SQLite Inner Join

Summary: this tutorial shows you how to use SQLite inner join clause to query data from multiple tables.

Introduction to SQLite inner join clause

In relational databases, data is often distributed in many related tables. A table is associated with another table using foreign keys (https://www.sqlitetutorial.net/sqlite-foreign-key/) .

To query data (https://www.sqlitetutorial.net/sqlite-select/) from multiple tables, you use `INNER JOIN` clause. The `INNER JOIN` clause combines columns from correlated tables.

Suppose you have two tables: A and B.

A has a1, a2, and f columns. B has b1, b2, and f column. The A table links to the B table using a foreign key column named f.

The following illustrates the syntax of the inner join clause:

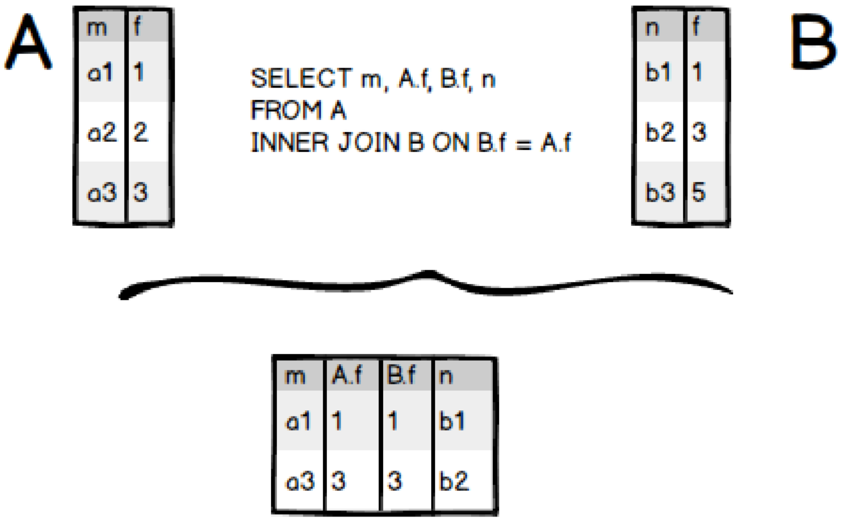
```
SELECT a1, a2, b1, b2
FROM A
INNER JOIN B on B.f = A.f;
```

For each row in the A table, the `INNER JOIN` clause compares the value of the f column with the value of the f column in the B table. If the value of the f column in the A table equals the value of the f column in the B table, it combines data from a1, a2, b1, b2, columns and includes this row in the result set.

In other words, the `INNER JOIN` clause returns rows from the A table that has the corresponding row in B table.

This logic is applied if you join more than 2 tables.

See the following example.



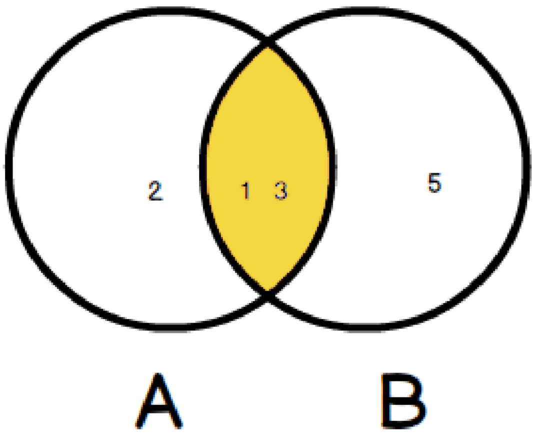
m	f
a1	1
a2	2
a3	3

n	f
b1	1
b2	3
b3	5

m	A.f	B.f	n
a1	1	1	b1
a3	3	3	b2

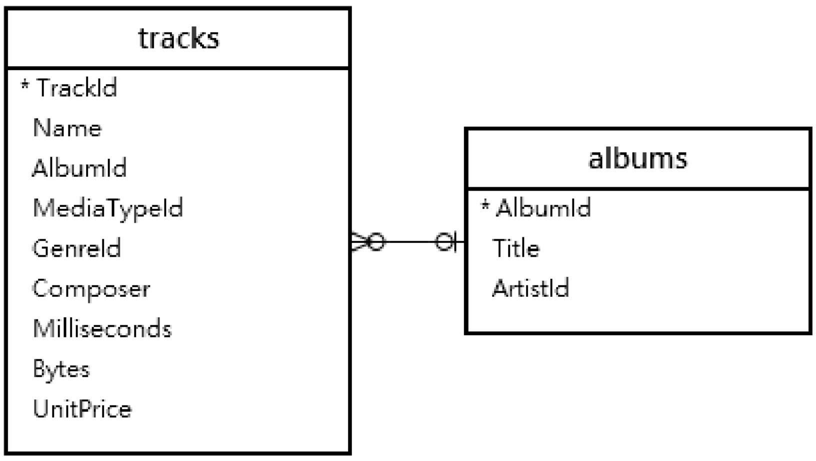
Only the rows in the A table: (a1,1), (a3,3) have the corresponding rows in the B table (b1,1), (b2,3) are included in the result set.

The following diagram illustrates the `INNER JOIN` clause:



SQLite INNER JOIN examples

Let's take a look at the `tracks` and `albums` tables in the sample database (https://www.sqlitetutorial.net/sqlite-sample-database/) . The `tracks` table links to the `albums` table via `AlbumId` column.

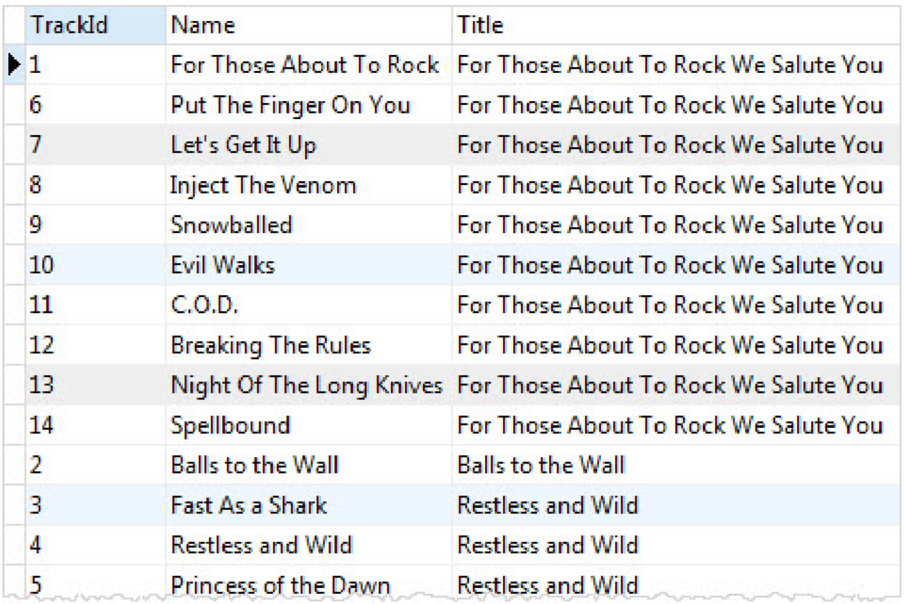


In the `tracks` table, the `AlbumId` column is a foreign key. And in the `albums` table, the `AlbumId` is the primary key (https://www.sqlitetutorial.net/sqlite-primary-key/) .

To query data from both `tracks` and `albums` tables, you use the following statement:

```
SELECT
        trackid,
        name,
        title
FROM
        tracks
INNER JOIN albums ON albums.albumid = tracks.albumid;
```

Try It ›



TrackId	Name	Title
1	For Those About To Rock	For Those About To Rock We Salute You
6	Put The Finger On You	For Those About To Rock We Salute You
7	Let's Get It Up	For Those About To Rock We Salute You
8	Inject The Venom	For Those About To Rock We Salute You
9	Snowballed	For Those About To Rock We Salute You
10	Evil Walks	For Those About To Rock We Salute You
11	C.O.D.	For Those About To Rock We Salute You
12	Breaking The Rules	For Those About To Rock We Salute You
13	Night Of The Long Knives	For Those About To Rock We Salute You
14	Spellbound	For Those About To Rock We Salute You
2	Balls to the Wall	Balls to the Wall
3	Fast As a Shark	Restless and Wild
4	Restless and Wild	Restless and Wild
5	Princess of the Dawn	Restless and Wild

For each row in the tracks table, SQLite uses the value in the `albumid` column of the `tracks` table to compare with the value in the `albumid` of the `albums` table. If SQLite finds a match, it combines data of rows in both tables in the result set.

You can include the `AlbumId` columns from both tables in the final result set to see the effect.

```
SELECT
    trackid,
    name,
    tracks.albumid AS album_id_tracks,
    albums.albumid AS album_id_albums,
    title
FROM
    tracks
    INNER JOIN albums ON albums.albumid = tracks.albumid;
```

Try It ›

SQLite inner join – 3 tables example

See the following tables: `tracks` `albums` and `artists`

One track belongs to one album and one album have many tracks. The `tracks` table associated with the `albums` table via `albumid` column.

One album belongs to one artist and one artist has one or many albums. The `albums` table links to the `artists` table via `artistid` column.

To query data from these tables, you need to use two inner join clauses in the `SELECT` (https://www.sqlitetutorial.net/sqlite-select/) statement as follows:

```
SELECT
    trackid,
    tracks.name AS track,
    albums.title AS album,
    artists.name AS artist
FROM
    tracks
    INNER JOIN albums ON albums.albumid = tracks.albumid
    INNER JOIN artists ON artists.artistid = albums.artistid;
```

Try It ›

You can use a WHERE clause (https://www.sqlitetutorial.net/sqlite-where/) to get the tracks and albums of the artist with id 10 as the following statement:

```
SELECT
        trackid,
        tracks.name AS Track,
        albums.title AS Album,
        artists.name AS Artist
FROM
        tracks
```

```
INNER JOIN albums ON albums.albumid = tracks.albumid
INNER JOIN artists ON artists.artistid = albums.artistid
WHERE
        artists.artistid = 10;
```

Try It ›

In this tutorial, you have learned how to use SQLite `INNER JOIN` clause to query data from multiple tables.