

LENGUAJE Y COMANDOS SQL

Lenguaje de manipulación de datos (DML)
Comando SELECT

Lenguaje de manipulación de datos

- Lenguaje de cierta complejidad que permite el manejo y procesamiento del contenido de la base de datos.
- Sentencias DML:
 - **Select**
 - **Insert**
 - **Update**
 - **Delete**

Comando SELECT

- Este comando permite recuperar datos de una o varias tablas.
- Esta sentencia es de la más compleja y potente de las sentencias SQL.

Sintaxis:

```
SELECT lista_de_campos  
[ FROM nombre_tabla ]  
[ WHERE condicion_individual]  
[ GROUP BY campos_a_agrupar ]  
[ HAVING condicion_grupo ]  
[ ORDER BY campo_a_ordenar [ ASC | DESC ] ]
```

Comando SELECT

FROM

- Se utiliza en conjunto con la **cláusula FROM** con la cual se indica en qué tabla o tablas se tiene que buscar la información.
- Ejemplo 1:

```
SELECT * FROM Products
```

- Ejemplo 2:

```
SELECT ProductID, ProductName, UnitPrice FROM  
Products
```

Comando SELECT

WHERE

- La **cláusula WHERE** permite seleccionar únicamente las filas que cumplan con una condición de selección especificada. Sólo se mostrarán las filas para las cuales la evaluación de la condición sea verdadera (TRUE).
- Los campos con valores NULL no se incluirán en las filas de resultado.
- La condición de la **cláusula WHERE** puede ser cualquier condición válida o combinación de condiciones utilizando operadores lógicos (NOT, AND, OR) y relacionales (=, <, >, <=, >=).

Comando SELECT

- Ejemplo 1:

```
SELECT ProductID, ProductName, UnitPrice  
FROM Products  
WHERE UnitPrice > 15
```

- Ejemplo 2:

```
SELECT ProductID, ProductName, UnitPrice  
FROM Products  
WHERE UnitPrice >= 15 AND UnitPrice <= 50
```

Comando SELECT

- Ejemplo 3:

```
SELECT ProductID, ProductName, UnitPrice  
FROM Products  
WHERE UnitPrice BETWEEN 15 AND 50
```

- Ejemplo 4:

```
SELECT ProductID, ProductName, UnitPrice  
FROM Products  
WHERE NOT UnitPrice > 15
```

Comando SELECT

- Ejemplo 5:

```
SELECT ProductID, ProductName, UnitPrice  
FROM Products  
WHERE ProductID > 15 OR UnitPrice < 10
```


Comando SELECT

LIKE

- Puede utilizarse la cláusula LIKE para formar patrones de comparación con caracteres comodín.
- Los comodines más utilizados son:
 - `_`: Cualquier carácter.
 - `%`: Cualquier cadena de cero o más caracteres
 - `[]`: Cualquier carácter individual del intervalo ([a-z])
- Puede utilizarse únicamente LIKE o NOT LIKE, indicando igual o no igual respectivamente

Comando SELECT

- Ejemplo 1:

```
SELECT EmployeeID, LastName FROM Employees  
WHERE LastName LIKE 'D%'
```

- Ejemplo 2:

```
SELECT EmployeeID, LastName FROM Employees  
WHERE LastName LIKE '%N'
```

- Ejemplo 3:

```
SELECT EmployeeID, LastName, Title FROM  
Employees WHERE Title LIKE '%SALES%'
```

Comando SELECT

- Ejemplo 4:

- Seleccionar los campos EmployeeID y LastName de
 - empleados EXCEPTO aquellos donde el dato almacenado
 - en LastName comience con la letra D

```
SELECT EmployeeID, LastName FROM Employees  
WHERE LastName NOT LIKE 'D%'
```

- Ejemplo 5:

- Seleccionar todas las ordenes de pedido donde el dato
 - almacenado en el campo OrderID termine con los
 - digitos 0248

```
SELECT OrderID FROM [Order Details]  
WHERE OrderID LIKE '_0248'
```

Comando SELECT

ORDER BY

- Se puede hacer uso de la cláusula ORDER BY para mostrar los datos de forma ordenada.
- Si se utiliza en conjunto la cláusula ASC, los registros se mostrarán en orden ascendente (de menor a mayor) según el campo(s) especificado(s) en la cláusula ORDER BY. Si se utiliza la cláusula DESC, los registros serán mostrados en orden descendente (de mayor a menor).

Comando SELECT

- Ejemplo 1:

```
SELECT ProductID, ProductName, UnitPrice  
FROM Products  
ORDER BY ProductID ASC
```

La instrucción ASC es la que viene por defecto por lo tanto es opcional

- Ejemplo 2:

```
SELECT ProductID, ProductName, UnitPrice  
FROM Products  
ORDER BY ProductID DESC
```

Comando SELECT

DISTINCT

- La cláusula DISTINCT especifica que los registros con ciertos datos duplicados sean ignorados en el resultado.

- Ejemplo:

```
SELECT DISTINCT OrderID FROM  
[Order Details]
```

Comando SELECT

TOP N

- TOP n, especifica que solo se mostrará el primer conjunto de filas del resultado de la consulta.
- El conjunto de filas puede ser un número o un porcentaje de las filas (TOP n PERCENT)
- TOP n WITH TIES: Esta cláusula permite incluir en la selección, todos los registros que tengan el mismo valor del campo por el que se ordena

Comando SELECT

- Ejemplo 1:

```
SELECT TOP 5 OrderID, ProductID, Quantity  
FROM [Order Details]
```

- Ejemplo 2:

```
SELECT TOP 2 WITH TIES OrderID, ProductID,  
Quantity  
FROM [Order Details]  
ORDER BY OrderID
```


Comando SELECT

- Ejemplo 3:

--En el ejemplo siguiente se mostraran el
--10% de todos los pedidos almacenados en
--la tabla Order Details

```
SELECT TOP 10 PERCENT OrderID, ProductID,  
Quantity  
FROM [Order Details]
```

Comando SELECT

Renombrar columnas de una consulta.

- En ocasiones en las consultas es necesario colocarles un sobrenombre a las columnas, ya que en ocasiones en algunos resultados de la consulta algunas columnas se obtienen a partir de operaciones o las columnas tienen nombres en inglés y así se puede poner un nombre más entendible.
- Para colocar el sobrenombre a la columna se utiliza la instrucción AS.

Comando SELECT

- Ejemplo 1:

```
--Seleccionar los datos almacenados en el campo CategoryName  
-- de la tabla Categories y renombrar  
-- a la columna con el nombre Nombre de Categoria  
SELECT CategoryName AS [Nombre de Categoria]  
FROM Categories
```

- Ejemplo 2:

```
--Se quiere conocer cual seria la fecha de envio (ShippedDate) --  
--con un retraso de 5 dias  
--Mostrar los los campos OrderID, OrderDate y ShippedDate de la  
--tabla Orders  
SELECT OrderId, OrderDate, ShippedDate, ShippedDate + 5 AS  
RetrasoEnvio  
FROM Orders
```