

PROJECT SQL

PROJECT NAME – CAR DATASET ANALYSIS

---car dataset analysis project in sql

```
--in the cclass table add two column "tax"and"mpg" before 'engineSize'  
select *from cclass  
alter table cclass add tax int  
alter table cclass add mpg int  
  
select*from cclass
```

	ID	model_ID	year	price	mileage	tax	mpg	engineSize	transmission_ID	fuel_ID
1	10669	25	2020	30495	1200	NULL	NULL	2	1	1
2	10670	25	2020	29989	1000	NULL	NULL	1.5	1	5
3	10671	25	2020	37899	500	NULL	NULL	2	1	1
4	10672	25	2019	30399	5000	NULL	NULL	2	1	1
5	10673	25	2019	29899	4500	NULL	NULL	2	1	1
6	10674	25	2020	30999	1000	NULL	NULL	2	1	1
7	10675	25	2020	35999	500	NULL	NULL	2	1	1
8	10676	25	2019	37990	1412	NULL	NULL	3	1	5
9	10677	25	2019	28990	3569	NULL	NULL	2	1	1
10	10678	25	2019	28990	3635	NULL	NULL	2	1	1
11	10679	25	2013	9995	44900	NULL	NULL	1.60000002384186	1	5
12	10680	25	2012	6995	88200	NULL	NULL	2.099999990463257	1	1

PROJECT SQL

```
--combain all the given table with the help of union all
SELECT *FROM audi UNION ALL SELECT*FROM BMW
UNION ALL SELECT*FROM hyundai UNION ALL SELECT*FROM merc union all select *from cclass
order by id
```

Results Messages										
	ID	model_ID	year	price	mileage	tax	mpg	engineSize	transmission_ID	fuel_ID
1	1	14	2017	12500	15735	150	55.4000015258789	1.39999997615814	2	5
2	2	19	2016	16500	36203	20	64.1999969482422	2	1	1
3	3	14	2016	11000	29946	30	55.4000015258789	1.39999997615814	2	5
4	4	17	2017	16800	25952	145	67.3000030517578	2	1	1
5	5	16	2019	17300	1998	145	49.5999984741211	1	2	5
6	6	14	2016	13900	32260	30	58.9000015258789	1.39999997615814	1	5
7	7	19	2016	13250	76788	30	61.4000015258789	2	1	1
8	8	17	2016	11750	75185	20	70.5999984741211	2	2	1
9	9	16	2015	10200	46112	20	60.0999984741211	1.39999997615814	2	5
10	10	14	2016	12000	22451	30	55.4000015258789	1.39999997615814	2	5
11	11	16	2017	16100	28955	145	58.9000015258789	1.39999997615814	2	5
12	12	19	2016	16500	52198	125	57.5999984741211	2	1	1

```
--a. Create an analysis to find income class of UK citizens based on price of Cars
```

```
--IMPORT THE uk per cap income TABLE
```

```
select*,case when price>=50000 then 'upper_class'
when price<=50000 and price>12000 then 'middle_class'
when price<=12000 then 'lower_class' end as sitizen_class_uk from(
select a.price,a.year,b.per_cap_income_year from
(SELECT *FROM audi UNION ALL SELECT*FROM BMW
UNION ALL SELECT*FROM hyundai UNION ALL SELECT*FROM merc union all select *from cclass)
as a left join
(select year(date) as year,round(NYGDPPCAPKDGGBR,2)as per_cap_income_year from [uk per cap income]) as b
on a.year=b.year )c;
```

	price	year	per_cap_income_year	sitizen_class_uk
1	12500	2017	46746.64	middle_class
2	16500	2016	46081.71	middle_class
3	11000	2016	46081.71	lower_class
4	16800	2017	46746.64	middle_class
5	17300	2019	47750.88	middle_class
6	13900	2016	46081.71	middle_class
7	13250	2016	46081.71	middle_class
8	11750	2016	46081.71	lower_class
9	10200	2015	45404.57	lower_class
10	12000	2016	46081.71	lower_class

```
--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
```

PROJECT SQL

```
--#price across the years
select sum(price) as car_price,year from(
SELECT *FROM audi UNION ALL SELECT*FROM BMW
UNION ALL SELECT*FROM hyndai UNION ALL SELECT*FROM merc union all select *from cclass
)c
group by year order by year;
```

	car_price	year
1	24999	1970
2	4450	1991
3	4750	1995
4	5995	1996
5	18595	1997
6	44659	1998
7	19135	1999
8	21774	2000
9	71579	2001
10	99989	2002
11	82511	2003

```
--#price changes across the years
select*,CAR_PRICE-LAG(CAR_PRICE)OVER(ORDER BY CAR_PRICE)AS CHANGE_PRICE FROM
(select sum(price) as car_price,year from(
SELECT *FROM audi UNION ALL SELECT*FROM BMW
UNION ALL SELECT*FROM hyndai UNION ALL SELECT*FROM merc union all select *from cclass
)c
group by year)C;
```

	car_price	year	CHANGE_PRICE
1	4450	1991	NULL
2	4750	1995	300
3	5995	1996	1245
4	18595	1997	12600
5	19135	1999	540
6	21774	2000	2639
7	24999	1970	3225
8	44659	1998	19660
9	71579	2001	26920
10	82511	2003	10932
11	99989	2002	17478

--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.

PROJECT SQL

```
--# number of sales for each year
select count(id) as car_SOLD,year from(
SELECT *FROM audi UNION ALL SELECT*FROM BMW
UNION ALL SELECT*FROM hyundai UNION ALL SELECT*FROM merc union all select *from cclass
)c
group by year order by year;
```

	car_SOLD	year
1	1	1970
2	1	1991
3	1	1995
4	1	1996
5	3	1997
6	7	1998
7	5	1999
8	6	2000
9	13	2001
10	22	2002
11	19	2003

```
--# CHANGE number of sales for each year
SELECT *, CAR_SOLD-LAG(CAR_SOLD)OVER(ORDER BY CAR_SOLD) AS CHANGE_CAR_SOLD FROM(
select count(id) as car_SOLD,year from(
SELECT *FROM audi UNION ALL SELECT*FROM BMW
UNION ALL SELECT*FROM hyundai UNION ALL SELECT*FROM merc union all select *from cclass
)c
group by year )C;
```

	car_SOLD	year	CHANGE_CAR_SOLD
1	1	1970	NULL
2	1	1996	0
3	1	1991	0
4	1	1995	0
5	3	1997	2
6	5	1999	2
7	6	2000	1
8	7	1998	1
9	13	2001	6
10	19	2003	6
11	22	2002	3

```
--#identify the categories which has seen significant jump in its sales
SELECT TOP 10 CAR_SOLD,year,CHANGE_CAR_SOLD FROM(
SELECT *, LAG(CAR_SOLD) OVER(ORDER BY CAR_SOLD DESC )-car_SOLD AS CHANGE_CAR_SOLD FROM(
select count(id) as car_SOLD,year from(
SELECT *FROM audi UNION ALL SELECT*FROM BMW
UNION ALL SELECT*FROM hyundai UNION ALL SELECT*FROM merc union all select *from cclass
)c
group by year )C)C ;
```

PROJECT SQL

	CAR_SOLD	YEAR	CHANGE_CAR_SOLD
1	13911	2019	NULL
2	7880	2017	6031
3	7078	2016	802
4	4528	2018	2550
5	3566	2015	962
6	2453	2020	1113
7	1760	2014	693
8	1124	2013	636
9	309	2012	815
10	168	2011	141

--# number of sales for each car models in year

```
select a.*,b.* from(
SELECT *FROM audi UNION ALL SELECT*FROM BMW
UNION ALL SELECT*FROM hyundai UNION ALL SELECT*FROM merc union all select *from cclass
) as a inner join models as b on a.model_id=b.model_id;
```

	ID	model_ID	year	price	mileage	tax	mpg	engineSize	transmission_ID	fuel_ID	model_ID	model_name
1	1	14	2017	12500	15735	150	55.4000015258789	1.39999997615814	2	5	14	A1
2	2	19	2016	16500	36203	20	64.1999969482422	2	1	1	19	A6
3	3	14	2016	11000	29946	30	55.4000015258789	1.39999997615814	2	5	14	A1
4	4	17	2017	16800	25952	145	67.3000030517578	2	1	1	17	A4
5	5	16	2019	17300	1998	145	49.5999984741211	1	2	5	16	A3
6	6	14	2016	13900	32260	30	58.9000015258789	1.39999997615814	1	5	14	A1
7	7	19	2016	13250	76788	30	61.4000015258789	2	1	1	19	A6
8	8	17	2016	11750	75185	20	70.5999984741211	2	2	1	17	A4
9	9	16	2015	10200	46112	20	60.0999984741211	1.39999997615814	2	5	16	A3
10	10	14	2016	12000	22451	30	55.4000015258789	1.39999997615814	2	5	14	A1
11	11	16	2017	16100	28955	145	58.9000015258789	1.39999997615814	2	5	16	A3

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

```
select top 10 a.id as sales_of_cars,a.mpg as fuel_efficiency,a.price,b.fueltype from(
SELECT *FROM audi UNION ALL SELECT*FROM BMW
UNION ALL SELECT*FROM hyundai UNION ALL SELECT*FROM merc union all select *from cclass
) as a left join fueltype as b on a.fuel_ID=b.fuel_id;
```

	sales_of_cars	fuel_efficiency	price	fueltype
1	1	55.4000015258789	12500	Petrol
2	2	64.1999969482422	16500	Diesel
3	3	55.4000015258789	11000	Petrol
4	4	67.3000030517578	16800	Diesel
5	5	49.5999984741211	17300	Petrol
6	6	58.9000015258789	13900	Petrol
7	7	61.4000015258789	13250	Diesel
8	8	70.5999984741211	11750	Diesel
9	9	60.0999984741211	10200	Petrol
10	10	55.4000015258789	12000	Petrol

PROJECT SQL

```
--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)
```

```
select top 20 count(id) as sales_car,year,petrol_price,diesel_price from  
(select a.*,b.petrol_price,b.diesel_price from  
(select a.*,b.fueltype from(  
SELECT *FROM audi UNION ALL SELECT*FROM BMW  
UNION ALL SELECT*FROM hyundai UNION ALL SELECT*FROM merc union all select *from cclass  
) as a left join fueltype as b on a.fuel_ID=b.fuel_id) as a left join  
(SELECT YEAR(DATE) AS YEAR,round(AVG(PETROL),2) AS petrol_price,  
round(avg(diesel),2)as diesel_price from fuel_price_in_uk  
group by YEAR(DATE) ) as b on a.year=b.year)c  
group by year,petrol_price,diesel_price order by year desc
```

	sales_car	year	petrol_price	diesel_price
1	2453	2020	114.09	119.46
2	13911	2019	124.97	131.71
3	4528	2018	125.03	130.04
4	7880	2017	117.51	120.21
5	7078	2016	108.95	110.42
6	3566	2015	111.03	115
7	1760	2014	127.42	133.47
8	1124	2013	134.32	140.59
9	309	2012	135.75	142.16
10	168	2011	133.41	138.81
11	146	2010	116.9	119.23

```
--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)
```

```
select top 10 count(a.id) as sales_car,sum(price)as sale_car_cost,b.model_name from(  
SELECT *FROM audi UNION ALL SELECT*FROM BMW  
UNION ALL SELECT*FROM hyundai UNION ALL SELECT*FROM merc union all select *from cclass  
) as a inner join models as b on a.model_id=b.model_id  
group by model_name order by sales_car desc
```

	sales_car	sale_car_cost	model_name
1	7646	181094291	C Class
2	2561	50834818	A Class
3	2443	48568936	3 Series
4	1969	31152869	1 Series
5	1953	49765215	E Class
6	1929	33581039	A3
7	1417	32589954	Q3
8	1381	27972777	A4
9	1347	19299480	A1
10	1300	20564328	Tucson

PROJECT SQL

--You are also asked to rank across all the models based on their total sales, average price, average mileage, average engine size, etc.

```
select dense_rank() over (order by count(a.id) desc) as sales_car,  
avg(price) as avg_price, b.model_name, avg(mileage) as avg_mileage  
, avg(engine_size) as avg_engine_size, count(a.id) as total_sales_car from(  
SELECT * FROM audi UNION ALL SELECT * FROM BMW  
UNION ALL SELECT * FROM hyundai UNION ALL SELECT * FROM merc union all select * from cclass  
) as a inner join models as b on a.model_id=b.model_id  
group by model_name
```

	sales_car	avg_price	model_name	avg_mileage	avg_engine_size	total_sales_car
1	1	23684	C Class	22279	2.03399159733363	7646
2	2	19849	A Class	20995	1.6585318006514	2561
3	3	19880	3 Series	31858	2.18309455600164	2443
4	4	15821	1 Series	27916	1.88486541511021	1969
5	5	25481	E Class	23993	2.26052225948418	1953
6	6	17408	A3	28891	1.64147226658431	1929
7	7	22999	Q3	21194	1.78045165378945	1417
8	8	20255	A4	29690	2.0135409088777	1381
9	9	14327	A1	24498	1.36391981602069	1347
10	10	15818	Tucson	21095	1.67346157018955	1300
11	11	19539	2 Series	17042	1.87347436940602	1229

--and now filter the top 5 basis their sales. Observe the identified models and provide your inference.

```
select top 5 dense_rank() over (order by count(a.id) desc) as sales_car,  
avg(price) as avg_price, b.model_name, avg(mileage) as avg_mileage  
, avg(engine_size) as avg_engine_size, count(a.id) as total_sales_car from(  
SELECT * FROM audi UNION ALL SELECT * FROM BMW  
UNION ALL SELECT * FROM hyundai UNION ALL SELECT * FROM merc union all select * from cclass  
) as a inner join models as b on a.model_id=b.model_id  
group by model_name
```

	sales_car	avg_price	model_name	avg_mileage	avg_engine_size	total_sales_car
1	1	23684	C Class	22279	2.03399159733363	7646
2	2	19849	A Class	20995	1.6585318006514	2561
3	3	19880	3 Series	31858	2.18309455600164	2443
4	4	15821	1 Series	27916	1.88486541511021	1969
5	5	25481	E Class	23993	2.26052225948418	1953