Lesson 7- SQL Advanced JOINs and Performance Tuning

Quiz 8: Self JOINs

Use Case: compare two events, one after another

**Self JOINs**

One of the most common use cases for self JOINs is in cases where two events occurred, one after another. As you may have noticed in the previous video, using inequalities in conjunction with self JOINs is common.

Modify the query from the previous video, which is pre-populated in the SQL Explorer below, to perform the same interval analysis except for the web\_events table. Also:

change the interval to 1 day to find those web events that occurred after, but not more than 1 day after, another web event

add a column for the channel variable in both instances of the table in your query

You can find more on the types of INTERVALS (and other date related functionality) in the Postgres documentation [**here**](https://www.postgresql.org/docs/8.2/static/functions-datetime.html).

SELECT w1.id AS w1\_id,

w1.account\_id AS w1\_account\_id,

w1.occurred\_at AS w1\_occurred\_at,

w1.channel AS w1\_channel,

w2.id AS w2\_id,

w2.account\_id AS w2\_account\_id,

w2.occurred\_at AS w2\_occurred\_at,

w2.channel AS w2\_channel

FROM web\_events w1

LEFT JOIN web\_events w2

ON w1.account\_id = w2.account\_id

AND w1.occurred\_at > w2.occurred\_at

AND w1.occurred\_at <= w2.occurred\_at + INTERVAL '1 day'

ORDER BY w1.account\_id, w2.occurred\_at;