# Project Overview

## Data Sources:

- The below site gives the different prices of the services given to the cars.
  https://engineerine.com/full-car-service-cost/

- The below site gives car prices and details about Nissan.
  https://www.autoauctionmall.com/learning-center/the-10-cheapest-new-cars-sold-in-the-us-in-2018

- The site below gives  Audi A3 car prices and more details about it.
  https://www.caranddriver.com/audi/a3-2022

- The site below gives  Mercedes car prices and more details about it.
  https://www.kbb.com/mercedes-benz/a-class/2021/

## References:

- https://media.pearsoncmg.com/ph/bp/bp_hoffer_mdm_13/index.html
- https://www.linkedin.com/learning/search?keywords=erd
- https://www.w3schools.com/sql/sql_groupby.asp
- https://www.sqlshack.com/learn-to-write-basic-sql-queries/
- Modern Database Management (13th Edition) - Authors: Jeffrey A. Hoffer, V. Ramesh, Heikki Topi
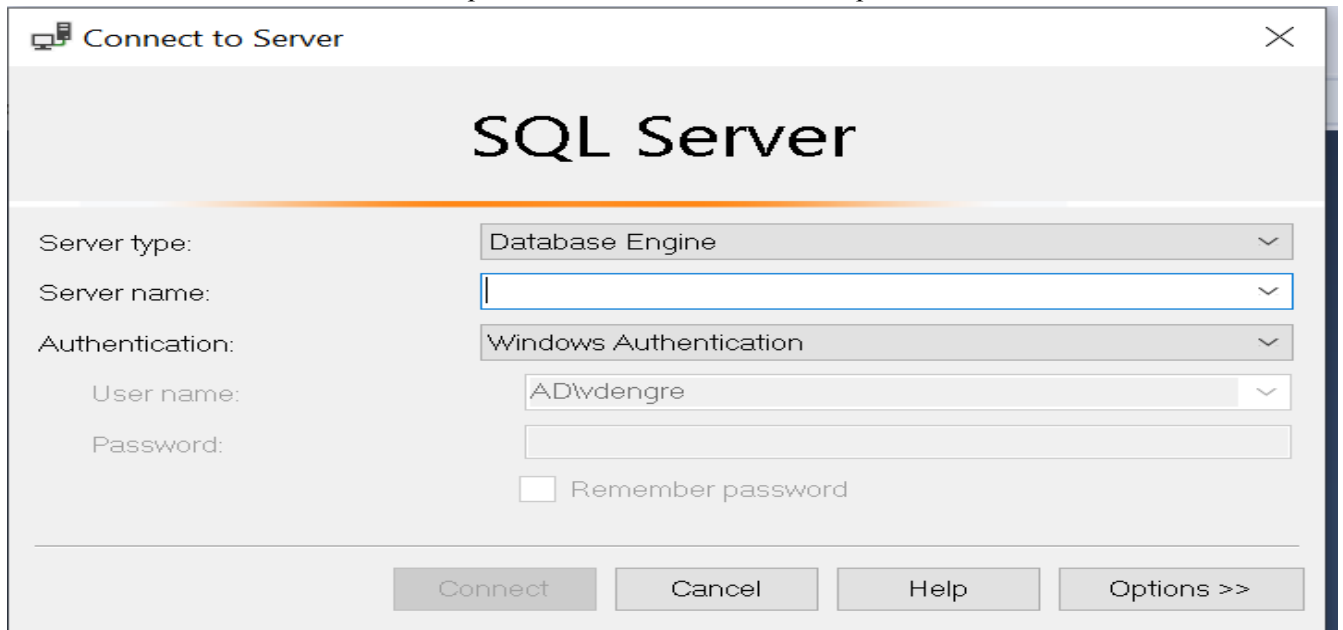
## Project Test:

### *Step 1:*

We will first be opening SQL in the VMware Horizon Client. Then we will be opening SQL Server Management Studio.
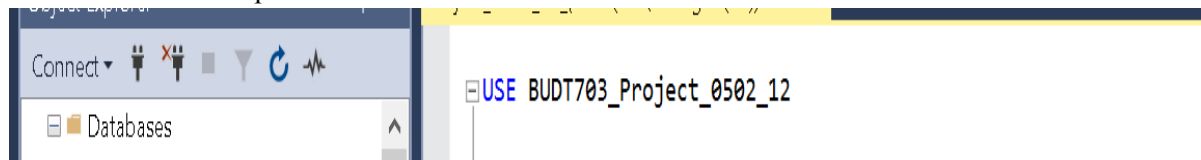


### *Step 2:*

To access the database we will need to put in the UMD server details as per the below screenshot.
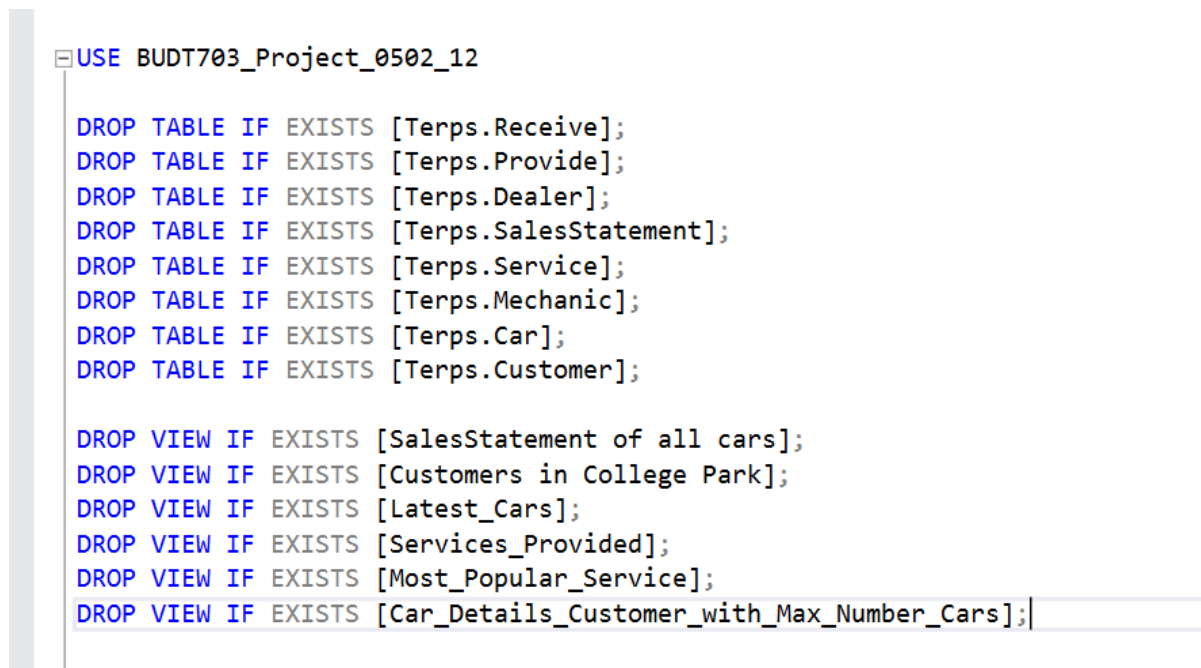
## Step 3:

Now we create a script where we use the database.



## Step 4:

We drop the tables if they exist, which needs to be created as per our ER Schema using drop table functionality.

```sql
USE BUDT703_Project_0502_12

DROP TABLE IF EXISTS [Terps.Receive];
DROP TABLE IF EXISTS [Terps.Provide];
DROP TABLE IF EXISTS [Terps.Dealer];
DROP TABLE IF EXISTS [Terps.SalesStatement];
DROP TABLE IF EXISTS [Terps.Service];
DROP TABLE IF EXISTS [Terps.Mechanic];
DROP TABLE IF EXISTS [Terps.Car];
DROP TABLE IF EXISTS [Terps.Customer];

DROP VIEW IF EXISTS [SalesStatement of all cars];
DROP VIEW IF EXISTS [Customers in College Park];
DROP VIEW IF EXISTS [Latest_Cars];
DROP VIEW IF EXISTS [Services_Provided];
DROP VIEW IF EXISTS [Most_Popular_Service];
DROP VIEW IF EXISTS [Car_Details_Customer_with_Max_Number_Cars];
```

## Step 5:
We create the tables as per our entities. We then add attributes to them and specify their data type and size. Next we even add any types of constraints if necessary and specify the primary key and also about adding or linking with a foreign key.

```sql
CREATE TABLE [Terps.Car] (
    carId CHAR (10) NOT NULL,
    carSerial VARCHAR (9),
    carModel VARCHAR (25),
    carColour VARCHAR (25),
    carManufactureYear CHAR (4),
    custId CHAR (10),
    carPrice DECIMAL (10,2),
        CONSTRAINT pk_Car_carId PRIMARY KEY (carId),
        CONSTRAINT fk_Car_custId FOREIGN KEY (custId)
            REFERENCES [Terps.Customer] (custId)
                ON DELETE SET NULL ON UPDATE CASCADE)
```

## Step 6:
Next step includes inserting values in each of the tables using insert keyword.

```sql
INSERT INTO [Terps.Customer] VALUES ('119090', 'John', 'Fernandes', '2407859691', '5660 Apt 12 Hyattsville MD 20783')
INSERT INTO [Terps.Customer] VALUES ('119089', 'Ramya', 'Thakkar', '2407859691', '48 Ave Apt 602 Hyattsville MD 20783')
INSERT INTO [Terps.Customer] VALUES ('119081', 'Karan', 'Gupta', '2409859691', '5990 pt 602 College Park MD 20740')
INSERT INTO [Terps.Customer] VALUES ('119099', 'Kartik', 'Shah', '2407859691', '7890 Ave Apt 602 College Park MD 20740')
INSERT INTO [Terps.Customer] VALUES ('119087', 'Ramani', 'Verma', '2407859341', '488 Ave Apt 304 College Park MD 20740')
INSERT INTO [Terps.Customer] VALUES ('119070', 'Tareek', 'Joshi', '2234783341', '890 Ave Apt 911 Green Belt MD 20737')
INSERT INTO [Terps.Customer] VALUES ('119082', 'Ramya', 'Shah', '2234859341', '908 Ave Apt 441 College Park MD 20740')
INSERT INTO [Terps.Customer] VALUES ('111178', 'Tanya', 'Dengre', '2234890341', '984 Ave Apt 811 Baltimore MD 20640')
INSERT INTO [Terps.Customer] VALUES ('112278', 'Sanya', 'Singh', '2299990341', '8489 Ave Apt 821 College Park MD 20740')
INSERT INTO [Terps.Customer] VALUES ('118090', 'Ron', 'Fernandes', '2489899691', '5645 Apt 19 Hyattsville MD 20783')
INSERT INTO [Terps.Customer] VALUES ('118190', 'Harry', 'Shah', '2489899999', '8125 Apt 20 College Park MD 20740')
INSERT INTO [Terps.Customer] VALUES ('118290', 'Ronak', 'Gupta', '2489669682', '8976 Apt 19 Green Belt MD 20737')

INSERT INTO [Terps.Car] VALUES ('129068', 'SD 1098', 'Nissan Versa Note S', 'Maroon', '2018', '119090', 16485.00)
INSERT INTO [Terps.Car] VALUES ('129067', 'MD 2267', 'Audi A3', 'Blue', '2020', '119089', 32341.00)
INSERT INTO [Terps.Car] VALUES ('129070', 'GD 2287', 'Audi A3', 'Black', '2020', '119089', 32341.00)
INSERT INTO [Terps.Car] VALUES ('128067', 'MD 3267', 'Nissan Versa Note S', 'Blue', '2018', '119089', 16485.00)
INSERT INTO [Terps.Car] VALUES ('129058', 'DD 3345', 'Audi A3', 'Black', '2020', '119089', 35895.00)
INSERT INTO [Terps.Car] VALUES ('129064', 'MD 3395', 'Audi Q8', 'Black', '2022', '119081', 38995.00)
INSERT INTO [Terps.Car] VALUES ('129034', 'MD 3305', 'Audi Q8', 'Black', '2022', '119087', 38995.00)
INSERT INTO [Terps.Car] VALUES ('122224', 'MD 3375', 'Mercedes','Black', '2021', '111178', 34488.00)
INSERT INTO [Terps.Car] VALUES ('123324', 'MD 3355', 'Mercedes','Red', '2020', '112278', 32467.00)
INSERT INTO [Terps.Car] VALUES ('129090', 'VA 1099', 'Ferrari', 'Red', '2021', '118090', 40578.00)
INSERT INTO [Terps.Car] VALUES ('128088', 'NY 1000', 'Land Rover','White', '2019', '118190', 37896.00)
INSERT INTO [Terps.Car] VALUES ('127070', 'MD 9999','Land Rover', 'Black', '2022', '118290', 40987.00)
```

## Step 7:

Now our tables have been created with values so we can query the table as per the requirement. We have used below questions to query them and the result will be displayed in the Result window.

## Query 1

```
--Question 1: Fetch the SalesStatement  for all the cars bought by Ramya Shah

CREATE VIEW [SalesStatement of all cars] AS
SELECT S.salStatementId AS 'Sales Statement Id', S.salStatementIDate AS 'Sales Statement Date'
FROM (([Terps.Receive] r
INNER JOIN [Terps.SalesStatement] S ON R.salStatementId = S.salStatementId)
INNER JOIN [Terps.Customer] C ON R.custId = C.custId)
WHERE C.custFirstName = 'Ramya'

SELECT * FROM [SalesStatement of all cars];

--Question 2: How many customers does Terp Motor Dealer have in 'College Park' area?
```

100 %

Results    Messages

| | Sales Statement Id | Sales Statement Date |
|---|---|---|
| 1 | 178879 | 2019-10-02 |
| 2 | 178985 | 2022-09-13 |
| 3 | 178989 | 2020-09-22 |
| 4 | 179989 | 2021-08-26 |

Query executed successfully.    doitsqlx.rhsmith.umd.edu,97...   AD\mthakkar (71)   BUDT703_Project_0502_12   00:00:00   4 rows

## Query 2

```
--Question 2: How many customers does Terp Motor Dealer have in 'College Park' area?
CREATE VIEW [Customers in College Park] AS
SELECT COUNT(*) AS 'Customers in College Park'
FROM [Terps.Customer] c
WHERE c.custAddress LIKE '%College%'

SELECT * FROM [Customers in College Park];

--Question 3: How many latest cars are there in showroom? (Manufactured in 2022)
```

100 %

Results    Messages

| | Customers in College Park |
|---|---|
| 1 | 6 |

Query executed successfully.    doitsqlx.rhsmith.umd.edu,97...   AD\mthakkar (71)   BUDT703_Project_0502_12   00:00:00   1 rows

## Query 3

```
--Question 3: How many latest cars are there in showroom? (Manufactured in 2022)

CREATE VIEW [Latest_Cars] AS
SELECT COUNT(c.carId) AS 'Number of Latest Cars'
FROM [Terps.Car] c
WHERE c.carManufactureYear = '2022' |

SELECT * FROM [Latest_Cars];

--Question 4: What are all the services provided by the mechanics?
```

100 %  ◄

⊞ Results  📜 Messages

| | Number of Latest Cars |
|---|---|
| 1 | 3 |

✅ Query executed successfully.     doitsqlx.rhsmith.umd.edu,97...  AD\mthakkar (71)  BUDT703_Project_0502_12  00:00:00  1 rows

**Query 4**

```
--Question 4: What are all the services provided by the mechanics?

CREATE VIEW [Services_Provided] AS
SELECT DISTINCT s.serId AS 'Service Id', s.serDescription AS 'Service Description', m.mecFirstName AS 'Mechanic First Name'
FROM (([Terps.Provide] p
INNER JOIN [Terps.Service] s ON p.serId = s.serId)
INNER JOIN [Terps.Mechanic] m ON p.mecId = m.mecId)

SELECT * FROM [Services_Provided]

--Question 5: What services cost more than a hundred dollars?
```

100 %  ◄

⊞ Results  📜 Messages

| | Service Id | Service Description | Mechanic First Name |
|---|---|---|---|
| 1 | 142245 | Tire Replacement | Raman |
| 2 | 142342 | Brake pad replacement | Vineet |
| 3 | 142346 | Car Wash | Peter |
| 4 | 142996 | Wiper replacement | Sherry |
| 5 | 144446 | Oil Change | Rohan |
| 6 | 147845 | Spark Plug Replacement | Parti |
| 7 | 147869 | Battery replacement | Shaan |

✅ Query executed successfully.     doitsqlx.rhsmith.umd.edu,97...  AD\mthakkar (71)  BUDT703_Project_0502_12  00:00:00  7 rows

**Query 5**

```sql
--Question 5: What services cost more than a hundred dollars?

CREATE VIEW [Costly_Services] AS
SELECT s.serId AS 'Service ID', s.serDescription AS 'Service Name', s.serPrice AS 'Service Price'
FROM [Terps.Service] s
WHERE s.serPrice > 100

SELECT * FROM Costly_Services

-- Question 6: What services have been provided on which cars by Terp Motor Dealers?

CREATE VIEW [Car_Services] AS
SELECT s.serId AS 'Service ID', s.serDescription AS 'Service Name', p.carId AS 'Car ID', c.carModel AS 'Car Model', c.carColour AS 'Car Color'
```

100 %

Results | Messages

| | Service ID | Service Name | Service Price |
|---|---|---|---|
| 1 | 142245 | Tire Replacement | 300.00 |
| 2 | 142342 | Brake pad replacement | 300.00 |
| 3 | 142346 | Car Wash | 200.00 |
| 4 | 147845 | Spark Plug Replacement | 400.00 |
| 5 | 147869 | Battery replacement | 200.00 |

Query executed successfully.    doitsqlx.rhsmith.umd.edu,97...  AD\mthakkar (71)  BUDT703_Project_0502_12  00:00:00  5 rows

**Query 6**

```sql
-- Question 6: What services have been provided on which cars by Terp Motor Dealers?

CREATE VIEW [Car_Services] AS
SELECT s.serId AS 'Service ID', s.serDescription AS 'Service Name', p.carId AS 'Car ID', c.carModel AS 'Car Model', c.carColour AS 'Car Color'
FROM [Terps.Service] s, [Terps.Provide] p, [Terps.Car] c
WHERE s.serId = p.serId AND p.carId = c.carId
GROUP BY s.serId, s.serDescription, p.carId, c.carModel, c.carColour

SELECT * FROM Car_Services

-- Question 7: Which customers have purchased more than 1 car from Terp Motor Dealers?

CREATE VIEW [Multiple_Cars] AS
```

100 %

Results | Messages

| | Service ID | Service Name | Car ID | Car Model | Car Color |
|---|---|---|---|---|---|
| 1 | 144446 | Oil Change | 122224 | Mercedes | Black |
| 2 | 142342 | Brake pad replacement | 123324 | Mercedes | Red |
| 3 | 142346 | Car Wash | 128067 | Nissan Versa Note S | Blue |
| 4 | 142996 | Wiper replacement | 128088 | Land Rover | White |
| 5 | 142245 | Tire Replacement | 129034 | Audi Q8 | Black |
| 6 | 142346 | Car Wash | 129058 | Audi A3 | Black |
| 7 | 147845 | Spark Plug Replacement | 129064 | Audi Q8 | Black |
| 8 | 142346 | Car Wash | 129067 | Audi A3 | Blue |
| 9 | 142346 | Car Wash | 129068 | Nissan Versa Note S | Maroon |
| 10 | 142346 | Car Wash | 129070 | Audi A3 | Black |
| 11 | 147869 | Battery replacement | 129090 | Ferrari | Red |

Query executed successfully.    doitsqlx.rhsmith.umd.edu,97...  AD\mthakkar (71)  BUDT703_Project_0502_12  00:00:00  11 rows

**Query 7**

```sql
-- Question 7: Which customers have purchased more than 1 car from Terp Motor Dealers?

CREATE VIEW [Multiple_Cars] AS
SELECT c.custId AS 'Customer ID', CONCAT( c.custFirstName,' ', c.custLastName) AS 'Customer Name', ca.carModel AS 'Car Model'
FROM [Terps.Customer] c, [Terps.Car] ca
WHERE c.custId = ca.custId AND ca.custId in (SELECT ca.custId
                                             FROM [Terps.Car] ca
                                             GROUP BY ca.custId
                                             HAVING COUNT(ca.custId) > 1)


SELECT * FROM [Multiple_Cars]
```

100 %

Results  Messages

|   | Customer ID | Customer Name | Car Model |
|---|---|---|---|
| 1 | 112278 | Sanya Singh | Mercedes |
| 2 | 118290 | Ronak Gupta | Land Rover |
| 3 | 119089 | Ramya Thakkar | Nissan Versa Note S |
| 4 | 112278 | Sanya Singh | Audi A3 |
| 5 | 119089 | Ramya Thakkar | Audi A3 |
| 6 | 118290 | Ronak Gupta | Audi A3 |

Query executed successfully.    doitsqlx.rhsmith.umd.edu,97...   AD\mthakkar (71)   BUDT703_Project_0502_12   00:00:00   6 rows