Activity Overview

You've just learned about `JOIN`s and aliasing. Now, you'll practice writing queries that join multiple tables to build more robust datasets and use aliasing to make your SQL queries clearer. In this activity, you will load and examine two tables from the World Bank's International Education dataset in order to identify and understand their keys. You will also review `JOIN`s and write your own query that includes `JOIN`s and aliases. Finally, you'll use a `JOIN` to answer a specific question about the data.

By learning how to apply `JOIN` statements and aliasing, you'll be able to fully harness the power of relational databases by combining data from tables linked by keys.

Step-By-Step Instructions

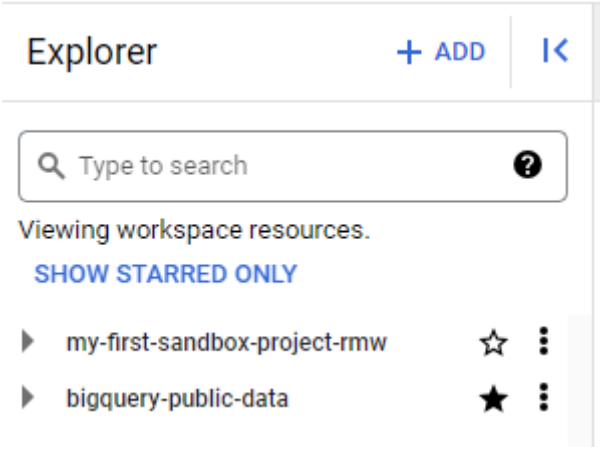
Follow the instructions to complete each step of the activity. Then answer the questions at the end of the activity before going to the next course item.

Step 1: Load the dataset

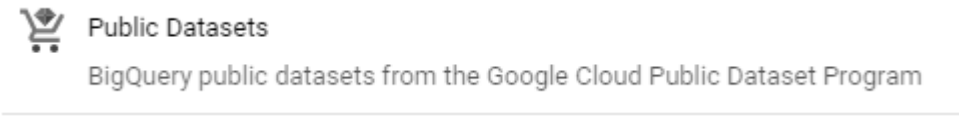
1. Log in to BigQuery Sandbox. If you have a free trial version of BigQuery, you can use that instead. On the BigQuery page, select the **Go to BigQuery** button.

Note: BigQuery Sandbox frequently updates its user interface. The latest changes may not be reflected in the screenshots presented in this activity, but the principles remain the same. Adapting to changes in software updates is an essential skill for data analysts, and it's helpful for you to practice troubleshooting. You can also reach out to your community of learners on the discussion forum for help.

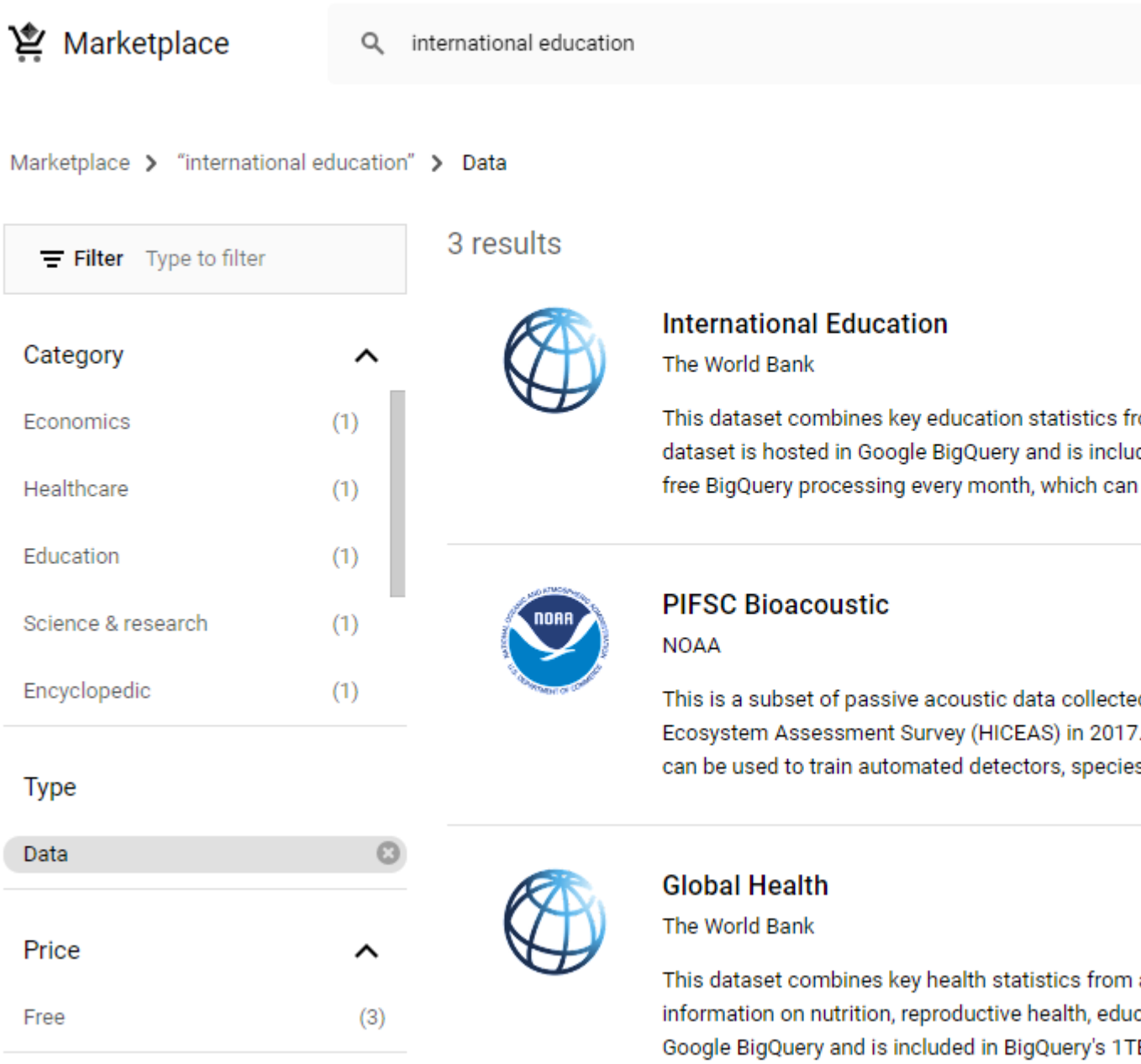
2. Now, you'll find the **Editor** interface. There are three major menus: the BigQuery navigation menu, the **Explorer** pane where you can search for datasets, and the details pane.



3. Select the + **ADD** button in the Explorer pane, then scroll down and select the **Public Datasets** option.



4. In the **Search Marketplace** text box, enter **international education** and press Return to find the search results.

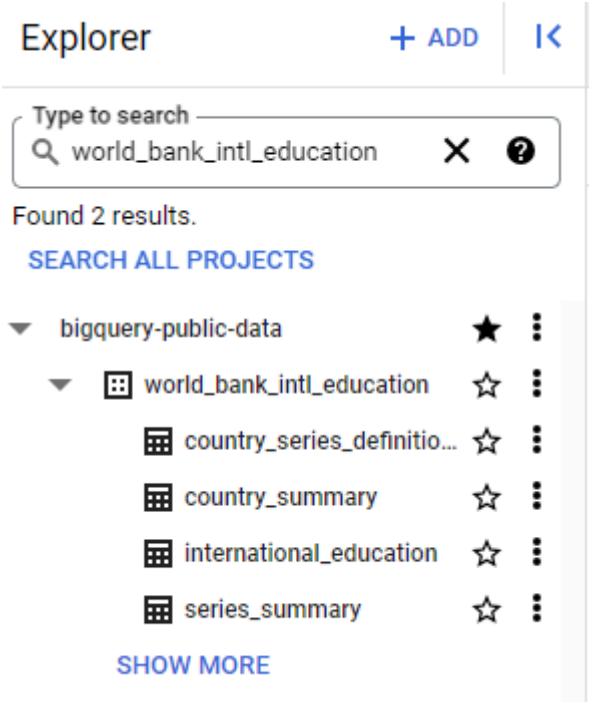


5. Select the World Bank's **International Education** dataset, which is the first result.

6. Select the **View Dataset** button to open the dataset in BigQuery in a new tab.

Note: You may want to star the **bigquery-public-data** resource in the **Explorer** pane. That way, you can access and browse the public datasets and tables more easily without having to navigate to the marketplace in the future.

7. In the Explorer pane, search for **world_bank_intl_education**. Expand the dataset to explore the tables it contains. You may also need to click on the **SHOW MORE** button for the additional tables to display.



Step 2: Identify and understand keys

Before you begin joining tables together, take a moment to consider how JOINs work: Two tables must be connected by their primary and foreign keys in order to join them. Keys are the most important elements of `JOIN`s—`JOIN`s function by combining tables based on those shared fields.

When designing a `JOIN` statement, these keys are listed in the `ON` statements as references to specific columns or fields within each table from the join: primary and foreign keys.

Consider two tables in the world_bank_intl_education dataset: **international_education** and **country_summary**. In order to understand how you might join these tables, take a moment to identify which columns you could use to combine them from each table. You can do that by examining the table schemas.

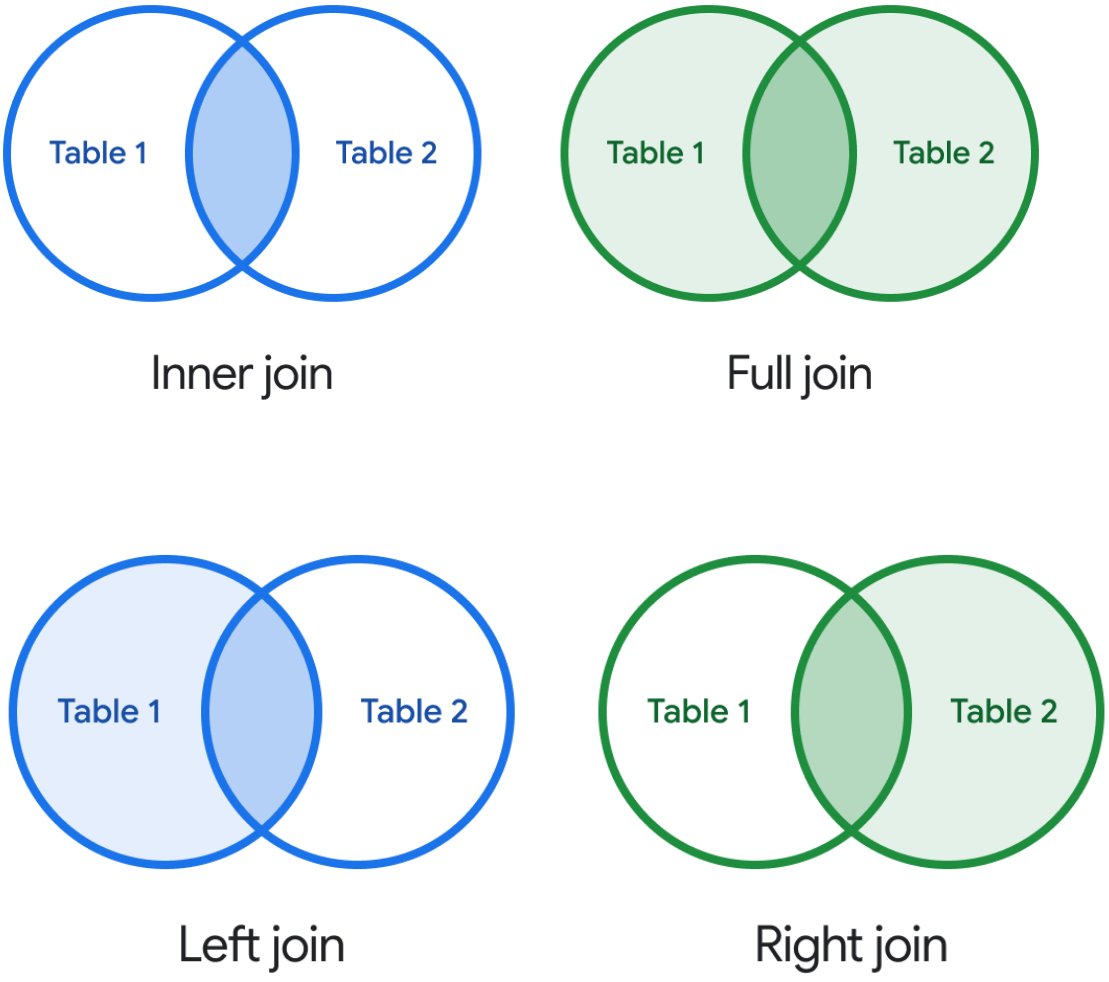
1. In the **Explorer** pane, select the **international_education** table. This will bring up the table's schema in the details pane. If the schema doesn't appear, select the **Schema** tab in the details pane. You might also check out the preview to view the data in the table.

2. Next, select the **country_summary** table and examine its schema. You'll find that the country_code column appears in both table schemas. You might also check out the preview.

What do you notice about the columns from each table?

Both of these tables share a field name: `country_code`. As you continue with this activity, you will use this common field for your JOIN as both your primary and foreign key. It's important to understand that foreign keys don't always have the same names across tables. If you're ever unsure if the columns are the same, you can always double-check. To do so, select the **Details** tab for each table and confirm that they contain the same kinds of information.

Step 3: Review JOINs

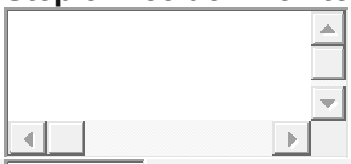


Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	region	secondary_edu_popu
1	South Asia	237541684.0
2	East Asia & Pacific	172016129.0
3	Sub-Saharan Africa	135639085.0
4	Europe & Central Asia	70181959.0
5	Latin America & Caribbean	67937467.0
6	Middle East & North Africa	44318682.0
7	North America	27003321.0

Step 6: Decide when to use INNER JOINs versus OUTER JOINs



1.
Question 1

Reflection

In the last query, you use a `LEFT JOIN` instead of an `INNER JOIN` to find the information you needed. Beneath the query results, you'll find that the number of rows in your joined table is 320. If you rerun the query with an `INNER JOIN` instead of a `LEFT JOIN`, how many rows does it return?

- 317
- 281
- 274

324

Correct

The query would return 317 rows. The row count drops from 320 with the `LEFT JOIN` to 317 with the `INNER JOIN`, because there are three rows where the team mascot is null. The `INNER JOIN` drops those rows which is why there is a difference between the two joins in this application.

1 / 1 point

2.
Question 2

In this activity, you used `JOIN` statements to combine d