

SQL Views

[« Previous](#)[Next »](#)

SQL CREATE VIEW Statement

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.



HTML

CSS

JAVASCRIPT

SQL

PYTHON

PHP

BOOTSTRAP

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

COLOR
PICKER

Note: A view always shows up-to-date data! The database engine recreates the data, using the view's SQL statement, every time a user queries a view.

HOW
TO

SQL Case
SQL Null Functions
SQL Stored Procedures
SQL Comments

SQL Database

SQL Create DB
SQL Drop DB
SQL Backup DB
SQL Create Table
SQL Drop Table
SQL Alter Table
SQL Constraints
SQL Not Null
SQL Unique
SQL Primary Key
SQL Foreign Key
SQL Check
SQL Default
SQL Index
SQL Auto Increment
SQL Dates
SQL Views
SQL Injection
SQL Hosting

SQL References

SQL Keywords
MySQL Functions
SQL Server Functions
MS Access Functions
SQL Operators
SQL Data Types
SQL Quick Ref

SQL Examples

SQL Examples
SQL Quiz

SQL CREATE VIEW

Examples

The following SQL creates a view that shows all customers from Brazil:

Example

```
CREATE VIEW [Brazil Customers] AS
SELECT CustomerName, ContactName
FROM Customers
WHERE Country = "Brazil";
```

[Try it Yourself »](#)

We can query the view above as follows:

Example

```
SELECT * FROM [Brazil Customers];
```

[Try it Yourself »](#)

SHARE



CERTIFIC

HTML
CSS
JavaScript
SQL
Python
PHP

```
    Products);
```

[Try it Yourself »](#)

jQuery
Bootstrap
XML

[Read More](#)

We can query the view above as follows:

Example

```
SELECT * FROM [Products Above Average  
Price];
```

[Try it Yourself »](#)

SQL Updating a View

A view can be updated with the CREATE OR REPLACE VIEW command.

SQL CREATE OR REPLACE VIEW Syntax

```
CREATE OR REPLACE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

The following SQL adds the "City" column to the "Brazil Customers" view:

Example

```
CREATE OR REPLACE VIEW [Brazil  
Customers] AS  
SELECT CustomerName, ContactName, City  
FROM Customers  
WHERE Country = "Brazil";
```

[Try it Yourself »](#)

SQL Dropping a View

A view is deleted with the DROP VIEW command.

SQL DROP VIEW Syntax

```
DROP VIEW view_name;
```

The following SQL drops the "Brazil Customers" view:

Example

```
DROP VIEW [Brazil Customers];
```

[Try it Yourself »](#)

[« Previous](#)

[Next »](#)

Top Tutorials	Top References	Top Examples	Web Certificates
HTML Tutorial	HTML Reference	HTML Examples	HTML Certificate
CSS Tutorial	CSS Reference	CSS Examples	CSS Certificate
JavaScript Tutorial	JavaScript Reference	JavaScript Examples	JavaScript Certificate
How To Tutorial	SQL Reference	How To Examples	SQL Certificate
SQL Tutorial	Python Reference	SQL Examples	Python Certificate
Python Tutorial	W3.CSS Reference	Python Examples	jQuery Certificate
W3.CSS Tutorial	Bootstrap Reference	W3.CSS Examples	PHP Certificate
Bootstrap Tutorial	PHP Reference	Bootstrap Examples	Bootstrap Certificate
PHP Tutorial	HTML Colors	PHP Examples	XML Certificate
jQuery Tutorial	jQuery Reference	jQuery Examples	
Java Tutorial	Java Reference	Java Examples	
C++ Tutorial	Angular Reference	XML Examples	
			Get Certified »

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2019 by

Refsnes Data. All Rights Reserved.

Powered by W3.CSS.

