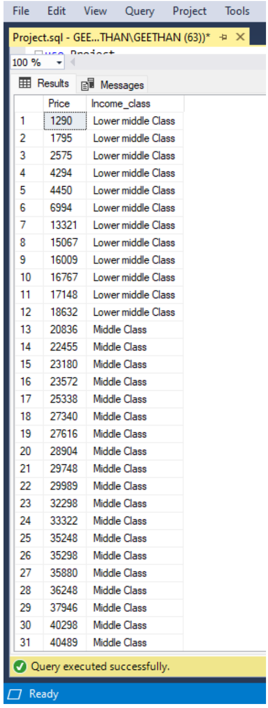
SQL PROJECT -CAR LAUNCH ANALYSIS IN UK MARKET

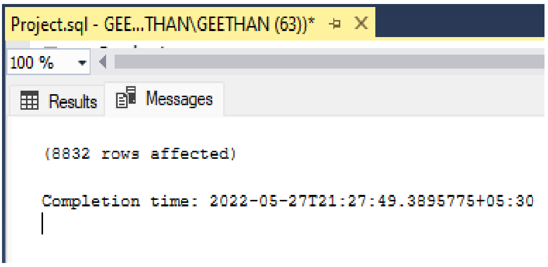
a. Create an analysis to find income class of UK citizens based on price of Cars(You can use per-capita income in UK from internet sources)

```
Select Price,
case when Price >70000 then 'Upper Class'
when price >= 49100 then 'Upper Middle Class'
when price >=20000 then'Middle Class'
Else 'Lower middle Class'
End as Income_class  from Audi union
Select Price,
case when Price >70000 then 'Upper Class'
when price >= 49100 then 'Upper Middle Class'
when price >=20000  then'Middle Class'
Else 'Lower middle Class'
End as Income_class  from bmw union
Select Price,
case when Price >70000 then 'Upper Class'
when price >= 49100 then 'Upper Middle Class'
when price >=20000 then'Middle Class'
Else 'Lower middle Class'
End as Income_class  from hyndai union
Select Price,
case when Price >70000 then 'Upper Class'
when price >= 49100 then 'Upper Middle Class'
when price >=20000 then'Middle Class'
Else 'Lower middle Class'
End as Income_class  from merc union
Select Price,
case when Price >70000 then 'Upper Class'
when price >= 49100 then 'Upper Middle Class'
when price >=20000 then'Middle Class'
Else 'Lower middle Class'
End as Income_class  from cclass;
```



	Price	Income_class
1	1290	Lower middle Class
2	1795	Lower middle Class
3	2575	Lower middle Class
4	4294	Lower middle Class
5	4450	Lower middle Class
6	6994	Lower middle Class
7	13321	Lower middle Class
8	15067	Lower middle Class
9	16009	Lower middle Class
10	16767	Lower middle Class
11	17148	Lower middle Class
12	18632	Lower middle Class
13	20836	Middle Class
14	22455	Middle Class
15	23180	Middle Class
16	23572	Middle Class
17	25338	Middle Class
18	27340	Middle Class
19	27616	Middle Class
20	28904	Middle Class
21	29748	Middle Class
22	29989	Middle Class
23	32298	Middle Class
24	33322	Middle Class
25	35248	Middle Class
26	35298	Middle Class
27	35880	Middle Class
28	36248	Middle Class
29	37946	Middle Class
30	40298	Middle Class
31	40489	Middle Class

Query executed successfully.



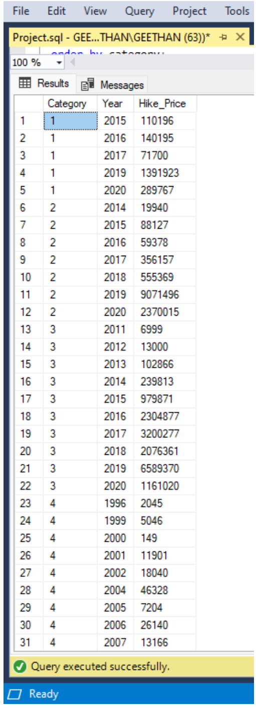
```
(8832 rows affected)

Completion time: 2022-05-27T21:27:49.3895775+05:30
```

b. Categorize the cars on the basis of their price(Create as many buckets as you want as per your understanding of data) and analyze the:

a. price changes across the years and identify the categories which has seen significant jump in its price

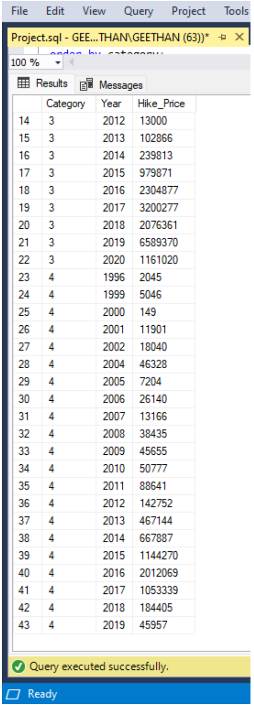
- For Bmw:

```
Select Category,Year,Sum(Difference_) as Hike_Price from(
 Select Year,(Total_Price-Lead_price) as Difference_ ,Category from
(select Year,Price as Total_Price ,lead(Price,1) over(Order by Year) as Lead_Price,case when
Price >70000 then '1'
when Price  >=41900 then '2'
when Price >=20000 then'3'
Else '4'
End as category
 from bmw )Aa where Total_Price>Lead_Price )BB
 group by year,category
 order by category,year;
```



	Category	Year	Hike_Price
1	1	2015	110196
2	1	2016	140195
3	1	2017	71700
4	1	2019	1391923
5	1	2020	289767
6	2	2014	19940
7	2	2015	88127
8	2	2016	59378
9	2	2017	356157
10	2	2018	555369
11	2	2019	9071496
12	2	2020	2370015
13	3	2011	6999
14	3	2012	13000
15	3	2013	102866
16	3	2014	239813
17	3	2015	979871
18	3	2016	2304877
19	3	2017	3200277
20	3	2018	2076361
21	3	2019	6589370
22	3	2020	1161020
23	4	1996	2045
24	4	1999	5046
25	4	2000	149
26	4	2001	11901
27	4	2002	18040
28	4	2004	46328
29	4	2005	7204
30	4	2006	26140
31	4	2007	13166

Query executed successfully.

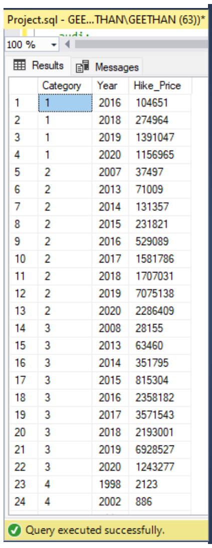


	Category	Year	Hike_Price
14	3	2012	13000
15	3	2013	102866
16	3	2014	239813
17	3	2015	979871
18	3	2016	2304877
19	3	2017	3200277
20	3	2018	2076361
21	3	2019	6589370
22	3	2020	1161020
23	4	1996	2045
24	4	1999	5046
25	4	2000	149
26	4	2001	11901
27	4	2002	18040
28	4	2004	46328
29	4	2005	7204
30	4	2006	26140
31	4	2007	13166
32	4	2008	38435
33	4	2009	45655
34	4	2010	50777
35	4	2011	88641
36	4	2012	142752
37	4	2013	467144
38	4	2014	667887
39	4	2015	1144270
40	4	2016	2012069
41	4	2017	1053339
42	4	2018	184405
43	4	2019	45957

Query executed successfully.

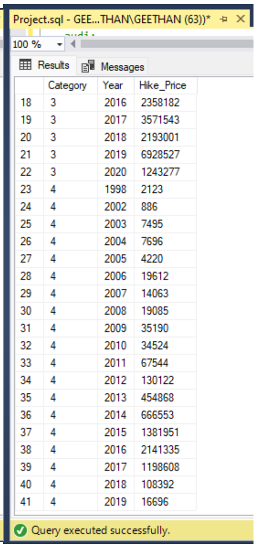
- For Audi:

```
Select Category,Year,Sum(Difference_) as Hike_Price from(
 Select Year,(Total_Price-Lead_price) as Difference_ ,Category from
(select Year,Price as Total_Price ,lead(Price,1) over(Order by Year) as Lead_Price,case when
Price >70000 then '1'
when Price  >=41900 then '2'
when Price >=20000 then'3'
Else '4'
End as category
 from audi)Aa where Total_Price>Lead_Price )BB
 group by year,category
 order by category,year;
```



	Category	Year	Hike_Price
1	1	2016	104651
2	1	2018	274964
3	1	2019	1391047
4	1	2020	1156965
5	2	2007	37497
6	2	2013	71009
7	2	2014	131357
8	2	2015	231821
9	2	2016	529089
10	2	2017	1581786
11	2	2018	1707031
12	2	2019	7075138
13	2	2020	2286409
14	3	2008	28155
15	3	2013	63460
16	3	2014	351795
17	3	2015	815304
18	3	2016	2358182
19	3	2017	3571543
20	3	2018	2193001
21	3	2019	6928527
22	3	2020	1243277
23	4	1998	2123
24	4	2002	886

Query executed successfully.

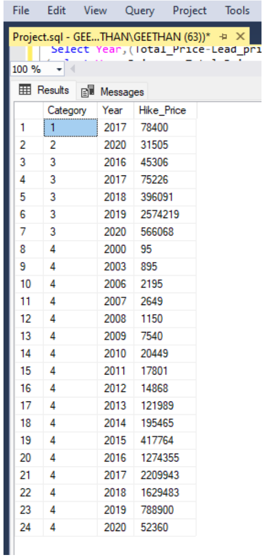


	Category	Year	Hike_Price
18	3	2016	2358182
19	3	2017	3571543
20	3	2018	2193001
21	3	2019	6928527
22	3	2020	1243277
23	4	1998	2123
24	4	2002	886
25	4	2003	7495
26	4	2004	7696
27	4	2005	4220
28	4	2006	19612
29	4	2007	14063
30	4	2008	19085
31	4	2009	35190
32	4	2010	34524
33	4	2011	67544
34	4	2012	130122
35	4	2013	454868
36	4	2014	666553
37	4	2015	1381951
38	4	2016	2141335
39	4	2017	1198608
40	4	2018	108392
41	4	2019	16696

Query executed successfully.

- For Hyndai:

```
Select Category,Year,Sum(Difference_) as Hike_Price from(
 Select Year,(Total_Price-Lead_price) as Difference_ ,Category from
(select Year,Price as Total_Price ,lead(Price,1) over(Order by Year) as Lead_Price,case when
Price >70000 then '1'
when Price  >=41900 then '2'
when Price >=20000 then'3'
Else '4'
End as category
 from hyndai )Aa where Total_Price>Lead_Price )BB
 group by year,category
 order by category,year;
```

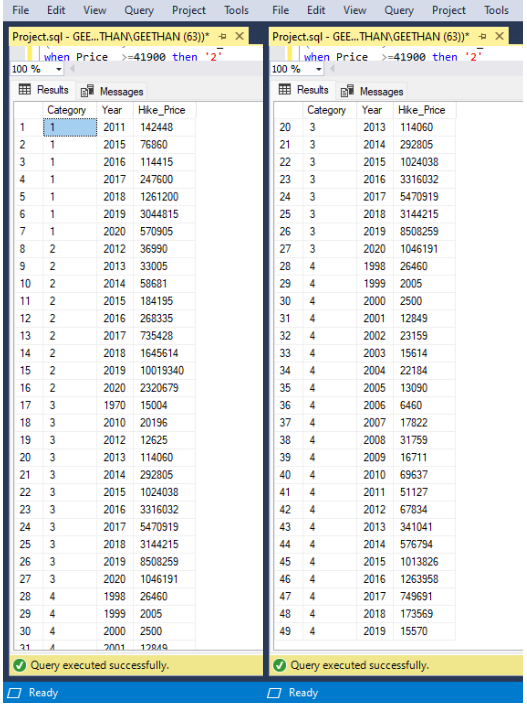


```
Select Year,(Total_Price-Lead_pri
```

	Category	Year	Hike_Price
1	1	2017	78400
2	2	2020	31505
3	3	2016	45306
4	3	2017	75226
5	3	2018	396091
6	3	2019	2574219
7	3	2020	566068
8	4	2000	95
9	4	2003	895
10	4	2006	2195
11	4	2007	2649
12	4	2008	1150
13	4	2009	7540
14	4	2010	20449
15	4	2011	17801
16	4	2012	14868
17	4	2013	121989
18	4	2014	195465
19	4	2015	417764
20	4	2016	1274355
21	4	2017	2209943
22	4	2018	1629483
23	4	2019	788900
24	4	2020	52360

- For Merc:

```
Select Category,Year,Sum(Difference_) as Hike_Price from(
 Select Year,(Total_Price-Lead_price) as Difference_ ,Category from
(select  Year,Price  as  Total_Price  ,lead(Price,1)  over(Order  by  Year)  as  Lead_Price,case  when
Price >70000 then '1'
when Price  >=41900 then '2'
when Price >=20000 then'3'
Else '4'
End as category
 from merc)Aa where Total_Price>Lead_Price )BB
 group by year,category
 order by category,year;
```


File Edit View Query Project Tools
Project.sql - GEE...THAN\GEETHAN (63))*
100 %
Results
Messages
Ready

```
when Price  >=41900 then '2'
```

	Category	Year	Hike_Price
1	1	2011	142448
2	1	2015	76860
3	1	2016	114415
4	1	2017	247600
5	1	2018	1261200
6	1	2019	3044815
7	1	2020	570905
8	2	2012	36990
9	2	2013	33005
10	2	2014	58681
11	2	2015	184195
12	2	2016	268335
13	2	2017	735428
14	2	2018	1645614
15	2	2019	10019340
16	2	2020	2320679
17	3	1970	15004
18	3	2010	20196
19	3	2012	12625
20	3	2013	114060
21	3	2014	292805
22	3	2015	1024038
23	3	2016	3316032
24	3	2017	5470919
25	3	2018	3144215
26	3	2019	8508259
27	3	2020	1046191
28	4	1998	26460
29	4	1999	2005
30	4	2000	2500
31	4	2001	12849

Query executed successfully.

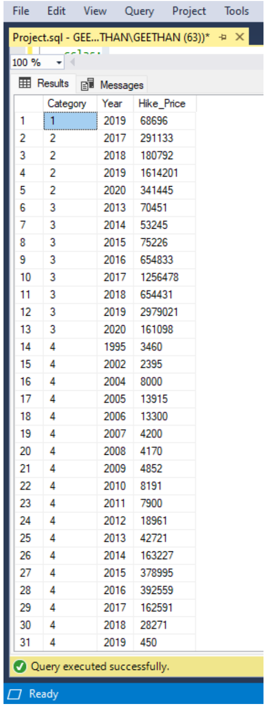
```
when Price  >=41900 then '2'
```

	Category	Year	Hike_Price
20	3	2013	114060
21	3	2014	292805
22	3	2015	1024038
23	3	2016	3316032
24	3	2017	5470919
25	3	2018	3144215
26	3	2019	8508259
27	3	2020	1046191
28	4	1998	26460
29	4	1999	2005
30	4	2000	2500
31	4	2001	12849
32	4	2002	23159
33	4	2003	15614
34	4	2004	22184
35	4	2005	13090
36	4	2006	6460
37	4	2007	17822
38	4	2008	31759
39	4	2009	16711
40	4	2010	69637
41	4	2011	51127
42	4	2012	67834
43	4	2013	341041
44	4	2014	576794
45	4	2015	1013826
46	4	2016	1263958
47	4	2017	749691
48	4	2018	173569
49	4	2019	15570

Query executed successfully.

- For cclass:

```
  Select Category,Year,Sum(Difference_) as Hike_Price from(
 Select Year,(Total_Price-Lead_price) as Difference_ ,Category from
(select  Year,Price  as  Total_Price  ,lead(Price,1)  over(Order  by  Year)  as  Lead_Price,case  when
Price >70000 then '1'
when Price  >=41900 then '2'
when Price >=20000 then'3'
Else '4'
End as category
 from cclass)Aa where Total_Price>Lead_Price )BB
 group by year,category
 order by category,year;
```



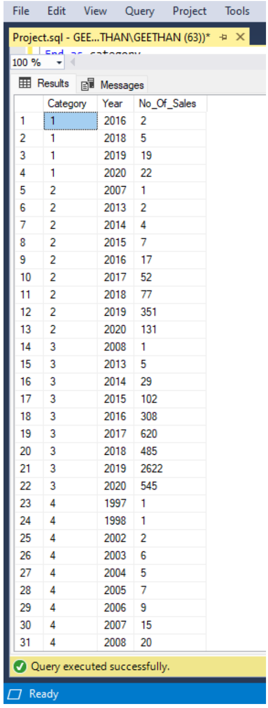
	Category	Year	Hike_Price
1	1	2019	68696
2	2	2017	291133
3	2	2018	180792
4	2	2019	1614201
5	2	2020	341445
6	3	2013	70451
7	3	2014	53245
8	3	2015	75226
9	3	2016	654833
10	3	2017	1256478
11	3	2018	654431
12	3	2019	2979021
13	3	2020	161098
14	4	1995	3460
15	4	2002	2395
16	4	2004	8000
17	4	2005	13915
18	4	2006	13300
19	4	2007	4200
20	4	2008	4170
21	4	2009	4852
22	4	2010	8191
23	4	2011	7900
24	4	2012	18961
25	4	2013	42721
26	4	2014	163227
27	4	2015	378995
28	4	2016	392559
29	4	2017	162591
30	4	2018	28271
31	4	2019	450

Query executed successfully.

b. changes in no of cars sold across the years and identify the categories which has seen significant jump in its sales

- For Audi:

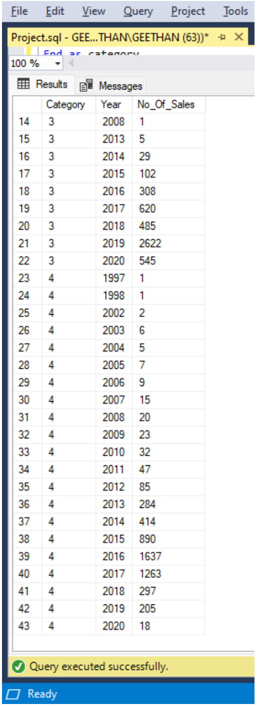
```
Select Category,Year,sum(Sales)as No_Of_Sales from
(Select Year,Price,Count(ID) as Sales,
case when Price >70000 then '1'
when Price  >=41900 then '2'
when Price >=20000 then'3'
Else '4'
End as category
from  audi group by year,price) Audi
group by year,category
Order by category,year;
```



```
End as category
```

	Category	Year	No_Of_Sales
1	1	2016	2
2	1	2018	5
3	1	2019	19
4	1	2020	22
5	2	2007	1
6	2	2013	2
7	2	2014	4
8	2	2015	7
9	2	2016	17
10	2	2017	52
11	2	2018	77
12	2	2019	351
13	2	2020	131
14	3	2008	1
15	3	2013	5
16	3	2014	29
17	3	2015	102
18	3	2016	308
19	3	2017	620
20	3	2018	485
21	3	2019	2622
22	3	2020	545
23	4	1997	1
24	4	1998	1
25	4	2002	2
26	4	2003	6
27	4	2004	5
28	4	2005	7
29	4	2006	9
30	4	2007	15
31	4	2008	20

Query executed successfully.



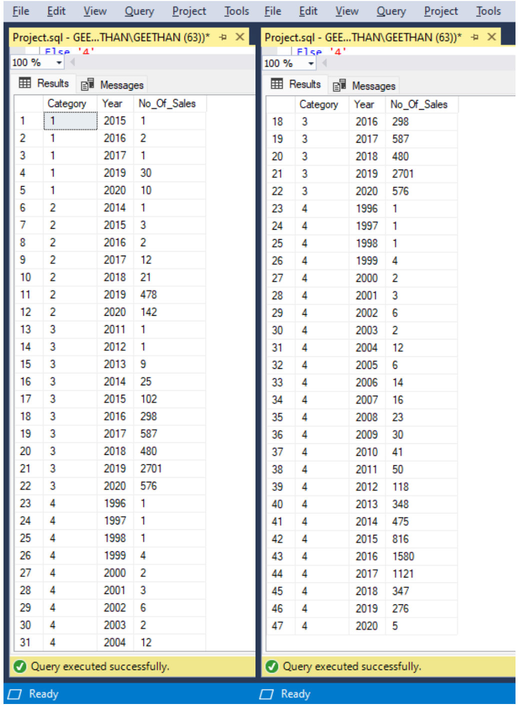
```
End as category
```

	Category	Year	No_Of_Sales
14	3	2008	1
15	3	2013	5
16	3	2014	29
17	3	2015	102
18	3	2016	308
19	3	2017	620
20	3	2018	485
21	3	2019	2622
22	3	2020	545
23	4	1997	1
24	4	1998	1
25	4	2002	2
26	4	2003	6
27	4	2004	5
28	4	2005	7
29	4	2006	9
30	4	2007	15
31	4	2008	20
32	4	2009	23
33	4	2010	32
34	4	2011	47
35	4	2012	85
36	4	2013	284
37	4	2014	414
38	4	2015	890
39	4	2016	1637
40	4	2017	1263
41	4	2018	297
42	4	2019	205
43	4	2020	18

Query executed successfully.

- For bmw:

```
Select Category,Year,sum(Sales)as No_Of_Sales from
(Select Year,Price,Count(ID) as Sales,
case when Price >70000 then '1'
when Price  >=41900 then '2'
when Price >=20000 then'3'
Else '4'
End as category
from  bmw group by year,price)bmw
group by year,category
Order by category,year;
```

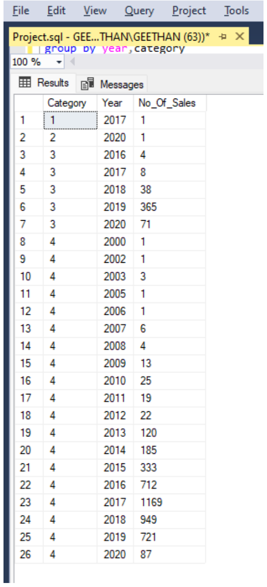

File Edit View Query Project Tools
Project.sql - GEE...THAN\GEETHAN (63))*
Else '4'
100 %
Results
Messages
Ready

	Category	Year	No_Of_Sales
1	1	2015	1
2	1	2016	2
3	1	2017	1
4	1	2019	30
5	1	2020	10
6	2	2014	1
7	2	2015	3
8	2	2016	2
9	2	2017	12
10	2	2018	21
11	2	2019	478
12	2	2020	142
13	3	2011	1
14	3	2012	1
15	3	2013	9
16	3	2014	25
17	3	2015	102
18	3	2016	298
19	3	2017	587
20	3	2018	480
21	3	2019	2701
22	3	2020	576
23	4	1996	1
24	4	1997	1
25	4	1998	1
26	4	1999	4
27	4	2000	2
28	4	2001	3
29	4	2002	6
30	4	2003	2
31	4	2004	12
32	4	2005	6
33	4	2006	14
34	4	2007	16
35	4	2008	23
36	4	2009	30
37	4	2010	41
38	4	2011	50
39	4	2012	118
40	4	2013	348
41	4	2014	475
42	4	2015	816
43	4	2016	1580
44	4	2017	1121
45	4	2018	347
46	4	2019	276
47	4	2020	5

Query executed successfully.

- For Hyndai:

```
Select Category,Year,sum(Sales)as No_Of_Sales from
(Select Year,Price,Count(ID) as Sales,
case when Price >70000 then '1'
when Price  >=41900 then '2'
when Price >=20000 then'3'
Else '4'
End as category
from  hyndai group by year,price)hyndai
group by year,category
Order by category,year;
```

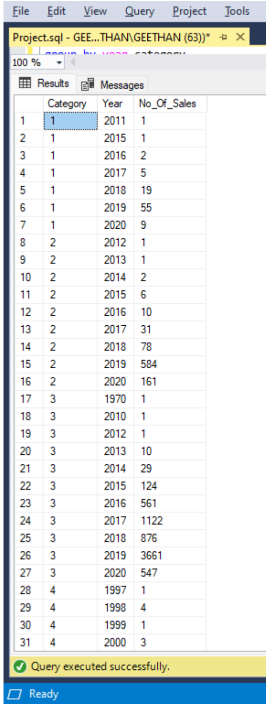


```
group by year,category
```

	Category	Year	No_Of_Sales
1	1	2017	1
2	2	2020	1
3	3	2016	4
4	3	2017	8
5	3	2018	38
6	3	2019	365
7	3	2020	71
8	4	2000	1
9	4	2002	1
10	4	2003	3
11	4	2005	1
12	4	2006	1
13	4	2007	6
14	4	2008	4
15	4	2009	13
16	4	2010	25
17	4	2011	19
18	4	2012	22
19	4	2013	120
20	4	2014	185
21	4	2015	333
22	4	2016	712
23	4	2017	1169
24	4	2018	949
25	4	2019	721
26	4	2020	87

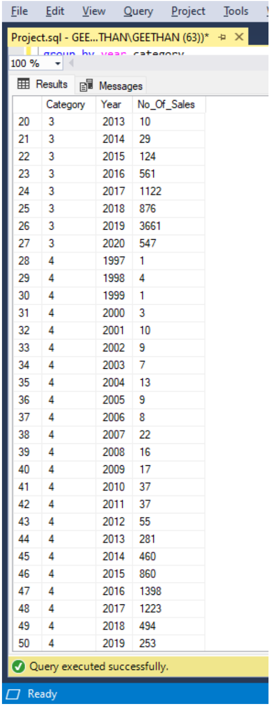
- For merc:

```
Select Category,Year,sum(Sales)as No_Of_Sales from
(Select Year,Price,Count(ID) as Sales,
case when Price >70000 then '1'
when Price  >=41900 then '2'
when Price >=20000 then'3'
Else '4'
End as category
from  merc group by year,price) merc
group by year,category
Order by category,year;
```



	Category	Year	No_Of_Sales
1	1	2011	1
2	1	2015	1
3	1	2016	2
4	1	2017	5
5	1	2018	19
6	1	2019	55
7	1	2020	9
8	2	2012	1
9	2	2013	1
10	2	2014	2
11	2	2015	6
12	2	2016	10
13	2	2017	31
14	2	2018	78
15	2	2019	584
16	2	2020	161
17	3	1970	1
18	3	2010	1
19	3	2012	1
20	3	2013	10
21	3	2014	29
22	3	2015	124
23	3	2016	561
24	3	2017	1122
25	3	2018	876
26	3	2019	3661
27	3	2020	547
28	4	1997	1
29	4	1998	4
30	4	1999	1
31	4	2000	3

Query executed successfully.

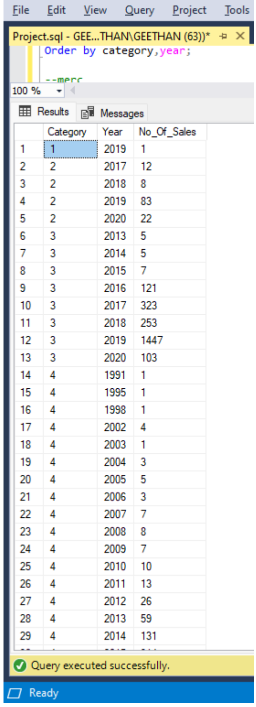


	Category	Year	No_Of_Sales
20	3	2013	10
21	3	2014	29
22	3	2015	124
23	3	2016	561
24	3	2017	1122
25	3	2018	876
26	3	2019	3661
27	3	2020	547
28	4	1997	1
29	4	1998	4
30	4	1999	1
31	4	2000	3
32	4	2001	10
33	4	2002	9
34	4	2003	7
35	4	2004	13
36	4	2005	9
37	4	2006	8
38	4	2007	22
39	4	2008	16
40	4	2009	17
41	4	2010	37
42	4	2011	37
43	4	2012	55
44	4	2013	281
45	4	2014	460
46	4	2015	860
47	4	2016	1398
48	4	2017	1223
49	4	2018	494
50	4	2019	253

Query executed successfully.

- ➢ For CClass:

```
Select Category,Year,sum(Sales)as No_Of_Sales from
(Select Year,Price,Count(ID) as Sales,
case when Price >70000 then '1'
when Price  >=41900 then '2'
when Price >=20000 then'3'
Else '4'
End as category
from  cclass group by year,price)cclass
group by year,category
Order by category,year;
```

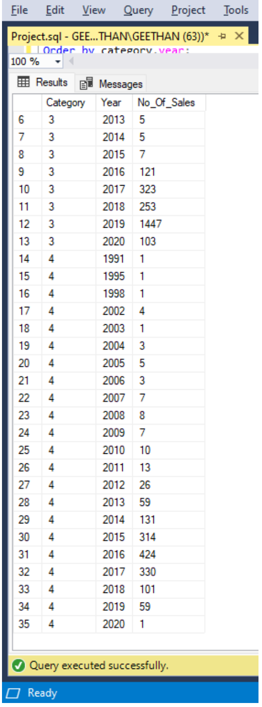


File Edit View Query Project Tools
Project.sql - GEE...THAN\GEETHAN (63))*
100 %
Results
Messages

```
Order by category,year;

--merc
```

	Category	Year	No_Of_Sales
1	1	2019	1
2	2	2017	12
3	2	2018	8
4	2	2019	83
5	2	2020	22
6	3	2013	5
7	3	2014	5
8	3	2015	7
9	3	2016	121
10	3	2017	323
11	3	2018	253
12	3	2019	1447
13	3	2020	103
14	4	1991	1
15	4	1995	1
16	4	1998	1
17	4	2002	4
18	4	2003	1
19	4	2004	3
20	4	2005	5
21	4	2006	3
22	4	2007	7
23	4	2008	8
24	4	2009	7
25	4	2010	10
26	4	2011	13
27	4	2012	26
28	4	2013	59
29	4	2014	131

Query executed successfully.



File Edit View Query Project Tools
Project.sql - GEE...THAN\GEETHAN (63))*
100 %
Results
Messages

```
Order by category,year;
```

	Category	Year	No_Of_Sales
6	3	2013	5
7	3	2014	5
8	3	2015	7
9	3	2016	121
10	3	2017	323
11	3	2018	253
12	3	2019	1447
13	3	2020	103
14	4	1991	1
15	4	1995	1
16	4	1998	1
17	4	2002	4
18	4	2003	1
19	4	2004	3
20	4	2005	5
21	4	2006	3
22	4	2007	7
23	4	2008	8
24	4	2009	7
25	4	2010	10
26	4	2011	13
27	4	2012	26
28	4	2013	59
29	4	2014	131
30	4	2015	314
31	4	2016	424
32	4	2017	330
33	4	2018	101
34	4	2019	59
35	4	2020	1

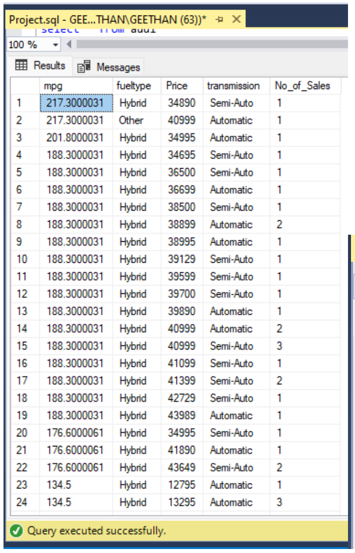
Query executed successfully.

- ✓ Using the above identified categories for both points (a) & (b), do a root cause analysis to identify the probable reason for their increase. For, e.g., Its fuel efficiency as compared to other types of car could be a reason.
- ✓ The price of car differs for various types of car with similar transmission.
- ✓ Increase of Fuel price makes customers to buy automatic and semi-automatic.
- ✓ Mileage of the car is compared to the price .
- ✓ Models that customer find suitable have more sales.
- ✓ Cars that have more Price has less Sales percentage compared to other category of cars.

c. Find relationship between fuel efficiency & price of car/sales of car/fuel type/, etc.

- ❖ For merc:

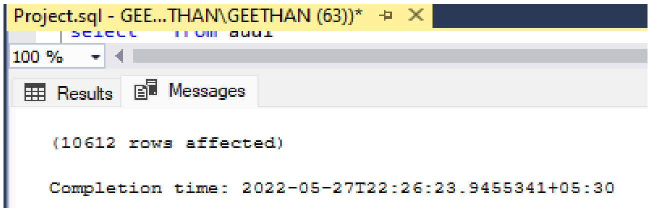
```
Select mpg,fueltype,Price,C.transmission,count(A.ID) as No_of_Sales from merc as A inner join
fueltype as B on A.fuel_ID
=B.fuel_ID inner join transmission as c on C.id=A.transmission_ID
Group by fueltype,price,mpg,C.transmission
order by mpg desc ;
```



Project.sql - GEE...THAN\GEETHAN (63))*

	mpg	fueltype	Price	transmission	No_of_Sales
1	217.3000031	Hybrid	34890	Semi-Auto	1
2	217.3000031	Other	40999	Automatic	1
3	201.8000031	Hybrid	34995	Automatic	1
4	188.3000031	Hybrid	34695	Semi-Auto	1
5	188.3000031	Hybrid	36500	Semi-Auto	1
6	188.3000031	Hybrid	36699	Automatic	1
7	188.3000031	Hybrid	38500	Semi-Auto	1
8	188.3000031	Hybrid	38899	Automatic	2
9	188.3000031	Hybrid	38995	Automatic	1
10	188.3000031	Hybrid	39129	Semi-Auto	1
11	188.3000031	Hybrid	39599	Semi-Auto	1
12	188.3000031	Hybrid	39700	Semi-Auto	1
13	188.3000031	Hybrid	39890	Automatic	1
14	188.3000031	Hybrid	40999	Automatic	2
15	188.3000031	Hybrid	40999	Semi-Auto	3
16	188.3000031	Hybrid	41099	Semi-Auto	1
17	188.3000031	Hybrid	41399	Semi-Auto	2
18	188.3000031	Hybrid	42729	Semi-Auto	1
19	188.3000031	Hybrid	43989	Automatic	1
20	176.6000061	Hybrid	34995	Semi-Auto	1
21	176.6000061	Hybrid	41890	Automatic	1
22	176.6000061	Hybrid	43649	Semi-Auto	2
23	134.5	Hybrid	12795	Automatic	1
24	134.5	Hybrid	13295	Automatic	3

Query executed successfully.

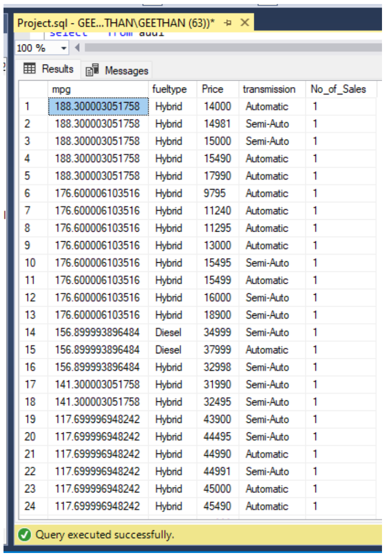


Project.sql - GEE...THAN\GEETHAN (63))*

```
(10612 rows affected)

Completion time: 2022-05-27T22:26:23.9455341+05:30
```

- ❖ For Audi:

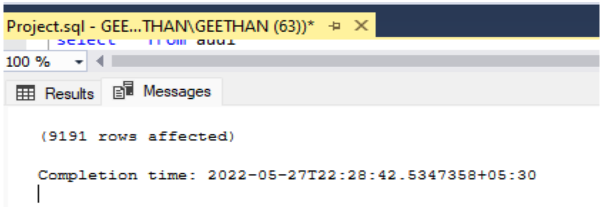
```
Select mpg,fueltype,Price,C.transmission,count(A.ID) as No_of_Sales from audi as A inner join
fueltype as B on A.fuel_ID
=B.fuel_ID inner join transmission as c on C.id=A.transmission_ID
Group by fueltype,price,mpg,C.transmission
order by mpg desc ;
```



Project.sql - GEE...THAN\GEETHAN (63))*

	mpg	fueltype	Price	transmission	No_of_Sales
1	188.300003051758	Hybrid	14000	Automatic	1
2	188.300003051758	Hybrid	14981	Semi-Auto	1
3	188.300003051758	Hybrid	15000	Semi-Auto	1
4	188.300003051758	Hybrid	15490	Automatic	1
5	188.300003051758	Hybrid	17990	Automatic	1
6	176.600006103516	Hybrid	9795	Automatic	1
7	176.600006103516	Hybrid	11240	Automatic	1
8	176.600006103516	Hybrid	11295	Automatic	1
9	176.600006103516	Hybrid	13000	Automatic	1
10	176.600006103516	Hybrid	15495	Semi-Auto	1
11	176.600006103516	Hybrid	15499	Automatic	1
12	176.600006103516	Hybrid	16000	Semi-Auto	1
13	176.600006103516	Hybrid	18900	Semi-Auto	1
14	156.899993896484	Diesel	34999	Semi-Auto	1
15	156.899993896484	Diesel	37999	Automatic	1
16	156.899993896484	Hybrid	32998	Semi-Auto	1
17	141.300003051758	Hybrid	31990	Semi-Auto	1
18	141.300003051758	Hybrid	32495	Semi-Auto	1
19	117.699996948242	Hybrid	43900	Semi-Auto	1
20	117.699996948242	Hybrid	44495	Semi-Auto	1
21	117.699996948242	Hybrid	44990	Automatic	1
22	117.699996948242	Hybrid	44991	Semi-Auto	1
23	117.699996948242	Hybrid	45000	Automatic	1
24	117.699996948242	Hybrid	45490	Automatic	1

Query executed successfully.

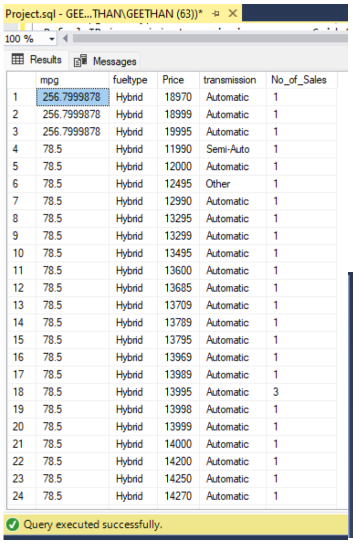


Project.sql - GEE...THAN\GEETHAN (63))*

```
(9191 rows affected)

Completion time: 2022-05-27T22:28:42.5347358+05:30
```

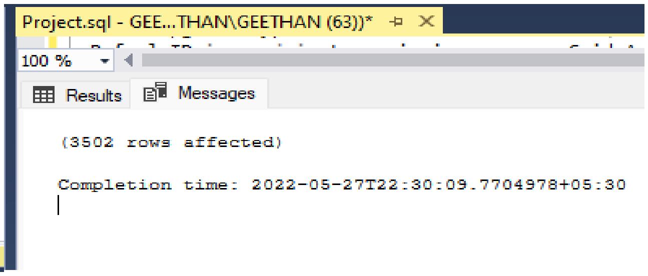
- ❖ For Hyndai:

```
Select mpg,fueltype,Price,C.transmission,count(A.ID) as No_of_Sales from hyndai as A inner join
fueltype as B on A.fuel_ID
=B.fuel_ID inner join transmission as c on C.id=A.transmission_ID
Group by fueltype,price,mpg,C.transmission
order by mpg desc ;
```



	mpg	fueltype	Price	transmission	No_of_Sales
1	256.7999878	Hybrid	18970	Automatic	1
2	256.7999878	Hybrid	18999	Automatic	1
3	256.7999878	Hybrid	19995	Automatic	1
4	78.5	Hybrid	11990	Semi-Auto	1
5	78.5	Hybrid	12000	Automatic	1
6	78.5	Hybrid	12495	Other	1
7	78.5	Hybrid	12990	Automatic	1
8	78.5	Hybrid	13295	Automatic	1
9	78.5	Hybrid	13299	Automatic	1
10	78.5	Hybrid	13495	Automatic	1
11	78.5	Hybrid	13600	Automatic	1
12	78.5	Hybrid	13685	Automatic	1
13	78.5	Hybrid	13709	Automatic	1
14	78.5	Hybrid	13789	Automatic	1
15	78.5	Hybrid	13795	Automatic	1
16	78.5	Hybrid	13969	Automatic	1
17	78.5	Hybrid	13989	Automatic	1
18	78.5	Hybrid	13995	Automatic	3
19	78.5	Hybrid	13998	Automatic	1
20	78.5	Hybrid	13999	Automatic	1
21	78.5	Hybrid	14000	Automatic	1
22	78.5	Hybrid	14200	Automatic	1
23	78.5	Hybrid	14250	Automatic	1
24	78.5	Hybrid	14270	Automatic	1

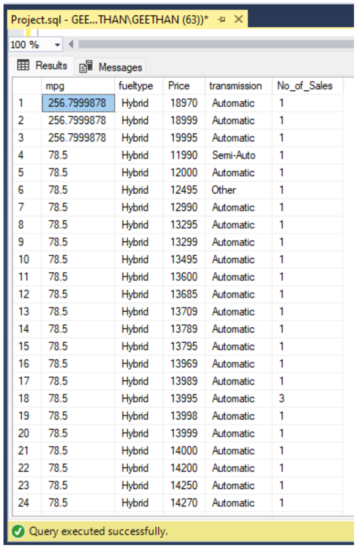
Query executed successfully.



```
(3502 rows affected)

Completion time: 2022-05-27T22:30:09.7704978+05:30
```

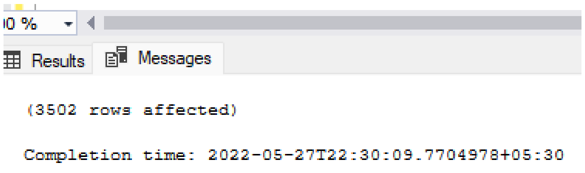
- For bmw:

Select mpg,fueltype,Price,C.transmission,count(A.ID) as No_of_Sales from bmw as A inner join fueltype as B on A.fuel_ID
=B.fuel_ID inner join transmission as c on C.id=A.transmission_ID
Group by fueltype,price,mpg,C.transmission
order by mpg desc ;



	mpg	fueltype	Price	transmission	No_of_Sales
1	256.7999878	Hybrid	18970	Automatic	1
2	256.7999878	Hybrid	18999	Automatic	1
3	256.7999878	Hybrid	19995	Automatic	1
4	78.5	Hybrid	11990	Semi-Auto	1
5	78.5	Hybrid	12000	Automatic	1
6	78.5	Hybrid	12495	Other	1
7	78.5	Hybrid	12990	Automatic	1
8	78.5	Hybrid	13295	Automatic	1
9	78.5	Hybrid	13299	Automatic	1
10	78.5	Hybrid	13495	Automatic	1
11	78.5	Hybrid	13600	Automatic	1
12	78.5	Hybrid	13685	Automatic	1
13	78.5	Hybrid	13709	Automatic	1
14	78.5	Hybrid	13789	Automatic	1
15	78.5	Hybrid	13795	Automatic	1
16	78.5	Hybrid	13969	Automatic	1
17	78.5	Hybrid	13989	Automatic	1
18	78.5	Hybrid	13995	Automatic	3
19	78.5	Hybrid	13998	Automatic	1
20	78.5	Hybrid	13999	Automatic	1
21	78.5	Hybrid	14000	Automatic	1
22	78.5	Hybrid	14200	Automatic	1
23	78.5	Hybrid	14250	Automatic	1
24	78.5	Hybrid	14270	Automatic	1

Query executed successfully.



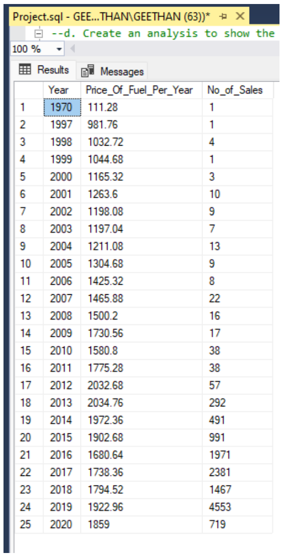
```
(3502 rows affected)

Completion time: 2022-05-27T22:30:09.7704978+05:30
```

d. Create an analysis to show the effect of fuel expenditure on the sales of car over the years(Get the fuel prices in UK through the years through internet sources)

- For Merc:

Select A.Year,C.Fuel_Price_Per_Year as Price_Of_Fuel_Per_Year,count(A.ID) as No_of_Sales from merc as A inner join fueltype as B on A.fuel_ID
=B.fuel_ID inner join fuelprice as c on A.year=c.year
group by A.year,C.Fuel_Price_Per_Year

```
order by A.year;
```



Project.sql - GEE...THAN\GEETHAN (63))*

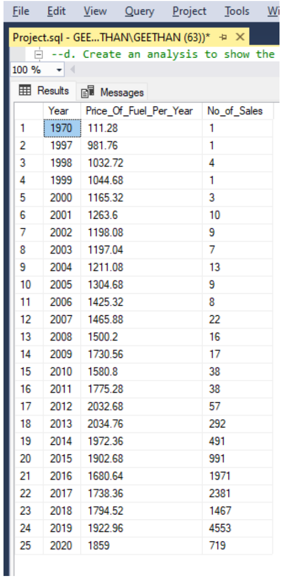
```
--d. Create an analysis to show the
```

100 %

	Year	Price_Of_Fuel_Per_Year	No_of_Sales
1	1970	111.28	1
2	1997	981.76	1
3	1998	1032.72	4
4	1999	1044.68	1
5	2000	1165.32	3
6	2001	1263.6	10
7	2002	1198.08	9
8	2003	1197.04	7
9	2004	1211.08	13
10	2005	1304.68	9
11	2006	1425.32	8
12	2007	1465.88	22
13	2008	1500.2	16
14	2009	1730.56	17
15	2010	1580.8	38
16	2011	1775.28	38
17	2012	2032.68	57
18	2013	2034.76	292
19	2014	1972.36	491
20	2015	1902.68	991
21	2016	1680.64	1971
22	2017	1738.36	2381
23	2018	1794.52	1467
24	2019	1922.96	4553
25	2020	1859	719

- For audi:

```
Select A.Year,C.Fuel_Price_Per_Year as Price_Of_Fuel_Per_Year,count(A.ID) as No_of_Sales
from audi as A inner join fueltype as B on A.fuel_ID
=B.fuel_ID inner join fuelprice as c on A.year=c.year
group by A.year,C.Fuel_Price_Per_Year
order by A.year;
```



Project.sql - GEE...THAN\GEETHAN (63))*

```
--d. Create an analysis to show the
```

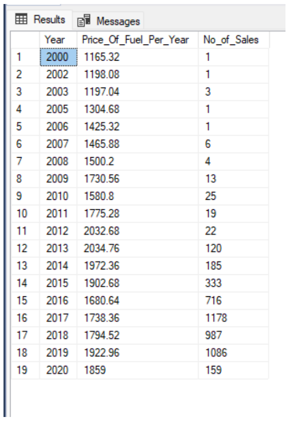
100 %

	Year	Price_Of_Fuel_Per_Year	No_of_Sales
1	1970	111.28	1
2	1997	981.76	1
3	1998	1032.72	4
4	1999	1044.68	1
5	2000	1165.32	3
6	2001	1263.6	10
7	2002	1198.08	9
8	2003	1197.04	7
9	2004	1211.08	13
10	2005	1304.68	9
11	2006	1425.32	8
12	2007	1465.88	22
13	2008	1500.2	16
14	2009	1730.56	17
15	2010	1580.8	38
16	2011	1775.28	38
17	2012	2032.68	57
18	2013	2034.76	292
19	2014	1972.36	491
20	2015	1902.68	991
21	2016	1680.64	1971
22	2017	1738.36	2381
23	2018	1794.52	1467
24	2019	1922.96	4553
25	2020	1859	719

- For Hyndai:

```
 Select A.Year,C.Fuel_Price_Per_Year as Price_Of_Fuel_Per_Year,count(A.ID) as No_of_Sales
from hyndai as A inner join fueltype as B on A.fuel_ID
```

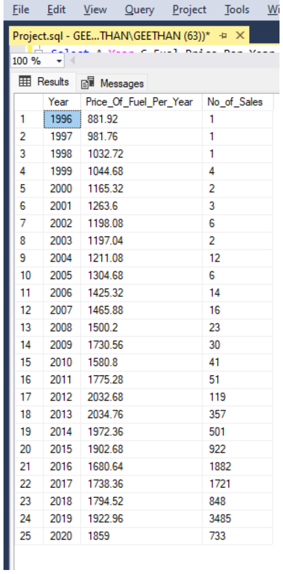
```
=B.fuel_ID inner join fuelprice as c on A.year=c.year
group by A.year,C.Fuel_Price_Per_Year
order by A.year;
```



	Year	Price_Of_Fuel_Per_Year	No_of_Sales
1	2000	1165.32	1
2	2002	1198.08	1
3	2003	1197.04	3
4	2005	1304.68	1
5	2006	1425.32	1
6	2007	1465.88	6
7	2008	1500.2	4
8	2009	1730.56	13
9	2010	1580.8	25
10	2011	1775.28	19
11	2012	2032.68	22
12	2013	2034.76	120
13	2014	1972.36	185
14	2015	1902.68	333
15	2016	1680.64	716
16	2017	1738.36	1178
17	2018	1794.52	987
18	2019	1922.96	1086
19	2020	1859	159

- For Bmw:

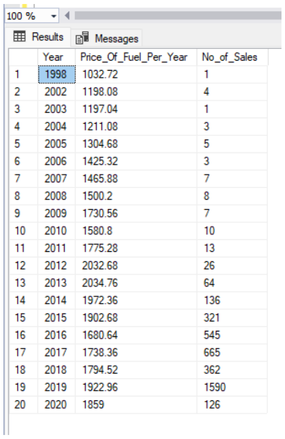
```
Select A.Year,C.Fuel_Price_Per_Year as Price_Of_Fuel_Per_Year,count(A.ID) as No_of_Sales
from bmw as A inner join fueltype as B on A.fuel_ID
=B.fuel_ID inner join fuelprice as c on A.year=c.year
group by A.year,C.Fuel_Price_Per_Year
order by A.year;
```



	Year	Price_Of_Fuel_Per_Year	No_of_Sales
1	1996	881.92	1
2	1997	981.76	1
3	1998	1032.72	1
4	1999	1044.68	4
5	2000	1165.32	2
6	2001	1263.6	3
7	2002	1198.08	6
8	2003	1197.04	2
9	2004	1211.08	12
10	2005	1304.68	6
11	2006	1425.32	14
12	2007	1465.88	16
13	2008	1500.2	23
14	2009	1730.56	30
15	2010	1580.8	41
16	2011	1775.28	51
17	2012	2032.68	119
18	2013	2034.76	357
19	2014	1972.36	501
20	2015	1902.68	922
21	2016	1680.64	1882
22	2017	1738.36	1721
23	2018	1794.52	848
24	2019	1922.96	3485
25	2020	1859	733

- For CClass:

```
Select A.Year,C.Fuel_Price_Per_Year as Price_Of_Fuel_Per_Year,count(A.ID) as No_of_Sales
from cclass as A inner join fueltype as B on A.fuel_ID
=B.fuel_ID inner join fuelprice as c on A.year=c.year
group by A.year,C.Fuel_Price_Per_Year
order by A.year
```

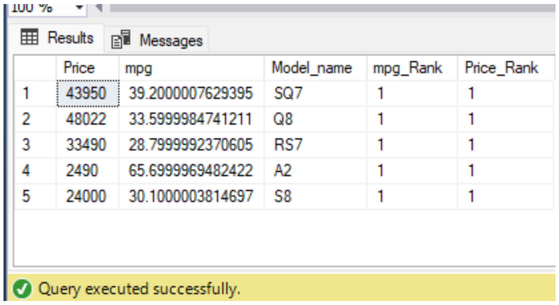


	Year	Price_Of_Fuel_Per_Year	No_of_Sales
1	1998	1032.72	1
2	2002	1198.08	4
3	2003	1197.04	1
4	2004	1211.08	3
5	2005	1304.68	5
6	2006	1425.32	3
7	2007	1465.88	7
8	2008	1500.2	8
9	2009	1730.56	7
10	2010	1580.8	10
11	2011	1775.28	13
12	2012	2032.68	26
13	2013	2034.76	64
14	2014	1972.36	136
15	2015	1902.68	321
16	2016	1680.64	545
17	2017	1738.36	665
18	2018	1794.52	362
19	2019	1922.96	1590
20	2020	1859	126

Using all of the above analysis, suggest cost and usage effective car type for the brand to launch(We can launch multiple types of car as well)

- For Audi:

```
Select * from(
select A.Price,A.mpg,B.Model_name,Dense_Rank() over(partition by B.Model_name order by A.mpg desc) as mpg_Rank,
Dense_Rank() over(partition by B.Model_name order by A.Price) as Price_Rank
from audi as a join models as b on a.model_ID=b.model_ID)A where Price_rank=1 and mpg_rank=1;
```

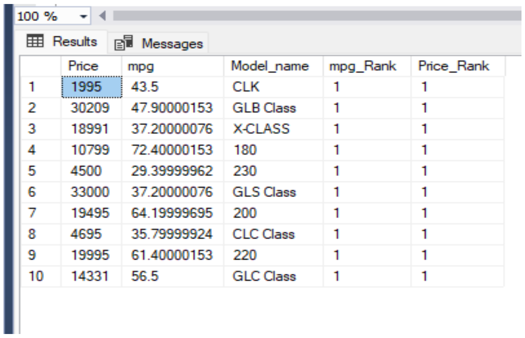


	Price	mpg	Model_name	mpg_Rank	Price_Rank
1	43950	39.2000007629395	SQ7	1	1
2	48022	33.5999984741211	Q8	1	1
3	33490	28.7999992370605	RS7	1	1
4	2490	65.6999969482422	A2	1	1
5	24000	30.1000003814697	S8	1	1

Query executed successfully.

- For merc:

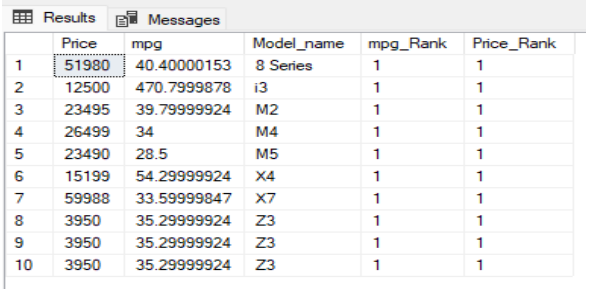
```
Select * from(
select A.Price,A.mpg,B.Model_name,Dense_Rank() over(partition by B.Model_name order by A.mpg desc) as mpg_Rank,
Dense_Rank() over(partition by B.Model_name order by A.Price) as Price_Rank
from merc as a join models as b on a.model_ID=b.model_ID)A where Price_rank=1 and mpg_rank=1;
```



	Price	mpg	Model_name	mpg_Rank	Price_Rank
1	1995	43.5	CLK	1	1
2	30209	47.90000153	GLB Class	1	1
3	18991	37.20000076	X-CLASS	1	1
4	10799	72.40000153	180	1	1
5	4500	29.39999962	230	1	1
6	33000	37.20000076	GLS Class	1	1
7	19495	64.19999695	200	1	1
8	4695	35.79999924	CLC Class	1	1
9	19995	61.40000153	220	1	1
10	14331	56.5	GLC Class	1	1

- For bmw:

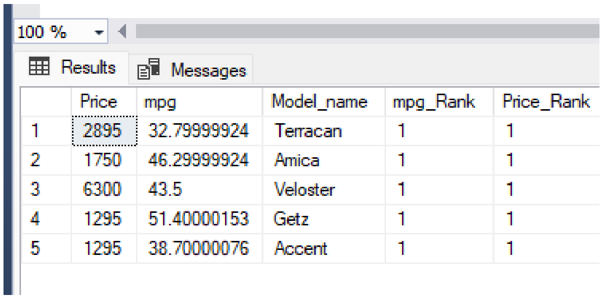
```
Select * from(
select A.Price,A.mpg,B.Model_name,Dense_Rank() over(partition by B.Model_name order by
A.mpg desc) as mpg_Rank,
Dense_Rank() over(partition by B.Model_name order by A.Price) as Price_Rank
from bmw as a join models as b on a.model_ID=b.model_ID)A where Price_rank=1 and
mpg_rank=1;
```



	Price	mpg	Model_name	mpg_Rank	Price_Rank
1	51980	40.40000153	8 Series	1	1
2	12500	470.7999878	i3	1	1
3	23495	39.79999924	M2	1	1
4	26499	34	M4	1	1
5	23490	28.5	M5	1	1
6	15199	54.29999924	X4	1	1
7	59988	33.59999847	X7	1	1
8	3950	35.29999924	Z3	1	1
9	3950	35.29999924	Z3	1	1
10	3950	35.29999924	Z3	1	1

- For Hyndai:

```
Select * from(
select A.Price,A.mpg,B.Model_name,Dense_Rank() over(partition by B.Model_name order by
A.mpg desc) as mpg_Rank,
Dense_Rank() over(partition by B.Model_name order by A.Price) as Price_Rank
from hyndai as a join models as b on a.model_ID=b.model_ID)A where Price_rank=1 and
mpg_rank=1;
```

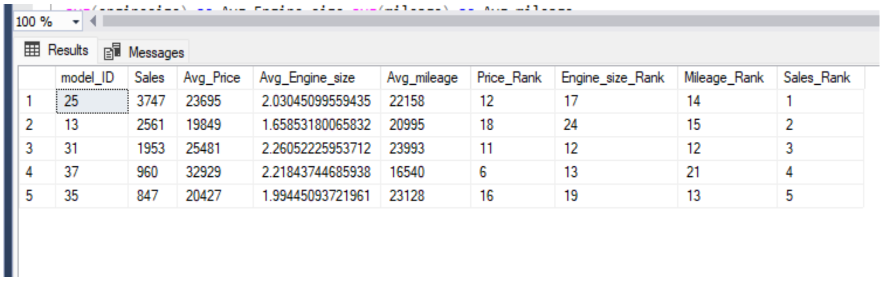


	Price	mpg	Model_name	mpg_Rank	Price_Rank
1	2895	32.79999924	Terracan	1	1
2	1750	46.29999924	Amica	1	1
3	6300	43.5	Veloster	1	1
4	1295	51.40000153	Getz	1	1
5	1295	38.70000076	Accent	1	1

You are also asked to rank across all the models based on their total sales, average price, average mileage, average engine size, etc. and now filter the top 5 basis their sales. Observe the identified models and provide your inference.

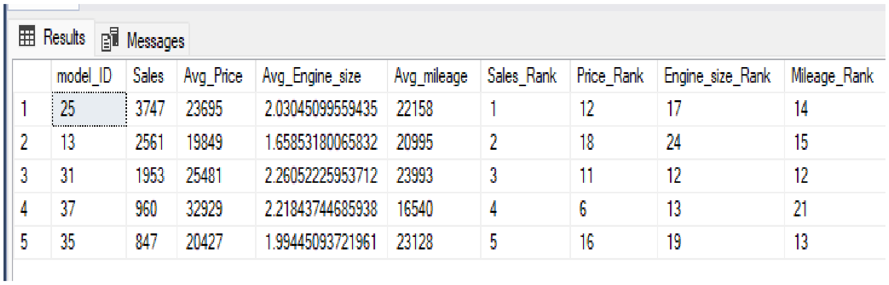
- For Merc:

```
select top 5 * from(
 select *,Dense_rank() over(order by Avg_Price desc)as Price_Rank,
 Dense_rank() over(order by Avg_Engine_size desc)as Engine_size_Rank,
 Dense_rank() over(order by Avg_mileage desc)as Mileage_Rank ,
 Dense_rank() over(order by Sales desc)as Sales_Rank from(
 Select model_ID,count(ID) as Sales ,avg(price) as Avg_Price,
 avg(enginesize) as Avg_Engine_size,avg(mileage) as Avg_mileage
 from merc group by model_ID) merc ) merc;
```



	model_ID	Sales	Avg_Price	Avg_Engine_size	Avg_mileage	Price_Rank	Engine_size_Rank	Mileage_Rank	Sales_Rank
1	25	3747	23695	2.03045099559435	22158	12	17	14	1
2	13	2561	19849	1.65853180065832	20995	18	24	15	2
3	31	1953	25481	2.26052225953712	23993	11	12	12	3
4	37	960	32929	2.21843744685938	16540	6	13	21	4
5	35	847	20427	1.99445093721961	23128	16	19	13	5

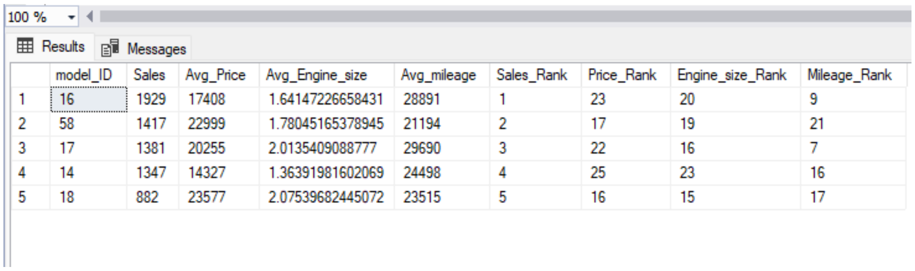
```
 --or--
 select * from(select*,Dense_rank() over(order by Sales desc)as Sales_Rank ,Dense_rank()
over(order by Avg_Price desc)as Price_Rank,
 Dense_rank() over(order by Avg_Engine_size desc)as Engine_size_Rank,
 Dense_rank() over(order by Avg_mileage desc)as Mileage_Rank
 from(
 Select model_ID,count(ID) as Sales ,avg(price) as Avg_Price,
 avg(enginesize) as Avg_Engine_size,avg(mileage) as Avg_mileage
 from merc group by model_ID)merc)merc where sales_rank<=5 order by Sales_Rank;
```



	model_ID	Sales	Avg_Price	Avg_Engine_size	Avg_mileage	Sales_Rank	Price_Rank	Engine_size_Rank	Mileage_Rank
1	25	3747	23695	2.03045099559435	22158	1	12	17	14
2	13	2561	19849	1.65853180065832	20995	2	18	24	15
3	31	1953	25481	2.26052225953712	23993	3	11	12	12
4	37	960	32929	2.21843744685938	16540	4	6	13	21
5	35	847	20427	1.99445093721961	23128	5	16	19	13

- For Audi:

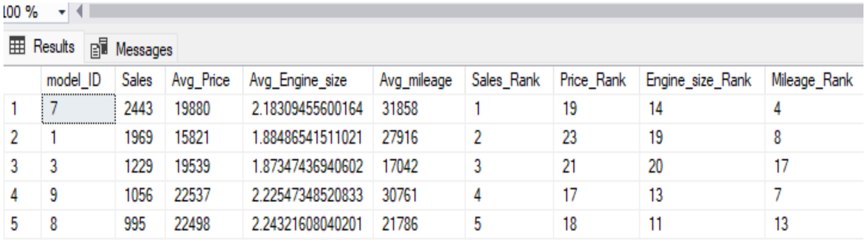
```
 select * from(select*,Dense_rank() over(order by Sales desc)as Sales_Rank ,Dense_rank()
over(order by Avg_Price desc)as Price_Rank,
 Dense_rank() over(order by Avg_Engine_size desc)as Engine_size_Rank,
 Dense_rank() over(order by Avg_mileage desc)as Mileage_Rank
 from(
 Select model_ID,count(ID) as Sales ,avg(price) as Avg_Price,
 avg(enginesize) as Avg_Engine_size,avg(mileage) as Avg_mileage
 from audi group by model_ID)audi)audi where sales_rank<=5 order by Sales_Rank;
```



	model_ID	Sales	Avg_Price	Avg_Engine_size	Avg_mileage	Sales_Rank	Price_Rank	Engine_size_Rank	Mileage_Rank
1	16	1929	17408	1.64147226658431	28891	1	23	20	9
2	58	1417	22999	1.78045165378945	21194	2	17	19	21
3	17	1381	20255	2.0135409088777	29690	3	22	16	7
4	14	1347	14327	1.36391981602069	24498	4	25	23	16
5	18	882	23577	2.07539682445072	23515	5	16	15	17

- ❖ For Bwm:

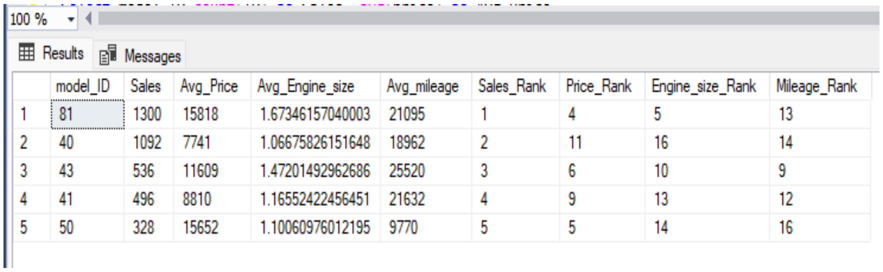
```
 select * from(select*,Dense_rank() over(order by Sales desc)as Sales_Rank ,Dense_rank()
over(order by Avg_Price desc)as Price_Rank,
 Dense_rank() over(order by Avg_Engine_size desc)as Engine_size_Rank,
 Dense_rank() over(order by Avg_mileage desc)as Mileage_Rank
 from(
 Select model_ID,count(ID) as Sales ,avg(price) as Avg_Price,
 avg(enginesize) as Avg_Engine_size,avg(mileage) as Avg_mileage
 from bmw group by model_ID)bmw)bmw where sales_rank<=5 order by Sales_Rank;
```



	model_ID	Sales	Avg_Price	Avg_Engine_size	Avg_mileage	Sales_Rank	Price_Rank	Engine_size_Rank	Mileage_Rank
1	7	2443	19880	2.18309455600164	31858	1	19	14	4
2	1	1969	15821	1.88486541511021	27916	2	23	19	8
3	3	1229	19539	1.87347436940602	17042	3	21	20	17
4	9	1056	22537	2.22547348520833	30761	4	17	13	7
5	8	995	22498	2.24321608040201	21786	5	18	11	13

For Hyndai:

```
 select * from(select*,Dense_rank() over(order by Sales desc)as Sales_Rank ,Dense_rank()
over(order by Avg_Price desc)as Price_Rank,
 Dense_rank() over(order by Avg_Engine_size desc)as Engine_size_Rank,
 Dense_rank() over(order by Avg_mileage desc)as Mileage_Rank
 from(
 Select model_ID,count(ID) as Sales ,avg(price) as Avg_Price,
 avg(enginesize) as Avg_Engine_size,avg(mileage) as Avg_mileage
 from hyndai group by model_ID)hyndai)hyndai where sales_rank<=5 order by Sales_Rank;
```



	model_ID	Sales	Avg_Price	Avg_Engine_size	Avg_mileage	Sales_Rank	Price_Rank	Engine_size_Rank	Mileage_Rank
1	81	1300	15818	1.67346157040003	21095	1	4	5	13
2	40	1092	7741	1.06675826151648	18962	2	11	16	14
3	43	536	11609	1.47201492962686	25520	3	6	10	9
4	41	496	8810	1.16552422456451	21632	4	9	13	12
5	50	328	15652	1.10060976012195	9770	5	5	14	16