

## Your first join - Inner Join with ON Clause

In this step, you are instructed to perform an inner join between the cities table on the left and the countries table on the right. The goal is to join the two tables using the country\_code column from cities and the code column from countries. This helps to combine relevant data from both tables.

Hint: In the cities table, the country code is named country\_code, but in the countries table, it's called code. Therefore, you'll have to use the ON clause to match these columns, as the USING command will not work in this case.

Below is the instruction and query output:

Learn / Courses / Joining Data In SQL

← Course Outline →

Daily XP 85

Exercise

### Your first join

Throughout this course, you'll be working with the `countries` database, which contains information about the most populous cities in the world, along with country-level economic, population, and geographic data. The database also contains information on languages spoken in each country.

You can see the different tables in this database to get a sense of what they contain by clicking on the corresponding tabs. Familiarize yourself with the fields that seem to be shared across tables before you continue with the course.

In this exercise, you'll use the `cities` and `countries` tables to build your first inner join. You'll start off by selecting all columns in step 1, performing your join in step 2, and then refining your join to choose specific columns in step 3.

Instructions 2/3 35 XP 2 3

- Perform an inner join with the `cities` table on the left and the `countries` table on the right; you **do not** need to alias tables here.
- Join **ON** the `country_code` and `code` columns, making sure you identify them with the correct table.

Take Hint (-10 XP)

query.sql

```
1 SELECT *
2 FROM cities
3 -- Inner join to countries
4 ---
5 -- Match on country codes
6 ---
```

Run Code

Submit Answer

query result

name	country_code	city_proper_pop	metroarea_pop	urbanarea_pop
Abidjan	CIV	4765000	null	4765000
Abu Dhabi	ARE	1145000	null	1145000
Abuja	NGA	1235880	6000000	1235880
Accra	GHA	2070463	4010054	2070463
Addis Ababa	ETH	3103673	4567857	3103673
Ahmedabad	IND	5570585	null	5570585
Alexandria	EGY	4616625	null	4616625
Algiers	DZA	3415811	5000000	3415811
Almaty	KAZ	1703404	null	1703404

Showing 100 out of 236 rows

## -- Full Answer in SQL

-- Select all columns from cities and countries

```
SELECT *
```

```
FROM cities
```

```
INNER JOIN countries
```

```
ON cities.country_code = countries.code;
```

The query uses an INNER JOIN to combine rows from the 'cities' and 'countries' tables based on a matching condition. The ON clause specifies that rows will be joined where 'cities.country\_code' matches 'countries.code'. Since the column names are different, the USING command cannot be used.