

Do Quizzes on udacity

1. **SELECT** indicates which column(s) you want to be given the data for.
1. **FROM** specifies from which table(s) you want to select the columns. Notice the columns need to exist in this table

SELECT occurred_at, account_id, channel <row/column/> → id+ amount As <new column name>

SELECT channel, AVG(events) AS average_events

SELECT DISTINCT statement is **used to return only distinct (different) values.**

HAVING clause in SQL **specifies that an SQL SELECT statement must only return rows where aggregate values meet the specified conditions.**

SELECT COUNT(accounts.id) AS <name_COUNT>

Calculations: COUNT, SUM, MIN, MAX, AVG,

DATE_TRUNC('month', o.occurred_at)(<year, month, day, second>)

DATE_PART('year', occurred_at(<year, month, day, second>)

⇒ **CASE WHEN** total >500 then 'Over 500'

WHEN total >300 then '301-500'

ELSE 'Small' **END AS** order_level

Concat(first_name, ' ', last_name) AS full_name *or*

first_name||' ', last_name) AS full_name_alt

FROM web_events <file/table>

FROM (SELECT DATE_TRUNC('day', occurred_at) AS day,
channel, COUNT(*) as events

GROUP BY account_id, country

Where account_id = 4251 AND/OR name!= 'USA' <condition/s>

can use =, LIKE, IN, NOT,

LIKE '%google%' (beginning to end). '%C' (beg), 's%' (end)

IN (<value>, <value>)

NOT IN (<value>, <value>)

NOT LIKE '%google%' (beginning to end). '%C' (beg), 's%' (end)

FROM orders

JOIN accounts <second file/table>

ON orders.account_id = accounts.id; <from

Order by <name> <DESC> (if ness, largest values)

LIMIT 15; <number of columns>

NULL means no data

primary vs foreign key: primary access the whole table, The foreign key provides the link between the two tables

SQL joins: Inner, outer(full, left, right,) bubble map