

## Inspecting a Relationship - Fixing Ambiguous Column Reference

In this exercise, you are exploring how to inspect a relationship between the 'countries' table and the 'languages' table. An error occurred due to an ambiguous column reference because the 'name' column exists in both tables. To resolve this, the alias for the 'countries' table (c) is used to explicitly specify the column.

Below is the instruction and initial setup:

Learn / Courses / Joining Data In SQL

← Course Outline →

Daily XP 415

Exercise

### Inspecting a relationship

You've just identified that the `countries` table has a many-to-many relationship with the `languages` table. That is, many languages can be spoken in a country, and a language can be spoken in many countries.

But, what is the best way to query all the different languages spoken in a country? Or, all the countries that speak a certain language?

Instructions 2/4

25 XP

2 3 4

- Now add an alias `c` for the `countries` table and perform an inner join with the `languages` table, `l`, on the right; join on `code` in line 8 with the `USING` keyword; include the language name, aliased as `language`.

Take Hint (-7 XP)

query.sql

Light Mode

```
1 -- Select country and language names (aliased)
2 SELECT c.name AS country, ---
3 -- From countries (aliased)
4 FROM countries ---
5 -- Join to languages (aliased)
6 ---
7 -- Use code as the joining field with the USING keyword
8 ---;
```

↶ Run Code Submit Answer

query result

languages

populations

currencies

cities

countries

economies

country

Afghanistan

Netherlands

Albania

Algeria

American Samoa

Andorra

Angola

Antigua and Barbuda

United Arab Emirates

Showing 100 out of 205 rows

### -- Full Answer in SQL

```
SELECT c.name AS country,  
       l.name AS language  
FROM countries AS c  
INNER JOIN languages AS l  
USING (code);
```

Explanation: This query resolves the ambiguity by explicitly specifying the alias 'c' for the 'countries' table and 'l' for the 'languages' table. The 'name' column from 'countries' is selected as 'c.name' and aliased as 'country', while the 'name' column from 'languages' is selected as 'l.name' and aliased as 'language'. The USING clause joins the tables on the 'code' column, retrieving the country name and associated language for each match.