

SQL Introduction

SQL SELECT (I)

SQL SELECT (II)

SQL ORDER BY

SQL GROUP BY

SQL LIKE

SQL Wildcards

SQL UNION

SQL Subquery

SQL ANY and ALL

SQL CASE

SQL HAVING

SQL EXISTS

SQL JOIN

SQL DATABASE & TABLE

SQL Insert, Update and Delete

SQL Constraints

SQL Additional Topics

Related Topics

SQL HAVING Clause

SQL COUNT()

SQL SELECT INTO Statement

SQL INSERT INTO SELECT Statement

SQL SUM() AND AVG()

SQL SELECT DISTINCT Statement

SQL GROUP BY

In this tutorial, we'll learn about GROUP BY in SQL with the help of examples.

In SQL, the `GROUP BY` clause is used to group rows by one or more columns. For example,

```
SELECT country, COUNT(*) AS number
FROM Customers
GROUP BY country;
```

Run Code >>

Here, the SQL command groups the rows by the `country` column, and counts the number of each country (because of the `COUNT()` function).

Note: The `GROUP BY` clause is used in conjunction with aggregate functions such as `MIN()`, `MAX()`, `SUM()`, `AVG()` and `COUNT()`, etc.

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SELECT country, COUNT(*) AS number
FROM Customers
GROUP BY country;

country	number
UAE	1
UK	2
USA	2

Example: SQL GROUP BY

Here, the column name of the `COUNT()` function in the result-set is `number`, because of the `AS` alias. To learn more, visit [SQL AS Alias](#).

Example Two: GROUP BY in SQL

Let's try to find the total amount for each customer who has placed an order.

```
SELECT customer_id, SUM(amount) AS total
FROM Orders
GROUP BY customer_id;
```

Run Code >>

Here, the SQL command sums the `amount` after grouping rows with `customer_id`.

Table: Orders

order_id	item	amount	customer_id
1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

SELECT customer_id, SUM(amount) AS total
FROM Orders
GROUP BY customer_id;

customer_id	total
1	400
2	250
3	12000
4	700

Example: SQL GROUP BY

GROUP BY Clause With JOIN in SQL

We can also use the `GROUP BY` clause with the `JOIN` clause. For example,

```
SELECT Customers.customer_id, Customers.first_name, Count(Orders.order_id) AS order_count
FROM Customers
LEFT JOIN Orders
ON Customers.customer_id = Orders.customer_id
GROUP BY Customers.customer_id;
```

Run Code >>

Here, the SQL command joins `Customers` and `Orders` tables, and groups the result set by `customer_id` (a customer). This gives us the number of orders that are placed by each customer.

Visit [SQL JOIN](#) to read more about the `JOIN` clause in SQL.

GROUP BY With Multiple Columns

`GROUP BY` can also be used to group rows based on multiple columns. For example,

```
SELECT country, state, MIN(age) as min_age
FROM Persons
GROUP BY country, state;
```

Run Code >>

Here, the SQL command groups all persons with similar `country` and `state`, and gives the minimum `age` of each group.

GROUP BY With HAVING Clause

We can use the `GROUP BY` clause with the `HAVING` clause to filter the result set based on aggregate functions. For example,

```
SELECT COUNT(customer_id), country
FROM Customers
GROUP BY country
HAVING COUNT(customer_id) > 1;
```

Run Code >>

Here, the SQL command:

- counts the number of rows by grouping them by `country`
- returns the result set if their count is greater than 1

To learn more, visit [SQL HAVING Clause](#).

Previous Tutorial:
SQL ORDER BY

Next Tutorial:
SQL LIKE >

Did you find this article helpful?

😊

😞

Related Tutorials

Programming

SQL HAVING Clause

Programming

SQL COUNT()

Programming

SQL SELECT INTO Statement

Programming

SQL SUM() AND AVG()