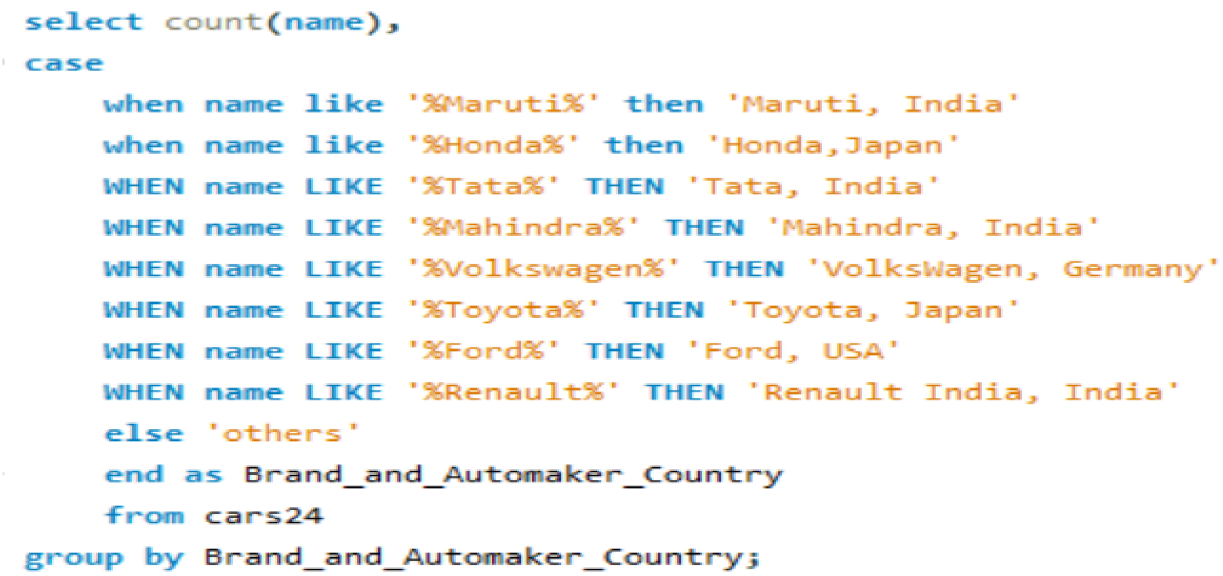
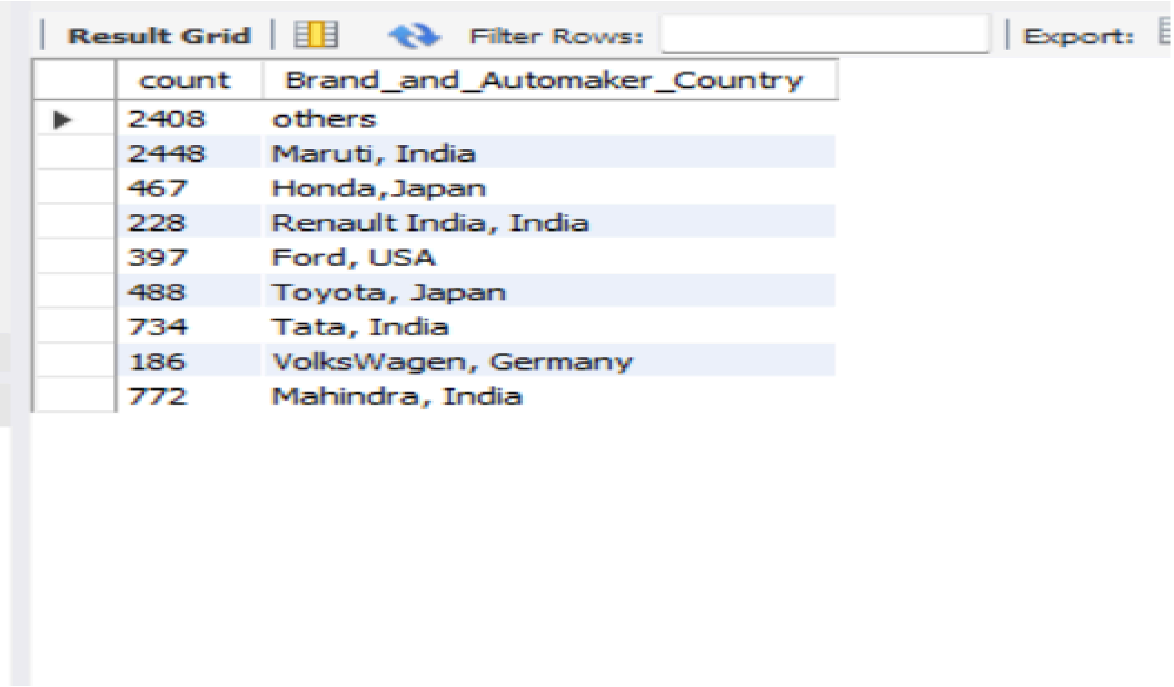
# CAP-DS-05 | SQL Mini Project L1

### Insight number 1 -

Write a query to count different cars according to their brands and their manufacturer country. Print Count(Count of specific car),Brand & Country column name as ‘Brand_and_Automaker_Country’ , it will represent the country of origin for each automaker based on the specified conditions.



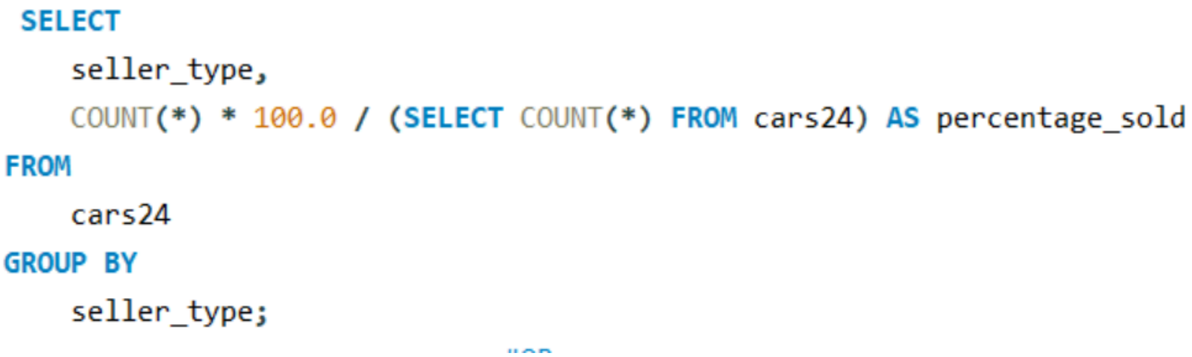
```
select count(name),
case
    when name like '%Maruti%' then 'Maruti, India'
    when name like '%Honda%' then 'Honda,Japan'
    WHEN name LIKE '%Tata%' THEN 'Tata, India'
    WHEN name LIKE '%Mahindra%' THEN 'Mahindra, India'
    WHEN name LIKE '%Volkswagen%' THEN 'VolksWagen, Germany'
    WHEN name LIKE '%Toyota%' THEN 'Toyota, Japan'
    WHEN name LIKE '%Ford%' THEN 'Ford, USA'
    WHEN name LIKE '%Renault%' THEN 'Renault India, India'
    else 'others'
    end as Brand_and_Automaker_Country
    from cars24
group by Brand_and_Automaker_Country;
```



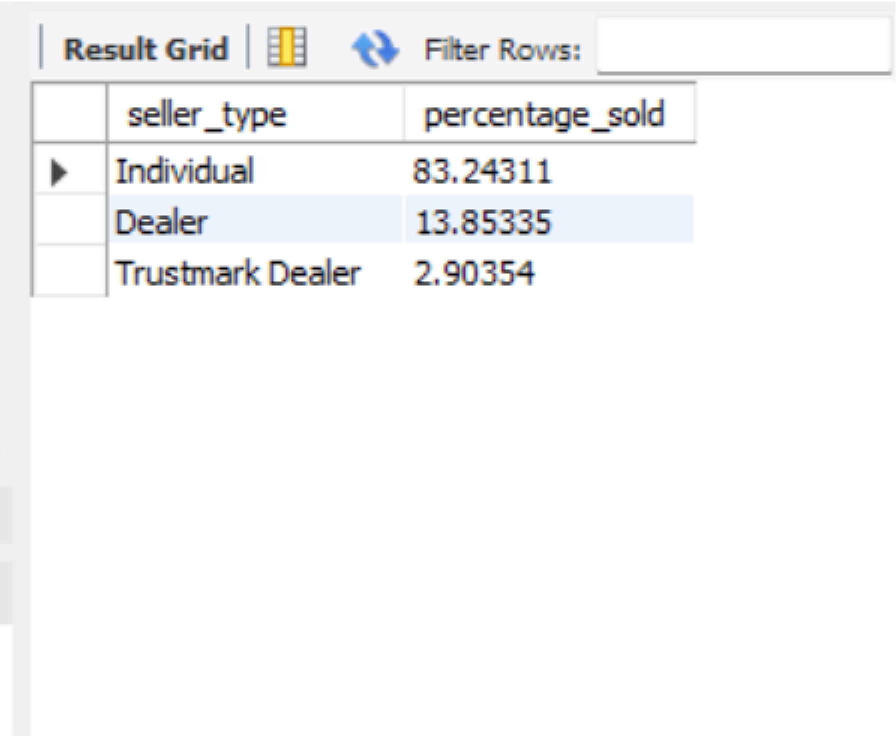
| count | Brand_and_Automaker_Country |
|---|---|
| 2408 | others |
| 2448 | Maruti, India |
| 467 | Honda,Japan |
| 228 | Renault India, India |
| 397 | Ford, USA |
| 488 | Toyota, Japan |
| 734 | Tata, India |
| 186 | VolksWagen, Germany |
| 772 | Mahindra, India |

## Insight number 2 -

Write a query to Calculate the percentage of cars sold by each seller type compared to the total number of cars.



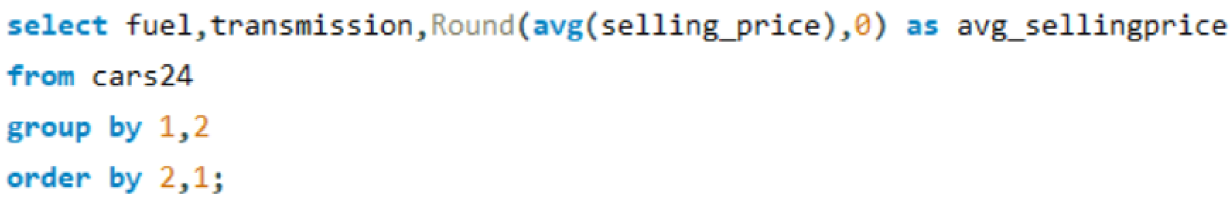
```
 SELECT
    seller_type,
    COUNT(*) * 100.0 / (SELECT COUNT(*) FROM cars24) AS percentage_sold
FROM
    cars24
GROUP BY
    seller_type;
```



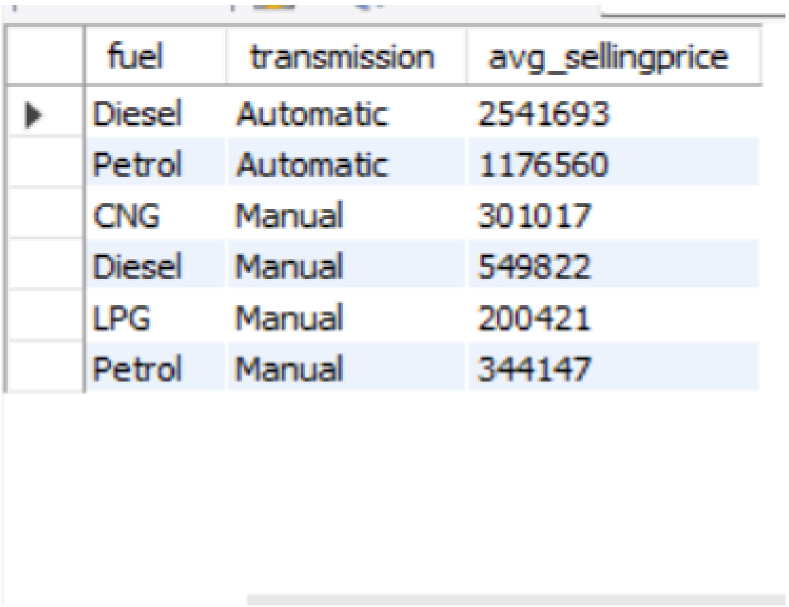
| seller_type | percentage_sold |
|---|---|
| Individual | 83.24311 |
| Dealer | 13.85335 |
| Trustmark Dealer | 2.90354 |

## Insight number 3 -

Write a query to find the average selling price for each combination of fuel type and transmission type. Print fuel, transmission, avg_sellingprice in your output, also round off avg_sellingprice to 0 places of decimal. Order your output by transmission in ascending order for the same transmission order by fuel.



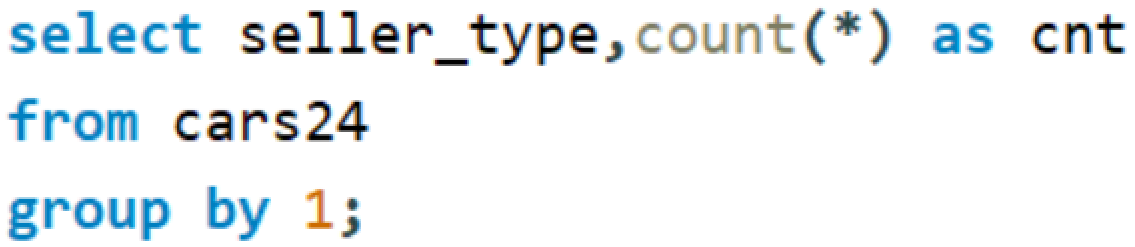
```
select fuel,transmission,Round(avg(selling_price),0) as avg_sellingprice
from cars24
group by 1,2
order by 2,1;
```



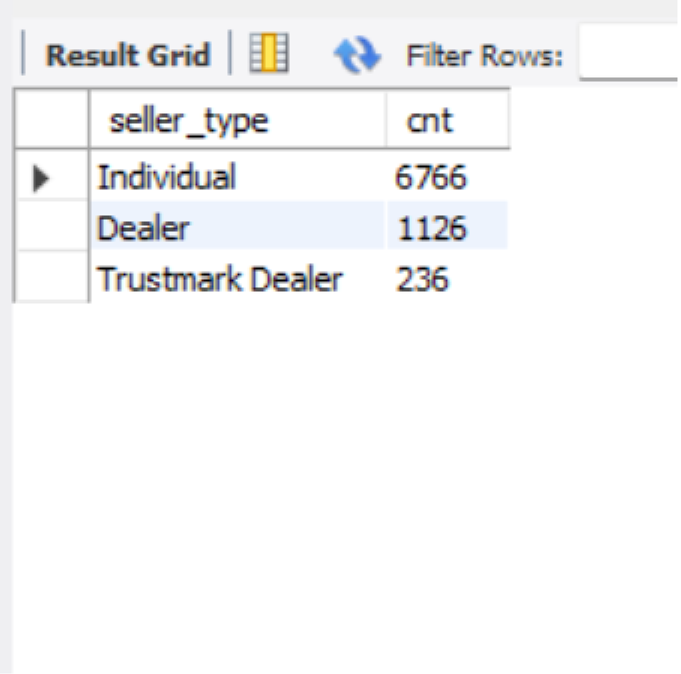
| | fuel | transmission | avg_sellingprice |
|---|---|---|---|
| ▶ | Diesel | Automatic | 2541693 |
| | Petrol | Automatic | 1176560 |
| | CNG | Manual | 301017 |
| | Diesel | Manual | 549822 |
| | LPG | Manual | 200421 |
| | Petrol | Manual | 344147 |

### Insight number 4 -

Write a query to Calculate the total number of cars sold by each seller type.



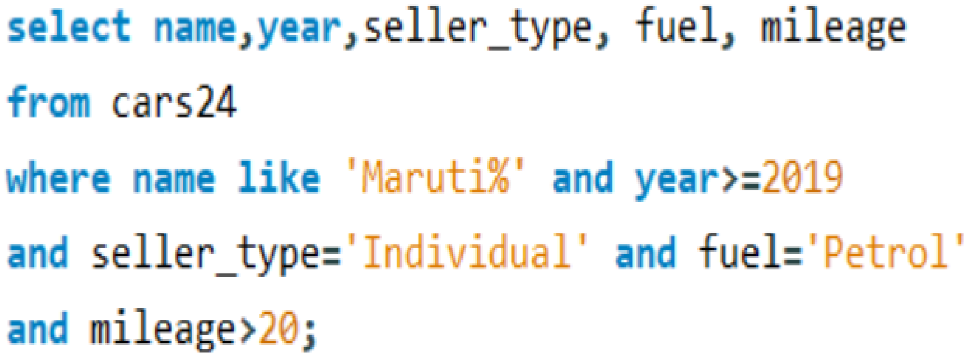
```
select seller_type,count(*) as cnt
from cars24
group by 1;
```



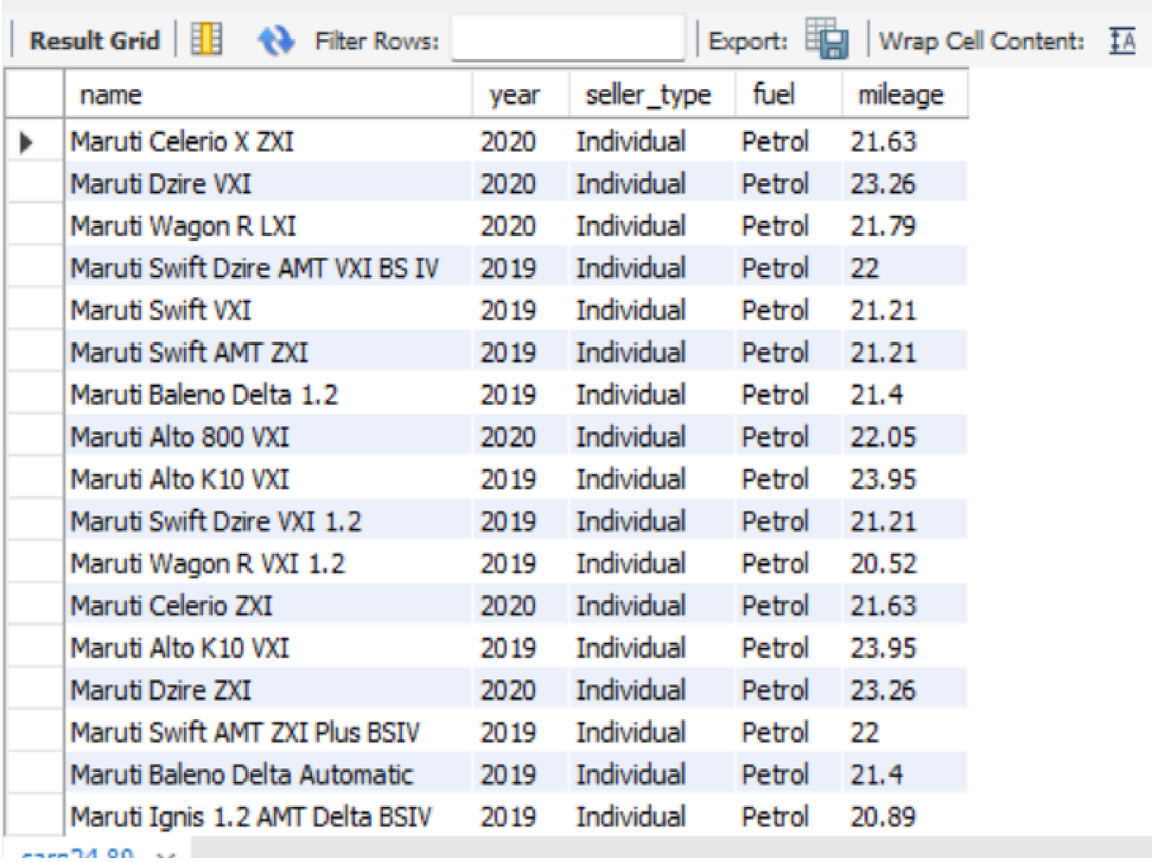
| seller_type | cnt |
|---|---|
| Individual | 6766 |
| Dealer | 1126 |
| Trustmark Dealer | 236 |

### Insight number 5 -

Suppose there's a  customer who only wants a Maruti Car from an individual seller whose year of manufacturing should not be less than 2019, with a fuel as Petrol that gives mileage greater than 20.
Write a query for it and Print name, year, seller_type, fuel, mileage in your output.



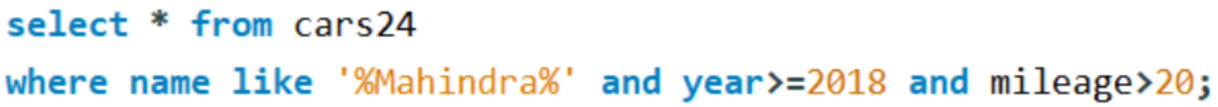
```
select name,year,seller_type, fuel, mileage
from cars24
where name like 'Maruti%' and year>=2019
and seller_type='Individual' and fuel='Petrol'
and mileage>20;
```



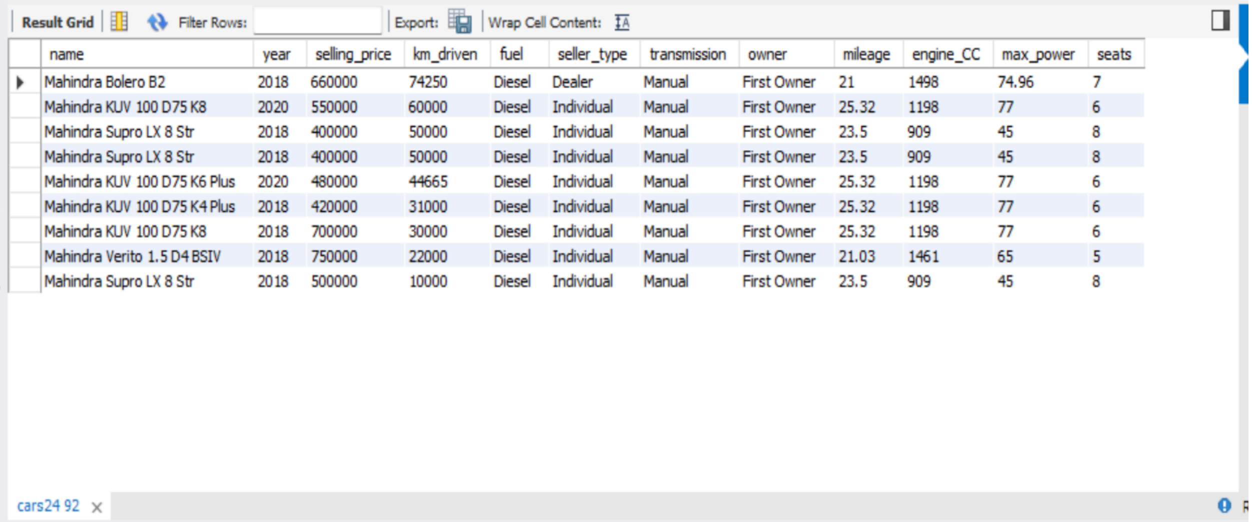
| name | year | seller_type | fuel | mileage |
|---|---|---|---|---|
| Maruti Celerio X ZXI | 2020 | Individual | Petrol | 21.63 |
| Maruti Dzire VXI | 2020 | Individual | Petrol | 23.26 |
| Maruti Wagon R LXI | 2020 | Individual | Petrol | 21.79 |
| Maruti Swift Dzire AMT VXI BS IV | 2019 | Individual | Petrol | 22 |
| Maruti Swift VXI | 2019 | Individual | Petrol | 21.21 |
| Maruti Swift AMT ZXI | 2019 | Individual | Petrol | 21.21 |
| Maruti Baleno Delta 1.2 | 2019 | Individual | Petrol | 21.4 |
| Maruti Alto 800 VXI | 2020 | Individual | Petrol | 22.05 |
| Maruti Alto K10 VXI | 2019 | Individual | Petrol | 23.95 |
| Maruti Swift Dzire VXI 1.2 | 2019 | Individual | Petrol | 21.21 |
| Maruti Wagon R VXI 1.2 | 2019 | Individual | Petrol | 20.52 |
| Maruti Celerio ZXI | 2020 | Individual | Petrol | 21.63 |
| Maruti Alto K10 VXI | 2019 | Individual | Petrol | 23.95 |
| Maruti Dzire ZXI | 2020 | Individual | Petrol | 23.26 |
| Maruti Swift AMT ZXI Plus BSIV | 2019 | Individual | Petrol | 22 |
| Maruti Baleno Delta Automatic | 2019 | Individual | Petrol | 21.4 |
| Maruti Ignis 1.2 AMT Delta BSIV | 2019 | Individual | Petrol | 20.89 |

### Insight number 6 -

Write a query for a customer who wants Mahindra Car whose manufacturer year is not less than 2018 and will give mileage of at least 20.



```
select * from cars24
where name like '%Mahindra%' and year>=2018 and mileage>20;
```



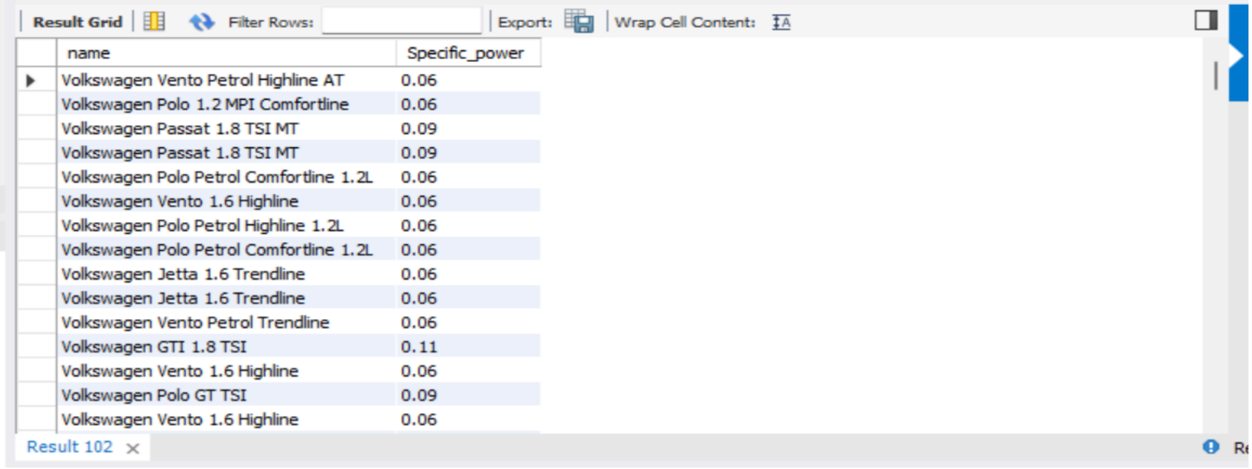
| name | year | selling_price | km_driven | fuel | seller_type | transmission | owner | mileage | engine_CC | max_power | seats |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Mahindra Bolero B2 | 2018 | 660000 | 74250 | Diesel | Dealer | Manual | First Owner | 21 | 1498 | 74.96 | 7 |
| Mahindra KUV 100 D75 K8 | 2020 | 550000 | 60000 | Diesel | Individual | Manual | First Owner | 25.32 | 1198 | 77 | 6 |
| Mahindra Supro LX 8 Str | 2018 | 400000 | 50000 | Diesel | Individual | Manual | First Owner | 23.5 | 909 | 45 | 8 |
| Mahindra Supro LX 8 Str | 2018 | 400000 | 50000 | Diesel | Individual | Manual | First Owner | 23.5 | 909 | 45 | 8 |
| Mahindra KUV 100 D75 K6 Plus | 2020 | 480000 | 44665 | Diesel | Individual | Manual | First Owner | 25.32 | 1198 | 77 | 6 |
| Mahindra KUV 100 D75 K4 Plus | 2018 | 420000 | 31000 | Diesel | Individual | Manual | First Owner | 25.32 | 1198 | 77 | 6 |
| Mahindra KUV 100 D75 K8 | 2018 | 700000 | 30000 | Diesel | Individual | Manual | First Owner | 25.32 | 1198 | 77 | 6 |
| Mahindra Verito 1.5 D4 BSIV | 2018 | 750000 | 22000 | Diesel | Individual | Manual | First Owner | 21.03 | 1461 | 65 | 5 |
| Mahindra Supro LX 8 Str | 2018 | 500000 | 10000 | Diesel | Individual | Manual | First Owner | 23.5 | 909 | 45 | 8 |

## Insight number 7 -

Write a query to Calculate the specific power for Volkswagen Cars

***Note: (Specific Power=Max Power/Engine Displacement)***

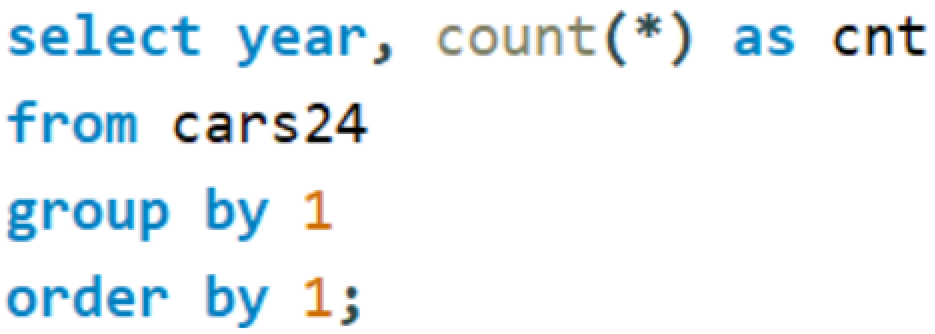
```
select name,Round((max_power/engine_CC),2) as Specific_power
from cars24
where name like 'Volkswagen%';
```



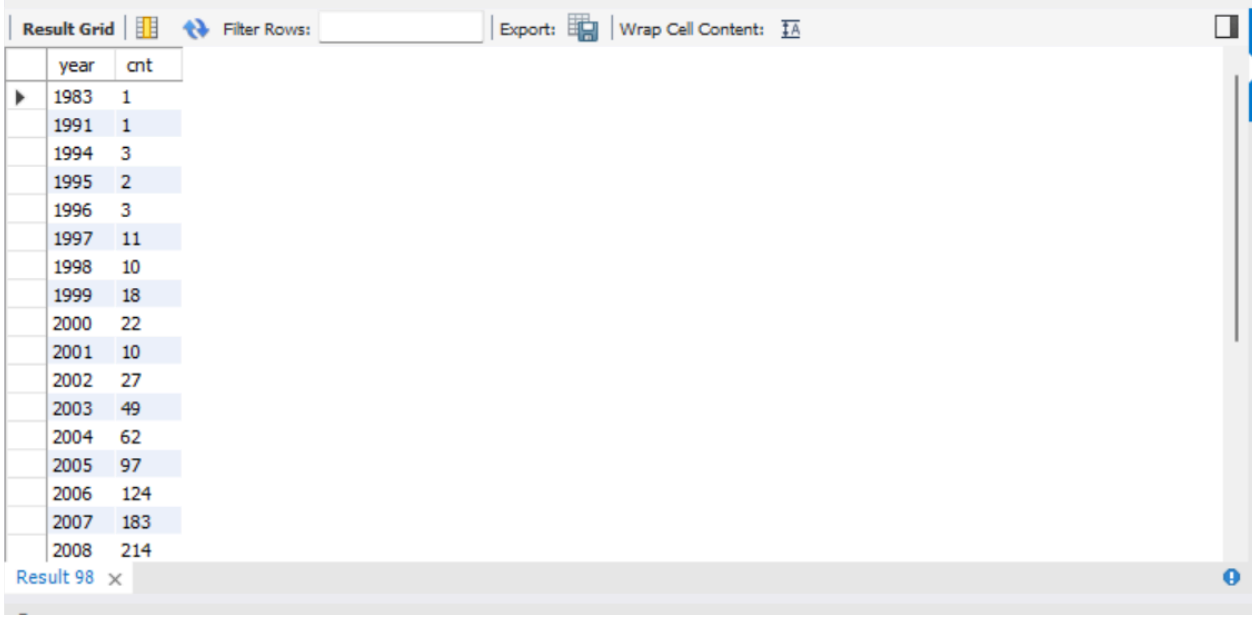
| name | Specific_power |
|---|---|
| Volkswagen Vento Petrol Highline AT | 0.06 |
| Volkswagen Polo 1.2 MPI Comfortline | 0.06 |
| Volkswagen Passat 1.8 TSI MT | 0.09 |
| Volkswagen Passat 1.8 TSI MT | 0.09 |
| Volkswagen Polo Petrol Comfortline 1.2L | 0.06 |
| Volkswagen Vento 1.6 Highline | 0.06 |
| Volkswagen Polo Petrol Highline 1.2L | 0.06 |
| Volkswagen Polo Petrol Comfortline 1.2L | 0.06 |
| Volkswagen Jetta 1.6 Trendline | 0.06 |
| Volkswagen Jetta 1.6 Trendline | 0.06 |
| Volkswagen Vento Petrol Trendline | 0.06 |
| Volkswagen GTI 1.8 TSI | 0.11 |
| Volkswagen Vento 1.6 Highline | 0.06 |
| Volkswagen Polo GT TSI | 0.09 |
| Volkswagen Vento 1.6 Highline | 0.06 |

## Insight number 8 -

Write a query to Count the no. of cars manufactured each year.Print year, count of number of cars as cnt and Order your output by year in ascending order.



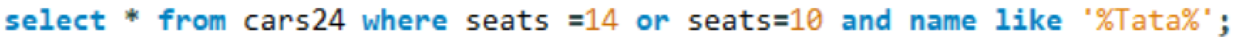
```sql
select year, count(*) as cnt
from cars24
group by 1
order by 1;
```



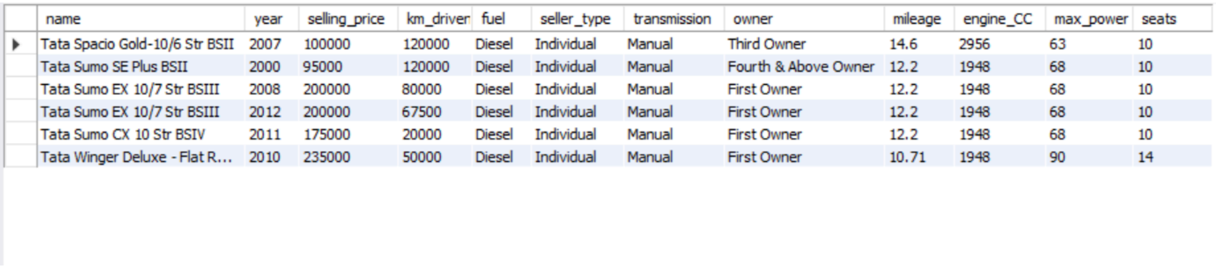
| year | cnt |
|---|---|
| 1983 | 1 |
| 1991 | 1 |
| 1994 | 3 |
| 1995 | 2 |
| 1996 | 3 |
| 1997 | 11 |
| 1998 | 10 |
| 1999 | 18 |
| 2000 | 22 |
| 2001 | 10 |
| 2002 | 27 |
| 2003 | 49 |
| 2004 | 62 |
| 2005 | 97 |
| 2006 | 124 |
| 2007 | 183 |
| 2008 | 214 |

### Insight number 9 -

Suppose a customer wants a Tata car with seats of either 14 or 10.Print full car details.



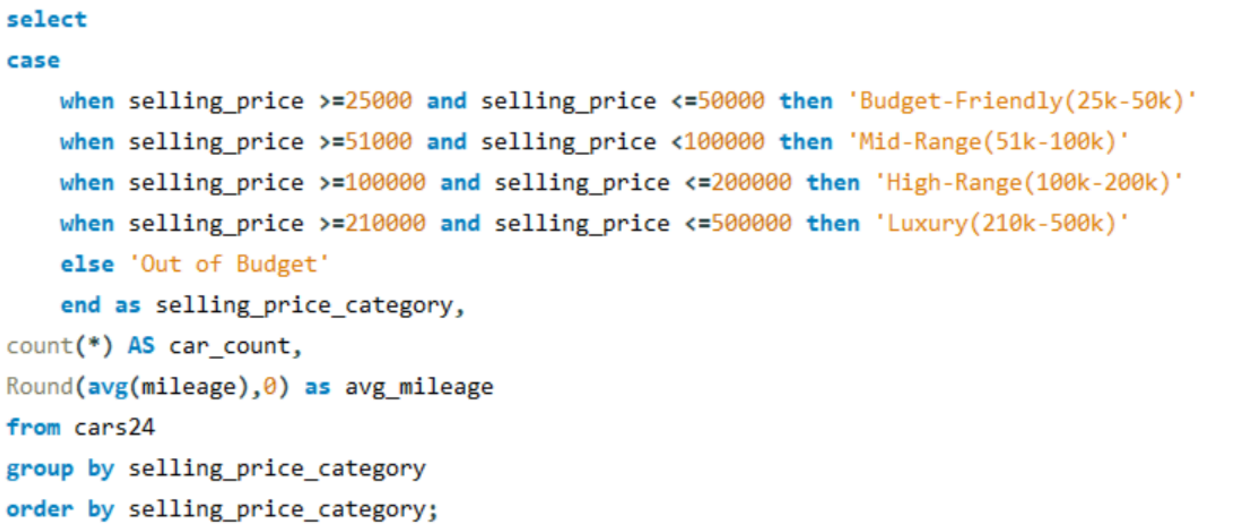
```sql
select * from cars24 where seats =14 or seats=10 and name like '%Tata%';
```



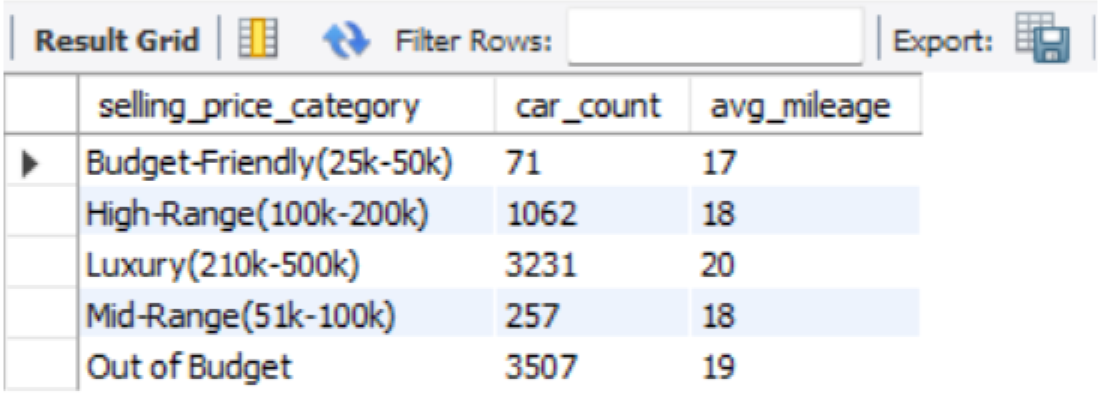
| name | year | selling_price | km_driver | fuel | seller_type | transmission | owner | mileage | engine_CC | max_power | seats |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Tata Spacio Gold-10/6 Str BSII | 2007 | 100000 | 120000 | Diesel | Individual | Manual | Third Owner | 14.6 | 2956 | 63 | 10 |
| Tata Sumo SE Plus BSII | 2000 | 95000 | 120000 | Diesel | Individual | Manual | Fourth & Above Owner | 12.2 | 1948 | 68 | 10 |
| Tata Sumo EX 10/7 Str BSIII | 2008 | 200000 | 80000 | Diesel | Individual | Manual | First Owner | 12.2 | 1948 | 68 | 10 |
| Tata Sumo EX 10/7 Str BSIII | 2012 | 200000 | 67500 | Diesel | Individual | Manual | First Owner | 12.2 | 1948 | 68 | 10 |
| Tata Sumo CX 10 Str BSIV | 2011 | 175000 | 20000 | Diesel | Individual | Manual | First Owner | 12.2 | 1948 | 68 | 10 |
| Tata Winger Deluxe - Flat R... | 2010 | 235000 | 50000 | Diesel | Individual | Manual | First Owner | 10.71 | 1948 | 90 | 14 |

## Insight number 10 -

Write a query to count different cars according to their selling_price criteria, for selling_price >=25000 and selling_price <=50000 then 'Budget-Friendly(25k-50k)',for selling_price >=51000 and selling_price <100000 then 'Mid-Range(51k-100k)', for selling_price >=100000 and selling_price <=200000 then 'High-Range(100k-200k)', for selling_price >=210000 and selling_price <=500000 then 'Luxury(210k-500k)', otherwise 'Out of Budget' and give column name as selling_price_category.Print selling_price_category, no. of cars in each category as car_count & avg mileage for each category as avg_mileage. Order your output by ascending order of selling_price_category.



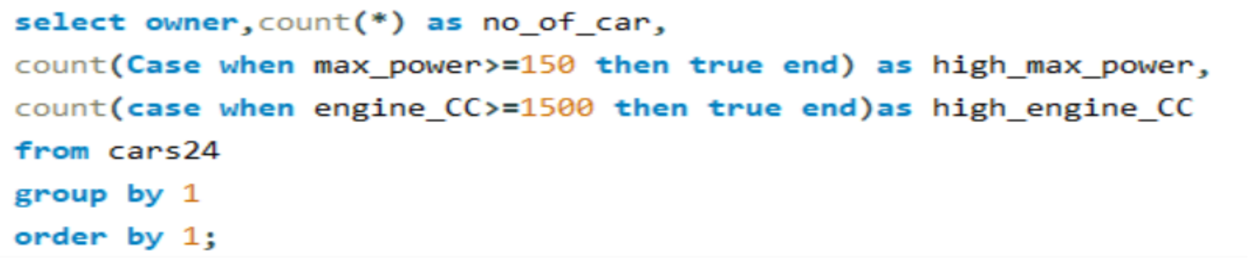
```
select
case
    when selling_price >=25000 and selling_price <=50000 then 'Budget-Friendly(25k-50k)'
    when selling_price >=51000 and selling_price <100000 then 'Mid-Range(51k-100k)'
    when selling_price >=100000 and selling_price <=200000 then 'High-Range(100k-200k)'
    when selling_price >=210000 and selling_price <=500000 then 'Luxury(210k-500k)'
    else 'Out of Budget'
    end as selling_price_category,
count(*) AS car_count,
Round(avg(mileage),0) as avg_mileage
from cars24
group by selling_price_category
order by selling_price_category;
```



| selling_price_category | car_count | avg_mileage |
|---|---|---|
| Budget-Friendly(25k-50k) | 71 | 17 |
| High-Range(100k-200k) | 1062 | 18 |
| Luxury(210k-500k) | 3231 | 20 |
| Mid-Range(51k-100k) | 257 | 18 |
| Out of Budget | 3507 | 19 |

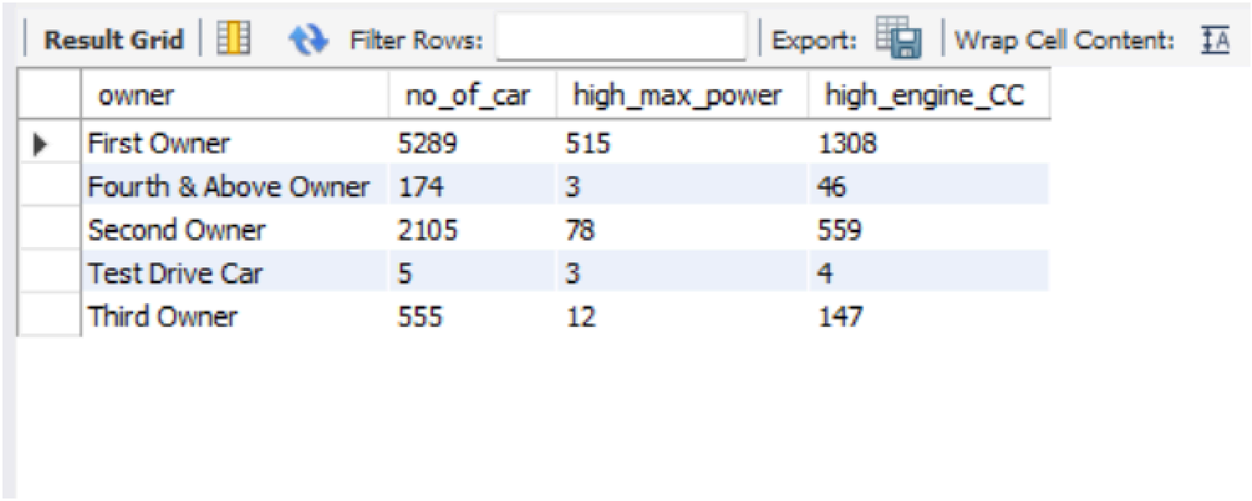
## Insight number 11 -

Write a query to count the number of cars having different owners, each car gives max_power of at least 150 and engine_cc at least 1500.Print Owner, Count of cars belonging to particular owner as no_of_car, max_power as high_max_power, engine_CC as high_engine_CC. Order your output by the owner in ascending order.



```
select owner,count(*) as no_of_car,
count(Case when max_power>=150 then true end) as high_max_power,
count(case when engine_CC>=1500 then true end)as high_engine_CC
from cars24
group by 1
order by 1;
```

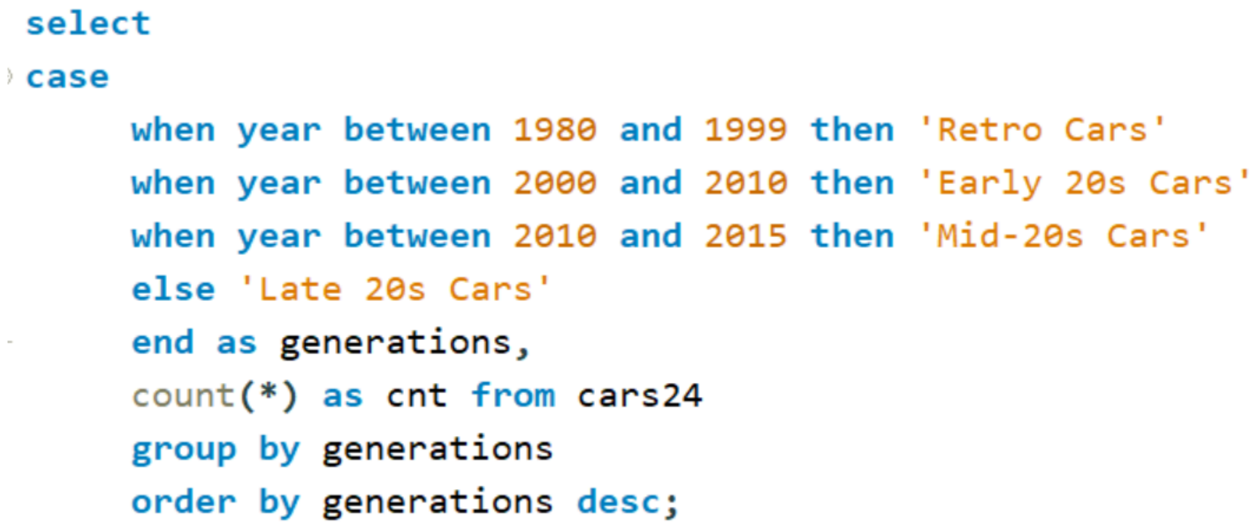
.



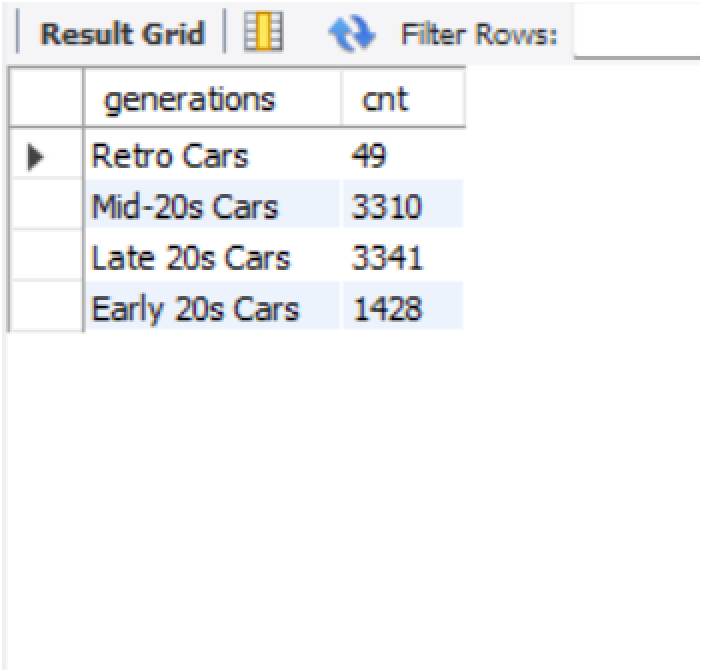
| owner | no_of_car | high_max_power | high_engine_CC |
|---|---|---|---|
| First Owner | 5289 | 515 | 1308 |
| Fourth & Above Owner | 174 | 3 | 46 |
| Second Owner | 2105 | 78 | 559 |
| Test Drive Car | 5 | 3 | 4 |
| Third Owner | 555 | 12 | 147 |

## Insight number 12 -

Write a query to count different cars that are separated based on their manufacturing year and categorize them as 1980-1999: Retro Cars,2000-2010: Early 20s Cars 2011-2015: Mid-20s Cars, 2015-2021: Late 20s Cars and give column name as generations. Order your output by descending order of generations.



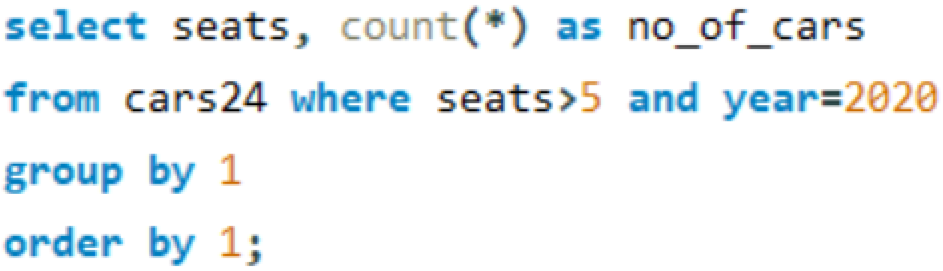
```
select
case
     when year between 1980 and 1999 then 'Retro Cars'
     when year between 2000 and 2010 then 'Early 20s Cars'
     when year between 2010 and 2015 then 'Mid-20s Cars'
     else 'Late 20s Cars'
     end as generations,
     count(*) as cnt from cars24
     group by generations
     order by generations desc;
```



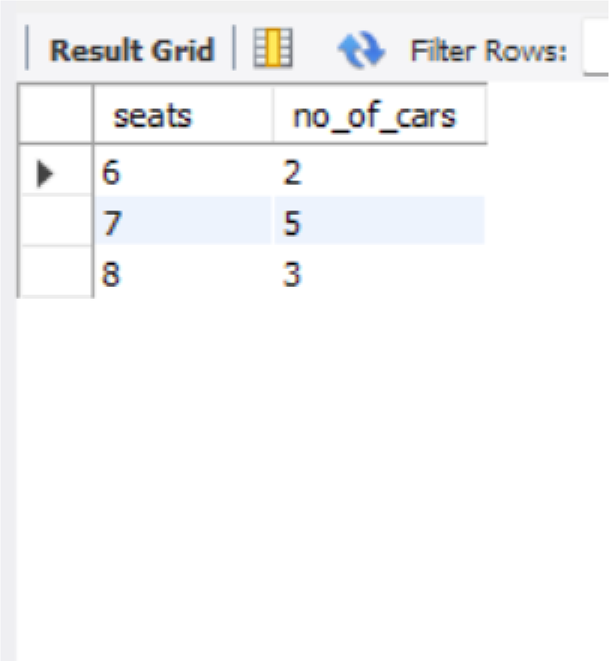
| generations | cnt |
|---|---|
| Retro Cars | 49 |
| Mid-20s Cars | 3310 |
| Late 20s Cars | 3341 |
| Early 20s Cars | 1428 |

### Insight number 13 -

count the no. of cars having seats more than 5, manufacturing year=2020. Print seats, count the number of cars in your output. Order your output by seats in ascending order.



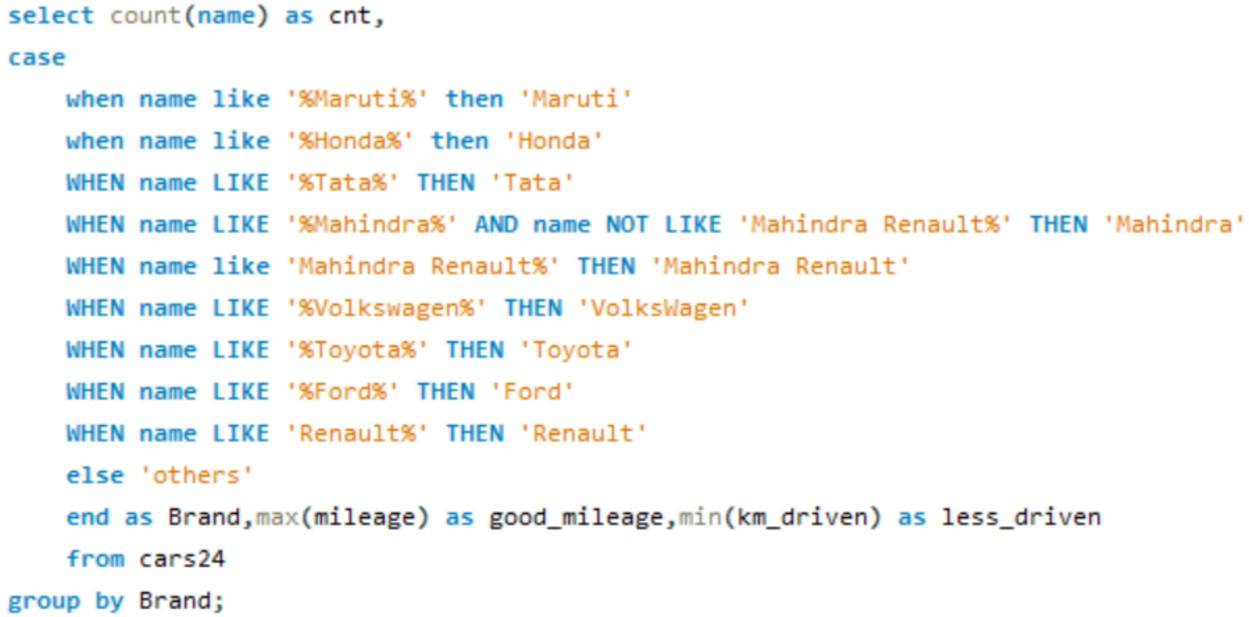
```
select seats, count(*) as no_of_cars
from cars24 where seats>5 and year=2020
group by 1
order by 1;
```



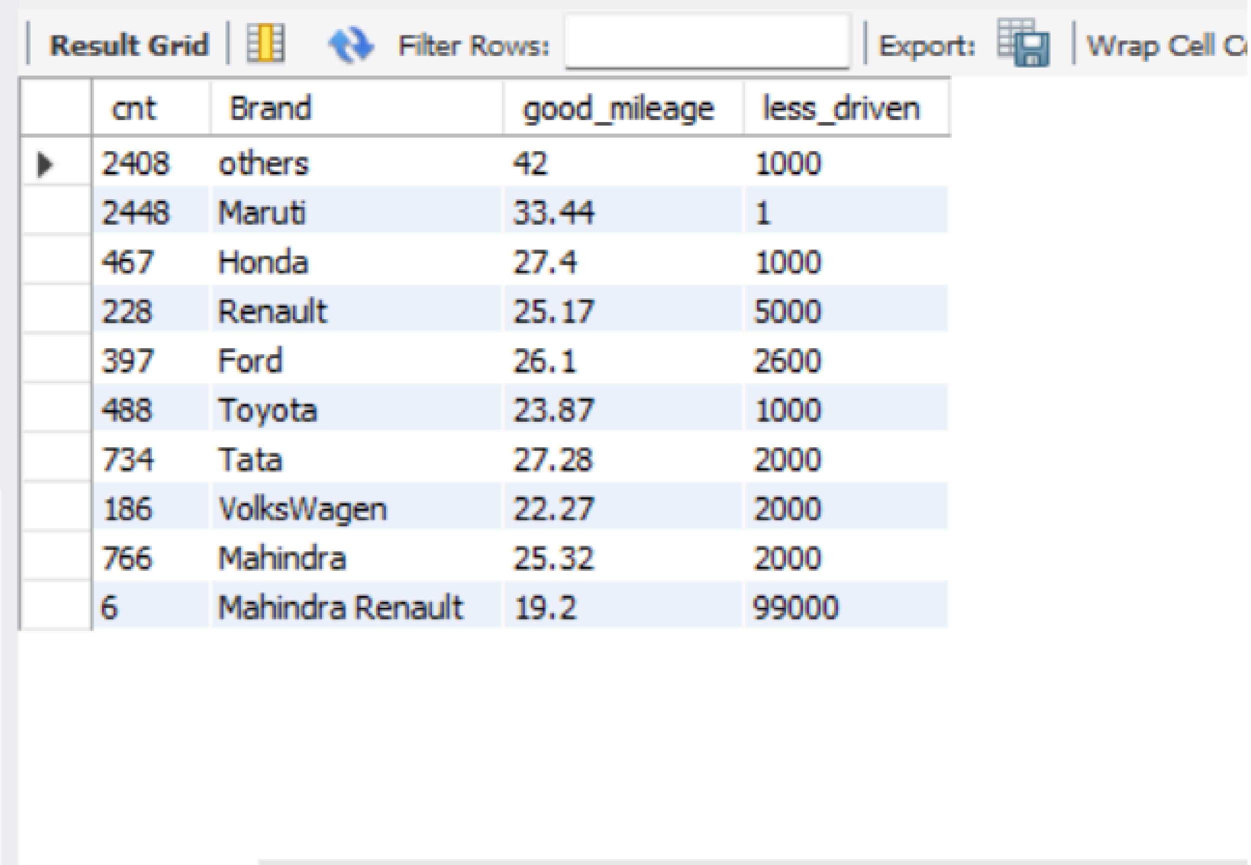
| seats | no_of_cars |
|---|---|
| 6 | 2 |
| 7 | 5 |
| 8 | 3 |

## Insight number 14 -

Suppose a customer wants a Car that will give good mileage and be less driven. Write a query for it. Print count of cars as cnt, Brand, mileage as good_mileage for each category and km_driven as less_driven.



```
select count(name) as cnt,
case
    when name like '%Maruti%' then 'Maruti'
    when name like '%Honda%' then 'Honda'
    WHEN name LIKE '%Tata%' THEN 'Tata'
    WHEN name LIKE '%Mahindra%' AND name NOT LIKE 'Mahindra Renault%' THEN 'Mahindra'
    WHEN name like 'Mahindra Renault%' THEN 'Mahindra Renault'
    WHEN name LIKE '%Volkswagen%' THEN 'VolksWagen'
    WHEN name LIKE '%Toyota%' THEN 'Toyota'
    WHEN name LIKE '%Ford%' THEN 'Ford'
    WHEN name LIKE 'Renault%' THEN 'Renault'
    else 'others'
    end as Brand,max(mileage) as good_mileage,min(km_driven) as less_driven
    from cars24
group by Brand;
```



| cnt | Brand | good_mileage | less_driven |
|---|---|---|---|
| 2408 | others | 42 | 1000 |
| 2448 | Maruti | 33.44 | 1 |
| 467 | Honda | 27.4 | 1000 |
| 228 | Renault | 25.17 | 5000 |
| 397 | Ford | 26.1 | 2600 |
| 488 | Toyota | 23.87 | 1000 |
| 734 | Tata | 27.28 | 2000 |
| 186 | VolksWagen | 22.27 | 2000 |
| 766 | Mahindra | 25.32 | 2000 |
| 6 | Mahindra Renault | 19.2 | 99000 |