SQL BOOTCAMP: SELECT, SORT, GROUP, AGGREGATE

GENERAL ASSEMBLY

SELECT [DISTINCT] *column_list*

FROM *table_name*	specifies which table to target
[WHERE where_expression**]**	specifies which rows to retrieve.
[GROUP BY column_list**]**	groups rows sharing a property so that an *aggregate function* can be applied to each group.
[HAVING where_expression**]**	selects among the groups defined by the **GROUP BY** clause.
[ORDER BY *column_list* **[DESC]]**	specifies an order in which to return the rows.
[LIMIT *n***]**;	specifies the maximum number, *n*, of results to return

COLUMN_LIST

A comma separated list of column names (e.g. first_name, last_name), or an asterisk, representing all columns

TABLE_NAME

The name of the table to extract information from. Use **SHOW TABLES** to find out what tables exist.

DISTINCT

If provided, return only the unique combinations of the requested columns

ORDER BY

Order the output by the given columns. Defaults to the smallest first, add **ASC** to reverse

WHERE EXPRESSIONS

- **x = A / x <> A** — *rows where x is equal / not equal to A*
- **x > A / x < A** — *rows where x is greater / less than A*
- **x >= A / x <= A** — *rows where x is greater / less than or equal to A*
- **x BETWEEN A AND B** — *rows where x is between A and B*
- **x IN (A, B, C)** — *rows where x matches any one of A, B or C*
- **x IS NULL / x IS NOT NULL** — *rows where x is (or is not) missing*
- **x LIKE A** — *rows where x matches a pattern: _ (a single character); % (0 or more characters)*

Combine multiple conditions with **AND** or **OR**.

GROUP BY

Rather than returning individual rows, group them together into groups that have the same values for the given columns

HAVING

Optionally specify a where expression saying what groups to keep

AGGREGATE FUNCTIONS

Define aggregate functions here
COUNT(x)
the number of rows in the group
SUM(x)
the sum of the values in the group
MAX(x) / MIN(x)
the group maximum / minimum
AVG(x) / STD(x)
the group average / standard deviation