

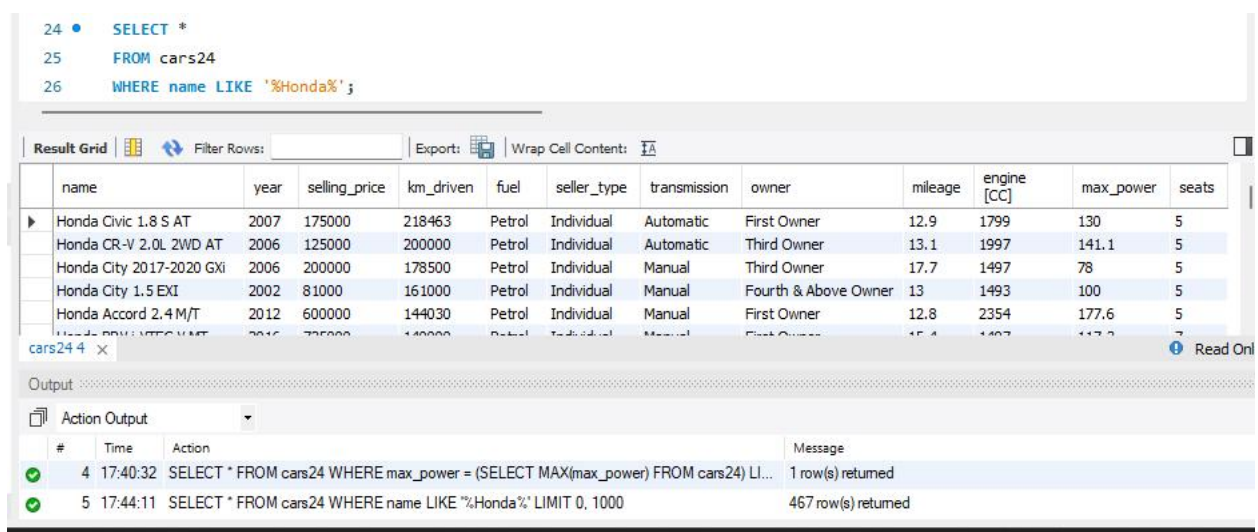
Project:1 SQL

1: This query is to find the details of specific brand of car honda .

SELECT *

FROM cars24

WHERE name LIKE '%Honda%'; --and from the number of rows returned we can find number of honda cars.



```
24 • SELECT *
25 FROM cars24
26 WHERE name LIKE '%Honda%';
```

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine [CC]	max_power	seats
▶	Honda Civic 1.8 S AT	2007	175000	218463	Petrol	Individual	Automatic	First Owner	12.9	1799	130	5
	Honda CR-V 2.0L 2WD AT	2006	125000	200000	Petrol	Individual	Automatic	Third Owner	13.1	1997	141.1	5
	Honda City 2017-2020 GXI	2006	200000	178500	Petrol	Individual	Manual	Third Owner	17.7	1497	78	5
	Honda City 1.5 EXI	2002	81000	161000	Petrol	Individual	Manual	Fourth & Above Owner	13	1493	100	5
	Honda Accord 2.4 M/T	2012	600000	144030	Petrol	Individual	Manual	First Owner	12.8	2354	177.6	5
	Honda CR-V MTFC VMT	2016	200000	140000	Petrol	Individual	Manual	First Owner	15.4	1497	117.2	5

cars24 4 x Read On

Output

Action Output

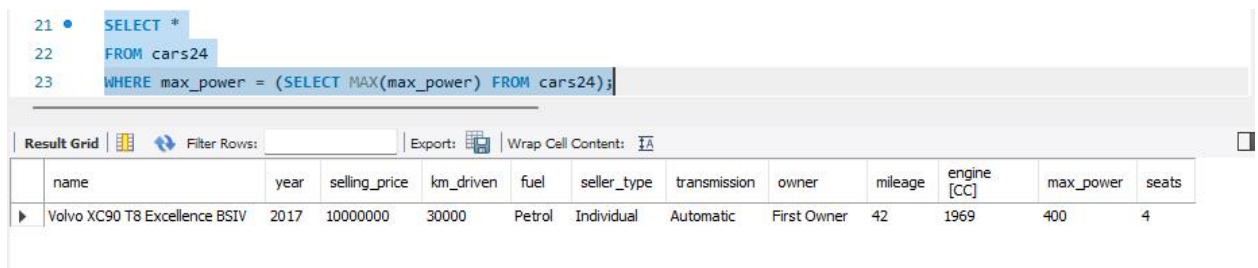
#	Time	Action	Message
✓ 4	17:40:32	SELECT * FROM cars24 WHERE max_power = (SELECT MAX(max_power) FROM cars24) LI...	1 row(s) returned
✓ 5	17:44:11	SELECT * FROM cars24 WHERE name LIKE "%Honda%" LIMIT 0, 1000	467 row(s) returned

2: this query is to find which car has the maximum power

SELECT *

FROM cars24

WHERE max_power = (SELECT MAX(max_power) FROM cars24);



```
21 • SELECT *
22 FROM cars24
23 WHERE max_power = (SELECT MAX(max_power) FROM cars24);
```

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine [CC]	max_power	seats
▶	Volvo XC90 T8 Excellence BSIV	2017	10000000	30000	Petrol	Individual	Automatic	First Owner	42	1969	400	4

3: this query is to find how many different types of sellers and their sales

SELECT seller_type, COUNT(*) AS num_sales

FROM cars24

GROUP BY seller_type;

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

```
9      # we need to find how many sales through dealer and Individual#
10 •   SELECT seller_type, COUNT(*) AS num_sales
11      FROM cars24
12      GROUP BY seller_type;
```

Below the editor is a 'Result Grid' showing the output of the query:

seller_type	num_sales
Individual	6766
Dealer	1126
Trustmark Dealer	236

The interface includes a 'Filter Rows' section, an 'Export' button, and a 'Wrap Cell Content' option.

4: this query is to find how many cars are there in each fuel type

SELECT fuel_type, COUNT(*) AS num_cars

FROM cars24

GROUP BY fuel_type;

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

```
13      # we need to find number of cars of each fuel type
14 •   SELECT fuel, COUNT(*) AS num_cars
15      FROM cars24
16      GROUP BY fuel;
```

Below the editor is a 'Result Grid' showing the output of the query:

fuel	num_cars
Petrol	3631
LPG	38
Diesel	4402
CNG	57

The interface includes a 'Filter Rows' section, an 'Export' button, and a 'Wrap Cell Content' option.

5: this query is to find number of sales in each year

SELECT year, COUNT(*) AS num_sales

FROM cars24

GROUP BY year

ORDER BY num_sales desc;

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

```
5 • SELECT year, COUNT(*) AS num_sales
6 FROM cars24
7 GROUP BY year
8 ORDER BY num_sales desc;
9
10
11
```

Below the editor is a 'Result Grid' showing the results of the query. The grid has two columns: 'year' and 'num_sales'. The data is sorted in descending order of 'num_sales'.

year	num_sales
2017	1018
2016	859
2018	807
2015	776
2013	670
2012	651

The interface includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' option. On the right side, there are buttons for 'Result Grid' and 'Form Editor'. A 'Read Only' status is indicated at the bottom right.

6: this query is to find how many new cars and how many used cars have been sold

SELECT

CASE

WHEN owner = 'First Owner' THEN 'New'

ELSE 'Second Hand'

END AS carsales_status,

COUNT(*) AS num_cars

FROM cars24

GROUP BY carsales_status ;

Query 1 x

```

9      END AS mileage_category,
10     COUNT(*) AS num_cars
11  FROM cars24
12  GROUP BY mileage_category;
13  • SELECT
14     CASE
15         WHEN owner = 'First Owner' THEN 'New'
16         ELSE 'Second Hand'
17     END AS carsales_status,
18     COUNT(*) AS num_cars
19  FROM cars24
20  GROUP BY carsales_status ;

```

Result Grid | Filter Rows: | Exports: | Wrap Cell Content: |

	carsales_status	num_cars
▶	Second Hand	2839
	New	5289

Result 11 x | Read Only

7: this query average selling price per year

SELECT year, AVG(selling_price) AS avg_selling_price

FROM cars24

GROUP BY year

ORDER BY avg_selling_price DESC;

```

35  • SELECT year, AVG(selling_price) AS avg_selling_price
36  FROM cars24
37  GROUP BY year
38  ORDER BY avg_selling_price DESC;

```

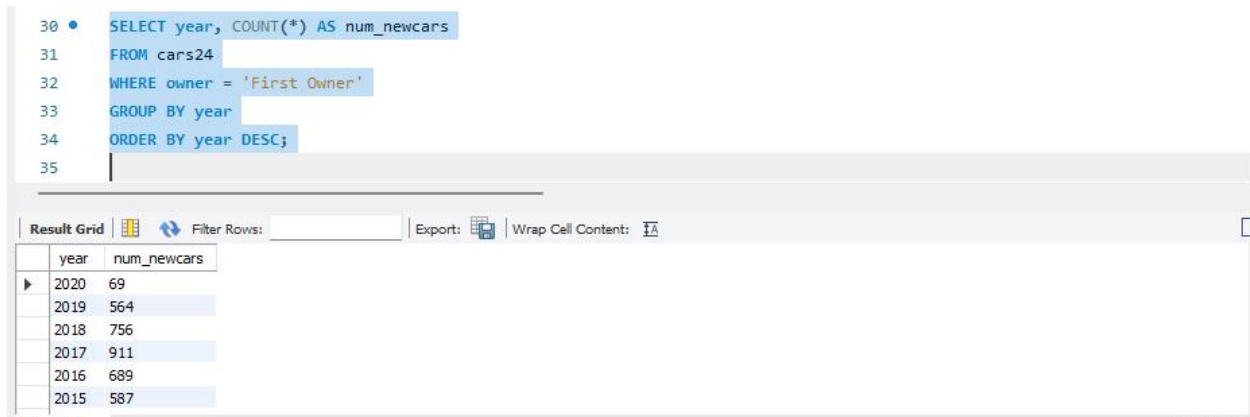
Result Grid | Filter Rows: | Exports: | Wrap Cell Content: |

	year	avg_selling_price
▶	2019	1776986.2504
	2018	957769.4919
	2017	889246.5305
	2020	885270.2297
	2016	699880.0605
	2015	596613.3492

Result 8 x | Read Only

8: this query is to find how many new cars are sold per year

```
SELECT year, COUNT(*) AS num_newcars
FROM cars24
WHERE owner = 'First Owner'
GROUP BY year
ORDER BY year DESC;
```



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

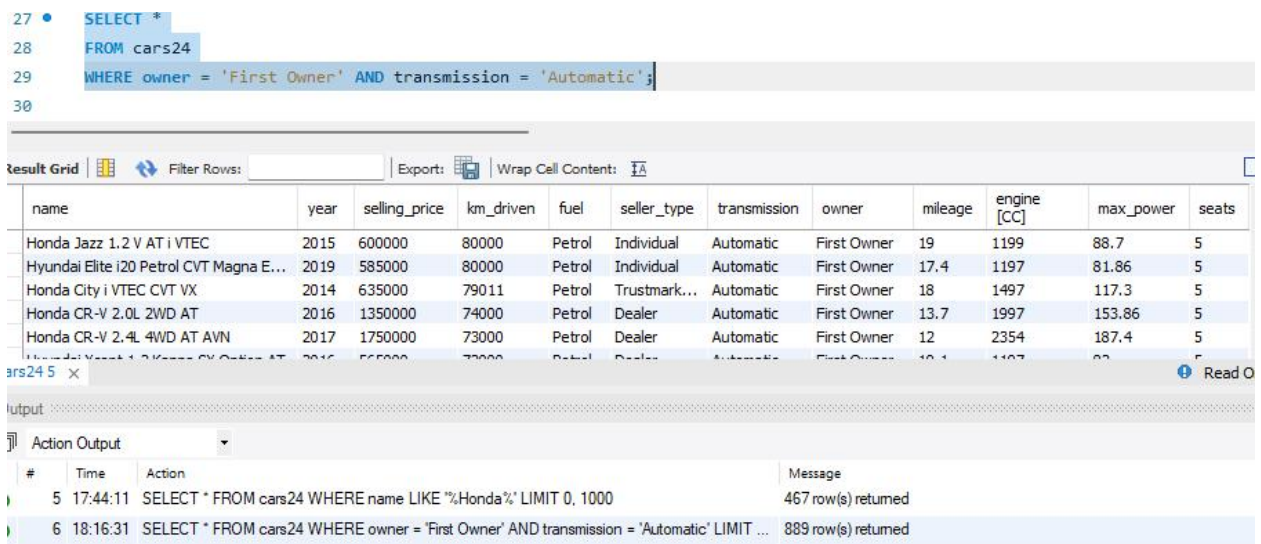
```
30 • SELECT year, COUNT(*) AS num_newcars
31 FROM cars24
32 WHERE owner = 'First Owner'
33 GROUP BY year
34 ORDER BY year DESC;
35
```

Below the query editor is the 'Result Grid' showing the results of the query:

year	num_newcars
2020	69
2019	564
2018	756
2017	911
2016	689
2015	587

9: this query is to find the list of automatic new cars

```
SELECT *
FROM cars24
WHERE owner = 'First Owner' AND transmission = 'Automatic';
```



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

```
27 • SELECT *
28 FROM cars24
29 WHERE owner = 'First Owner' AND transmission = 'Automatic';
30
```

Below the query editor is the 'Result Grid' showing the results of the query:

name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine [CC]	max_power	seats
Honda Jazz 1.2 V AT i VTEC	2015	600000	80000	Petrol	Individual	Automatic	First Owner	19	1199	88.7	5
Hyundai Elite i20 Petrol CVT Magna E...	2019	585000	80000	Petrol	Individual	Automatic	First Owner	17.4	1197	81.86	5
Honda City i VTEC CVT VX	2014	635000	79011	Petrol	Trustmark...	Automatic	First Owner	18	1497	117.3	5
Honda CR-V 2.0L 2WD AT	2016	1350000	74000	Petrol	Dealer	Automatic	First Owner	13.7	1997	153.86	5
Honda CR-V 2.4L 4WD AT AVN	2017	1750000	73000	Petrol	Dealer	Automatic	First Owner	12	2354	187.4	5

Below the result grid is the 'Action Output' section showing the execution of the query:

#	Time	Action	Message
5	17:44:11	SELECT * FROM cars24 WHERE name LIKE "%Honda%" LIMIT 0, 1000	467 row(s) returned
6	18:16:31	SELECT * FROM cars24 WHERE owner = 'First Owner' AND transmission = 'Automatic' LIMIT ...	889 row(s) returned

10: this query is to find how many cars are in different mileage categories

```
select * from cars24;
```

```
SELECT
```

```
    CASE
```

```
        WHEN mileage >= 16 THEN 'High Mileage'
```

```
        WHEN mileage > 11 THEN 'Normal Mileage'
```

```
        WHEN mileage <= 11 THEN 'Low Mileage'
```

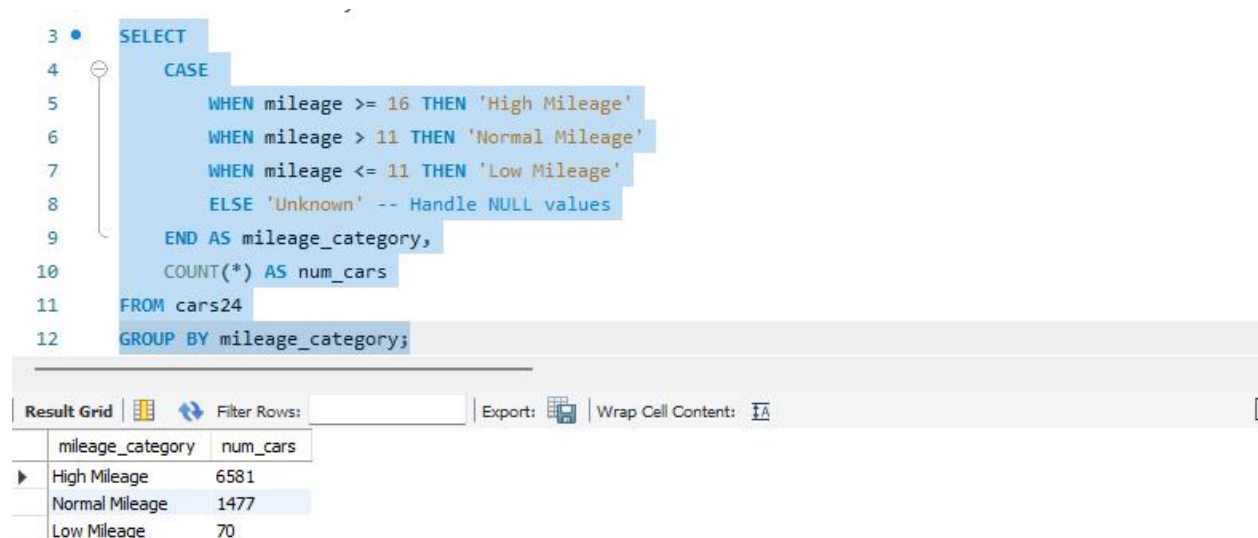
```
        ELSE 'Unknown' -- Handle NULL values
```

```
    END AS mileage_category,
```

```
    COUNT(*) AS num_cars
```

```
FROM cars24
```

```
GROUP BY mileage_category;
```



The screenshot shows a SQL IDE interface. The top pane displays a SQL query with line numbers 3 through 12. The query is a SELECT statement with a CASE expression to categorize mileage into 'High Mileage', 'Normal Mileage', 'Low Mileage', or 'Unknown'. It also counts the number of cars in each category. The bottom pane shows the 'Result Grid' with columns 'mileage_category' and 'num_cars'. The results are as follows:

mileage_category	num_cars
High Mileage	6581
Normal Mileage	1477
Low Mileage	70