

Apéndice A: Introducción al lenguaje SQL.

A.1. Comandos

A.2. Cláusulas

A.3. Operadores Lógicos.

A.4. Operadores de Comparación.

A.5. Funciones de Agregado

Apéndice A: Introducción al lenguaje SQL.

Hasta la década de los 80, las personas que preparaban las consultas e informes de una base de datos debían ser programadores. Al aparecer las bases de datos con lenguajes de consulta sencillos y estandarizados, semejantes al lenguaje natural, el proceso de consulta puede hacerlo cualquier usuario mediante un lenguaje escrito asequible.

El lenguaje de gestión de bases de datos más conocido en la actualidad es el SQL, *Structured Query Language*, que es un lenguaje estándar internacional, comúnmente aceptado por los fabricantes de generadores de bases de datos. En concreto, el BDE de C++ Builder es capaz de usar el lenguaje SQL.

El lenguaje SQL puede trabajar con estructura cliente/servidor sobre una red de ordenadores. El ordenador cliente es el que inicia la consulta; el ordenador servidor es que atiende esa consulta. El cliente utiliza toda su capacidad de proceso para trabajar; se limita a solicitar datos al ordenador servidor, sin depender para nada más del exterior. Estas peticiones y las respuestas son transferencias de textos que cada ordenador cliente se encarga de sacar por pantalla, presentar en informes tabulados, imprimir, guardar, etc., dejando el servidor libre.

SQL permite:

- Definir una base de datos mediante tablas
- Almacenar información en tablas.
- Seleccionar la información que sea necesaria de la base de datos.
- Realizar cambios en la información y estructura de los datos.
- Combinar y calcular datos para conseguir la información necesaria.

Como su nombre indica, el SQL nos **permite** realizar **consultas a la base de datos**. Pero, como vemos, el nombre se queda corto ya que SQL además realiza funciones de **definición, control y gestión de la base de datos**. Las sentencias SQL se

clasifican según su finalidad dando origen a tres ‘lenguajes’ o mejor dicho sublenguajes:

- α **DDL** (Data Description Language), **lenguaje de definición** de datos, incluye órdenes para definir, modificar o borrar las tablas en las que se almacenan los datos y de las relaciones entre estas. (Es el que más varía de un sistema a otro)
- α **DCL** (Data Control Language), **lenguaje de control** de datos, contiene elementos útiles para trabajar en un entorno multiusuario, en el que es importante la protección de los datos, la seguridad de las tablas y el establecimiento de restricciones en el acceso, así como elementos para coordinar la compartición de datos por parte de usuarios concurrentes, asegurando que no interfieren unos con otros.
- α **DML** (Data Manipulation Language), **lenguaje de manipulación** de datos, nos permite recuperar los datos almacenados en la base de datos y también incluye órdenes para permitir al usuario actualizar la base de datos añadiendo nuevos datos, suprimiendo datos antiguos o modificando datos previamente almacenados.

Una sentencia SQL es como una **frase** (escrita en **inglés**) con la que decimos **lo que queremos obtener y de donde obtenerlo**.

Todas las sentencias empiezan con un **verbo** (palabra reservada que indica la acción a realizar), seguido del resto de **cláusulas**, algunas **obligatorias** y otras **opcionales** que completan la frase. Todas las sentencias siguen una **sintaxis** para que se puedan ejecutar correctamente, para describir esa sintaxis utilizaremos un **diagrama sintáctico** como el que se muestra a continuación.

Cómo interpretar un diagrama sintáctico

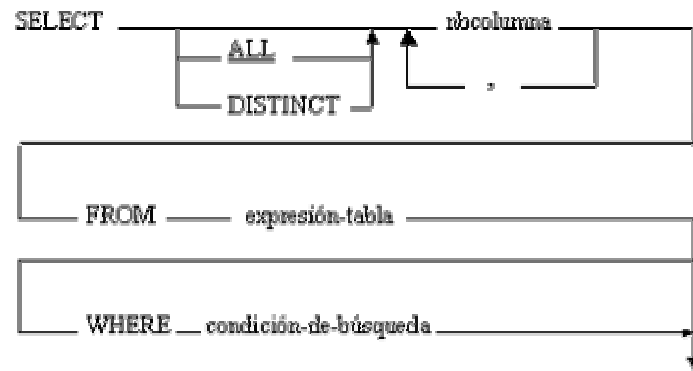


Fig. A.1: Interpretación de una sentencia SQL.

Las palabras que aparecen en mayúsculas son palabras reservadas se tienen que poner tal cual y no se pueden utilizar para otro fin, por ejemplo, en el diagrama de la figura tenemos las palabras reservadas **SELECT**, **ALL**, **DISTINCT**, **FROM**, **WHERE**.

Las palabras en minúsculas son variables que el usuario deberá sustituir por un dato concreto. En el diagrama tenemos nbcolumna, expresion-tabla y condicion-de-busqueda.

Una sentencia válida se construye siguiendo la línea a través del diagrama hasta el punto que marca el final. Las líneas se siguen de **izquierda a derecha y de arriba abajo**. Cuando se quiere alterar el orden normal se indica con una **flecha**.

¿Cómo se interpretaría el diagrama sintáctico de la figura?

Hay que empezar por la palabra **SELECT**, después puedes poner **ALL** o bien **DISTINCT** o nada, a continuación un nombre de columna, o varios separados por comas, a continuación la palabra **FROM** y una expresión-tabla, y por último de forma opcional puedes incluir la cláusula **WHERE** con una condición-de-búsqueda.

Por ejemplo:

SELECT ALL col1,col2,col3 FROM mitabla

SELECT col1,col2,col3 FROM mitabla

SELECT DISTINCT col1 FROM mitabla

SELECT col1,col2 FROM mitabla WHERE col2 = 0

Todas estas sentencias se podrían escribir y no darían lugar a errores sintácticos. Cuando una palabra opcional está **subrayada**, esto indica que ese es el **valor por defecto** (el valor que se asume si no se pone nada). En el ejemplo anterior las dos primeras sentencias son equivalentes (en el diagrama ALL aparece subrayada)

Otra forma de analizar una sentencia SQL es como sigue: está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

A.1. Comandos

Existen dos tipos de comandos SQL:

- α los DDL que permiten crear y definir nuevas bases de datos, campos e índices.
- α los DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Comandos DDL	
Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos e índices
DROP	Empleado para eliminar tablas e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

Comandos DML	
Comando	Descripción
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.
UPDATE	Utilizado para modificar los valores de los campos y registros especificados
DELETE	Utilizado para eliminar registros de una tabla de una base de datos

A.2. Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Cláusula	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

A.3. Operadores Lógicos.

Operador	Uso
AND	Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
OR	Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.

A.4. Operadores de Comparación.

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor ó Igual que
>=	Mayor ó Igual que
=	Igual que

BETWEEN	Utilizado para especificar un intervalo de valores.
LIKE	Utilizado en la comparación de un modelo
In	Utilizado para especificar registros de una base de datos

A.5. Funciones de Agregado

Las funciones de agregado se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

Función	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado