[BigQuery - SQL join](https://stackoverflow.com/questions/70194352/querying-stackoverflow-public-dataset-on-bigquery-sql-multiple-join)

**Left join using Public dataset**

Left join returns all records (rows) from the left table and the matching record from the right table.

**Tip**: LEFT JOIN and LEFT OUTER JOIN both are same.

bigquery-public-data.stackoverflow.posts\_questions

bigquery-public-data.stackoverflow.posts\_answers

* Same question with multiple answers(use left join because you can't have an answer without a question):

SELECT

  q.tags question\_tag,

  q.id question\_id,

  q.title question\_title,

  q.body question\_body,

  a.id answer\_id,

  a.body

FROM

  `bigquery-public-data.stackoverflow.posts\_questions` AS q

LEFT JOIN

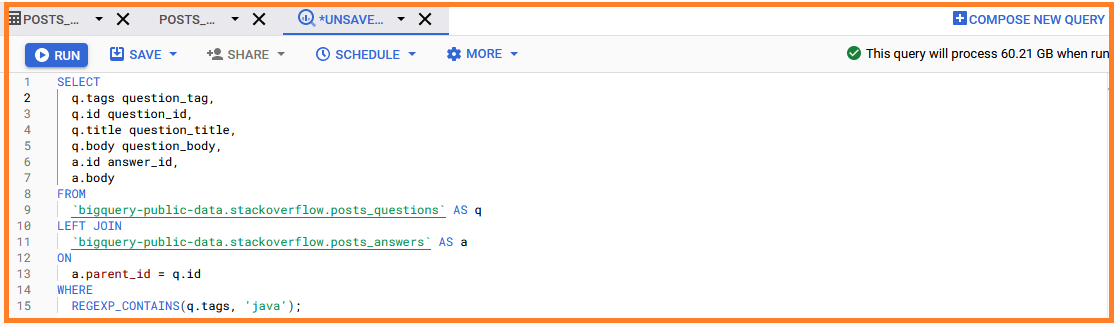
  `bigquery-public-data.stackoverflow.posts\_answers` AS a

ON

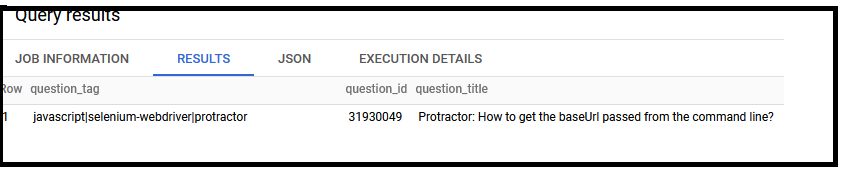
  a.parent\_id = q.id

WHERE

  REGEXP\_CONTAINS(q.tags, 'java');



RESULT



**Right join** **using Public dataset**

The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.

**Tip**: RIGTH JOIN and RIGHT OUTER JOIN both are same.

bigquery-public-data.austin\_bikeshare.bikeshare\_trips

bigquery-public-data.austin\_bikeshare.bikeshare\_stations

**bikeshare\_trips** contains records of all public bike hire trips in San Francisco

**sffd\_service\_calls** contains records of all the San Francisco 311 (non-emergency) calls made since 2008

* Find which bikeshare locations had the most traffic accidents associated with them?

SELECT

s.zipcode\_of\_incident,

s.city AS City,

s.station\_area,

COUNT(b.trip\_id) AS total\_trips,

s.call\_type,

COUNT(s.call\_type) AS total\_calls

FROM

`bigquery-public-data.san\_francisco.bikeshare\_trips` b

RIGHT JOIN

`bigquery-public-data.san\_francisco.sffd\_service\_calls` s

ON

b.zip\_code = s.zipcode\_of\_incident

WHERE

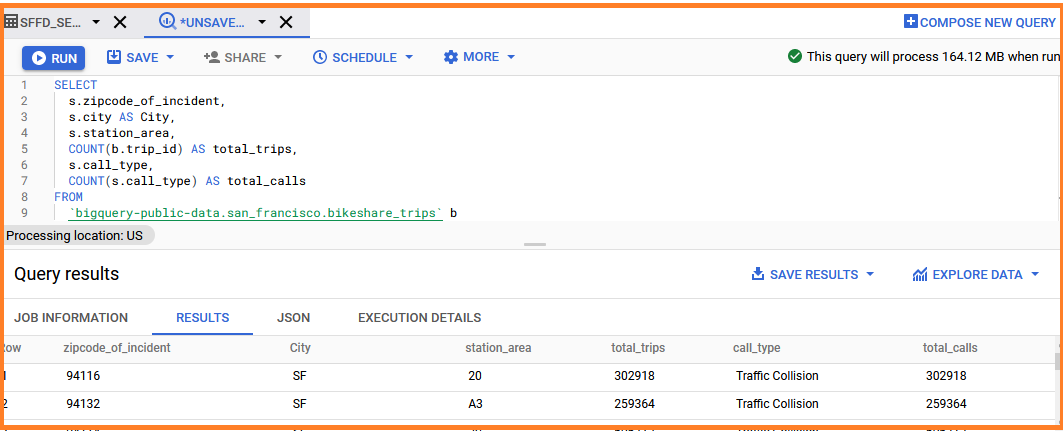
call\_type = 'Traffic Collision'

GROUP BY

1,2,3,5

LIMIT

10;



**Inner join** **using Public dataset**

The INNER JOIN keyword selects records that have matching values in both tables.

bigquery-public-data.baseball.games\_wide

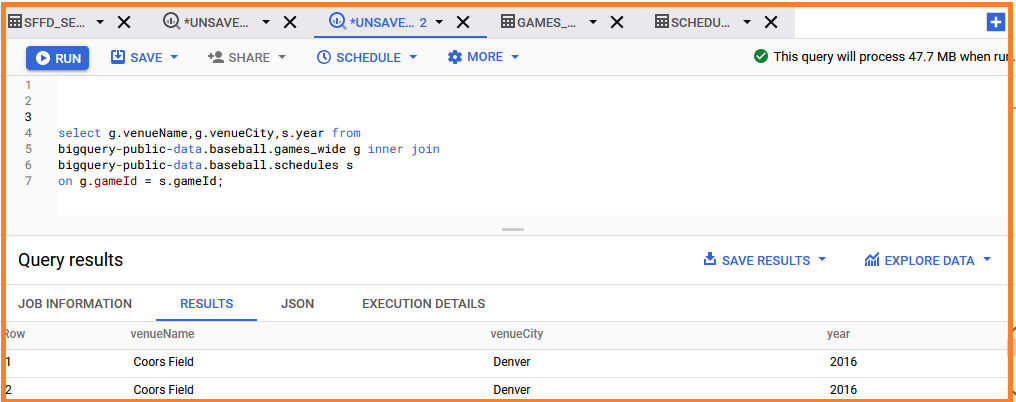
bigquery-public-data.baseball.schedules

select g.venueName,g.venueCity,s.year from

bigquery-public-data.baseball.games\_wide g inner join

bigquery-public-data.baseball.schedules s

on g.gameId = s.gameId;



**Full Outer join** **using Public dataset**

The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.

**Tip:** FULL OUTER JOIN and FULL JOIN are the same.

SELECT

g.venueName,

g.venueCity,

s.year

FROM

`bigquery-public-data.baseball.games\_wide` g

FULL OUTER JOIN

`bigquery-public-data.baseball.schedules` s

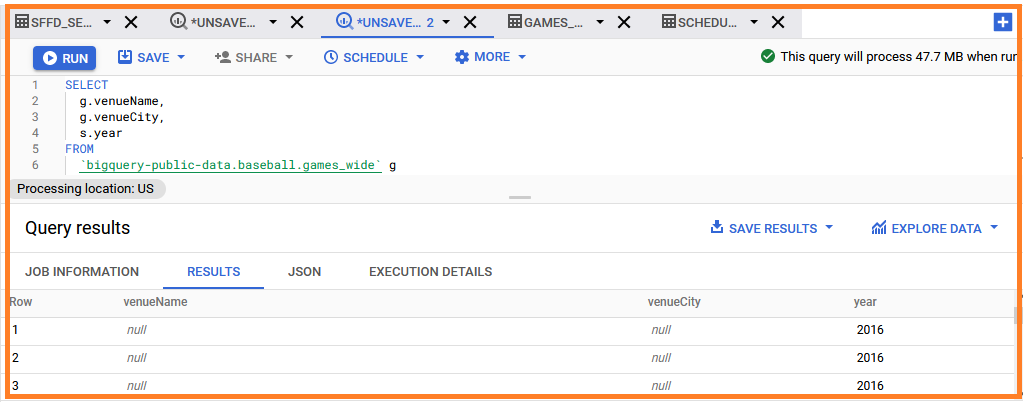
ON

g.gameId = s.gameId

order by g.venueCity;

bigquery-public-data.baseball.games\_wide

bigquery-public-data.baseball.schedules



**WITH Clause**

## Benefits of using the WITH Clause

The WITH clause allows you to reduce joins and logically sequence your code.

* It does this by creating temporary tables (technically they are [views](https://cloud.google.com/bigquery/docs/views-intro)) that are usable by your main SQL statement so that you can break your code into smaller and easier to understand snippets.
* You can use it to create one or more SELECT statements, give each a name, and then reference them in your main query.

WITH

hurricanes AS (

SELECT

NAME,

iso\_time,

latitude,

longitude,

usa\_sshs

FROM

`bigquery-public-data.noaa\_hurricanes.hurricanes`

WHERE

season = '2010'

AND basin = 'NA')

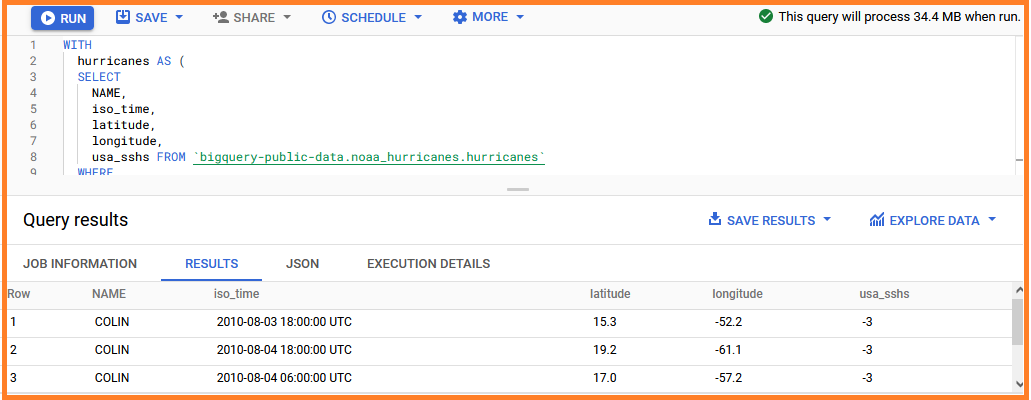
SELECT

\*

FROM

hurricanes

LIMIT 5;



**WITH clause with ARRAY\_AGG**

* In group the result returns a bunch of rows and we do use an aggregate function such as sum( ) or avg( ) of the rows to come down to one value.
* To retain all the rows use ARRAY\_AGG(). In this array, we can give more than one field.
* The array\_agg function needs to be used on columns.

WITH

hurricanes AS (

SELECT

name,

ARRAY\_AGG(STRUCT (iso\_time,

latitude,

longitude,

usa\_sshs )

ORDER BY

iso\_time ASC ) AS track

FROM

`bigquery-public-data.noaa\_hurricanes.hurricanes`

WHERE

season = '2010'

AND basin = 'NA'

GROUP BY

name )

SELECT

\*

FROM

hurricanes

LIMIT

5 ;

