
Referencia SQL: Datos de las tablas

En esta sección se presenta una referencia de las sentencias de SQL para crear, modificar y eliminar una tabla de la base de datos.

Consultas de una tabla

La sentencia SELECT se utiliza para seleccionar datos desde la base de datos. Los datos devueltos son almacenados en una tabla resultado, llamada conjunto resultado.

La sintaxis básica es:

```
SELECT column1, column2, ...  
FROM table_name;
```

Donde *column1*, *column2*, etc. son las columnas de la tabla que queremos seleccionar. Si queremos seleccionar todas las columnas de la tabla podemos utilizar asterisco (*). Por ejemplo:

```
SELECT * FROM libro;
```

La sentencia SELECT DISTINCT se utiliza para devolver solo los valores diferentes. En una tabla, muchas veces una columna contiene muchos valores duplicados, y a veces solo queremos una lista de los valores diferentes.

La sintaxis básica es:

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

Por ejemplo, podemos pedir las categorías de los libros que tenemos registrados:

```
SELECT DISTINCT categoria_id FROM libro;
```

La cláusula WHERE se utiliza para filtrar registros, es decir, se utiliza para extraer solo aquellos registros que cumplen con una condición específica.

La sintaxis básica es:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

Donde *condition* es una expresión que debe evaluar a verdadero o falso. Por ejemplo, si quiero los libros con precio mayor a 100 puedo realizar la siguiente consulta:

```
SELECT * FROM libro where precio > 100;
```

Además, al ser una expresión *booleana*, se pueden combinar distintas expresiones con los operadores lógicos AND, OR y NOT. Con lo cual, podría escribir expresiones más complejas como, por ejemplo:

```
-- libros con precio entre 90 y 200  
SELECT * FROM libro where precio >= 90 AND precio <= 200;  
  
-- libros con precios menores a 100 y libros con precios mayores a 200  
SELECT * FROM libro where precio < 100 OR precio > 200;  
  
-- libros que no son de la categoría con id igual a 1  
SELECT * FROM libro where NOT categoria_id = 1;
```

Por otro lado, también podemos ordenar los registros resultados. Para esto utilizamos la palabra clave ORDER BY que ordena el conjunto resultado de manera ascendente o descendente según se indique. Por defecto, los registros se ordenan en orden ascendente.

La sintaxis básica es:

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

Cuando se ordena por varias columnas, primero se ordena por la columna que aparece primero (más a la izquierda), luego por la segunda, y así siguiendo. Y el orden será ascendente si no se indica nada o si se indica ASC y de modo descendente si se indica DESC.

Por ejemplo, ordenamos los libros por precio con orden ascendente en primera consulta y por orden descendente en la segunda consulta.

```
SELECT * FROM libro ORDER BY precio;  
  
SELECT * FROM libro ORDER BY precio DESC;
```

Insertión de datos en una tabla

La sentencia INSERT INTO se utiliza para insertar un nuevo registro en una tabla. Hay dos formas de escribirlo. La primera forma, especifica los nombres de las columnas y los valores a insertar en cada columna.

La sintaxis es:

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

Por ejemplo, agregamos la categoría de libros con nombre “Ciencia Ficción”:

```
INSERT INTO categoria_libro (NOMBRE) VALUES ('Ciencia Ficción');
```

Notar que no hace falta indicar la columna CODIGO porque es autoincremental.

Si se insertan valores para todas las columnas de la tabla, entonces no es necesario especificar los nombres de las columnas en la consulta SQL. Sin embargo, hay que asegurarse que el orden de los valores sean el mismo que el orden de las columnas en la tabla.

La sintaxis para este caso queda:

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

Por ejemplo, creamos la categoría de libros con nombre “Terror” y CODIGO 4:

```
INSERT INTO categoria_libro VALUES (4, 'Terror');
```

Actualización de datos en una tabla

La sentencia UPDATE se utiliza para modificar un registro existente en una tabla.

La sintaxis es:

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

Donde se actualizan de la tabla *table_name* las columnas *column1*, *column2*, etc. con los valores *value1*, *value2*, etc. respectivamente para los registros que cumplen con la condición *condition*.

Por ejemplo, modificamos el nombre de la categoría de libros con código 4, asignándole como nuevo nombre “Acción”:

```
UPDATE categoria_libro SET NOMBRE = 'Acción' WHERE CODIGO = 4;
```

Borrado de datos de una tabla

La sentencia DELETE se utiliza para borrar registros existentes en una tabla.

La sintaxis básica es:

```
DELETE FROM table_name WHERE condition;
```

Con lo cual, se borran todos los registros de la tabla *table_name* que cumplan con la condición *condition*.

Por ejemplo, si queremos borrar la categoría de libros con CODIGO 4, entonces ejecutamos la siguiente consulta:

```
DELETE FROM categoria_libro WHERE CODIGO = 4;
```

Además, es posible borrar todos los registros en una tabla sin borrar la tabla. Esto significa que la estructura de la tabla, las columnas y los índices quedan intactos. Por ejemplo, con la siguiente consulta se borran todas las categorías de los libros:

```
DELETE FROM categoria_libro;
```