Data to Insights: Unioning and Joining Datasets v1.1 | Google Cloud Skills Boost

Qwiklabs : 7-9 minutes

Overview

JOINs enrich your dataset by potentially adding fields (horizontally). UNIONs append more data to your table (vertically). When you understand the relationships between your tables, use UNIONS to append records to a consolidated table, and JOINs to enrich your results with data from multiple sources.

This lab focuses on how to create new reporting tables using SQL JOINS and UNIONs.

Objectives

In this lab you learn how to perform the following tasks:

- Describe Unioning and Joining Datasets.
- Describe Joining Tables.
- Describe Working with NULLs.

Setup and requirements

For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Sign in to Qwiklabs using an **incognito window**.
2. Note the lab's access time (for example, `1:15:00`), and make sure you can finish within that time. There is no pause feature. You can restart if needed, but you have to start at the beginning.
3. When ready, click **Start lab**.
4. Note your lab credentials (**Username** and **Password**). You will use them to sign in to the Google Cloud Console.
5. Click **Open Google Console**.
6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts. If you use other credentials, you'll receive errors or **incur charges**.
7. Accept the terms and skip the recovery resource page.

Task 1. Practice unioning and joining datasets

Open BigQuery Console

1. In the Google Cloud Console, select **Navigation menu** > **BigQuery**.

The **Welcome to BigQuery in the Cloud Console** message box opens. This message box provides a link to the quickstart guide and lists UI updates.

2. Click **Done**.

3. Compose the query in BigQuery **EDITOR**.

4. Ensure `#standardSQL` is set as your first line of code.

5. Write a Query that will count the number of tax filings by calendar year for all IRS Form 990 filings.

6. Use the below partially-written query as a guide.

Hint: You will need to use Table Wildcards `*` with `_TABLE_SUFFIX`.

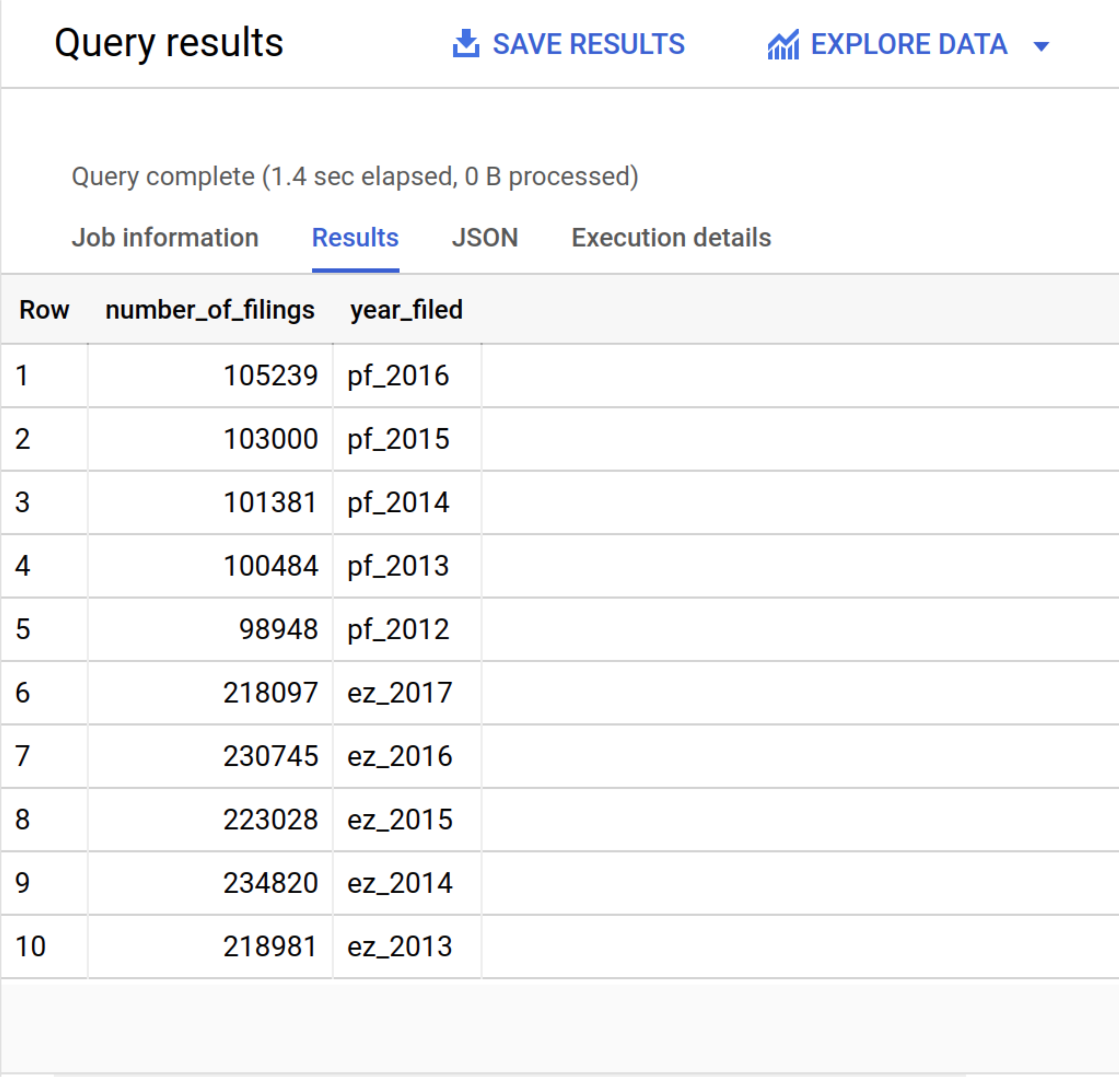
```
#standardSQL # UNION Wildcard and returning a table suffix SELECT COUNT(*) as number_of_filings, AS year_filed FROM `bigquery-public-data.irs_990.irs_990` GROUP BY year_filed ORDER BY year_filed DESC
```

7. Compare with the below solution:

```
#standardSQL # UNION Wildcard and returning a table suffix SELECT COUNT(*) as number_of_filings, _TABLE_SUFFIX AS year_filed FROM `bigquery-public-data.irs_990.irs_990_*` GROUP BY year_filed ORDER BY year_filed DESC
```

8. **Run** the query and confirm against the results below.

Result:



Query complete (1.4 sec elapsed, 0 B processed)

Row	number_of_filings	year_filed
1	105239	pf_2016
2	103000	pf_2015
3	101381	pf_2014
4	100484	pf_2013
5	98948	pf_2012
6	218097	ez_2017
7	230745	ez_2016
8	223028	ez_2015
9	234820	ez_2014
10	218981	ez_2013

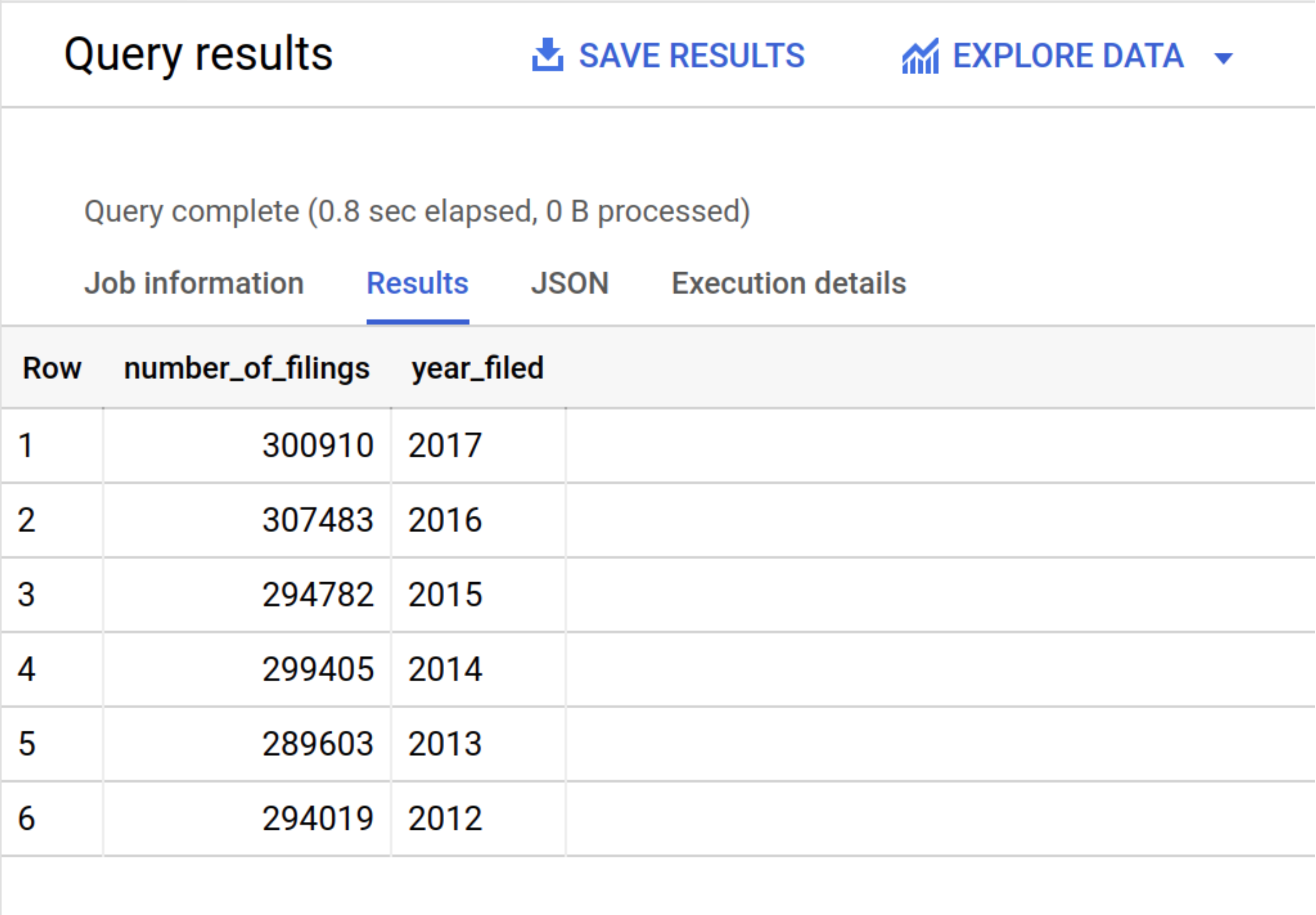
9. **Modify** the query you just wrote to only include the IRS tables with the following format: `irs_990_`YYYY (i.e. filter out pf, ez, ein). Start with the partially completed query below:

#standardSQL # UNION Wildcard and returning a table suffix SELECT COUNT(*) as number_of_filings, CONCAT(,_TABLE_SUFFIX) AS year_filed FROM `bigquery-public-data.irs_990.irs_990_*` GROUP BY year_filed ORDER BY year_filed DESC

10. Compare with the below solution:

#standardSQL # UNION Wildcard and returning a table suffix SELECT COUNT(*) as number_of_filings, CONCAT("2",_TABLE_SUFFIX) AS year_filed FROM `bigquery-public-data.irs_990.irs_990_2*` GROUP BY year_filed ORDER BY year_filed DESC

11. **Run** the query and confirm the result:



Query complete (0.8 sec elapsed, 0 B processed)

Row	number_of_filings	year_filed
1	300910	2017
2	307483	2016
3	294782	2015
4	299405	2014
5	289603	2013
6	294019	2012

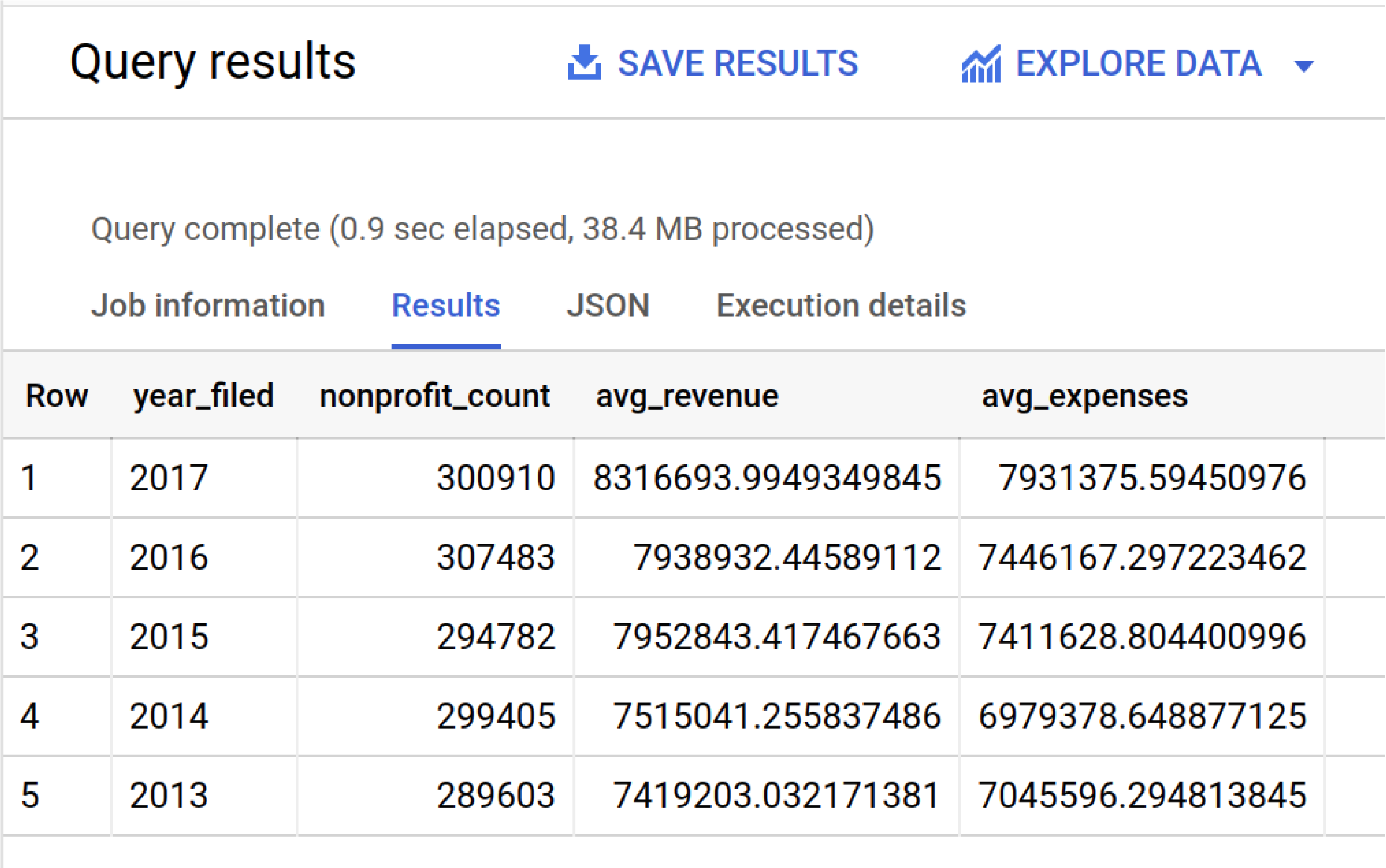
12. Lastly, modify your query to only include tax filings from tables on or after 2013. Also include average `totrevenue` and average `totfuncexpns` as additional metrics.

Hint: Consider using `_TABLE_SUFFIX` in a filter.

13. Compare with the below solution:

```
#standardSQL # count of filings, revenue, expenses since 2013 SELECT CONCAT("20",_TABLE_SUFFIX)
AS year_filed, COUNT(ein) AS nonprofit_count, AVG(totrevenue) AS avg_revenue, AVG(totfuncexpns) AS
avg_expenses FROM `bigquery-public-data.irs_990.irs_990_20*` WHERE _TABLE_SUFFIX >= '13'
GROUP BY year_filed ORDER BY year_filed DESC
```

14. **Run** the query and confirm the result:



Query complete (0.9 sec elapsed, 38.4 MB processed)

Row	year_filed	nonprofit_count	avg_revenue	avg_expenses
1	2017	300910	8316693.9949349845	7931375.59450976
2	2016	307483	7938932.44589112	7446167.297223462
3	2015	294782	7952843.417467663	7411628.804400996
4	2014	299405	7515041.255837486	6979378.648877125
5	2013	289603	7419203.032171381	7045596.294813845

Task 2. Practice joining tables

Find the Org Names of all EINs for 2015 with some revenue or expenses. You will need to join tax filing table data with the organization details table.

1. Start with the below query and fill in the tables, join condition, and any filter you will need:

```
#standard SQL # Find the Org Names of all EINs for 2015 with some revenue or expenses, limit 100
SELECT tax.ein AS tax_ein, org.ein AS org_ein, org.name, tax.totrevenue, tax.totfuncexpns FROM AS tax
JOIN AS org ON tax.ein = WHERE > 0 LIMIT 100;
```

2. Compare your query to the below solution:

```
#standardSQL # Find the Org Names of all EINs for 2015 with some revenue or expenses, limit 100
SELECT tax.ein AS tax_ein, org.ein AS org_ein, org.name, tax.totrevenue, tax.totfuncexpns FROM
`bigquery-public-data.irs_990.irs_990_2015` AS tax JOIN `bigquery-public-data.irs_990.irs_990_ein` AS org
ON tax.ein = org.ein WHERE tax.totrevenue + tax.totfuncexpns > 0 LIMIT 100;
```

3. **Run** the Query.

4. Confirm the results show 100 records, the names of the Organization, and at least some expenses or revenues.

5. Clear the BigQuery **EDITOR**.

Task 3. Practicing working with NULLs

Write a query to find where tax records exist for 2015 but no corresponding Org Name.

1. Fill out the partially written starter query below:

```
#standardSQL # Find where tax records exist for 2015 but no corresponding Org Name SELECT tax.ein AS
tax_ein, org.ein AS org_ein, org.name, tax.totrevenue, tax.totfuncexpns FROM `bigquery-public-
data.irs_990.irs_990_2015` tax FULL # Complete the JOIN `bigquery-public-data.irs_990.irs_990_ein` org
ON WHERE IS NULL # put tax.ein or org.ein to check here (one is correct)
```

2. Compare your solution to the below:

```
#standardSQL # Find where tax records exist for 2015 but no corresponding Org Name SELECT tax.ein AS
tax_ein, org.ein AS org_ein, org.name, tax.totrevenue, tax.totfuncexpns FROM `bigquery-public-
data.irs_990.irs_990_2015` tax FULL JOIN `bigquery-public-data.irs_990.irs_990_ein` org ON tax.ein =
org.ein WHERE org.ein IS NULL
```

3. **Run** the Query.

Question: How many tax filings occurred in 2015 but have no corresponding record in the Organization Details table?

Answer: 14,123 (your answer may be higher as new EIN numbers get added to the public base table)

Congratulations!

You have completed the **UNIONING and JOINING Datasets** lab.

Learning review

- Use Union Wildcards to treat multiple tables as a single group
- Use _TABLE_SUFFIX to filter wildcard tables and create calculated fields with the table name
- FULL JOINs (also called FULL OUTER JOINs) include all records from each joined table regardless of whether there are matches on the join key
- Having non-unique join keys can result in an unintentional CROSS JOIN (more output rows than input rows) which should be avoided
- Use COUNT() and GROUP BY to determine if a key field is indeed unique

End your lab

When you have completed your lab, click **End Lab**. Google Cloud Skills Boost removes the resources you've used and cleans the account for you.

You will be given an opportunity to rate the lab experience. Select the applicable number of stars, type a comment, and then click **Submit**.

The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral

- 4 stars = Satisfied
- 5 stars = Very satisfied

You can close the dialog box if you don't want to provide feedback.

For feedback, suggestions, or corrections, please use the **Support** tab.