Step-by-Step: Explore how JOINs work

This reading provides you with the steps the instructor performs in the following video, Explore how JOINs work. The video teaches you how to use `JOIN` in SQL to aggregate data in databases.

Keep this step-by-step guide open as you watch the video. It can serve as a helpful reference tool if you need additional context or clarification while following the video steps. This is not a graded activity, but you can complete these steps to practice the skills demonstrated in the video.

What you'll need

In order to follow along with the instructor, you will need the employee dataset uploaded into your project space. If you haven't already uploaded this data, follow the instructions in the Upload the employee dataset to BigQuery reading.

Common `JOIN`s

This video explores exactly how `JOIN`s work. A `JOIN` is a SQL clause that is used to combine rows from two or more tables based on a related column. The instructor discusses the different types of `JOIN`s in more detail in the video; here's a quick reference you can review as you follow along:

- `INNER JOIN`: a function that returns records with matching values in both tables
- `LEFT JOIN`: a function that returns all the records from the left table (first mentioned) and only the matching records from the right table (second mentioned)
- `RIGHT JOIN`: a function that returns all records from the right table (second mentioned) and only the matching records from the left table (first mentioned).
- `OUTER JOIN`: a function that combines the `RIGHT JOIN` and `LEFT JOIN` to return all matching records in both tables.

Example 1: `INNER JOIN`

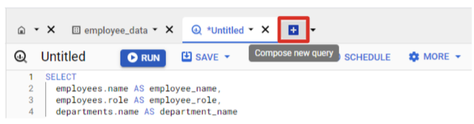
In the video, the instructor uses BigQuery to join the employees and departments tables. The following steps take you through typing the query into the query window. If you prefer, you can copy and paste the following query into the query window instead.

1. In BigQuery, select the **COMPOSE A NEW QUERY** button. BigQuery opens a query window where you can enter your query. The instructor has already executed some queries so their starting line number may be different from yours. If you've just opened BigQuery, your line number will be 1. It's okay if your line numbers aren't the same as what's in the video.
2. In line 1 of the query window, enter **SELECT** and then press **Enter**.
3. Press Tab and then in line 2 enter **employees.name AS employee_name**, then press **Enter**.
4. In line 3, enter **employees.role AS employee_role**, then press **Enter**.
5. In line 4, enter **departments.name AS department_name**, then press **Enter**.
6. In line 5, press **Backspace** to stop indenting, enter **FROM**, then press **Enter**.
7. In line 6, press **Tab**, then enter **[your-project-name].employee_data.employees**, replacing **[your-project-name]** with the name of the BigQuery project to which you added the **employee_data** dataset.
8. Continuing in line 6, enter **AS employees**. Press **Enter**.
9. In line 7, press **Backspace** to stop indenting, enter **INNER JOIN**, then press **Enter**.
10. In line 8, press **Tab**, then enter **[your-project-name].employee_data.departments**, replacing **[your-project-name]** with the name of the BigQuery project to which you added the **employee_data** dataset.
11. Continuing in line 8, add **AS departments**. Press **Enter**.
12. In line 9, enter **ON employees.department_id = departments.department_id**.
13. Run the query by selecting the **Run** button.

```
SELECT
    employees.name AS employee_name,
    employees.role AS employee_role,
    departments.name AS department_name
FROM
    [your-project-id].employee_data.employees AS employees
INNER JOIN
    [your-project-id].employee_data.departments AS departments
    ON employees.department_id = departments.department_id
```

Example 2: `LEFT JOIN`

Start a new query in BigQuery by opening a new query window. To open a new query window, select the **+** button.



Now, you'll create a new query that uses `LEFT JOIN`.

1. Using the query window tabs to navigate between queries, copy and paste the query from Example 1 into the new query window.
2. In line 7, replace `INNER JOIN` with `LEFT JOIN`.
3. Run the query by selecting the **Run** button.

If you prefer, you can copy and paste the following query into the query window in BigQuery.

```
SELECT
    employees.name AS employee_name,
    employees.role AS employee_role,
    departments.name AS department_name
FROM
    [your-project-id].employee_data.employees AS employees
LEFT JOIN
    [your-project-id].employee_data.departments AS departments
    ON employees.department_id = departments.department_id
```

Example 3: `RIGHT JOIN`

Now, you'll create a new query that uses `RIGHT JOIN`. Open a new query window.

1. Using the query window tabs to navigate between queries, copy and paste the query from Example 2 into the new query window.
2. In line 7, replace `LEFT JOIN` with `RIGHT JOIN`.
3. Run the query by selecting the **Run** button.

If you prefer, you can copy and paste the following query into the query window in BigQuery.

```
SELECT
    employees.name AS employee_name,
    employees.role AS employee_role,
    departments.name AS department_name
FROM
    [your-project-id].employee_data.employees AS employees
RIGHT JOIN
    [your-project-id].employee_data.departments AS departments
    ON employees.department_id = departments.department_id
```

Example 4: `FULL OUTER JOIN`

Now, you'll create a new query that uses `FULL OUTER JOIN`.

1. Using the query window tabs to navigate between queries, copy and paste the query from Example 3 into the new query window.
2. In line 7, replace `RIGHT JOIN` with `FULL OUTER JOIN`.
3. Run the query by selecting the **Run** button.

If you prefer, you can copy and paste the following query into the query window in BigQuery.

```
SELECT
    employees.name AS employee_name,
    employees.role AS employee_role,
    departments.name AS department_name
FROM
    [your-project-id].employee_data.employees AS employees
OUTER JOIN
    [your-project-id].employee_data.departments AS departments
      ON employees.department_id = departments.department_id
```