PROJECT

CAR LAUNCH ANALYSIS

PROJECT QUERY

```
--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 A.model_id,c.model_name,A.price,B.per_capita_income ,B.year,
concat(abs((A.price)*100/per_capita_income),'%') as perc from audi as A
inner join per_capita as B on A.year=B.year
inner join models as c on a.model_ID = c.model_id

create view brands as
select A.model_id,c.model_name,A.price,B.per_capita_income ,B.year,
concat(abs((A.price)*100/per_capita_income),'%') as perc from audi as A
inner join per_capita as B on A.year=B.year
inner join models as c on a.model_ID = c.model_id

select * from brands
-----------------------------------------------
select *, case when perc>='130%' then 'rich'
when perc >='70%' then 'middle_class'
when perc >='0%' then 'poor'
end as class_of_customer
from brands
order by year desc
```

```
-----------------------------------------------
select *, case when perc>='130%' then 'rich'
when perc >='70%' then 'middle_class'
when perc >='0%' then 'poor'
end as class_of_customer
from brands
order by year desc
-----------------------------------------------------------------------------------------------
```

108 %

Results | Messages

	model_id	model_name	price	per_capita_income	year	perc	class_of_customer
1	60	Q7	62985	30504	2020	206%	rich
2	59	Q5	47895	30504	2020	157%	rich
3	17	A4	31985	30504	2020	104%	poor
4	59	Q5	37985	30504	2020	124%	poor
5	60	Q7	82995	30504	2020	272%	rich
6	16	A3	35995	30504	2020	118%	poor
7	17	A4	34798	30504	2020	114%	poor
8	17	A4	40780	30504	2020	133%	rich
9	17	A4	31985	30504	2020	104%	poor
10	18	A5	37331	30504	2020	122%	poor
11	18	A5	39454	30504	2020	129%	poor
12	18	A5	37507	30504	2020	122%	poor
13	58	Q3	33462	30504	2020	109%	poor

Query executed successfully

```
-----------------------------------------------------------------------------------------
--------------------------------
```

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

```
create VIEW pricee as
SELECT id, year, model_id ,model_name, price, car_name FROM cars WHERE year > 2017
create VIEW priceel as
SELECT avg(price) as avg_price, year, model_name, car_name from pricee GROUP BY
model_name, year,car_name

SELECT car_name, model_name, [2018],[2019],[2020],
([2019]-[2018]) as jump_1, ([2020]-[2019]) as jump_2
FROM priceel
pivot (avg(avg_price) for year in ([2018], [2019], [2020])) as a

select * from priceel
```

```
SELECT car_name, model_name, [2018],[2019],[2020],
([2019]-[2018]) as jump_1, ([2020]-[2019]) as jump_2
FROM priceel
pivot (avg(avg_price) for year in ([2018], [2019], [2020])) as a

select * from priceel
```

108 %

Results | Messages

	car_name	model_name	2018	2019	2020	jump_1	jump_2
1	Audi	A1	15469	21200	22781	5731	1581
2	Audi	A3	20557	23551	26276	2994	2725
3	Audi	A4	21412	27663	35607	6251	7944
4	Audi	A5	25175	29437	31704	4262	2267
5	Audi	A6	25446	33582	39498	8136	5916
6	Audi	A7	34375	39828	44974	5453	5146
7	Audi	A8	37575	43315	55627	5740	12312
8	Audi	Q2	20037	24135	27119	4098	2984
9	Audi	Q3	23265	29707	32982	6442	3275
10	Audi	Q5	32474	35999	43268	3525	7269
11	Audi	Q7	50751	49370	63820	-1381	14450
12	Audi	Q8	55712	59022	69206	3310	10184
13	Audi	R8	94963	122742	138965	27779	16223

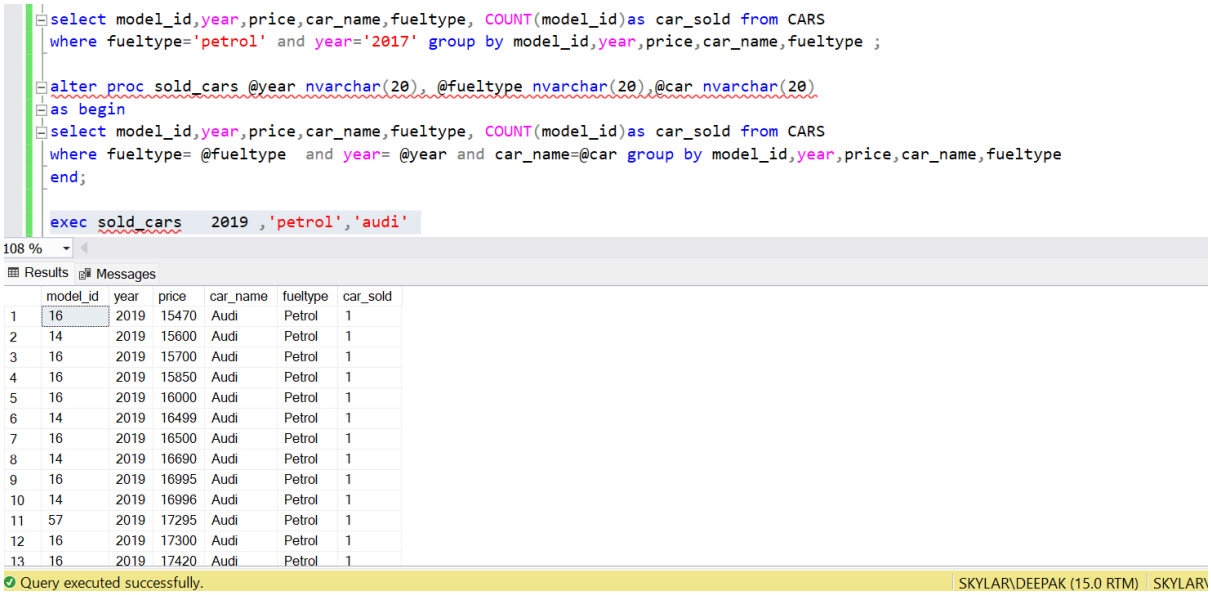
```
-------------------------
```

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

```
select model_id,year,price,car_name,fueltype, COUNT(model_id)as car_sold from CARS
where fueltype='petrol' and year='2017' group by model_id,year,price,car_name,fueltype
;

alter proc sold_cars @year nvarchar(20), @fueltype nvarchar(20),@car nvarchar(20)
as begin
select model_id,year,price,car_name,fueltype, COUNT(model_id)as car_sold from CARS
where fueltype= @fueltype  and year= @year and car_name=@car group by
model_id,year,price,car_name,fueltype
end;

exec sold_cars   2019 ,'petrol','audi'
select * from cars order by id
```



	model_id	year	price	car_name	fueltype	car_sold
1	16	2019	15470	Audi	Petrol	1
2	14	2019	15600	Audi	Petrol	1
3	16	2019	15700	Audi	Petrol	1
4	16	2019	15850	Audi	Petrol	1
5	16	2019	16000	Audi	Petrol	1
6	14	2019	16499	Audi	Petrol	1
7	16	2019	16500	Audi	Petrol	1
8	14	2019	16690	Audi	Petrol	1
9	16	2019	16995	Audi	Petrol	1
10	14	2019	16996	Audi	Petrol	1
11	57	2019	17295	Audi	Petrol	1
12	16	2019	17300	Audi	Petrol	1
13	16	2019	17420	Audi	Petrol	1

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

```sql
select * into recent from cars
where year >'2016';

select * from recent;

select * from cars;
select max(mileage) from recent
select min(mileage) from recent
select max(price) from recent
select min(Price) from recent


select count(id)as cars_sold_petrol from recent
where car_name = 'bmw' and fueltype = 'petrol';

select count(id)cars_sold_diesel from recent
where car_name = 'bmw' and fueltype = 'diesel';
```

```sql
select count(id)as cars_sold_petrol from recent
where car_name = 'bmw' and fueltype = 'petrol';

select count(id)cars_sold_diesel from recent
where car_name = 'bmw' and fueltype = 'diesel';

select *,count(id) over(partition by fueltype order by mileage desc) from recent
```

108 %

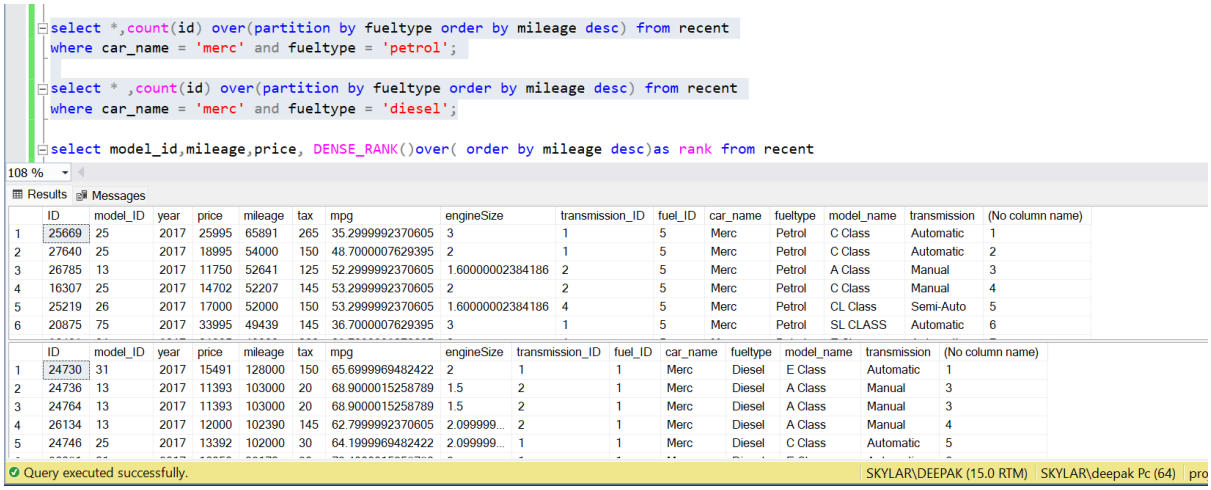
Results | Messages

	cars_sold_petrol
1	2587

	cars_sold_diesel
1	3968

```sql
select *,count(id) over(partition by fueltype order by mileage desc) from recent
where car_name = 'merc' and fueltype = 'petrol';

select * ,count(id) over(partition by fueltype order by mileage desc) from recent
where car_name = 'merc' and fueltype = 'diesel';
```



```sql
select *,count(id) over(partition by fueltype order by mileage desc) from recent
where car_name = 'merc' and fueltype = 'petrol';

select * ,count(id) over(partition by fueltype order by mileage desc) from recent
where car_name = 'merc' and fueltype = 'diesel';

select model_id,mileage,price, DENSE_RANK()over( order by mileage desc)as rank from recent
```

108 %

Results | Messages

	ID	model_ID	year	price	mileage	tax	mpg	engineSize	transmission_ID	fuel_ID	car_name	fueltype	model_name	transmission	(No column name)
1	25669	25	2017	25995	65891	265	35.2999992370605	3	1	5	Merc	Petrol	C Class	Automatic	1
2	27640	25	2017	18995	54000	150	48.7000007629395	2	1	5	Merc	Petrol	C Class	Automatic	2
3	26785	13	2017	11750	52641	125	52.2999992370605	1.60000002384186	2	5	Merc	Petrol	A Class	Manual	3
4	16307	25	2017	14702	52207	145	53.2999992370605	2	2	5	Merc	Petrol	C Class	Manual	4
5	25219	26	2017	17000	52000	150	53.2999992370605	1.60000002384186	4	5	Merc	Petrol	CL Class	Semi-Auto	5
6	20875	75	2017	33995	49439	145	36.7000007629395	3	1	5	Merc	Petrol	SL CLASS	Automatic	6

	ID	model_ID	year	price	mileage	tax	mpg	engineSize	transmission_ID	fuel_ID	car_name	fueltype	model_name	transmission	(No column name)
1	24730	31	2017	15491	128000	150	65.6999969482422	2	1	1	Merc	Diesel	E Class	Automatic	1
2	24736	13	2017	11393	103000	20	68.9000015258789	1.5	2	1	Merc	Diesel	A Class	Manual	3
3	24764	13	2017	11393	103000	20	68.9000015258789	1.5	2	1	Merc	Diesel	A Class	Manual	3
4	26134	13	2017	12000	102390	145	62.7999992370605	2.099999...	2	1	Merc	Diesel	A Class	Manual	4
5	24746	25	2017	13392	102000	30	64.1999969482422	2.099999...	1	1	Merc	Diesel	C Class	Automatic	5

Query executed successfully. SKYLAR\DEEPAK (15.0 RTM) SKYLAR\deepak Pc (64) pro

```sql
where car_name = 'merc';
--------------------------------------------------------------------------------
------------
```

```
--D 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.
```

```sql
----avg_price
select proc avg_price @car_name varchar(50) ,@year nvarchar(20)
as begin
select top 5 year, model_name, car_name, avg(price) as avg_price ,
DENSE_RANK() over (order by avg(price)desc)as rank from cars
where year= @year and car_name= @car_name group by year, model_name, car_name
end;
exec avg_price merc,2019;
```

```sql
----avg_price
select proc avg_price @car_name varchar(50) ,@year nvarchar(20)
as begin
select top 5 year, model_name, car_name, avg(price) as avg_price ,
DENSE_RANK() over (order by avg(price)desc)as rank from cars
where year= @year and car_name= @car_name group by year, model_name, car_name
end;
exec avg_price merc,2019;
--------------------------------
```

108 %

Results | Messages

	year	model_name	car_name	avg_price	rank
1	2019	G Class	Merc	121225	1
2	2019	S Class	Merc	61611	2
3	2019	GLS Class	Merc	52499	3
4	2019	GLE Class	Merc	51873	4
5	2019	SL CLASS	Merc	43683	5

```sql
--------------------------------
-----avg_mileage
select proc avg_mileage @car_name varchar(50) ,@year nvarchar(20)
as begin
select top 5 year, model_name, car_name, avg(mileage) as avg_mileage ,
DENSE_RANK() over (order by avg(mileage)desc)as rank from cars
where year= @year and car_name= @car_name group by year, model_name, car_name
end;
exec avg_mileage merc,2019;
```

```sql
-----avg_mileage
select proc avg_mileage @car_name varchar(50) ,@year nvarchar(20)
as begin
select top 5 year, model_name, car_name, avg(mileage) as avg_mileage ,
DENSE_RANK() over (order by avg(mileage)desc)as rank from cars
where year= @year and car_name= @car_name group by year, model_name, car_name
end;
exec avg_mileage merc,2019;
-----------------------------------
```

08 %

Results | Messages

	year	model_name	car_name	avg_mileage	rank
1	2019	G Class	Merc	10345	1
2	2019	GLS Class	Merc	10088	2
3	2019	GLA Class	Merc	8225	3
4	2019	CLA Class	Merc	7895	4
5	2019	E Class	Merc	7757	5

```sql
-----------------------------------
```

```sql
----avg_enginesize
create proc engine @car_name varchar(50) ,@year nvarchar(20)
as begin
select top 5 year, model_name, car_name, avg(enginesize) as avg_enginesize ,
DENSE_RANK() over (order by avg(enginesize)desc)as rank from cars
where year= @year and car_name= @car_name group by year, model_name, car_name
end;
exec engine merc,2019;
```

```sql
----avg_enginesize
create proc engine @car_name varchar(50) ,@year nvarchar(20)
as begin
select top 5 year, model_name, car_name, avg(enginesize) as avg_enginesize ,
DENSE_RANK() over (order by avg(enginesize)desc)as rank from cars
where year= @year and car_name= @car_name group by year, model_name, car_name
end;
exec engine merc,2019;
--------------------------------
```

108 %

Results | Messages

	year	model_name	car_name	avg_enginesize	rank
1	2019	G Class	Merc	3.5	1
2	2019	S Class	Merc	3.16428572109767	2
3	2019	GLS Class	Merc	3	3
4	2019	SL CLASS	Merc	2.89782606648362	4
5	2019	CLS Class	Merc	2.75806459303825	5

```sql
--------------------------------
---count_sales
create proc count_sales @car_name varchar(50) ,@year nvarchar(20)
as begin
select top 5 year, model_name, car_name, count(model_name) as count_sales ,
DENSE_RANK() over (order by count(model_name)desc)as rank from cars
where year= @year and car_name= @car_name group by year, model_name, car_name
end;
exec count_sales merc,2019;
```

```sql
---count_sales
create proc count_sales @car_name varchar(50) ,@year nvarchar(20)
as begin
select top 5 year, model_name, car_name, count(model_name) as count_sales ,
DENSE_RANK() over (order by count(model_name)desc)as rank from cars
where year= @year and car_name= @car_name group by year, model_name, car_name
end;
exec count_sales merc,2019;
```

108 %

Results | Messages

	year	model_name	car_name	count_sales	rank
1	2019	C Class	Merc	1551	1
2	2019	A Class	Merc	745	2
3	2019	E Class	Merc	687	3
4	2019	GLC Class	Merc	393	4
5	2019	B Class	Merc	296	5

ADDITIONAL QUERIES FOR MAKING UNITED TABLE CARS

```
select* from audi
select* from cclass
alter table cclass add  tax int  null
alter table cclass add  mpg float  null

alter table cclass add car_name nvarchar(20) not null default 'CClass'


create view cclass1 as
select
id,model_id,year,price,mileage,tax,mpg,enginesize,transmission_id,fuel_id,car_name
from cclass

create view brandtype as
select* from audi union
select* from bmw union
select* from hyndai union
select* from merc union
select* from cclass1
select * from fueltype
select* from models
select * from transmission
select * from cars



CREATE view cars as
select a.*,b.fueltype,c.model_name,d.transmission from brandtype as a left join
fueltype as b on a.fuel_ID= b.fuel_ID
left join models as c on a.model_ID= c.model_ID
left join transmission d on a.transmission_ID= d.ID

select * from cars order by id
```