

---

---

# SQL CHEAT SHEET

Max DELLA LIBERA

# FRAMEWORK

---

-- This is the order of all the SQL keywords we have seen so far!

SELECT

columns,

aggregates AS friendly\_name

FROM table1 t1

JOIN table2 t2 ON t1.column = t2.matching\_column

WHERE conditions

GROUP BY columns

HAVING conditions on aggregates

ORDER BY columns

LIMIT 100;

# FRAMEWORK - EXAMPLE

---

-- Try it yourself!

SELECT

st.name,

sa.category\_name,

SUM(total) AS total\_sales

FROM stores st

LEFT JOIN sales sa ON sa.store = st.store

WHERE st.store\_status = 'A'

GROUP BY 1,2

HAVING SUM(total) > 1000

ORDER BY 3 DESC

LIMIT 100;

# THE FUNDAMENTALS

---

**SELECT \* : TABLE PREVIEW** *(all the columns in the table for the first 100 rows)*

```
SELECT * FROM table LIMIT 100;
```

**SELECT COLUMNS: PREVIEW OF SELECTED COLUMNS**

```
SELECT column1, column2  
FROM table LIMIT 100;
```

**DISTINCT: FIND UNIQUE COMBINATIONS**

```
SELECT DISTINCT column1, column2  
FROM table LIMIT 100;
```

# COMMENTS

---

```
-- This is a one line comment
```

```
SELECT * FROM table; -- This is a comment at the end of the line
```

```
/* This is  
   a comment  
   on multiple lines  
*/
```

```
SELECT * /*inline comment (not recommended)*/ FROM table;
```

---

# FILTERING

---

## WHERE WITH AND, OR

SELECT \* FROM table WHERE column1=x LIMIT 100;

SELECT \* FROM table WHERE column1=x AND column2>x LIMIT 100;

SELECT \* FROM table WHERE column1=x OR column2>x LIMIT 100;

*etc...*

## OPERATORS

=, !=	Equal, not equal to
>, >=	More than, more than or equal to
<, <=	Less than, less than or equal to
BETWEEN	Between two values. E.g.: age BETWEEN 15 AND 35
IN	In a list. E.g.: country IN ( 'Luxembourg' , 'France' , 'Italy' , 'Japan' )
LIKE, NOT LIKE	Similar string (or not similar). E.g.: category_name LIKE '%WHISK%'
%	Wildcard. From the example above, match WHISKEY, OLD WHISKY, WHISKIES etc.

# BASICS

---

**ORDER BY: SORTING** *(by ascending or descending order)*

```
SELECT * FROM table ORDER BY column1, column2 DESC LIMIT 100;
```

```
SELECT * FROM table ORDER BY 1, 2 DESC LIMIT 100;
```

*In "ORDER BY" you can either use the name of the columns, or the position of the columns.*

**:: OR CAST() TO CHANGE COLUMN TYPE**

```
SELECT column1::NUMERIC FROM table LIMIT 100;
```

```
SELECT CAST(column1 AS NUMERIC) FROM table LIMIT 100;
```

---

# AGGREGATE FUNCTIONS

---

We talk about aggregate functions when we need to look into several rows or objects to get an information. For example, we need to look over all the rows to get the `MAX(age)` or `SUM(age)`.

## FUNCTIONS

`MIN()`, `MAX()`, `AVG()`, `SUM()`, `COUNT()`

## GROUP BY: SAME AS A PIVOT TABLE

`SELECT SUM(sales) FROM table; -- This returns only one number`

`SELECT store, SUM(sales) FROM table GROUP BY store;`

`SELECT store, SUM(sales) FROM table GROUP BY 1;`

*The last two queries returns the sum of sales for each store.*

## HAVING: FILTER ON AGGREGATES

`SELECT store, SUM(sales) FROM table`

`GROUP BY store HAVING MAX(sales) > 100; -- Only store with >100 sales`



---

# UNIONS

---

## UNION TO MERGE TABLES

Sometimes your data will be split into several similar tables. This is usually done for very big data set to improve performance or reduce costs.

For example, when you have a lot of sales across stores all over the world, you would have one sale table per day.

To group the tables together, you need UNION

```
SELECT * FROM table_day1  
UNION  
SELECT * FROM table_day2;
```

---

# UNIONS

---

## UNION TO TRANSPOSE

Sometimes your data will be split into several similar tables. This is usually done for very big data set to improve performance or reduce costs.

For example, when you have a lot of sales across stores all over the world, you would have one sale table per day.

To group the tables together, you need UNION

```
SELECT * FROM table_day1  
UNION  
SELECT * FROM table_day2;
```

# JOINS

---

## INNER/LEFT/RIGHT JOIN

```
SELECT *  
FROM table1 t1  
INNER/LEFT/RIGHT JOIN table2 t2 ON t1.column = t2.matching_column;
```

You can also use USING instead of ON as below:

```
SELECT *  
FROM table1 t1  
INNER/LEFT/RIGHT JOIN table2 t2 USING(column_name);
```

---

# KEEP PRACTICING!

Max DELLA LIBERA