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 statements and functions to a view and present the data as if the data were coming from one single table.

A view is created with the CREATE VIEW statement.

CREATE VIEW Syntax

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

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

MySQL 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';

We can query the view above as follows:

Example

SELECT * FROM [Brazil Customers];

The following SQL creates a view that selects every product in the "Products" table with a price higher than the average price:

Example

CREATE VIEW [Products Above Average Price] AS
SELECT ProductName, Price
FROM Products
WHERE Price > (SELECT AVG(Price) FROM Products);

We can query the view above as follows:

Example

SELECT * FROM [Products Above Average Price];

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

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';

MySQL Dropping a View

A view is deleted with the DROP VIEW statement.

DROP VIEW Syntax

DROP VIEW *view_name*;

The following SQL drops the "Brazil Customers" view:

Example

DROP VIEW [Brazil Customers];