



UD3

CONSULTA Y MODIFICACIÓN DE BASES DE DATOS

MP_0484

Bases de Datos

3.3 Consultas básicas II

Conjuntos de valores

Hasta ahora comparábamos una columna con un posible valor, pero hemos de tener en cuenta que también se puede comparar con varios valores.

- **IN** nos permite comparar con una lista de valores, para verificar si pertenece o no a esa lista (figura 1).

[NOT] IN (lista valores separados por comas)

```

7 • select empapellido, dptono
8   from emple
9   where dptono IN (20,30)
10  order by dptono;

```

| empapellido | dptono |
|-------------|--------|
| SANCHEZ | 20 |
| JIMÉNEZ | 20 |
| GIL | 20 |
| ALONSO | 20 |
| FERNÁNDEZ | 20 |
| ARROYO | 30 |
| SALA | 30 |
| MARTÍN | 30 |
| NEGRO | 30 |
| TOVAR | 30 |
| JIMENO | 30 |

Figura 1. Uso de IN

De la misma forma, podemos usar el operador not IN para seleccionar los valores que se excluyan de la lista (figura 2).

```

select EmpApellido, dptono
from emple
where dptono not IN (10,20);

```

| EmpApellido | dptono |
|-------------|--------|
| ARROYO | 30 |
| SALA | 30 |
| MARTÍN | 30 |
| NEGRO | 30 |
| TOVAR | 30 |
| JIMENO | 30 |

Figura 2. Uso de NOT IN

- **BETWEEN** por su parte comprueba si el valor está comprendido entre un rango de valores, desde uno inicial a uno final (figura 3).

[NOT] BETWEEN valor_inicial AND valor_final

| | | |
|---|-------------|---------|
| <code>select empapellido, salario</code> | empapellido | salario |
| <code>from emple</code> | JIMÉNEZ | 3800 |
| <code>where salario between 2000 and 6000;</code> | NEGRO | 3700 |
| | CEREZO | 3100 |
| | GIL | 3900 |
| | FERNÁNDEZ | 3900 |

Figura 3. Uso de BETWEEN

Funciones en SQL

Ahora vamos a conocer algunas de las funciones más comunes que se utilizan en SQL, con las cuales podremos llevar a cabo operaciones con números y cadenas de caracteres algo más avanzadas. En primer lugar (figura 4) tenemos las operaciones matemáticas básicas, tales como el valor absoluto, el módulo o resto y la potencia, entre otros.

| | |
|------------------------|---|
| Valores simples | ABS(n) |
| | % Devuelve el resto de la división. |
| | POWER(m, exponente) Calcula la potencia. |
| | ROUND(número[,m]) Redondea. |
| | SIGN(valor) Devuelve el signo (+1, 0, -1). |
| | SQRT(n) Devuelve la raíz cuadrada. |

Figura 4. Operaciones matemáticas básicas

En la figura 5 podemos ver operaciones que aplican para conjuntos de números.

| | |
|--------------------------|---|
| Grupos de valores | AVG(n) Calcula la media, sin contar los NULL. |
| | COUNT(* expresión) * Cuenta el número de filas, expresión cuenta los no nulos. |
| | MAX(expresión) Selecciona el valor máximo. |
| | MIN(expresión) Selecciona el valor mínimo. |
| | SUM(expresión) Obtiene la suma de los valores de expresión. |
| | VARIANCE(expresión) Obtiene la varianza de expresión. |

Figura 5. Operaciones matemáticas sobre conjuntos de números

Por último, veamos en la figura 6 las operaciones que podemos realizar con cadenas de caracteres.

| | |
|-------------------------|---|
| Valores carácter | CHAR(n) Devuelve el código ASCII entero de n. |
| | CONCAT(cad1, cad2,...) Concatena cadenas de caracteres. |
| | LOWER(cad) Convierte a minúsculas. |
| | UPPER(cad) Convierte a mayúsculas. |
| | LEFT(Cad, n), RIGHT (Cad, n) Coge n caracteres de la cadena izquierda o derecha respectivamente. |
| | REPLACE(CadOrigen, CadBuscar, CadCambio) Sustituye. |

Figura 6. Operaciones con cadenas de caracteres

Existen muchas más funciones. El objetivo no es conocerlas todas, sino saber que existen y tener en cuenta la documentación de referencia de SQL para buscarlas cuando las necesitemos. Por ejemplo, la función LENGTH(cadena): devuelve el número de caracteres de la cadena, excluyendo los espacios en blanco al final (figura 7). Veamos a continuación algunas aplicaciones prácticas de estas funciones.

| 4 • <code>select empapellido,length(EmpApellido)</code> | | | | | | | | | | | | | | | | | | | |
|---|---|-------------|---------------------|---------|---|--------|---|------|---|---------|---|--------|---|-------|---|--------|---|-----|---|
| 5 <code>from emple;</code> | | | | | | | | | | | | | | | | | | | |
| | <table> <tr> <th>empapellido</th><th>length(EmpApellido)</th></tr> <tr><td>SANCHEZ</td><td>7</td></tr> <tr><td>ARROYO</td><td>6</td></tr> <tr><td>SALA</td><td>4</td></tr> <tr><td>JIMÉNEZ</td><td>8</td></tr> <tr><td>MARTÍN</td><td>7</td></tr> <tr><td>NEGRO</td><td>5</td></tr> <tr><td>CEREZO</td><td>6</td></tr> <tr><td>GIL</td><td>3</td></tr> </table> | empapellido | length(EmpApellido) | SANCHEZ | 7 | ARROYO | 6 | SALA | 4 | JIMÉNEZ | 8 | MARTÍN | 7 | NEGRO | 5 | CEREZO | 6 | GIL | 3 |
| empapellido | length(EmpApellido) | | | | | | | | | | | | | | | | | | |
| SANCHEZ | 7 | | | | | | | | | | | | | | | | | | |
| ARROYO | 6 | | | | | | | | | | | | | | | | | | |
| SALA | 4 | | | | | | | | | | | | | | | | | | |
| JIMÉNEZ | 8 | | | | | | | | | | | | | | | | | | |
| MARTÍN | 7 | | | | | | | | | | | | | | | | | | |
| NEGRO | 5 | | | | | | | | | | | | | | | | | | |
| CEREZO | 6 | | | | | | | | | | | | | | | | | | |
| GIL | 3 | | | | | | | | | | | | | | | | | | |

Figura 7. Uso de la función LENGTH

En la figura 8, podemos ver como la función LOWER nos pasa las cadenas a minúsculas.

| <code>select empapellido, lower(EmpApellido)</code> | | | | | | | | | | | | | | | | | | | | | |
|---|--|-------------|--------------------|---------|---------|--------|--------|------|------|---------|---------|--------|--------|-------|-------|--------|--------|-----|-----|-----|-----|
| <code>from emple;</code> | | | | | | | | | | | | | | | | | | | | | |
| | <table> <tr> <th>empapellido</th><th>lower(EmpApellido)</th></tr> <tr><td>SANCHEZ</td><td>sanchez</td></tr> <tr><td>ARROYO</td><td>arroyo</td></tr> <tr><td>SALA</td><td>sala</td></tr> <tr><td>JIMÉNEZ</td><td>jiménez</td></tr> <tr><td>MARTÍN</td><td>martín</td></tr> <tr><td>NEGRO</td><td>negro</td></tr> <tr><td>CEREZO</td><td>cerezo</td></tr> <tr><td>GIL</td><td>gil</td></tr> <tr><td>REY</td><td>rey</td></tr> </table> | empapellido | lower(EmpApellido) | SANCHEZ | sanchez | ARROYO | arroyo | SALA | sala | JIMÉNEZ | jiménez | MARTÍN | martín | NEGRO | negro | CEREZO | cerezo | GIL | gil | REY | rey |
| empapellido | lower(EmpApellido) | | | | | | | | | | | | | | | | | | | | |
| SANCHEZ | sanchez | | | | | | | | | | | | | | | | | | | | |
| ARROYO | arroyo | | | | | | | | | | | | | | | | | | | | |
| SALA | sala | | | | | | | | | | | | | | | | | | | | |
| JIMÉNEZ | jiménez | | | | | | | | | | | | | | | | | | | | |
| MARTÍN | martín | | | | | | | | | | | | | | | | | | | | |
| NEGRO | negro | | | | | | | | | | | | | | | | | | | | |
| CEREZO | cerezo | | | | | | | | | | | | | | | | | | | | |
| GIL | gil | | | | | | | | | | | | | | | | | | | | |
| REY | rey | | | | | | | | | | | | | | | | | | | | |

Figura 8. Uso de la función LOWER

En la figura 9, vemos el uso de la función COUNT.



```

6 • select count(oficio) "Nº oficios"
7   from emple;

```

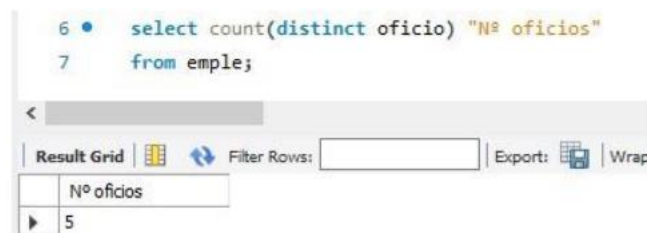
Result Grid | Filter Rows: | Export:

| Nº oficios |
|------------|
| 14 |

| EmpNo | EmpApellido | Oficio | EmpleDir | FechaAlta | Salario | Co |
|-------|-------------|------------|----------|------------|---------|----|
| 7369 | SANCHEZ | EMPLEADO | 7902 | 1980-12-12 | 1500 | 1 |
| 7499 | ARROYO | VENDEDOR | 7698 | 2001-02-22 | 1600 | 1 |
| 7521 | SALA | VENDEDOR | 7698 | 2000-02-02 | 1600 | 1 |
| 7566 | JIMÉNEZ | DIRECTOR | 7839 | 1981-02-02 | 3800 | 1 |
| 7654 | MARTÍN | VENDEDOR | 7698 | 2001-10-10 | 1600 | 1 |
| 7698 | NEGRO | DIRECTOR | 7839 | 1995-12-12 | 3700 | 1 |
| 7782 | CEREZO | DIRECTOR | 7839 | 2015-02-02 | 3100 | 1 |
| 7788 | GIL | ANALISTA | 7566 | 2010-01-05 | 3900 | 1 |
| 7839 | REY | PRESIDENTE | 7839 | 2002-05-05 | 6500 | 1 |
| 7844 | TOVAR | VENDEDOR | 7698 | 2001-06-06 | 1950 | 0 |
| 7876 | ALONSO | EMPLEADO | 7788 | 2003-04-06 | 1430 | 1 |
| 7900 | JIMENO | EMPLEADO | 7698 | 2005-08-08 | 1200 | 1 |
| 7902 | FERNÁNDEZ | ANALISTA | 7566 | 2000-03-12 | 3900 | 1 |
| 7934 | MUÑOZ | EMPLEADO | 7782 | 2003-09-09 | 1600 | 1 |

Figura 9. Uso de la función COUNT

Si combinamos COUNT y DISTINCT, podemos contar el número de oficios distintos que hay en la empresa (figura 10).



```

6 • select count(distinct oficio) "Nº oficios"
7   from emple;

```

Result Grid | Filter Rows: | Export: | Wrap

| Nº oficios |
|------------|
| 5 |

Figura 10. Uso combinado de COUNT y DISTINCT

Utilizando MAX podemos conocer el valor máximo de una serie, en el caso de la figura 11 nos permite conocer el salario más elevado.



```

8 • select max(salario)
9   from emple;

```

Result Grid | Filter Rows: | Export:

| max(salario) |
|--------------|
| 6500 |

Figura 11. Uso de la función MAX