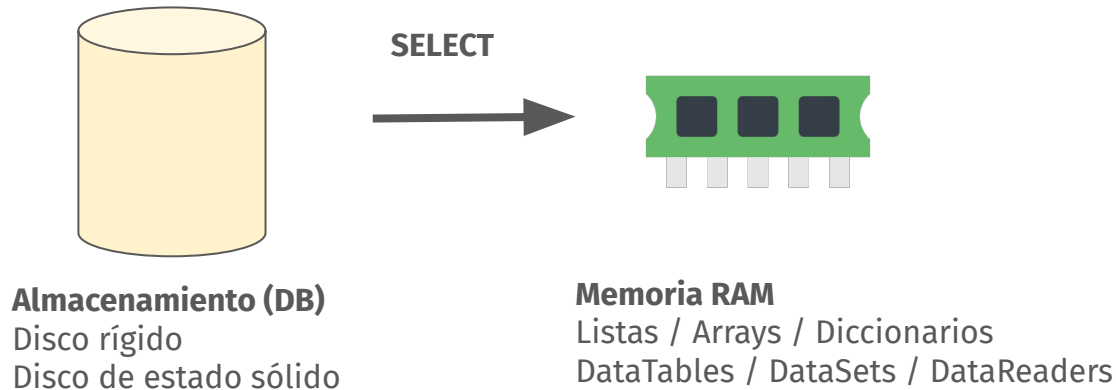


# **Base de Datos II**

Consultas de selección

# Consultas de selección

Son consultas que nos permiten seleccionar información de una o más tablas. Permiten realizar filtros, ordenamientos, transformaciones y resúmenes de los datos seleccionados.



# Sintaxis básica

```
SELECT Columna1, Columna2, Columna3 FROM MiTabla
```

```
SELECT Apellidos, Nombres, Sueldo FROM Empleados
```



Selecciona los apellidos, nombres y sueldos de la tabla empleados

```
SELECT * FROM Empleados
```



Selecciona todas las columnas de la tabla empleados

# DISTINCT

Remueve filas duplicadas de la selección

```
SELECT DISTINCT Columna1, Columna2, Columna3 FROM MiTabla
```

```
SELECT DISTINCT Sueldo FROM Empleados
```

10000	10000
20000	20000
20000	30000
30000	
30000	

Si hay más de un empleado que cobra exactamente el mismo sueldo, entonces dicho sueldo no se repetirá en el resultado de la selección, apareciendo una sola vez.

```
SELECT Sueldo, Moneda From Empleados;
```

```
→ 1000 USD  
1000000 ARS  
→ 1000 USD  
1000 ARS
```

```
1000 USD  
1000000 ARS  
1000 ARS
```

Select Distinct Sueldo, Moneda  
From Empleados

# ORDER BY

Realiza un ordenamiento del resultado a partir de uno o varios criterios.

```
SELECT Columna1, Columna2, Columna3 FROM MiTabla ORDER BY  
Columna1 ASC
```

```
SELECT Apellidos, Nombres, Nacimiento FROM Empleados ORDER BY  
Nacimiento ASC
```

Ordena el listado por fecha de nacimiento. El criterio es ascendente por lo que aparecerán primeros los que nacieron hace más tiempo y últimos los que nacieron recientemente.

## Criterios:

ASC → Ascendente

DESC → Descendente

# TOP

Permite limitar la cantidad de registros que devolverá la consulta

```
SELECT TOP 5 Columna1, Columna2 FROM MiTabla ORDER BY Columna1 ASC
```

```
SELECT TOP 3 Apellidos, Nombres, AñoIngreso  
FROM Empleados ORDER BY AñoIngreso ASC
```

Obtiene apellidos, nombres y año de ingreso de los 3 empleados más antiguos

# WHERE

Excluye filas del resultado de la selección a partir de uno o varias condiciones que deben cumplirse

```
SELECT Columna1, Columna2 FROM MiTabla WHERE Columna1 = 100
```

```
SELECT Apellidos, Nombres, Nacimiento FROM Empleados  
WHERE Apellidos = 'Lopez' OR Apellidos = 'Perez'
```

Selecciona los apellidos, nombres y fecha de nacimiento de los empleados cuyo apellido sea Lopez o Perez.

# OPERADORES

La cláusula WHERE acepta los operadores relacionales para establecer proposiciones lógicas y operadores lógicos para combinar varias proposiciones lógicas

```
SELECT Apellidos, Nombres, Nacimiento FROM Empleados  
WHERE AñoIngreso > 1990 AND Nombres <> 'Maria'  
!=
```

Selecciona los apellidos, nombres y fecha de nacimiento de los empleados que registren un año de ingreso posterior a 1990 y cuyos nombres no sean Maria.



# OPERADORES

El lenguaje SQL dispone de otros tipos de operadores para simplificar operaciones entre expresiones

Operador	Descripción
BETWEEN	Permite comparar entre rangos cerrados → [Desde; Hasta]
IN	Permite comparar un conjunto de valores separados por coma
IS NULL	Permite determinar si un valor es o no nulo
LIKE	Permite buscar patrones en el valor de comparación

# Anexo: Comodines en LIKE

Comodín	Descripción	Ejemplo	Coincidencias
%	Sustituye cero, uno o varios caracteres	nombre LIKE 'A%'	"Ana", "Andrés", "Antonio"
_	Sustituye un solo carácter	nombre LIKE 'J_n'	"Jan", "Jon", "Jen", "J n"
[ ]	Coincide con cualquier carácter dentro de los corchetes (solo en SQL Server)	nombre LIKE 'Jo[nm]'	"Jon", "Jom"
[ ^ ]	Excluye cualquier carácter dentro de los corchetes (solo en SQL Server)	nombre LIKE 'Jo[^n]'	"Jom", "Jop" (pero no "Jon")
-	Define un rango de caracteres dentro de [ ] (solo en SQL Server)	nombre LIKE 'J[a-c]n'	"Jan", "Jbn", "Jcn"