



### Caso práctico



#### 7 A continuación se muestran ejemplos de uso de la cláusula LIKE:

LIKE 'Director'      la cadena 'Director'.

LIKE 'M%'            cualquier cadena que empiece por 'M'.

LIKE '%X%'           cualquier cadena que contenga una 'X'.

LIKE '\_\_\_M'          cualquier cadena de 3 caracteres terminada en 'M'.

LIKE 'N\_'            una cadena de 2 caracteres que empiece por 'N'.

LIKE '\_R%'           cualquier cadena cuyo segundo carácter sea una 'R'.

Hemos de tener en cuenta que las mayúsculas y minúsculas son significativas ('m' no es lo mismo que 'M') y que las constantes alfanuméricas deben encerrarse siempre entre comillas simples.

- A partir de la tabla EMPLE, obtén aquellos apellidos que empiecen por una 'J':

```
SQL> SELECT APELLIDO FROM EMPLE WHERE APELLIDO LIKE 'J%';
APELLIDO
-----
JIMENEZ
JIMENO
```

- Obtén aquellos apellidos que tengan una 'R' en la segunda posición:

```
SQL> SELECT APELLIDO FROM EMPLE WHERE APELLIDO LIKE '_R%';
APELLIDO
-----
ARROYO
```

- Obtén aquellos apellidos que empiecen por 'A' y tengan una 'O' en su interior:

```
SQL> SELECT APELLIDO FROM EMPLE WHERE APELLIDO LIKE 'A%O%';
APELLIDO
-----
ARROYO
ALONSO
```

A continuación, consideramos la tabla LIBRERÍA, cuya descripción es la siguiente:

SQL> DESC LIBRERIA		
Nombre	¿Nulo?	Tipo
-----	-----	-----
TEMA	NOT NULL	CHAR (15)
ESTANTE		CHAR (1)
EJEMPLARES		NUMBER (2)

(Continúa)



### 3. Introducción a SQL

#### 3.11 NULL y NOT NULL

(Continuación)

- Consultamos las filas de la tabla LIBRERIA cuyo tema sea 'LABORES'; usamos el operador '=':

```
SQL> SELECT * FROM LIBRERIA WHERE TEMA='LABORES';  
TEMA          E EJEMPLARES
```

```
-----
```

- Hacemos lo mismo, pero ahora manejando el operador LIKE:

```
SQL> SELECT * FROM LIBRERIA WHERE TEMA LIKE 'LABORES';
```

ninguna fila seleccionada

La consulta no devuelve nada, debido a que la columna TEMA está definida con el tipo CHAR(15). El tipo CHAR rellena blancos a la derecha hasta formar la longitud de 15 caracteres. Para que funcione la consulta tendremos que utilizar el comodín %:

```
SQL> SELECT * FROM LIBRERIA WHERE TEMA LIKE 'LABORES%';
```

```
TEMA          E EJEMPLARES  
-----  
LABORES      B          20
```

Si la columna TEMA fuese de tipo VARCHAR2, no sería necesario usar el comodín % con el operador LIKE.

## 3.11 NULL y NOT NULL

Se dice que una columna de una fila es NULL si está completamente vacía. Para comprobar si el valor de una columna es nulo empleamos la expresión: *columna IS NULL*. Si queremos saber si el valor de una columna no es nulo, usamos la expresión: *columna IS NOT NULL*. Cuando comparamos con valores nulos o no nulos no podemos utilizar los operadores de igualdad, mayor o menor.

Por ejemplo, a partir de la tabla EMPLE, consultamos los apellidos de los empleados cuya comisión es nula:

```
SQL> SELECT APELLIDO FROM EMPLE WHERE COMISION IS NULL;
```

Si queremos consultar los apellidos de los empleados cuya comisión no sea nula teclearemos esto:

```
SQL> SELECT APELLIDO FROM EMPLE WHERE COMISION IS NOT NULL;
```



## 3.12 Comprobaciones con conjuntos de valores

Hasta ahora, todas las comprobaciones lógicas que hemos visto comparan una columna o expresión con un valor. Por ejemplo:

```
OFICIO = 'ANALISTA' AND DEPT_NO=10.
```

Pero también podemos comparar una columna o una expresión con una lista de valores utilizando los operadores **IN** y **BETWEEN**.

### A. Operador IN

El operador **IN** nos permite comprobar si una expresión pertenece o no (**NOT**) a un conjunto de valores, haciendo posible la realización de comparaciones múltiples. Su formato es:

```
<expresión> [NOT] IN (lista de valores separados por comas)
```

#### Caso práctico



##### 8 La lista de valores está formada por números:

- Consulta los apellidos de la tabla EMPLE cuyo número de departamento sea 10 o 30:

```
SQL> SELECT APELLIDO FROM EMPLE WHERE DEPT_NO IN(10,30);
```

- Consulta los apellidos de la tabla EMPLE cuyo número de departamento no sea ni 10 ni 30:

```
SQL> SELECT APELLIDO FROM EMPLE WHERE DEPT_NO NOT IN(10,30);
```

La lista de valores está formada por cadenas:

- Consulta los apellidos de la tabla EMPLE cuyo oficio sea 'VENDEDOR', 'ANALISTA' o 'EMPLEADO':

```
SQL> SELECT APELLIDO FROM EMPLE WHERE OFICIO IN ('VENDEDOR', 'ANALISTA', 'EMPLEADO');
```

- Consulta los apellidos de la tabla EMPLE cuyo oficio no sea ni 'VENDEDOR' ni 'ANALISTA' ni 'EMPLEADO':

```
SQL> SELECT APELLIDO FROM EMPLE WHERE OFICIO NOT IN ('VENDEDOR', 'ANALISTA', 'EMPLEADO');
```



### 3. Introducción a SQL

#### 3.12 Comprobaciones con conjuntos de valores

## B. Operador BETWEEN

El operador **BETWEEN** comprueba si un valor está comprendido o no (**NOT**) dentro de un rango de valores, desde un valor inicial a un valor final. Su formato es:

<expresión> [**NOT**] BETWEEN valor\_inicial AND valor\_final



### Caso práctico

- 9 A partir de la tabla EMPLE, obtén el apellido y el salario de los empleados cuyo salario esté comprendido entre 1500 y 2000:

```
SQL> SELECT APELLIDO, SALARIO FROM EMPLE WHERE SALARIO BETWEEN 1500 AND 2000;
```

APELLIDO	SALARIO
-----	-----
ARROYO	1500
SALA	1625
MARTIN	1600
MUÑOZ	1690

En esta sentencia se visualizan los empleados cuyo salario es 1500, 2000 o cualquier valor comprendido entre ambos.

- A partir de la tabla EMPLE, obtener el apellido y el salario de los empleados cuyo SALARIO no esté comprendido entre 1500 y 2000:

```
SQL> SELECT APELLIDO, SALARIO FROM EMPLE WHERE SALARIO NOT BETWEEN 1500 AND 2000;
```

APELLIDO	SALARIO		
-----	-----		
SANCHEZ	1040	REY	4100
JIMENEZ	2900	TOVAR	1350
NEGRO	3005	ALONSO	1430
CEREZO	2885	JIMENO	1335
GIL	3000	FERNANDEZ	2900

10 filas seleccionadas.

Con esta sentencia tenemos los empleados cuyo SALARIO es menor que 1500 o mayor que 2000.