

# SQL Analytics I

## Módulo II



## Temario

- SQL - Definición
- SQL - Estructura del Lenguaje
- SQL - Lenguaje de Manipulación de Datos (DML)
  - Árbol de Evaluación
  - SELECT (TOP | DISTINCT)
  - FROM
  - WHERE
  - ORDER BY

SQL

Definición

*SQL (Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas*

## SQL

## Definición

- SQL es el lenguaje más ampliamente implementado para las bases de datos relacionales
- De la misma manera que las matemáticas son el lenguaje de la ciencia, SQL es el lenguaje de las bases de datos relacionales
- SQL no solamente permite leer, modificar, insertar y eliminar datos dentro de la base de datos, sino que también se utiliza para definir y alterar su estructura

## SQL

## Estructura del Lenguaje

- Dentro de SQL encontramos los siguientes lenguajes
  - DDL (*Data Definition Lenguaje*)
  - **DML (*Data Manipulation Lenguaje*)**
  - DCL (*Data Control Lenguaje*)

## DDL (Data Definition Language)

- Es un Lenguaje que permite modificar la estructura de las tablas de una base de datos
- Permite por ejemplo:
  - Crear una tabla
  - Borrar una tabla
  - Vaciar el contenido de una tabla
  - Renombrar una tabla
  - Alterar los campos de una tabla

## DML (Data Manipulation Lenguaje)

- Es un Lenguaje que permite la manipulación de los datos
- Permite por ejemplo:
  - [Consultar registros](#)
  - Insertar registros en la Base de Datos
  - Actualizar registros
  - Borrar registros

## DCL (Data Control Lenguaje)

- El Lenguaje de Control de Datos, permite a un Administrador conceder derechos y establecer restricciones a los usuarios sobre las distintas bases de datos
- Básicamente permite:
  - Dar permisos a uno o varios usuarios o roles para realizar tareas determinadas
  - Eliminar permisos que previamente se han concedido
  - Modificar permisos



## SQL

## Estructura del Lenguaje

- SQL es un lenguaje declarativo. O sea, que especifica qué es lo que se quiere y no cómo conseguirlo
- El orden de ejecución interno de una sentencia puede afectar seriamente la eficiencia, por lo que es necesario llevar a cabo una optimización antes de su ejecución
- Utiliza Comandos orientados al manejo de conjuntos de registros

## SQL

## Estructura del Lenguaje

- No distingue entre mayúsculas y minúsculas, da igual como se escriba, siempre y cuando cada comando utilizado este correctamente escrito

## SQL

## Lenguaje de Manipulación de Datos

- Es utilizado principalmente para obtener datos:

*¿Qué datos se desean obtener?*

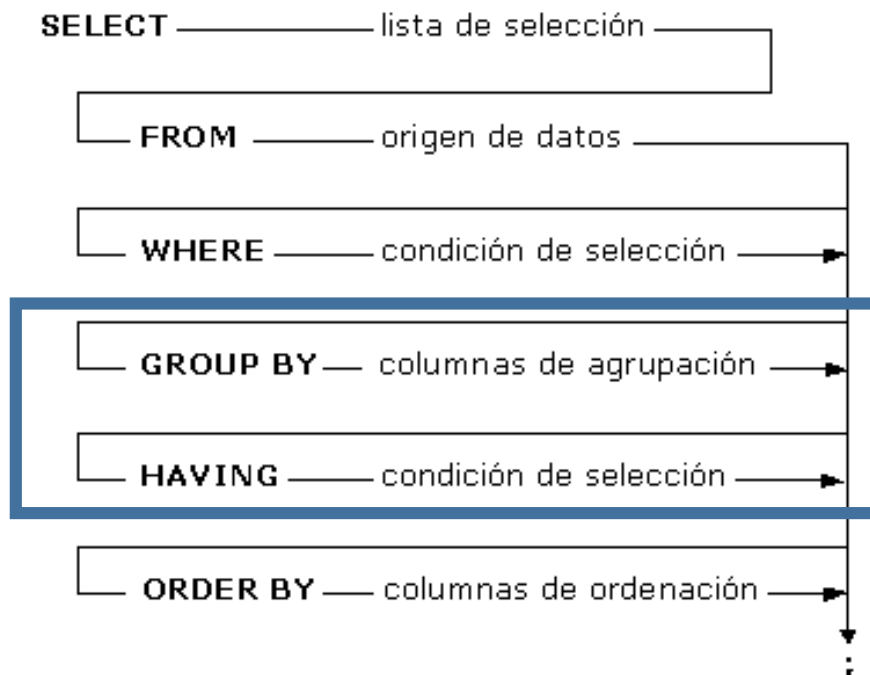
- Las bases de datos almacenan información de negocios.
- Es de suma importancia saber cómo obtener esta información y que sea realmente relevante para la empresa, entonces:

*¿Qué información se desea obtener?*

## SQL

## Lenguaje de Manipulación de Datos

- Sintaxis:



**Se utilizan principalmente con funciones de agregación (se verán más adelante)**

- Orden de ejecución:

1. FROM clause
2. ON clause
3. OUTER clause
4. WHERE clause
5. GROUP BY clause
6. HAVING clause
7. SELECT clause
8. DISTINCT clause
9. ORDER BY clause
10. TOP clause

```
SELECT DISTINCT <TOP_specification> <select_list>
FROM <left_table>
    <join_type> JOIN <right_table>
    ON <join_condition>
WHERE <where_condition>
GROUP BY <group_by_list>
HAVING <having_condition>
ORDER BY <order_by_list>
```

## SQL

## Lenguaje de Manipulación de Datos

- **SELECT**

- La sentencia SELECT permite indicar los campos que se requieren de la base de datos (los campos de las tablas necesarios para nuestra consulta y las funciones sobre estos campos)

- **SELECT DISTINCT**

- La sentencia SELECT DISTINCT permite obtener todos los registros (tuplas) no duplicados de una tabla

- **SELECT TOP**

- Permite indicar el número o porcentaje de registros que queremos obtener de la consulta

- FROM

- Enumera las tablas y vistas que contienen las columnas a las que se hace referencia en la lista de selección
- Los nombres de las tablas y vistas se pueden sustituir por alias mediante la cláusula AS
- **La cláusula FROM es una lista separada por comas de nombres de tablas, nombres de vistas y cláusulas JOIN (se verá más adelante)**



## SQL

## Lenguaje de Manipulación de Datos

- FROM

- La cláusula FROM "convierte" a un conjunto de tablas (clientes, teléfonos, facturas, etc) en una sola tabla en donde un registro será la unidad de análisis de nuestro interés (en general son facturas, productos, clientes, etc)

## SQL

## Lenguaje de Manipulación de Datos

- Consulta Básica de SQL (1 tabla):

```
SELECT [Nombre Campo 1], [Nombre Campo 2], [Nombre Campo n]  
FROM [Nombre Tabla 1]
```

Una consulta básica de SQL permite obtener datos de una tabla de manera sencilla, seleccionando los campos que se requieren (SELECT) e indicando la tabla correspondiente (FROM)

## SQL

## Lenguaje de Manipulación de Datos

- Consulta Básica de SQL

- Ejemplo:

```
SELECT FirstName, LastName
```

```
FROM Person.Person
```

- Esta consulta permite obtener nombre y apellido de todos los registros de una tabla llamada “*Person.Person*”
  - A simple vista no es información relevante pero en el contexto de una empresa puede ser la información de nombre y apellido de todos sus clientes:

***¿Tiene valor ésta información?***

FirstName	LastName
Pilar	Ackerman
Janaina Barreiro Gambaro	Bueno
Janeth	Esteves
Adrian	Dumitrascu
Alvaro	De Matos Miranda Filho
Gustavo	Camargo
Ciro	Bauer
Luis	Bonifaz
Humberto	Acevedo
José	Saraiva
Anibal	Sousa
Ramón	Cai
José	Lopez
Jésus	Alonso
Amy	Li
Jo	Zimmernan
Karin	Zimprich
Juanita	Zimmernan

## SQL

## Lenguaje de Manipulación de Datos

- Consulta Básica de SQL

- Ejemplo:

**SELECT \***

**FROM** Person.EmailAddress

- Utilizando el carácter “\*” luego del SELECT me permite indicar todos los campos de una tabla
  - En este ejemplo se obtienen todos los registros de la tabla “*Person.EmailAddress*”

Results		Messages	
	BusinessEntityID	EmailAddressID	EmailAddress
1	1	1	ken0@adventure-works.com
2	2	2	tem0@adventure-works.com
3	3	3	roberto0@adventure-works.com
4	4	4	rob0@adventure-works.com
5	5	5	gail0@adventure-works.com
6	6	6	jossef0@adventure-works.com
7	7	7	dylan0@adventure-works.com
8	8	8	diane1@adventure-works.com
9	9	9	gigi0@adventure-works.com
10	10	10	michael6@adventure-works.c...
11	11	11	ovidiu0@adventure-works.com
12	12	12	thierry0@adventure-works.com

## SQL

## Lenguaje de Manipulación de Datos

- **SELECT DISTINCT**

- Ejemplo:

```
SELECT DISTINCT JobTitle  
FROM HumanResources.Employee
```

- En este caso se quiere obtener los distintos tipos de cargos de los empleados de la Empresa
- Utilizando únicamente SELECT se obtendrían cargos repetidos, dado que es muy factible que existan empleados que ostenten el mismo cargo

	JobTitle
1	Accountant
2	Accounts Manager
3	Accounts Payable Specialist
4	Accounts Receivable Specialist
5	Application Specialist
6	Assistant to the Chief Financial Officer
7	Benefits Specialist
8	Buyer
9	Chief Executive Officer
10	Chief Financial Officer
11	Control Specialist
12	Database Administrator
13	Design Engineer
14	Document Control Assistant

- WHERE

- La cláusula WHERE permite establecer condicionales (filtros) para la selección

```
SELECT [Nombre Campo 1], [Nombre Campo 2], [Nombre Campo n]  
FROM [Nombre Tabla 1], [Nombre Tabla 2], [Nombre Tabla n]  
WHERE [Condición 1] AND [Condición 2] AND [Condición n]
```

## SQL

## Lenguaje de Manipulación de Datos

- WHERE

- Ejemplo:

```
SELECT FirstName, LastName
```

```
FROM Person.Person
```

```
WHERE FirstName = 'José'
```

- En este caso se quiere obtener sólo el nombre y apellido de las personas que se llaman José

Nótese que José se encuentra entre comillas simples dado que es un tipo de dato: Cadena de Caracteres

	FirstName	LastName
1	José	Hernandez
2	José	Jackson
3	José	Lopez
4	José	Martinez
5	José	Miller
6	José	Perez
7	José	Rodriguez
8	José	Saraiva
9	José	Saraiva

- ORDER BY

- La cláusula ORDER BY permite ordenar los resultados obtenidos por una o más columnas de manera ascendente o descendente

```
SELECT [Nombre Campo 1], [Nombre Campo 2], [Nombre Campo n]  
FROM [Nombre Tabla 1], [Nombre Tabla 2], [Nombre Tabla n]  
WHERE [Condición 1] AND [Condición 2] AND [Condición n]  
ORDER BY [NombreCampo1] ASC | DESC , [NombreCampo2] ASC | DESC
```



## SQL

## Lenguaje de Manipulación de Datos

- ORDER BY

- Ejemplo:

```
SELECT FirstName, LastName
```

```
FROM Person.Person
```

```
ORDER BY LastName DESC
```

- Para este ejemplo se ordena por apellido de manera descendente

Vale aclarar que si se omite el modificador ASC o DESC automáticamente se ordena por el campo indicado de manera ascendente (ASC)

	FirstName	LastName
1	Michael	Zwilling
2	Michael	Zwilling
3	Jake	Zukowski
4	Judy	Zugelder
5	Patricia	Zubaty
6	Carla	Zubaty
7	Karin	Zimprich
8	Karin	Zimprich
9	Tiffany	Zimmeman
10	Marc	Zimmeman
11	Krystal	Zimmeman

## SQL

## Lenguaje de Manipulación de Datos

- **SELECT TOP**

- Ejemplo:

```
SELECT TOP 10 FirstName, LastName
```

```
FROM Person.Person
```

- En este caso se quiere obtener el nombre y apellido de los primeros 10 empleados de la tabla “*Person.Person*”
- Utilizando únicamente SELECT se obtendrían todos los empleados (19972 empleados en la tabla)

Se puede utilizar también porcentaje , Ej: **SELECT TOP 5 PERCENT** , obteniendo el 5% de la cantidad de registros de la tabla

	FirstName	LastName
1	Syed	Abbas
2	Catherine	Abel
3	Kim	Abercrombie
4	Kim	Abercrombie
5	Kim	Abercrombie
6	Hazem	Abolrous
7	Sam	Abolrous
8	Humberto	Acevedo
9	Gustavo	Achong
10	Pilar	Ackerman

- SELECT TOP

- Ejemplo:

```
SELECT TOP 10 BusinessEntityID,  
              LoginID,  
              BirthDate  
FROM HumanResources.Employee  
ORDER BY BirthDate
```

	BusinessEntityID	LoginID	BirthDate
1	274	adventure-works\stephen0	1941-11-17
2	84	adventure-works\frank3	1942-04-03
3	256	adventure-works\frank2	1942-06-13
4	5	adventure-works\gail0	1942-10-29
5	140	adventure-works\prasanna0	1943-06-01
6	231	adventure-works\jo1	1944-05-25
7	63	adventure-works\maciej0	1945-03-02
8	40	adventure-works\jolynn0	1946-02-16
9	49	adventure-works\barry0	1946-04-27
10	53	adventure-works\diane0	1946-04-30

- En este caso se desea obtener ciertos datos de los primeros 10 empleados con mayor edad de la tabla *"HumanResources.Employee"*

## Resumen Módulo II

- SQL - Definición
- SQL - Estructura del Lenguaje
- SQL - Lenguaje de Manipulación de Datos (DML)
  - Árbol de Evaluación
  - SELECT (TOP | DISTINCT)
  - FROM
  - WHERE
  - ORDER BY