

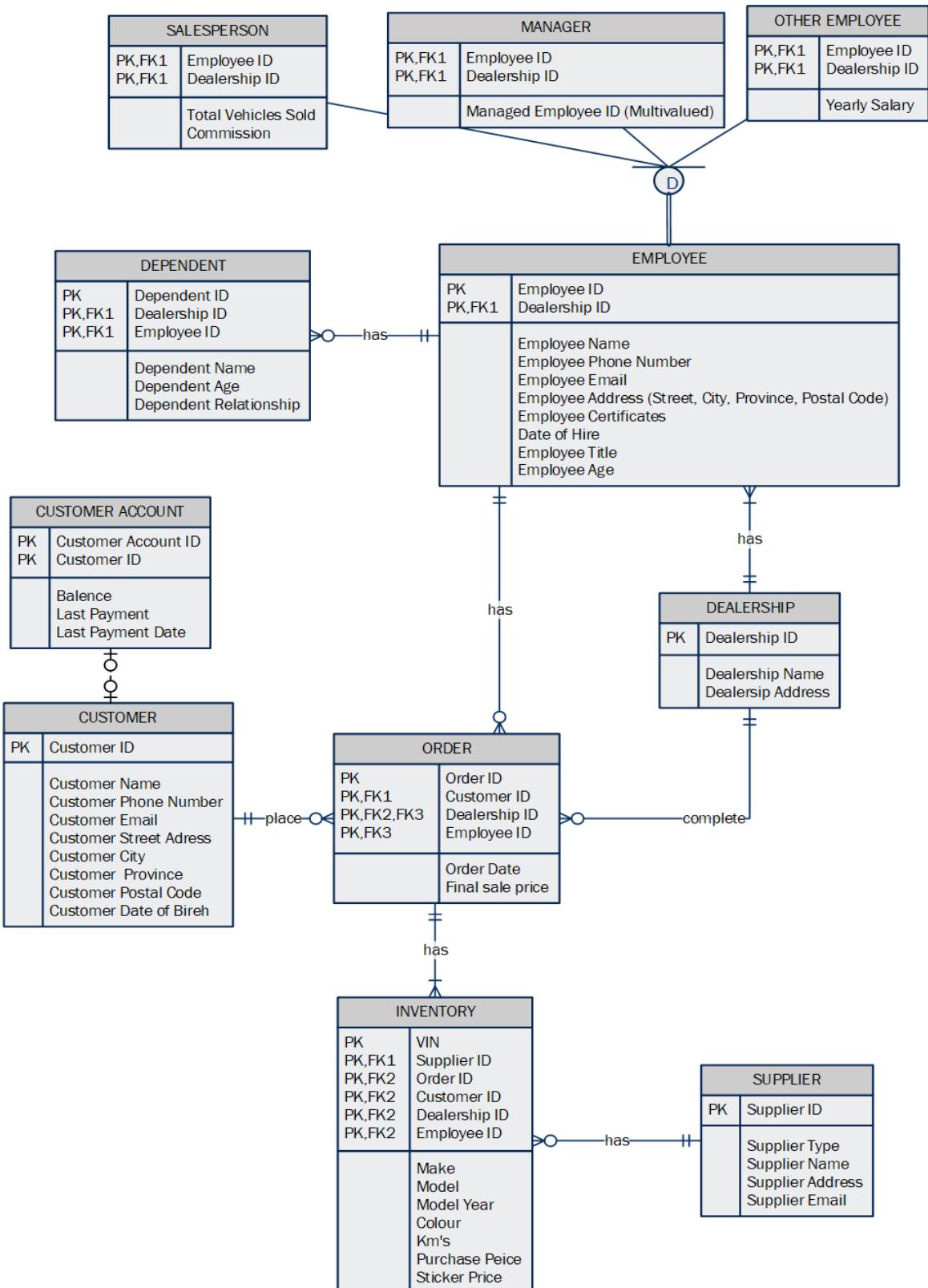
Automotive Dealership Database Design

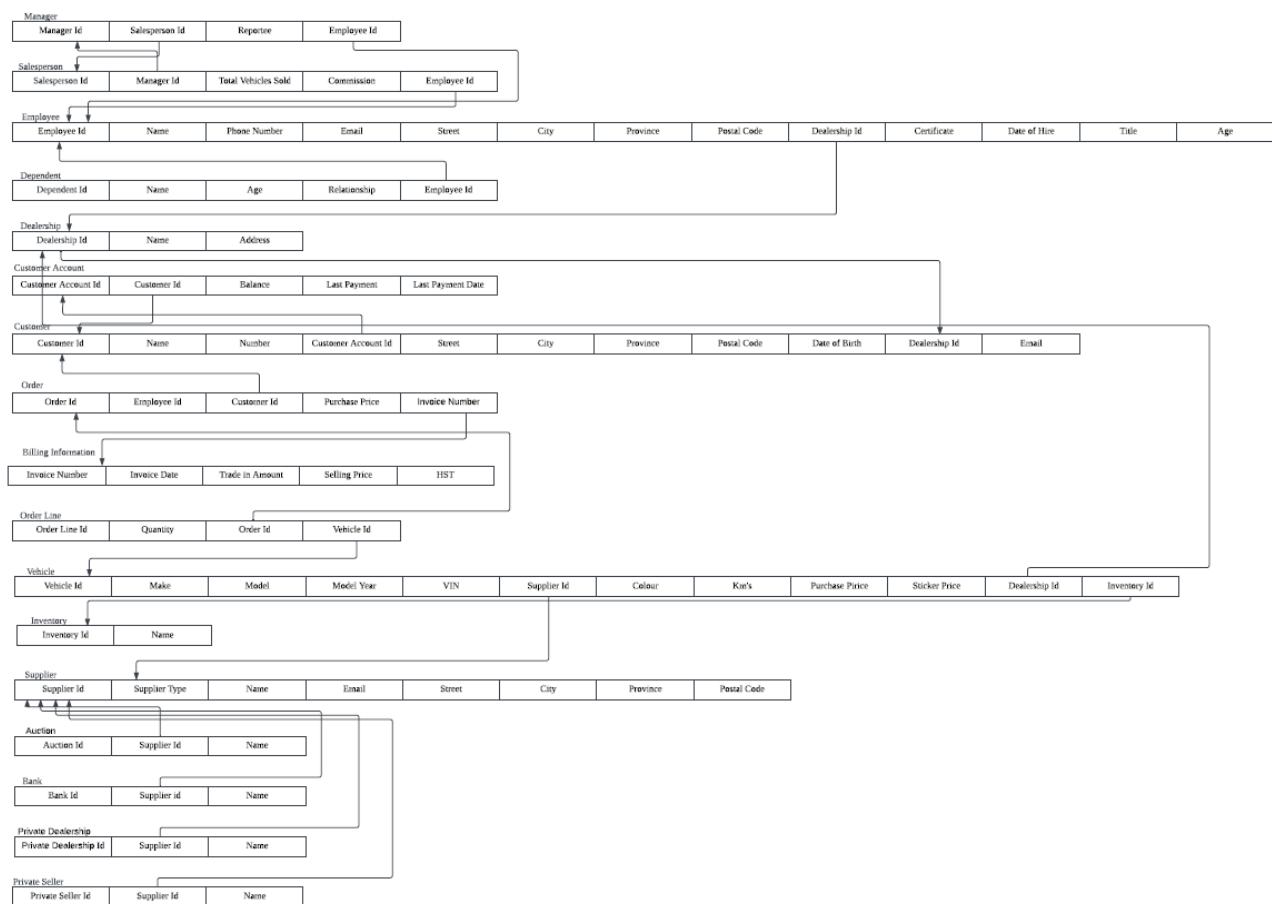
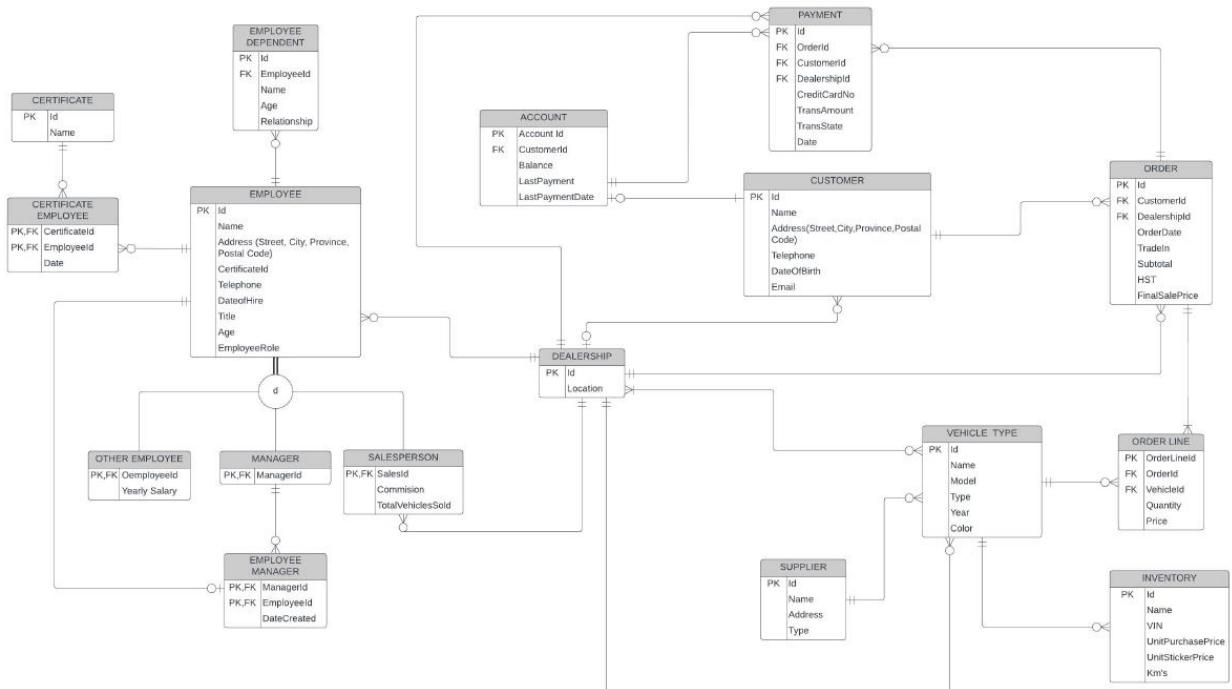
-Edison Wang

Overview:

- Design a database management system for a chain of stores that sell used vehicles. Owner requires this system to track details about inventory, employees, sales and customers, etc.
- Database Name: “WonderfulWheels”.
- Star Schema

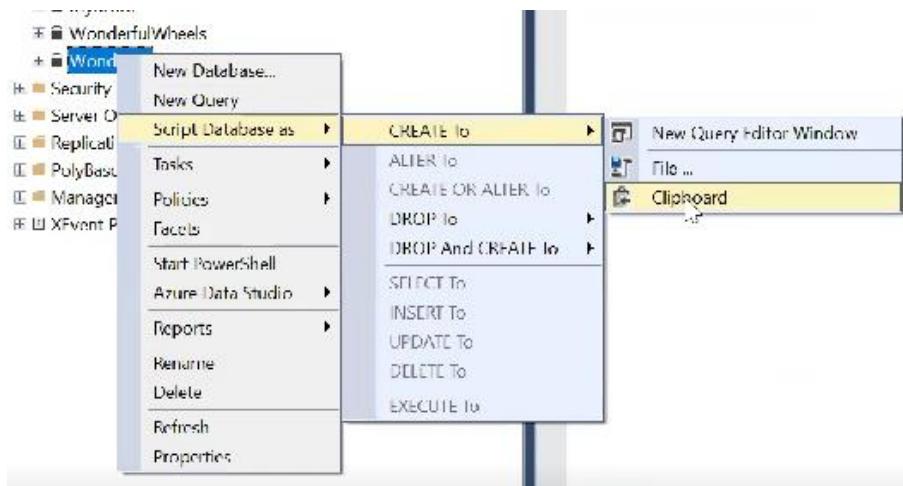
Step1: Design Entity Relationship diagrams:





Step2: Database Creation with constraints:

1. Right click “Databases”: New Database -> WonderfulWheels -> OK



2. Create tables:

WonderfulWheels -> Tables -> New Table -> Fill column names -> DataType -> [NULL]

Right click on column name -> Set as primary key

Continue filling other columns

3. Setting Default Value: (column properties)

The screenshot displays the 'Table Designer' window for a table named 'dbo.Table_1'. The 'StreetAddress' column is selected. In the 'Column Properties' pane, the 'Default Value or Binding' property is set to 'NULL'. Other properties shown include '(Name)', 'Allow Nulls' (set to 'Yes'), 'Data Type' (set to 'varchar'), and 'Length' (set to '50').

Column Name	Data Type	Allow Nulls
LocationID	int	<input type="checkbox"/>
StreetAddress	varchar(50)	<input checked="" type="checkbox"/>
City	varchar(50)	<input checked="" type="checkbox"/>
Province	varchar(50)	<input checked="" type="checkbox"/>
PostalCode	varchar(50)	<input checked="" type="checkbox"/>

Column Properties

(General)

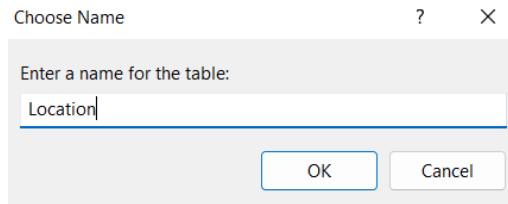
- (Name): StreetAddress
- Allow Nulls: Yes
- Data Type: varchar
- Default Value or Binding: NULL
- Length: 50

Table Designer

4. If need auto generated IDs:

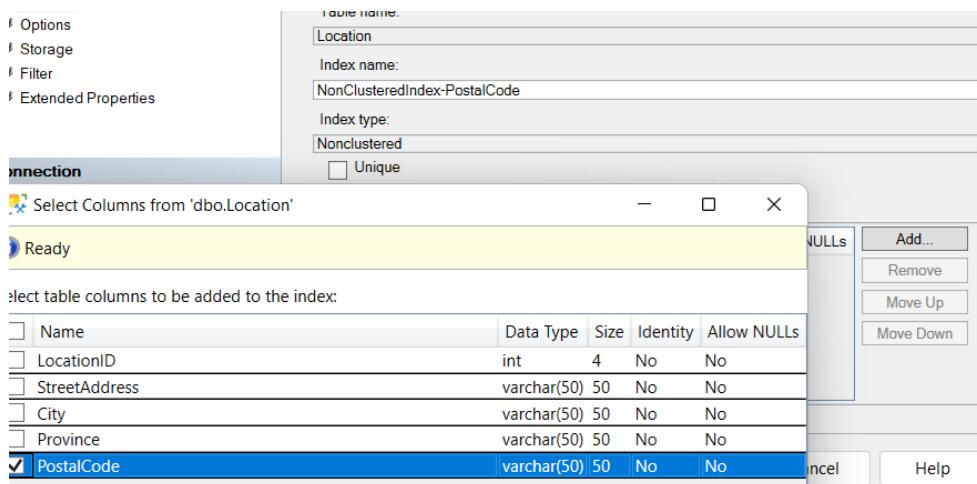
Click on ColName -> Column Properties -> Identity Specification -> Double Click (YES) -> Set Seed(Starting Value) and Increments.

5. Close window when ready -> Rename the table -> Refresh



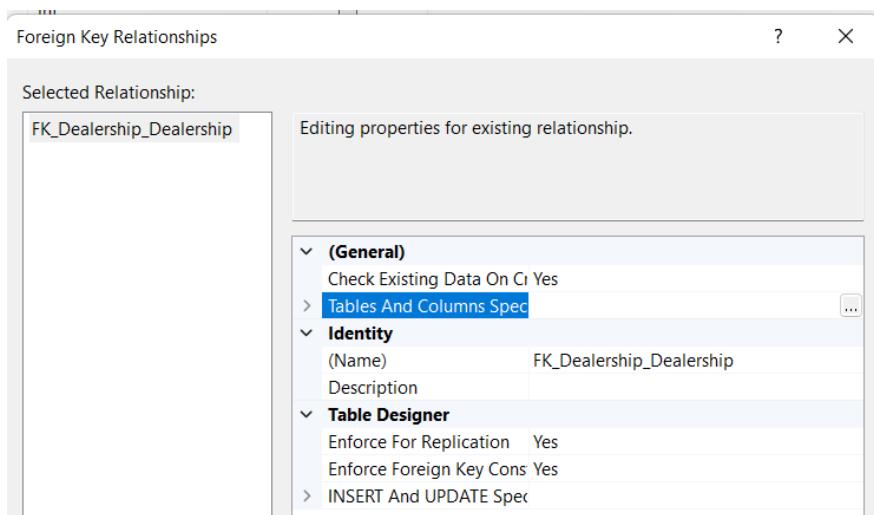
6. Index:

Double Click on Table -> Right click Indexes -> New [Non Clustered Index] -> Rename Index -> Add the desired columns:

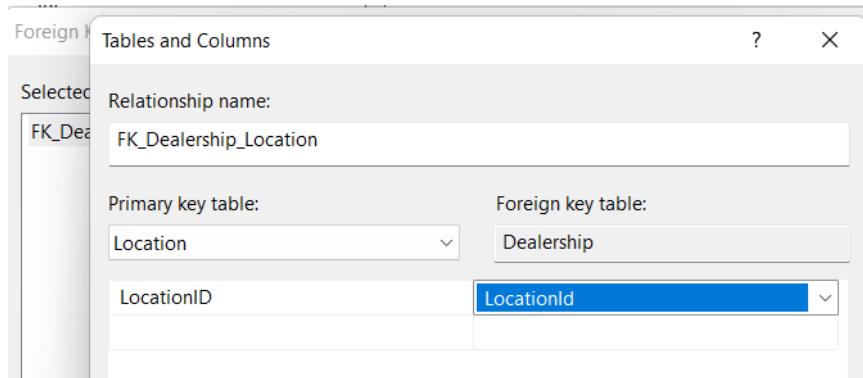


7. Setting Foreign Keys: (Column can not be NULL)

Inside the table with foreign key: right click in black -> Relationships (Add)

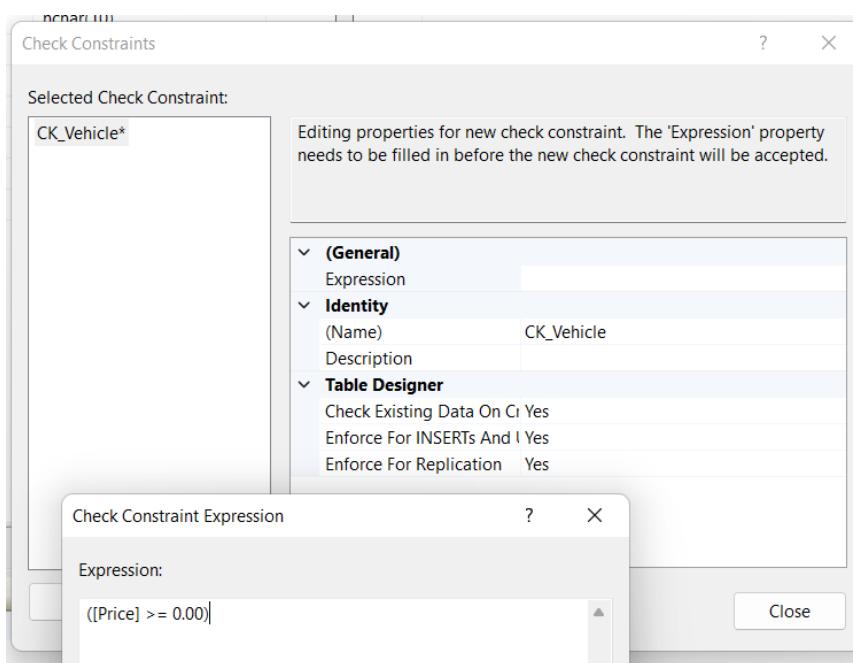


Click on the 3 dots



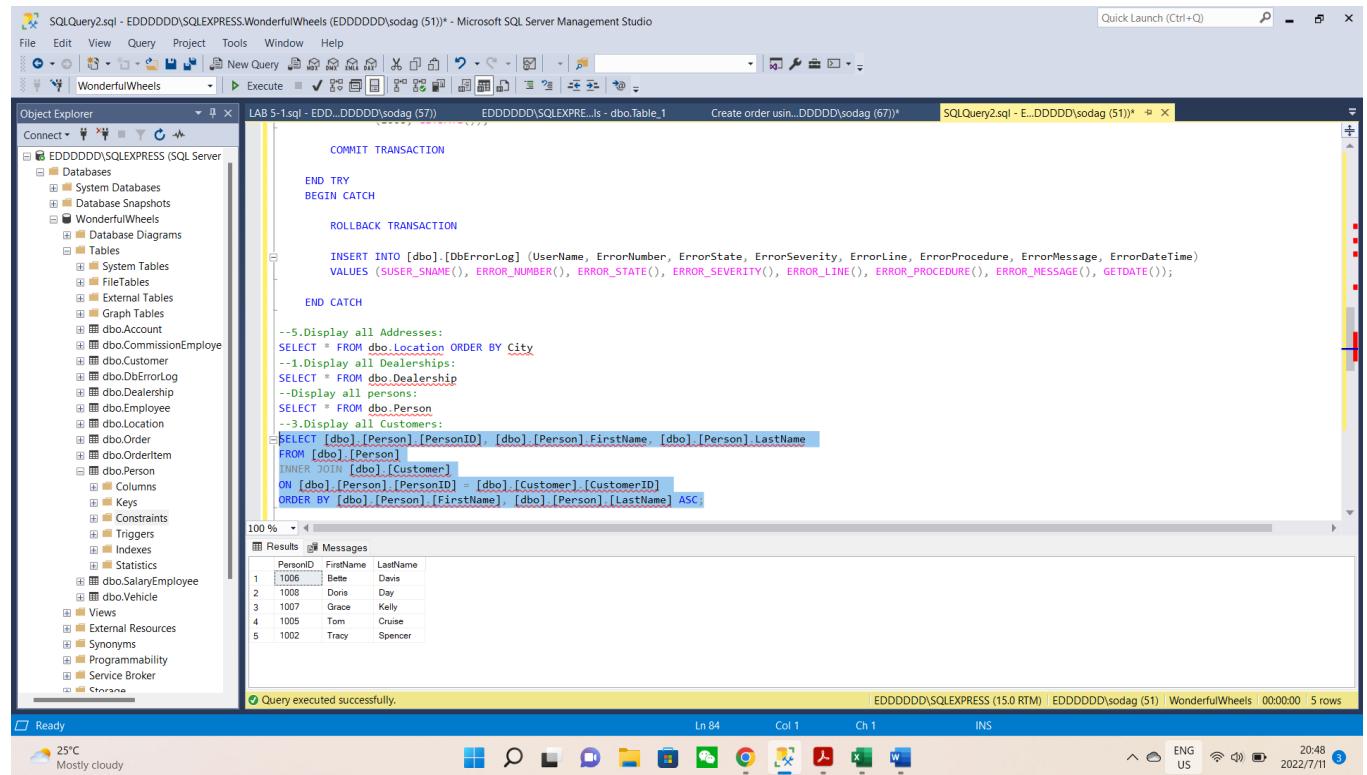
8. Adding Constraints:

Double click Table -> Constraints -> New -> Rename and add Expression.



Step3: Data import with validation rule testing

1) Display all Dealerships:



SQLQuery2.sql - EDDDDDD\SQLEXPRESS.WonderfulWheels (EDDDDDDD\sodag (51))* - Microsoft SQL Server Management Studio

```

COMMIT TRANSACTION
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION
    INSERT INTO [dbo].[DbErrorLog] (UserName, ErrorNumber, ErrorState, ErrorSeverity, ErrorLine, ErrorProcedure, ErrorMessage, ErrorDateTime)
    VALUES (SUSER_SNAME(), ERROR_NUMBER(), ERROR_STATE(), ERROR_SEVERITY(), ERROR_LINE(), ERROR_PROCEDURE(), ERROR_MESSAGE(), GETDATE());
END CATCH
--5.Display all Addresses:
SELECT * FROM dbo.Location ORDER BY City
--1.Display all Dealerships:
SELECT * FROM dbo.Dealership
--Display all persons:
SELECT * FROM dbo.Person
--3.Display all Customers:
SELECT [dbo].[Person].[PersonID], [dbo].[Person].[FirstName], [dbo].[Person].[LastName]
FROM [dbo].[Person]
INNER JOIN [dbo].[Customer]
ON [dbo].[Person].[PersonID] = [dbo].[Customer].[CustomerID]
ORDER BY [dbo].[Person].[FirstName], [dbo].[Person].[LastName] ASC

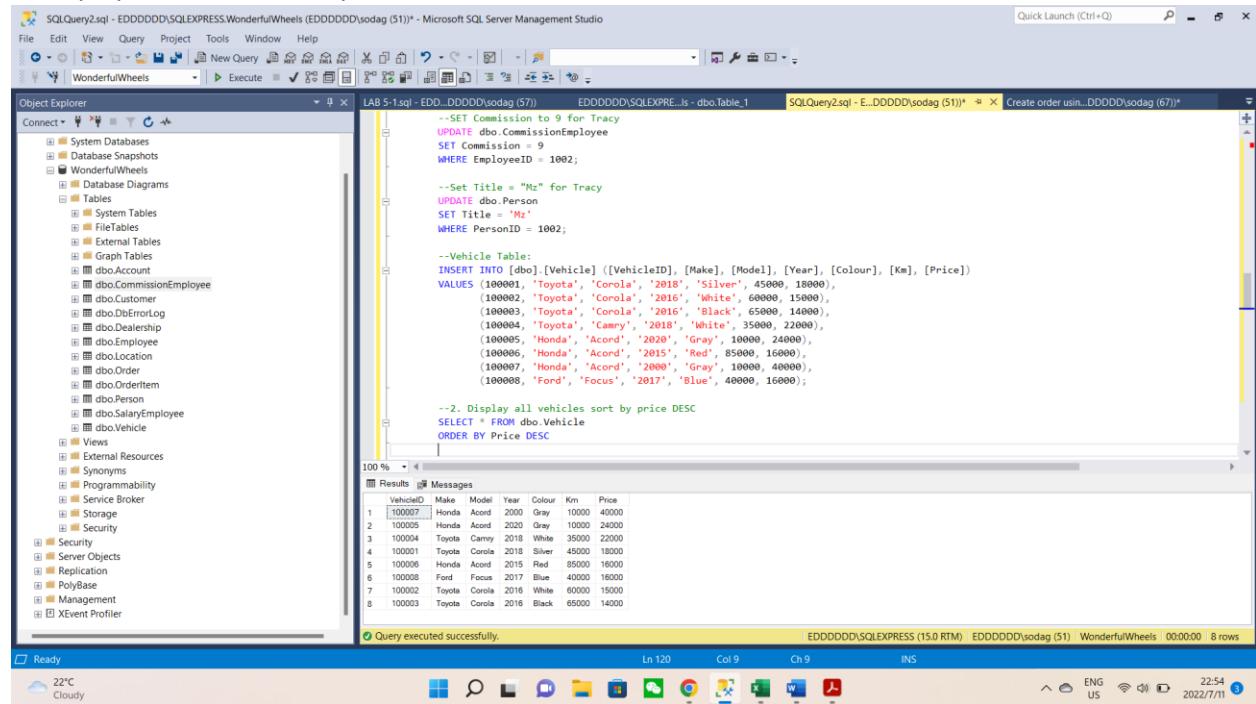
```

Results

PersonID	FirstName	LastName
1006	Belle	Davis
1008	Doris	Day
1007	Grace	Kelly
1005	Tom	Cruise
1002	Tracy	Spencer

Query executed successfully.

2) Display all vehicles sort by Price DESC:



SQLQuery2.sql - EDDDDDD\SQLEXPRESS.WonderfulWheels (EDDDDDDD\sodag (51))* - Microsoft SQL Server Management Studio

```

--SET Commission to 9 for Tracy
UPDATE dbo.CommissionEmployee
SET Commission = 9
WHERE EmployeeID = 1002;

--Set Title = "Ms" for Tracy
UPDATE dbo.Person
SET Title = 'Ms'
WHERE PersonID = 1002;

--Vehicle Table:
INSERT INTO [dbo].[Vehicle] ([VehicleID], [Make], [Model], [Year], [Colour], [Km], [Price])
VALUES (100001, 'Toyota', 'Corolla', '2018', 'Silver', 45000, 18000),
       (100002, 'Toyota', 'Corolla', '2016', 'White', 60000, 15000),
       (100003, 'Toyota', 'Corolla', '2016', 'Black', 65000, 14000),
       (100004, 'Toyota', 'Corolla', '2018', 'White', 35000, 22000),
       (100005, 'Honda', 'Accord', '2020', 'Gray', 10000, 24000),
       (100006, 'Honda', 'Accord', '2015', 'Red', 85000, 16000),
       (100007, 'Honda', 'Accord', '2009', 'Gray', 10000, 40000),
       (100008, 'Ford', 'Focus', '2017', 'Blue', 40000, 16000),
       (100002, 'Toyota', 'Corolla', '2016', 'White', 60000, 15000),
       (100003, 'Toyota', 'Corolla', '2016', 'Black', 65000, 14000);

--2. Display all vehicles sort by price DESC
SELECT * FROM dbo.Vehicle
ORDER BY Price DESC

```

Results

VehicleID	Make	Model	Year	Colour	Km	Price
100007	Honda	Accord	2009	Gray	10000	40000
100006	Honda	Accord	2015	Red	85000	16000
100004	Toyota	Corolla	2018	White	35000	22000
100001	Toyota	Corolla	2016	Silver	45000	18000
100005	Honda	Accord	2020	Gray	10000	24000
100008	Ford	Focus	2017	Blue	40000	16000
100002	Toyota	Corolla	2016	White	60000	15000
100003	Toyota	Corolla	2016	Black	65000	14000

Query executed successfully.

3) Display all customers order by first, last names asc:

```
--Customer Table:  
SET IDENTITY_INSERT dbo.Customer OFF;  
  
--Employee Table:  
SET IDENTITY_INSERT dbo.Employee ON;  
  
INSERT INTO [dbo].[Employee] ([EmployeeID], [EmpDealID], [HireDate], [Role], [ManagerID])  
VALUES  
    (1000, 1, GETDATE(), 'Manager', NULL),  
    (1001, 1, GETDATE(), 'OfficeAdmin', 1000),  
    (1002, 1, GETDATE(), 'Sales', 1000),  
    (1003, 1, GETDATE(), 'Sales', 1000),  
    (1004, 1, GETDATE(), 'Sales', 1000);
```

) %

Messages

Msg 515, Level 16, State 2, Line 78
Cannot insert the value NULL into column 'ManagerID', table 'WonderfulWheels.dbo.Employee'; column does not allow nulls. INSERT fails.
The statement has been terminated.

Completion time: 2022-07-11T21:12:09.5325710-04:00

4.) Display all employees order by first, last names asc:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists database objects like External Tables, Graph Tables, dbo.Account, dbo.CommissionEmployee, dbo.Customer, dbo.DealerShip, dbo.Employee, dbo.Location, dbo.Order, dbo.OrderItem, dbo.Person, and dbo.Vehicle. The main pane displays a query window with the following code and results:

```
--Customer Table:  
SET IDENTITY_INSERT dbo.Customer OFF;  
  
--Employee Table:  
SET IDENTITY_INSERT dbo.Employee ON;  
  
--Allow NULL to prevent error:  
ALTER TABLE dbo.Employee ALTER COLUMN ManagerID INT NULL;  
  
INSERT INTO [dbo].[Employee] ([EmployeeID], [EmpDealID], [HireDate], [Role], [ManagerID])  
VALUES  
    (1000, 2, GETDATE(), 'Manager', NULL),  
    (1001, 2, GETDATE(), 'OfficeAdmin', 1000),  
    (1002, 2, GETDATE(), 'Sales', 1000),  
    (1003, 2, GETDATE(), 'Sales', 1000),  
    (1004, 2, GETDATE(), 'Sales', 1000);  
  
--4.Display all Employees:  
SELECT [dbo].[Employee].[EmployeeID], [dbo].[Person].[FirstName], [dbo].[Person].[LastName]  
FROM [dbo].[Person]  
INNER JOIN [dbo].[Employee]  
ON [dbo].[Person].[PersonID] = [dbo].[Employee].[EmployeeID]  
ORDER BY [dbo].[Person].[FirstName], [dbo].[Person].[LastName] ASC;
```

The results grid shows the following data:

EmployeeID	FirstName	LastName
1003	James	Stewart
1000	John	Smith
1001	Mary	Brown
1004	Paul	Newman
1002	Tracy	Spencer

Query executed successfully.

5) Display Addresses sort by City:

SQLQuery2.sql - EDDDDDD\SQLEXPRESS.WonderfulWheels (EDDDDD(sodag (51)) - Microsoft SQL Server Management Studio

```

USE WonderfulWheels
GO

-- Location Table:
BEGIN TRY

    BEGIN TRANSACTION

    INSERT INTO [dbo].[Location] ([StreetAddress], [City], [Province], [PostalCode])
    VALUES ('221 King St W', 'Kitchener', 'Ontario', 'N2B3C6'),
           ('77 Victoria St N', 'Cambridge', 'Ontario', 'N1Z8B8'),
           ('100 White Oak Rd', 'London', 'Ontario', 'L9B1N2');

    COMMIT TRANSACTION

END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION
END CATCH

--Test:
SELECT * FROM dbo.Location ORDER BY City

```

Results

LocationID	StreetAddress	City	Province	PostalCode
1	77 Victoria St N	Cambridge	Ontario	N1Z8B8
2	221 King St W	Kitchener	Ontario	N2B3C6
3	100 White Oak Rd	London	Ontario	L9B1N2

Query executed successfully.

6. Update Tracy Spencer commission to 9: ERROR OCCURRED

SQLQuery2.sql - EDDDDDD\SQLEXPRESS.WonderfulWheels (EDDDDD(sodag (51)) - Microsoft SQL Server Management Studio

```

ALTER TABLE dbo.Employee ALTER COLUMN ManagerID INT NULL;

INSERT INTO [dbo].[Employee] ([EmployeeID], [EmpDealID], [HireDate], [Role], [ManagerID])
VALUES (1000, 2, GETDATE(), 'Manager', NULL),
       (1001, 2, GETDATE(), 'OfficeAdmin', 1000),
       (1002, 2, GETDATE(), 'Sales', 1000),
       (1003, 2, GETDATE(), 'Sales', 1000),
       (1004, 2, GETDATE(), 'Sales', 1000);

--CommissionEmployee Table:
SET IDENTITY_INSERT dbo.Employee OFF;
SET IDENTITY_INSERT dbo.CommissionEmployee ON;

INSERT INTO [dbo].[CommissionEmployee] ([EmployeeID], [Commission])
VALUES (1002, 13),
       (1003, 15),
       (1004, 10);

UPDATE dbo.CommissionEmployee
SET Commission = 9
WHERE EmployeeID = 1002;

COMMIT TRANSACTION

END TRY
BEGIN CATCH
END CATCH

```

Messages

Msg 847, Level 16, State 0, Line 96
The UPDATE statement conflicted with the CHECK constraint "CK_Comm". The conflict occurred in database "WonderfulWheels", table "dbo.CommissionEmployee", column 'Commission'.
The statement has been terminated.

Completion time: 2022-07-11T22:39:09.4092933-04:00

Query completed with errors.

7. Update Tracy's Title to 'Mz', error occurred:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'WonderfulWheels' is selected. In the center pane, a query window contains the following script:

```
--CommissionEmployee Table:  
SET IDENTITY_INSERT dbo.Employee OFF;  
SET IDENTITY_INSERT dbo.CommissionEmployee ON;  
  
INSERT INTO [dbo].[CommissionEmployee] ([EmployeeID], [Commission])  
VALUES (1002, 13),  
       (1003, 15),  
       (1004, 10);  
  
--SET Commission to 9 for Tracy  
UPDATE dbo.CommissionEmployee  
SET Commission = 9  
WHERE EmployeeID = 1002;  
  
--Set Title = "Mz" for Tracy  
UPDATE dbo.Person  
SET Title = 'Mz'  
WHERE PersonID = 1002;  
  
COMMIT TRANSACTION  
  
END TRY  
BEGIN CATCH  
END CATCH
```

In the Messages pane, an error message is displayed:

```
Msg 191, Level 16, State 0, Line 12  
The UPDATE statement conflicted with the CHECK constraint "CK_Title". The conflict occurred in database "WonderfulWheels", table "dbo.Person", column 'Title'.  
The statement has been terminated.
```

The status bar at the bottom indicates the completion time as 2022-07-11T22:36:21.7448963-04:00.

8. Display vehicles that are not sold:

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'WonderfulWheels' is selected. In the center pane, a query window contains the following script:

```
--Order Table:  
INSERT INTO [dbo].[Order] ([OrderID], [OrderCusID], [OrdEmpID], [OrderDate], [OrdDealID])  
VALUES (100, 1005, 1002, GETDATE(), 2),  
      (101, 1006, 1002, GETDATE(), 2),  
      (102, 1002, 1003, GETDATE(), 2);  
  
--OrderItem Table:  
SET IDENTITY_INSERT dbo.CommissionEmployee OFF;  
SET IDENTITY_INSERT dbo.OrderItem ON;  
  
INSERT INTO [dbo].[OrderItem] ([OrderID], [VehicleID], [FinalSalePrice])  
VALUES (100, 100001, 17500),  
      (100, 100004, 21000),  
      (101, 100008, 15000),  
      (102, 100006, 15000);  
  
--Display vehicles that are not sold.  
SELECT * FROM dbo.Vehicle V  
LEFT OUTER JOIN dbo.OrderItem O  
ON V.VehicleID = O.VehicleID  
WHERE O.VehicleID IS NULL
```

In the Results pane, the output of the final SELECT statement is shown:

VehicleID	Make	Model	Year	Colour	Km	Price	OrderID	VehicleID	FinalSalePrice
1	Toyota	Corolla	2016	White	60000	15000	NULL	NULL	NULL
2	Toyota	Corolla	2016	Black	65000	14000	NULL	NULL	NULL
3	Honda	Accord	2020	Grey	10000	24000	NULL	NULL	NULL
4	Honda	Accord	2020	Grey	10000	40000	NULL	NULL	NULL

The status bar at the bottom indicates the query was executed successfully at 2022-07-12 1:01.

Step4: Data Warehousing

1: Create Data Warehouse:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'WonderfulWheelsDW'. The main query window contains T-SQL code to create a data warehouse:

```
-- PROG8400 LAB7
-- GROUP:MIC
-- GROUP MEMBERS: KEYAO WANG, YUHANG LI, YAZHENG GUO.

-- Step1: CREATE DATA WAREHOUSE: WonderfulWheelsDW
USE master
GO

IF DB_ID(N'WonderfulWheelsDW') IS NOT NULL
    DROP DATABASE WonderfulWheelsDW;
GO
CREATE DATABASE WonderfulWheelsDW
GO
```

The results pane shows the execution status: "Query executed successfully." The message bar at the bottom indicates the session details: EDDDDDD/SQLEXPRESS (15.0 RTM) | EDDDDDD/sodag (74) | WonderfulWheels | 00:00:00 | 3 rows.

2: ETL into table Dim_CommissionEmployee

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'WonderfulWheelsDW'. The main query window contains T-SQL code for loading data into the 'Dim_CommissionEmployee' table:

```
USE WonderfulWheels
GO

-- Step2: Load CommissionEmployee into data warehouse
SELECT
    IDENTITY(INT,1,1) AS EmployeeSK,
    p.PersonID AS EmployeeAK,
    p.FirstName,
    p.LastName,
    p.Email,
    p.DateOfBirth,
    p.Title,
    e.HireDate,
    e.[Role] AS EmpRole,
    ce.Commission,
    e.ManagerID,
    NULL AS ExpirationDate,
    GETDATE() AS LoadDate
INTO [WonderfulWheelsDW].[dbo].[Dim_CommissionEmployee]
FROM [dbo].[Person] p
JOIN [dbo].[Employee] e ON p.PersonID = e.EmployeeID
JOIN [dbo].[CommissionEmployee] ce ON e.EmployeeID = ce.EmployeeID

--Test
SELECT * FROM [WonderfulWheelsDW].[dbo].[Dim_CommissionEmployee]
```

The results pane shows the data being inserted into the table. The message bar at the bottom indicates the session details: EDDDDDD/SQLEXPRESS (15.0 RTM) | EDDDDDD/sodag (74) | WonderfulWheels | 00:00:00 | 3 rows.

3: Load Customer into data warehouse.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'WonderfulWheelsDW'. The main pane contains a query window with the following T-SQL code:

```
-- Step3: Load Customer into data warehouse
SELECT
    IDENTITY(INT,1,1) AS CustomerSK,
    p.PersonID AS CustomerAK,
    p.FirstName,
    p.LastName,
    p.Email,
    p.DateOfBirth,
    p.Title,
    c.RegistrationDate AS RegDate,
    NULL AS ExpirationDate,
    GETDATE() AS LoadDate
INTO [WonderfulWheelsDW].[dbo].[Dim_Customer]
FROM [dbo].[Person] p
JOIN [dbo].[Customer] c ON p.PersonID = c.CustomerID

-- Test
SELECT * FROM [WonderfulWheelsDW].[dbo].[Dim_Customer]
```

The results pane shows a table with 5 rows of data, which is then confirmed by a message: "Query executed successfully."

	CustomerSK	CustomerAK	FirstName	LastName	Email	DatedBirth	Title	RegDate	ExpirationDate	LoadDate
1	1002	Tracy	Spencer	tespencer@email.com	1999-07-22	Ms	2022-07-11 20:36:56.693	NULL	2022-07-28 13:44:04.100	
2	1005	Tom	Cruise	touise@email.com	1982-03-22	Mr	2022-07-11 20:36:56.693	NULL	2022-07-28 13:44:04.100	
3	1006	Bette	Davis	bdavis@email.com	1952-09-01	Ms	2022-07-11 20:36:56.693	NULL	2022-07-28 13:44:04.100	
4	1007	Grace	Kelly	gkelly@email.com	1973-06-09	Ms	2022-07-11 20:36:56.693	NULL	2022-07-28 13:44:04.100	
5	1008	Doris	Day	dday@email.com	1962-03-22	Mr	2022-07-11 20:36:56.693	NULL	2022-07-28 13:44:04.100	

4: Load Vehicle into Data Warehouse:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'WonderfulWheelsDW'. The main pane contains a query window with the following T-SQL code:

```
-- Step4: Load Vehicle into Data Warehouse:
SELECT
    IDENTITY(INT,1,1) AS VehicleSK,
    VehicleID AS VehicleAK,
    Make,
    Model,
    [Year],
    Colour,
    Km,
    Price,
    NULL AS ExpirationDate,
    GETDATE() AS LoadDate
INTO [WonderfulWheelsDW].[dbo].[Dim_Vehicle]
FROM [dbo].[Vehicle]

-- Test
SELECT * FROM [WonderfulWheelsDW].[dbo].[Dim_Vehicle]
```

The results pane shows a table with 8 rows of data, which is then confirmed by a message: "Query executed successfully."

	VehicleSK	VehicleAK	Make	Model	Year	Colour	Km	Price	ExpirationDate	LoadDate
1	100001	Toyota	Corolla	2018	Silver	45000	18000	NULL	2022-07-28 13:52:48.413	
2	100002	Toyota	Corolla	2016	White	60000	15000	NULL	2022-07-28 13:52:48.413	
3	100003	Toyota	Corolla	2016	Black	65000	14000	NULL	2022-07-28 13:52:48.413	
4	100004	Toyota	Camry	2018	White	35000	22000	NULL	2022-07-28 13:52:48.413	
5	100005	Honda	Accord	2020	Gray	10000	24000	NULL	2022-07-28 13:52:48.413	
6	100006	Honda	Accord	2015	Red	85000	16000	NULL	2022-07-28 13:52:48.413	
7	100007	Honda	Accord	2000	Gray	10000	40000	NULL	2022-07-28 13:52:48.413	
8	100008	Ford	Focus	2017	Blue	40000	16000	NULL	2022-07-28 13:52:48.413	

5: Load Dealership into Data warehouse:

```

SQLQuery1.sql - EDDDDDD\SQLEXPRESS.WonderfulWheels (EDDDDDDD\sodag (74)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer
Connect Databases System Databases Database Snapshots WonderfulWheels Database Diagrams Tables System Tables FileTables External Tables Graph Tables dbo.Dim.CommissionEmployee dbo.Dim.Customer dbo.Dim.Dealership Columns Keys Constraints Triggers Indexes Statistics Views External Resources Synonyms Programmability Concurrency
SQLQuery1.sql - E_DDDDD(sodag (74))*
GO
-- Step5: Load Dealership into Data warehouse:
SELECT
    IDENTITY(INT,1,1) AS DealershipSK,
    d.DealershipID AS DealershipAK,
    l.StreetAddress,
    l.City,
    l.Province,
    l.PostalCode,
    NULL AS ExpirationDate,
    GETDATE() AS LoadDate
INTO [WonderfulWheelsDW].[dbo].[Dim_Dealership]
FROM [dbo].[Location] l
JOIN [dbo].[Dealership] d ON l.LocationID = d.LocationId
-- Test
SELECT * FROM [WonderfulWheelsDW].[dbo].[Dim_Dealership]
GO

```

Results grid (Messages)

	DealershipSK	DealershipAK	StreetAddress	City	Province	PostalCode	ExpirationDate	LoadDate
1	221 King St W	Kitchener	Ontario	N9B 3C6	NULL	2022-07-29 14:07:35.177		
2	77 Victoria St N	Cambridge	Ontario	N1Z 8B8	NULL	2022-07-28 14:07:35.177		
3	100 White Oak Rd	London	Ontario	N9B 1W2	NULL	2022-07-28 14:07:35.177		

Query executed successfully.

6: Create Fact Table: Fact_Sales.

```

SQLQuery1.sql - EDDDDDD\SQLEXPRESS.WonderfulWheels (EDDDDDDD\sodag (74)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer
Connect Databases System Databases Database Snapshots WonderfulWheels Database Diagrams Tables System Tables FileTables External Tables Graph Tables dbo.Dim.CommissionEmployee dbo.Dim.Customer dbo.Dim.Dealership dbo.Dim.Vehicle dbo.Fact_Sales Columns Keys Constraints Triggers Indexes Statistics Views External Resources Summaries
SQLQuery1.sql - E_DDDDD(sodag (74))*
GO
-- Step6: Create Fact Table: Fact_Sales.
SELECT
    o.OrderID,
    IDENTITY(INT,1,1) AS OrderItemID,
    EmployeeSK,
    CustomerSK,
    VehicleSK,
    DealershipSK,
    oi.FinalSalePrice,
    o.OrderDate AS OrderDateSK,
    oi.FinalSalePrice * ce.Commission AS Commission
INTO [WonderfulWheelsDW].[dbo].[Fact_Sales]
FROM [dbo].[Order] o
LEFT JOIN [dbo].[Employee] e ON o.OrgEmpID = e.EmployeeID
LEFT JOIN [dbo].[Dealership] d ON o.OrgDealID = d.DealershipID
LEFT JOIN [dbo].[Customer] c ON o.OrderCusID = c.CustomerID
LEFT JOIN [dbo].[OrderItem] oi ON o.OrderID = oi.OrderID
LEFT JOIN [dbo].[Vehicle] v ON oi.VehicleID = v.VehicleID
-- Join to Dimension to get SKs:
JOIN [WonderfulWheelsDW].[dbo].[Dim_CommissionEmployee] ce ON o.OrgEmpID = EmployeeAK
JOIN [WonderfulWheelsDW].[dbo].[Dim_Customer] ON o.OrderCusID = CustomerAK
JOIN [WonderfulWheelsDW].[dbo].[Dim_Vehicle] ON oi.VehicleID = VehicleAK
JOIN [WonderfulWheelsDW].[dbo].[Dim_Dealership] ON o.OrgDealID = DealershipAK
-- Test
SELECT * FROM [WonderfulWheelsDW].[dbo].[Fact_Sales]
GO

```

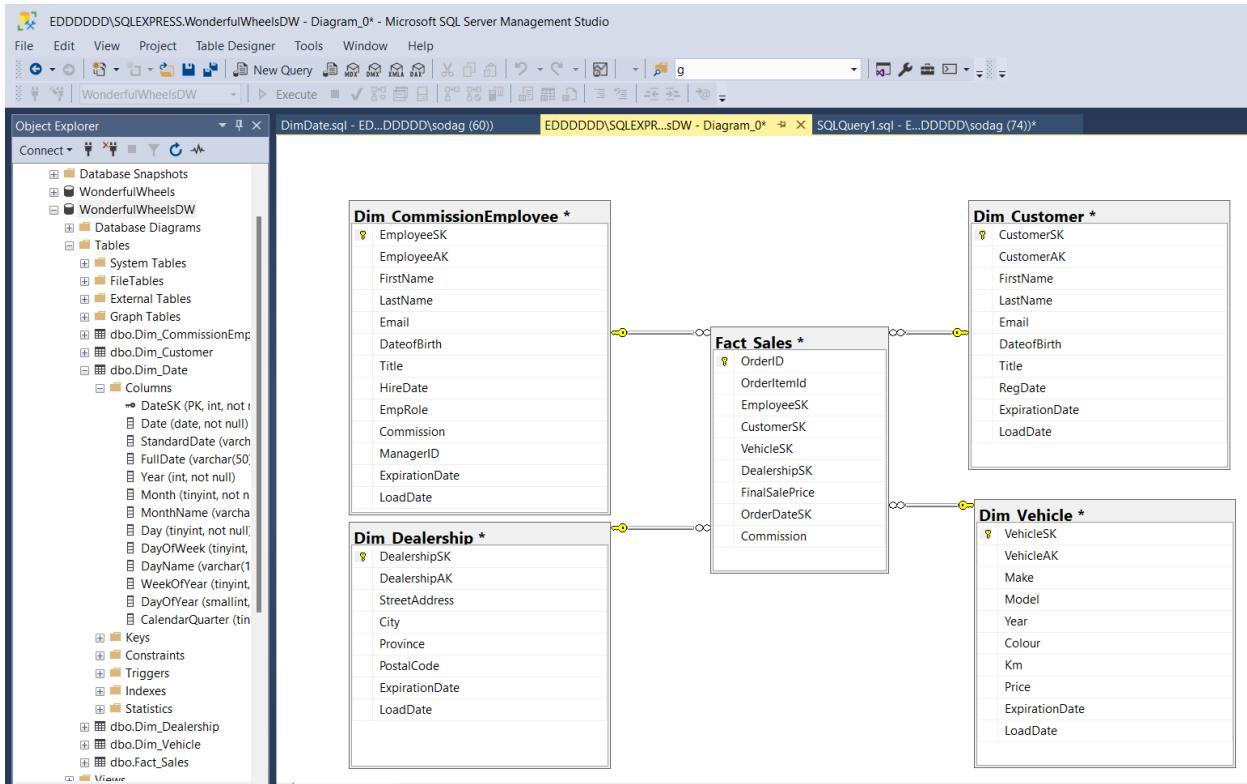
Results grid (Messages)

	OrderID	OrderItemID	EmployeeSK	CustomerSK	VehicleSK	DealershipSK	FinalSalePrice	OrderDateSK	Commission	
1	102	1	1	6	1	15000	2022-07-11	225000		
2	100	2	1	2	1	17500	2022-07-11	227500		
3	100	3	1	2	4	1	21000	2022-07-11	273000	
4	101	4	1	3	8	1	15000	2022-07-11	195000	

Query executed successfully.

7: Include Dim_Date table:

8: Star Schema



Step5: Data Warehouse & BI

IN SQL:

-Create view with Employee, Customer, and Vehicle Sales information:

```

CREATE VIEW dbo.vwEmployeeSales
AS
SELECT
    CE.FirstName + ' ' + CE.LastName AS 'Employee Name',
    CE.Email,
    DATEDIFF(YEAR, CE.DateOfBirth, GETDATE()) AS 'Age',
    DATEDIFF(YEAR, CE.HireDate, GETDATE()) AS 'Years of service',
    CE.Commission,
    FinalSalePrice = CE.Commission/100 AS 'Sale Commission',
    C.FirstName + ' ' + C.LastName AS 'Customer Name',
    C.RegDate,
    C.Title,
    D.DealershipSK AS 'Dealership',
    D.StreetAddress AS 'Dealership Street Address',
    D.City AS 'Dealership City',
    D.Province AS 'Dealership Province',
    V.Make,
    V.Model,
    V.[Year],
    V.Km,
    V.Price,
    F.FinalSalePrice,
    YEAR(F.OrderDateSK) AS 'Sale Year',
    MONTH(F.OrderDateSK) AS 'Sale Month'

FROM Dim_CollectionEmployee CE
LEFT JOIN [dbo].[Fact_Sales] F ON CE.EmployeeSK = F.EmployeeSK
LEFT JOIN [dbo].[Dim_Customer] C ON C.CustomerSK = F.CustomerSK
LEFT JOIN [dbo].[Dim_Dealership] D ON D.DealershipSK = F.DealershipSK
LEFT JOIN [dbo].[Dim_Vehicle] V ON V.VehicleSK = F.VehicleSK

```

-Check View Table:

```

SELECT TOP (1000) [Employee Name]
      ,[Email]
      ,[Age]
      ,[Years of service]
      ,[Commission]
      ,[Sale Commission]
      ,[Customer Name]
      ,[RegDate]
      ,[Title]
      ,[Dealership]
      ,[Dealership Street Address]
      ,[Dealership City]
      ,[Dealership Province]
      ,[Make]
      ,[Model]
      ,[Year]
      ,[Km]
      ,[Price]
      ,[FinalSalePrice]
      ,[OrderDateSK]
  FROM [WonderfulWheelsDW].[dbo].[vwEmployeeSales]

```

	Employee Name	Email	Age	Years of service	Commission	Sale Commission	Customer Name	RegDate	Title	Dealership	Dealership Street Address	Dealership City	Dealership Province	Make	Model	Year	Km
1	Tracy Spencer	tspencer@email.com	24	17	13	2275	Tom Cruise	2013-05-05 00:00:00.000	Mr.	1	221 King St W	Kitchener	Ontario	Toyota	Corolla	2013	45k
2	Tracy Spencer	tspencer@email.com	24	17	13	2700	Tom Cruise	2012-05-05 00:00:00.000	Mr.	1	221 King St W	Kitchener	Ontario	Toyota	Camry	2012	35k
3	Tracy Spencer	tspencer@email.com	24	17	13	1950	Bette Davis	2013-05-05 00:00:00.000	Ms.	1	221 King St W	Kitchener	Ontario	Ford	Focus	2017	40k
4	James Stewart	jstewart@email.com	26	10	15	2250	Tracy Spencer	2010-05-05 00:00:00.000	Ms.	1	221 King St W	Kitchener	Ontario	Honda	Accord	2015	65k
5	Paul Newman	pnewman@email.com	30	2	10	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

-Test View:

SQL Server Management Studio - EDDDDDD\SQLEXPRESS.WonderfulWheelsDW (EDDDDD) [sodag (71)*]

```

-- Step 2 Test it
DECLARE @FirstName VARCHAR(100),
        @LastName VARCHAR(100),
        @Year CHAR(4),
        @Month CHAR(4)

SET @FirstName = ''
SET @LastName = ''
SET @Year = '2022'
SET @Month = '7'

SELECT *
FROM dbo.vwEmployeeSales
WHERE ([Employee Name] = @FirstName + ' ' + @LastName OR @FirstName = '')
    AND ([Sale Year] = @Year OR @Year IS NULL)
    AND ([Sale Month] = @Month OR @Month IS NULL)

```

Results

Employee Name	Email	Age	Years of service	Commission	Sale Commission	Customer Name	RegDate	Title	Dealership	Dealership Street Address	Dealership City	Dealership Province	Make	Model	Year	Km
Tracy Spencer	tspencer@email.com	24	17	13	2275	Tom Cruise	2012-05-05 00:00:00.000	Mr	1	221 King St W	Kitchener	Ontario	Toyota	Corolla	2018	45000

Query executed successfully.

-Create Main Procedure:

And Run Test1: `EXEC dbo.spGetEmployeeSales ''','NULL,NULL`
Null value which returns all rows.

SQL Server Management Studio - EDDDDDD\SQLEXPRESS.WonderfulWheelsDW (EDDDDD) [sodag (71)*]

```

USE [WonderfulWheelsDW]
GO
/*
Test1: NULL INPUT
    EXEC dbo.spGetEmployeeSales ''','NULL,NULL
*/
CREATE PROCEDURE dbo.spGetEmployeeSales
(
    @FirstName VARCHAR(100),
    @LastName VARCHAR(100),
    @Year CHAR(4),
    @Month CHAR(4)
)
AS
BEGIN
    SELECT *
    FROM dbo.vwEmployeeSales
    WHERE ([Employee Name] LIKE '%' + @FirstName + '%' + @LastName + '%' OR (@FirstName = '' AND @LastName = ''))
        AND ([Sale Year] = @Year OR @Year IS NULL)
        AND ([Sale Month] = @Month OR @Month IS NULL)
    ORDER BY [Employee Name]
END
GO

```

Results

Employee Name	Email	Age	Years of service	Commission	Sale Commission	Customer Name	RegDate	Title	Dealership	Dealership Street Address	Dealership City	Dealership Province	Make	Model	Year	Km
James Stewart	jstewart@email.com	26	10	15	2250	Tracy Spencer	2010-05-05 00:00:00.000	Ms	1	221 King St W	Kitchener	Ontario	Honda	Accord	2019	85000
Paul Newman	pnewman@email...	30	2	10	NULL	NULL	NULL	N.	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Tracy Spencer	tspencer@email...	24	17	13	2275	Tom Cruise	2012-05-05 00:00:00.000	Mr	1	221 King St W	Kitchener	Ontario	Toyota	Corolla	2018	45000
Tracy Spencer	tspencer@email...	24	17	13	2730	Tom Cruise	2012-05-05 00:00:00.000	Mr	1	221 King St W	Kitchener	Ontario	Toyota	Camry	2018	35000
Tracy Spencer	tspencer@email...	24	17	13	1950	Bette Davis	2013-05-05 00:00:00.000	Ms	1	221 King St W	Kitchener	Ontario	Ford	Focus	2017	40000

Query executed successfully.

-Run Procedure Test2: `EXEC dbo.spGetEmployeeSales 'Ja','','NULL,NULL`
Which only returns James record.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists databases, tables, and other objects under the 'EDDDDDD\SQLEXPRESS' connection. The main window contains four tabs: 'SQLQuery7.sql - E_DDDDDD\sqlag (65)', 'Sales Report View... DDDDD\sqlag (68)', 'SQLQuery6.sql - E_DDDDDD\sqlag (55)', and 'SQLQuery1.sql - E_DDDDDD\sqlag (71)*'. The 'SQLQuery1.sql' tab is active, displaying the following T-SQL code:

```
USE [WonderfulWheelsDW]
/*
Test1: NULL INPUT
EXEC dbo.spGetEmployeeSales '','',NULL,NULL
Test2: NULL INPUT
EXEC dbo.spGetEmployeeSales 'Ja','','NULL,NULL
*/
CREATE PROCEDURE dbo.spGetEmployeeSales
(
    @FirstName VARCHAR(100),
    @LastName VARCHAR(100),
    @Year CHAR(4),
    @Month CHAR(4)
)
AS
BEGIN
    SELECT *
    FROM dbo.vwEmployeeSales
    WHERE ([Employee Name] LIKE '%' + @FirstName + '%' + @LastName + '%' OR (@FirstName = '' AND @LastName = ''))
        AND ([Sale Year] = @Year OR @Year IS NULL)
        AND ([Sale Month] = @Month OR @Month IS NULL)
    ORDER BY [Employee Name]
END
GO
```

The results pane shows a single row of data for James Stewart:

Employee Name	Email	Age	Years of service	Commission	Sale Commission	Customer Name	RegDate	Title	Dealership	Dealership Street Address	Dealership City	Dealership Province	Make	Model	Year	Km
James Stewart	jstewart@email.com	26	10	15	2250	Tracy Spencer	2010-05-05 00:00:00.000	Ms	1	221 King St W	Kitchener	Ontario	Honda	Accord	2015	85000

Below the results, a message indicates: 'Query executed successfully.'