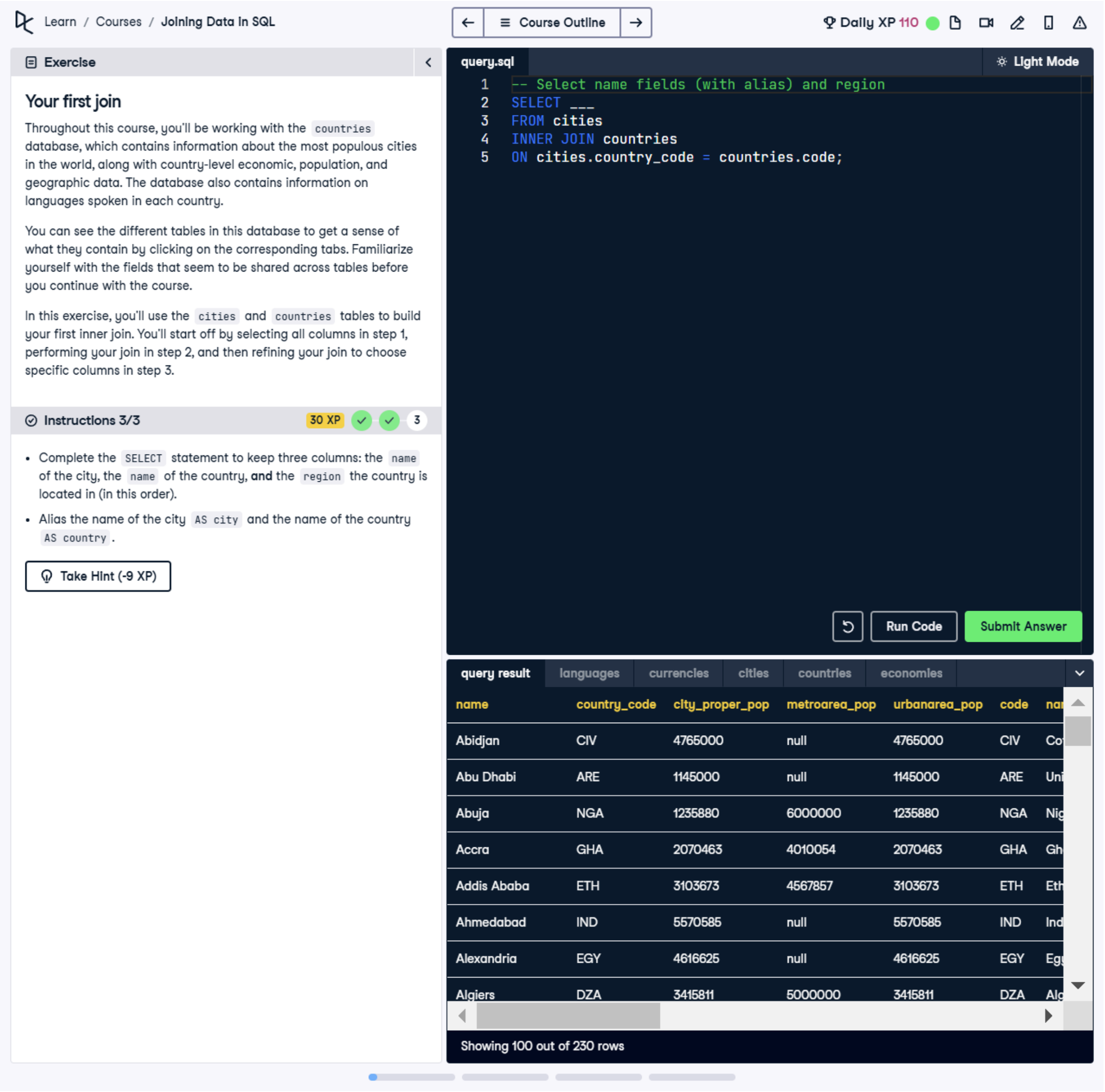
Your first join - SELECT with Aliases and Region

In this step, you are instructed to refine your query by selecting specific columns: the name of the city, the name of the country, and the region the country is located in. Additionally, you need to alias the name of the city as 'city' and the name of the country as 'country'.

Below is the instruction and query output:



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 3/3 30 XP ✓ ✓ 3

- Complete the `SELECT` statement to keep three columns: the `name` of the city, the `name` of the country, **and** the `region` the country is located in (in this order).
- Alias the name of the city `AS city` and the name of the country `AS country`.

```
-- Select name fields (with alias) and region
SELECT ___
FROM cities
INNER JOIN countries
ON cities.country_code = countries.code;
```

name	country_code	city_proper_pop	metroarea_pop	urbanarea_pop	code	nar
Abidjan	CIV	4765000	null	4765000	CIV	Co
Abu Dhabi	ARE	1145000	null	1145000	ARE	Uni
Abuja	NGA	1235880	6000000	1235880	NGA	Nig
Accra	GHA	2070463	4010054	2070463	GHA	Gh
Addis Ababa	ETH	3103673	4567857	3103673	ETH	Eth
Ahmedabad	IND	5570585	null	5570585	IND	Ind
Alexandria	EGY	4616625	null	4616625	EGY	Eg
Algiers	DZA	3415811	5000000	3415811	DZA	Alg

Showing 100 out of 230 rows

-- Full Answer in SQL

-- Select specific fields with aliases
SELECT cities.name AS city,

```
    countries.name AS country,
    countries.region
FROM cities
INNER JOIN countries
ON cities.country_code = countries.code;
```

The query selects three specific columns with aliases: the city name is aliased as 'city', the country name as 'country', and the region is included without an alias. The INNER JOIN combines rows from the 'cities' and 'countries' tables based on matching 'country_code' in 'cities' with 'code' in 'countries'.