

Laboratorio de Informática



MANUAL BÁSICO DEL LENGUAJE SQL

ESCUELA COLOMBIANA DE INGENIERÍA

JULIO GARAVITO

LABORATORIO DE INFORMÁTICA

BOGOTÁ D. C.

2007-2

TABLA DE CONTENIDO

INTRODUCCIÓN	3
1. COMANDOS	4
1.1 Comandos DLL	4
1.2 Comandos DML	4
2. CLÁUSULAS	5
2.1 Comando Descripción	5
3. OPERADORES	6
3.1 Operadores Lógicos	6
3.2 Operadores de Comparación	6
4. FUNCIONES DE AGREGADO	7
5. CONSULTAS	8
5.1 Consultas de Selección	8
5.1.1 Consultas básicas	8
5.1.2 Ordenar los registros	8
5.2 Consultas con Predicado	8
6. ALIAS	9
7. RECUPERAR INFORMACIÓN DE UNA BASE DE DATOS EXTERNA	10
7.1 Criterios de Selección	10
8. TIPOS DE DATOS	11
9. ESTRUCTURAS DE LAS TABLAS	12
9.1 Creación de Tablas Nuevas	12
10. TIPO DE ÍNDICE	13

INTRODUCCIÓN

Este manual básico del lenguaje SQL está diseñado para toda aquella persona que esté interesada en un aprendizaje de los comandos básicos del lenguaje SQL.

A continuación se presentan lo que son comandos DDL y DML, las cláusulas, los operadores (lógicos y de comparación), funciones de agregado, consultas, tipos de datos, estructuras de las tablas y tipos de índices.

1. COMANDOS

Existen dos tipos de comandos **SQL**:

- **DLL**: permiten crear y definir nuevas bases de datos, campos e índices.
- **DML**: permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

1.1 Comandos DLL

- **CREATE** Crea nuevas tablas, campos e índices
- **DROP** Elimina tablas e índices
- **ALTER** Modifica las tablas agregando campos o cambiando la definición de los campos.

1.2 Comandos DML

- **SELECT** Consulta registros de la base de datos que satisfagan un criterio determinado
- **INSERT** Carga lotes de datos en la base de datos en una única operación.
- **UPDATE** Modifica los valores de los campos y registros especificados
- **DELETE** Elimina registros de una tabla de una base de datos

2. CLÁUSULAS

Son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

2.1 Comando Descripción

- **FROM** Especifica la tabla de la cual se van a seleccionar los registros.
- **WHERE** Especifica las condiciones que deben reunir los registros que se van a seleccionar.
- **GROUP BY** Separa los registros seleccionados en grupos específicos.
- **HAVING** Expresa la condición que debe satisfacer cada grupo.
- **ORDER BY** Ordena los registros seleccionados de acuerdo con un orden específico.

3. OPERADORES

3.1 Operadores Lógicos

- **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.

3.2 Operadores de Comparación

- **<** Menor que
- **>** Mayor que
- **< >** Distinto de
- **<=** Menor ó Igual que
- **>=** Mayor ó Igual que
- **BETWEEN** Especifica un intervalo de valores.
- **LIKE** Comparación de un modelo
- **In** Especifica registros de una base de datos

4. 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.

- **AVG** Calcula el promedio de los valores de un campo determinado
- **COUNT** Devuelve el número de registros de la selección
- **SUM** Devuelve la suma de todos los valores de un campo determinado
- **MAX** Devuelve el valor más alto de un campo especificado
- **MIN** Devuelve el valor más bajo de un campo especificado

5. CONSULTAS

5.1 Consultas de Selección

Las consultas de selección se utilizan para indicar al motor de datos que devuelva información de las bases de datos, esta información es devuelta en forma de conjunto de registros que se pueden almacenar en un objeto recordset. Este conjunto de registros es modificable.

5.1.1 Consultas básicas

La sintaxis básica de una consulta de selección es la siguiente:

```
SELECT Campos FROM Tabla;
```

En donde campos es la lista de campos que se deseen recuperar y tabla es el origen de los mismos, por ejemplo:

```
SELECT Nombre, Teléfono FROM Clientes;
```

Esta consulta devuelve un recordset con el campo nombre y teléfono de la tabla clientes.

5.1.2 Ordenar los registros

Se puede especificar el orden en que se desean recuperar los registros de las tablas mediante la cláusula ORDER BY Lista de Campos. En donde Lista de campos representa los campos a ordenar.

5.2 Consultas con Predicado

El predicado se incluye entre la cláusula y el primer nombre del campo a recuperar, los posibles predicados son:

- **ALL** Devuelve todos los campos de la tabla
- **TOP** Devuelve un determinado número de registros de la tabla
- **DISTINCT** Omite los registros cuyos campos seleccionados coincidan totalmente
- **DISTINCROW** Omite los registros duplicados basándose en la totalidad del registro y no sólo en los campos seleccionados.

6. ALIAS

En determinadas circunstancias es necesario asignar un nombre a alguna columna determinada de un conjunto devuelto.

Para ello se tiene la palabra reservada **AS** que se encarga de asignar el nombre que deseamos a la columna deseada. Podemos hacer que la columna devuelta por la consulta, en lugar de llamarse apellido (igual que el campo devuelto) se llame Empleado.

En este caso procederíamos de la siguiente forma:

```
SELECT DISTINCTROW Apellido AS Empleado FROM Empleados;
```

7. RECUPERAR INFORMACIÓN DE UNA BASE DE DATOS EXTERNA

En ocasiones es necesario la recuperación de información que se encuentra contenida en una tabla que no se encuentra en la base de datos que ejecutará la consulta o que en ese momento no se encuentra abierta, esta situación la podemos salvar con la palabra reservada IN de la siguiente forma:

```
SELECT DISTINCTROW Apellido AS Empleado FROM Empleados  
IN 'c:\databases\gestion.mdb';
```

En donde c:\databases\gestion.mdb es la base de datos que contiene la tabla Empleados.

7.1 Criterios de Selección

Los criterios de selección o las posibilidades de filtrar los registros se realizan con el fin de recuperar solamente aquellos que cumplan unas condiciones preestablecidas.

8. TIPOS DE DATOS

Los tipos de datos SQL se clasifican en 13 tipos de datos primarios y de varios sinónimos válidos reconocidos por dichos tipos de datos. Tipos de datos primarios:

- **BINARY** 1 byte Para consultas sobre tabla adjunta de productos de bases de datos que definen un tipo de datos Binario.
- **BIT** 1 byte Valores Si/No ó True/False
- **BYTE** 1 byte Un valor entero entre 0 y 255.
- **COUNTER** 4 bytes Un número incrementado automáticamente (de tipo Long)
- **CURRENCY** 8 bytes Un entero escalable entre 922.337.203.685.477,5808 y 922.337.203.685.477,5807.
- **DATETIME** 8 bytes Un valor de fecha u hora entre los años 100 y 9999.
- **SINGLE** 4 bytes Un valor en punto flotante de precisión simple con un rango de -3.402823×10^{38} a $-1.401298 \times 10^{-45}$ para valores negativos, 1.401298×10^{-45} a 3.402823×10^{38} para valores positivos, y 0.
- **DOUBLE** 8 bytes Un valor en punto flotante de doble precisión con un rango de $-1.79769313486232 \times 10^{308}$ a $-4.94065645841247 \times 10^{-324}$ para valores negativos, $4.94065645841247 \times 10^{-324}$ a $1.79769313486232 \times 10^{308}$ para valores positivos, y 0.
- **SHORT** 2 bytes Un entero corto entre -32,768 y 32,767.
- **LONG** 4 bytes Un entero largo entre - 2,147,483,648 y 2,147,483,647.
- **LONGTEXT** 1 byte por carácter. De cero a un máximo de 1.2 gigabytes.
- **LONGBYNARY** Según se necesite. De cero 1 gigabyte. Utilizado para objetos OLE.
- **TEXT** 1 byte por carácter. De cero a 255 caracteres.

9. ESTRUCTURAS DE LAS TABLAS

9.1 Creación de Tablas Nuevas

Si se está utilizando el motor de datos de Microsoft para acceder a bases de datos access, sólo se puede emplear esta instrucción para crear bases de datos propias de access.

Su sintaxis es:

```
[TABLE] consulta1 UNION [ALL] [TABLE] consulta2 [UNION [ALL] [TABLE]
consultan [ ... ]]
```

En donde:

- **tabla** Nombre de la tabla que se desea modificar.
- **campo** Nombre del campo que se va a añadir o eliminar.
- **tipo** Es el tipo de campo que se va a añadir.
- **tamaño** Tamaño del campo que se va a añadir (sólo para campos de texto).
- **índice** Nombre del índice del campo (cuando se crean campos) o el nombre del índice de la tabla que se desea eliminar.
- **Índice multicampo** Nombre del índice del campo multicampo (cuando se crean campos) o el nombre del índice de la tabla que se desea eliminar.

10. TIPO DE ÍNDICE

- **UNIQUE** Genera un índice de clave única. Lo que implica que los registros de la tabla no pueden contener el mismo valor en los campos indexados.
- **PRIMARY KEY** Genera un índice primario el campo o los campos especificados. Todos los campos de la clave principal deben ser únicos y no nulos, cada tabla sólo puede contener una única clave principal.
- **FOREIGN KEY** Genera un índice externo (toma como valor del índice campos contenidos en otras tablas). Si la clave principal de la tabla externa consta de más de un campo, se debe utilizar una definición de índice de múltiples campos, listando todos los campos de referencia, el nombre de la tabla externa, y los nombres de los campos referenciados en la tabla externa en el mismo orden que los campos de referencia listados.

Si los campos referenciados son la clave principal de la tabla externa, no tiene que especificar los campos referenciados, predeterminado por valor, el motor Jet se comporta como si la clave principal de la tabla externa fueran los campos referenciados .