

Base de Datos desde CERO

BLOQUE 02B

MCT. MCSA. MCP.
Ing. Gianfranco Manrique



Agenda

1. Uso de ORDER BY
2. Filtrado con la clausula WHERE
3. Predicados y Operadores
4. Filtrado con TOP
5. Filtrado con OFFSET-FETCH
6. Trabajando con valores desconocidos

1. Uso de ORDER BY

```
SELECT <LISTA_SELECT>  
FROM <TABLA_FUENTE>  
ORDER BY <LISTA_ORDER_BY>  
ASC|DESC;
```

- Utilícelo para ordenar filas de manera ascendente (ASC) y descendente (DESC)
- Puede utilizarlo con nombres de columnas, alias y posición (No recomendado).
- Puede utilizarlo también con columnas que no son parte del SELECT.

2. Filtrado con la clausula WHERE

```
SELECT <LISTA_SELECT>  
FROM <TABLA_FUENTE>  
WHERE <CONDICION_BUSQUEDA>  
      <COLUMNA>  
      <OPERADOR>  
      <EXPRESION>
```

- Los WHERE son construidas desde una condición de búsqueda y escritas como expresiones de #predicado.
- Un predicado requiere ser expresado como una condición lógica evaluada en VERDADERO o FALSO.
- Sólo las filas donde la evaluación del predicado sea VERDADERO serán pasadas a través del #filtro.

3. Predicados y Operadores

- Además de los operadores lógicos o constantes en la clausula WHERE.

OPERADOR	EJEMPLO
AND	Combina dos expresiones booleanas y retorna VERDADERO solamente si ambas expresiones lo son.
OR	Combina dos expresiones booleanas y retorna VERDADERO si alguna de las expresiones lo es.
NOT	Revierde el resultado de una condición de búsqueda
IN	Determina si un valor específico coincide con cualquier valor de una lista.
BETWEEN	Especifica un rango de valores a evaluar.
LIKE	Determina si una cadena de caracteres coincide con un patrón específico.

4. Filtrado con TOP

- Utilice TOP para limitar el número o porcentaje de filas devueltas.
- TOP WITH TIES incluye filas con valores que coinciden con la última fila.

```
SELECT TOP (N)|TOP (N) PERCENT  
FROM <TABLA_FUENTE>  
WHERE <CONDICION_BUSQUEDA>
```

```
SELECT TOP (N)|TOP (N) PERCENT  
WITH TIES  
FROM <TABLA_FUENTE>  
WHERE <CONDICION_BUSQUEDA>
```

5. Filtrado con OFFSET-FETCH

- Funciona similar a TOP pero basado en un número de registros y en un ordenamiento.
- Aplicable para paginado de información.

```
SELECT <LISTA_SELECT>  
FROM <TABLA_FUENTE>  
WHERE <CONDICION_BUSQUEDA>  
ORDER BY <LISTA_ORDER_BY>  
OFFSET <VALOR_OFFSET> ROW(S)  
FETCH FIRST|NEXT <VALOR_FETCH> ROW(S) ONLY
```

6. Trabajando con valores desconocidos

- Clausulas como ON, WHERE y HAVING filtran NULLs.
- Restricciones del tipo CHECK aceptan NULLs.
- ORDER BY, DISTINCT tratan NULLs como iguales.

EXPRESION IS [NOT] NULL