



# Arquitectura y Administración de bases de datos con SQL 2021

Héctor Manuel Garduño Castañeda

Diciembre, 2021



# Contenido

ORDER BY

LIMIT

Alias



La instrucción **ORDER BY** es utilizada para ordenar los registros de un conjunto de resultados (recuerda que un conjunto de resultados es el resultado de hacer un select). Solo puede ser utilizada con **SELECT**.

```
SELECT <<nombre de la columna>>  
FROM <<nombre de la tabla>>  
(WHERE si es que quieres una condición)  
ORDER BY <<alguna columna>>[ASC,DESC], <<alguna  
columna>>[ASC,DESC],...;
```



+ En caso de haber empates  
que ordene respecto a otra  
columna

+ Un "conjunto de resultados" es el output de un **SELECT**



## Práctica

De la tabla *customer*, mostrar aquellos registros cuyo estado es California ordenados alfabéticamente por nombre; mostrar todos los registros ordenados alfabéticamente, pero en orden inverso, por nombre; mostrar todos los registros cuya edad es mayor a 25 ordenados en ascenso por ciudad y en descenso por nombre; mostrar todos los registros ordenados por edad.

```
SELECT * FROM customer
```

```
WHERE state = 'California' ORDER BY Customer_name;
```

+ Si no se especifica si es ASC O  
DESC asumirá ASC

```
SELECT * FROM customer
```

```
WHERE state = 'California' ORDER BY Customer_name ASC;
```

```
SELECT * FROM customer ORDER BY Customer_name DESC;
```

```
SELECT * FROM customer
```

```
WHERE age>25 ORDER BY City ASC, Customer_name DESC;
```

```
SELECT * FROM customer ORDER BY age;
```



+ Cuando haya empate de 'age', no definimos criterio de desempate y SQL los pone de alguna forma (NO USAMOS WHERE)

+ Cuando exista empate en 'city', el criterio de desempate será 'customer\_name' en orden DESC

Cuando solo queremos ver una pequeña parte de una tabla, podemos utilizar la sentencia **LIMIT**. Esta nos sirve para evitar mostrar el conjunto de resultados de una consulta.

```
SELECT <<nombre de la columna>>  
FROM <<nombre de la tabla>>  
(WHERE si es que quieres una condición)  
(ORDER BY si quieres algún orden) } opcionales  
LIMIT <<contador de filas>>;
```

+ Funciona como el head(n) de Python



## Práctica

De la tabla *customer*, muestra los primeros 8 registros cuya edad es al menos 25 ordenados en forma descendente por edad; muestra los primeros 10 registros cuya edad es al menos 25 ordenados en forma ascendente por edad.

```
SELECT * FROM customer  
WHERE age >= 25 ORDER BY age DESC  
LIMIT 8;
```

```
SELECT * FROM customer  
WHERE age >= 25 ORDER BY age  
LIMIT 10;
```

+ Podemos evitar poner ASC pues SQL  
(al omitir el valor) toma ASC como el valor  
default



La palabra clave **AS** es utilizada para dar un nombre provisional (alias) a una columna o una tabla. No es como tal un cambio de nombre, sino solamente un apodo provisional.

```
SELECT <<nombre de la columna>> AS <<alias de la columna>>  
FROM <<nombre de la tabla>>  
(Instrucciones extra de filtrado, orden y limit);  
( WHERE, ORDER BY, LIMIT, etc.)
```

+ Los outputs de los SELECT se llaman "query" y NO son una tabla, pero a estos query se les puede poner un alias con la sentencia AS



# Práctica

- + Notar que es con DOBLE comillas, pero un alias no tiene comillas (nombre)
- + Aquellos que llevan comillas es para que nos permitiera poner espacios en los alias, pero al momento de hacer alguna consulta lanza error con esos alias que tienen espacio  
ORDER BY Edad cliente ; por ejemplo
- + Los alias solo se quedan en la query que estamos haciendo (no afectan para nada la tabla original)

El siguiente código nos devolverá las columnas seleccionadas pero con un alias.

```
SELECT customer_id AS "Num de cliente", customer_name AS nombre,  
age AS "Edad cliente"  
FROM customer;
```

- + El output es una tabla que muestra las columnas con esos nombres modificados, solo muestra las columnas a las que les aplicamos un alias
- + Desde SELECT hasta FROM estamos haciendo una query normal, como la hemos venido haciendo en el curso, entonces podemos ponerle todo de lo que ya sabemos, PERO ahora debemos hacer uso de los alias que asignamos a esas columnas:

```
SELECT customer_id AS "Num de cliente", customer_name AS nombre, age AS "Edad cliente"  
FROM customer  
ORDER BY nombre;
```

