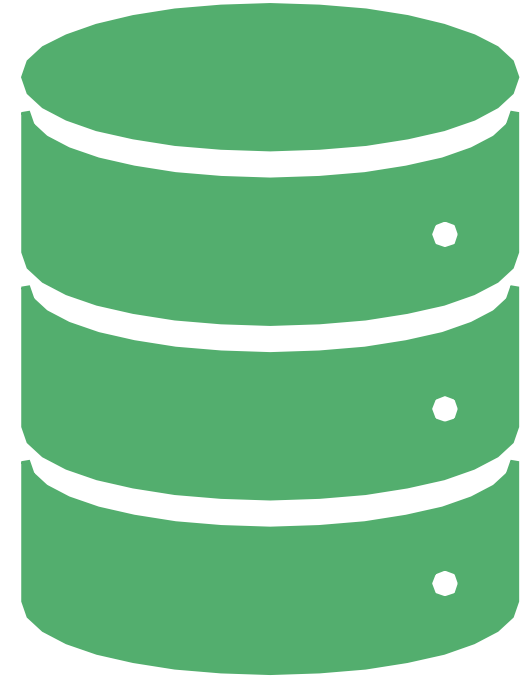


LAB 2: SQL CRASH COURSE

QUERIES



QUERY

- *Solicitud de datos a una(s) tablas alguna base de datos.*
- *Esencial para un Data Scientist.*

KAHOOT: COMPETENCIA

- 1 minuto por pregunta.
- Seleccionar resultado obtenido por el query.
- 1 ganador: 5 netos extra.
- Empate: 2 netos extra.
- Subir notebook como laboratorio #2 a GIT.

A decorative wavy line on the left side of the slide, consisting of a green line and a white line on a black background.

I.

SELECCIONANDO COLUMNAS

SELECT

- Seleccionar determinadas columnas de una tabla.
- **SELECT** field **FROM** table;
 - SELECT y FROM son keywords.
 - No son *case sensitive*.

- BUENA PRÁCTICA:

SELECT field
FROM table;

SELECT DISTINCT

- Resultados duplicados en las consultas.
- Sirve para obtener los valores únicos.

SELECT DISTINCT

field

FROM table;

COUNT

- Contar número de filas en un Query.
- Combinable con **DISTINCT**

```
SELECT COUNT(*)  
FROM table;
```

```
SELECT COUNT(field)  
FROM table;
```



**FILTRANDO
RESULTADOS**

WHERE

- Permite filtrar basado en texto y valores numéricos.

SELECT field

FROM table

WHERE field = 2;

SELECT field

FROM table

WHERE field = 'The Beatles' ;

= *equal*

<> *not equal*

< *less than*

> *greater than*

<= *less than or equal to*

>= *greater than or equal to*

WHERE AND

- Permite filtrar en múltiples condiciones.

SELECT *

FROM table

WHERE field_1 = 2

AND field_2 ='The Beatles';

SELECT *

FROM table

WHERE field_1 > 5

AND field_1 < 10;

= equal

<> not equal

< less than

> greater than

<= less than or equal to

>= greater than or equal to

BETWEEN / IN

- Permite buscar dentro de un rango
- *Es inclusivo*

```
SELECT *  
FROM table  
WHERE field_I  
BETWEEN I AND 5;
```

```
SELECT *  
FROM table  
WHERE field_I  
IN (1,3,5,7);
```

ORDER BY

- Ordena los resultados por un campo en específico.
- Por default, es en orden ascendente.

```
SELECT *  
FROM table  
ORDER BY field_1,  
field_2;
```

```
SELECT *  
FROM table  
ORDER BY field_1  
DESC;
```

GROUP BY

- Permite realizar agregaciones por algún parámetro en específico.
- **SELECT** field_1,field_2, **COUNT**(*)
FROM table
GROUP BY field_2;

HAVING

- WHERE no se puede utilizar al momento de hacer una agregacion.
- Si queremos filtrar por agregacion, utilizamos **HAVING**

```
SELECT field_1, field_2,  
COUNT(*)  
FROM table  
  
GROUP BY field_2  
  
HAVING COUNT(field_1) >  
100;
```