

Create database sales12

```
CREATE TABLE sales12 (  
    order_id INT PRIMARY KEY,  
    order_date DATE,  
    amount DECIMAL(10,2),  
    product_id INT  
);
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(1, '2023-01-05',  
250.50, 3)
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(2, '2023-01-15',  
120.00, 5)
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(3, '2023-02-02',  
450.75, 2)
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(4, '2023-02-18',  
300.40, 7)
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(5, '2023-03-03',  
99.99, 1)
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(26, '2023-03-20',  
175.25, 4)
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(7, '2023-04-10',  
510.60, 8)
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(8, '2023-04-22',  
200.00, 2)
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(9, '2023-05-05',  
320.10, 6)
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(10, '2023-05-28',  
410.75, 9)
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(27, '2024-06-07',  
150.30, 1)
```

```
INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(12, '2023-06-18',  
220.45, 3)
```

```

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(13,'2023-07-09',
499.99, 7)

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(14,'2023-07-21',
275.00, 5)

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(35,'2024-08-12',
600.80, 10)

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(16,'2023-08-25',
340.25, 2)

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(37,'2023-09-06',
190.75, 4)

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(18,'2024-09-27',
420.00, 8)

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(19,'2025-10-11',
510.20, 6)

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(20,'2025-10-29',
289.99, 9)

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(21,'2025-11-14',
135.50, 5)

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(22,'2024-11-25',
455.75, 3)

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(23,'2022-12-05',
610.40, 7)

INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(24,'2023-12-18',
275.60, 1)

```

Query History	
8	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(1, '2023-01-05', 250.50, 3)
9	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(2, '2023-01-15', 120.00, 5)
10	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(3, '2023-02-02', 450.75, 2)
11	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(4, '2023-02-18', 300.40, 7)
12	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(5, '2023-03-03', 99.99, 1)
13	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(26, '2023-03-20', 175.25, 4)
14	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(7, '2023-04-10', 510.60, 8)
15	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(8, '2023-04-22', 200.00, 2)
16	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(9, '2023-05-05', 320.10, 6)
17	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(10, '2023-05-28', 410.75, 9)
18	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(27, '2024-06-07', 150.30, 1)
19	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(12, '2023-06-18', 220.45, 3)
20	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(13, '2023-07-09', 499.99, 7)
21	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(14, '2023-07-21', 275.00, 5)
22	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(35, '2024-08-12', 600.80, 10)
23	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(16, '2023-08-25', 340.25, 2)
24	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(37, '2023-09-06', 190.75, 4)
25	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(18, '2024-09-27', 420.00, 8)
26	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(19, '2025-10-11', 510.20, 6)
27	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(20, '2025-10-29', 289.99, 9)
28	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(21, '2025-11-14', 135.50, 5)
29	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(22, '2024-11-25', 455.75, 3)
30	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(23, '2022-12-05', 610.40, 7)
31	INSERT INTO sales12 (order_id, order_date, amount, product_id) VALUES(24, '2023-12-18', 275.60, 1)

```
select * from sales12
```

SQL

Data Output Messages Notifications

Showing rows: 1 to 24 Page No: 1 of 1

	order_id [PK] integer	order_date date	amount numeric (10,2)	product_id integer
1	1	2023-01-05	250.50	3
2	2	2023-01-15	120.00	5
3	3	2023-02-02	450.75	2
4	4	2023-02-18	300.40	7
5	5	2023-03-03	99.99	1
6	26	2023-03-20	175.25	4
7	7	2023-04-10	510.60	8
8	8	2023-04-22	200.00	2
9	9	2023-05-05	320.10	6
10	10	2023-05-28	410.75	9
11	27	2024-06-07	150.30	1
12	12	2023-06-18	220.45	3
13	13	2023-07-09	499.99	7
14	14	2023-07-21	275.00	5
15	35	2024-08-12	600.80	10
16	16	2023-08-25	340.25	2
Total rows: 24	Query complete 00:00:00.203			CRLF Ln 32, Col :

SELECT

EXTRACT(YEAR FROM order\_date) AS order\_year,

EXTRACT(MONTH FROM order\_date) AS order\_month,

SUM(amount) AS total\_revenue,

COUNT(DISTINCT order\_id) AS total\_orders

FROM sales12

GROUP BY EXTRACT(YEAR FROM order\_date), EXTRACT(MONTH FROM order\_date)

ORDER BY order\_year, order\_month;

Data Output Messages Notifications

SQL

	order_id [PK] integer	order_date date	amount numeric (10,2)	product_id integer
1	1	2023-01-05	250.50	3
2	2	2023-01-15	120.00	5
3	3	2023-02-02	450.75	2
4	4	2023-02-18	300.40	7
5	5	2023-03-03	99.99	1
6	26	2023-03-20	175.25	4
7	7	2023-04-10	510.60	8
8	8	2023-04-22	200.00	2
9	9	2023-05-05	320.10	6
10	10	2023-05-28	410.75	9
11	27	2024-06-07	150.30	1
12	12	2023-06-18	220.45	3
13	13	2023-07-09	499.99	7
14	14	2023-07-21	275.00	5
15	35	2024-08-12	600.80	10
16	16	2023-08-25	340.25	2
Total rows: 24	Query complete 00:00:00.147			

SELECT

EXTRACT(YEAR FROM order\_date) AS order\_year,

EXTRACT(MONTH FROM order\_date) AS order\_month,

SUM(amount) AS total\_revenue,

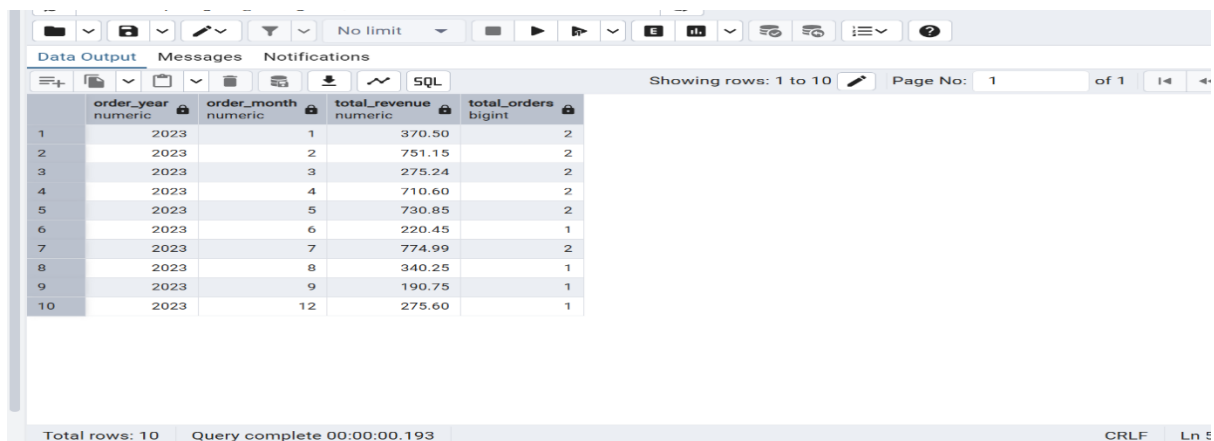
COUNT(DISTINCT order\_id) AS total\_orders

FROM sales12

WHERE EXTRACT(YEAR FROM order\_date) = 2023

GROUP BY EXTRACT(YEAR FROM order\_date), EXTRACT(MONTH FROM order\_date)

ORDER BY order\_year, order\_month;



	order_year numeric	order_month numeric	total_revenue numeric	total_orders bigint
1	2023	1	370.50	2
2	2023	2	751.15	2
3	2023	3	275.24	2
4	2023	4	710.60	2
5	2023	5	730.85	2
6	2023	6	220.45	1
7	2023	7	774.99	2
8	2023	8	340.25	1
9	2023	9	190.75	1
10	2023	12	275.60	1

Total rows: 10    Query complete 00:00:00.193    CRLF    Ln 5

SELECT

EXTRACT(YEAR FROM order\_date) AS order\_year,

EXTRACT(MONTH FROM order\_date) AS order\_month,

SUM(amount) AS total\_revenue,

COUNT(DISTINCT order\_id) AS total\_orders

FROM sales12

WHERE EXTRACT(YEAR FROM order\_date) = 2023

GROUP BY order\_year, order\_month

ORDER BY order\_year, order\_month;



▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

▼

<

SELECT

EXTRACT(YEAR FROM order\_date) AS order\_year,

EXTRACT(MONTH FROM order\_date) AS order\_month,

SUM(amount) AS total\_revenue, -- revenue aggregation

COUNT(DISTINCT order\_id) AS total\_orders

FROM sales12

WHERE EXTRACT(YEAR FROM order\_date) = 2024

GROUP BY order\_year, order\_month

ORDER BY order\_year, order\_month;

Query

Query History

Scratch Pad

```
72 SELECT
73     EXTRACT(YEAR FROM order_date) AS order_year,
74     EXTRACT(MONTH FROM order_date) AS order_month,
75     SUM(amount) AS total_revenue,      -- revenue aggregation
76     COUNT(DISTINCT order_id) AS total_orders
77 FROM sales12
78 WHERE EXTRACT(YEAR FROM order_date) = 2024
79 GROUP BY order_year, order_month
80 ORDER BY order_year, order_month;
81
```

Data Output

Messages

Notifications

Showing rows: 1 to 4

Page No: 1

	order_year numeric	order_month numeric	total_revenue numeric	total_orders bigint
1	2024	6	150.30	1
2	2024	8	600.80	1
3	2024	9	420.00	1
4	2024	11	455.75	1

Total rows: 4    Query complete 00:00:00.145

SELECT

```
EXTRACT(YEAR FROM order_date) AS order_year,  
EXTRACT(MONTH FROM order_date) AS order_month,  
SUM(amount) AS total_revenue,  
COUNT(DISTINCT order_id) AS total_orders
```

FROM sales12

WHERE EXTRACT(YEAR FROM order\_date) = 2023

GROUP BY order\_year, order\_month

ORDER BY order\_year, order\_month;

	order_year numeric	order_month numeric	total_revenue numeric	total_orders bigint
1	2023	1	370.50	2
2	2023	2	751.15	2
3	2023	3	275.24	2
4	2023	4	710.60	2
5	2023	5	730.85	2
6	2023	6	220.45	1
7	2023	7	774.99	2
8	2023	8	340.25	1
9	2023	9	190.75	1
10	2023	12	275.60	1

SELECT

```
EXTRACT(YEAR FROM order_date) AS order_year,  
EXTRACT(MONTH FROM order_date) AS order_month,  
SUM(amount) AS total_revenue,  
COUNT(DISTINCT order_id) AS total_orders
```

FROM sales12

WHERE EXTRACT(YEAR FROM order\_date) = 2023

GROUP BY order\_year, order\_month

ORDER BY order\_year, order\_month;

	order_year numeric	order_month numeric	total_revenue numeric	total_orders bigint
1	2023	1	370.50	2
2	2023	2	751.15	2
3	2023	3	275.24	2
4	2023	4	710.60	2
5	2023	5	730.85	2
6	2023	6	220.45	1
7	2023	7	774.99	2
8	2023	8	340.25	1
9	2023	9	190.75	1
10	2023	12	275.60	1

SELECT

EXTRACT(YEAR FROM order\_date) AS order\_year,

EXTRACT(MONTH FROM order\_date) AS order\_month,

SUM(amount) AS total\_revenue,

COUNT(DISTINCT order\_id) AS total\_orders

FROM sales12

WHERE EXTRACT(YEAR FROM order\_date) = 2025

GROUP BY order\_year, order\_month

ORDER BY order\_year, order\_month;

	order_year numeric	order_month numeric	total_revenue numeric	total_orders bigint
1	2025	10	800.19	2
2	2025	11	135.50	1

SELECT

EXTRACT(YEAR FROM order\_date) AS order\_year,



```

EXTRACT(MONTH FROM order_date) AS order_month,

SUM(amount) AS total_revenue,

COUNT(DISTINCT order_id) AS total_orders

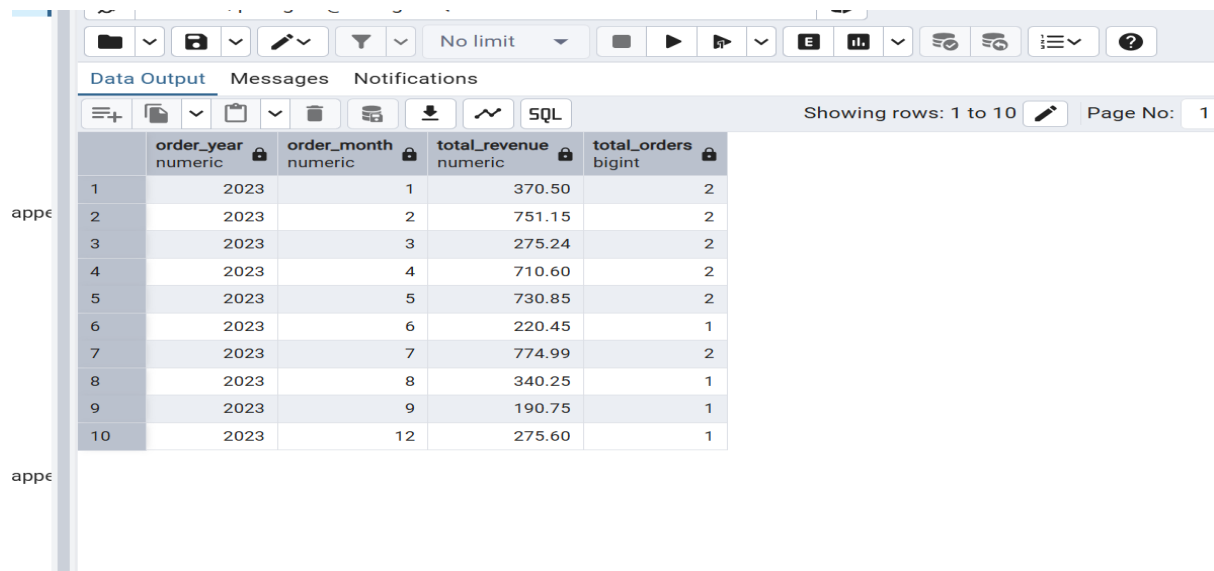
FROM sales12

WHERE EXTRACT(YEAR FROM order_date) = 2023

GROUP BY order_year, order_month

ORDER BY order_year, order_month;

```



	order_year numeric	order_month numeric	total_revenue numeric	total_orders bigint
1	2023	1	370.50	2
2	2023	2	751.15	2
3	2023	3	275.24	2
4	2023	4	710.60	2
5	2023	5	730.85	2
6	2023	6	220.45	1
7	2023	7	774.99	2
8	2023	8	340.25	1
9	2023	9	190.75	1
10	2023	12	275.60	1

```

SELECT

EXTRACT(YEAR FROM order_date) AS order_year,

EXTRACT(MONTH FROM order_date) AS order_month,

SUM(amount) AS total_revenue,

COUNT(DISTINCT order_id) AS total_orders

FROM sales12

WHERE EXTRACT(YEAR FROM order_date) = 2024

GROUP BY order_year, order_month

ORDER BY order_year, order_month;

```

sales12/postgres@PostgreSQL 18

Query Query History Scratch Pad X

```
122 SELECT
123     EXTRACT(YEAR FROM order_date) AS order_year,
124     EXTRACT(MONTH FROM order_date) AS order_month,
125     SUM(amount) AS total_revenue,
126     COUNT(DISTINCT order_id) AS total_orders
127 FROM sales12
128 WHERE EXTRACT(YEAR FROM order_date) = 2024
129 GROUP BY order_year, order_month
130 ORDER BY order_year, order_month;
131
```

Data Output Messages Notifications

Showing rows: 1 to 4 Page No: 1 of 1

	order_year numeric	order_month numeric	total_revenue numeric	total_orders bigint
1	2024	6	150.30	1
2	2024	8	600.80	1
3	2024	9	420.00	1
4	2024	11	455.75	1

✓ Successfully run. Total query runtime: 131 msec. 4 rows affected.

SELECT

EXTRACT(YEAR FROM order\_date) AS order\_year,

EXTRACT(MONTH FROM order\_date) AS order\_month,

SUM(amount) AS total\_revenue,

COUNT(DISTINCT order\_id) AS total\_orders

FROM sales12

WHERE EXTRACT(YEAR FROM order\_date) = 2023

AND EXTRACT(MONTH FROM order\_date) = 3

GROUP BY order\_year, order\_month

ORDER BY order\_year, order\_month;

Query Query History Scratch Pad X

```
132 SELECT
133     EXTRACT(YEAR FROM order_date) AS order_year,
134     EXTRACT(MONTH FROM order_date) AS order_month,
135     SUM(amount) AS total_revenue,
136     COUNT(DISTINCT order_id) AS total_orders
137 FROM sales12
138 WHERE EXTRACT(YEAR FROM order_date) = 2023
139     AND EXTRACT(MONTH FROM order_date) = 3
140 GROUP BY order_year, order_month
141 ORDER BY order_year, order_month;
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	order_year numeric	order_month numeric	total_revenue numeric	total_orders bigint
1	2023	3	275.24	2

✓ Successfully run. Total query runtime: 165 msec. 1 rows affected.

SELECT

```
EXTRACT(YEAR FROM order_date) AS order_year,  
EXTRACT(MONTH FROM order_date) AS order_month,  
SUM(amount) AS total_revenue,  
COUNT(DISTINCT order_id) AS total_orders
```

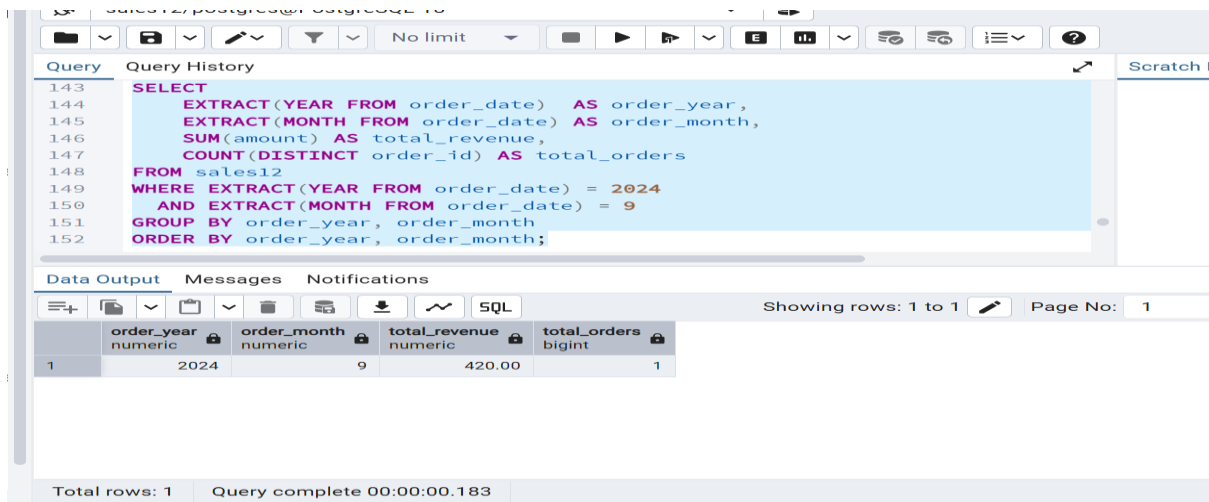
FROM sales12

WHERE EXTRACT(YEAR FROM order\_date) = 2024

AND EXTRACT(MONTH FROM order\_date) = 9

GROUP BY order\_year, order\_month

ORDER BY order\_year, order\_month;



The screenshot shows a PostgreSQL query editor interface. The top section displays the SQL query being executed. Below the query, the 'Data Output' tab is active, showing a single row of results. The results table has four columns: 'order\_year' (numeric), 'order\_month' (numeric), 'total\_revenue' (numeric), and 'total\_orders' (bigint). The values for the single row are 2024, 9, 420.00, and 1 respectively. The status bar at the bottom indicates 'Total rows: 1' and 'Query complete 00:00:00.183'.

```
SELECT  
    EXTRACT(YEAR FROM order_date) AS order_year,  
    EXTRACT(MONTH FROM order_date) AS order_month,  
    SUM(amount) AS total_revenue,  
    COUNT(DISTINCT order_id) AS total_orders  
FROM sales12  
WHERE EXTRACT(YEAR FROM order_date) = 2024  
    AND EXTRACT(MONTH FROM order_date) = 9  
GROUP BY order_year, order_month  
ORDER BY order_year, order_month;
```

order_year numeric	order_month numeric	total_revenue numeric	total_orders bigint
2024	9	420.00	1

Total rows: 1    Query complete 00:00:00.183

SELECT

```
EXTRACT(YEAR FROM order_date) AS order_year,  
EXTRACT(MONTH FROM order_date) AS order_month,  
SUM(amount) AS total_revenue,  
COUNT(DISTINCT order_id) AS total_orders
```

FROM sales12

WHERE order\_date BETWEEN '2023-01-01' AND '2023-03-31'

GROUP BY order\_year, order\_month

ORDER BY order\_year, order\_month;

The screenshot shows a SQL IDE interface. At the top, there's a toolbar with various icons for file operations, execution, and settings. Below the toolbar, the 'Query' tab is active, displaying a SQL query. The query is as follows:

```
SELECT
    EXTRACT(YEAR FROM order_date) AS order_year,
    EXTRACT(MONTH FROM order_date) AS order_month,
    SUM(amount) AS total_revenue,
    COUNT(DISTINCT order_id) AS total_orders
FROM sales12
WHERE order_date BETWEEN '2023-01-01' AND '2023-03-31'
GROUP BY order_year, order_month
ORDER BY order_year, order_month;
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following columns: order\_year (numeric), order\_month (numeric), total\_revenue (numeric), and total\_orders (bigint). The table contains three rows of data.

	order_year numeric	order_month numeric	total_revenue numeric	total_orders bigint
1	2023	1	370.50	2
2	2023	2	751.15	2
3	2023	3	275.24	2

At the bottom of the interface, there's a status bar showing 'Total rows: 3', 'Query complete 00:00:00.237', and 'CRLF'.

SELECT

EXTRACT(YEAR FROM order\_date) AS order\_year,

EXTRACT(MONTH FROM order\_date) AS order\_month,

SUM(amount) AS total\_revenue,

COUNT(DISTINCT order\_id) AS total\_orders

FROM sales12

WHERE order\_date BETWEEN '2025-01-01' AND '2025-12-31'

GROUP BY order\_year, order\_month

ORDER BY order\_year, order\_month;

Query Query History Execute script F5 Scratch Pad

```
164 SELECT
165     EXTRACT(YEAR FROM order_date) AS order_year,
166     EXTRACT(MONTH FROM order_date) AS order_month,
167     SUM(amount) AS total_revenue,
168     COUNT(DISTINCT order_id) AS total_orders
169 FROM sales12
170 WHERE order_date BETWEEN '2025-01-01' AND '2025-12-31'
171 GROUP BY order_year, order_month
172 ORDER BY order_year, order_month;
173
```

Data Output Messages Notifications

+

📄

▼

📋

▼

🗑️

📦

⬇️

📈

SQL

Showing rows: 2 Page No: 1

	order_year numeric 🔒	order_month numeric 🔒	total_revenue numeric 🔒	total_orders bigint 🔒
1	2025	10	800.19	2
2	2025	11	135.50	1