



UD3

CONSULTA Y MODIFICACIÓN DE BASES DE DATOS

MP_0484

Bases de Datos

3.2 Consultas básicas

Introducción

La versión original de SQL fue desarrollada por IBM bajo el nombre Sqel en un proyecto llamado System R. SQL (structured query language), viene de Sqel. Actualmente, SQL es el estándar usado en las bases de datos relacionales. Referentes de estandarización como ISO (Organización Internacional de Estandarización) o ANSI (Instituto Nacional Estadounidense de Estándares) avalan este publicando dicho estándar y normas sobre él.

Una de sus características principales es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de las BD, así como hacer cambios en ellas. Se considera un lenguaje sencillo, ya que no hay que especificar cómo hacer las cosas, solo es necesario decir qué es lo que se quiere; son lenguajes funcionales o lógicos no procedimentales, cuya programación es más compleja.

Consultas simples

La forma de acceder a los datos de la BD es realizar consultas sobre esta. Para ello, se dispone de la sentencia select la cual podrá ser más o menos compleja en función de los datos a seleccionar.

```
SELECT [ALL|DISTINCT]
        {column1, column2,...,column | *}
FROM nombreTabla1 [,nombreTabla2,...]
[WHERE Condición]
[ORDER BY column [DESC|ASC] [,column [DESC|ASC] ]
[GROUP BY column]
[HAVING where condición];
```

- **SELECT** será la sentencia más usada en BD relacional, sirve para seleccionar, es obligatorio comenzar así. Seguido del select van los nombres de las columnas que vamos a seleccionar separadas por comas.
- **FROM** aquí hay que indicar la tabla de la que vamos a coger los valores. Si anteriormente nos hemos posicionado en una BD (comando use NombreBD), solamente podremos el nombre de la tabla; si no nos hemos posicionado, hace falta indicar el nombre de la base de datos y el nombre de la tabla, separadas estas por un punto (.): (nombreBD.nombreTabla).

El resultado de select es el conjunto de las filas de la tabla y de las columnas seleccionadas. Este resultado no se almacena en la BD, es una respuesta visual en forma de tabla que representa los valores que en ese momento tiene la BD. Veamos a continuación un ejemplo (figura 1).

```
select EmpApellido, Oficio
from emple_depart.emple;
```

EmpApellido	Oficio
SANCHEZ	EMPLEADO
ARROYO	VENDEDOR
SALA	VENDEDOR
JIMÉNEZ	DIRECTOR
MARTÍN	VENDEDOR
NEGRO	DIRECTOR
CEREZO	DIRECTOR
GIL	ANALISTA
REY	PRESIDENTE
TOVAR	VENDEDOR
ALONSO	EMPLEADO
JIMENO	EMPLEADO
FERNÁNDEZ	ANALISTA
MUÑOZ	EMPLEADO

Figura 1. Consulta básica

- **WHERE** si agregamos la cláusula where a la sentencia select, nos va a permitir elegir qué filas de la tabla queremos mostrar, en concreto, mostrará solo las filas que cumplan la condición where. Podemos ver un ejemplo en la figura 2.

```
select EmpApellido
from emple
where oficio ="vendedor";
```

EmpApellido
ARROYO
SALA
MARTÍN
TOVAR

Figura 2. Consulta empleando WHERE

- **ORDER BY** especifica el criterio de clasificación del resultado de la consulta. Los valores posibles son asc (ascendente, es la ordenación por defecto) o desc (descendente). Podemos ver un ejemplo en la figura 3.

```
select EmpApellido, Oficio
from emple
order by EmpApellido;
```

EmpApellido	Oficio
ALONSO	EMPLEADO
ARROYO	VENDEDOR
CEREZO	DIRECTOR
FERNÁNDEZ	ANALISTA
GIL	ANALISTA
JIMÉNEZ	DIRECTOR
JIMENO	EMPLEADO
MARTÍN	VENDEDOR
MUÑOZ	EMPLEADO
NEGRO	DIRECTOR
REY	PRESIDENTE
SALA	VENDEDOR
SANCHEZ	EMPLEADO
TOVAR	VENDEDOR

Figura 3. Consulta con resultados en orden ascendente

- **ALL y DISTINCT** selecciona todas las posibles filas, aunque los valores se repitan o no, según se ponga all, para que aparezcan todas las filas (es el valor por defecto) o distinct, para que no se repitan. Distinct y distinctrow son parámetros sinónimos, así que siempre usaremos distinct (figura 4).





13 • select oficio	15 • select distinct oficio																					
14 from emple;	16 from emple;																					
<div>Result Grid   Filter Rows: <input type="text"/></div> <table><tr><th>oficio</th></tr><tr><td>EMPLEADO</td></tr><tr><td>VENDEDOR</td></tr><tr><td>VENDEDOR</td></tr><tr><td>DIRECTOR</td></tr><tr><td>VENDEDOR</td></tr><tr><td>DIRECTOR</td></tr><tr><td>DIRECTOR</td></tr><tr><td>ANALISTA</td></tr><tr><td>PRESIDENTE</td></tr><tr><td>VENDEDOR</td></tr><tr><td>EMPLEADO</td></tr><tr><td>EMPLEADO</td></tr><tr><td>ANALISTA</td></tr><tr><td>EMPLEADO</td></tr></table>	oficio	EMPLEADO	VENDEDOR	VENDEDOR	DIRECTOR	VENDEDOR	DIRECTOR	DIRECTOR	ANALISTA	PRESIDENTE	VENDEDOR	EMPLEADO	EMPLEADO	ANALISTA	EMPLEADO	<div>Result Grid   Filter Rows: <input type="text"/></div> <table><tr><th>oficio</th></tr><tr><td>EMPLEADO</td></tr><tr><td>VENDEDOR</td></tr><tr><td>DIRECTOR</td></tr><tr><td>ANALISTA</td></tr><tr><td>PRESIDENTE</td></tr></table>	oficio	EMPLEADO	VENDEDOR	DIRECTOR	ANALISTA	PRESIDENTE
oficio																						
EMPLEADO																						
VENDEDOR																						
VENDEDOR																						
DIRECTOR																						
VENDEDOR																						
DIRECTOR																						
DIRECTOR																						
ANALISTA																						
PRESIDENTE																						
VENDEDOR																						
EMPLEADO																						
EMPLEADO																						
ANALISTA																						
EMPLEADO																						
oficio																						
EMPLEADO																						
VENDEDOR																						
DIRECTOR																						
ANALISTA																						
PRESIDENTE																						

Figura 4. Uso de all y distinct

- **ALIAS** si el nombre de la columna resulta demasiado poco descriptivo, se puede crear un texto para describir dicha columna en el propio select. Para ello, detrás del nombre del campo se debe poner entre comillas el texto que queremos que aparezca como descriptor de la columna (figura 5).

```
select EmpApellido "Apellido"
from emple;
```

	Apellido
▶	SANCHEZ
	ARROYO
	SALA
	JIMÉNEZ
	MARTÍN
	NEGRO
	CEREZO
	GIL
	REY
	TOVAR
	ALONSO
	JIMENO
	FERNÁN...
	MUÑOZ

Figura 5. Utilizando un alias en la consulta

- **CARÁCTER *** usar el carácter * en lugar de indicar los campos de la tabla, va a devolver todas las columnas de la tabla. Es una forma de mostrar la tabla sin nombrar las columnas (figura 6).

```
select *
from depart;
```

DptNo	DptoNombre	DptoLoc
10	CONTABILIDAD	SEVILLA
20	INVESTIGACIÓN	MADRID
30	VENTAS	BARCELONA
40	PRODUCCIÓN	BILBAO

Figura 6. Uso de * en consultas

- **Operadores y comparadores** hasta ahora hemos utilizado como columnas de la tabla resultado el nombre del campo, pero utilizando operadores podemos crear nuevas columnas de la tabla resultado que sean una combinación de estas. Estos operadores también nos van a permitir realizar comparaciones que se pueden aplicar en where y otras combinaciones más (figura 7).

OPERADORES		
Aritméticos	+	Suma
	-	Resta
	*	Multipliación
	/	División
Comparación	=	Igual a
	>	Mayor que
	<	Menor que
	>=	Mayor igual
	<=	Menos igual
	!= o <>	Distinto de
Lógicos	AND	Devuelve valor verdadero cuando las dos premisas son ciertas
	OR	Devuelve valor verdadero cuando una de las premisas es cierta.
	NOT	Devuelve valor verdadero si el resultado es falso y viceversa.

Figura 7. Operadores en SQL

Con ellos, podemos crear consultas más complejas (figura 8).

```
select *
from notas
where Notal=7 and (notal+nota2+nota3)/3 > 6;
```

	CodAlu	NombreAlu	Notal	Nota2	Nota3
▶	6	Benito Martín, Luis	7	6	8
*	NULL	NULL	NULL	NULL	NULL

Figura 8. Consulta con operadores

- **LIKE** si bien con el operador = podemos realizar comparaciones entre cadenas de caracteres, el operador like nos va a permitir realizar esta misma consulta y usar ciertos caracteres especiales que dan mucha más flexibilidad (figura 9).

<div>LIKE</div> <div> <div>%</div> <div>Representa cualquier valor entre 0 a n caracteres.</div> </div> <div> <div>_</div> <div>Representa cualquier valor que ocupe una posición.</div> </div>	Like 'DIRECTOR'
	Representa el texto «DIRECTOR».
	LIKE 'M%'
	Representa cualquier cadena que empiece por «M».
	LIKE '%M%'
	Representa cualquier cadena que contenga una «M».
	LIKE '_M'
	Representa cualquier cadena de dos posiciones que termine en M.
	LIKE 'N__'
	Representa cualquier cadena de tres posiciones que empiece por N.
	LIKE '_R%'
	Representa cualquier cadena donde la 2.ª posición sea una R.

Figura 9. Uso del operador LIKE

Veamos a continuación un uso práctico del operador like (figura 10).

```

17 • select *
18   from emple
19   where EmpApellido like 'J%';
20

```

EmpNo	EmpApellido	Oficio	EmpleDir	FechaAlta	Salario
7566	JIMÉNEZ	DIRECTOR	7839	1981-02-02	3800
7900	JIMENO	EMPLEADO	7698	2005-08-08	1200

Figura 10. Ejemplo de consulta empleando LIKE

- **NULL y NOT NULL** con estos operadores vamos a comprobar si los campos están vacíos o no lo están. Cuando un campo no tiene contenido puede que signifique algo importante: si no tiene fecha de devolución, será que todavía no se ha devuelto; si no tienen nota, será que no se ha presentado, etc.

```

select *
from notas
where nota1 is null or nota2 is null or nota3 is null;

```

CodAlu	NombreAlu	Nota1	Nota2	Nota3
8	Corregidor Sánchez, Ana	NULL	9	8
9	Díaz Sánchez, María	NULL	NULL	NULL
10	Fernández Peña, Luis	4	5	NULL
NULL	NULL	NULL	NULL	NULL

Figura 11. Uso de NULL