

## Condiciones de selección

Las **condiciones de selección** son las condiciones que pueden aparecer en la cláusula **WHERE**.

En SQL tenemos cinco condiciones básicas:

el **test de comparación**

el **test de rango**

el **test de pertenencia a un conjunto**

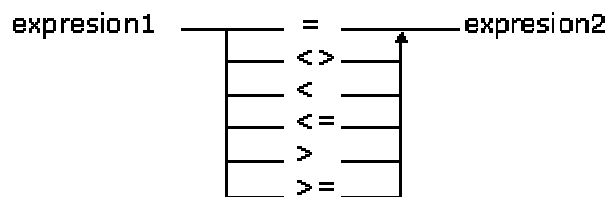
el **test de valor nulo**

el **test de correspondencia con patrón**.

el **test de comparación**.

Compara el valor de una expresión con el valor de otra.

La sintaxis es la siguiente:



= igual que

<> distinto de

< menor que

<= menor o igual

> mayor que

>= mayor o igual

---

**SELECT numemp, nombre**

Lista los empleados cuyas ventas superan su cuota

**FROM empleados**

**WHERE ventas > cuota**

---

**SELECT numemp, nombre**

Lista los empleados contratados antes del año 88 (cuya fecha de contrato sea anterior al 1 de enero de 1988).

**FROM empleados**

---

---

**WHERE contrato < #01/01/1988#**

¡¡Ojo!!, las fechas entre almohadillas # # deben estar con el formato mes,dia,año aunque tengamos definido otro formato para nuestras fechas.

---

**SELECT numemp, nombre**

**FROM empleados**

**WHERE YEAR(contrato) < 1988**

Este ejemplo obtiene lo mismo que el anterior pero utiliza la función year(). Obtiene los empleados cuyo año de la fecha de contrato sea menor que 1988.

---

**SELECT oficina**

**FROM oficinas**

**WHERE ventas < objetivo \* 0.8**

Lista las oficinas cuyas ventas estén por debajo del 80% de su objetivo.

Hay que utilizar siempre el punto decimal aunque tengamos definida la coma como separador de decimales.

---

**SELECT oficina**

**FROM oficinas**

**WHERE dir = 108**

Lista las oficinas dirigidas por el empleado 108.

---

### Test de rango (BETWEEN).

Examina si el **valor** de la expresión está **comprendido entre** los **dos valores** definidos por exp1 y exp2.

Tiene la siguiente sintaxis:

expresion 

---

**SELECT numemp, nombre**

**FROM empleados**

**WHERE ventas BETWEEN 100000  
AND 500000**

Lista los empleados cuyas ventas estén comprendidas entre 100.000 y 500.00

---

**SELECT numemp, nombre**

**FROM empleados**

**WHERE (ventas >= 100000) AND  
(ventas <= 500000)**

Obtenemos lo mismo que en el ejemplo anterior. Los paréntesis son opcionales.

---

### Test de pertenencia a conjunto (IN)

Examina si el **valor** de la expresión es uno de los valores **incluidos en la lista de valores**.

Tiene la siguiente sintaxis:

Expresion — IN — ( valor ) →  
                                  ↑  
                                  └─ , ─┘

---

**SELECT numemp, nombre, oficina**      Lista los empleados de las oficinas 12, 14 y 16  
**FROM empleados**  
**WHERE oficina IN (12,14,16)**

---

**SELECT numemp, nombre**                      Obtenemos lo mismo que en el ejemplo anterior.  
**FROM empleados**                              Los paréntesis son opcionales.  
**WHERE (oficina = 12) OR (oficina = 14) OR (oficina = 16)**

---

#### Test de valor nulo (IS NULL)

Una condición de selección puede dar como resultado el valor verdadero TRUE, falso FALSE o nulo NULL.

Cuando una **columna** que interviene en una condición de selección **contiene** el **valor nulo**, el **resultado** de la condición no es verdadero ni falso, sino **nulo**, **sea cual sea el test** que se haya utilizado. Por eso si queremos listar las filas que tienen valor en una determinada columna, no podemos utilizar el test de comparación, la condición oficina = null devuelve el valor nulo sea cual sea el valor contenido en oficina. Si queremos preguntar si una columna contiene el valor nulo debemos utilizar un **test especial**, el test de valor nulo.

Tiene la siguiente sintaxis:

nbcolumna — IS — ( NOT ) — NULL — →

Ejemplos:

---

**SELECT oficina, ciudad**                      Lista las oficinas que no tienen director.  
**FROM oficinas**  
**WHERE dir IS NULL**

---

**SELECT numemp, nombre**                      Lista los empleados asignados a alguna oficina  
**FROM empleados**                              (los que tienen un valor en la columna oficina).  
**WHERE oficina IS NOT NULL**

---

#### Test de correspondencia con patrón (LIKE)

Se utiliza cuando queremos **utilizar caracteres comodines** para formar el valor con el comparar.

Tiene la siguiente sintaxis:

nbcolumna  $\begin{array}{c} \text{---} \\ \text{---} \end{array}$  **LIKE**  $\begin{array}{c} \text{---} \\ \text{---} \end{array}$  patron  $\rightarrow$   
 $\begin{array}{c} \text{---} \\ \text{---} \end{array}$  **NOT**  $\begin{array}{c} \text{---} \\ \text{---} \end{array}$

Los comodines más usados son los siguientes:

? representa un carácter cualquiera

\* representa cero o más caracteres

# representa un dígito cualquiera (0-9)

Ejemplos:

---

<b>SELECT numemp, nombre</b>	Lista los empleados cuyo nombre empiece por Luis (Luis seguido de cero o más caracteres).
<b>FROM empleados</b>	
<b>WHERE nombre LIKE 'Luis*'</b>	

---

<b>SELECT numemp, nombre</b>	Lista los empleados cuyo nombre contiene Luis, en este caso también saldría los empleados José Luis (cero o más caracteres seguidos de LUIS y seguido de cero o más caracteres).
<b>FROM empleados</b>	
<b>WHERE nombre LIKE '*Luis*'</b>	

---

<b>SELECT numemp, nombre</b>	Lista los empleados cuyo nombre contenga una a como tercera letra (dos caracteres, la letra a, y cero o más caracteres).
<b>FROM empleados</b>	
<b>WHERE nombre LIKE '??a*'</b>	

---