* COUNT(): count the number of rows
* SUM(): the sum of the values in a column
* MAX()/MIN(): the largest/smallest value
* AVG(): the average of the values in a column
* ROUND(): round the values in the column

COUNT() is a function that takes the name of a column as an argument and counts the number of non-empty values in that column.

SELECT COUNT(\*)

FROM fake\_apps

WHERE price = 0;

SELECT SUM(downloads)

FROM fake\_apps;

SELECT MAX(downloads) /MIN

FROM fake\_apps;

SELECT AVG(price)

FROM fake\_apps;

ROUND() function takes two arguments inside the parenthesis:

1. a column name
2. an integer

SELECT ROUND(AVG(price), 2)

FROM fake\_apps;

GROUP BY is a clause in SQL that is used with aggregate functions. It is used in collaboration with the SELECT statement to arrange identical data into *groups*.

The GROUP BY statement comes after any WHERE statements, but before ORDER BY or LIMIT.

SELECT price, COUNT(\*)

FROM fake\_apps

WHERE downloads > 20000

GROUP BY price;

SELECT category, SUM(downloads)

FROM fake\_apps

GROUP BY category;

We can’t use WHERE here because we don’t want to filter the rows; we want to *filter groups*. This is where HAVING comes in.

* When we want to limit the results of a query based on values of the individual rows, use WHERE.
* When we want to limit the results of a query based on an aggregate property, use HAVING.

HAVING statement always comes after GROUP BY, but before ORDER BY and LIMIT.

SELECT price,

ROUND(AVG(downloads)),

COUNT(\*)

FROM fake\_apps

GROUP BY price

HAVING COUNT(\*) > 10;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SELECT first\_name, COUNT(\*) AS count

FROM users

GROUP BY first\_name

ORDER BY count DESC;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SELECT ROUND(watch\_duration\_in\_minutes,0) as duration, COUNT(\*) as count

FROM watch\_history

GROUP BY duration ORDER BY duration ASC;

SELECT user\_id, SUM(amount) AS amount

FROM payments

WHERE status = 'paid'

GROUP BY user\_id

ORDER BY amount DESC;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SELECT user\_id, sum(watch\_duration\_in\_minutes) AS time

FROM watch\_history

GROUP BY user\_id

HAVING time > 400;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SELECT

MAX(watch\_duration\_in\_minutes) AS max,

MIN(watch\_duration\_in\_minutes) AS min

FROM watch\_history;