



LEARN SQLITE

sql database engine

SQLite Tutorial

- SQLite - Home
- SQLite - Overview
- SQLite - Installation
- SQLite - Commands
- SQLite - Syntax
- SQLite - Data Type
- SQLite - CREATE Database
- SQLite - ATTACH Database
- SQLite - DETACH Database
- SQLite - CREATE Table
- SQLite - DROP Table
- SQLite - INSERT Query
- SQLite - SELECT Query
- SQLite - Operators
- SQLite - Expressions
- SQLite - WHERE Clause
- SQLite - AND & OR Clauses
- SQLite - UPDATE Query
- SQLite - DELETE Query
- SQLite - LIKE Clause
- SQLite - GLOB Clause
- SQLite - LIMIT Clause
- SQLite - ORDER By Clause
- SQLite - GROUP By Clause
- SQLite - HAVING Clause
- SQLite - DISTINCT Keyword

Advanced SQLite

- SQLite - PRAGMA
- SQLite - Constraints
- SQLite - JOINS
- SQLite - UNIONS Clause
- SQLite - NULL Values
- SQLite - ALIAS Syntax
- SQLite - Triggers
- SQLite - Indexes
- SQLite - INDEXED By Clause
- SQLite - ALTER Command
- SQLite - TRUNCATE Command
- SQLite - Views
- SQLite - Transactions
- SQLite - Subqueries
- SQLite - AUTOINCREMENT
- SQLite - Injection
- SQLite - EXPLAIN
- SQLite - VACUUM
- SQLite - Date & Time
- SQLite - Useful Functions

SQLite Interfaces

- SQLite - C/C++
- SQLite - Java
- SQLite - PHP
- SQLite - Perl
- SQLite - Python

SQLite Useful Resources

- SQLite - Quick Guide
- SQLite - Useful Resources
- SQLite - Discussion

Selected Reading

- UPSC IAS Exams Notes
- Developer's Best Practices
- Questions and Answers
- Effective Resume Writing
- HR Interview Questions
- Computer Glossary
- Who is Who

SQLite - GROUP BY Clause

Advertisements



Previous Page



Next Page

SQLite **GROUP BY** clause is used in collaboration with the SELECT statement to arrange identical data into groups.

GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

Syntax

Following is the basic syntax of GROUP BY clause. GROUP BY clause must follow the conditions in the WHERE clause and must precede ORDER BY clause if one is used.

```
SELECT column-list
FROM table_name
WHERE [ conditions ]
GROUP BY column1, column2....columnN
ORDER BY column1, column2....columnN
```

You can use more than one column in the GROUP BY clause. Make sure whatever column you are using to group, that column should be available in the column-list.

Example

Consider COMPANY table with the following records.

ID	NAME	AGE	ADDRESS	SALARY
1	Paul	32	California	20000.0
2	Allen	25	Texas	15000.0
3	Teddy	23	Norway	20000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	85000.0
6	Kim	22	South-Hall	45000.0
7	James	24	Houston	10000.0

If you want to know the total amount of salary on each customer, then GROUP BY query will be as follows –

```
sqlite> SELECT NAME, SUM(SALARY) FROM COMPANY GROUP BY NAME;
```

This will produce the following result –

NAME	SUM(SALARY)
Allen	15000.0
David	85000.0
James	10000.0
Kim	45000.0
Mark	65000.0
Paul	20000.0
Teddy	20000.0

Now, let us create three more records in COMPANY table using the following INSERT statements.

```
INSERT INTO COMPANY VALUES (8, 'Paul', 24, 'Houston', 20000.00 );
INSERT INTO COMPANY VALUES (9, 'James', 44, 'Norway', 5000.00 );
INSERT INTO COMPANY VALUES (10, 'James', 45, 'Texas', 5000.00 );
```

Now, our table has the following records with duplicate names.

ID	NAME	AGE	ADDRESS	SALARY
1	Paul	32	California	20000.0
2	Allen	25	Texas	15000.0
3	Teddy	23	Norway	20000.0
4	Mark	25	Rich-Mond	65000.0
5	David	27	Texas	85000.0
6	Kim	22	South-Hall	45000.0
7	James	24	Houston	10000.0
8	Paul	24	Houston	20000.0
9	James	44	Norway	5000.0
10	James	45	Texas	5000.0

Again, let us use the same statement to group-by all the records using NAME column as follows –

```
sqlite> SELECT NAME, SUM(SALARY) FROM COMPANY GROUP BY NAME ORDER BY NAME;
```

This will produce the following result.

NAME	SUM(SALARY)
Allen	15000
David	85000
James	20000
Kim	45000
Mark	65000
Paul	40000
Teddy	20000

Let us use ORDER BY clause along with GROUP BY clause as follows –

```
sqlite> SELECT NAME, SUM(SALARY)
FROM COMPANY GROUP BY NAME ORDER BY NAME DESC;
```

This will produce the following result.

NAME	SUM(SALARY)
Teddy	20000
Paul	40000
Mark	65000
Kim	45000
James	20000
David	85000
Allen	15000

Previous Page

Print Page

Next Page

Advertisements



About us

Terms of use

Privacy Policy

FAQ's

Teach with Us

Contact