

# SQL SERVER APLICADO (SSA010)

Ariel Alexis Fierro Sáez  
[afierrosaez@gmail.com](mailto:afierrosaez@gmail.com)

# SQL

- El lenguaje de consulta estructurado (SQL) es un lenguaje de bases de datos normalizado, utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones sobre los datos o sobre la estructura de los mismos. Pero sin embargo cada motor de bases de datos tiene sus peculiaridades y lo hace diferente de cada motor, por lo tanto el lenguaje SQL normalizado (ANSI) no tendrá la misma respuesta en los diferentes motores de bases de datos.

# Componentes SQL

- Esta compuesto por comandos, clausulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

# Tipos de Comandos

- Las sentencias SQL se dividen, principalmente en las siguientes categorías:
  - DDL que permiten crear y definir nuevas bases de datos, campos e índices.
  - DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

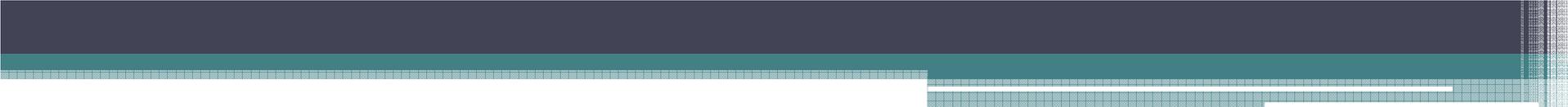
# Sentencias de Manipulación de Datos (DML)

- Seleccionan, manipulan o manejan los datos en los objetos de los esquemas ya existentes, esto permite hacer:
  - > Mostrar los datos de una o mas tablas, o de una o mas vistas.
  - > Modificar los valores de atributos en filas existentes.
  - > Eliminar filas de tablas o vistas.
  - > Agregar nuevas filas a una tabla o vista.
  - > Bloquear una tabla o vista, limitando temporalmente el acceso de otros usuarios.
  - > Ver el plan de ejecución de una sentencia SQL.

# Sentencias de Definición de datos

## DDL

- Definen, modifican o alteran la estructura y eliminan objetos del esquema:
  - Crear, alterar o modificar y eliminar objetos y otras estructuras de bases de datos.
  - Renombrarlas.
  - Eliminar los datos del objeto.
  - Obtener estadísticas.
  - Autorizar y revocar privilegios y roles.



# Instrucciones Select

- Esta instrucción la utilizamos para obtener información de la base de datos, nos permite sacar reportes y es el principal comando. Tiene otras aplicaciones que veremos mas adelante.

# Sintaxis Select

Select [atributos que deseamos mostrar]  
Into [variable]  
From [tablas utilizadas]  
Where [condiciones de filtrado]  
Group by [atributos de agrupación]  
Having [condiciones sobre los grupos]

Ejemplos:

```
select id_rol from rol;  
select * from rol;
```

# Modificadores

- Distinct, devuelve solamente filas únicas.
- TOP n [porcentaje], devuelve solamente las n primeras filas o el porcentaje.

```
Select distinct nombre  
From usuario
```

```
Select top 5 nombre  
From usuario
```

# Modificadores

```
Select top 50 percent id-rol  
From rol
```

Veamos los alias:

```
Select id_rol as "Rol"  
From rol
```

# Funciones

- Función que retorna el Maximo

```
Select MAX(salario) as “Mejor Pagado”
```

```
From salarios
```

```
go
```

# Funciones

- Min, selecciona el mínimo

```
Select MIN(salario) as Peor_Pagado  
From salarios
```

# Funciones

- Order By

```
Select emp_id, salario as Sueldo_Pagado  
From salarios  
Order by salario
```

## Usando tablas en FROM

- Se pueden utilizar también consultas y asignándoles un alias para luego utilizarlas en la consulta principal, es una forma alternativa a utilizar sub consultas.

```
Select id_empleado  
from Empleado e, (Select id_empleado from  
Ventas) as v  
where e.id_empleado <> v.id_empleado
```

# Operadores de Comparación

<code>=</code>	<i>Igual a</i>
<code>&gt;, &gt;=</code>	<i>Mayor a, Mayor o igual a</i>
<code>&lt;, &lt;=</code>	<i>Menor a, Menor o Igual a</i>
<code>&lt;&gt;</code>	<i>Diferente</i>
<code>!&gt;</code>	<i>no es mayor que</i>
<code>!&lt;</code>	<i>no es menor que</i>
<code>BETWEEN...AND...</code>	<i>Entre dos valores inclusive</i>
<code>IN(lista)</code>	<i>Empareja valores con la lista</i>
<code>LIKE</code>	<i>Empareja con un patrón</i>
<code>IS NULL</code>	<i>Es un valor nulo?</i>
<code>NOT</code>	<i>Verifica que no cumpla la condición</i>

# Operadores de Comparación

- *Compara todos los valores con los indicados en la lista, también puede ir una consulta, pero los datos resultantes deberán ser del mismo tipo.*

```
SELECT nombre,salario,nss_superv  
FROM empleado  
WHERE nss_superv IN ('888665555','333445555');
```

# LIKE y sus Comodines

*LIKE ejecuta búsquedas en base a patrones:*

- % coincide con una cadena de cero o mas caracteres.
- \_ indica un único carácter.
- [ ] coincide con el carácter único o el rango o conjunto.
- [^] Carácter comodín no en rango.

*WHERE title LIKE '%computer%' busca todos los títulos de libros que contengan la palabra 'computer' en el título.*

*WHERE au\_fname LIKE '\_ean' busca todos los nombres de cuatro letras que terminen en ean (Dean, Sean, etc.)*

*WHERE au\_lname LIKE '[C-P]arsen' busca apellidos de autores que terminen en arsen y empiecen por cualquier carácter individual entre C y P, como Carsen, Larsen, Karsen, etc.*

*WHERE au\_lname LIKE 'de[^l]%' busca todos los apellidos de autores que empiecen por de y en los que la siguiente letra no sea l.*

# Operador de Comparación

*Compara valores entre ambos datos, el mínimo el mayor, ambos son inclusivos.*

```
SELECT nombre,salario  
FROM empleado  
WHERE salario BETWEEN 1000 AND 1500;
```

# Operadores Lógicos

- *AND*                      *Ambos componentes son verdaderos*
  - *OR*                         *Verdad si alguno es TRUE*
  - *NOT*                        *Niega la condición*
- 
- *Primero se realizan los operadores de comparación, luego NOT, AND y OR (Precedencia de operadores)*

# Agrupaciones

- Group by se utiliza para agrupar filas y sacar datos de ellas, como por ejemplo:

```
Select d.departamento_id, sum(s.salario),
```

```
  s.empleado_id
```

```
From departamento as d, salarios as s
```

```
Group by d.departamento
```

# Condiciones de Agrupaciones

- Having se utiliza de manera parecida que el where, solo que las condiciones del having se aplican a los grupos:

```
Select d.departamento_id, sum(s.salario),  
       s.empleado_id  
From departamento as d, salarios as s  
Group by d.departamento  
Having sum(s.salario) > 20000
```

# Ejercicios

- El Administrador del Cinemark ha decidido generar una base de datos de registro de las películas que se exhiben. Para ello se deben considerar los siguientes requisitos:
  1. El cine está formado por varias salas que tienen un número, m<sup>2</sup>, cantidad de butacas, sistema de sonido y número de puertas.
  2. Las películas tienen un título, director, duración, restricción de edad, fecha de inicio y de término de exhibición.
  3. Las películas se pueden exhibir en varias salas en horarios y días diferentes. Así como también en cada sala se pueden exhibir varias películas en días y horarios diferentes.
  4. Se requiere saber cuántas personas ven cada película cada vez que se exhibe

# Ejercicios

- La empresa Metro S.A. quiere llevar su sistema de transporte a una base de datos. Considere lo siguiente:
- Existen diferentes líneas del metro, de cada línea se quiere conocer su número y longitud total.
- Cada línea está conformada por varias estaciones y una estación puede pertenecer a más de una línea. Las estaciones tienen un nombre.
- Una línea de metro tiene una estación de origen y una estación de destino final. Además una línea puede tener varias estaciones intermedias.
- A las líneas de metro se asignan uno o más trenes, con su número de tren y capacidad. Los trenes operan en una línea.
- Además existen trabajadores del metro (rut, nombre, dirección). Estos trabajadores son de uno de los siguientes tipos: administrativos, operarios o conductores.
- De los administrativos interesa conocer el cargo que desempeña. Además un empleado administrativo puede estar encargado una estación, teniendo cada estación un único encargado.
- Los operarios están asignados a una línea en particular, existiendo en cada línea varios operarios.
- Los conductores están asignados a un tren en particular, teniendo cada tren uno o más conductores.