

SQL MINI PROJECT L1

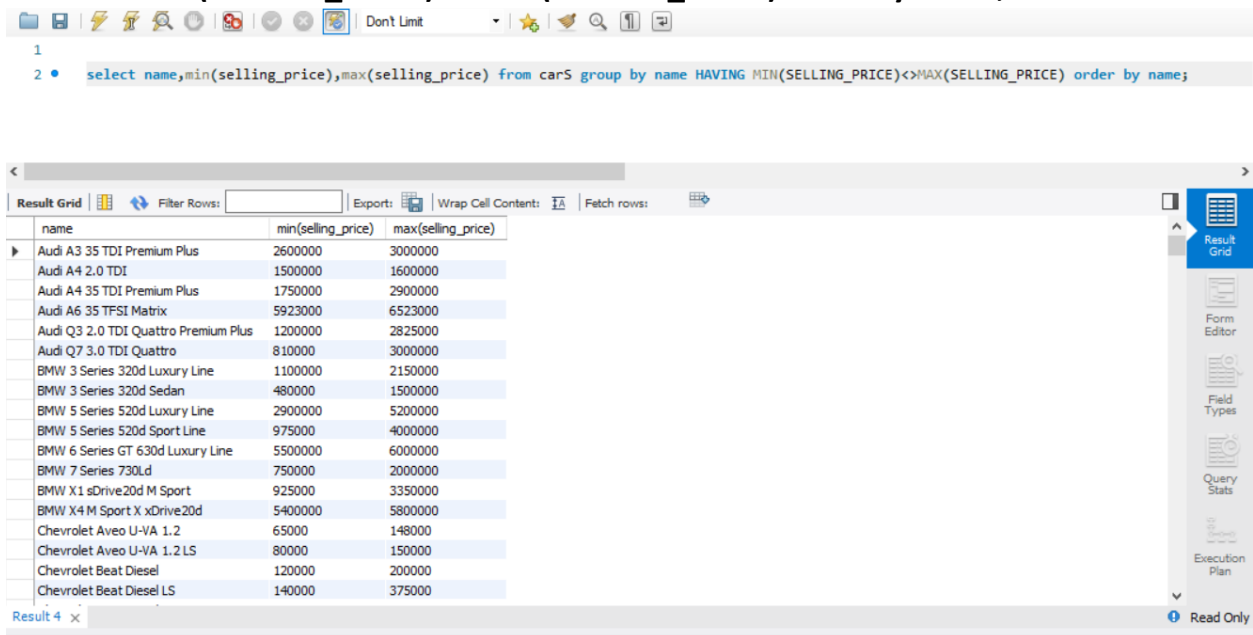
1-Which car names have different selling prices in the table, and what are the lowest and highest prices for each of them?

QUERY

```
select name,min(selling_price),max(selling_price)
```

```
from cars group by name
```

```
HAVING MIN(SELLING_PRICE)<>MAX(SELLING_PRICE) order by name;
```



The screenshot shows a SQL query editor with a toolbar at the top. The query is entered in the main text area and is numbered 2. Below the query, a 'Result Grid' is displayed, showing the results of the query. The grid has three columns: 'name', 'min(selling_price)', and 'max(selling_price)'. The results are listed in a table format with alternating light blue and white rows. The table includes car models and their corresponding minimum and maximum selling prices. The results are ordered by name. The interface also includes a 'Filter Rows' section, an 'Export' button, and a 'Fetch rows' button. On the right side, there is a sidebar with icons for 'Result Grid', 'Form Editor', 'Field Types', 'Query Stats', and 'Execution Plan'. At the bottom right, there is a 'Read Only' indicator.

name	min(selling_price)	max(selling_price)
Audi A3 35 TDI Premium Plus	2600000	3000000
Audi A4 2.0 TDI	1500000	1600000
Audi A4 35 TDI Premium Plus	1750000	2900000
Audi A6 35 TFSI Matrix	5923000	6523000
Audi Q3 2.0 TDI Quattro Premium Plus	1200000	2825000
Audi Q7 3.0 TDI Quattro	810000	3000000
BMW 3 Series 320d Luxury Line	1100000	2150000
BMW 3 Series 320d Sedan	480000	1500000
BMW 5 Series 520d Luxury Line	2900000	5200000
BMW 5 Series 520d Sport Line	975000	4000000
BMW 6 Series GT 630d Luxury Line	5500000	6000000
BMW 7 Series 730Ld	750000	2000000
BMW X1 sDrive20d M Sport	925000	3350000
BMW X4 M Sport X xDrive20d	5400000	5800000
Chevrolet Aveo U-VA 1.2	65000	148000
Chevrolet Aveo U-VA 1.2 LS	80000	150000
Chevrolet Beat Diesel	120000	200000
Chevrolet Beat Diesel LS	140000	375000

2-How many cars of each brand are there in the table, and which brand has the least number of cars?

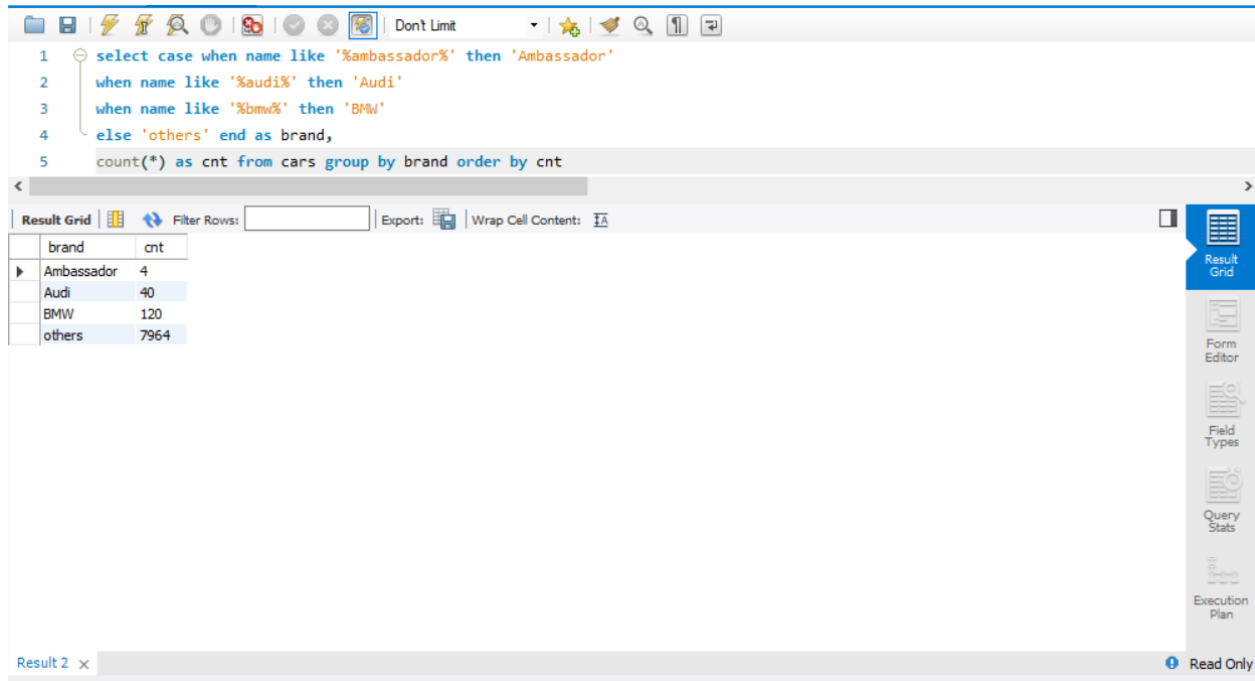
QUERY

```
select case when name like '%ambassador%' then 'Ambassador'
```

```
when name like '%audi%' then 'Audi'
```

```
when name like '%bmw%' then 'BMW'
```

else 'others' end as brand,count(*) as cnt from cars group by brand order by cnt



The screenshot shows a SQL query editor with the following query:

```
1 select case when name like '%ambassador%' then 'Ambassador'
2 when name like '%audi%' then 'Audi'
3 when name like '%bmw%' then 'BMW'
4 else 'others' end as brand,
5 count(*) as cnt from cars group by brand order by cnt
```

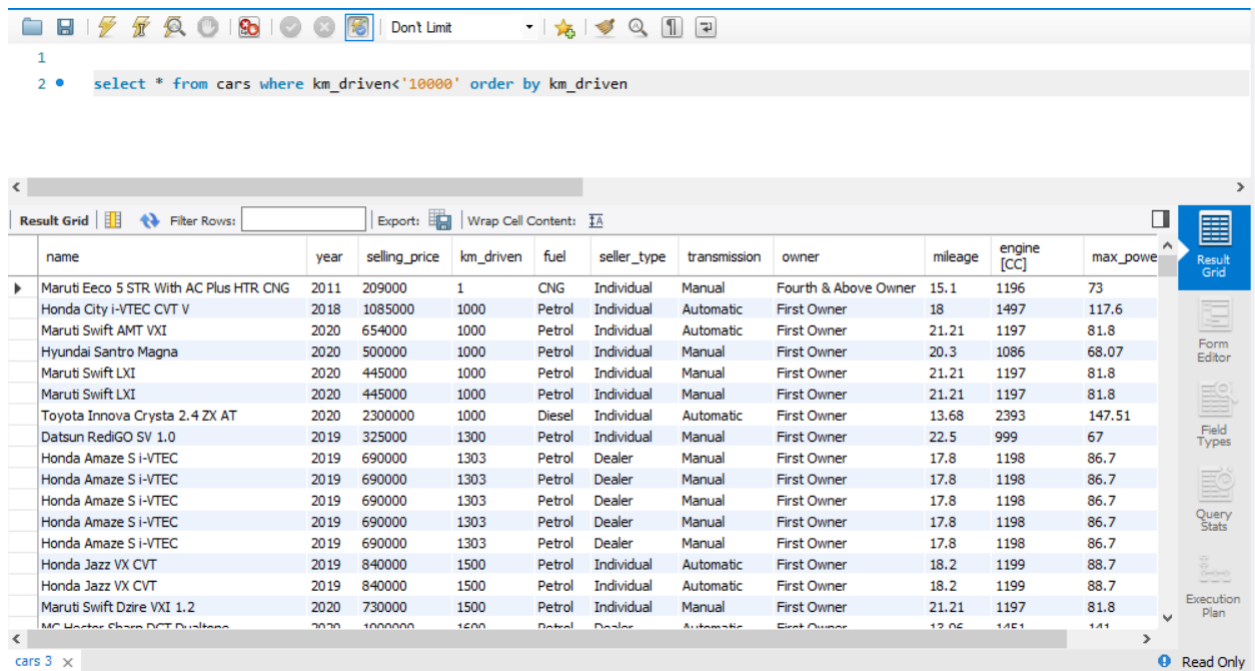
The result grid shows the following data:

brand	cnt
Ambassador	4
Audi	40
BMW	120
others	7964

3-WHICH CAR HAS DRIVEN LESS THAN 10000 KM.

QUERY

select * from cars where km_driven<'10000' order by km_driven



The screenshot shows a SQL query editor with the following query:

```
1
2 • select * from cars where km_driven<'10000' order by km_driven
```

The result grid shows the following data:

name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine [CC]	max_power
Maruti Eeco 5 STR With AC Plus HTR CNG	2011	209000	1	CNG	Individual	Manual	Fourth & Above Owner	15.1	1196	73
Honda City i-VTEC CVT V	2018	1085000	1000	Petrol	Individual	Automatic	First Owner	18	1497	117.6
Maruti Swift AMT VXi	2020	654000	1000	Petrol	Individual	Automatic	First Owner	21.21	1197	81.8
Hyundai Santro Magna	2020	500000	1000	Petrol	Individual	Manual	First Owner	20.3	1086	68.07
Maruti Swift LXI	2020	445000	1000	Petrol	Individual	Manual	First Owner	21.21	1197	81.8
Maruti Swift LXI	2020	445000	1000	Petrol	Individual	Manual	First Owner	21.21	1197	81.8
Toyota Innova Crysta 2.4 ZX AT	2020	2300000	1000	Diesel	Individual	Automatic	First Owner	13.68	2393	147.51
Datsun RediGO SV 1.0	2019	325000	1300	Petrol	Individual	Manual	First Owner	22.5	999	67
Honda Amaze S i-VTEC	2019	690000	1303	Petrol	Dealer	Manual	First Owner	17.8	1198	86.7
Honda Amaze S i-VTEC	2019	690000	1303	Petrol	Dealer	Manual	First Owner	17.8	1198	86.7
Honda Amaze S i-VTEC	2019	690000	1303	Petrol	Dealer	Manual	First Owner	17.8	1198	86.7
Honda Amaze S i-VTEC	2019	690000	1303	Petrol	Dealer	Manual	First Owner	17.8	1198	86.7
Honda Jazz VX CVT	2019	840000	1500	Petrol	Individual	Automatic	First Owner	18.2	1199	88.7
Honda Jazz VX CVT	2019	840000	1500	Petrol	Individual	Automatic	First Owner	18.2	1199	88.7
Maruti Swift Dzire VXi 1.2	2020	730000	1500	Petrol	Individual	Manual	First Owner	21.21	1197	81.8
MG Hector Sharp DCT Dually	2020	1000000	1600	Petrol	Dealer	Automatic	First Owner	12.05	1451	141

4- How many cars of each fuel type are there in the table, and which fuel type has the least number of cars?

QUERY

SELECT FUEL,COUNT(*) AS CNT FROM CARS GROUP BY FUEL ORDER BY CNT

The screenshot shows a database query interface. The query bar contains the text: `SELECT FUEL,COUNT(*) AS CNT FROM CARS GROUP BY FUEL ORDER BY CNT`. Below the query bar, the results are displayed in a table with two columns: FUEL and CNT. The results are ordered by CNT in ascending order.

FUEL	CNT
LPG	38
CNG	57
Petrol	3631
Diesel	4402

5- Print all information about a particular car.

QUERY

SELECT * FROM CARS WHERE NAME LIKE '%WAGON R%' ORDER BY SELLING_PRICE

The screenshot shows a database query interface. The query bar contains the text: `SELECT * FROM CARS WHERE NAME LIKE '%WAGON R%' ORDER BY SELLING_PRICE`. Below the query bar, the results are displayed in a table with 12 columns: name, year, selling_price, km_driven, fuel, seller_type, transmission, owner, mileage, engine [CC], max_power, and seats. The results are ordered by selling_price in ascending order.

name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine [CC]	max_power	seats
Maruti Wagon R. VXi	2004	33351	90000	Petrol	Individual	Manual	Third Owner	18.9	998	67.1	5
Maruti Wagon R. LXI	2003	50000	147000	Petrol	Individual	Manual	Fourth & Above Owner	18.9	998	67.1	5
Maruti Wagon R. LXI	2003	55000	167000	Petrol	Individual	Manual	Second Owner	18.9	998	67.1	5
Maruti Wagon R. LX	2006	65000	198000	Petrol	Individual	Manual	Second Owner	18.9	998	67.1	5
Maruti Wagon R. LXI	2005	65000	90000	Petrol	Individual	Manual	Second Owner	18.9	998	67.1	5
Maruti Wagon R. VXi	2003	65000	70000	Petrol	Individual	Manual	Fourth & Above Owner	18.9	998	67.1	5
Maruti Wagon R. LXI	2005	65000	35000	Petrol	Individual	Manual	First Owner	18.9	998	67.1	5
Maruti Wagon R. LXI	2005	65000	35000	Petrol	Individual	Manual	First Owner	18.9	998	67.1	5
Maruti Wagon R. LXI	2005	70000	188000	Petrol	Individual	Manual	Fourth & Above Owner	18.9	998	67.1	5
Maruti Wagon R. LXI	2006	70000	110000	Petrol	Individual	Manual	Fourth & Above Owner	18.9	998	67.1	5

6-WHICH IS THE MOST VALUED CAR IN THE TABLE?

QUERRY

SELECT name, year, selling_price FROM cars ORDER BY selling_price DESC LIMIT 1;

The screenshot shows a database query interface. At the top, there is a toolbar with various icons. Below the toolbar, the query is entered in a text area: `SELECT name, year, selling_price FROM cars ORDER BY selling_price DESC LIMIT 1;`. Below the query area, there is a section for the results. It includes a 'Result Grid' button, a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' button. The results are displayed in a table with the following data:

name	year	selling_price
Volvo XC90 T8 Excellence BSIV	2017	10000000

On the right side of the interface, there are buttons for 'Result Grid' and 'Form Editor'. At the bottom, there is a tab labeled 'cars 1' and a 'Read Only' status indicator.

7- WHAT IS THE OLDEST YEAR OF CAR IN CARS TABLE?

SELECT name, year, selling_price FROM cars ORDER BY YEAR ASC LIMIT 1;

The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons for file operations, query execution, and settings. Below the toolbar, the SQL query is entered in a text area: `1 SELECT name, year, selling_price FROM cars ORDER BY year ASC LIMIT 1;`. The query is highlighted in blue. Below the query area, there is a section for the results. It includes a 'Result Grid' tab, a 'Filter Rows' input field, and an 'Export' button. The 'Result Grid' is active, displaying a table with the following data:

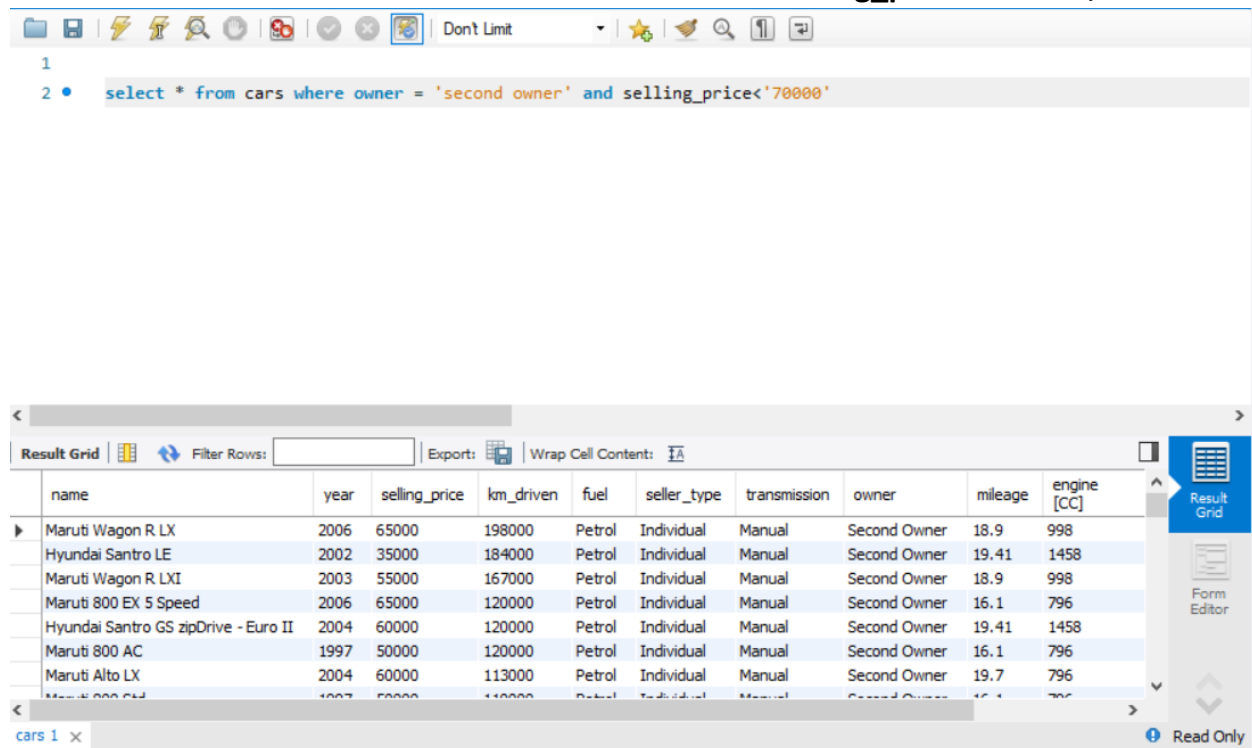
	name	year	selling_price
▶	Mahindra Willys CJ 3B 4X4	1983	300000

On the right side of the interface, there is a vertical toolbar with buttons for 'Result Grid' (selected), 'Form Editor', and a 'Read Only' status indicator at the bottom right.

8- Which car is selling by second owner and price below 70000

Query

select * from cars where owner = 'second owner' and selling_price<'70000' ;



The screenshot shows a database query interface. At the top, there's a toolbar with various icons. Below it, a SQL query is entered in a text area:

```
1
2 • select * from cars where owner = 'second owner' and selling_price<'70000'
```

Below the query, a table of results is displayed. The table has columns: name, year, selling_price, km_driven, fuel, seller_type, transmission, owner, mileage, and engine [CC]. The results are filtered to show only cars owned by 'Second Owner' with a selling price less than 70,000.

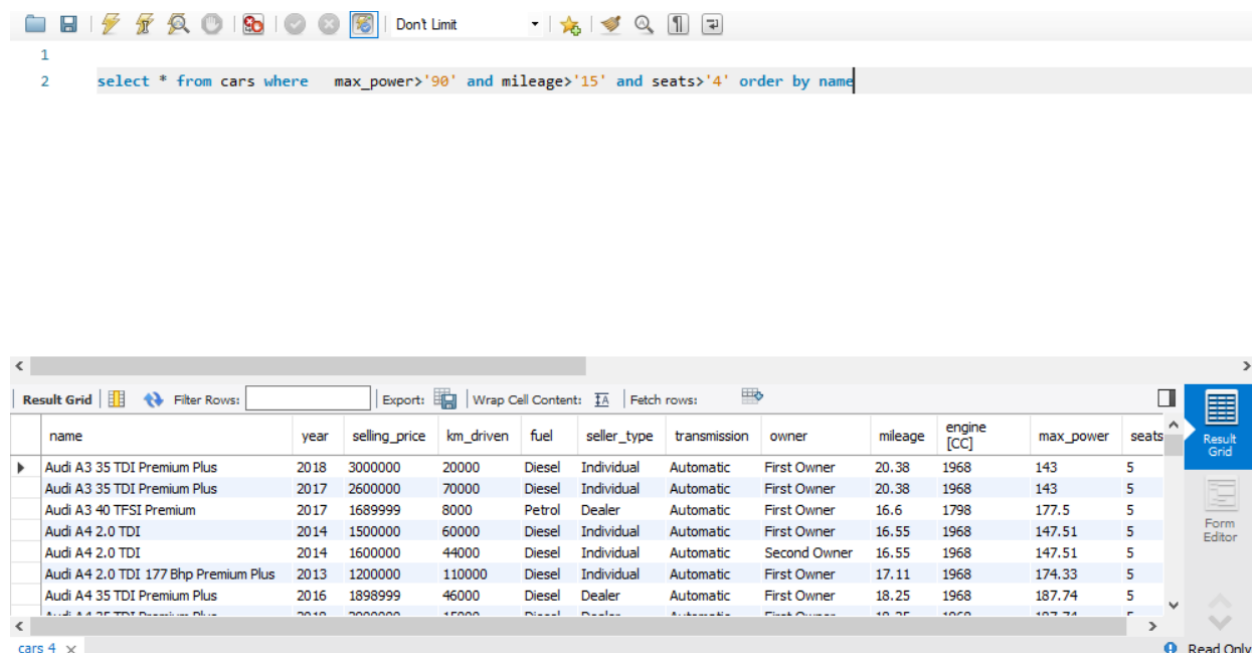
name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine [CC]
Maruti Wagon R LX	2006	65000	198000	Petrol	Individual	Manual	Second Owner	18.9	998
Hyundai Santro LE	2002	35000	184000	Petrol	Individual	Manual	Second Owner	19.41	1458
Maruti Wagon R LXI	2003	55000	167000	Petrol	Individual	Manual	Second Owner	18.9	998
Maruti 800 EX 5 Speed	2006	65000	120000	Petrol	Individual	Manual	Second Owner	16.1	796
Hyundai Santro GS zipDrive - Euro II	2004	60000	120000	Petrol	Individual	Manual	Second Owner	19.41	1458
Maruti 800 AC	1997	50000	120000	Petrol	Individual	Manual	Second Owner	16.1	796
Maruti Alto LX	2004	60000	113000	Petrol	Individual	Manual	Second Owner	19.7	796

The interface also includes a 'Result Grid' button, a 'Form Editor' button, and a 'Read Only' indicator.

9- Which cars have high performance, good fuel efficiency, Enough seats and what are their details?

QUERY

select * from cars where max_power>'90' and mileage>'15' and seats>'4';



The screenshot shows a database query interface. At the top, there's a toolbar with various icons. Below it, a SQL query is entered in a text area:

```
1
2 select * from cars where max_power>'90' and mileage>'15' and seats>'4' order by name
```

Below the query, a table of results is displayed. The table has columns: name, year, selling_price, km_driven, fuel, seller_type, transmission, owner, mileage, engine [CC], max_power, and seats. The results are ordered by name and filtered to show only cars with max_power > 90, mileage > 15, and seats > 4.

name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine [CC]	max_power	seats
Audi A3 35 TDI Premium Plus	2018	3000000	20000	Diesel	Individual	Automatic	First Owner	20.38	1968	143	5
Audi A3 35 TDI Premium Plus	2017	2600000	70000	Diesel	Individual	Automatic	First Owner	20.38	1968	143	5
Audi A3 40 TFSI Premium	2017	1689999	8000	Petrol	Dealer	Automatic	First Owner	16.6	1798	177.5	5
Audi A4 2.0 TDI	2014	1500000	60000	Diesel	Individual	Automatic	First Owner	16.55	1968	147.51	5
Audi A4 2.0 TDI	2014	1600000	44000	Diesel	Individual	Automatic	Second Owner	16.55	1968	147.51	5
Audi A4 2.0 TDI 177 Bhp Premium Plus	2013	1200000	110000	Diesel	Individual	Automatic	First Owner	17.11	1968	174.33	5
Audi A4 35 TDI Premium Plus	2016	1898999	46000	Diesel	Dealer	Automatic	First Owner	18.25	1968	187.74	5

The interface also includes a 'Result Grid' button, a 'Form Editor' button, and a 'Read Only' indicator.

10-caterise the table in 3 different price ranges low(below 70000),medium(between 70000 and 700000) and high(more than 700000).

QUERY

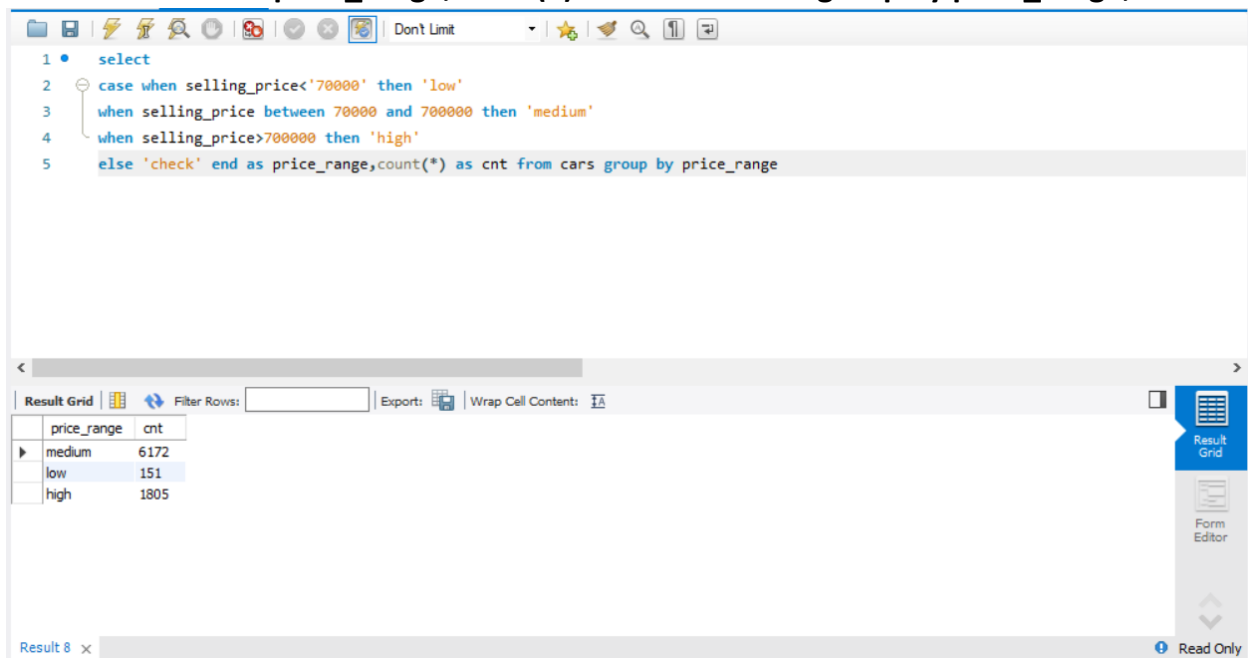
select

case when selling_price<'70000' then 'low'

when selling_price between 70000 and 700000 then 'medium'

when selling_price>700000 then 'high'

else 'check' end as price_range,count(*) as cnt from cars group by price_range;



The screenshot shows a SQL query editor with a query that categorizes cars into three price ranges: low, medium, and high. The query is as follows:

```
1 select
2 case when selling_price<'70000' then 'low'
3 when selling_price between 70000 and 700000 then 'medium'
4 when selling_price>700000 then 'high'
5 else 'check' end as price_range,count(*) as cnt from cars group by price_range
```

The results are displayed in a table with two columns: price_range and cnt.

price_range	cnt
medium	6172
low	151
high	1805

The interface includes a toolbar at the top with various icons, a "Filter Rows" section, and an "Export" button. The bottom right corner shows a "Result Grid" button and a "Form Editor" button.

11- Count every car's quantity in the table

Query

SELECT `NAME`, COUNT(*) AS `NUM`

FROM cars

GROUP BY `NAME`

order by num desc

[illegible]