

# SQLITE - VIEWS

A view is nothing more than a SQLite statement that is stored in the database with an associated name. A view is actually a composition of a table in the form of a predefined SQLite query.

A view can contain all rows of a table or selected rows from one or more tables. A view can be created from one or many tables which depends on the written SQLite query to create a view.

Views which are kind of virtual tables, allow users to do the following:

- Structure data in a way that users or classes of users find natural or intuitive.
- Restrict access to the data such that a user can only see limited data instead of complete table.
- Summarize data from various tables which can be used to generate reports.

SQLite views are read-only and so you may not execute a DELETE, INSERT or UPDATE statement on a view. But you can create a trigger on a view that fires on an attempt to DELETE, INSERT, or UPDATE a view and do what you need in the body of the trigger.

## Creating Views:

The SQLite views are created using the **CREATE VIEW** statement. The SQLite views can be created from a single table, multiple tables, or another view.

The basic CREATE VIEW syntax is as follows:

```
CREATE [TEMP | TEMPORARY] VIEW view_name AS
SELECT column1, column2.....
FROM table_name
WHERE [condition];
```

You can include multiple tables in your SELECT statement in very similar way as you use them in normal SQL SELECT query. If the optional TEMP or TEMPORARY keyword is present, the view will be created in the temp database.

## Example:

Consider [COMPANY](#) table is having 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

Now, following is an example to create a view from COMPANY table. This view would be used to have only few columns from COMPANY table:

```
sqlite> CREATE VIEW COMPANY_VIEW AS
SELECT ID, NAME, AGE
FROM COMPANY;
```

Now, you can query COMPANY\_VIEW in similar way as you query an actual table. Following is the example:

```
sqlite> SELECT * FROM COMPANY_VIEW;
```

This would produce the following result:

ID	NAME	AGE
1	Paul	32
2	Allen	25
3	Teddy	23
4	Mark	25
5	David	27
6	Kim	22
7	James	24

## Dropping Views:

To drop a view, simply use the DROP VIEW statement with the **view\_name**. The basic DROP VIEW syntax is as follows:

```
sqlite> DROP VIEW view_name;
```

Following command will delete COMPANY\_VIEW view, which we created in the last section:

```
sqlite> DROP VIEW COMPANY_VIEW;
```