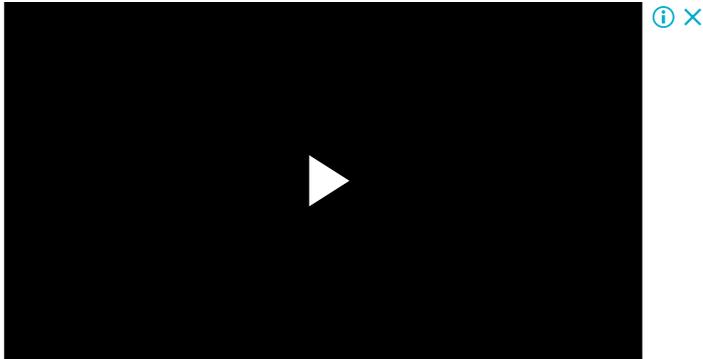


SQLite Join

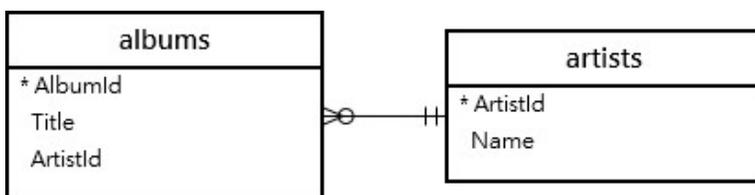


改用 Google Pixel 3a 手機

AR 步行導航讓你不再多走冤枉路。

Summary: in this tutorial, you will learn about various kinds of SQLite joins to query data from two or more tables.

For the demonstration, we will use the `artists` and `albums` tables from the [sample database](#).



An artist can have zero or many albums while an album belongs to one artist.

To query data from both `artists` and `albums` tables, you can use an `INNER JOIN`, `LEFT JOIN`, or `CROSS JOIN` clause. Each join clause determines how SQLite uses data from one table to match with rows in another table.

Note that SQLite doesn't directly support the `RIGHT JOIN` and `FULL OUTER JOIN`.



SQLite INNER JOIN

The following statement returns the album titles and their artist names:

```
1 SELECT
2     Title,
3     Name
4 FROM
5     albums
6 INNER JOIN artists
7     ON artists.ArtistId = albums.ArtistId;
```

Here is the partial output:

Title	Name
For Those About To Rock We Salute You	AC/DC
Balls to the Wall	Accept
Restless and Wild	Accept
Let There Be Rock	AC/DC
Big Ones	Aerosmith
Jagged Little Pill	Alanis Morissette
Facelift	Alice In Chains
Warner 25 Anos	Antônio Carlos Jobim
Plays Metallica By Four Cellos	Apocalyptica
Audioslave	Audioslave
Out Of Exile	Audioslave
BackBeat Soundtrack	BackBeat
The Best Of Billy Cobham	Billy Cobham
Alcohol Fueled Brewtality Live! [Disc 1]	Black Label Society
Alcohol Fueled Brewtality Live! [Disc 2]	Black Label Society
Black Sabbath	Black Sabbath
Black Sabbath Vol. 4 (Remaster)	Black Sabbath

In this example, the `INNER JOIN` clause matches each row from the `albums` table with every row from the `artists` table based on the join condition (`artists.ArtistId = albums.ArtistId`) specified after the `ON` keyword.

If the join condition evaluates to true (or 1), the columns of rows from both `albums` and `artists` tables are included in the result set.

This query uses table aliases (`l` for the `albums` table and `r` for `artists` table) to shorten the query:

```
1 SELECT
2     l.Title,
3     r.Name
```

```
4 FROM
5   albums l
6 INNER JOIN artists r ON
7   r.ArtistId = l.ArtistId;
```

[HOME](#)

[VIEWS](#)

[INDEXES](#)

[TRIGGERS](#)

[FUNCTIONS](#) ▾

[INTERFACES](#) ▾

[TRY IT](#)

In case the column names of joined tables are the same e.g., `ArtistId`, you can use the `USING` syntax as follows:

```
1 SELECT
2   Title,
3   Name
4 FROM
5   albums
6 INNER JOIN artists USING(ArtistId);
```

The clause `USING(ArtistId)` is equivalent to the clause `ON artists.ArtistId = albums.ArtistId`.

SQLite `LEFT JOIN`

This statement selects the artist names and album titles from the `artists` and `albums` tables using the `LEFT JOIN` clause:

```
1 SELECT
2   Name,
3   Title
4 FROM
5   artists
6 LEFT JOIN albums ON
7   artists.ArtistId = albums.ArtistId
8 ORDER BY Name;
```

Here is the output:



Name	Title
A Cor Do Som	[NULL]
AC/DC	For Those About To Rock We Salute You
AC/DC	Let There Be Rock
Aaron Copland & London Symphony Orchestra	A Copland Celebration, Vol. I
Aaron Goldberg	Worlds
Academy of St. Martin in the Fields & Sir Neville Marriner	The World of Classical Favourites
Academy of St. Martin in the Fields Chamber Ensemble & Sir Neville Marriner	Sir Neville Marriner: A Celebration
Academy of St. Martin in the Fields, John Birch, Sir Neville Marriner	Fauré: Requiem, Ravel: Pavane & Others
Academy of St. Martin in the Fields, Sir Neville Marriner & John Birch	Bach: Orchestral Suites Nos. 1 - 4
Academy of St. Martin in the Fields, Sir Neville Marriner & John Birch	[NULL]
Accept	Balls to the Wall
Accept	Restless and Wild
Adrian Leaper & Doreen de Feis	Górecki: Symphony No. 3
Aerosmith	Big Ones
Aerosmith & Sierra Leone's Refugee Allstars	[NULL]
Aisha Duo	Quiet Songs
Alanis Morissette	Jagged Little Pill

The `LEFT JOIN` clause selects data starting from the left table (`artists`) and matching rows in the right table (`albums`) based on the join condition (`artists.ArtistId = albums.ArtistId`).

The left join returns all rows from the `artists` table (or left table) and the matching rows from the `albums` table (or right table).

If a row from the left table doesn't have a matching row in the right table, SQLite includes columns of the rows in the left table and `NULL` for the columns of the right table.

Similar to the `INNER JOIN` clause, you can use the `USING` syntax for the join condition as follows:

```

1 SELECT
2   Name,
3   Title
4 FROM
5   artists
6 LEFT JOIN albums USING (ArtistId)
7 ORDER BY
8   Name;
```

If you want to find artists who don't have any albums, you can add a `WHERE` clause as shown in the following query:

```

1 SELECT
2   Name,
```

```

3  Title
4  FROM
5  artists
6  LEFT JOIN albums ON
7  artists.ArtistId = albums.ArtistId
8  WHERE Title IS NULL
9  ORDER BY Name;

```

This picture shows the partial output:

Name	Title
A Cor Do Som	[NULL]
Academy of St. Martin in the Fields, Sir Neville Marriner & William Bennett	[NULL]
Aerosmith & Sierra Leone's Refugee Allstars	[NULL]
Avril Lavigne	[NULL]
Azymuth	[NULL]
Baby Consuelo	[NULL]
Banda Black Rio	[NULL]
Barão Vermelho	[NULL]
Bebel Gilberto	[NULL]
Ben Harper	[NULL]
Big & Rich	[NULL]
Black Eyed Peas	[NULL]
Charlie Brown Jr.	[NULL]
Christina Aguilera featuring BigElf	[NULL]
Corinne Bailey Rae	[NULL]
DJ Dolores & Orchestra Santa Massa	[NULL]

Generally, this type of query allows you to find rows that are available in the left table but don't have corresponding rows in the right table.

Note that `LEFT JOIN` and `LEFT OUTER JOIN` are synonyms.

SQLite `CROSS JOIN`

The `CROSS JOIN` clause creates a [Cartesian product](#) of rows from the joined tables.

Unlike the `INNER JOIN` and `LEFT JOIN` clauses, a `CROSS JOIN` doesn't have a join condition. Here is the basic syntax of the `CROSS JOIN` clause:

```

1  SELECT
2  select_list
3  FROM table1
4  CROSS JOIN table2;

```

The `CROSS JOIN` combines every row from the first table (`table1`) with every row from the second table (`table2`) to form the result set.

If the first table has `M` rows, the second table has `N` rows, the final result will have `M*N` rows. [TRY IT](#)

A practical example of the `CROSS JOIN` clause is to combine two sets of data for forming an initial data set for further processing. For example, you have a list of products and months, and you want to make a plan when you can sell which products.

The following script creates the `products` and `calendars` tables:

```
1 CREATE TABLE products(  
2     product text NOT null  
3 );  
4  
5 INSERT INTO products(product)  
6 VALUES('P1'),('P2'),('P3');  
7  
8  
9  
10 CREATE TABLE calendars(  
11     y int NOT NULL,  
12     m int NOT NULL  
13 );  
14  
15 INSERT INTO calendars(y,m)  
16 VALUES  
17     (2019,1),  
18     (2019,2),  
19     (2019,3),  
20     (2019,4),  
21     (2019,5),  
22     (2019,6),  
23     (2019,7),  
24     (2019,8),  
25     (2019,9),  
26     (2019,10),  
27     (2019,11),  
28     (2019,12);
```

This query uses the `CROSS JOIN` clause to combine the products with the months:

```
1 SELECT *  
2 FROM products  
3 CROSS JOIN calendars;
```

Here is the output:

product	y	m
A	2,019	1
A	2,019	2
A	2,019	3
A	2,019	4
A	2,019	5
A	2,019	6
A	2,019	7
A	2,019	8
A	2,019	9
A	2,019	10
A	2,019	11
A	2,019	12
B	2,019	1
B	2,019	2
B	2,019	3
B	2,019	4
B	2,019	5
B	2,019	6
B	2,019	7
B	2,019	8
B	2,019	9
B	2,019	10
B	2,019	11
B	2,019	12
C	2,019	1
C	2,019	2
C	2,019	3
C	2,019	4
C	2,019	5
C	2,019	6
C	2,019	7
C	2,019	8
C	2,019	9
C	2,019	10
C	2,019	11
C	2,019	12

HOME

VIEWS

INDEXES

TRIGGERS

FUNCTIONS

INTERFACES

TRY IT

In this tutorial, you have learned various kind of SQLite joins that allow you to query from multiple tables.

Was this tutorial helpful ?

Yes

No



DIY玩手作 · 共度好時光

花花手染推薦 · 限時優惠倒數中！

◀◀ Previous Tutorial:
[SQLite IS NULL](#)

Next Tutorial: [SQLite Inner Join](#) ▶▶

Search this website

JGDQ 聯達外語

學費返還 勵學方案

12/1起停賣!!

最後活動 倒數!!!

GETTING STARTED

[What Is SQLite](#)

[Download & Install SQLite](#)

[SQLite Sample Database](#)



[HOME](#)

[VIEWS](#)

[INDEXES](#)

[TRIGGERS](#)

SQLITE TUTORIAL

[FUNCTIONS](#) ▾

[INTERFACES](#) ▾

[TRY IT](#)

[SQLite Select](#)

[SQLite Order By](#)

[SQLite Select Distinct](#)

[SQLite Where](#)

[SQLite Limit](#)

[SQLite BETWEEN](#)

[SQLite IN](#)

[SQLite Like](#)

[SQLite IS NULL](#)

[SQLite GLOB](#)

[SQLite Join](#)

[SQLite Inner Join](#)

[SQLite Left Join](#)

[SQLite Cross Join](#)

[SQLite Self-Join](#)

[SQLite Full Outer Join](#)

[SQLite Group By](#)

[SQLite Having](#)

[SQLite Union](#)

[SQLite Except](#)

[SQLite Intersect](#)

[SQLite Subquery](#)

[SQLite EXISTS](#)

[SQLite Case](#)

[SQLite Insert](#)



[HOME](#)

[VIEWS](#)

[INDEXES](#)

[TRIGGERS](#)

[SQLite Update](#)

[SQLite Delete](#)

[SQLite Replace](#)

[FUNCTIONS](#) ▾

[INTERFACES](#) ▾

[TRY IT](#)

[SQLite Transaction](#)



SQLITE DATA DEFINITION

[SQLite Data Types](#)

[SQLite Date & Time](#)

[SQLite Create Table](#)

[SQLite Primary Key](#)

[SQLite Foreign Key](#)

[SQLite NOT NULL Constraint](#)

[SQLite UNIQUE Constraint](#)

[SQLite CHECK Constraint](#)

[SQLite AUTOINCREMENT](#)

[SQLite Alter Table](#)

[SQLite Rename Column](#)

[SQLite Drop Table](#)

[SQLite Create View](#)



[HOME](#)

[VIEWS](#)

[INDEXES](#)

[TRIGGERS](#)

[SQLite Drop View](#)

[SQLite Index](#)

[SQLite Expression-based Index](#)

[FUNCTIONS](#) ▾

[INTERFACES](#) ▾

[TRY IT](#)

[SQLite Trigger](#)

[SQLite VACUUM](#)

[SQLite Transaction](#)

[SQLite Full-text Search](#)



SQLITE TOOLS

[SQLite Commands](#)

[SQLite Show Tables](#)

[SQLite Describe Table](#)

[SQLite Dump](#)

[SQLite Import CSV](#)

[SQLite Export CSV](#)

SQLITE FUNCTIONS

[SQLite AVG](#)

[SQLite COUNT](#)



[SQLite MAX](#)[SQLite MIN](#)[SQLite SUM
FUNCTIONS](#) ▾[INTERFACES](#) ▾[TRY IT](#)[HOME](#)[VIEWS](#)[INDEXES](#)[TRIGGERS](#)

[SQLITE INTERFACES](#)[SQLite PHP](#)[SQLite Node.js](#)[SQLite Java](#)[SQLite Python](#)

ABOUT SQLITE TUTORIAL

SQLite Tutorial website helps you master SQLite quickly and easily. It explains the complex concepts in simple and easy-to-understand ways so that you can both understand SQLite fast and know how to apply it in your software development work more effectively.

LOOKING FOR A TUTORIAL...

If you did not find the tutorial that you are looking for, you can use the following search box. In case the tutorial is not available, you can request for it using the [request for a SQLite tutorial form](#).

RECENT TUTORIALS

[SQLite Concat](#)[SQLite INSTEAD OF Triggers](#)[SQLite Join](#)[SQLite IS NULL](#)[SQLite Rename Column](#)[SQLite DROP VIEW](#)[SQLite Window Frame](#)[SQLite CUME_DIST](#)[SITE LINKS](#)[Home](#)[About](#)[Contact](#)[Resources](#)[Privacy Policy](#)

Copyright © 2019 [SQLite Tutorial](#). All rights Reserved.

[HOME](#)

[VIEWS](#)

[INDEXES](#)

[TRIGGERS](#)

[FUNCTIONS](#) ▾

[INTERFACES](#) ▾

[TRY IT](#)

