



**Argentina
programa
4.0**



Ministerio de Economía
Argentina

Secretaría de
Economía del Conocimiento

***primero
la gente***

Clase 19: Bases de datos Relacionales

El lenguaje SQL

Agenda de hoy

- A. EL LENGUAJE SQL
 - a. Subcategorías / Sublenguajes
 - i. DQL - DML
 - ii. DDL
 - iii. DCL
 - iv. TCL
- B. La cláusula SELECT
 - a. FROM
 - b. LIMIT
 - i. diferencias con SQL SERVER
 - c. ORDER
 - i. simple
 - ii. compuesto
- C. DISTINCT



El lenguaje SQL

Tal como venimos hablando desde el momento en el cual comenzamos a interactuar con bases de datos, el lenguaje SQL es un lenguaje de programación dedicado a trabajar con la manipulación de datos en general.

Este nació junto con la propuesta de Bases de datos Relacionales, en la década del 60', y se convirtió en un lenguaje regulado, regido por el Estándar ANSI SQL.



El lenguaje SQL

El proceso de establecer un Estándar para SQL surgió dado que entre los 70's y principio de los 80's comenzaron a abundar los motores de Bases de datos en el mercado de software, jactándose cada uno de ellos de utilizar el lenguaje SQL.

Dado que muchos lenguajes comenzaron a desviar la propuesta original del lenguaje en sí, SQL comenzó a ser regulado para que toda empresa de software que se jacte de dar soporte SQL en su base de datos, debiera cumplimentar con dicho Estándar.



El lenguaje SQL

El proceso de establecer un Estándar para SQL surgió dado que entre los 70's y principio de los 80's comenzaron a abundar los motores de Bases de datos en el mercado de software, jactándose cada uno de ellos de utilizar el lenguaje SQL.

Dado que muchos lenguajes comenzaron a desviar la propuesta original del lenguaje en sí, SQL comenzó a ser regulado para que toda empresa de software que se jacte de dar soporte SQL en su base de datos, debiera cumplimentar con dicho Estándar.



El lenguaje SQL

Como vimos en nuestro encuentro anterior, SQL se apoya sobre el modelo de datos estructurados, almacenando el mismo en Tablas, y definiendo un tipo de datos “*tipado*”, para cada uno de los campos que conforman cada tabla.

Si bien SQL cuenta con otros tantos objetos dentro de una base de datos (*Vistas, Procedimientos Almacenados, Funciones Almacenadas*), las tablas son el punto de partida para que el resto de los objetos funcionen correctamente en torno a éstas.

A 6x4 grid with a pink header row and blue body rows. The grid is composed of 24 cells arranged in 6 rows and 4 columns. The first row is pink, and the remaining five rows are blue. The grid is outlined with a thick dark blue border.

El lenguaje SQL

Y toda la interacción, operaciones y manejo general de una base de datos, y de cualquiera de sus objetos, SQL lo logra utilizando el lenguaje SQL en sí.

Este es un lenguaje fácil y práctico de aprender. Su estructura está representada por una serie de instrucciones, o cláusulas, establecidas en formato imperativo, y definidas con un lenguaje muy cercano al idioma inglés.



El lenguaje SQL

Existen diferentes tipos de operaciones que este lenguaje nos permite realizar, podemos destacar a:

- Ejecutar consultas para recuperar datos
- agregar, modificar y/o eliminar registros
- crear nuevas bb.dd.
- crear nuevas tablas, vistas, funciones
- crear programas (*usando Stored Procedures*)
- modificar vistas de datos, funciones, tablas
- Definir permisos de uso de los objetos



El lenguaje SQL

Además, las diferentes operaciones que se pueden realizar en una base de datos SQL, están definidas bajo diferentes denominaciones regidas por siglas.

Estas siglas permiten identificar las diferentes funcionalidades que cada base de datos pone a nuestra disposición.

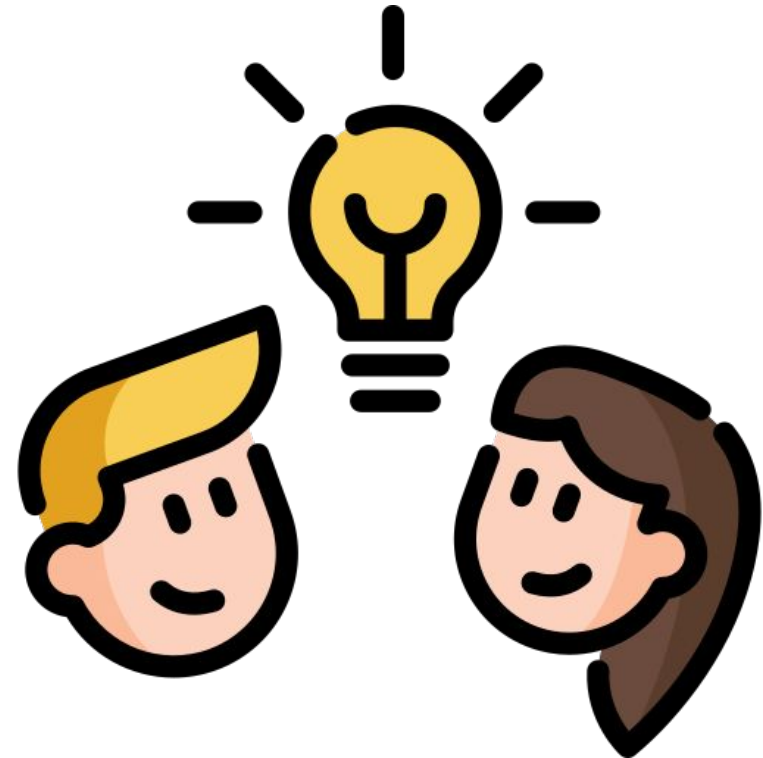
Las funcionalidades son denominadas Sublenguajes.



El lenguaje SQL

En resumen, todo lo que podemos hacer con mysql workbench también lo podemos hacer directamente con el lenguaje sql.

Existe una sentencia SQL para todo lo mencionado anteriormente, y más también. Si bien nos vamos a concentrar en la parte más importante, cubriremos gran parte del lenguaje base, que incluso se usará para operaciones o acciones más complejas.



Sublenguajes / Subcategorías

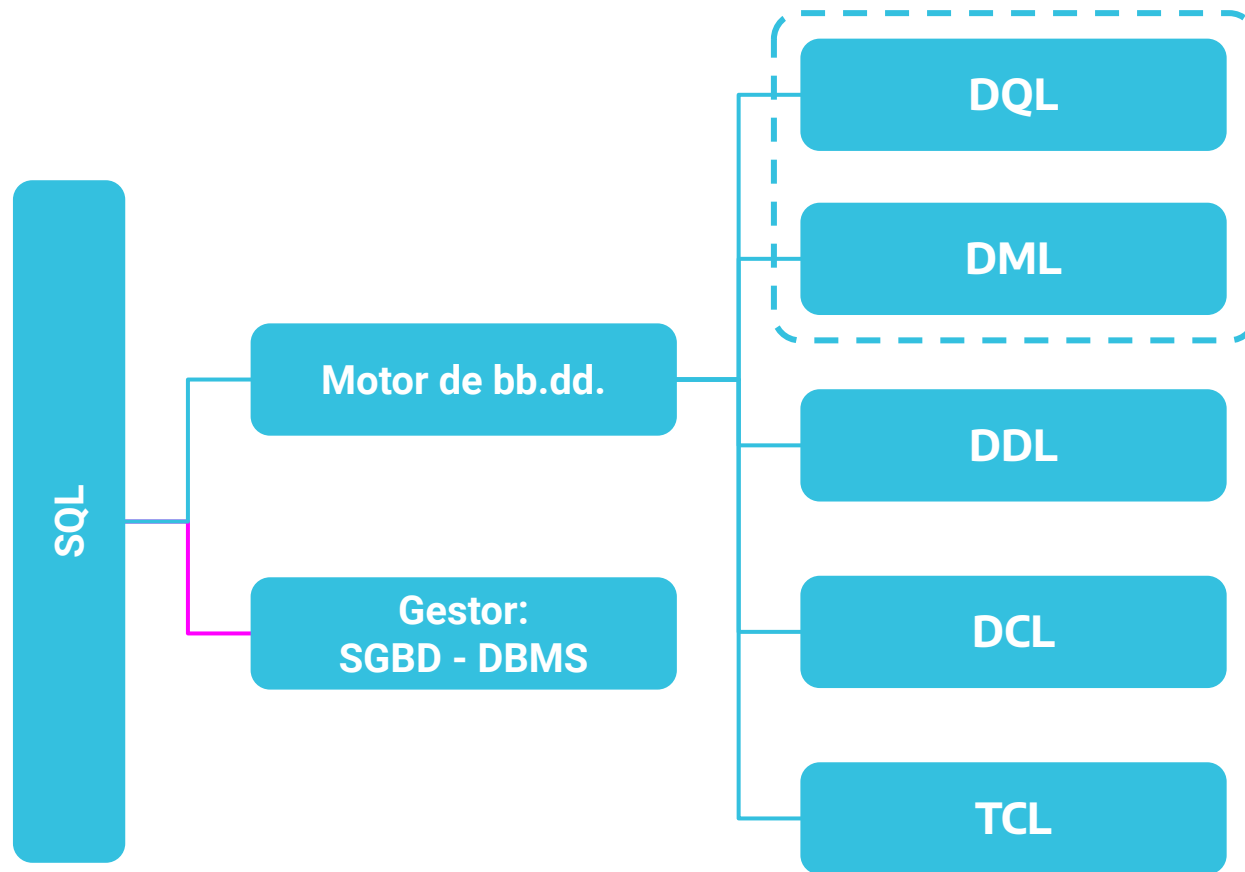
Sublenguajes / Subcategorías

Veamos a continuación un gráfico de una base de datos SQL, en donde representaremos todos los componentes que conforman a las mismas.

Cada sublenguaje, también denominados subcategorías, permiten ordenar rápidamente el tipo de operaciones que estas agrupan.

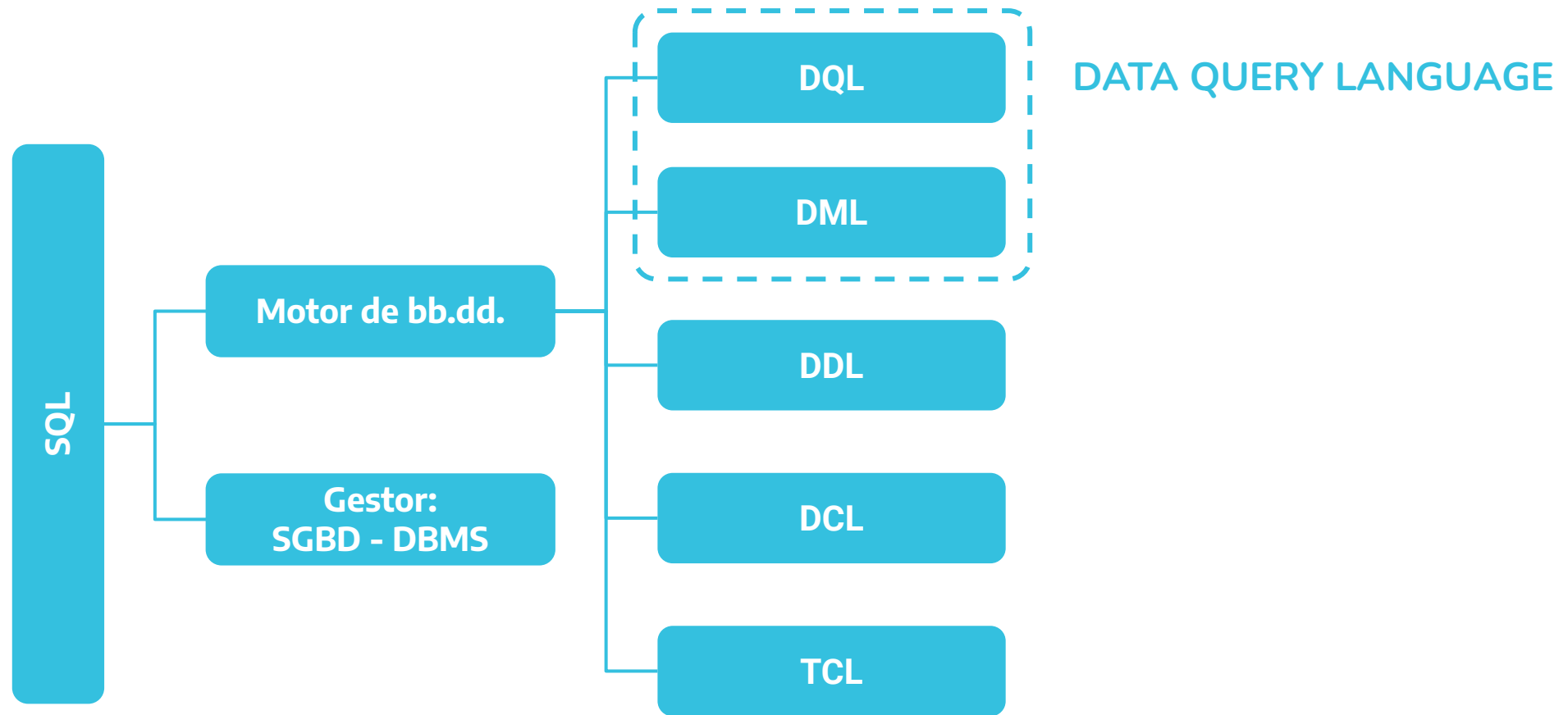


Sublenguajes / Subcategorías



Estas siglas representan a las diferentes subcategorías / sublenguajes integrados en SQL.

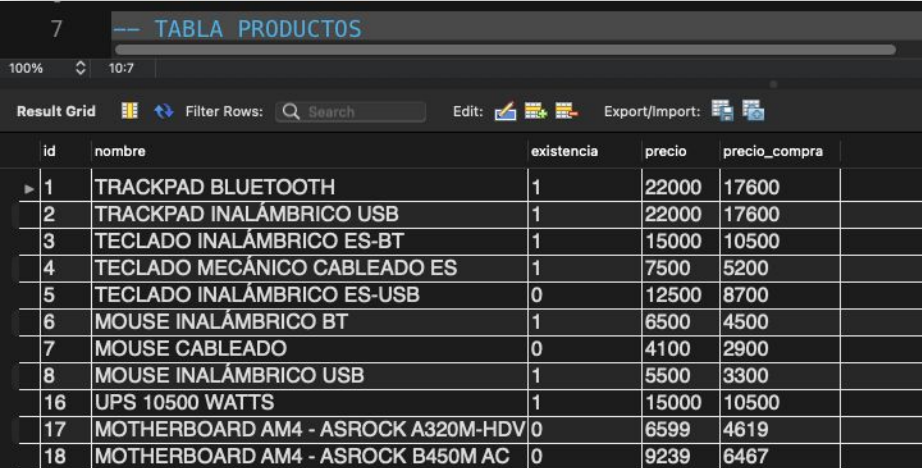
Sublenguajes / Subcategorías



Sublenguajes / Subcategorías

Hasta el momento, todo el manejo de información sobre una base de datos SQL lo realizamos a través de MySQL Workbench.

Seguiremos utilizando este gestor, tal como venimos haciendo hasta el momento, con la diferencia de que integraremos el sublenguaje DQL para solicitarle que realice operaciones mediante cláusulas SQL.



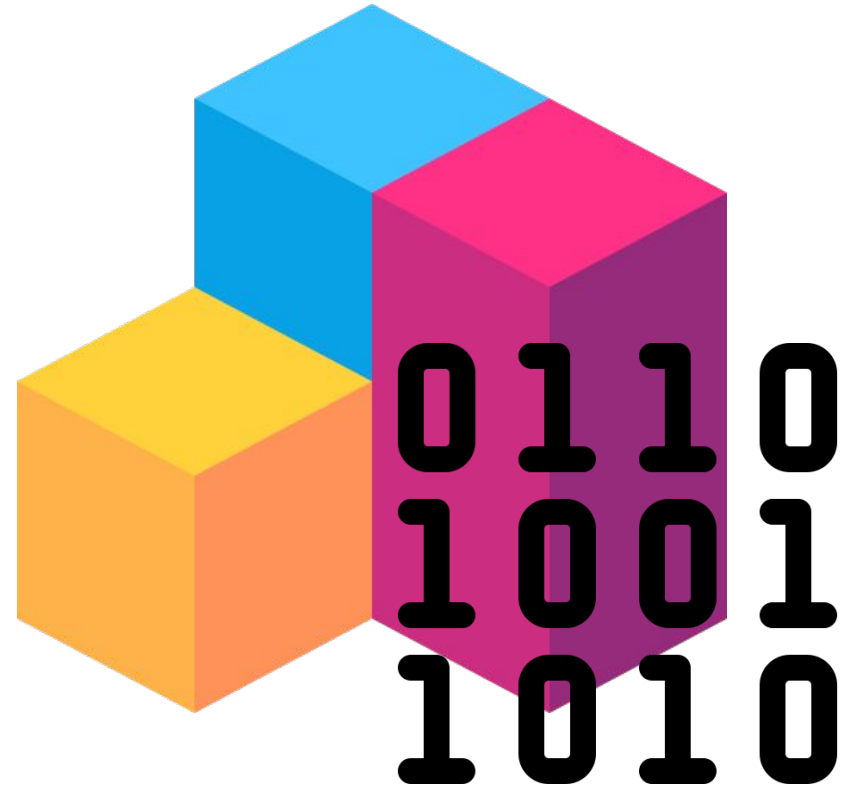
The screenshot shows the MySQL Workbench interface with the 'TABLA PRODUCTOS' table open. The table contains 18 rows of product data. The columns are: id, nombre, existencia, precio, and precio_compra. The data is as follows:

id	nombre	existencia	precio	precio_compra
1	TRACKPAD BLUETOOTH	1	22000	17600
2	TRACKPAD INALÁMBRICO USB	1	22000	17600
3	TECLADO INALÁMBRICO ES-BT	1	15000	10500
4	TECLADO MECÁNICO CABLEADO ES	1	7500	5200
5	TECLADO INALÁMBRICO ES-USB	0	12500	8700
6	MOUSE INALÁMBRICO BT	1	6500	4500
7	MOUSE CABLEADO	0	4100	2900
8	MOUSE INALÁMBRICO USB	1	5500	3300
16	UPS 10500 WATTS	1	15000	10500
17	MOTHERBOARD AM4 - ASROCK A320M-HDV	0	6599	4619
18	MOTHERBOARD AM4 - ASROCK B450M AC	0	9239	6467

Sublenguajes / Subcategorías

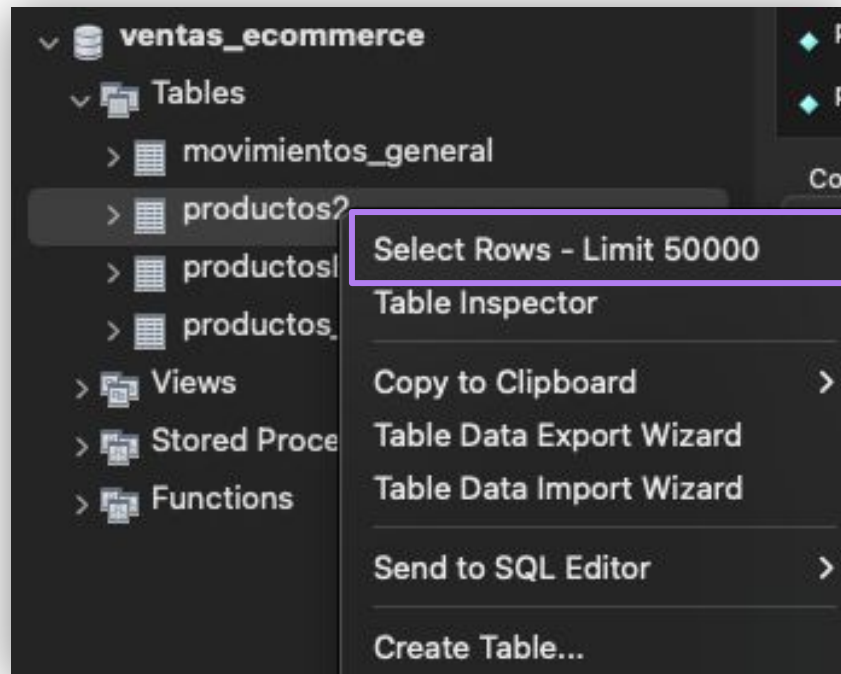
Una cláusula también se denomina “*comando o sentencia*”, son palabras reservadas del lenguaje, que permiten ejecutar acciones sobre la bb.dd.

Gracias a estas, nosotras como usuarias técnicas/ administradoras / programadoras, operaremos sobre la bb.dd. de acuerdo a la tarea que debamos realizar.



La cláusula SELECT

La cláusula SELECT



Cuando usamos MySQL Workbench para abrir una tabla y mirar su información, pulsamos sobre la opción **SELECT ROWS x000...**

Así se ejecuta la consulta sobre la tabla y visualizamos la misma en el entorno de trabajo de MySQL Workbench.

En la parte superior de la ventana de MySQL Workbench, veremos que se arma una consulta de selección, SQL.

La cláusula SELECT

Pongamos foco en la estructura; luego veremos cada palabra en detalle.

Para consultar determinados campos de una tabla:



SELECT - FROM parcial

```
SELECT nombre, apellido, dni FROM clientes_banco;
```

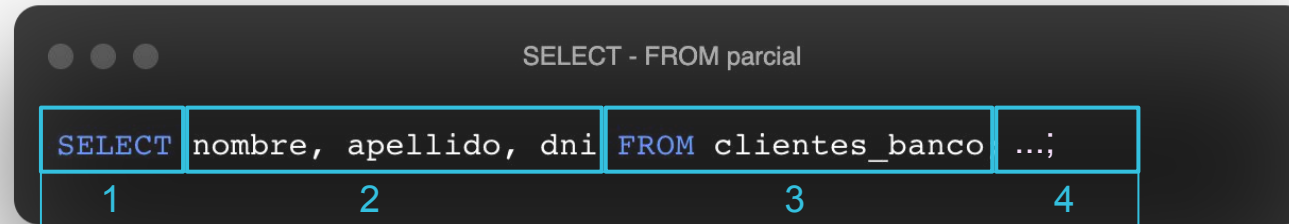
Para consultar todos los campos de una tabla:



SELECT - FROM completo

```
SELECT * FROM clientes_banco;
```

La cláusula SELECT



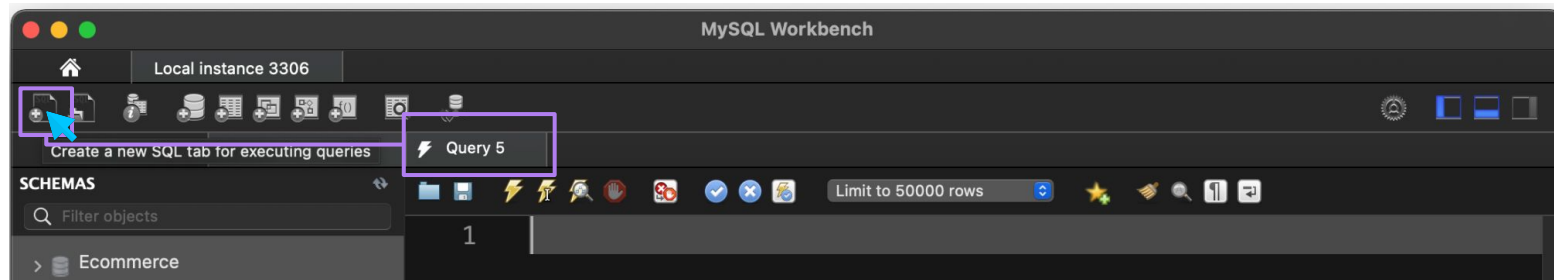
Las operaciones en SQL siguen una estructura que describe la operación que deseamos realizar. Si bien encontraremos consultas muy sencillas y otras más complejas, existen algunos elementos que aparecen con mayor frecuencia:

1. **Acción (keyword):** crear, seleccionar, insertar, actualizar, eliminar, etc.
2. **Porción donde operaremos:** puede ser uno o más campos o un asterisco (*) para seleccionar todos los campos, o traer una o más tablas, etcétera
3. **Tabla a la cual queremos acceder:** la identificamos escribiendo el nombre
4. **Condiciones:** establecer criterios para operar sólo sobre los registros que los cumplan

La cláusula SELECT

- Las sentencias SQL no son sensibles a las mayúsculas y minúsculas.
No obstante, es importante respetarlas al colocar el nombre de un campo o tabla.
- Cada sistema de bases de datos tiene sus particularidades sintácticas. Sin embargo, si conocemos la base del SQL podremos adaptarnos sin dificultades.
- Cada consulta finaliza con punto y coma (;)
- Previo a comenzar a escribir consultas en MySQL Workbench, debemos seleccionar la base de datos con la cual vamos a trabajar, a través de la cláusula **USE**.

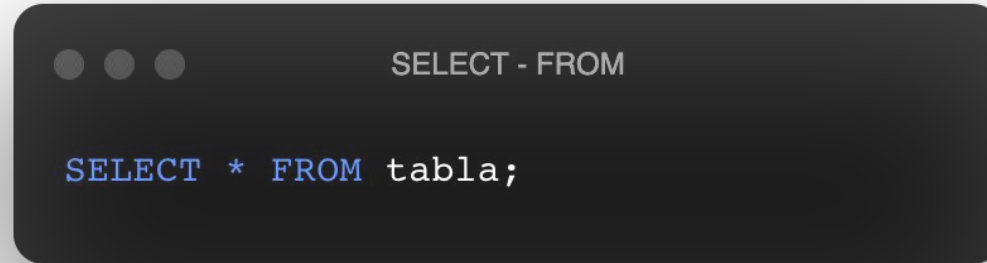
La cláusula SELECT



En MySQL Workbench disponemos en la barra de herramientas superior, la opción de iniciar una nueva pestaña (*tab*) de SQL (*de script*) para ejecutar consultas.

Pulsemos el botón para crear una pestaña y comenzar a trabajar.

La cláusula SELECT



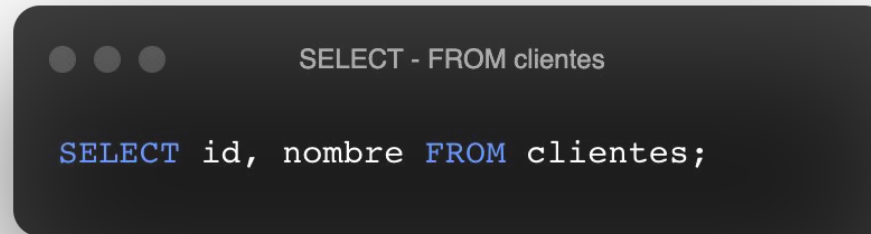
```
SELECT - FROM

SELECT * FROM tabla;
```

La sentencia SELECT, como lo indica su nombre, permite seleccionar información a extraer desde una tabla y visualizar la información resultante.

La cláusula **FROM** complementa a **SELECT**. Esta declara la tabla desde la cual se va a extraer la información.

La cláusula SELECT

A dark-themed terminal window with three gray window control buttons (minimize, maximize, close) in the top-left corner. The title bar text reads "SELECT - FROM clientes". The main area contains a SQL query in a light blue monospace font: "SELECT id, nombre FROM clientes;".

```
SELECT - FROM clientes

SELECT id, nombre FROM clientes;
```

Resultado de la consulta:

id	nombre
1	Marissa Mayer
2	Susan Wojcicki
3	Mira Murati
4	Melinda Gates

La cláusula SELECT

● ● ● SELECT - FROM clientes campos según conveniencia

```
SELECT DNI, nombre, id FROM clientes;
```

Manejo de campos de una tabla:

El orden de los campos es irrelevante.

Podemos definir el que necesitemos en primer lugar, más allá de la posición donde éste haya sido definido cuando se construyó la tabla.

Al visualizar el resultado, el orden de los campos será tal como lo pedimos en la consulta.

ORDER BY

ORDER BY

Dentro de toda consulta SQL existe la posibilidad de aplicar un ordenamiento sobre la información a mostrar.

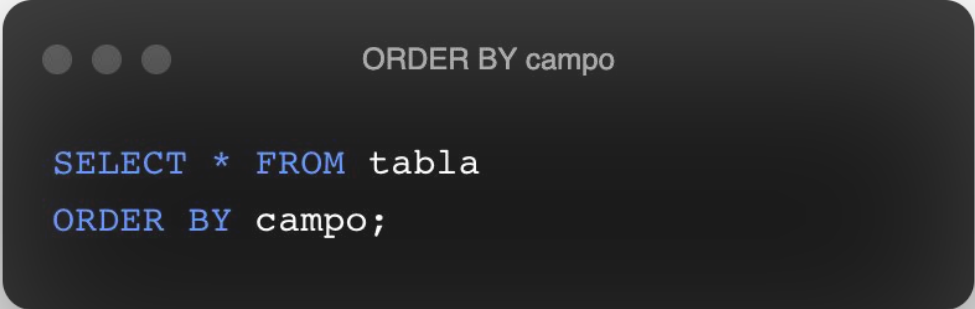
De acuerdo a nuestro propósito para con el tratamiento de la información, podemos listar la misma ordenándola por un campo específico.



ORDER BY

Para esto, debemos agregar a la consulta de selección, **la cláusula ORDER BY**.

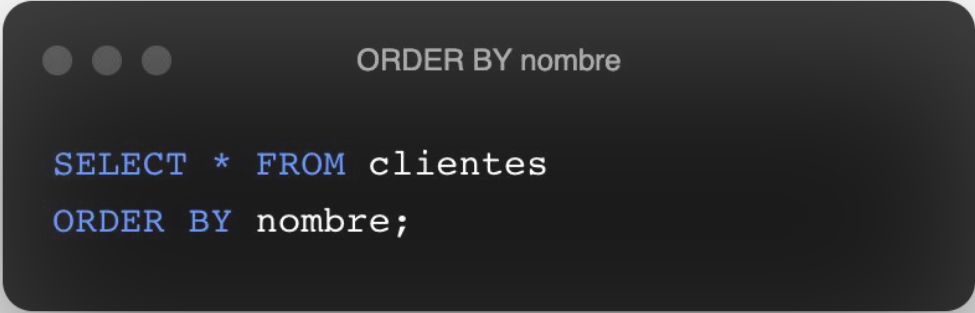
La misma, se ubica al final de la consulta de selección, definiendo a continuación de esta cláusula, el campo en cuestión por el cual queremos ordenar los datos, previo a visualizarlos.



```
SELECT * FROM tabla  
ORDER BY campo;
```

ORDER BY

En cuanto a ordenamiento refiere, es indistinto pedir uno, dos, o todos los campos en la consulta de selección y luego especificar en el ordenamiento, un campo que tal vez no necesitamos incluir en la consulta.



```
ORDER BY nombre

SELECT * FROM clientes
ORDER BY nombre;
```

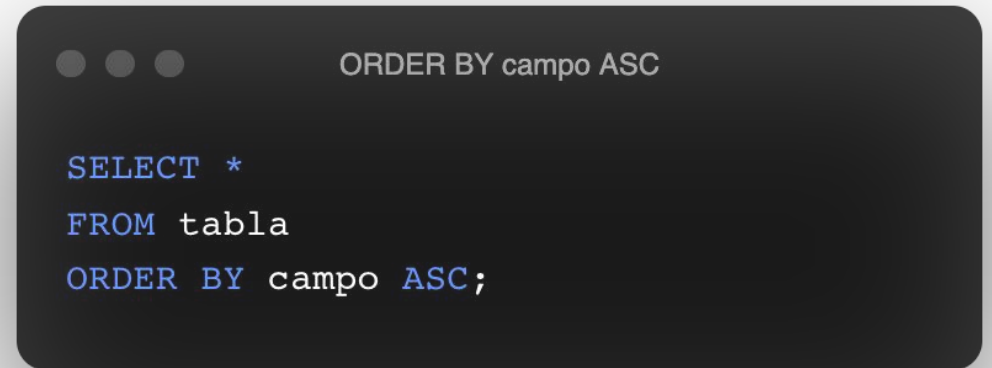
Tipo de ordenamiento

ORDER BY

También, dentro del ordenamiento de los datos, podemos especificar la forma en la cual queremos ordenarlos.

- Forma ascendente
- Forma descendente

Existen palabras reservadas para esta operación.



```
ORDER BY campo ASC

SELECT *
FROM tabla
ORDER BY campo ASC;
```


ORDER BY

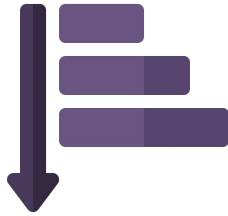
Orden ascendente (*opcional*)

```
ORDER BY nombre ASC  
  
SELECT *  
FROM clientes  
ORDER BY nombre ASC;
```

Orden descendente (*obligatorio*)

```
ORDER BY DNI DESC  
  
SELECT *  
FROM clientes  
ORDER BY DNI DESC;
```

ORDER BY



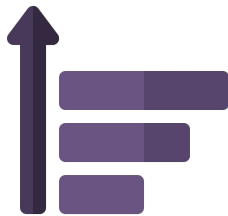
ASC ordena:

Textos varios: [A - Z]

Números: [0 - 9]

Fechas: [21-01-2022 - 25-05-2022]

ó [2022-01-21 - 2022-05-25]



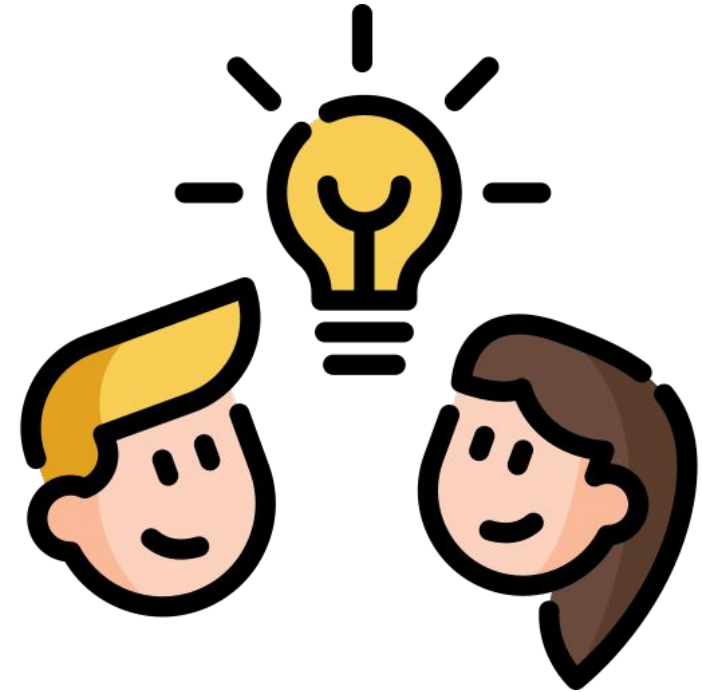
DESC ordena:

Textos varios: [Z - A]

Números: [9 - 0]

Fechas: [25-05-2022 - 21-01-2022]

ó [2022-05-25 - 2022-01-21]



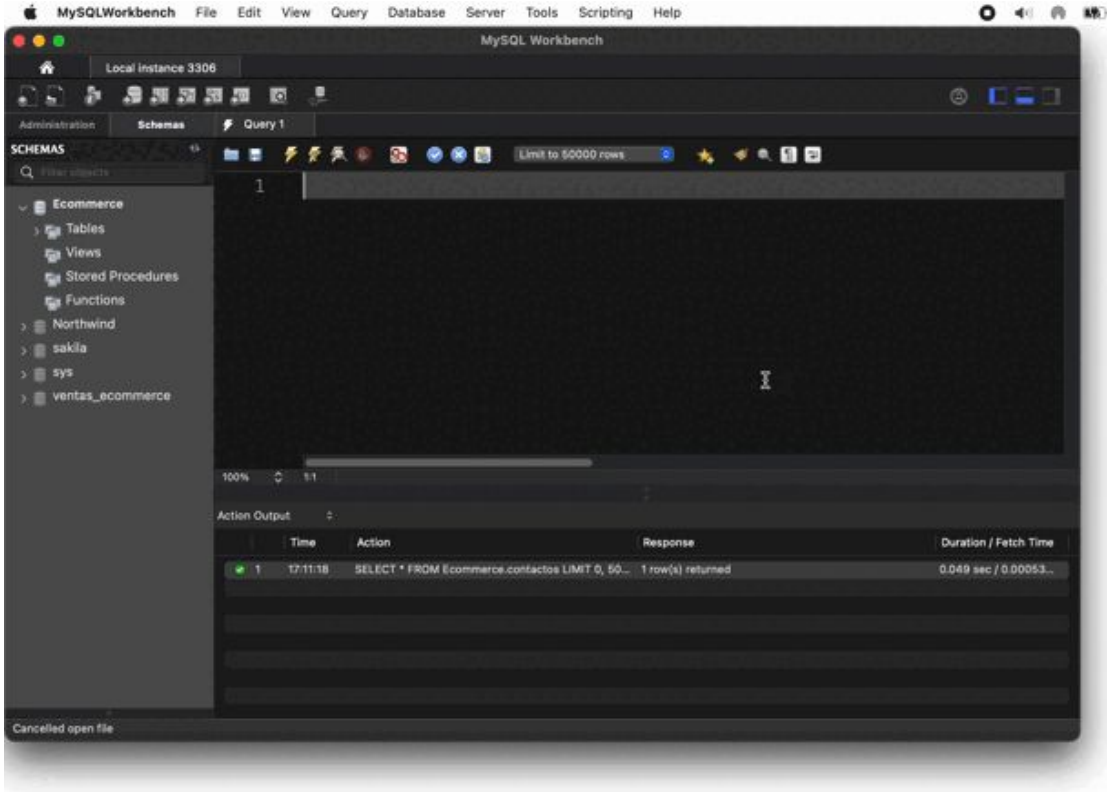
Sección práctica

Para poder abordar prácticas sobre datos contundentes, instalaremos una base de datos MySQL en nuestro entorno de trabajo denominada Northwind.

La misma ya posee una estructura de datos y una cantidad importante de registros, lo cual nos permitirá realizar diferentes tipos de consultas utilizando el lenguaje SQL.



Prácticas



- Instalaremos la **base de datos Northwind** en MySQL Workbench para comenzar a realizar prácticas.
- Descarga el **archivo .sql** compartido por la Profe
- Ábrelo con MySQL Workbench y pulsa el botón **Execute the Script:** (menú **Query > Execute**)

Tiempo estimado: **10 minutos**

Como referencia

Northwind es una base de datos de ejemplo que se utilizó en la documentación y los ejemplos de Microsoft Access desde las primeras versiones de esta aplicación de software.

Representa una empresa ficticia que vende productos a clientes y realiza transacciones comerciales, y contiene tablas de productos, categorías de productos, proveedores, clientes, pedidos, detalles de pedidos y empleados.

Northwind es una excelente herramienta para aprender y practicar la administración de bases de datos, la escritura de consultas y la creación de métricas. Debido a que ha sido utilizada en muchos ejemplos y tutoriales, es fácil encontrar recursos y guías online para trabajar con Northwind.

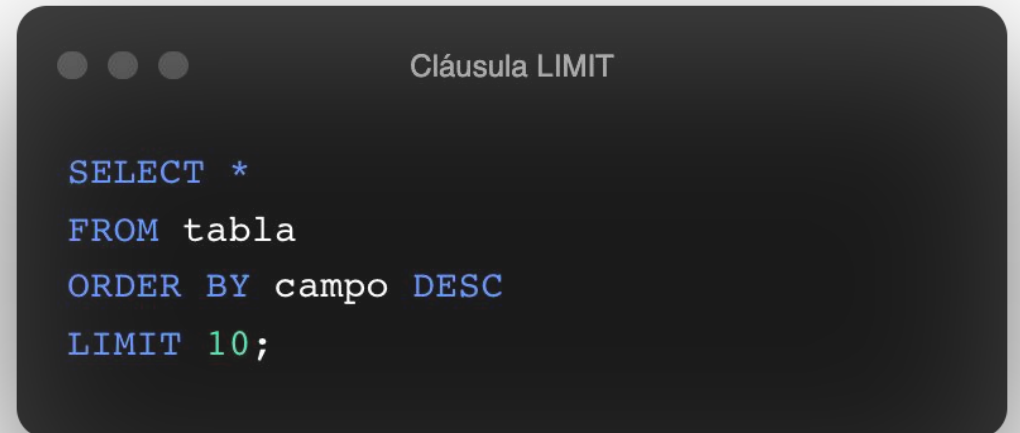


LIMIT

LIMIT

Corresponde a una cláusula SQL que se utiliza en las consultas de selección, para limitar el total de registros a visualizar.

La misma tiene algunas particularidades que debemos conocer para poder ejecutar consultas de selección efectivas.



```
SELECT *  
FROM tabla  
ORDER BY campo DESC  
LIMIT 10;
```

LIMIT

Puede recibir un parámetro, o dos.

Cuando definimos un solo parámetro en la consulta, se limitará a traer el total de registros correspondientes a dicho parámetro.

LIMIT CON 1 PARAMETRO

```
SELECT *  
FROM clientes  
ORDER BY DNI DESC  
LIMIT 10;
```


LIMIT

Cuando definimos LIMIT con dos parámetros, le indicamos a través del primero de ellos, desde qué registro deseamos empezar a visualizar la información.

El segundo parámetro pasa a ser el que limitará el total de registros a mostrar.

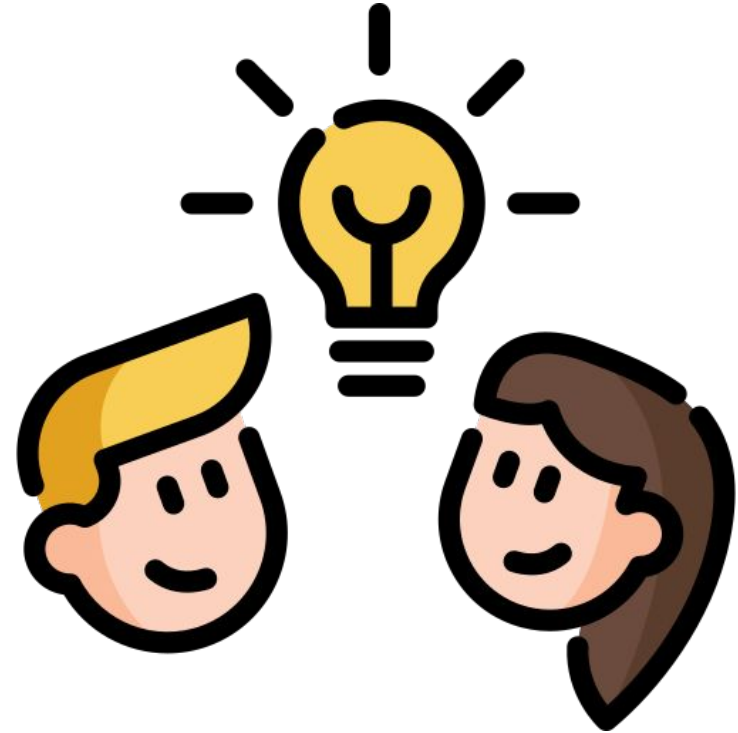
LIMIT CON 2 PARAMETROS

```
SELECT *  
FROM clientes  
ORDER BY DNI DESC  
LIMIT 1, 10;
```

LIMIT

Las tablas en MySQL comienzan a contar los registros en forma de Array de datos, como en los lenguajes de programación. El primer registro de la tabla tendrá la posición número 0, el segundo registro tendrá la posición número 1, y así sucesivamente.

Debemos prestar atención con esto porque podemos ejecutar una consulta de selección importante, dejando de lado el primero de los registros de ésta si especificamos el número 1 como primer parámetro.

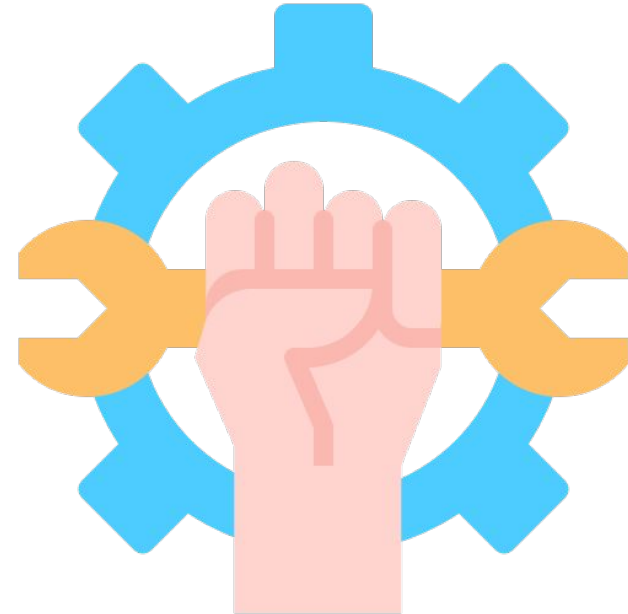


LIMIT

Aquí encontramos la primera diferencia entre MySQL con otro motor, como ser SQL Server, ya que en este último no existe la cláusula **LIMIT**.

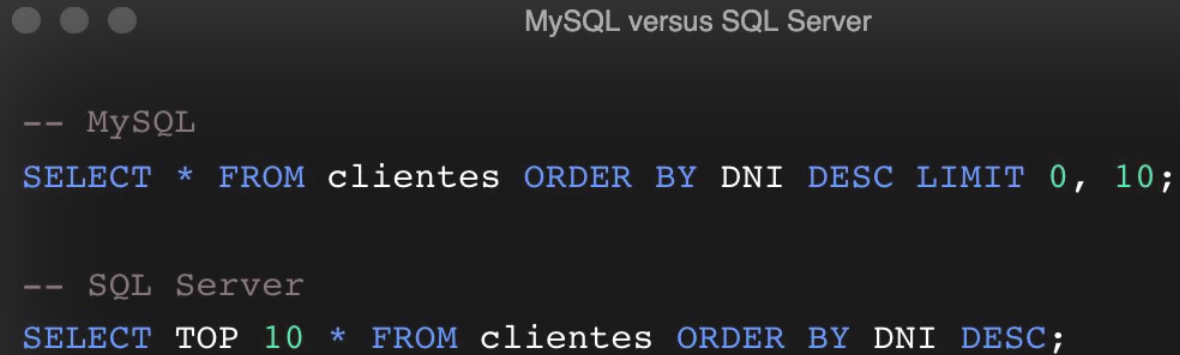
En su defecto, incluye la cláusula **TOP n** para limitar el total de registros que deseamos obtener en la consulta de selección.

Y solo recibe un único parámetro; el total numérico.



LIMIT

Referencia comparativa entre MySQL y SQL Server.



```
MySQL versus SQL Server

-- MySQL
SELECT * FROM clientes ORDER BY DNI DESC LIMIT 0, 10;

-- SQL Server
SELECT TOP 10 * FROM clientes ORDER BY DNI DESC;
```

DISTINCT

DISTINCT

La cláusula **DISTINCT** se usa en combinación con **SELECT** y nos permite filtrar de una consulta de selección, aquellos registros que pueden estar repetidos en la tabla, en base a un dato en particular.

La cláusula DISTINCT

```
SELECT DISTINCT campo  
FROM tabla;
```

DISTINCT

Debemos sumar al menos el nombre de un campo. También podemos sumar más de un campo dentro de la cláusula **DISTINCT**.

A mayor cantidad de campos definidos, más laxo será el resultado de la consulta.



La cláusula DISTINCT con más campos

```
SELECT DISTINCT campo, otroCampo, otroMas  
FROM tabla;
```

DISTINCT

Analizando una tabla como la siguiente, donde parte de sus datos se repiten:

¿Se imaginan cuál es el resultado de la primera consulta? 🙋

idCliente	nombre	apellido
1	Juana	Manso
2	Juana	Manso
3	Juana	Manso
4	Juana	Manso

¿Y el resultado de la segunda consulta? 🙋

Consulta 1 DISTINCT

```
-- consulta 1
SELECT DISTINCT idCliente, nombre, apellido
FROM clientes;
```

Consulta 2 DISTINCT

```
-- consulta 2
SELECT DISTINCT nombre, apellido
FROM clientes;
```


DISTINCT

Resultados:

idCliente	nombre	apellido
1	Juana	Manso
2	Juana	Manso
3	Juana	Manso
4	Juana	Manso

nombre	apellido
Juana	Manso

Consulta 1 DISTINCT

```
-- consulta 1
SELECT DISTINCT idCliente, nombre, apellido
FROM clientes;
```

Consulta 2 DISTINCT

```
-- consulta 2
SELECT DISTINCT nombre, apellido
FROM clientes;
```

Sección práctica

Pongámonos a tono con la base de datos Northwind.

Para ello, desarrollarás las siguientes consultas de ejecución sobre una tabla específica, de acuerdo a las indicaciones que verás en el siguiente Slide.



Prácticas

1. Ejecuta una consulta de selección sobre todos los campos de la tabla **Customers**
2. Ejecuta una consulta de selección de los siguientes campos de la tabla **Customers**:
 - **CustomerID, CompanyName, ContactName, ContactTitle, City, Phone**
 - Ordena esta consulta por el campo **CompanyName**
3. Ejecuta una consulta de selección sobre los siguientes campos de la tabla **Customers**:
 - **CustomerID, CompanyName, ContactName, ContactTitle**
 - Ordena esta consulta por el campo **ContactName** de forma descendente
4. Ejecuta una consulta de selección sobre todos los campos de la tabla **Customers**:
 - Ordena esta consulta por el campo **CustomerID**
 - Limita el total de registros a visualizar a 20
5. Ejecuta una consulta de selección sobre todos los campos de la tabla **Customers**:
 - Ordena esta consulta por el campo **ContactName**
 - **Limita el total de registros a visualizar: muestra solo 10 registros a partir del cliente número 10 de esta consulta resultante**

Muchas gracias.



Ministerio de Economía
Argentina

Secretaría de
Economía del Conocimiento

*primero
la gente*