Project Summary

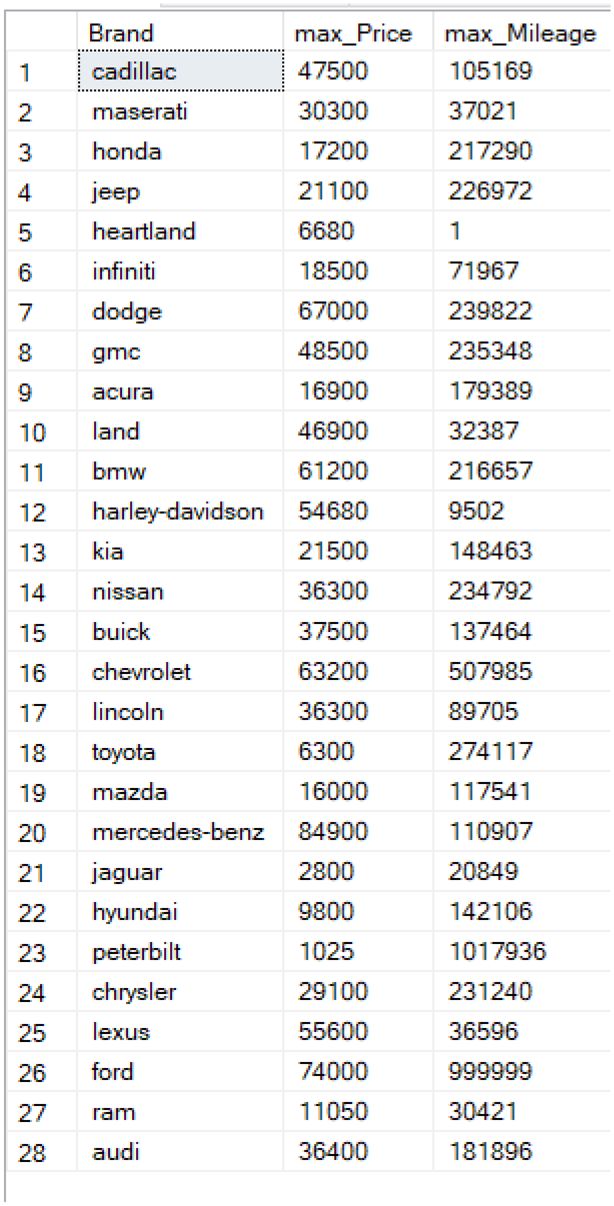
The basis of our database is to provide information for a customer based on their budget and color preferences. We created a dealership table that contained the id of the dealership, the address, the type of dealership and the name. We also created a car table from a dataset of cars, each car has a dealer id as a foreing key, meaning that it belongs to a certain dealership.

Some key features are the relationship between the car table and the dealership table. The dealership table primary key is a forein key in the car table, this allows us to do queries that can find a dealership that meets certain criteria such as a car that fits a customer's budget and is the color of the customers preference, or a dealership that is closest to the customer. The car table contains many cars that way the customer can have more freedom to choose a vehicle that meets their preference.

Some critiques we had were not creating some relation between the customer, cars and car dealer table. We could have created a relation by adding a table called CarOrder. In this table we could have had OrderId as primary key, customer id as a foreign key and type of order as in a car purchase or rental. However due to schedule differences and deadlines we were not able to add this feature. There were also some table design issues that we did not address, for example for customer and car dealership tables the whole address, city and zip code are stored in one column, when we should have stored city and zip code in a different column, that way we could query just the cities of dealerships if needed for the customer. Some disadvantages of using Microsoft SQL is that it is platform dependent, it has support only for windows, this was a problem because each of us had a different operating system, Linux, MacOS, and Windows. Microsoft SQL server management studio is for professional use and must be purchased, however there is a free version for developers that has limited functionality. This is unlike some free open source software such as MySQL or PostgreSQL. There were some advantages such as creating the database and importing tables was easy and intuitive, Microsoft SQL also gives a prompt to make sure the primary key is correct and that all the data types of the columns where set correctly. The entire IDE is easy to use, however at first it was confusing why some queries would not be saved, this was figured out eventually. Some critiques we had on our SQL queries was double checking for correctness. One of the queries that we wrote required the intersection of two tables instead of union, this query returned the wrong information. Some queries such as deleting cars that had mileage and price as 0 were necessary, since this would not make sense in the real world.

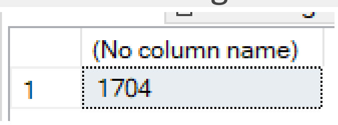
Some important lessons learned include creating schema and EER diagrams for a particular database that we designed using open source software. Creating the EER diagram really helped us understand how our database was going to work. Another lesson is that designing and implementing queries can take some time and several attempts. It is not easy to immediately come up with a query; some research is sometimes needed to find the most efficient but readable query. The most interesting lesson learned was how to create and obtain fake data that could have some semblance of being real and useful in demonstrating a concept. Overall, this program was intended to create a connection between both car dealerships and a database of cars of how many cars belong to each dealership. This can have multiple uses in the automotive industry considering the flexibility of this program.

```
SELECT Brand,MAX(Price) AS max_Price, MAX(Mileage) AS
max_Mileage
FROM Cars
GROUP BY Brand;
```



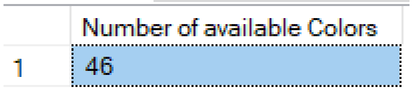
	Brand	max_Price	max_Mileage
1	cadillac	47500	105169
2	maserati	30300	37021
3	honda	17200	217290
4	jeep	21100	226972
5	heartland	6680	1
6	infiniti	18500	71967
7	dodge	67000	239822
8	gmc	48500	235348
9	acura	16900	179389
10	land	46900	32387
11	bmw	61200	216657
12	harley-davidson	54680	9502
13	kia	21500	148463
14	nissan	36300	234792
15	buick	37500	137464
16	chevrolet	63200	507985
17	lincoln	36300	89705
18	toyota	6300	274117
19	mazda	16000	117541
20	mercedes-benz	84900	110907
21	jaguar	2800	20849
22	hyundai	9800	142106
23	peterbilt	1025	1017936
24	chrysler	29100	231240
25	lexus	55600	36596
26	ford	74000	999999
27	ram	11050	30421
28	audi	36400	181896

```
SELECT COUNT(CarID)
FROM Cars
WHERE Mileage<= 50000
```



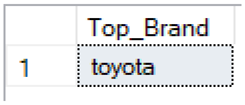
	(No column name)
1	1704

```
SELECT COUNT(DISTINCT Color) AS "Number of available
Colors"
FROM Cars;
```



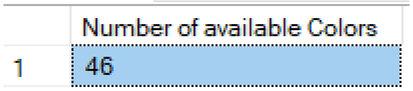
	Number of available Colors
1	46

```
SELECT TOP 1 Brand AS Top_Brand
FROM Cars
GROUP BY Brand
ORDER BY Top_Brand DESC;
```



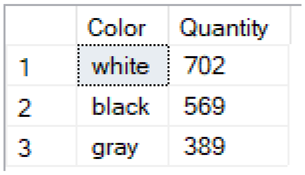
	Top_Brand
1	toyota

```
SELECT COUNT(DISTINCT Color) AS "Number of available
Colors"
FROM Cars;
```



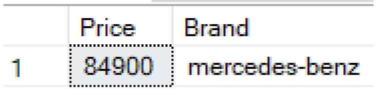
	Number of available Colors
1	46

```
SELECT TOP 3 Color, COUNT(Brand) AS Quantity
FROM Cars
GROUP BY Color
ORDER BY Quantity DESC;
```



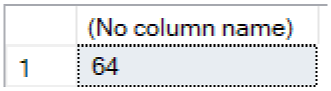
	Color	Quantity
1	white	702
2	black	569
3	gray	389

```
SELECT Price, Brand FROM Cars
WHERE Price = (
SELECT TOP 1 MAX(Price) AS Most_expensive_car
FROM Cars
GROUP BY Price
ORDER BY Most_expensive_car DESC);
```



	Price	Brand
1	84900	mercedes-benz

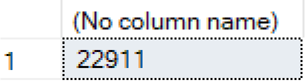
```
SELECT COUNT(*) FROM Cars
WHERE Year<=2010
```



	(No column name)
1	64

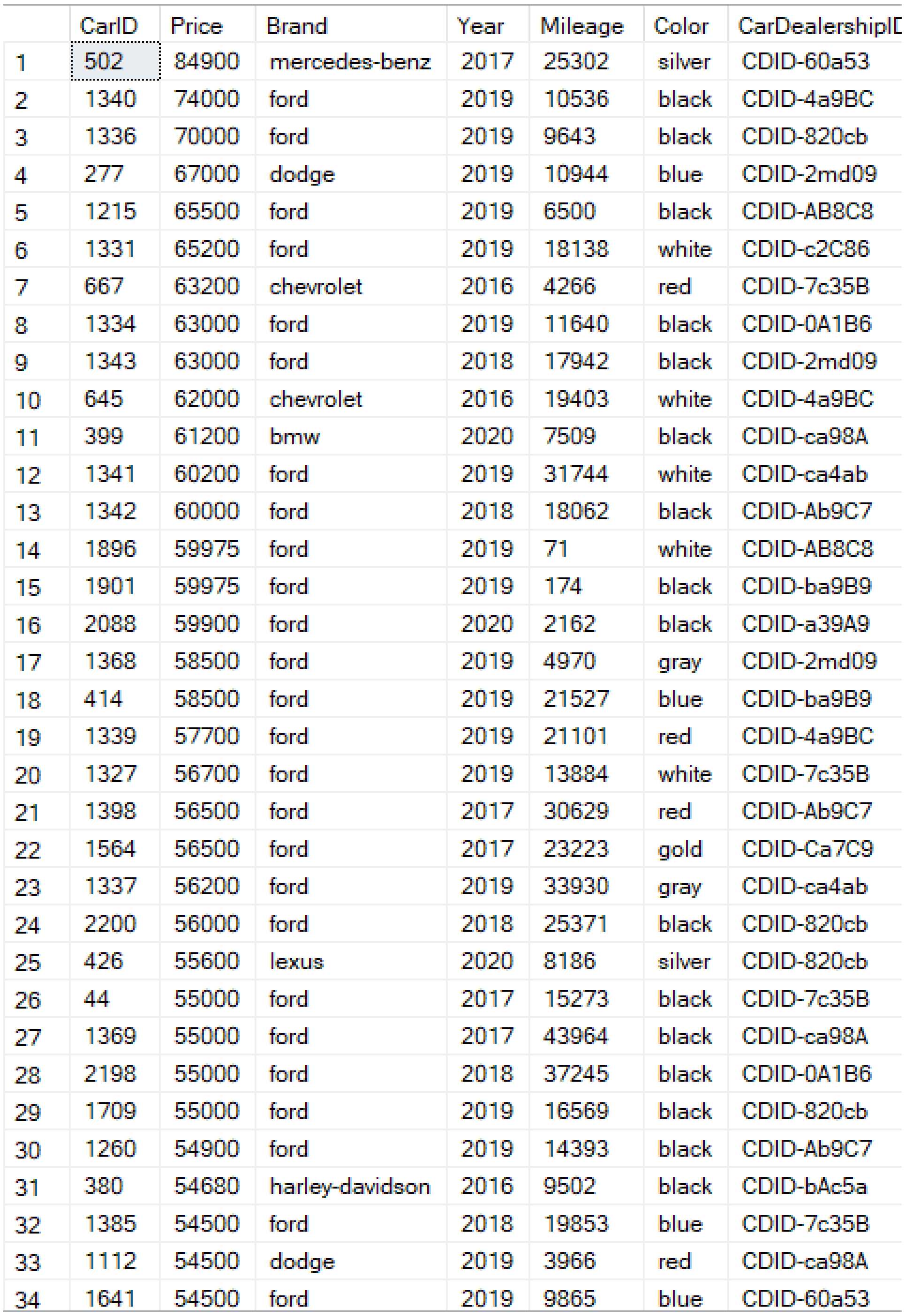
```
SELECT AVG(Price)
```

```
FROM Cars
WHERE Mileage<=1000;
```



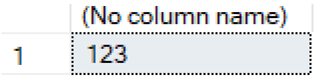
```
  (No column name)
1 22911
```

```
SELECT *
FROM Cars
WHERE Price>10000
ORDER BY Price DESC
```



	CarID	Price	Brand	Year	Mileage	Color	CarDealershipI
1	502	84900	mercedes-benz	2017	25302	silver	CDID-60a53
2	1340	74000	ford	2019	10536	black	CDID-4a9BC
3	1336	70000	ford	2019	9643	black	CDID-820cb
4	277	67000	dodge	2019	10944	blue	CDID-2md09
5	1215	65500	ford	2019	6500	black	CDID-AB8C8
6	1331	65200	ford	2019	18138	white	CDID-c2C86
7	667	63200	chevrolet	2016	4266	red	CDID-7c35B
8	1334	63000	ford	2019	11640	black	CDID-0A1B6
9	1343	63000	ford	2018	17942	black	CDID-2md09
10	645	62000	chevrolet	2016	19403	white	CDID-4a9BC
11	399	61200	bmw	2020	7509	black	CDID-ca98A
12	1341	60200	ford	2019	31744	white	CDID-ca4ab
13	1342	60000	ford	2018	18062	black	CDID-Ab9C7
14	1896	59975	ford	2019	71	white	CDID-AB8C8
15	1901	59975	ford	2019	174	black	CDID-ba9B9
16	2088	59900	ford	2020	2162	black	CDID-a39A9
17	1368	58500	ford	2019	4970	gray	CDID-2md09
18	414	58500	ford	2019	21527	blue	CDID-ba9B9
19	1339	57700	ford	2019	21101	red	CDID-4a9BC
20	1327	56700	ford	2019	13884	white	CDID-7c35B
21	1398	56500	ford	2017	30629	red	CDID-Ab9C7
22	1564	56500	ford	2017	23223	gold	CDID-Ca7C9
23	1337	56200	ford	2019	33930	gray	CDID-ca4ab
24	2200	56000	ford	2018	25371	black	CDID-820cb
25	426	55600	lexus	2020	8186	silver	CDID-820cb
26	44	55000	ford	2017	15273	black	CDID-7c35B
27	1369	55000	ford	2017	43964	black	CDID-ca98A
28	2198	55000	ford	2018	37245	black	CDID-0A1B6
29	1709	55000	ford	2019	16569	black	CDID-820cb
30	1260	54900	ford	2019	14393	black	CDID-Ab9C7
31	380	54680	harley-davidson	2016	9502	black	CDID-bAc5a
32	1385	54500	ford	2018	19853	blue	CDID-7c35B
33	1112	54500	dodge	2019	3966	red	CDID-ca98A
34	1641	54500	ford	2019	9865	blue	CDID-60a53

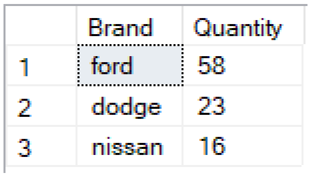
```
SELECT COUNT(CarID) FROM Cars
WHERE Cars.CarDealershipID = (SELECT CarDealershipID
FROM [CarDealer ] WHERE [CarDealer ].CarDealershipName =
'Auto Trade Corporation')
```



	(No column name)
1	123

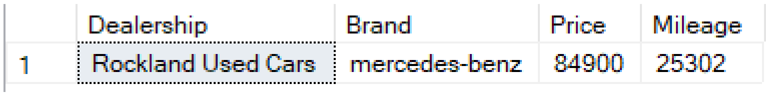
```
SELECT TOP 3 Brand, COUNT(Brand) AS Quantity
```

```
FROM Cars
WHERE Cars.CarDealershipID = (SELECT CarDealershipID
FROM [CarDealer ] WHERE [CarDealer ].CarDealershipName =
'Auto Trade Corporation')
GROUP BY Brand
ORDER BY Quantity DESC;
```



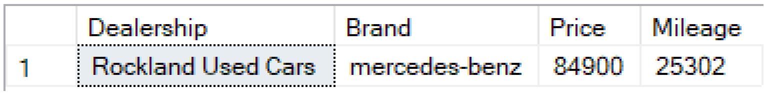
	Brand	Quantity
1	ford	58
2	dodge	23
3	nissan	16

```
DECLARE @CDID nvarchar(50);
SET @CDID = (SELECT CarDealershipID FROM Cars
WHERE Price = (
SELECT TOP 1 MAX(Price)
FROM Cars
GROUP BY Price
ORDER BY Price DESC))

SELECT TOP 1 CD.CarDealershipName AS Dealership,
C.Brand, C.Price, C.Mileage
FROM [CarDealer ] CD, Cars C
WHERE CD.CarDealershipID =@CDID
ORDER BY Price DESC;
```



	Dealership	Brand	Price	Mileage
1	Rockland Used Cars	mercedes-benz	84900	25302

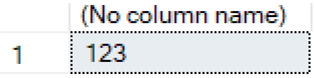
```
SELECT CD.CarDealershipName AS Dealership, CD.Address,
C.Price, C.Mileage, C.Color
FROM [CarDealer ] CD, Cars C
WHERE C.Color =(SELECT ColorPreference FROM Customer
WHERE CustomerName='Peter Brown')
AND C.Price<=(SELECT Budget FROM Customer WHERE
CustomerName='Peter Brown')
```



	Dealership	Brand	Price	Mileage
1	Rockland Used Cars	mercedes-benz	84900	25302

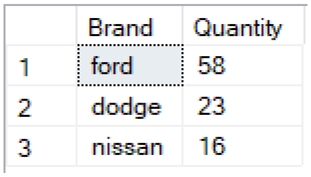
```
SELECT COUNT(CarID) FROM Cars
```

```
WHERE Cars.CarDealershipID = (SELECT CarDealershipID
FROM [CarDealer ] WHERE [CarDealer ].CarDealershipName =
'Auto Trade Corporation')
```



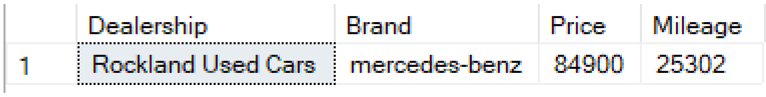
	(No column name)
1	123

```
SELECT TOP 3 Brand, COUNT(Brand) AS Quantity
FROM Cars
WHERE Cars.CarDealershipID = (SELECT CarDealershipID
FROM [CarDealer ] WHERE [CarDealer ].CarDealershipName =
'Auto Trade Corporation')
GROUP BY Brand
ORDER BY Quantity DESC;
```



	Brand	Quantity
1	ford	58
2	dodge	23
3	nissan	16

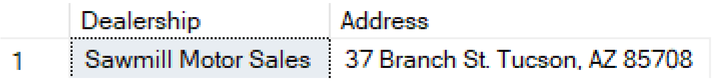
```
DECLARE @CDID nvarchar(50);
SET @CDID = (SELECT CarDealershipID FROM Cars
WHERE Price = (
SELECT TOP 1 MAX(Price)
FROM Cars
GROUP BY Price
ORDER BY Price DESC))

SELECT TOP 1 CD.CarDealershipName AS Dealership,
C.Brand, C.Price, C.Mileage
FROM [CarDealer ] CD, Cars C
WHERE CD.CarDealershipID =@CDID
ORDER BY Price DESC;
```



	Dealership	Brand	Price	Mileage
1	Rockland Used Cars	mercedes-benz	84900	25302

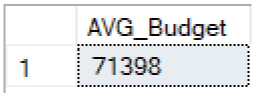
```
DECLARE @temp nvarchar(50)
SET @temp = (SELECT Address FROM Customer WHERE
CustomerName = 'Clay Serrano')
DECLARE @ZIP_Code nvarchar(50)
SET @ZIP_Code =  RIGHT(@temp, 5)
```

```
SELECT CarDealershipName AS Dealership, Address FROM
[CarDealer ]
WHERE Address LIKE '%, AZ '+@ZIP_Code
```



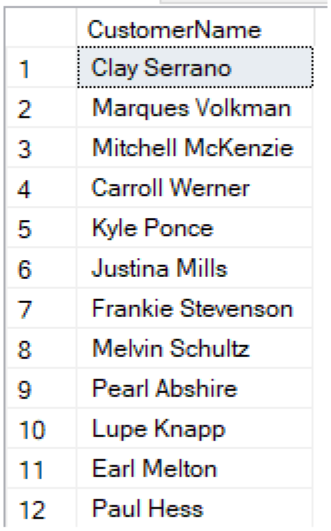
	Dealership	Address
1	Sawmill Motor Sales	37 Branch St. Tucson, AZ 85708

```
SELECT AVG(Budget) AS AVG_Budget FROM Customer
```



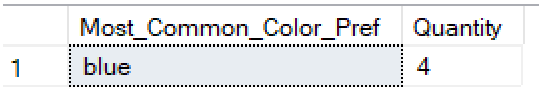
	AVG_Budget
1	71398

```
SELECT CustomerName FROM Customer WHERE
Birthdate>'19800101'
```



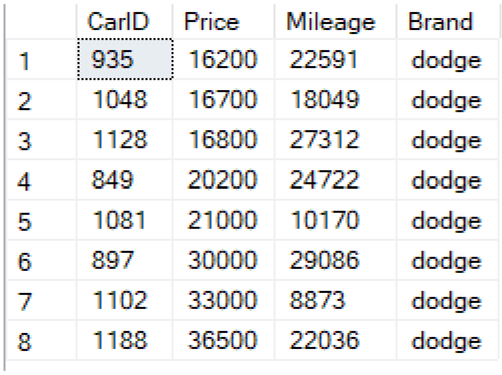
	CustomerName
1	Clay Serrano
2	Marques Volkman
3	Mitchell McKenzie
4	Carroll Werner
5	Kyle Ponce
6	Justina Mills
7	Frankie Stevenson
8	Melvin Schultz
9	Pearl Abshire
10	Lupe Knapp
11	Earl Melton
12	Paul Hess

```
SELECT TOP 1 ColorPreference AS Most_Common_Color_Pref,
COUNT(ColorPreference) AS Quantity
FROM Customer
GROUP BY ColorPreference
ORDER BY Quantity DESC;
```



	Most_Common_Color_Pref	Quantity
1	blue	4

```
DECLARE @CustomerBudget int
SET @CustomerBudget = (SELECT Budget FROM Customer WHERE
CustomerName = 'Marques Volkman')
SELECT CarID, Price, Mileage,Brand FROM Cars
WHERE CarDealershipID = (SELECT CarDealershipID FROM
[CarDealer ] WHERE CarDealershipName = 'Wide World Inc')
AND Price<=@CustomerBudget
AND Mileage<=30000
AND Brand = 'dodge'
AND Year>2015
ORDER BY Price
```



	CarID	Price	Mileage	Brand
1	935	16200	22591	dodge
2	1048	16700	18049	dodge
3	1128	16800	27312	dodge
4	849	20200	24722	dodge
5	1081	21000	10170	dodge
6	897	30000	29086	dodge
7	1102	33000	8873	dodge
8	1188	36500	22036	dodge