

## Inspecting a Relationship - Countries and Languages

In this exercise, you are exploring how to inspect a relationship between the 'countries' table and the 'languages' table. Specifically, this task involves selecting the country name aliased as 'country' from the 'countries' table.

Below is the instruction and initial setup:

The screenshot shows a web-based SQL learning interface. At the top, there's a navigation bar with 'Learn / Courses / Joining Data In SQL' and a 'Course Outline' button. The main content area is divided into two panels. The left panel, titled 'Exercise', contains the following text: '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 1/4' (with a progress bar showing 1 of 4 steps), and a list item: 'Select the country `name`, aliased as `country`, from the `countries` table.' Below this is a 'Take Hint (-7 XP)' button. The right panel is a code editor titled 'query.sql' with a 'Light Mode' toggle. It contains the SQL query: `-- Select country (aliased) from countries`. Below the code editor are 'Run Code' and 'Submit Answer' buttons. At the bottom, there's a 'query result' section with tabs for 'languages', 'populations', 'currencies', 'cities', 'countries', and 'economies'. The 'query result' tab is selected, showing 'No query executed yet...' and 'Showing 0 out of 0 rows'.

## -- Full Answer in SQL

-- Select country name with alias  
SELECT name AS country  
FROM countries;

This query selects the 'name' column from the 'countries' table and aliases it as 'country'. Aliasing the column as 'country' ensures clarity when referencing this column in query outputs.