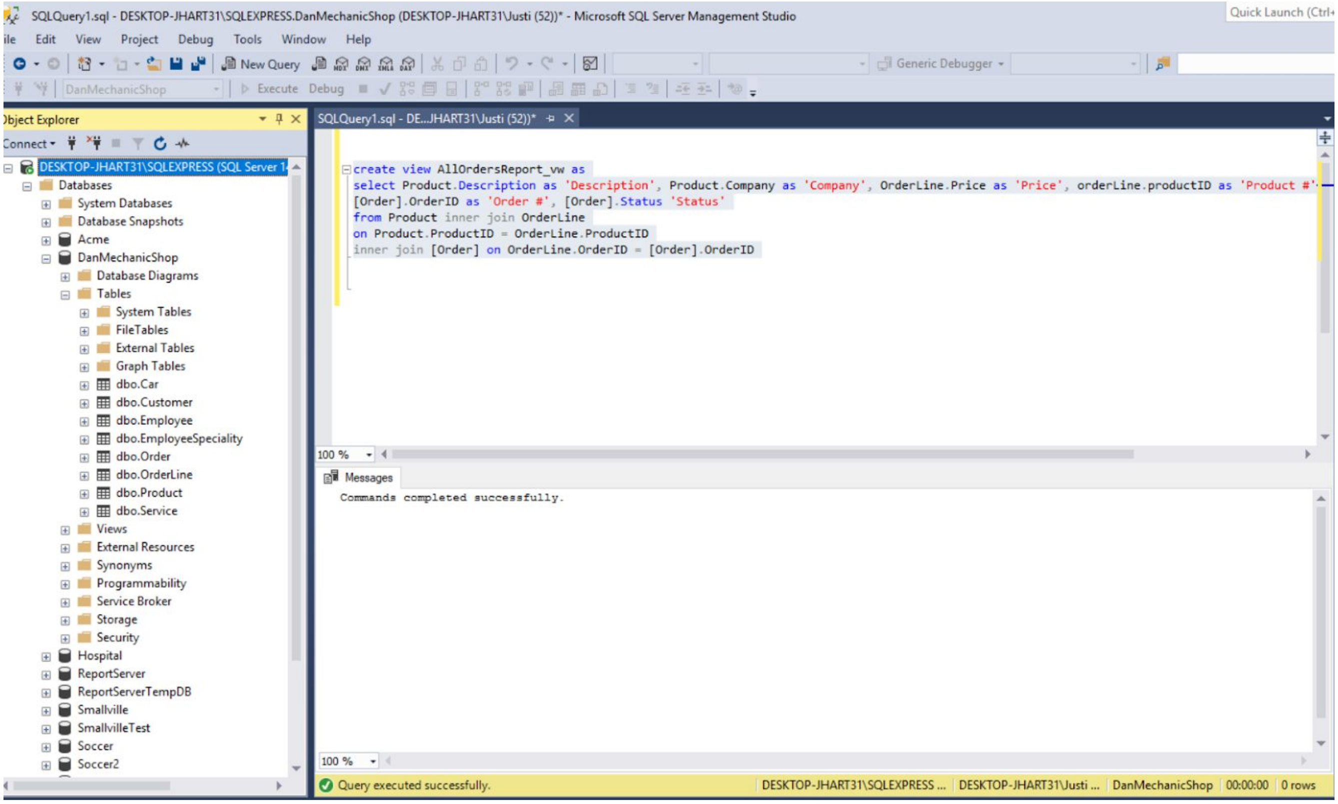
	Last Name	Phone	Worker	Results	Car Number	Car Make
1	Cochrane	407-620-7118	102	Repaired the Bumper	106	Honda
2	Cochrane	407-620-7118	102	Radio sound issues	203	Nissan
3	Lamar	565-343-4353	202	Tires	204	Ford
4	Lamar	565-343-4353	202	Bumper	205	Lambo
5	Will	520-614-8989	104	issue with the Windows	208	Toyota

Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS | DESKTOP-JHART31\Justi | DanMechanicShop | 00:00:00 | 5 rows



Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
LastName	[Last N...	Employee	☑						
PhoneNumber	Phone	Employee	☑						
WorkID	Worker	Service	☑						
Results		Service	☑						
CarID	[Car N...	Car	☑						

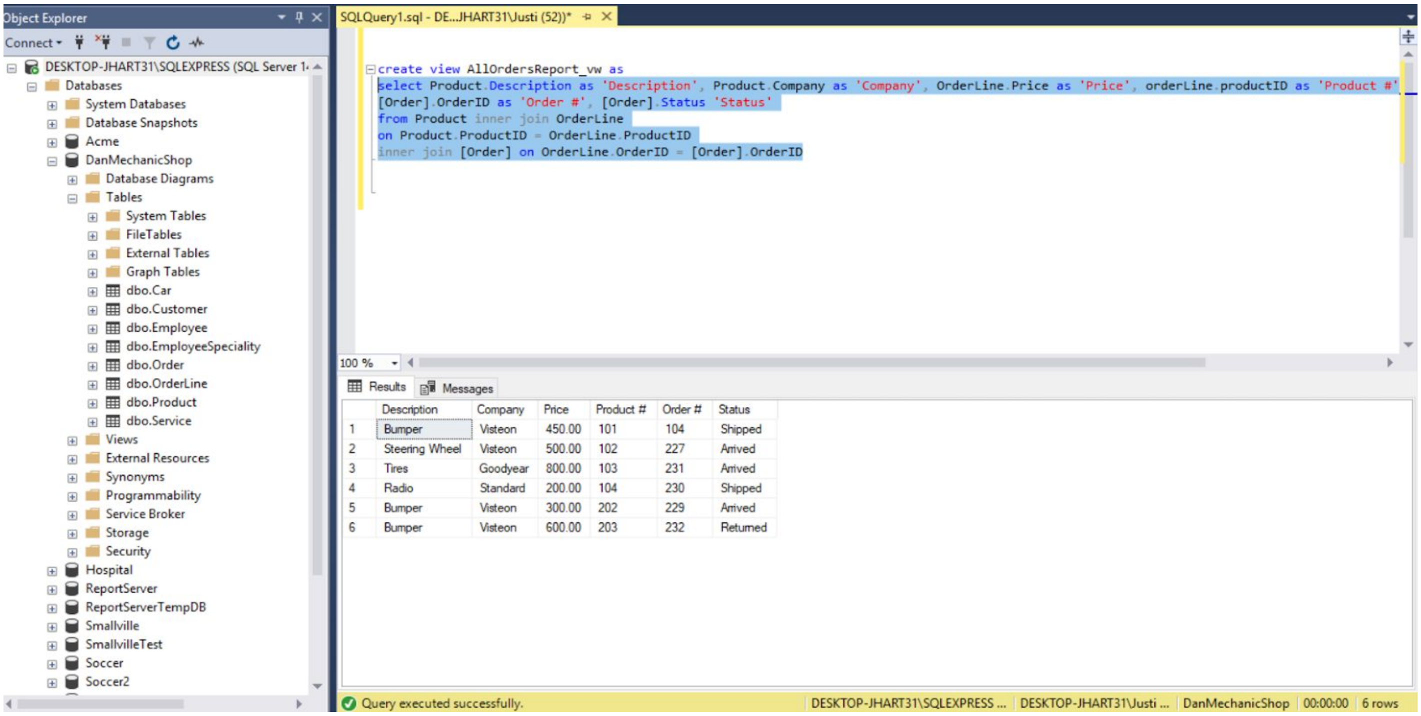
```
SELECT      dbo.Employee.LastName AS [Last Name], dbo.Employee.PhoneNumber AS Phone, dbo.Service.WorkID AS Worker, dbo.Service.Results, dbo.Car.CarID AS [Car Numbe
FROM          dbo.Employee INNER JOIN
                 dbo.Service ON dbo.Employee.WorkID = dbo.Service.WorkID INNER JOIN
                 dbo.Car ON dbo.Service.CarID = dbo.Car.CarID
```


SQLQuery1.sql - DESKTOP-JHART31\SQLEXPRESS.DanMechanicShop (DESKTOP-JHART31\Justi (52))* - Microsoft SQL Server Management Studio
Object Explorer
DanMechanicShop
Tables
Query executed successfully.
DESKTOP-JHART31\SQLEXPRESS ...
DESKTOP-JHART31\Justi ...
DanMechanicShop
00:00:00
0 rows

```
create view AllOrdersReport_vw as
select Product.Description as 'Description', Product.Company as 'Company', OrderLine.Price as 'Price', orderLine.productID as 'Product #'
[Order].OrderID as 'Order #', [Order].Status 'Status'
from Product inner join OrderLine
on Product.ProductID = OrderLine.ProductID
inner join [Order] on OrderLine.OrderID = [Order].OrderID
```

Messages

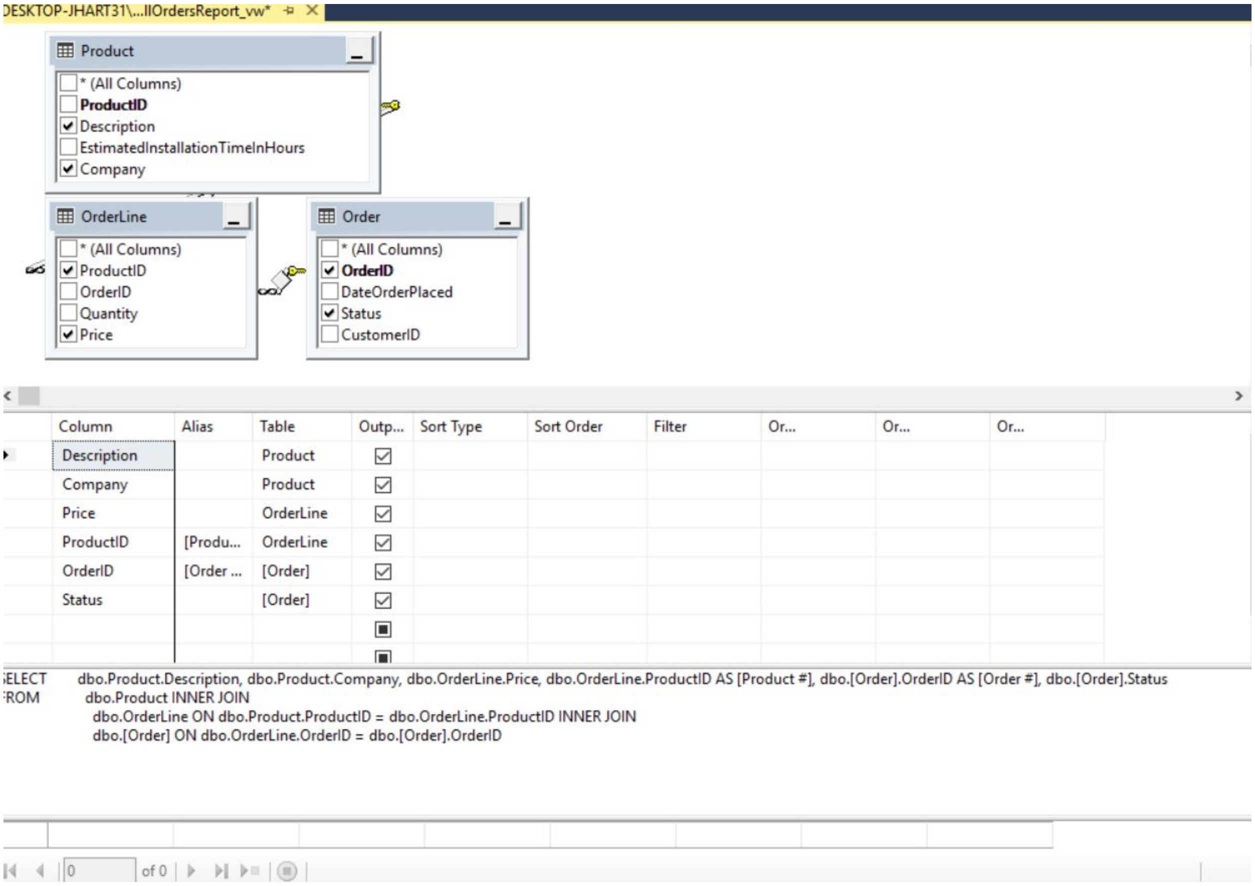
```
Commands completed successfully.
```


Object Explorer
SQLQuery1.sql - DE...JHART31\Justi (52))*
Query executed successfully.
DESKTOP-JHART31\SQLEXPRESS ...
DESKTOP-JHART31\Justi ...
DanMechanicShop
00:00:00
6 rows

```
create view AllOrdersReport_vw as
select Product.Description as 'Description', Product.Company as 'Company', OrderLine.Price as 'Price', orderLine.productID as 'Product #'
[Order].OrderID as 'Order #', [Order].Status 'Status'
from Product inner join OrderLine
on Product.ProductID = OrderLine.ProductID
inner join [Order] on OrderLine.OrderID = [Order].OrderID
```

Results

	Description	Company	Price	Product #	Order #	Status
1	Bumper	Visteon	450.00	101	104	Shipped
2	Steering Wheel	Visteon	500.00	102	227	Arrived
3	Tires	Goodyear	800.00	103	231	Arrived
4	Radio	Standard	200.00	104	230	Shipped
5	Bumper	Visteon	300.00	202	229	Arrived
6	Bumper	Visteon	600.00	203	232	Returned



DESKTOP-JHART31\...llOrdersReport_vw*

Product
- □ * (All Columns)
- □ **ProductID**
- ☑ Description
- □ EstimatedInstallationTimeInHours
- ☑ Company

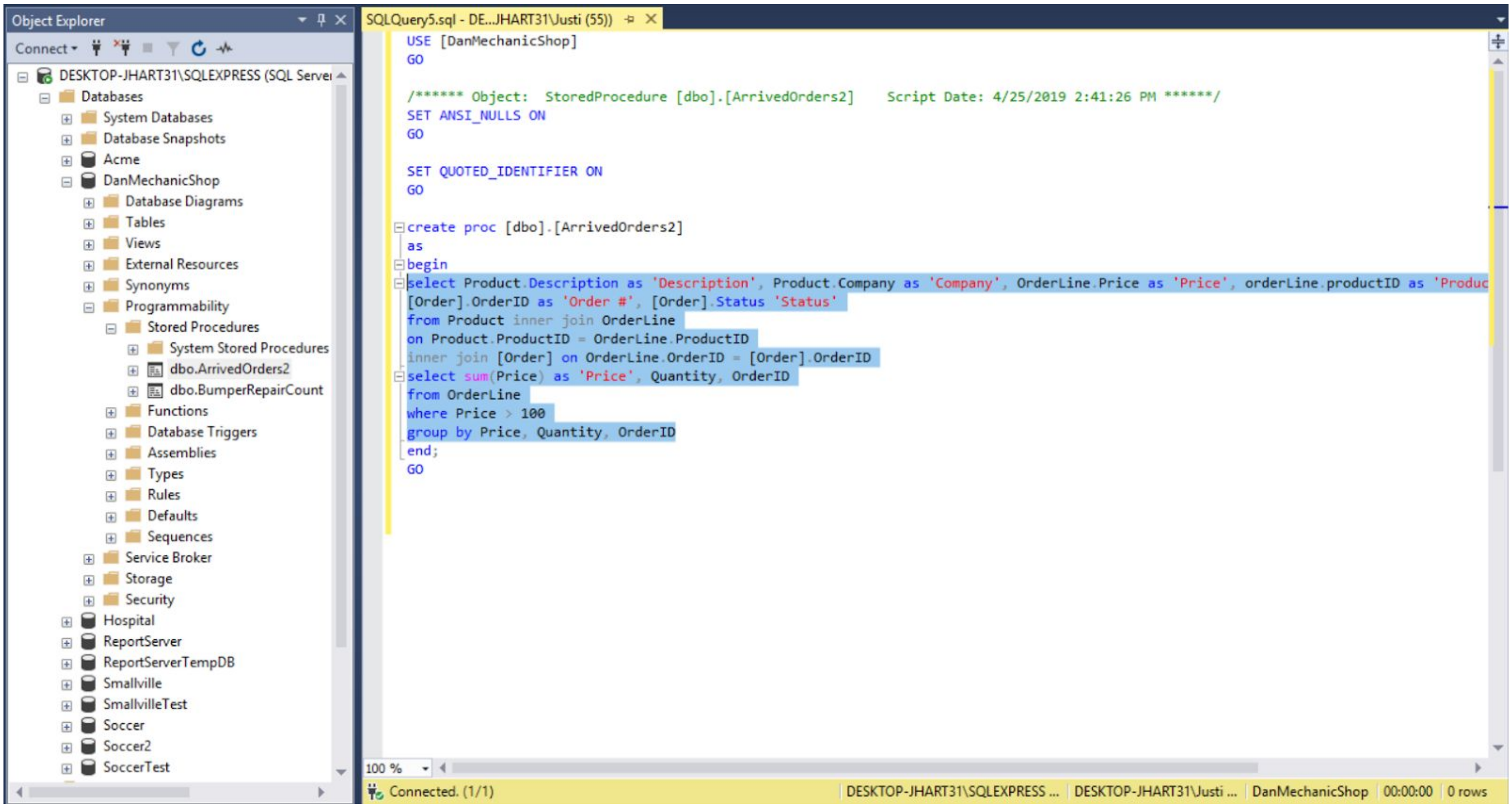
OrderLine
- □ * (All Columns)
- ☑ ProductID
- □ OrderID
- □ Quantity
- ☑ Price

Order
- □ * (All Columns)
- ☑ **OrderID**
- □ DateOrderPlaced
- ☑ Status
- □ CustomerID

Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
Description		Product	☑						
Company		Product	☑						
Price		OrderLine	☑						
ProductID	[Produ...	OrderLine	☑						
OrderID	[Order ...	[Order]	☑						
Status		[Order]	☑						
			▣						
			▣						

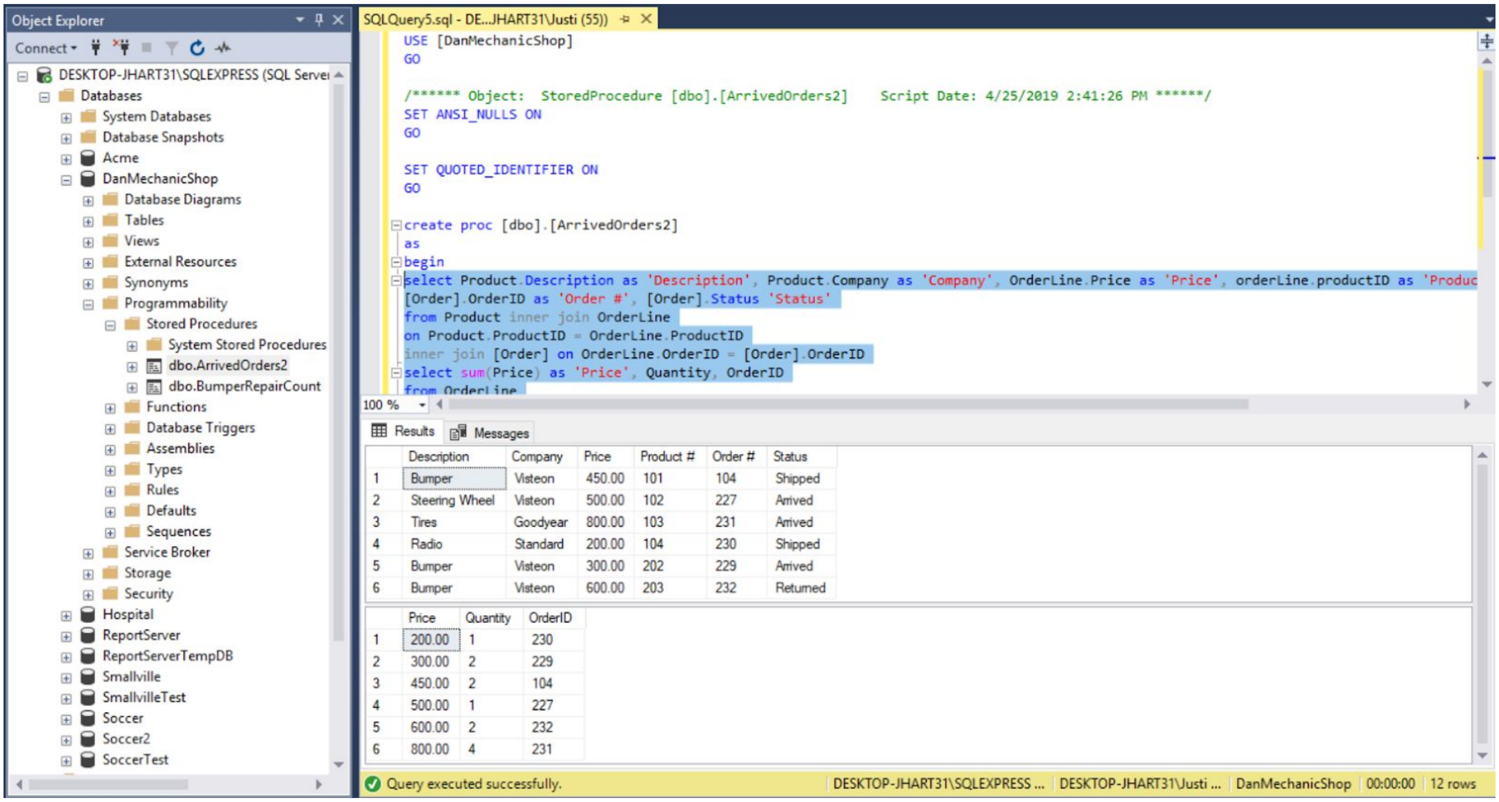
```
SELECT    dbo.Product.Description, dbo.Product.Company, dbo.OrderLine.Price, dbo.OrderLine.ProductID AS [Product #], dbo.[Order].OrderID AS [Order #], dbo.[Order].Status
FROM       dbo.Product INNER JOIN
             dbo.OrderLine ON dbo.Product.ProductID = dbo.OrderLine.ProductID INNER JOIN
             dbo.[Order] ON dbo.OrderLine.OrderID = dbo.[Order].OrderID
```

SPROCS



```
USE [DanMechanicShop]
GO

/****** Object:  StoredProcedure [dbo].[ArrivedOrders2]    Script Date: 4/25/2019 2:41:26 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

create proc [dbo].[ArrivedOrders2]
as
begin
select Product.Description as 'Description', Product.Company as 'Company', OrderLine.Price as 'Price', orderLine.productID as 'Produc
[Order].OrderID as 'Order #', [Order].Status 'Status'
from Product inner join OrderLine
on Product.ProductID = OrderLine.ProductID
inner join [Order] on OrderLine.OrderID = [Order].OrderID
select sum(Price) as 'Price', Quantity, OrderID
from OrderLine
where Price > 100
group by Price, Quantity, OrderID
end;
GO
```

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

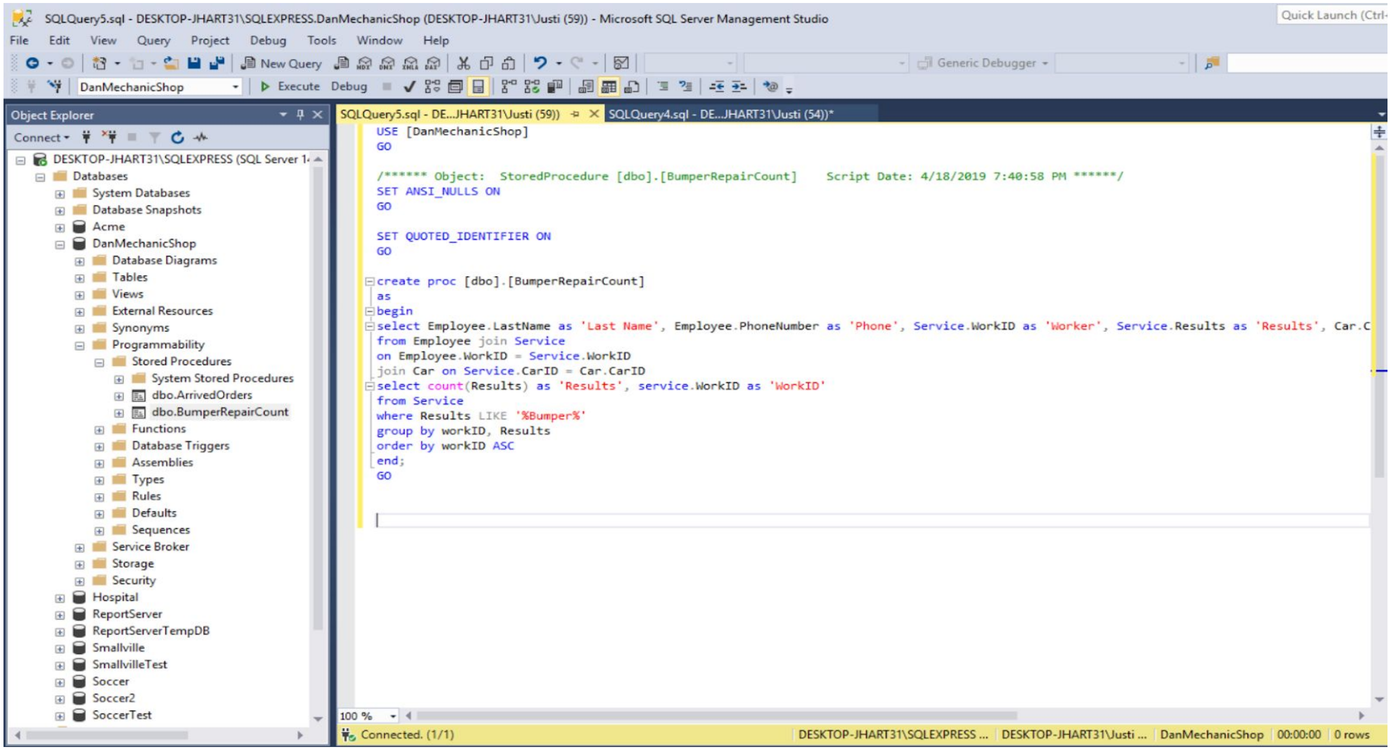


```
USE [DanMechanicShop]
GO

/****** Object:  StoredProcedure [dbo].[ArrivedOrders2]    Script Date: 4/25/2019 2:41:26 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

create proc [dbo].[ArrivedOrders2]
as
begin
select Product.Description as 'Description', Product.Company as 'Company', OrderLine.Price as 'Price', orderLine.productID as 'Produc
[Order].OrderID as 'Order #', [Order].Status 'Status'
from Product inner join OrderLine
on Product.ProductID = OrderLine.ProductID
inner join [Order] on OrderLine.OrderID = [Order].OrderID
select sum(Price) as 'Price', Quantity, OrderID
from OrderLine
```

	Description	Company	Price	Product #	Order #	Status
1	Bumper	Visteon	450.00	101	104	Shipped
2	Steering Wheel	Visteon	500.00	102	227	Arrived
3	Tires	Goodyear	800.00	103	231	Arrived
4	Radio	Standard	200.00	104	230	Shipped
5	Bumper	Visteon	300.00	202	229	Arrived
6	Bumper	Visteon	600.00	203	232	Returned

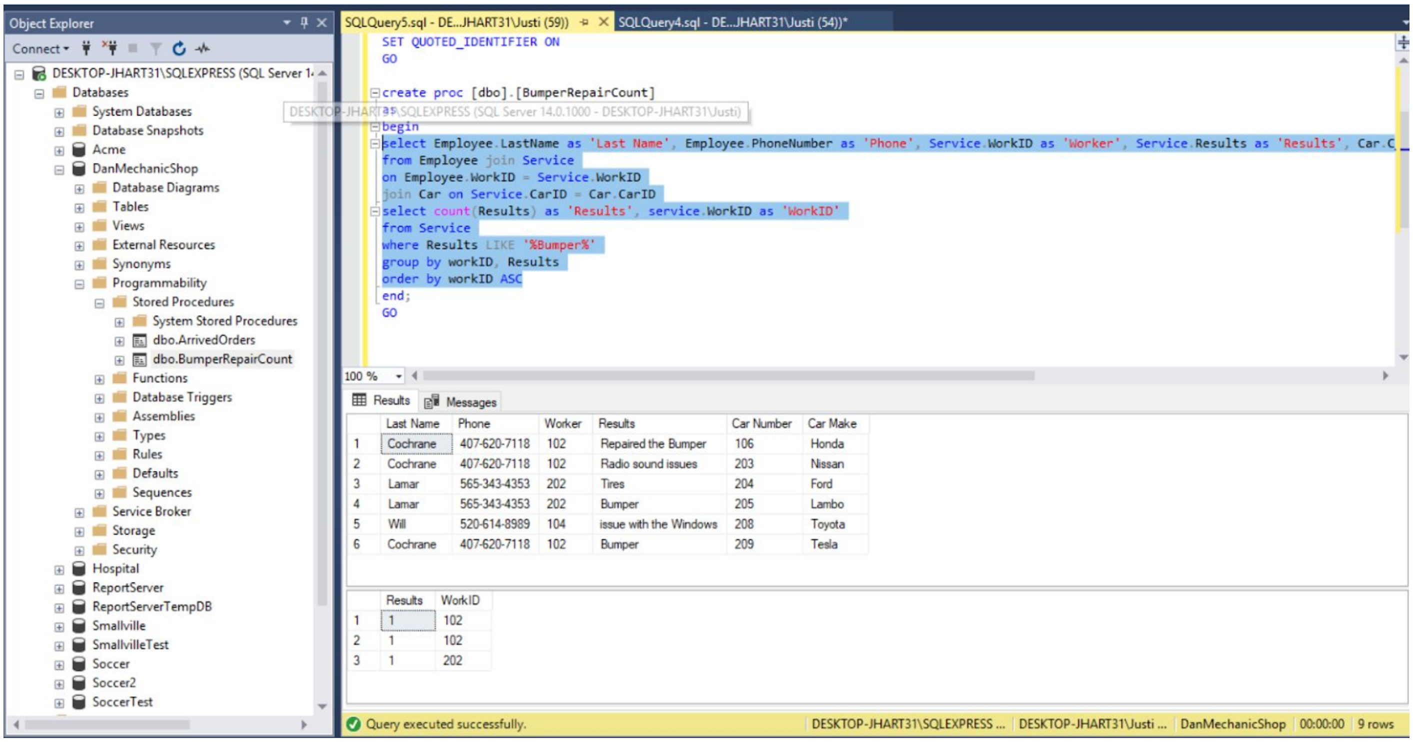
	Price	Quantity	OrderID
1	200.00	1	230
2	300.00	2	229
3	450.00	2	104
4	500.00	1	227
5	600.00	2	232
6	800.00	4	231

Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS ... | DESKTOP-JHART31\Justi ... | DanMechanicShop | 00:00:00 | 12 rows


SQLQuery5.sql - DESKTOP-JHART31\SQLEXPRESS.DanMechanicShop (DESKTOP-JHART31\Justi (59)) - Microsoft SQL Server Management Studio

```sql
USE [DanMechanicShop]
GO

/****** Object:  StoredProcedure [dbo].[BumperRepairCount]    Script Date: 4/18/2019 7:40:58 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

create proc [dbo].[BumperRepairCount]
as
begin
select Employee.LastName as 'Last Name', Employee.PhoneNumber as 'Phone', Service.WorkID as 'Worker', Service.Results as 'Results', Car.C
from Employee join Service
on Employee.WorkID = Service.WorkID
join Car on Service.CarID = Car.CarID
select count(Results) as 'Results', service.WorkID as 'WorkID'
from Service
where Results LIKE '%Bumper%'
group by workID, Results
order by workID ASC
end;
GO
```

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



DESKTOP-JHART31\SQLEXPRESS (SQL Server 14.0.1000 - DESKTOP-JHART31\Justi)

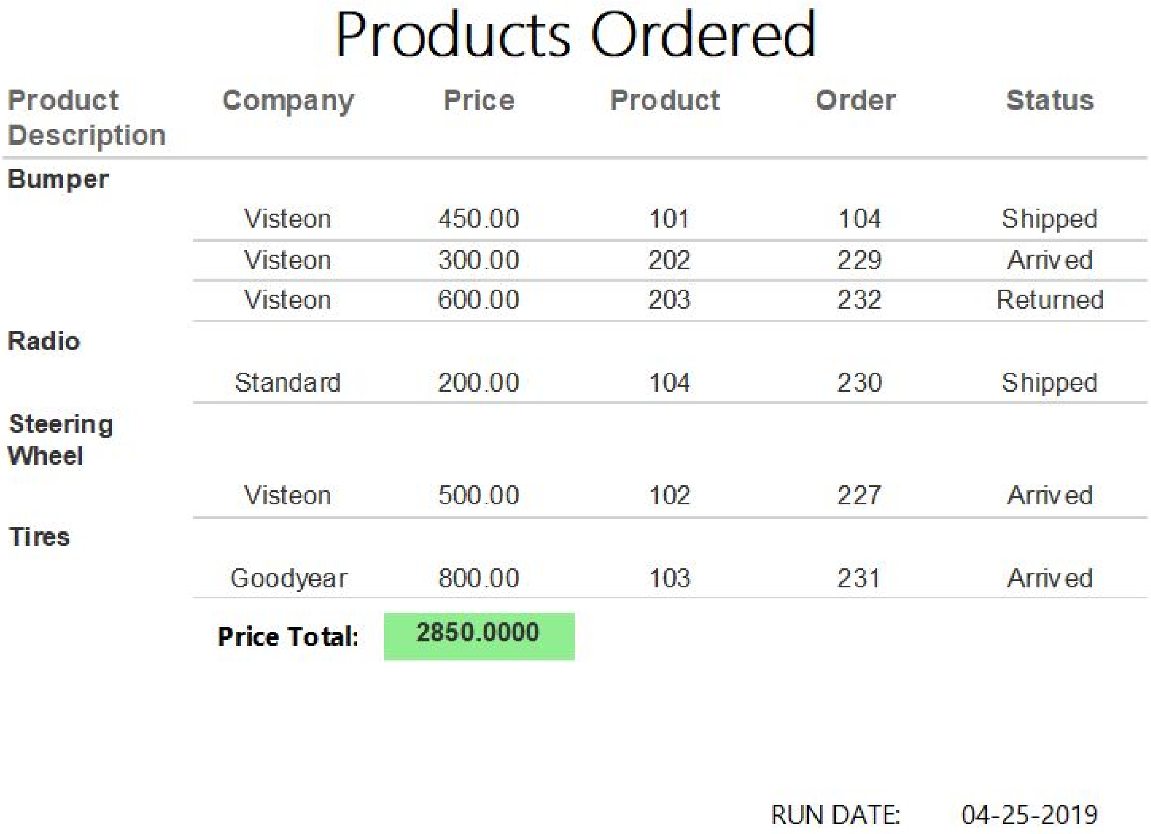
```sql
SET QUOTED_IDENTIFIER ON
GO

create proc [dbo].[BumperRepairCount]
as
begin
select Employee.LastName as 'Last Name', Employee.PhoneNumber as 'Phone', Service.WorkID as 'Worker', Service.Results as 'Results', Car.C
from Employee join Service
on Employee.WorkID = Service.WorkID
join Car on Service.CarID = Car.CarID
select count(Results) as 'Results', service.WorkID as 'WorkID'
from Service
where Results LIKE '%Bumper%'
group by workID, Results
order by workID ASC
end;
GO
```

	Last Name	Phone	Worker	Results	Car Number	Car Make
1	Cochrane	407-620-7118	102	Repaired the Bumper	106	Honda
2	Cochrane	407-620-7118	102	Radio sound issues	203	Nissan
3	Lamar	565-343-4353	202	Tires	204	Ford
4	Lamar	565-343-4353	202	Bumper	205	Lambo
5	Will	520-614-8989	104	issue with the Windows	208	Toyota
6	Cochrane	407-620-7118	102	Bumper	209	Tesla

	Results	WorkID
1	1	102
2	1	102
3	1	202

Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS ... | DESKTOP-JHART31\Justi ... | DanMechanicShop | 00:00:00 | 9 rows

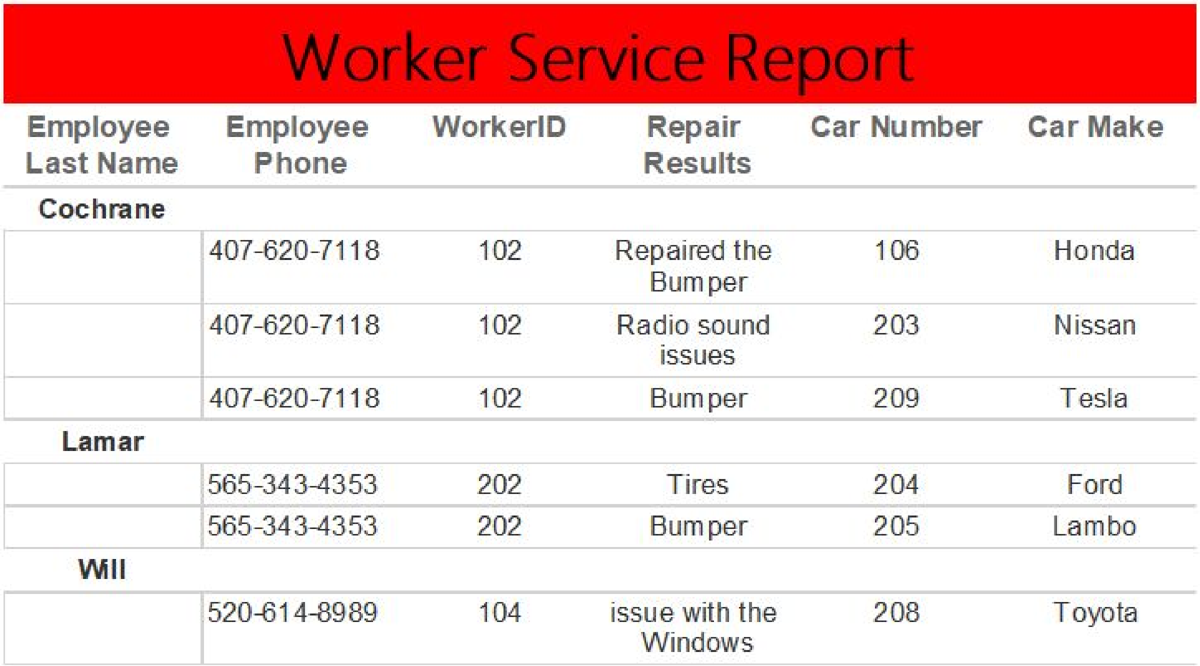
SSRS Reports



Products Ordered

Product Description	Company	Price	Product	Order	Status
Bumper					
	Visteon	450.00	101	104	Shipped
	Visteon	300.00	202	229	Arrived
	Visteon	600.00	203	232	Returned
Radio					
	Standard	200.00	104	230	Shipped
Steering Wheel					
	Visteon	500.00	102	227	Arrived
Tires					
	Goodyear	800.00	103	231	Arrived
	Price Total:	**2850.0000**			

RUN DATE: 04-25-2019



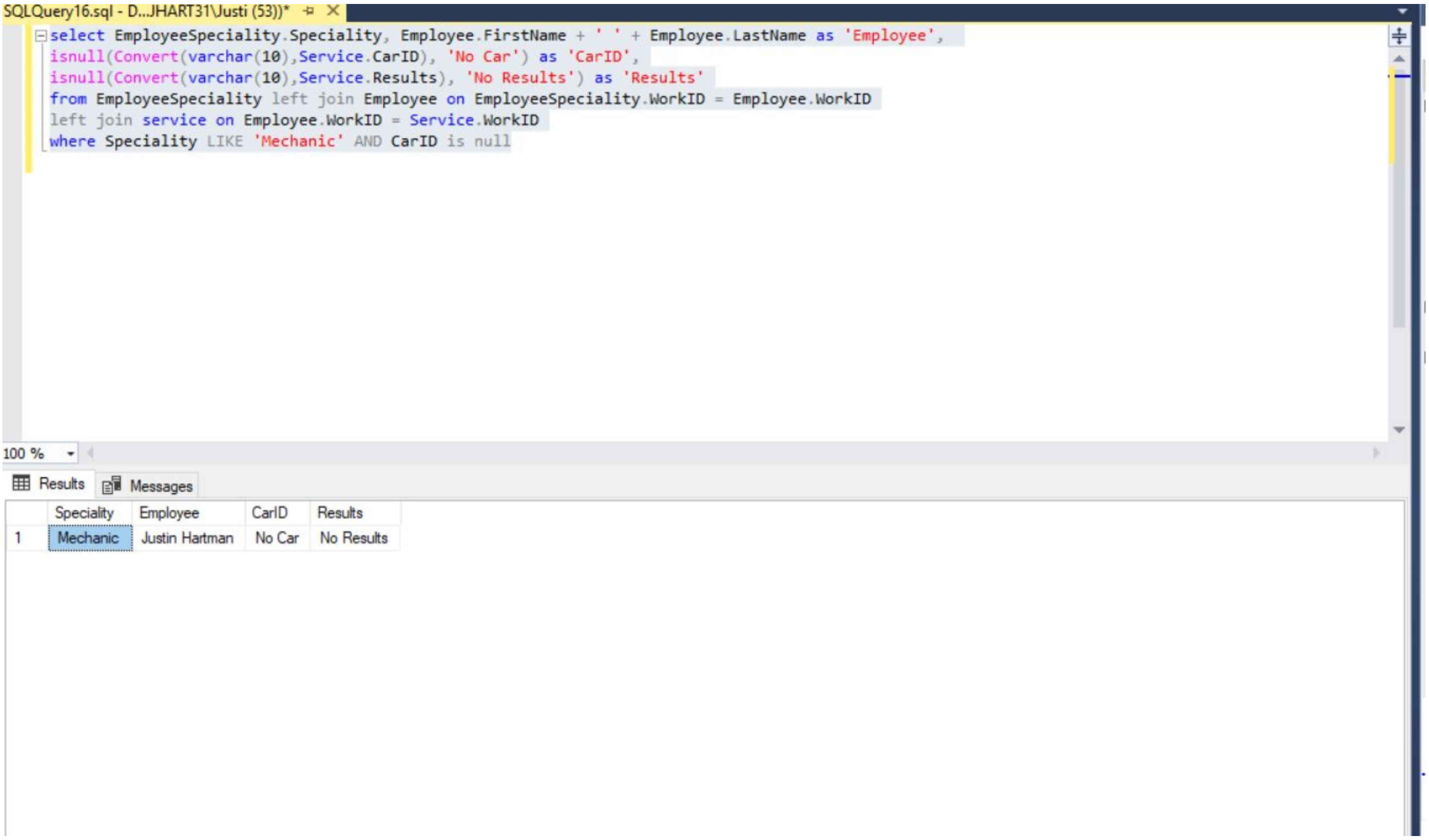
Worker Service Report

Employee Last Name	Employee Phone	WorkerID	Repair Results	Car Number	Car Make
Cochrane					
	407-620-7118	102	Repaired the Bumper	106	Honda
	407-620-7118	102	Radio sound issues	203	Nissan
	407-620-7118	102	Bumper	209	Tesla
Lamar					
	565-343-4353	202	Tires	204	Ford
	565-343-4353	202	Bumper	205	Lambo
Will					
	520-614-8989	104	issue with the Windows	208	Toyota

User Acceptance Test Queries

Question: Show me all the employee's who are new mechanics and have not worked on a car yet.

Answer:



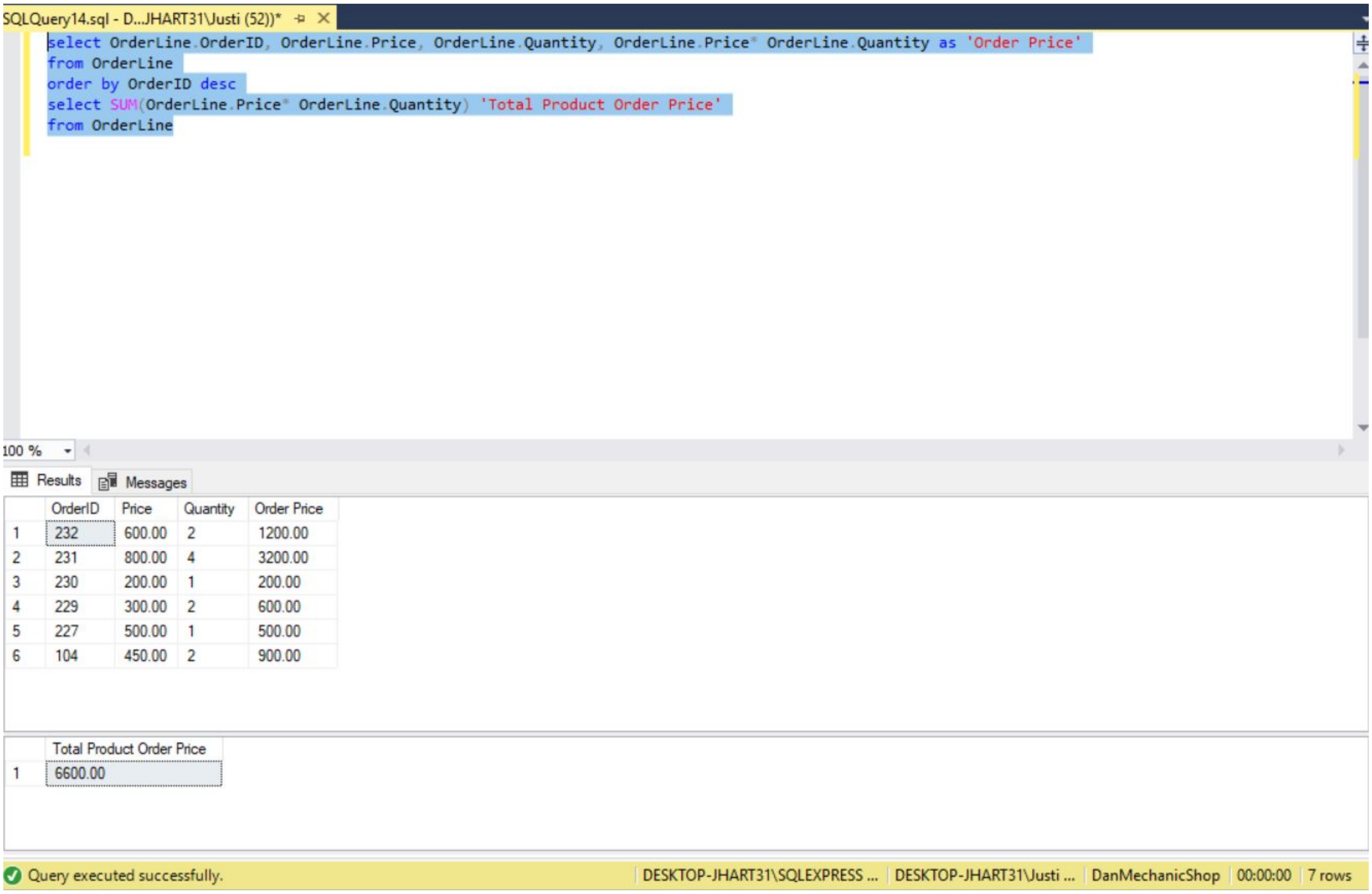
SQLQuery16.sql - D...JHART31\Justi (53))*

```
select EmployeeSpeciality.Speciality, Employee.FirstName + ' ' + Employee.LastName as 'Employee',
isnull(Convert(varchar(10),Service.CarID), 'No Car') as 'CarID',
isnull(Convert(varchar(10),Service.Results), 'No Results') as 'Results'
from EmployeeSpeciality left join Employee on EmployeeSpeciality.WorkID = Employee.WorkID
left join service on Employee.WorkID = Service.WorkID
where Speciality LIKE 'Mechanic' AND CarID is null
```

	Speciality	Employee	CarID	Results
1	Mechanic	Justin Hartman	No Car	No Results

Question: We recently ran a report on total product price per each item. We now want to factor in quantity. List the OrderID, Price, Quantity and a total order price using quantity.

Answers:



```
select OrderLine.OrderID, OrderLine.Price, OrderLine.Quantity, OrderLine.Price* OrderLine.Quantity as 'Order Price'
from OrderLine
order by OrderID desc
select SUM(OrderLine.Price* OrderLine.Quantity) 'Total Product Order Price'
from OrderLine
```

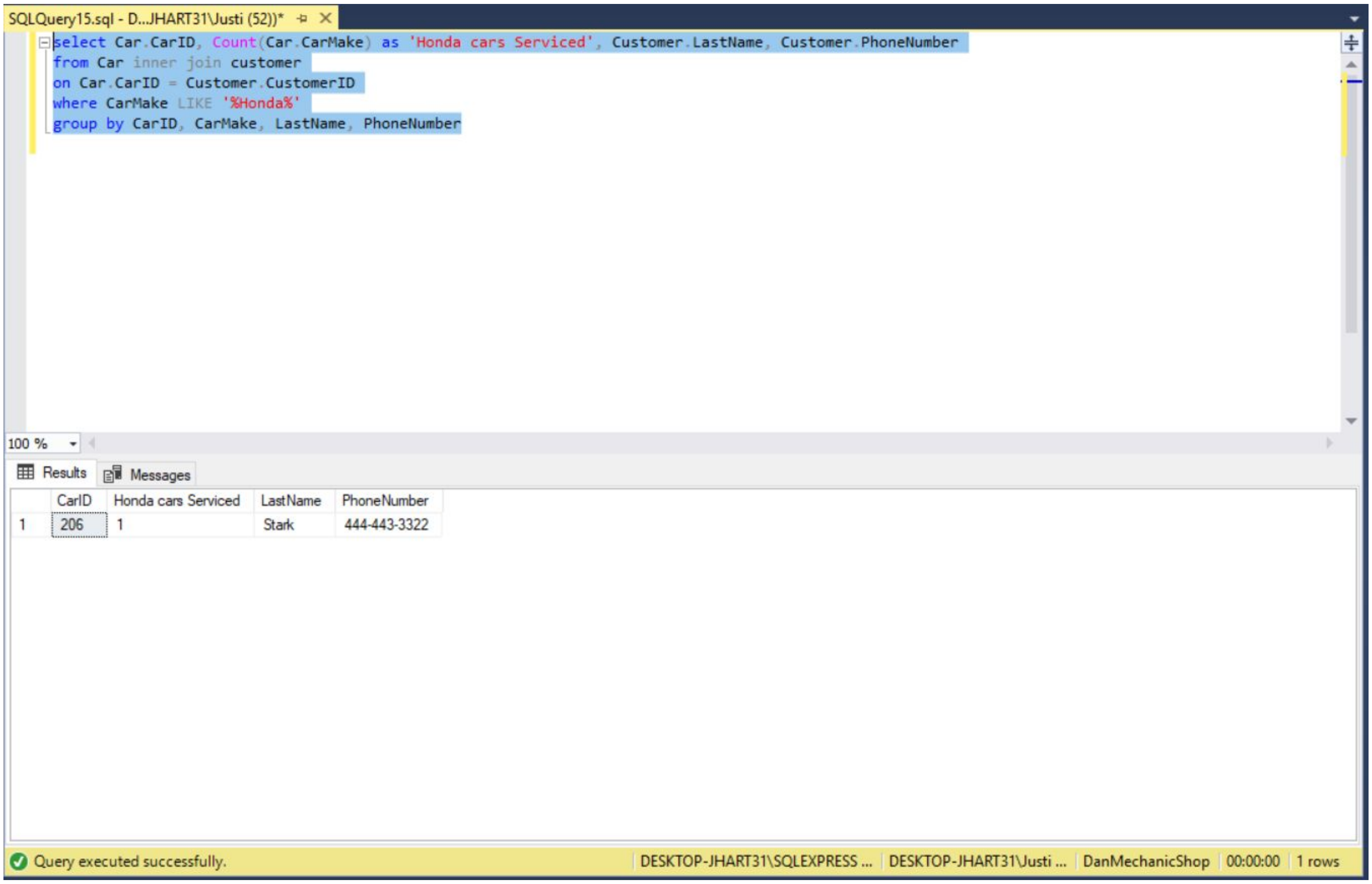
	OrderID	Price	Quantity	Order Price
1	232	600.00	2	1200.00
2	231	800.00	4	3200.00
3	230	200.00	1	200.00
4	229	300.00	2	600.00
5	227	500.00	1	500.00
6	104	450.00	2	900.00

	Total Product Order Price
1	6600.00

Query executed successfully. | DESKTOP-JHART31\SQLEXPRESS ... | DESKTOP-JHART31\Justi ... | DanMechanicShop | 00:00:00 | 7 rows

Question: Show me a count of all Honda’s that have been serviced, give me the customer last name and phone number.

Answer:



```sql
select Car.CarID, Count(Car.CarMake) as 'Honda cars Serviced', Customer.LastName, Customer.PhoneNumber
from Car inner join customer
on Car.CarID = Customer.CustomerID
where CarMake LIKE '%Honda%'
group by CarID, CarMake, LastName, PhoneNumber
```

	CarID	Honda cars Serviced	LastName	PhoneNumber
1	206	1	Stark	444-443-3322

```
Query executed successfully.    DESKTOP-JHART31\SQLEXPRESS ... | DESKTOP-JHART31\Justi ... | DanMechanicShop | 00:00:00 | 1 rows
```