```sql
-- First, we will select everything to have a look at how our data looks
select *
from car_sales_project;
```

Results | Messages

	Manufacturer	Model	Unit_Sales	Price	Year_Resale_Value	Retention	Retention_Value	Engine_Size	Horsepower	HP_Level	Vehicl
	Mercury	Mountaineer	27609	27560	20430	74%	GOOD	4	210	Medium HP	Car
	Mercury	Villager	20380	22510	14795	66%	POOR	3.29999995231628	170	Medium HP	Car
	Oldsmobile	Alero	80255	18270	11459	63%	POOR	2.40000009536743	150	Medium HP	Passe
	Saturn	SW	5223	14290	10790	76%	GOOD	1.89999997615814	124	Medium HP	Passe
	Mercedes-B	SLK	7998	38900	32436	83%	GOOD	2.29999995231628	190	Medium HP	Passe
	Nissan	Xterra	54158	22799	13093	57%	POOR	3.29999995231628	170	Medium HP	Car
	Volkswagen	Golf	9761	14900	11425	77%	GOOD	2	115	Medium HP	Passe
	Mercedes-B	M-Class	28976	35300	28720	81%	GOOD	3.20000004768372	215	Medium HP	Car
	Toyota	Camry	247994	17518	13245	76%	GOOD	2.20000004768372	133	Medium HP	Passe
0	Oldsmobile	Aurora	14690	36229	19890	55%	POOR	4	250	Medium HP	Passe
1	Volvo	S40	16957	23400	14334	61%	POOR	1.89999997615814	160	Medium HP	Passe
2	Nissan	Quest	27308	26399	15380	58%	POOR	3.29999995231628	170	Medium HP	Car
3	Toyota	4Runner	68411	22288	19425	87%	GOOD	2.70000004768372	150	Medium HP	Car
4	Acura	RL	8588	42000	29725	71%	GOOD	3.5	210	Medium HP	Passe
5	Saturn	SC	24546	12535	10590	84%	GOOD	1.89999997615814	100	Medium HP	Passe
6	Mercedes-B	SL-Class	3311	82600	58600	71%	GOOD	5	302	High HP	Passe
7	Oldsmobile	Intrigue	38554	24150	17100	71%	GOOD	3.5	215	Medium HP	Passe
8	Volkswagen	GTI	5596	17500	13760	79%	GOOD	2	115	Medium HP	Passe

```sql
-- From above, we see that some columns such as Engine_Size have inconsistent decimal places
--Q2: Write a query to round Engine_size, power_perf_factor, wheelbase, length, width, curb_weight, fuel_capacity to two decimals

SELECT
    ROUND(Engine_Size, 2) AS Engine_Size,
    ROUND(Power_Perf_Factor, 2) AS Power_Perf_Factor,
    ROUND(Wheelbase, 2) AS Wheelbase,
    ROUND(Width, 2) AS Width,
    ROUND(Length, 2) AS Length,
    ROUND(Curb_Weight, 2) AS Curb_Weight,
    ROUND(Fuel_Capacity, 2) AS Fuel_Capacity
FROM car_sales_project;
```

Results | Messages

	Engine_Size	Power_Perf_Factor	Wheelbase	Width	Length	Curb_Weight	Fuel_Capacity
1	4	85.95	111.6	70.2	190.1	3.88	21
2	3.3	69.67	112.2	74.9	194.7	3.94	20
3	2.4	60.73	107	70.1	186.7	2.96	15
4	1.9	49.87	102.4	66.4	176.9	2.45	12.1
5	2.3	82.81	94.5	67.5	157.9	3.06	15.9
6	3.3	69.78	104.3	70.4	178	3.82	19.4
7	2	46.94	98.9	68.3	163.3	2.77	14.5
8	3.2	90.5	111	72.2	180.6	4.39	19
9	2.2	54.37	105.2	70.1	188.5	3	18.5
10	4	103.44	113.8	74.4	205.4	3.97	18.5
11	1.9	66.11	100.5	67.6	176.6	3	15.8
12	3.3	71.17	112.2	74.9	194.8	3.99	20
13	2.7	62.36	105.3	66.5	183.3	3.44	18.5
14	3.5	91.39	114.6	71.4	196.6	3.85	18
15	1.9	40.7	102.4	66.4	180	2.37	12.1
16	5	139.98	99	71.3	177.1	4.13	21.1
17	3.5	86.27	109	73.6	195.9	3.45	18
18	2	47.95	98.9	68.3	163.3	2.76	14.6
19	2.8	84	111.4	70.9	188	3.47	18.5

```sql
-- Q3: We updated the above, however, when we selected all(*) data again, it showed data was unchanged
-- The reason is that this was done for displaying purposes. We can either create a new table with updated da
-- Now, we will directly update the table, but be cautious if you have a real world project, this is irrevers
UPDATE car_sales_project
SET
    Engine_Size = ROUND(Engine_Size, 2),
    Power_Perf_Factor = ROUND(Power_perf_Factor, 2),
    Wheelbase = ROUND(Wheelbase, 2),
    Width = ROUND(Width, 2),
    Length = ROUND(Length, 2),
    Curb_Weight = ROUND(Curb_Weight, 2),
    Fuel_Capacity = ROUND(Fuel_Capacity, 2)
```

100 %

Results | Messages

	_Value	Engine_Size	Horsepower	HP_Level	Vehicle_type	Fuel_Efficiency	Power_Perf_Factor	Wheelbase	Width	Length	Curb_Weight
1		4	210	Medium HP	Car	18	85.95	111.6	70.2	190.1	3.88
2		3.3	170	Medium HP	Car	21	69.67	112.2	74.9	194.7	3.94
3		2.4	150	Medium HP	Passenger	27	60.73	107	70.1	186.7	2.96
4		1.9	124	Medium HP	Passenger	31	49.87	102.4	66.4	176.9	2.45
5		2.3	190	Medium HP	Passenger	26	82.81	94.5	67.5	157.9	3.06
6		3.3	170	Medium HP	Car	18	69.78	104.3	70.4	178	3.82
7		2	115	Medium HP	Passenger	26	46.94	98.9	68.3	163.3	2.77
8		3.2	215	Medium HP	Car	20	90.5	111	72.2	180.6	4.39
9		2.2	133	Medium HP	Passenger	27	54.37	105.2	70.1	188.5	3
10		4	250	Medium HP	Passenger	22	103.44	113.8	74.4	205.4	3.97
11		1.9	160	Medium HP	Passenger	25	66.11	100.5	67.6	176.6	3
12		3.3	170	Medium HP	Car	21	71.17	112.2	74.9	194.8	3.99
13		2.7	150	Medium HP	Car	23	62.36	105.3	66.5	183.3	3.44

```sql
--Q4: Write a query to check for null values in column
SELECT *
FROM car_sales_project
WHERE Manufacturer IS NULL OR Model IS NULL OR Unit_Sales IS NULL OR Price IS NULL OR Year_Resale_Value IS NULL
OR Retention IS NULL OR Retention_Value IS NULL OR Engine_Size IS NULL OR Horsepower IS NULL
OR HP_Level IS NULL OR Vehicle_type IS NULL OR Fuel_Efficiency IS NULL OR Power_Perf_Factor IS NULL
OR Wheelbase IS NULL OR Width IS NULL OR Length IS NULL OR Curb_Weight IS NULL
OR Fuel_Capacity IS NULL OR Latest_Launch IS NULL;
```

100 %

Results | Messages

Manufacturer	Model	Unit_Sales	Price	Year_Resale_Value	Retention	Retention_Value	Engine_Size	Horsepower	HP_Level	Vehicle_type	Fuel_Efficiency	Power

```sql
-- Q5: Qrite a query to return not only null, but also EMPTY and SPACES in the column Manufacturer
-- ISNULL() function will return any spaces, empty or null values
SELECT Manufacturer FROM car_sales_project
WHERE ISNULL(Manufacturer, '') = ''
```

100 %

Results | Messages

Manufacturer

```sql
-- Q6: Write a query to check for the length of data in a specified column
-- the function will return any empty rows with no data

SELECT * FROM car_sales_project
WHERE DATALENGTH(Model) = 0
```

100 %

Results | Messages

Manufacturer	Model	Unit_Sales	Price	Year_Resale_Value	Retention	Retention_Value	Engine_Size	Horsepower	HP_Level	Vehicle_type	Fuel_Efficiency	Power

```sql
--Q7: write a query to find any possible ZEROEs (0) in the columns manufacturer, model, unit_sales and Price
-- Note: For manufacturer and model, we use "LIKE" since these are varchar and not int
SELECT Manufacturer, Model, Unit_Sales, Price
FROM car_sales_project
WHERE Manufacturer LIKE '0' OR Model LIKE '0' OR Unit_Sales = 0 OR Price = 0;
```

100 %

Results | Messages

Manufacturer	Model	Unit_Sales	Price

```sql
--Q8: write a new query to create new table customer_info with columns: customer_id, firt_name, last_name, city, state

CREATE TABLE customer_info (
customer_id INT PRIMARY KEY,
first_name VARCHAR(50),
last_name VARCHAR(50),
city VARCHAR(50),
state VARCHAR(50)
);
```

100 %

Messages

```
Commands completed successfully.

Completion time: 2024-09-17T15:16:35.1921600-05:00
```

100 %

```sql
--Q9: Write a query to add some values to the table we created above, customer_info
-- Note: we can write more values as long we have right syntax and structure
INSERT INTO customer_info
(customer_id, first_name, last_name, city, state)
VALUES
(1, 'Mike', 'Sema', 'Dallas','Texas'),
(2, 'Mira','Hiba','Sinjar','Ninevah');
```

100 %

Messages

```
(2 rows affected)

Completion time: 2024-09-17T15:28:45.9559764-05:00
```

100 %

```sql
-- 10: Write a query to update values in customer_info table, 'Mike', 'Sema' to 'Mola', 'Hamida'

UPDATE customer_info
SET first_name = 'Mola', last_name = 'Hamida'
WHERE first_name = 'Mike' AND last_name = 'Sema';
```

100 %

Messages

```
(1 row affected)

Completion time: 2024-09-17T15:36:47.2350231-05:00
```

100 %

```sql
-- Q11: write a query to selecta and retrieve everything in the new table we created, customer_info

SELECT * FROM customer_info
```

100 %

Results | Messages

	customer_id	first_name	last_name	city	state
1	1	Mola	Hamida	Dallas	Texas
2	2	Mira	Hiba	Sinjar	Ninevah

Query executed successfully. | DESKTOP-MPNEKS0\SQLEXPRESS ... | DESKTOP-MPNEKS0\Jallal... | car_sales | 00:00:00 | 2 rows

```sql
--Q12: The table above, customer_info, we created it for practice purposes, we will now write queries to delete it
-- write a query to delete column first_name


ALTER TABLE customer_info
DROP COLUMN first_name
```

100 %

Messages

```
Commands completed successfully.

Completion time: 2024-09-17T17:22:46.7319823-05:00
```

100 %

```sql
-- Q13: write a query to delete table customer_info

DROP TABLE customer_info
```

100 %

Messages

```
Commands completed successfully.

Completion time: 2024-09-17T17:25:52.9282570-05:00
```

100 %

```sql
-- Q14: Write a query to rerieve info from columns: manufacturer, model, engine_size and retention with no repeated values
SELECT DISTINCT Manufacturer, Model, Engine_Size, Retention

FROM car_sales_project;
```

100 %

Results | Messages

	Manufacturer	Model	Engine_Size	Retention
1	Acura	Integra	1.8	76%
2	Acura	RL	3.5	71%
3	Acura	TL	3.2	70%
4	Audi	A4	1.8	93%
5	Audi	A6	2.8	69%
6	Audi	A8	4.2	63%
7	BMW	323i	2.5	91%
8	BMW	328i	2.8	86%
9	BMW	528i	2.8	93%
10	Buick	Century	3.1	57%
11	Buick	LeSabre	3.8	48%
12	Buick	Park Avenue	3.8	63%

```sql
-- Q15: write a query to retrieve only BMW and retention over 70%, limit results to 8
-- Note: limit is NOT supported in standard SQL SERVER, so we will use TOP AND ORDER BY

SELECT TOP 8 *
FROM car_sales_project
WHERE Manufacturer = 'BMW' AND Retention > '70%'
ORDER BY Retention DESC;
```

100 %

Results | Messages

	Manufacturer	Model	Unit_Sales	Price	Year_Resale_Value	Retention	Retention_Value	Engine_Size	Horsepower	HP_Level	Vehicle_type	Fuel_Efficier
1	BMW	528i	17527	38900	36125	93%	GOOD	2.8	193	Medium HP	Passenger	25
2	BMW	323i	19747	26990	24500	91%	GOOD	2.5	170	Medium HP	Passenger	26
3	BMW	328i	9231	33400	28675	86%	GOOD	2.8	193	Medium HP	Passenger	24

```sql
-- Q16: Write a query to retrieve data where unit sales are at least 20000, retention_value is Good and Engine_size is at least 1
SELECT DISTINCT Unit_Sales, Retention_Value, Engine_Size
FROM car_sales_project
WHERE Unit_Sales >= 20000 AND Retention_Value = 'GOOD' AND Engine_Size >= 1
ORDER BY Engine_Size;
```

100 %

Results | Messages

	Unit_Sales	Retention_Value	Engine_Size
1	199685	GOOD	1.6
2	20397	GOOD	1.8
3	33269	GOOD	1.8
4	51102	GOOD	1.8
5	142535	GOOD	1.8
6	24546	GOOD	1.9
7	80620	GOOD	1.9
8	25106	GOOD	2
9	26529	GOOD	2
10	32306	GOOD	2
11	49463	GOOD	2

```sql
-- Q17: write a query to return data of cars where horsepower is at least 210, vehicle type is car and fuel ifficiency is at least 15

SELECT Horsepower, Vehicle_type, Fuel_Efficiency
FROM car_sales_project
WHERE Horsepower >= 210 AND Vehicle_type = 'Car' AND Fuel_Efficiency >= 15
ORDER BY Fuel_Efficiency;
```

100 %

Results | Messages

	Horsepower	Vehicle_type	Fuel_Efficiency
1	230	Car	15
2	255	Car	15
3	230	Car	15
4	300	Car	15
5	240	Car	16
6	230	Car	17
7	230	Car	17
8	220	Car	18
9	210	Car	18
10	210	Car	19
11	215	Car	20

```sql
-- Q18: write a query to get data of cars as follow: retention value is poor and hp level is at least medium
SELECT Retention_Value, HP_Level
FROM car_sales_project
WHERE Retention_Value = 'POOR' AND (HP_Level = 'Medium HP' OR HP_Level = 'High HP');
```

100 %

Results | Messages

	Retention_Value	HP_Level
1	POOR	Medium HP
2	POOR	Medium HP
3	POOR	Medium HP
4	POOR	Medium HP
5	POOR	Medium HP
6	POOR	Medium HP
7	POOR	Medium HP
8	POOR	Medium HP
9	POOR	Medium HP
10	POOR	Medium HP
11	POOR	Medium HP
12	POOR	Medium HP
13	POOR	Medium HP
14	POOR	Medium HP

```sql
-- Q19: write a query to return all car details but only when after date 12/30/2011

SELECT *
FROM car_sales_project
WHERE Latest_Launch > '2011-12-30';
```

100 %

Results | Messages

	Manufacturer	Model	Unit_Sales	Price	Year_Resale_Value	Retention	Retention_Value	Engine_Size	Horsepower	HP_Level	Vehicle_type	Fuel_Efficiency	Power_F
1	Lexus	RX300	51238	34605	29140	84%	GOOD	3	220	Medium HP	Car	21	91.94
2	Dodge	Ram Wagon	16767	21315	15510	73%	GOOD	3.9	175	Medium HP	Car	15	71.14
3	Ford	Ranger	220650	12050	7850	65%	POOR	2.5	119	Medium HP	Car	23	47.39
4	Chrysler	Sebring Coupe	7854	19840	12360	62%	POOR	2.5	163	Medium HP	Passenger	24	65.96
5	Mitsubishi	Montero	11337	31807	19125	60%	POOR	3.5	200	Medium HP	Car	18	83.92
6	Pontiac	Sunfire	51645	21610	13790	64%	POOR	2.4	150	Medium HP	Passenger	27	62.02
7	BMW	328i	9231	33400	28675	86%	GOOD	2.8	193	Medium HP	Passenger	24	81.88
8	Mitsubishi	Galant	55616	17357	10595	61%	POOR	2.4	145	Medium HP	Passenger	25	58.61
9	Ford	Mustang	113369	21560	12760	59%	POOR	3.8	190	Medium HP	Passenger	24	76.51
10	Acura	Integra	16919	21500	16360	76%	GOOD	1.8	140	Medium HP	Passenger	28	58.28
11	Honda	Odyssey	76029	26000	19490	75%	GOOD	3.5	210	Medium HP	Car	23	85.22
12	Chrysler	300M	30696	29185	18425	63%	POOR	3.5	253	Medium HP	Passenger	23	101.66
13	Porsche	Boxter	8982	41430	41250	100%	GOOD	2.7	217	Medium HP	Passenger	22	93.44

```sql
--Q20: Write a query to show all car data where year_resale_value is more than 26000 and vehicle type is car

SELECT *
FROM car_sales_project
WHERE Year_Resale_Value > 26000 AND Vehicle_type = 'Car';
```

100 %

Results | Messages

	Manufacturer	Model	Unit_Sales	Price	Year_Resale_Value	Retention	Retention_Value	Engine_Size	Horsepower	HP_Level	Vehicle_type	Fuel_Efficiency	Power_Perf_F
1	Mercedes-B	M-Class	28976	35300	28720	81%	GOOD	3.2	215	Medium HP	Car	20	90.5
2	Toyota	Land Cruiser	9835	51728	34080	66%	POOR	4.7	230	Medium HP	Car	15	102.53
3	Lexus	RX300	51238	34605	29140	84%	GOOD	3	220	Medium HP	Car	21	91.94
4	Cadillac	Escalade	14785	46225	31245	68%	POOR	5.7	255	Medium HP	Car	15	109.51
5	Lexus	LX470	9126	60105	43870	73%	GOOD	4.7	230	Medium HP	Car	15	105.76

Query executed successfully. | DESKTOP-MPNEKS0\SQLEXPRESS ... | DESKTOP-MPNEKS0\Jallal... | car_sales | 00:00:00 | 5 rows

```sql
--Q21: write a query to show all car details where Model is M-Class or RX300

SELECT *
FROM car_sales_project
WHERE Model = 'M-Class' OR Model = 'RX300';
```

100 %

Results | Messages

	Manufacturer	Model	Unit_Sales	Price	Year_Resale_Value	Retention	Retention_Value	Engine_Size	Horsepower	HP_Level	Vehicle_type	Fuel_Efficiency	Power_Perf_Fact
1	Mercedes-B	M-Class	28976	35300	28720	81%	GOOD	3.2	215	Medium HP	Car	20	90.5
2	Lexus	RX300	51238	34605	29140	84%	GOOD	3	220	Medium HP	Car	21	91.94

```sql
--Q22: write a query to retrieve info where Power factor is at least 70 and wheelbase is at least 100

SELECT *
FROM car_sales_project
WHERE Power_Perf_Factor >= 70 AND Wheelbase >= 100;
```

100 %

Results | Messages

	Manufacturer	Model	Unit_Sales	Price	Year_Resale_Value	Retention	Retention_Value	Engine_Size	Horsepower	HP_Level	Vehicle_typ
1	Mercury	Mountaineer	27609	27560	20430	74%	GOOD	4	210	Medium HP	Car
2	Mercedes-B	M-Class	28976	35300	28720	81%	GOOD	3.2	215	Medium HP	Car
3	Oldsmobile	Aurora	14690	36229	19890	55%	POOR	4	250	Medium HP	Passenger
4	Nissan	Quest	27308	26399	15380	58%	POOR	3.3	170	Medium HP	Car
5	Acura	RL	8588	42000	29725	71%	GOOD	3.5	210	Medium HP	Passenger
6	Oldsmobile	Intrigue	38554	24150	17100	71%	GOOD	3.5	215	Medium HP	Passenger
7	BMW	528i	17527	38900	36125	93%	GOOD	2.8	193	Medium HP	Passenger
8	Mercedes-B	CL500	954	85500	63423	74%	GOOD	5	302	High HP	Passenger
9	Mercedes-B	C-Class	18392	31750	26050	82%	GOOD	2.3	185	Medium HP	Passenger
10	Volvo	C70	3493	45500	34175	75%	GOOD	2.3	236	Medium HP	Passenger
11	Cadillac	Seville	15943	44475	27100	61%	POOR	4.6	275	Medium HP	Passenger
12	Nissan	Maxima	79853	26249	15125	58%	POOR	3	222	Medium HP	Passenger
13	Pontiac	Bonneville	35945	23755	13225	56%	POOR	3.8	205	Medium HP	Passenger

```sql
--Q23: Write a query to return car data but only of models Quest, RL and C70

SELECT *
FROM car_sales_project
WHERE Model = 'Quest' OR Model = 'RL' OR Model = 'C70';
```

100 %

Results | Messages

	Manufacturer	Model	Unit_Sales	Price	Year_Resale_Value	Retention	Retention_Value	Engine_Size	Horsepower	HP_Level	Vehicle_type	Fuel_
1	Nissan	Quest	27308	26399	15380	58%	POOR	3.3	170	Medium HP	Car	21
2	Acura	RL	8588	42000	29725	71%	GOOD	3.5	210	Medium HP	Passenger	22
3	Volvo	C70	3493	45500	34175	75%	GOOD	2.3	236	Medium HP	Passenger	23

```sql
--Q24: write a query and show data of cars as follow: wheel at least 80, Width at least 65 and Length more than 160

SELECT *
FROM car_sales_project WHERE Wheelbase >= 80 AND Width >= 65 AND Length > 160;
```

100 %

Results | Messages

	orsepower	HP_Level	Vehicle_type	Fuel_Efficiency	Power_Perf_Factor	Wheelbase	Width	Length	Curb_Weight	Fuel_Capacity	Latest_Launch
1	10	Medium HP	Car	18	85.95	111.6	70.2	190.1	3.88	21	2008-02-13
2	70	Medium HP	Car	21	69.67	112.2	74.9	194.7	3.94	20	2009-10-20
3	50	Medium HP	Passenger	27	60.73	107	70.1	186.7	2.96	15	2009-10-20
4	24	Medium HP	Passenger	31	49.87	102.4	66.4	176.9	2.45	12.1	2011-01-15
5	70	Medium HP	Car	18	69.78	104.3	70.4	178	3.82	19.4	2011-01-24
6	15	Medium HP	Passenger	26	46.94	98.9	68.3	163.3	2.77	14.5	2011-01-24
7	15	Medium HP	Car	20	90.5	111	72.2	180.6	4.39	19	2011-02-10
8	33	Medium HP	Passenger	27	54.37	105.2	70.1	188.5	3	18.5	2011-02-10
9	50	Medium HP	Passenger	22	103.44	113.8	74.4	205.4	3.97	18.5	2011-02-18
10	30	Medium HP	Passenger	25	66.11	100.5	67.6	176.6	3	15.8	2011-02-18
11	70	Medium HP	Car	21	71.17	112.2	74.9	194.8	3.99	20	2011-03-07
12	50	Medium HP	Car	23	62.36	105.3	66.5	183.3	3.44	18.5	2011-03-07
13	10	Medium HP	Passenger	22	91.39	114.6	71.4	196.6	3.85	18	2011-03-10

```sql
--Q25: Write a query to show data of Curb weight of cars is less than 3 and fuel capacity of cars is more than 16

SELECT *
FROM car_sales_project
WHERE Curb_Weight < 3 AND Fuel_Capacity > 16;
```

100 %

Results | Messages

	Manufacturer	Model	Unit_Sales	Price	Year_Resale_Value	Retention	Retention_Value	Engine_Size	Horsepower	HP_Level	Vehicle_type	Fue
1	Saab	3-Sep	12115	26100	18255	70%	GOOD	2	185	Medium HP	Passenger	23
2	Mitsubishi	Galant	55616	17357	10595	61%	POOR	2.4	145	Medium HP	Passenger	25
3	Porsche	Boxter	8982	41430	41250	100%	GOOD	2.7	217	Medium HP	Passenger	22
4	Honda	Accord	230902	15350	13210	86%	GOOD	2.3	135	Medium HP	Passenger	27