

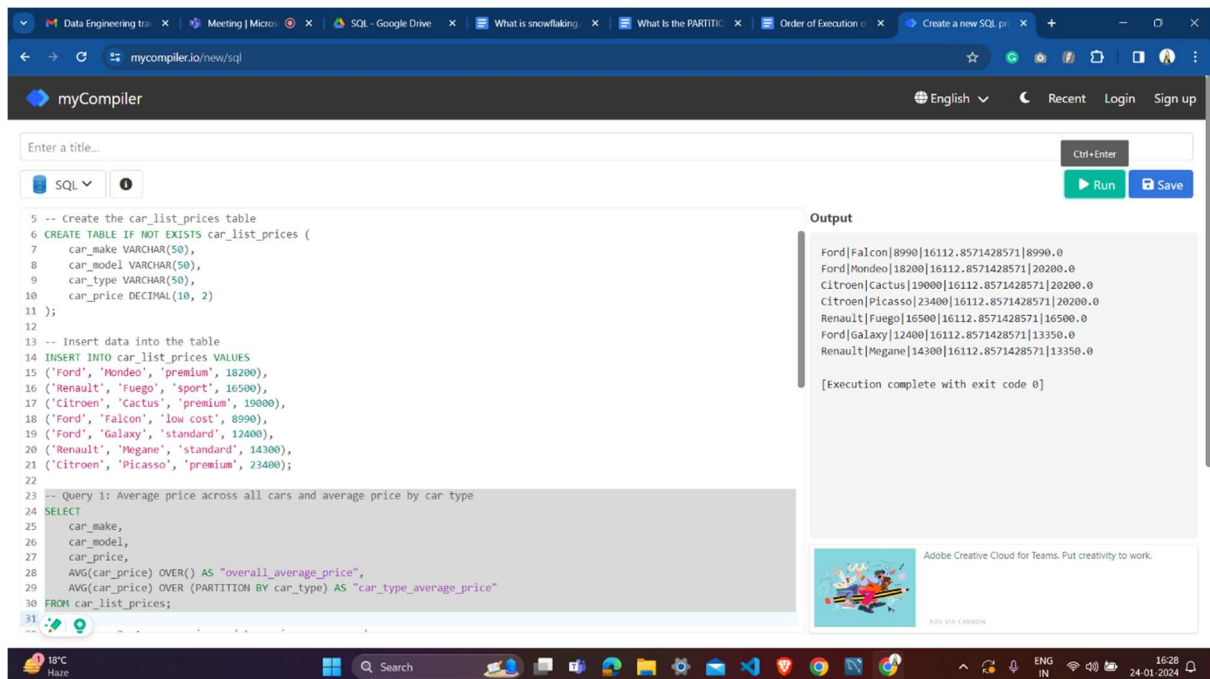
Name: Abhishek Kanoujia

DATA ENGINEERING BATCH 1

DAY 6 ASSIGNMENT

What Is the PARTITION BY Clause in SQL?

The SQL `PARTITION BY` expression is a subclause of the `OVER` clause, which is used in almost all invocations of window functions like `AVG()`, `MAX()`, and `RANK()`. As many readers probably know, window functions operate on window frames which are sets of rows that can be different for each record in the query result. This is where the SQL `PARTITION BY` subclause comes in: it is used to define which records to make part of the window frame associated with each record of the result.



The screenshot shows a web browser window with the URL `mycompiler.io/new/sql`. The page has a dark theme and a header with the 'myCompiler' logo and navigation links. The main area contains a SQL editor with the following code:

```
5 -- Create the car_list_prices table
6 CREATE TABLE IF NOT EXISTS car_list_prices (
7   car_make VARCHAR(50),
8   car_model VARCHAR(50),
9   car_type VARCHAR(50),
10  car_price DECIMAL(10, 2)
11 );
12
13 -- Insert data into the table
14 INSERT INTO car_list_prices VALUES
15 ('Ford', 'Mondeo', 'premium', 18200),
16 ('Renault', 'Fuego', 'sport', 16500),
17 ('Citroen', 'Cactus', 'premium', 19000),
18 ('Ford', 'Falcon', 'low cost', 8990),
19 ('Ford', 'Galaxy', 'standard', 12400),
20 ('Renault', 'Megane', 'standard', 14300),
21 ('Citroen', 'Picasso', 'premium', 23400);
22
23 -- Query 1: Average price across all cars and average price by car type
24 SELECT
25   car_make,
26   car_model,
27   car_price,
28   AVG(car_price) OVER() AS "overall_average_price",
29   AVG(car_price) OVER (PARTITION BY car_type) AS "car_type_average_price"
30 FROM car_list_prices;
```

The output of the query is displayed on the right side of the editor:

```
Ford|Falcon|8990|16112.8571428571|8990.0
Ford|Mondeo|18200|16112.8571428571|20200.0
Citroen|Cactus|19000|16112.8571428571|20200.0
Citroen|Picasso|23400|16112.8571428571|20200.0
Renault|Fuego|16500|16112.8571428571|16500.0
Ford|Galaxy|12400|16112.8571428571|13350.0
Renault|Megane|14300|16112.8571428571|13350.0
```

Below the output, it states "[Execution complete with exit code 0]". At the bottom of the page, there is a Windows taskbar showing the time as 16:28 on 24-01-2024.

myCompiler

Enter a title... Ctrl+Enter

SQL ?

Run Save

```
27 -- car_price,
28 -- AVG(car_price) OVER() AS "overall_average_price",
29 -- AVG(car_price) OVER (PARTITION BY car_type) AS "car_type_average_price"
30 -- FROM car_list_prices;
31
32
33
34
35
36 -- Query 2: Average price and top price per car make
37 SELECT
38     car_make,
39     AVG(car_price) AS average_price,
40     MAX(car_price) AS top_price
41 FROM car_list_prices
42 GROUP BY car_make;
```

Output

```
Citroen|21200.0|23400
Ford|13196.6666666667|18200
Renault|15400.0|16500

[Execution complete with exit code 0]
```

Adobe Creative Cloud for Teams. Put creativity to work.

myCompiler

Enter a title... Ctrl+Enter

SQL ?

Run Save

```
427
50
51
52
53
54
55
56 -- Query 3: Average make price using window functions
57 SELECT
58     car_make,
59     car_model,
60     car_price,
61     AVG(car_price) OVER (PARTITION BY car_make) AS average_make
62 FROM car_list_prices;
63
64
65
66
67
68
69
70 -- Query 4: Monthly variation in car prices
71 -- WITH year_month_data AS (
72 --     SELECT DISTINCT
73 --         strftime('%Y', 'now') AS year,
74 --         strftime('%m', 'now') AS month,
75 --         AVG(car_price) AS average_price
76 --     FROM car_list_prices
```

Output

```
Citroen|Cactus|19000|21200.0
Citroen|Picasso|23400|21200.0
Ford|Mondeo|18200|13196.6666666667
Ford|Falcon|8990|13196.6666666667
Ford|Galaxy|12400|13196.6666666667
Renault|Fuego|16500|15400.0
Renault|Megane|14300|15400.0

[Execution complete with exit code 0]
```

Adobe Creative Cloud for Teams. Put creativity to work.

myCompiler

English Recent Login Sign up

Enter a title...

Ctrl+Enter

SQL

61 -- AVG(car_price) OVER (PARTITION BY car_make) AS average_make
62 -- FROM car_list_prices;
63
64
65
66
67
68
69
70 -- Query 4: Monthly variation in car prices
71 WITH year_month_data AS (
72 SELECT DISTINCT
73 strftime('%Y', 'now') AS year,
74 strftime('%m', 'now') AS month,
75 AVG(car_price) AS average_price
76 FROM car_list_prices
77 GROUP BY 1, 2
78)
79 SELECT year,
80 month,
81 average_price,
82 LAG(average_price) OVER (ORDER BY year, month) AS avg_price_previous_month,
83 average_price - LAG(average_price) OVER (ORDER BY year, month) AS avg_price_delta
84 FROM year_month_data;
85
86
87
88

Output
2024|01|16112.8571428571|

[Execution complete with exit code 0]

myCompiler

English Recent Login Sign up

Enter a title...

Ctrl+Enter

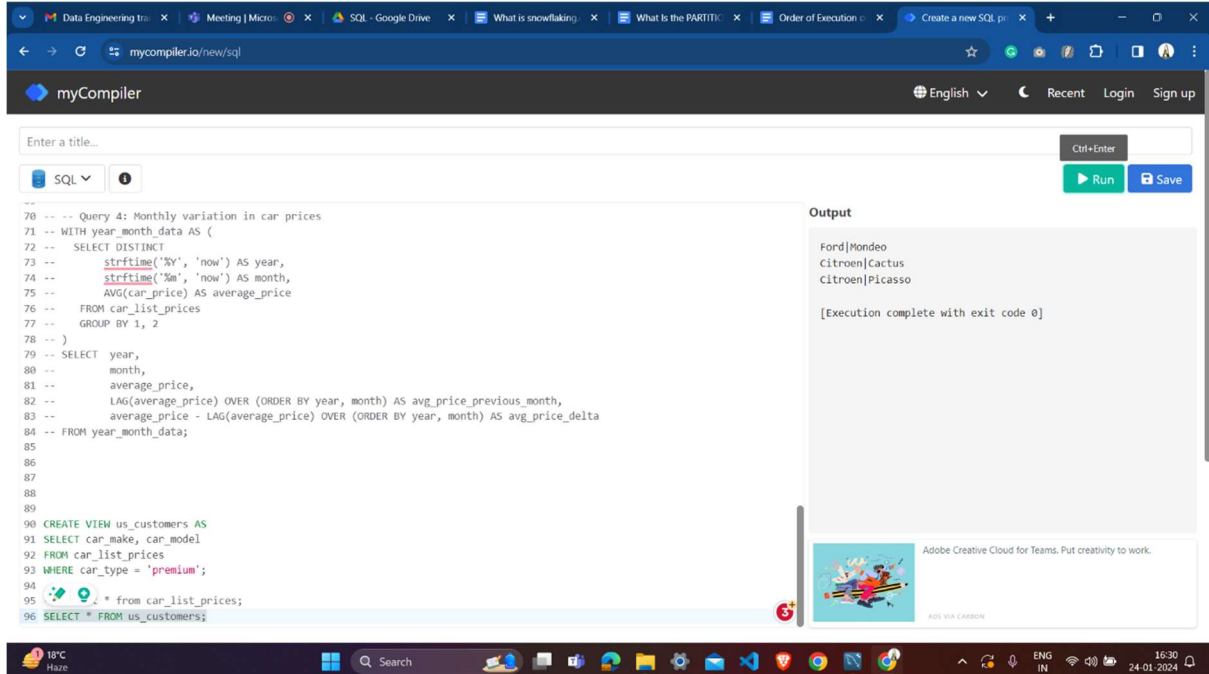
SQL

70 -- Query 4: Monthly variation in car prices
71 -- WITH year_month_data AS (
72 -- SELECT DISTINCT
73 -- strftime('%Y', 'now') AS year,
74 -- strftime('%m', 'now') AS month,
75 -- AVG(car_price) AS average_price
76 -- FROM car_list_prices
77 -- GROUP BY 1, 2
78 --)
79 -- SELECT year,
80 -- month,
81 -- average_price,
82 -- LAG(average_price) OVER (ORDER BY year, month) AS avg_price_previous_month,
83 -- average_price - LAG(average_price) OVER (ORDER BY year, month) AS avg_price_delta
84 -- FROM year_month_data;
85
86
87
88
89
90 CREATE VIEW us_customers AS
91 SELECT car_make, car_model
92 FROM car_list_prices
93 WHERE car_type = 'premium';
94
95 select * from car_list_prices;
96
97 * FROM us_customers;

Output
Ford|Mondeo|premium|18200
Renault|Fuego|sport|16500
Citroen|Cactus|premium|19000
Ford|Falcon|low cost|8990
Ford|Galaxy|standard|12400
Renault|Megane|standard|14300
Citroen|Picasso|premium|23400

[Execution complete with exit code 0]

Material View in SQL syntax and example:



The screenshot shows the myCompiler.io web interface. The browser address bar displays `mycompiler.io/new/sql`. The page header includes the myCompiler logo, a language dropdown set to English, and links for Recent, Login, and Sign up. Below the header, there is a text input field for a title and buttons for Ctrl+Enter, Run, and Save. The main area is split into two panels. The left panel contains SQL code, and the right panel shows the output.

SQL Code:

```
-- Query 4: Monthly variation in car prices
-- WITH year_month_data AS (
--   SELECT DISTINCT
--     strftime('%Y', 'now') AS year,
--     strftime('%m', 'now') AS month,
--     AVG(car_price) AS average_price
--   FROM car_list_prices
--   GROUP BY 1, 2
-- )
-- SELECT year,
--        month,
--        average_price,
--        LAG(average_price) OVER (ORDER BY year, month) AS avg_price_previous_month,
--        average_price - LAG(average_price) OVER (ORDER BY year, month) AS avg_price_delta
-- FROM year_month_data;

CREATE VIEW us_customers AS
SELECT car_make, car_model
FROM car_list_prices
WHERE car_type = 'premium';

SELECT * FROM us_customers;
```

Output:

```
Ford|Mondeo
Citroen|Cactus
Citroen|Picasso

[Execution complete with exit code 0]
```

At the bottom of the interface, there is a Windows taskbar showing the system clock as 18:30 on 24-01-2024, and a temperature of 18°C with a haze condition.