
SQL - Predicados



I. Predicados

➔ *Introducción*

- ❑ **Predicado** ≈ Expresa una condición entre valores. Su resultado puede ser *Verdadero*, *Falso* o *Desconocido*.
- ❑ Los predicados se pueden clasificar en:
 - **Simple**
 - **Compuestos:** Son combinaciones de otros predicados (simples o compuestos, con los operadores lógicos AND, OR y NOT)
- ❑ **SELECT Subordinada(anidada):** Sentencia SELECT que está dentro de otra. Cuando se utiliza en un predicado, si su valor es único, es decir, una tabla con una única fila y una única columna, se dice que es una **Select Escalar** (se utiliza en predicados de comparación).

I. Predicados

➔ *Lógica Trivaluada*

- La existencia de valores *Nulos* o *Desconocidos* en las columnas de una tabla, obliga a definir cuál será el comportamiento de las operaciones con esos valores:
 - De comparación
 - Aritméticas
 - Lógicas

I. Predicados

→ Comportamiento *Nulos*

□ Nulos en operaciones de **comparación**

```
valor1 oper valor2
```

- oper: =, <, >, <>, <=, >=
- Tabla de Verdad

Operandos	Resultado
Alguno o ambos operandos son <i>Nulo</i> o <i>Desconocido</i>	<i>Desconocido</i>
Ninguno de los operandos es <i>Nulo</i> o <i>Desconocido</i>	valor1 oper valor2

I. Predicados

→ Comportamiento *Nulos*

□ Nulos en operaciones **aritméticas**

```
valor1 oper valor2
```

- oper: +, -, *, /
- Tabla de Verdad

Operandos	Resultado
Alguno o ambos operandos son <i>Nulo</i> o <i>Desconocido</i>	<i>Desconocido</i>
Ninguno de los operandos es <i>Nulo</i> o <i>Desconocido</i>	valor1 oper valor2

I. Predicados

→ *Lógica Trivaluada (LTV)*

□ LTV en operaciones **lógicas**

valor1 **oper** valor2

- oper: AND, OR, NOT
- Tablas de Verdad

AND	V	F	D
V	V	F	D
F	F	F	F
D	D	F	D

OR	V	F	D
V	V	V	V
F	V	F	D
D	V	D	D

NOT	V	F	D
	F	V	D

II. Predicados Básicos

□ SINTAXIS

```
expr1 oper expr2
```

- **oper:** =, <, >, <>, <=, >=
- Se especifican en la cláusula WHERE (y en otras)
- **Ejemplo:** *Obtener por orden alfabético los nombres de los empleados cuyos sueldos igualan o superan al de Claudia Fierro en más de un 50%*

SELECT escalar (Un único valor)

```
SELECT NOMEM  
FROM TEMPLE  
WHERE SALAR >=  
  
ORDER BY NOMEM
```

```
(SELECT SALAR*1.5  
FROM TEMPLE  
WHERE NOMEM = 'FIERRO, CLAUDIA')
```

II. Predicados Básicos

→ *Valores Nulos*

□ **RECORDAR:**

- La comparación se realiza teniendo en cuenta que si alguno de los comparandos es **Nulo** → el predicado es **Desconocido**.
- Una sentencia subordinada que produce como resultado una tabla vacía, su valor como expresión es **Nulo**.

- **Ejemplo:** *Obtener por orden alfabético los nombres de los empleados cuyos sueldos igualan o superan al 50% del de Antonio López*

```
SELECT NOMEM
FROM TEMPLE
WHERE SALAR >= (SELECT SALAR*0.5
                FROM TEMPLE
                WHERE NOMEM = 'LOPEZ, ANTONIO')
ORDER BY NOMEM
```


III. Predicado NULL

□ SINTAXIS

```
expr1 IS [NOT] NULL
```

- Permite averiguar si el valor de una expresión es o no *Nulo*.
- Evaluación del predicado:

expr1	Resultado
Nulo	Verdadero
No Nulo	Falso

III. Predicado NULL

□ SINTAXIS

```
expr1 IS [NOT] NULL
```

- **Ejemplo:** *Obtener los nombres de los departamentos que no dependen funcionalmente de otros.*



IV. Predicados CUANTIFICADOS

□ SINTAXIS

SELECT subordinada



```
expr1 oper [SOME | ALL] (sent1)
```

- **oper:** =, <, >, <>, <=, >= que [al menos|que todos] un miembro de (sent1)
- Compara un valor (expr1) con la colección de filas que resulta de una SELECT Subordinada (sent1).
- Al contrario que ocurría en los predicados básicos, se admite que el resultado de la SELECT Subordinada tenga **varias filas**, pero **una sola columna** si ésta va precedida de SOME o ALL.
- P. ej: *Obtener por orden alfabético los nombres de los empleados cuyo salario supera al máximo salario de los empleados del departamento 122*

```
SELECT NOMEM FROM TEMPLE
WHERE SALAR > ALL (SELECT SALAR
                   FROM TEMPLE
                   WHERE NUMDE = 122)
ORDER BY NOMEM
```

IV. Predicados CUANTIFICADOS

□ SINTAXIS

SELECT subordinada

`expr1 oper [SOME] (sent1)`

- oper "que al menos un miembro" de (sent1)
- $\text{expr1} > \text{SOME}(150, 230) \approx (\text{expr1} > 150) \text{ OR } (\text{expr1} > 230)$
- Evaluación del predicado cuantificado SOME:

sent1	Result. SOME
Tabla con 1 ó más filas, y al menos una de ellas al compararlas dan Verdadero	V
Tabla con 1 ó más filas, y todas ellas al compararlas dan Falso	F
Tabla con 1 ó más filas, y ninguna de ellas al compararlas da Verdadero y alguna da Desconocido	D
Tabla con cero filas (vacía)	F

IV. Predicados CUANTIFICADOS

■ SINTAXIS

SELECT subordinada

`expr1 oper [ALL] (sent1)`

- oper “que todos los miembros” de (sent1)
- $\text{expr1} > \text{ALL}(150, 230) \approx (\text{expr1} > 150) \text{ AND } (\text{expr1} > 230)$
- Evaluación del predicado cuantificado ALL:

sent1	Result. ALL
Tabla con 1 ó más filas, y todas ellas al compararlas dan Verdadero	V
Tabla con 1 ó más filas, y al menos una de ellas al compararlas da Falso	F
Tabla con 1 ó más filas, y ninguna de ellas al compararlas da Falso y alguna da Desconocido	D
Tabla con cero filas (vacía)	V

IV. Predicados CUANTIFICADOS

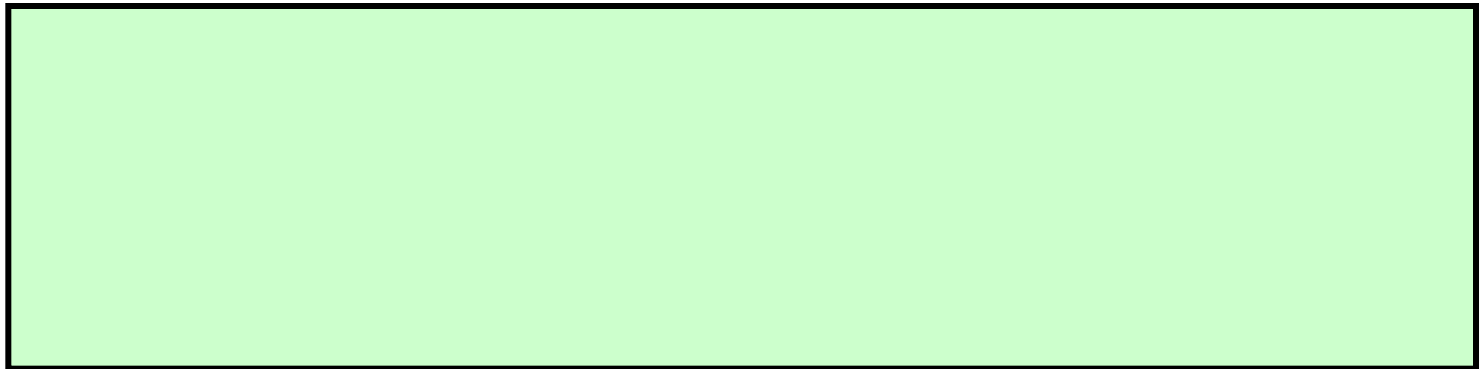
→ *Ejemplo*

□ SINTAXIS

SELECT subordinada

```
expr1 oper [SOME | ALL] (sent1)
```

- **Ejemplo:** *Obtener por orden alfabético los nombres de los empleados cuyo salario supera al máximo salario de los empleados del departamento 123.*



IV. Predicados CUANTIFICADOS

→ Ejemplo

□ SINTAXIS

SELECT subordinada

```
expr1 oper [SOME | ALL] (sent1)
```

- **Ejemplo:** *Obtener por orden alfabético los nombres de los empleados cuyo sueldo supera en tres veces y media o más al mínimo salario de los empleados del departamento 122.*



V. Predicado *BETWEEN*

□ SINTAXIS

```
expr1 [NOT] BETWEEN expr2 AND expr3
```

- Permite comparar si un valor está comprendido entre otros dos, ambos inclusive, o no.
- $V1 \text{ BETWEEN } V2 \text{ AND } V3 \approx (V1 \geq V2) \text{ AND } (V1 \leq V3)$
- Evaluación del predicado **BETWEEN**:

	Result. BETWEEN
Si alguno de los valores es Nulo	<i>Desconocido</i>
Si ninguno de los valores es Nulo	$(V1 \geq V2) \text{ AND } (V1 \leq V3)$

V. Predicado *BETWEEN*

□ SINTAXIS

```
expr1 [NOT] BETWEEN expr2 AND expr3
```

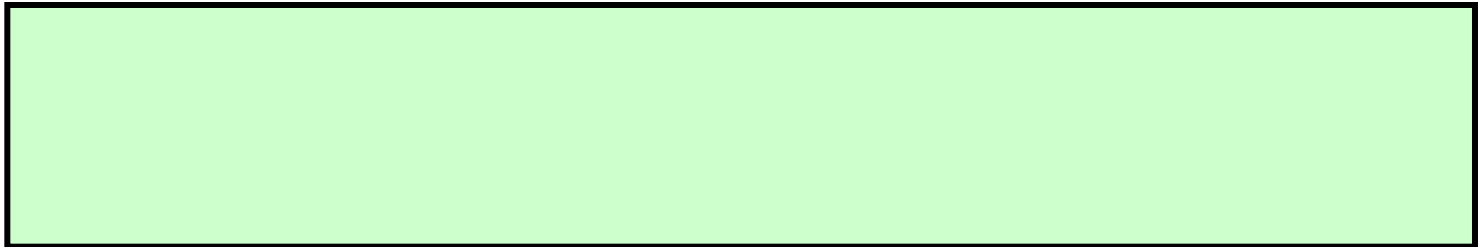
- Permite comparar si un valor está comprendido entre otros dos, ambos inclusive, o no.
- $V1 \text{ NOT BETWEEN } V2 \text{ AND } V3 \approx V1 < V2 \text{ OR } V1 > V3$
- Evaluación del predicado **NOT BETWEEN**:

	Result. NOT BETWEEN
Si V1 es <i>Nulo</i>	D
Si V1 no es <i>Nulo</i> y V2 y V3 son <i>Nulos</i>	D
Si V1 no es <i>Nulo</i> y V2 es <i>Nulo</i> y V3 no es <i>Nulo</i>	V si $V1 > V3$
Si V1 no es <i>Nulo</i> y V3 es <i>Nulo</i> y V2 no es <i>Nulo</i>	V si $V1 < V2$

V. Predicado *BETWEEN*

➔ *Ejemplo*

- ▣ **Ejemplo 1:** *Obtener por orden alfabético los nombres de los empleados cuyo salario está entre 2500 y 3000 euros.*



VI. Predicado *LIKE* (Operaciones sobre cadenas)

□ SINTAXIS


```
expr1 [NOT] LIKE expr2
```

- El tipo de datos de `expr1` y `expr2` debe ser *Texto*.
- Es una comparación de igualdad pero permite comodines:
 - %: se puede sustituir por cualquier número de caracteres (0 ó más)
 - _: se sustituye por 1 carácter
- Ejemplos:
 - Ej1: `SELECT NOMEM FROM TEMPLE WHERE NOMEM LIKE '_'`: buscaría los nombres de 1 carácter
 - Ej2: `SELECT NOMEM FROM TEMPLE WHERE NOMEM LIKE '%'`: buscaría los nombres de 1 ó más caracteres
- Evaluación del Predicado

	X LIKE Y
X o Y Nulos	Desconocido
X e Y vacíos (cadenas de longitud 0)	Verdadero
X vacío e Y no vacío	Falso
X no vacío e Y vacío	Falso
X e Y no vacíos ni nulos	Verdadero si el valor de X está incluido entre los que se pueden generar a partir Y con los caracteres comodín

VI. Predicado *LIKE*

- ▣ **Ejemplo 1:** *Obtener por orden alfabético los nombres de los empleados cuyo primer apellido es Mora o empieza por Mora.*

A large, empty rectangular box with a black border, intended for the results of the SQL query for Example 1.

- ▣ **Ejemplo 2:** *Obtener por orden alfabético los nombres de los empleados cuyo nombre de pila empieza por Valeriana.*

A large, empty rectangular box with a black border, intended for the results of the SQL query for Example 2.

VII. Predicado *IN*

SELECT subordinada de 1 única columna

■ SINTAXIS

```
[expr1 | (expr3[,...])] [NOT] IN [(sent1) | (expr2 [,exprm])]
```

- Permite preguntar si el valor de *expr1* pertenece a un conjunto de valores especificados detrás de la cláusula **IN** o no (**NOT**)
- El resultado de *sent1* puede tener cualquier número de filas, inclusive cero (tabla vacía), pero una sola columna.
- *expr1* **IN**(150,230) \approx *expr1* **=SOME**(150,230) \approx (*expr1*=150) **OR** (*expr1*=230)
- *expr1* **NOT IN**(150,230) \approx *expr1* **<>ALL**(150,230) \approx (*expr1*<>150) **AND** (*expr1*<>230)

- **Ejemplo:** *Obtener por orden alfabético los nombres de los empleados cuya extensión telefónica es 250 ó 750*



VII. Predicado *IN*

□ SINTAXIS

SELECT subordinada de 1 única columna

```
[expr1 | (expr3 [, ...])] IN [(sent1) | (expr2 [, exprm])]
```

- Permite preguntar si el valor de *expr1* pertenece a un conjunto de valores especificados detrás de la cláusula **IN**.
- El resultado de *sent1* puede tener cualquier número de filas, inclusive cero (tabla vacía), pero una sola columna.
- *expr1* **IN** (150,230) \approx *expr1* **=SOME** (150,230) \approx (*expr1*=150) **OR** (*expr1*=230)
- **Evaluación del Predicado:** *expr1* **IN** listaValores

	IN
<i>expr1</i> no es Nulo y es igual a alguno de los valores de la lista de valores (listaValores)	V
<i>expr1</i> no es Nulo y NO es igual a ninguno de los valores de la lista de valores (listaValores)	F
<i>expr1</i> es Nulo	D

VII. Predicado *NOT IN*

□ SINTAXIS

SELECT subordinada de 1 única columna

```
[expr1 | (expr3 [, ...])] NOT IN [(sent1) | (expr2 [, exprm])]
```

- Permite preguntar si el valor de *expr1* **no** pertenece a un conjunto de valores especificados detrás de la cláusula **IN**
- El resultado de *sent1* puede tener cualquier número de filas, inclusive cero (tabla vacía), pero una sola columna.
- *expr1* **NOT IN**(150,230) \approx *expr1* **<>ALL**(150,230) \approx (*expr1***<>**150) **AND** (*expr1***<>**230)
- **Evaluación del Predicado:** *expr1* **NOT IN** listaValores

	NOT IN
<i>expr1</i> no es Nulo y es igual a alguno de los valores de la lista de valores (listaValores)	F
<i>expr1</i> no es Nulo y NO es igual a ninguno de los valores de la lista de valores (listaValores)	V
alguno de los valores de listaValores es Nulo	D
<i>expr1</i> es Nulo	D

VII. Predicado *IN*

➔ *Ejemplo*

- *Obtener por orden alfabético los nombres de los empleados que trabajan en el mismo departamento que Adriana Alba o Dorotea Flor.*

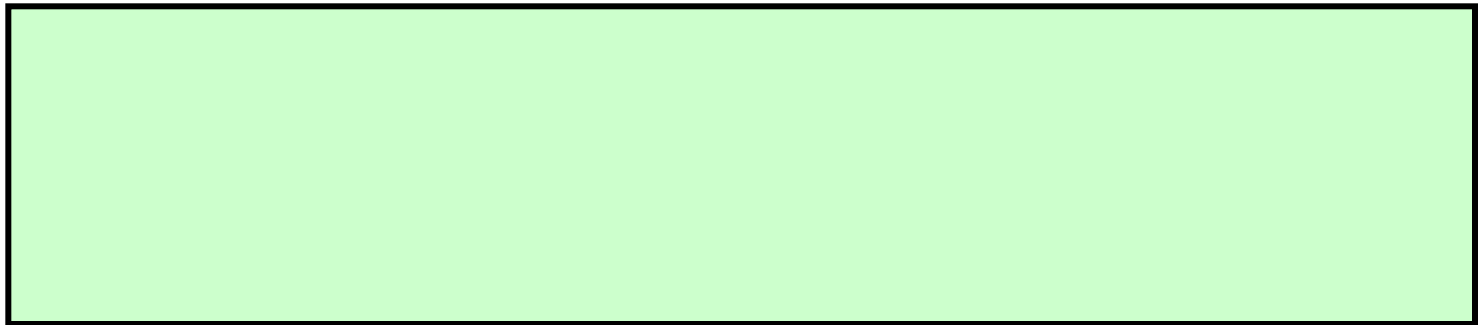


VIII. Predicado *EXISTS*

□ SINTAXIS

EXISTS (sent)

- Permite averiguar si la sentencia subordinada tiene una o más filas (*Verdadero*), o es una tabla vacía (*Falso*).
- **Ejemplo:** *Obtener los nombres de los centros de trabajo, siempre y cuando haya alguno en la calle Atocha.*



VIII. Predicado *EXISTS*

- **NOT EXISTS:**

- Al contrario que **EXISTS**:

- Devuelve Verdadero: Si no hay tuplas en el resultado de la consulta
 - Devuelve Falso: En caso de que haya tuplas en el resultado

IX. Predicados Compuestos

- Son combinaciones de otros predicados, simples o compuestos, con los operadores lógicos AND, OR y NOT.
- **Ejemplo:** *Obtener por orden alfabético los nombres y comisiones de los empleados del departamento 110 si hay en él algún empleado que tenga comisión.*

