

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 Squel en un proyecto llamado System R. SQL (structured query language), viene de Squel. 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]
{colum1, colum2,...,column|*}

FROM nombreTabla1 [,nombreTabla2,...]
[WHERE Condicion]
[ORDER BY colum [DESC|ASC] [,colum [DESC|ASC]]
[GROUP BY colum]
[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).



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.



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.

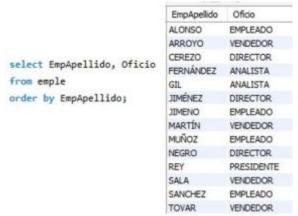


Figura 3. Consulta con resultados en orden ascendente



 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).

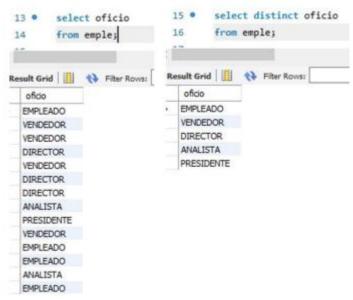


Figura 4. Uso de all y distinct

 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).

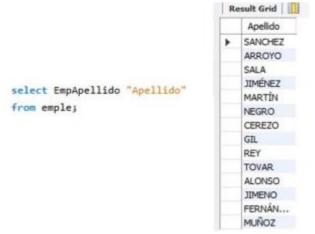


Figura 5. Utilizando un alias en la consulta

 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).



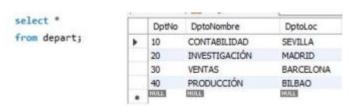


Figura 6. Uso de * en consultas

 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).

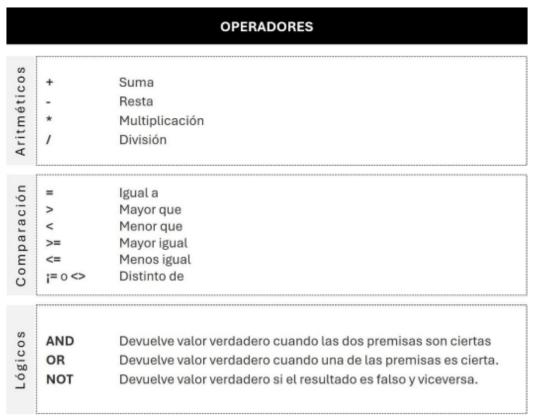


Figura 7. Operadores en SQL

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

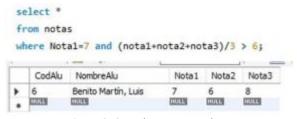


Figura 8. Consulta con operadores



• **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).

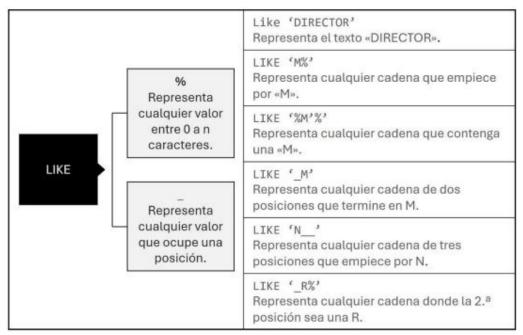


Figura 9. Uso del operador LIKE

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



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.



Figura 11. Uso de NULL