Module 9

Joining and Merging Datasets

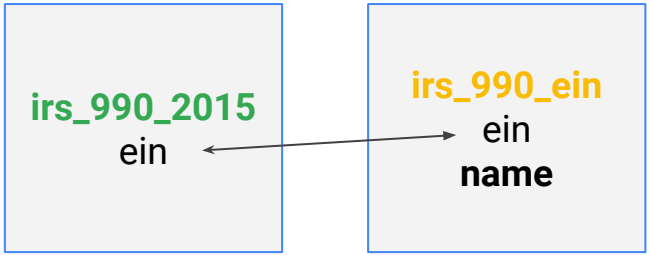
In this module we will:

- **Merge Historical Data Tables with UNION**
- Introduce Table Wildcards for Easy Merges
- Review Data Schemas: Linking Data Across Multiple Tables
- Walkthrough JOIN Examples and Pitfalls

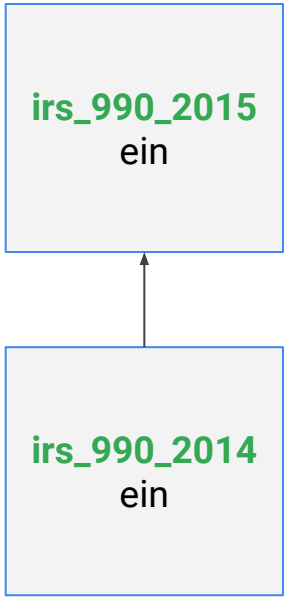
Google Cloud

Enriching your Dataset through JOINs and UNIONs

JOINs give you fields from different tables

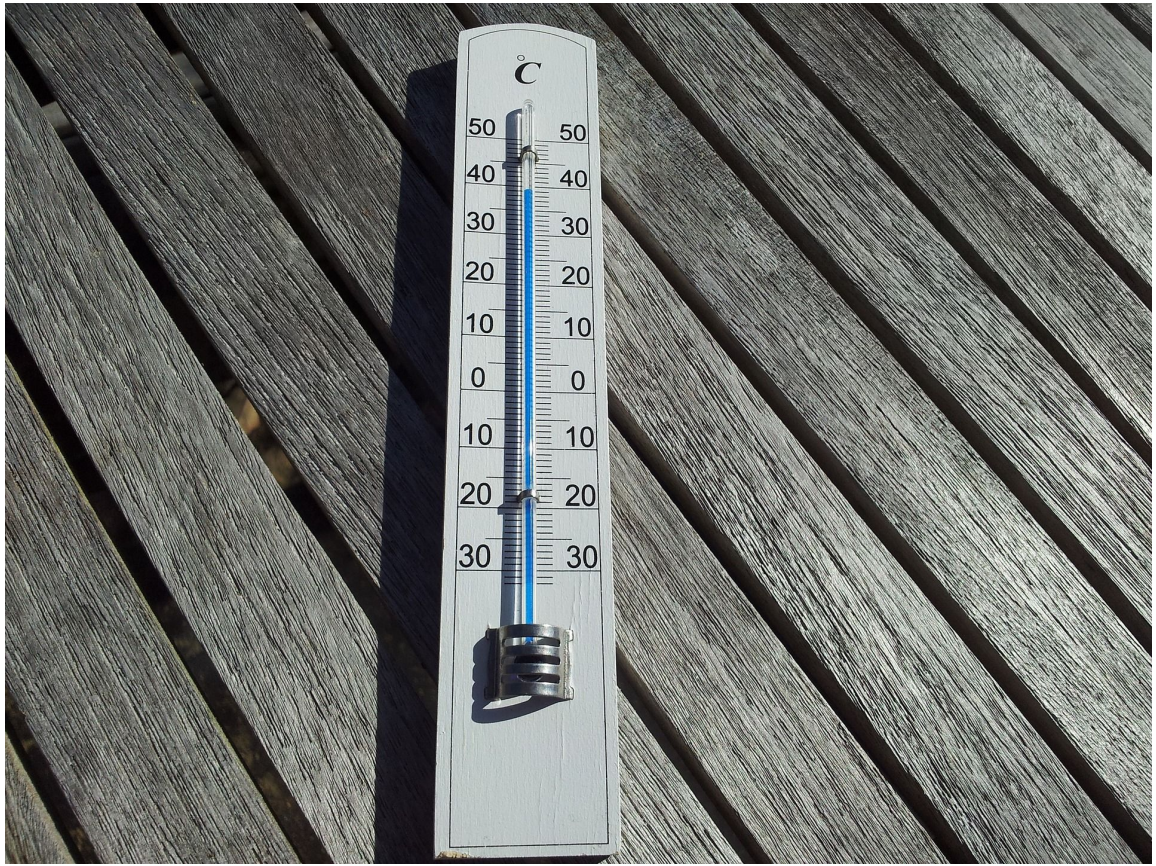


UNIONs add more records to your table



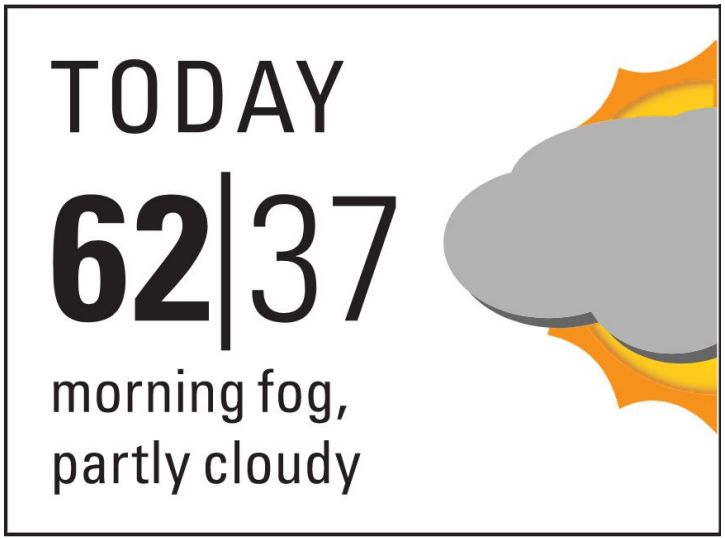
Walkthrough Example

Joining and Merging *Temperature* and *Weather Station* Data



Two Types of Tables in the **NOAA Weather Dataset**

Daily Temperature Readings



Weather Recording Station Locations

Victoria, Australia

Wake Island Harbor

Two Types of Tables in the **NOAA Weather Dataset**

Daily Temperature Readings

- noaa_gsod
  - **gsod1929**
  - gsod1930
  - gsod1931
  - gsod1932
  - gsod1933
  - gsod1934
  - gsod1935
  - gsod1936
  - gsod1937
  - gsod1938
  - gsod1939
  - gsod1940
  - gsod1941
  - gsod1942
  - gsod1943
  - gsod1944
  - gsod1945
  - gsod1946
  - gsod1947
  - gsod1948
  - gsod1949
  - gsod1950
  - gsod1951
  - gsod1952
  - gsod1953
  - gsod1954
  - gsod1955
  - gsod1956
  - gsod1956
  - gsod1957
  - gsod1958
  - gsod1959
  - gsod1960
  - gsod1961 … current
  - gsod1962
  - gsod1963
  - gsod1964
  - gsod1965
  - gsod1966
  - gsod1967
  - gsod1968
  - gsod1969

Weather Recording Station Locations

stations

Results | Explanation | Job Information

Row	usaf	wban	name	country	state	call	lat	lon	elev	begin	end
1	912450	41606	WAKE ISLAND AIRFLD	WQ	PC	PWAK	19.283	166.65	+0003.7	19451231	20100731
2	912460	41606	WAKE ISLAND AIRFIELD	WQ	UM	PWAK	19.283	166.65	+0007.0	20100801	20170805
3	999999	41606	WAKE ISLAND	WQ	PC	PWAK	19.283	166.65	+0003.7	19491031	19721231
4	912450	99999	WAKE ISLAND AIRFLD	WQ			19.283	166.65	+0004.0	20000101	20100818
5	997387	99999	WAKE ISLAND	WQ			19.28	166.62	+0005.0	20050517	20170804

Table | JSON

What is our **Unique Identifier** for a Weather Station?

```
#standardSQL
# Is usaf unique over time?
  SELECT
    COUNT(usaf) AS total_stations,
    COUNT(DISTINCT usaf) AS
distinct_stations
FROM
`bigquery-public-data.noaa_gsod.stations`;
```

Row	total_count	distinct_count
1	30016	26453

no

Weather Recording Station Locations

stations

Results | Explanation | Job Information

Row	usaf	wban	name	country	state	call	lat	lon	elev	begin	end
1	912450	41606	WAKE ISLAND AIRFLD	WQ	PC	PWAK	19.283	166.65	+0003.7	19451231	20100731
2	912460	41606	WAKE ISLAND AIRFIELD	WQ	UM	PWAK	19.283	166.65	+0007.0	20100801	20170805
3	999999	41606	WAKE ISLAND	WQ	PC	PWAK	19.283	166.65	+0003.7	19491031	19721231
4	912450	99999	WAKE ISLAND AIRFLD	WQ			19.283	166.65	+0004.0	20000101	20100818
5	997387	99999	WAKE ISLAND	WQ			19.28	166.62	+0005.0	20050517	20170804

Table | JSON

We need to use a **combination key**

Weather Recording Station Locations

```
#standardSQL
# Is usaf wban combo unique over time?
  SELECT
    COUNT(CONCAT(usaf,wban)) AS total_stations,
    COUNT(DISTINCT CONCAT(usaf,wban)) AS distinct_stations
FROM `bigquery-public-data.noaa_gsod.stations`;
```

stations

Row	total_stations	distinct_stations
1	30016	30016

Yes

Results | Explanation | Job Information

Row	usaf	wban	name	country	state	call	lat	lon	elev	begin	end
1	912450	41606	WAKE ISLAND AIRFLD	WQ	PC	PWAK	19.283	166.65	+0003.7	19451231	20100731
2	912460	41606	WAKE ISLAND AIRFIELD	WQ	UM	PWAK	19.283	166.65	+0007.0	20100801	20170805
3	999999	41606	WAKE ISLAND	WQ	PC	PWAK	19.283	166.65	+0003.7	19491031	19721231
4	912450	99999	WAKE ISLAND AIRFLD	WQ			19.283	166.65	+0004.0	20000101	20100818
5	997387	99999	WAKE ISLAND	WQ			19.28	166.62	+0005.0	20050517	20170804

Table | JSON

Join and Union your Data for Enriched Insights

Daily Temperature Readings

- noaa_gsod
  - **gsod1929**
  - gsod1930
  - gsod1931
  - gsod1932
  - gsod1933
  - gsod1934
  - gsod1935
  - gsod1936
  - gsod1937
  - gsod1938
  - gsod1939
  - gsod1940
  - gsod1941
  - gsod1942
  - gsod1943
  - gsod1944
  - gsod1945
  - gsod1946
  - gsod1947
  - gsod1948
  - gsod1949
  - gsod1950
  - gsod1951
  - gsod1952
  - gsod1953
  - gsod1954
  - gsod1955
  - gsod1956
  - gsod1956
  - gsod1957
  - gsod1958
  - gsod1959
  - gsod1960
  - gsod1961
  - gsod1962
  - gsod1963
  - gsod1964
  - gsod1965
  - gsod1966
  - gsod1967
  - gsod1968
  - gsod1969

… current

How are we going to JOIN so many tables?

Can’t we combine the temperature readings across years somehow?

Weather Recording Station Locations

stations

Introducing UNION for Vertically Merging your Data

Daily Temperature Readings

gsod1929

gsod1930

gsod1929

stn	wban	temp	year
030050	99999	49	1929
030050	99999	45.7	1929
030050	99999	48.2	1929

UNION

gsod1930

stn	wban	temp	year
037770	99999	50.7	1930
030910	99999	56	1930
038560	99999	53.2	1930

=

Gsod1929 UNION gsod1930

stn	wban	temp	year
030050	99999	49	**1929**
030050	99999	45.7	**1929**
030050	99999	48.2	**1929**
037770	99999	50.7	1930
030910	99999	56	1930
038560	99999	53.2	1930

Introducing UNION for Vertically Merging your Data

gsod1929

gsod1930

```
#standardSQL
SELECT
  stn,
  wban,
  temp,
  year
FROM

`bigquery-public-data.noaa_gsod.gsod1929`
  UNION DISTINCT
(SELECT stn, wban, temp, year FROM
`bigquery-public-data.noaa_gsod.gsod1930`)
```

Gsod1929 UNION gsod1930

stn	wban	temp	year
030050	99999	49	**1929**
030050	99999	45.7	**1929**
030050	99999	48.2	**1929**
037770	99999	50.7	1930
030910	99999	56	1930
038560	99999	53.2	1930

UNION DISTINCT removes duplicates whereas UNION ALL keeps every record

Wait a minute....

- **gsod1929**
- gsod1930
- gsod1931
- gsod1932
- gsod1933
- gsod1934
- gsod1935
- gsod1936
- gsod1937
- gsod1938
- gsod1939

```
#standardSQL
SELECT
  stn,
  wban,
  temp,
  year
FROM
  `bigquery-public-data.noaa_gsod.gsod1929`
  UNION DISTINCT
(SELECT stn,wban,temp,year FROM
`bigquery-public-data.noaa_gsod.gsod1930`)
  UNION DISTINCT
(SELECT stn,wban,temp,year FROM
`bigquery-public-data.noaa_gsod.gsod1931`)
  UNION DISTINCT
(SELECT stn,wban,temp,year FROM
`bigquery-public-data.noaa_gsod.gsod1932`)
# This is getting out of hand...
```

.. I don't want to type 100 Unions

Module 9

Joining and Merging Datasets

In this module we will:

- Merge Historical Data Tables with UNION
- **Introduce Table Wildcards for Easy Merges**
- Review Data Schemas: Linking Data Across Multiple Tables
- Walkthrough JOIN Examples and Pitfalls

Google Cloud

Make your UNIONs Easier with the **Table Wildcard ***

```
#standardSQL
SELECT
  stn,
  wban,
  temp,
  year
FROM

`bigquery-public-data.noaa_gsod.gsod1929`
  UNION DISTINCT
`bigquery-public-data.noaa_gsod.gsod1930`
  UNION DISTINCT
`bigquery-public-data.noaa_gsod.gsod1931`
  UNION DISTINCT
`bigquery-public-data.noaa_gsod.gsod1932`
# This is getting out of hand...
```

```
#standardSQL
SELECT
  stn,
  wban,
  temp,
  year
FROM

`bigquery-public-data.noaa_gsod.gsod*`
# All gsod tables
```

Filtering with a Table Wildcard * and _TABLE_SUFFIX_

Use _TABLE_SUFFIX to filter out tables included

Be as granular as you can

- e.g. .gsod2* instead of .gsod* if you only care about the year 2000 onward

```
#standardSQL
SELECT
  stn,
  wban,
  temp,
  year
FROM

`bigquery-public-data.noaa_gsod.gsod*`

# All gsod tables after 1950
WHERE _TABLE_SUFFIX > '1950'
```

Filtering with a **Table Wildcard** * and **_TABLE_SUFFIX_**

- Use Table Wildcard * vs writing many UNIONs
- Use _TABLE_SUFFIX to filter out tables wildcard included
- Use _TABLE_SUFFIX in your SELECT statements with CONCAT( )

Avoid **Union Pitfalls** like Brittle Schemas

- Duplicate Records among tables (Use UNION DISTINCT vs UNION ALL)
- Changing Schemas and Field Names over time.
- Mismatched count of columns in your UNION

Review of What We've Done so Far

```
FROM `bigquery-public-data.noaa_gsod.gsod*`
```

stn	wban	temp	year
030050	99999	49	**1929**
030050	99999	45.7	**1929**
030050	99999	48.2	**1929**
...	...	...	....
037770	99999	50.7	**2017**
030910	99999	56	**2017**
038560	99999	53.2	**2017**

- We are merging all historical gsod tables into one UNION'd table through a **Table Wildcard**

How do we **Enrich** our Temperature Data with Station Details?

```
FROM `bigquery-public-data.noaa_gsod.gsod*`
```

stn	wban	temp	year	name	state	country
030050	99999	49	**1929**			
030050	99999	45.7	**1929**			
030050	99999	48.2	**1929**			
...	...	...	....			
037770	99999	50.7	**2017**			
030910	99999	56	**2017**			
038560	99999	53.2	**2017**			

??

... by **JOIN**ing with data in other tables

Module 9

Joining and Merging Datasets

In this module we will:

- Merge Historical Data Tables with UNION
- Introduce Table Wildcards for Easy Merges
- **Review Data Schemas: Linking Data Across Multiple Tables**
- Walkthrough JOIN Examples and Pitfalls

Google Cloud

What is a **JOIN**?

Combine **data from separate tables** that share a comment element **into one table**

```
#standardSQL
SELECT
  a.stn,
  a.wban,
  a.temp,
  a.year,
  b.name,
  b.state,
  b.country
FROM
  `bigquery-public-data.noaa_gsod.gsod*` AS a
JOIN
  `bigquery-public-data.noaa_gsod.stations` AS b
ON
  a.stn=b.usaf
  AND a.wban=b.wban

WHERE
  # Filter data
  state IS NOT NULL
  AND country='US'
  AND _TABLE_SUFFIX > '2015'
```

What is a **JOIN**?

Fields from Temperature Tables

Fields from Station Details Table

Join Type

Join Condition

```
#standardSQL
SELECT
  a.stn,
  a.wban,
  a.temp,
  a.year,
  b.name,
  b.state,
  b.country
FROM
  `bigquery-public-data.noaa_gsod.gsod*` AS a
JOIN
  `bigquery-public-data.noaa_gsod.stations` AS b
ON
  a.stn = b.usaf
  AND a.wban = b.wban

WHERE
  # Filter data
  state IS NOT NULL
  AND country='US'
  AND _TABLE_SUFFIX > '2015'
```

Module 9

Joining and Merging Datasets

In this module we will:

- Merge Historical Data Tables with UNION
- Introduce Table Wildcards for Easy Merges
- Review Data Schemas: Linking Data Across Multiple Tables
- **Walkthrough JOIN Examples and Pitfalls**

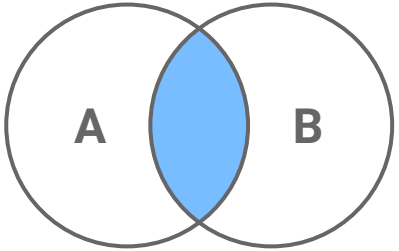
Google Cloud

Different Types of Joins

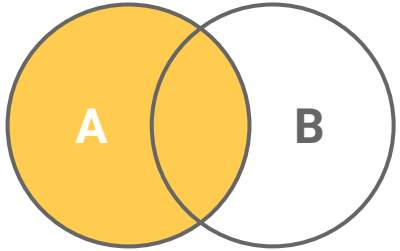
INNER JOIN	Returns rows from multiple tables where join condition is met
LEFT JOIN	Returns all rows from the left table and matched rows from the right table
RIGHT JOIN	Returns all rows from the right table and matched rows from the left table
OUTER JOIN	Returns all rows from all tables and unmatched rows are displayed as NULL

Joins represented via Venn diagram

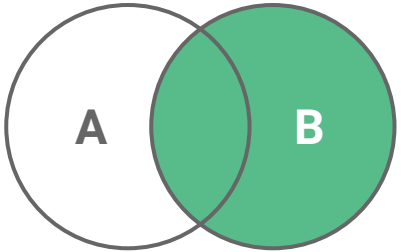
INNER JOIN



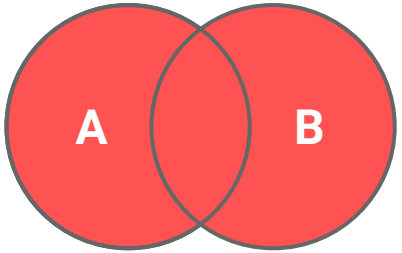
LEFT JOIN



RIGHT JOIN



FULL JOIN



Pitfall: Joining on Non-Unique Fields Explodes your Dataset

- Doing a many-to-many JOIN could result in more rows than either of your initial tables
- This is a primary reason for exceeding your resource cap in BigQuery (unintentionally high compute)
- Know your dataset and the relationships between your tables before joining

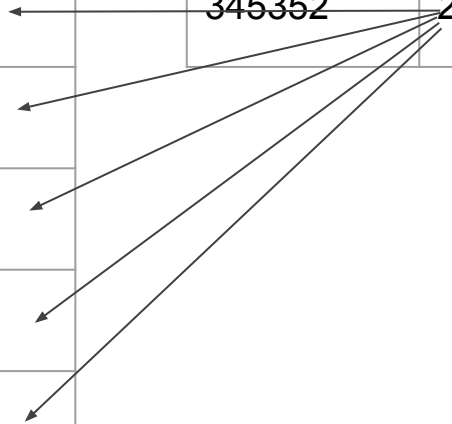
Pitfall: Joining on Non-Unique Fields Explodes your Dataset

Woah, what happened here?

Pitfall: Joining on Non-Unique Fields Explodes your Dataset

irs_990_2015	
ein	tax_pd
112321445	201412
192934562	201412
123454972	201412
123449568	201412
242342422	201412

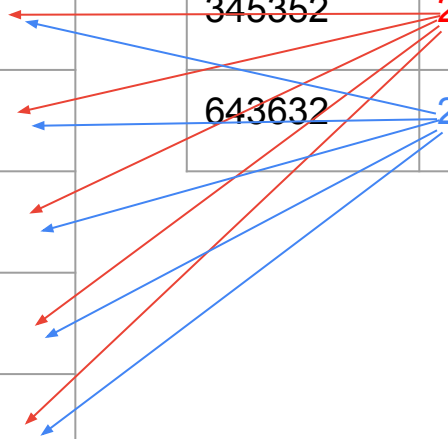
irs_990_ein	
ein	tax_period
345352	201412



Pitfall: Creating an **Unintentional Cross Join**

irs_990_2015	
ein	tax_pd
112321445	201412
192934562	201412
123454972	201412
123449568	201412
242342422	201412

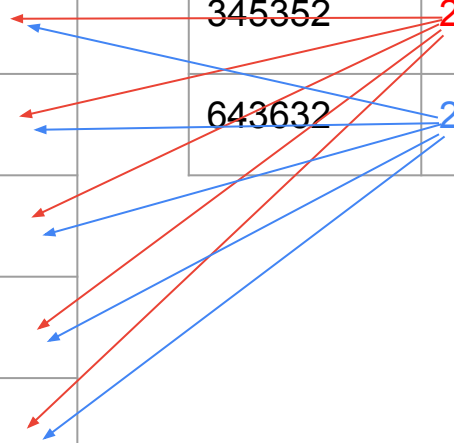
irs_990_ein	
ein	tax_period
345352	201412
643632	201412



Pitfall: Cross Joins Multiply your Data

irs_990_2015	
ein	tax_pd
112321445	201412
192934562	201412
123454972	201412
123449568	201412
242342422	201412

irs_990_ein	
ein	tax_period
345352	201412
643632	201412



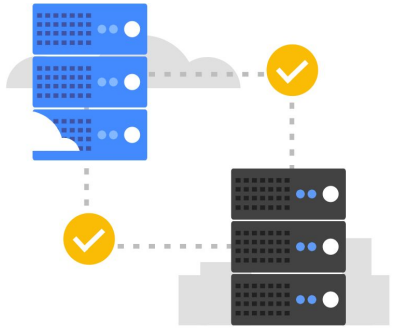
ein	tax_pd
112321445	**201412**
112321445	**201412**
192934562	201412
192934562	201412
123454972	**201412**
123454972	**201412**
123449568	201412
123449568	201412
242342422	**201412**
242342422	**201412**

Pitfall: Understand your Data Model and Relationships

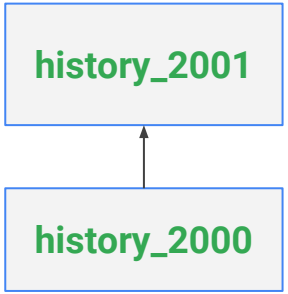
- Understand your data relationship before joining 1:1, N:1, 1:N, N:N
- Use CONCAT( ) to create composite key fields if no unique fields exist or join on more than one field
- Ensure your key fields are distinct (deduplicate)

Summary: Mashup your datasets with Joins and Unions

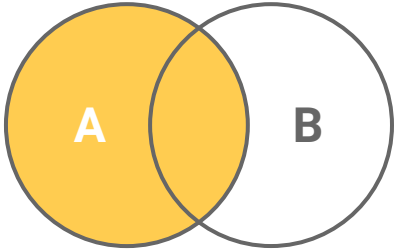
Finding the unique record identifier(s) in table is critical



Spend time exploring the data relationship model between tables



Use UNION wildcards and _TABLE_SUFFIX_ to quickly add records to a consolidated table

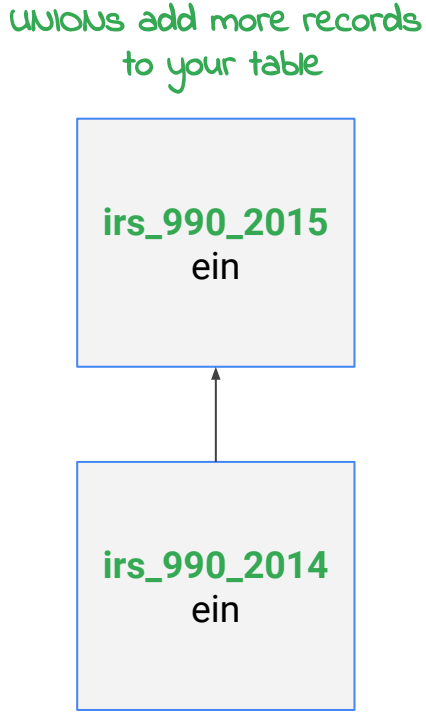


Use JOINs to enrich data across multiple tables

Lab 8
UNIONING and JOINING Datasets

UNIONING and JOINING Datasets

In this lab, you will learn how to apply SQL UNIONs and JOINs to enrich your dataset.



Google Cloud