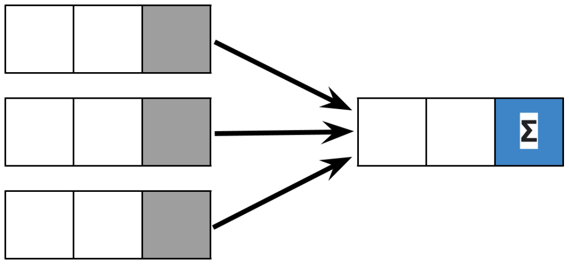
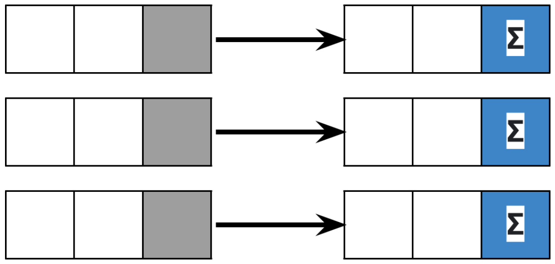
# Window Functions in SQL - How they work? with examples

## What is a Window Function?

A Window Function is used to perform a calculation on an aggregate value based on a set of rows and return multiple rows for each group.

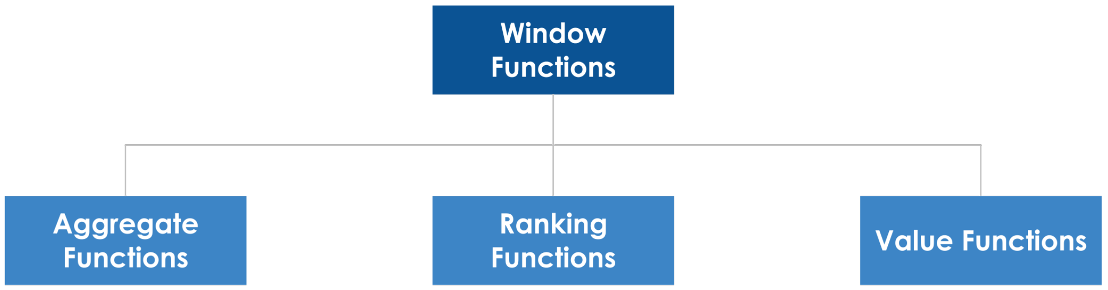


AGGREGATE FUNCTIONS



WINDOW FUNCTIONS

## Types of Window Fucntions



- COUNT()
- SUM()
- AVG()

- ROW_NUMBER()
- RANK()
- DENSE_RANK()
- CUME_DIST()
- NTILE()

- LAG()
- LEAD()
- FIRST_VALUE()
- LAST_VALUE()

| Examples | Concept |
|---|---|
| Example-1 | COUNT |
| Example-2 | SUM |
| Example-3 | AVG |
| Example-4 | ROW_NUMBER |
| Example-5 | RANK |
| Example-6 | DENSE_RANK |
| Example-7 | NTILE |
| Example-8 | LEAD |
| Example-9 | LAG |
| Example-10 | FIRST_VALUE |
| Example-11 | LAST_VLAUE |

1 **Find the the total number of clients per state and city**

| Revenue | | | |
|---|---|---|---|
| state | city | client_id | sales |
| California | Los Angeles | 1001 | 10000 |
| California | Los Angeles | 1002 | 10000 |
| California | San Diego | 1003 | 30000 |
| California | San Diego | 1004 | 40000 |
| Texas | Houston | 1005 | 60000 |
| Texas | Houston | 1006 | 50000 |
| Texas | Austin | 1007 | 60000 |
| Texas | Austin | 1008 | 50000 |

**Let's first partition the data by states and count the number of rows in each partition**

| state | city | client_id | sales | no_of_clients |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 4 |
| California | Los Angeles | 1002 | 10000 | 4 |
| California | San Diego | 1003 | 30000 | 4 |
| California | San Diego | 1004 | 40000 | 4 |
| Texas | Houston | 1005 | 60000 | 4 |
| Texas | Houston | 1006 | 50000 | 4 |
| Texas | Austin | 1007 | 60000 | 4 |
| Texas | Austin | 1008 | 50000 | 4 |

| state | no_of_clients |
|---|---|

|  |  |
|---|---|
| California | 4 |
| California | 4 |
| California | 4 |
| California | 4 |
| Texas | 4 |
| Texas | 4 |
| Texas | 4 |
| Texas | 4 |

**Now, let's partition the data by cities and count the number of rows in each partition**

| state | city | client_id | sales | no_of_clients |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 2 |
| California | Los Angeles | 1002 | 10000 | 2 |
| California | San Diego | 1003 | 30000 | 2 |
| California | San Diego | 1004 | 40000 | 2 |
| Texas | Houston | 1005 | 60000 | 2 |
| Texas | Houston | 1006 | 50000 | 2 |
| Texas | Austin | 1007 | 60000 | 2 |
| Texas | Austin | 1008 | 50000 | 2 |

| city | no_of_clients |
|---|---|
| Los Angeles | 2 |
| Los Angeles | 2 |
| San Diego | 2 |
| San Diego | 2 |

|  |  |
|---|---|
| Houston | 2 |
| Houston | 2 |
| Austin | 2 |
| Austin | 2 |

**How can I write a query to get this answer programmatically?**

```
SELECT client_id, state, city,
   COUNT(client_id) OVER(Partition by state) as clients_state,
   COUNT(client_id) OVER(Partition by city) as clients_city
FROM revenue;
```

| client_id | state | city | clients_state | clients_city |
|---|---|---|---|---|
| 1001 | California | Los Angeles | 4 | 2 |
| 1002 | California | Los Angeles | 4 | 2 |
| 1003 | California | San Diego | 4 | 2 |
| 1004 | California | San Diego | 4 | 2 |
| 1005 | Texas | Houston | 4 | 2 |
| 1006 | Texas | Houston | 4 | 2 |
| 1007 | Texas | Austin | 4 | 2 |
| 1008 | Texas | Austin | 4 | 2 |

2 Write a query to show the total amount of sales in each state and the total amount of sales in each city

| Revenue | | | |
|---|---|---|---|
| state | city | client_id | sales |
| California | Los Angeles | 1001 | 10000 |
| California | Los Angeles | 1002 | 10000 |
| California | San Diego | 1003 | 30000 |
| California | San Diego | 1004 | 40000 |
| Texas | Houston | 1005 | 60000 |
| Texas | Houston | 1006 | 50000 |
| Texas | Austin | 1007 | 60000 |
| Texas | Austin | 1008 | 50000 |

Let's first partition the data by states and find the sum of sales in each partition

| state | city | client_id | sales | state_sales |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 90000 |
| California | Los Angeles | 1002 | 10000 | 90000 |
| California | San Diego | 1003 | 30000 | 90000 |
| California | San Diego | 1004 | 40000 | 90000 |
| Texas | Houston | 1005 | 60000 | 220000 |
| Texas | Houston | 1006 | 50000 | 220000 |
| Texas | Austin | 1007 | 60000 | 220000 |
| Texas | Austin | 1008 | 50000 | 220000 |

| state | state_sales |
|---|---|

| | |
|---|---|
| California | 90000 |
| California | 90000 |
| California | 90000 |
| California | 90000 |
| Texas | 220000 |
| Texas | 220000 |
| Texas | 220000 |
| Texas | 220000 |

**Now, let's partition the data by cities and find the sum of sales in each partition**

| state | city | client_id | sales | city_sales |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 20000 |
| California | Los Angeles | 1002 | 10000 | 20000 |
| California | San Diego | 1003 | 30000 | 70000 |
| California | San Diego | 1004 | 40000 | 70000 |
| Texas | Houston | 1005 | 60000 | 110000 |
| Texas | Houston | 1006 | 50000 | 110000 |
| Texas | Austin | 1007 | 60000 | 110000 |
| Texas | Austin | 1008 | 50000 | 110000 |

| city | city_sales |
|---|---|
| Los Angeles | 20000 |
| Los Angeles | 20000 |
| San Diego | 70000 |
| San Diego | 70000 |

|  |  |
|---|---|
| Houston | 110000 |
| Houston | 110000 |
| Austin | 110000 |
| Austin | 110000 |

**How can I write a query to get this answer programmatically?**

```
SELECT state, city,
   SUM(sales) OVER(Partition by state) as state_sales,
   SUM(sales) OVER(Partition by city) as city_sales
FROM revenue
ORDER BY client_id;
```

| state | city | state_sales | city_sales |
|---|---|---|---|
| California | Los Angeles | 90000 | 20000 |
| California | Los Angeles | 90000 | 20000 |
| California | San Diego | 90000 | 70000 |
| California | San Diego | 90000 | 70000 |
| Texas | Houston | 220000 | 110000 |
| Texas | Houston | 220000 | 110000 |
| Texas | Austin | 220000 | 110000 |
| Texas | Austin | 220000 | 110000 |

3 Write a query to show the average amount of sales in each state and the average amount of sales in each city

| Revenue | | | |
|---|---|---|---|
| state | city | client_id | sales |
| California | Los Angeles | 1001 | 10000 |
| California | Los Angeles | 1002 | 10000 |
| California | San Diego | 1003 | 30000 |
| California | San Diego | 1004 | 40000 |
| Texas | Houston | 1005 | 60000 |
| Texas | Houston | 1006 | 50000 |
| Texas | Austin | 1007 | 60000 |
| Texas | Austin | 1008 | 50000 |

Let's first partition the data by states and find the average amount of sales in each partition

| state | city | client_id | sales | avg_state |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 22500 |
| California | Los Angeles | 1002 | 10000 | 22500 |
| California | San Diego | 1003 | 30000 | 22500 |
| California | San Diego | 1004 | 40000 | 22500 |
| Texas | Houston | 1005 | 60000 | 55000 |
| Texas | Houston | 1006 | 50000 | 55000 |
| Texas | Austin | 1007 | 60000 | 55000 |
| Texas | Austin | 1008 | 50000 | 55000 |

| state | avg_state |
|---|---|

| | |
|---|---|
| California | 22500 |
| California | 22500 |
| California | 22500 |
| California | 22500 |
| Texas | 55000 |
| Texas | 55000 |
| Texas | 55000 |
| Texas | 55000 |

Now, let's partition the data by cities and find the average amount of sales in each partition

| state | city | client_id | sales | city_sales |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 10000 |
| California | Los Angeles | 1002 | 10000 | 10000 |
| California | San Diego | 1003 | 30000 | 35000 |
| California | San Diego | 1004 | 40000 | 35000 |
| Texas | Houston | 1005 | 60000 | 55000 |
| Texas | Houston | 1006 | 50000 | 55000 |
| Texas | Austin | 1007 | 60000 | 55000 |
| Texas | Austin | 1008 | 50000 | 55000 |

| city | avg_city |
|---|---|
| Los Angeles | 10000 |
| Los Angeles | 10000 |
| San Diego | 35000 |
| San Diego | 35000 |

| | |
|---|---|
| Houston | 55000 |
| Houston | 55000 |
| Austin | 55000 |
| Austin | 55000 |

**How can I write a query to get this answer programmatically?**

```
SELECT client_id, state, city,
    AVG(sales) OVER(Partition by state) as avg_state,
    AVG(sales) OVER(Partition by city) as avg_city
FROM revenue;
```

| client_id | state | city | avg_state | avg_city |
|---|---|---|---|---|
| 1001 | California | Los Angeles | 22500 | 10000 |
| 1002 | California | Los Angeles | 22500 | 10000 |
| 1003 | California | San Diego | 22500 | 35000 |
| 1004 | California | San Diego | 22500 | 35000 |
| 1005 | Texas | Houston | 55000 | 55000 |
| 1006 | Texas | Houston | 55000 | 55000 |
| 1007 | Texas | Austin | 55000 | 55000 |
| 1008 | Texas | Austin | 55000 | 55000 |

4 Write a query to assign row numbers to the revenue table based on states

| Revenue | | | |
|---|---|---|---|
| state | city | client_id | sales |
| California | Los Angeles | 1001 | 10000 |
| California | Los Angeles | 1002 | 10000 |
| California | San Diego | 1003 | 30000 |
| California | San Diego | 1004 | 40000 |
| Texas | Houston | 1005 | 60000 |
| Texas | Houston | 1006 | 50000 |
| Texas | Austin | 1007 | 60000 |
| Texas | Austin | 1008 | 50000 |

Let's first partition the data by states and assign row numbers to each row in the partition

| state | city | client_id | sales | row_no |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 1 |
| California | Los Angeles | 1002 | 10000 | 2 |
| California | San Diego | 1003 | 30000 | 3 |
| California | San Diego | 1004 | 40000 | 4 |
| Texas | Houston | 1005 | 60000 | 1 |
| Texas | Houston | 1006 | 50000 | 2 |
| Texas | Austin | 1007 | 60000 | 3 |
| Texas | Austin | 1008 | 50000 | 4 |

| state | row_no |
|---|---|
| California | 1 |
| California | 2 |
| California | 3 |
| California | 4 |
| Texas | 1 |
| Texas | 2 |
| Texas | 3 |
| Texas | 4 |

**How can I write a query to get this answer programmatically?**

```
SELECT *,
   ROW_NUMBER() OVER(Partition by state) as row_no
FROM revenue;
```

| state | city | client_id | sales | row_no |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 1 |
| California | Los Angeles | 1002 | 10000 | 2 |
| California | San Diego | 1003 | 30000 | 3 |
| California | San Diego | 1004 | 40000 | 4 |
| Texas | Houston | 1005 | 60000 | 1 |
| Texas | Houston | 1006 | 50000 | 2 |
| Texas | Austin | 1007 | 60000 | 3 |
| Texas | Austin | 1008 | 50000 | 4 |

5 Write a query to assign rank based on the sales in each state

| Revenue | | | |
|---|---|---|---|
| state | city | client_id | sales |
| California | Los Angeles | 1001 | 10000 |
| California | Los Angeles | 1002 | 10000 |
| California | San Diego | 1003 | 30000 |
| California | San Diego | 1004 | 40000 |
| Texas | Houston | 1005 | 60000 |
| Texas | Houston | 1006 | 50000 |
| Texas | Austin | 1007 | 60000 |
| Texas | Austin | 1008 | 50000 |

Let's rank the data by partitioning the states and sorting sales in ascending order

| state | city | client_id | sales | rnk |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 1 |
| California | Los Angeles | 1002 | 10000 | 1 |
| California | San Diego | 1003 | 30000 | 3 |
| California | San Diego | 1004 | 40000 | 4 |
| Texas | Houston | 1006 | 50000 | 1 |
| Texas | Austin | 1008 | 50000 | 1 |
| Texas | Houston | 1005 | 60000 | 3 |
| Texas | Austin | 1007 | 60000 | 3 |

How can I write a query to get this answer programmatically?

```
SELECT *,
   RANK() OVER(Partition by state ORDER BY sales) as rnk
FROM revenue;
```

| state | city | client_id | sales | rnk |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 1 |
| California | Los Angeles | 1002 | 10000 | 1 |
| California | San Diego | 1003 | 30000 | 3 |
| California | San Diego | 1004 | 40000 | 4 |
| Texas | Houston | 1006 | 50000 | 1 |
| Texas | Austin | 1008 | 50000 | 1 |
| Texas | Houston | 1005 | 60000 | 3 |
| Texas | Austin | 1007 | 60000 | 3 |

6 Write a query to assign rank based on the sales of each state

| Revenue | | | |
|---|---|---|---|
| state | city | client_id | sales |
| California | Los Angeles | 1001 | 10000 |
| California | Los Angeles | 1002 | 10000 |
| California | San Diego | 1003 | 30000 |
| California | San Diego | 1004 | 40000 |
| Texas | Houston | 1005 | 60000 |
| Texas | Houston | 1006 | 50000 |
| Texas | Austin | 1007 | 60000 |
| Texas | Austin | 1008 | 50000 |

Let's rank the data by partitioning the states and sorting sales in ascending order

| state | city | client_id | sales | dense_rnk |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 1 |
| California | Los Angeles | 1002 | 10000 | 1 |
| California | San Diego | 1003 | 30000 | 2 |
| California | San Diego | 1004 | 40000 | 3 |
| Texas | Houston | 1006 | 50000 | 1 |
| Texas | Austin | 1008 | 50000 | 1 |
| Texas | Houston | 1005 | 60000 | 2 |
| Texas | Austin | 1007 | 60000 | 2 |

How can I write a query to get this answer programmatically?

```
SELECT *,
    DENSE_RANK() OVER(Partition by state ORDER BY sales) as dense_rnk
FROM revenue;
```

| state | city | client_id | sales | dense_rnk |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 1 |

NOTE:

Unlike the RANK() function, the DENSE_RANK() function returns consecutive rank values.

| | | | | |
|---|---|---|---|---|
| California | Los Angeles | 1002 | 10000 | 1 |
| California | San Diego | 1003 | 30000 | 2 |
| California | San Diego | 1004 | 40000 | 3 |
| Texas | Houston | 1006 | 50000 | 1 |
| Texas | Austin | 1008 | 50000 | 1 |
| Texas | Houston | 1005 | 60000 | 2 |
| Texas | Austin | 1007 | 60000 | 2 |

8 Write a query to bucket the data into 3 buckets based on the range of sales

| Revenue | | | |
|---|---|---|---|
| state | city | client_id | sales |
| California | Los Angeles | 1001 | 10000 |
| California | Los Angeles | 1002 | 10000 |
| California | San Diego | 1003 | 30000 |
| California | San Diego | 1004 | 40000 |
| Texas | Houston | 1005 | 60000 |
| Texas | Houston | 1006 | 50000 |
| Texas | Austin | 1007 | 60000 |
| Texas | Austin | 1008 | 50000 |

Let's first order the data by sales and then group them into three groups based on the range of sales

| state | city | client_id | sales | tier |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 1 |
| California | Los Angeles | 1002 | 10000 | 1 |
| California | San Diego | 1003 | 30000 | 1 |
| California | San Diego | 1004 | 40000 | 2 |
| Texas | Houston | 1006 | 50000 | 2 |
| Texas | Austin | 1008 | 50000 | 2 |
| Texas | Houston | 1005 | 60000 | 3 |
| Texas | Austin | 1007 | 60000 | 3 |

How can I write a query to get this answer programmatically?

```
SELECT *,
   NTILE(3) OVER(ORDER BY sales) as tier
FROM revenue;
```

| client_id | state | city | sales | tier |
|---|---|---|---|---|
| California | Los Angeles | 1001 | 10000 | 1 |
| California | Los Angeles | 1002 | 10000 | 1 |
| California | San Diego | 1003 | 30000 | 1 |
| California | San Diego | 1004 | 40000 | 2 |
| Texas | Houston | 1006 | 50000 | 2 |
| Texas | Austin | 1008 | 50000 | 2 |
| Texas | Houston | 1005 | 60000 | 3 |
| Texas | Austin | 1007 | 60000 | 3 |

9 Write a query to find the sales of the following month in each of the states

| Revenue | | |
|---|---|---|
| state | month | sales |
| California | 4 | 30000 |
| California | 1 | 10000 |
| California | 3 | 40000 |
| California | 2 | 10000 |
| Texas | 1 | 50000 |
| Texas | 4 | 60000 |
| Texas | 3 | 60000 |
| Texas | 2 | 50000 |

Let's first partition the states and then sort the data by month

| state | month | sales |
|---|---|---|
| California | 1 | 10000 |
| California | 2 | 10000 |
| California | 3 | 40000 |
| California | 4 | 30000 |
| Texas | 1 | 50000 |
| Texas | 2 | 50000 |
| Texas | 3 | 60000 |
| Texas | 4 | 60000 |

Now, let's find the sales of the following month

| state | month | sales | next_sales |
|---|---|---|---|
| California | 1 | 10000 | 10000 |
| California | 2 | 10000 | 40000 |
| California | 3 | 40000 | 30000 |
| California | 4 | 30000 | *Null* |
| Texas | 1 | 50000 | 50000 |
| Texas | 2 | 50000 | 60000 |
| Texas | 3 | 60000 | 60000 |
| Texas | 4 | 60000 | *Null* |

**How can I write a query to get this answer programmatically?**

```
SELECT *,
   LEAD(sales) OVER(PARTITION BY state ORDER BY sales) as next_sales
FROM revenue;
```

| state | city | sales | next_sales |
|---|---|---|---|
| California | 1 | 10000 | 10000 |
| California | 2 | 10000 | 40000 |
| California | 3 | 40000 | 30000 |
| California | 4 | 30000 | *Null* |
| Texas | 1 | 50000 | 50000 |
| Texas | 2 | 50000 | 60000 |
| Texas | 3 | 60000 | 60000 |
| Texas | 4 | 60000 | *Null* |

10 Write a query to find the sales of the previous month in each of the states

| Revenue | | |
|---|---|---|
| state | month | sales |
| California | 4 | 30000 |
| California | 1 | 10000 |
| California | 3 | 40000 |
| California | 2 | 10000 |
| Texas | 1 | 50000 |
| Texas | 4 | 60000 |
| Texas | 3 | 60000 |
| Texas | 2 | 50000 |

Let's first partition the states and then sort the data by month

| state | month | sales |
|---|---|---|
| California | 1 | 10000 |
| California | 2 | 10000 |
| California | 3 | 40000 |
| California | 4 | 30000 |
| Texas | 1 | 50000 |
| Texas | 2 | 50000 |
| Texas | 3 | 60000 |
| Texas | 4 | 60000 |

Now, let's find the sales of the previous month

| state | month | sales | prev_sales |
|---|---|---|---|
| California | 1 | 10000 | *Null* |
| California | 2 | 10000 | 10000 |
| California | 3 | 40000 | 10000 |
| California | 4 | 30000 | 40000 |
| Texas | 1 | 50000 | *Null* |
| Texas | 2 | 50000 | 50000 |
| Texas | 3 | 60000 | 50000 |
| Texas | 4 | 60000 | 60000 |

**How can I write a query to get this answer programmatically?**

```
SELECT *,
    LAG(sales) OVER(PARTITION BY state ORDER BY sales) as prev_sales
FROM revenue;
```

| state | city | sales | prev_sales |
|---|---|---|---|
| California | 1 | 10000 | 10000 |
| California | 2 | 10000 | 40000 |
| California | 3 | 40000 | 30000 |
| California | 4 | 30000 | *Null* |
| Texas | 1 | 50000 | 50000 |
| Texas | 2 | 50000 | 60000 |
| Texas | 3 | 60000 | 60000 |
| Texas | 4 | 60000 | *Null* |

11 Write a query to find the very first sale in each of the states

| Revenue | | |
|---|---|---|
| state | month | sales |
| California | 4 | 30000 |
| California | 1 | 10000 |
| California | 3 | 40000 |
| California | 2 | 10000 |
| Texas | 1 | 50000 |
| Texas | 4 | 60000 |
| Texas | 3 | 60000 |
| Texas | 2 | 50000 |

Let's first partition the states and then sort the data by month

| state | month | sales |
|---|---|---|
| California | 1 | 10000 |
| California | 2 | 10000 |
| California | 3 | 40000 |
| California | 4 | 30000 |
| Texas | 1 | 50000 |
| Texas | 2 | 50000 |
| Texas | 3 | 60000 |
| Texas | 4 | 60000 |

Now, let's find the sales of the first month

| state | month | sales | first_sales |
|---|---|---|---|
| California | 1 | 10000 | 10000 |
| California | 2 | 10000 | 10000 |
| California | 3 | 40000 | 10000 |
| California | 4 | 30000 | 10000 |
| Texas | 1 | 50000 | 50000 |
| Texas | 2 | 50000 | 50000 |
| Texas | 3 | 60000 | 50000 |
| Texas | 4 | 60000 | 50000 |

**How can I write a query to get this answer programmatically?**

```
SELECT *,
   FIRST_VALUE(sales) OVER(PARTITION BY state ORDER BY sales) as first_sales
FROM revenue;
```

| state | city | sales | first_sales |
|---|---|---|---|
| California | 1 | 10000 | 10000 |
| California | 2 | 10000 | 10000 |
| California | 3 | 40000 | 10000 |
| California | 4 | 30000 | 10000 |
| Texas | 1 | 50000 | 50000 |
| Texas | 2 | 50000 | 50000 |
| Texas | 3 | 60000 | 50000 |
| Texas | 4 | 60000 | 50000 |

12 Write a query to find the very last sale in each of the states

| Revenue | | |
|---|---|---|
| state | month | sales |
| California | 4 | 30000 |
| California | 1 | 10000 |
| California | 3 | 40000 |
| California | 2 | 10000 |
| Texas | 1 | 50000 |
| Texas | 4 | 60000 |
| Texas | 3 | 60000 |
| Texas | 2 | 50000 |

Let's first partition the states and then sort the data by month

| state | month | sales |
|---|---|---|
| California | 1 | 10000 |
| California | 2 | 10000 |
| California | 3 | 40000 |
| California | 4 | 30000 |
| Texas | 1 | 50000 |
| Texas | 2 | 50000 |
| Texas | 3 | 60000 |
| Texas | 4 | 60000 |

Now, let's find the sales of the last month

| state | month | sales | last_sale |
|---|---|---|---|
| California | 1 | 10000 | 30000 |
| California | 2 | 10000 | 30000 |
| California | 3 | 40000 | 30000 |
| California | 4 | 30000 | 30000 |
| Texas | 1 | 50000 | 60000 |
| Texas | 2 | 50000 | 60000 |
| Texas | 3 | 60000 | 60000 |
| Texas | 4 | 60000 | 60000 |

**How can I write a query to get this answer programmatically?**

```
SELECT *,
    LAST_VALUE(sales) OVER(PARTITION BY state ORDER BY sales) as last_sale
FROM revenue;
```

| state | city | sales | last_sale |
|---|---|---|---|
| California | 1 | 10000 | 30000 |
| California | 2 | 10000 | 30000 |
| California | 3 | 40000 | 30000 |
| California | 4 | 30000 | 30000 |
| Texas | 1 | 50000 | 60000 |
| Texas | 2 | 50000 | 60000 |
| Texas | 3 | 60000 | 60000 |
| Texas | 4 | 60000 | 60000 |