Tables

Announcements

Joining Tables

Joining Two Tables

A **JOIN** (or a comma) combines rows of two tables

titles

tconst	title	year
tt0058331	Mary Poppins	1964
tt14205554	KPop Demon Hunters	2025
tt8404614	The Two Popes	2019

ratings

tconst	avgRating	numVotes
tt0058331	7.8	195397
tt14205554	7.6	95933

SELECT * **FROM** titles **JOIN** ratings;

> **Cross Join:** all combos of a row from **A** & a row from **B**

tconst	title	year	tconst	avgRating	numVotes
tt0058331	Mary Poppins	1964	tt0058331	7.8	195397
tt0058331	Mary Poppins	1964	tt14205554	7.6	95933
tt14205554	KPop Demon Hunters	2025	tt0058331	7.8	195397
tt14205554	KPop Demon Hunters	2025	tt14205554	7.6	95933
tt8404614	The Two Popes	2019	tt0058331	7.8	195397
tt8404614	The Two Popes	2019	tt14205554	7.6	95933

Joining Two Tables

A `JOIN` (or a comma) combines rows of two tables

titles

tconst	title	year
tt0058331	Mary Poppins	1964
tt14205554	KPop Demon Hunters	2025
tt8404614	The Two Popes	2019

ratings

tconst	avgRating	numVotes
tt0058331	7.8	195397
tt14205554	7.6	95933

`SELECT * FROM titles JOIN ratings `**`ON titles.tconst=ratings.tconst;`**

tconst	title	year	tconst	avgRating	numVotes
tt0058331	Mary Poppins	1964	tt0058331	7.8	195397
tt14205554	KPop Demon Hunters	2025	tt14205554	7.6	95933

Match up a row from **titles** with a row from **ratings** only if the **tconst** values match

Joining Two Tables

```
SELECT title, averageRating FROM titles JOIN ratings ON titles.tconst=ratings.tconst LIMIT 3;


    title                           averageRating
    _____  _____
    The Kid                         8.2
    Nosferatu: A Symphony of Horror  7.8
    The Gold Rush                   8.1


SELECT title, averageRating FROM titles, ratings WHERE titles.tconst=ratings.tconst LIMIT 3;
```

> Can use a ,
> instead of **JOIN**

> Can put conditions
> in **WHERE**

```
Explicit syntax (JOIN and ON) is now far more common, but you'll find both forms in old code
```

Joining Two Tables: Actors and Characters

`SELECT` [expression] `FROM` [table] `JOIN` [table] `ON` [condition];

principals

tconst	ordering	nconst	category	character
tt0012349	1	nm0000122	actor	A Tramp
tt0012349	2	nm0701012	actor	The Woman

names

nconst	name	birth	death	profession	knownforTitles
nm0000002	Lauren Bacall	1924	2014	actress,miscellaneous,soundtrack	tt0037382,tt0075213,tt0038355,tt0117057
nm0000004	John Belushi	1949	1982	actor,writer,music_department	tt0072562,tt0077975,tt0080455,tt0078723

Select each actor's name and character, for each character they've played

`SELECT` names.name, principals.character

`FROM` principals `JOIN` names

`ON` principals.nconst=names.nconst   pollev.com/cs61a

`WHERE` principals.category = "actor";

String Expressions

（Demo）

String Expressions

String values can be combined to form longer strings

```
sqlite> SELECT "hello," || " world";
hello, world
```

Strings can be searched for a partial match

```
sqlite> SELECT title FROM titles where title LIKE "%dog%";
Dog Day Afternoon
Reservoir Dogs
Ghost Dog: The Way of the Samurai
Dogville
Slumdog Millionaire
Hachi: A Dog's Tale
Isle of Dogs
```

String manipulation is much more difficult in SQL than in Python

(Demo)

String Practice: Highest Rated Comedies

Get the `titles` and `ratings` of the `10` highest rated comedies

titles

tconst	title	year	runtime	genres
tt8404614	The Two Popes	2019	125	Biography,Drama
tt0012349	The Kid	1921	68	Comedy,Drama,Family

ratings

tconst	avgRating	numVotes
tt8613070	8.0	123438
tt8404614	7.6	144516

```
sqlite> SELECT "hello," || " world";
hello, world
```

title **LIKE** "%dog%"

Concatenate strings

Contains "dog" (case-insensitive)

SELECT title, avgRating

    FROM titles **JOIN** ratings **ON** titles.tconst=ratings.tconst

    WHERE genres **LIKE** "%Comedy%"

    ORDER BY averageRating **DESC limit 10**;

pollev.com/cs61a

Aliases and Dot Expressions

Joining a Table with Itself

A table can be joined by itself to compare one row to another row; aliases are required

Create a table of remakes that have the same title

```
SELECT old.title, old.year AS first, new.year AS second
  FROM titles AS old JOIN titles AS new
    ON old.title=new.title AND old.year < new.year;
```

Alias (**AS**) needed to disambiguate tables

pollev.com/cs61a

title	first	second
How to Train Your Dragon	2010	2025
The Girl with the Dragon Tattoo	2009	2011

(Demo)

Example: You Choose?

IMDB Tables

titles

tconst	title	year	runtime	genres
tt8404614	The Two Popes	2019	125	Biography,Drama
tt8503618	Hamilton	2020	2020	Biography,Drama,History

ratings

tconst	avgRating	numVotes
tt8613070	8.0	123438
tt8404614	7.6	144516

principals

tconst	ordering	nconst	category	character
tt0012349	1	nm0000122	actor	A Tramp
tt0012349	2	nm0701012	actor	The Woman

names

nconst	name	birth	death	profession	knownforTitles
nm0000002	Lauren Bacall	1924	2014	actress,miscellaneous,soundtrack	tt0037382,tt0075213,tt0038355,tt0117057
nm0000004	John Belushi	1949	1982	actor,writer,music_department	tt0072562,tt0077975,tt0080455,tt0078723

What question do you want to write a query to answer? pollev.com/cs61a

IMDB Tables

titles

tconst	title	year	runtime	genres
tt8404614	The Two Popes	2019	125	Biography,Drama
tt8503618	Hamilton	2020	2020	Biography,Drama,History

ratings

tconst	avgRating	numVotes
tt8613070	8.0	123438
tt8404614	7.6	144516

principals

tconst	ordering	nconst	category	character
tt0012349	1	nm0000122	actor	A Tramp
tt0012349	2	nm0701012	actor	The Woman

names

nconst	name	birth	death	profession	knownforTitles
nm0000002	Lauren Bacall	1924	2014	actress,miscellaneous,soundtrack	tt0037382,tt0075213,tt0038355,tt0117057
nm0000004	John Belushi	1949	1982	actor,writer,music_department	tt0072562,tt0077975,tt0080455,tt0078723

What question do you want to write a query to answer?

```
SELECT _____
    FROM _____
    WHERE _____
    ORDER BY _____ LIMIT 10;
```