ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

“RESUMEN DEL LENGUAJE SQL”

AUTORÍA
JOSEFA PÉREZ DOMINGUEZ

TEMÁTICA
INFORMATICA

ETAPA
CICLO FORMATIVO DE GRADO SUPERIOR Y MEDIO DE INFORMATICA

Resumen

Con esta publicación muestra un resumen de la sintaxis del lenguaje SQL, para que aquellas personas que trabajen con este lenguaje puedan ver de manera rápida la sintaxis de las principales sentencias.

Palabras clave

Lenguaje SQL, elementos del lenguaje, creación de tablas, actualización de tablas, consultas de tablas, subconsultas, consultas multitablas y consultas dentro de otras consultas

1. LENGUAJE SQL

SQL significa lenguaje estructurado de consultas (Structured Query Language).

Es un lenguaje relacional que opera sobre relaciones (tablas) y da como resultado otra relación.

Los principales sistemas gestores de base de datos relacionales incorporan un motor SQL en el servidor de base de datos, así como herramientas clientes que permiten enviar comandos SQL para que sean procesadas por el motor del servidor. Todas las tareas de gestión de la base de datos pueden realizarse utilizando sentencias SQL.

Con el lenguaje SQL podemos hacer:

- Consultar los datos de la base de datos.
- Insertar, modificar y borrar datos.
- Crear, modificar y borrar objetos de la base de datos.
- Controlar el acceso a la información.
- Garantizar la consistencia de los datos.

SQL es un lenguaje muy extendido que incorporan muchos lenguajes de programación para acceder a una base de datos.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

Vamos a explicar las consideraciones que debemos tener en cuenta, dentro en la sintaxis de las sentencias SQL que presentaremos a lo largo del artículo:

- Las palabras reservadas del lenguaje SQL aparecen en mayúsculas.
- Los nombres de los objetos (tablas, campos, etc) aparecen en formato TipoTitulo.
- { } indica elección obligatoria entre varios elementos.
- | separa los elementos de la elección.
- [ ] encierran un elemento opcional.
- ; es el separador de instrucciones.

2. ELEMENTOS DEL LENGUAJE.

Los elementos del lenguaje SQL serian:

- Identificadores: son los nombres que reciben los objetos de la base de datos y la propia base de datos.
- Palabras reservadas: son las palabras que tienen un significado especial para el gestor de la base de datos que no pueden ser utilizados como identificadores.

3. DATOS.

Los datos pueden ser de los siguientes tipos:

- Datos constantes o literales. Que pueden ser de tres tipos:
  - Constantes numéricas: cadena de dígitos que puede llevar un punto decimal y que puede estar precedidos por un signo + ó -.
  - Constantes de cadena: cadena de caracteres encerrada entre comillas simples.
  - Constante fecha.
- Datos variables. Se definen indicando su nombre (NombreColumna) y el tipo de datos que almacenaran. Vamos a indicar algunos de los tipos de datos mas utilizados:
  - Datos numéricos: INT[(num)] o INTEGER[(num)] para indicar un número entero, FLOAT(escala, precisión) de esta forma indicamos un número en coma flotante y NUMERIC(escala, precisión) se especifica el número de dígitos totales y el número de posiciones decimales.
  - Datos alfanuméricos o cadenas de caracteres: CHAR(long) guarda una cadena de caracteres de longitud fija, VARCHAR(long) almacena una cadena de caracteres de longitud variable,

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

TEXT este tipo de datos se corresponde con un texto de 65535 caracteres y LONGTEXT que es un texto de longitud máxima de 4 Gigas.

- Fecha: DATE con el formato 'YYYY-MM-DD', DATETIME con el siguiente formato 'YYYY-MM-DD HH-MM-SS' y TIME con la forma 'HH:MM:SS'.
- Binarios: BOOLEAN que se corresponde 1 a verdadero y 0 a falso.

4. OPERADORES.

Los operadores serian de los siguientes tipos:

- Operadores aritméticos: +, -, *, /, Div (división entera)
- Operadores de comparación: =, != ó <> (distinto), <, >, <= menor o igual, >= mayor o igual, BETWEEN valor1 AND valor2, su valor será verdadero si el valor a comparar es mayor o igual que valor1 y menor o igual que valor2 y falso en caso contrario, IN (lista de valores separados por coma) da como resultado verdadero si el valor a comparar esta dentro de la lista de valores y falso en caso contrario, IS NULL se obtiene como resultado verdadero si el valor del dato comparado es nulo y falso en caso contrario, y por ultimo el operador LIKE que nos permite comparar dos cadenas.
- Operadores lógicos: NOT, AND, OR y XOR.
- Prioridad de los operadores. Nos indica que operación se realiza primero cuando en una expresión hay varios operadores. Lo vamos a representar a través de una tabla:

Prioridad	Operador
1º	*, /, DIV
2º	+, -
3ª	=, !=, <, >, <=, >=, IS, LIKE, BETWEEN, IN
4º	NOT
5º	AND
6º	OR, XOR

5. FUNCIONES PREDEFINIDAS.

Son funciones que incorpora el gestor y que dan potencia al lenguaje. Existen funciones predefinidas para operar con todo tipo de datos.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

6. VALORES NULOS (NULL).

La ausencia de valor se expresa con el valor nulo (NULL) que no es igual al valor 0 o a una cadena vacía.

Habrá que tener espacial cuidado a la hora de operar con columnas de una tabla que pueden contener valores nulos.

7. EXPRESIONES Y CONDICIONES.

Es un conjunto de variables, constantes o literales, funciones, operadores y paréntesis (se utilizan para variar la prioridad de los operadores):

8. RESTRICCIONES EN LAS TABLAS.

Las restricciones nos permiten relacionar las tablas entre si y poner restricciones a los valores que pueden tomar los atributos de esa tablas.

Tipos de restricciones:

- NOT NULL. Obliga a la existencia de un valor.
- DEFAULT. Proporciona un valor por defecto si en la inserción no se ha especificado un valor para ese atributo.
- PRIMARY KEY. Indica que una o varias columnas identifican de manera unívoca cada fila de una tabla.
- FOREIGN KEY. Es una columna o conjunto de columnas que tiene un valor que hace referencia a una fila de otra tabla.
- UNIQUE. Indica que esta columna o conjunto de columnas debe tener un valor único, este valor se comprobará al hacer una inserción.
- CHECK. Comprueba que el valor introducido para una determinada columna cumple una condición.

9. INTEGRIDAD REFERENCIAL.

Las claves ajenas nos permiten mantener la integridad referencial en la base de datos relacional. La columna o columnas que hemos definido como clave ajena deben tomar valores que se correspondan con un valor existente de la clave referenciada. ¿Qué sucede si borramos o modificamos el valor de la clave primaria referenciada? El sistema debe de impedir realizar acciones que dejarían la base de datos sin integridad referencial.

Será necesario hacer estas operaciones, para mantener la integridad de los datos al borrar (DELETE) o modificar (UPDATE) una fila referenciada, por eso cuando definimos una clave ajena debemos seleccionar alguna de estas opciones:

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

- RESTRICT. Es la opción por defecto, no se nos permite llevar a cabo el borrado o la modificación.
- CASCADE. El borrado o modificación de una fila de la tabla referenciada lleva consigo el borrado o modificación en cascada de las filas de la tabla que contiene la clave ajena.
- SET NULL. El borrado o modificación de una fila de la tabla referenciada lleva consigo poner NULL en los valores de las claves ajenas de las filas de la tabla referenciada.
- SET DEFAULT. Pone por defecto un valor en las claves ajenas de la tabla referenciada.
- NO ACTION. El borrado o modificación de una fila de la tabla referenciada solo se produce si no existe ese valor en la tabla que contiene la clave ajena.

10. CREACION DE TABLAS.

La sintaxis sería:

CREATE TABLE [IF NOT EXISTS] NombreTabla

(NombreColumna TipoDato [Restricción1]

[, NombreColumna TipoDato [Restricción1…..]]

[, Restricción2 [, Restricción2, ..]);

NombreTabla es el identificador que identifica a la tabla que se va a crear.

NombreColumna es el identificador utilizado para darle nombre a la columna de la tabla.

TipoDato indicamos el tipo de datos que se va a almacenar en la columna.

La Restricción1 es la definición de la restricción a nivel de columna. Se define de la siguiente forma:

- CLAVE PRIMARIA — PRIMARY KEY
- POSIBILIDAD DE NULO — NULL | NOT NULL
- VALOR POR DEFECTO — DEFAULT ValorDefecto
- UNICIDAD — UNIQUE
- COMPROBACION DE VALORES — CHECK (Expresión)
- CLAVE AJENA — REFERENCES NombreTabla [(NombreColumna)]
- AUTO INCREMENTO — AUTO_INCREMENT

Por el contrario, la Restriccion2 se define a nivel de tabla. La forma de definir estas restricciones a nivel de tabla sería:

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

- PRIMARY KEY
  [CONSTRAINT [NombreConstraint]] PRIMARY KEY (NombreColumna [,NombreColumna …])
- UNIQUE
  [CONSTRAINT [NombreConstraint]] UNIQUE (NombreColumna [,NombreColumna …])
- CHECK
  [CONSTRAINT [NombreConstraint]] CHECK (Expresión)
- FOREIGN KEY
  [CONSTRAINT [NombreConstraint]]
  FOREIGN KEY (NombreColumna [,NombreColumna …])
  REFERENCES NombreTabla [(NombreColumna [, NombreColumna ….])]
  [ON DELETE {CASCADE | SET NULL| NO ACTION |SET DEFAULT |RESTRICT}]
  [ON UPDATE {CASCADE | SET NULL| NO ACTION |SET DEFAULT |RESTRICT}]

11. MODIFICACIÓN DE LA DEFINICION DE UNA TABLA.

Una vez creada una tabla podemos tener la necesidad de modificarla.

La sintaxis sería:

ALTER TABLE NombreTabla

EspecificaciónModificación [, EspecificaciónModificación …];

NombreTabla indicamos el nombre de la tabla que deseamos modificar.

EspecificaciónModificacion especificamos las modificaciones que deseamos realizar en la tabla, estas serían:

- Añadir una columna.
  ADD [COLUMN] NombreColumna TipoDato [Restriccion1]
- Añadir una restricción.
  ADD [CONSTRAINT [NombreConstraint]]
  PRIMARY KEY (NombreColumna [, NombreColumna …])
  ADD [CONSTRAINT [NombreConstraint]]
  FOREIGN KEY (NombreColumna [, NombreColumna …])

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

ADD [CONSTRAINT [NombreConstraint]]
UNIQUE (NombreColumna [, NombreColumna …])

- Borrar una columna.
  DROP [COLUMN] NombreColumna
- Borrar una restricción.
  DROP PRIMARY KEY
  DROP FOREIGN KEY NombreConstraint
- Modificar una columna sin cambiar el nombre.
  MODIFY [COLUMN] NombreColumna TipoDato [Restriccion1]
- Modificar la definición de una columna y su nombre.
  CHANGE [COLUMN] NombreColumnaAntiguo NombreColumnaNuevo TipoDatos [Restriccion1]
- Renombrar una tabla.
  RENAME [TO] NombreTablaNuevo

12. ELIMINAR UNA TABLA.

La sintaxis de esta orden sería:
DROP TABLE [IF EXISTS] NombreTabla;

IF EXISTS previene de errores si la tabla a borrar no existe.

13. ACTUALIZACIONES DE LAS TABLAS.

Cuando hablamos de actualización nos estamos refiriendo a las siguientes operaciones:

- Insertar una nueva fila en la base de datos. La sintaxis sería:
  INSERT INTO NombreTabla [(NombreColumna [,NombreColumna …])]
  VALUES (Expresión [, Expresión …]);

  Donde:
  NombreTabla es el nombre de la tabla donde vamos a insertar una fila.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

NombreColumna, especificaremos las columnas que queremos insertar.

Expresión indica las expresiones cuyos valores se van a insertar, existiendo una correspondencia con la lista de columnas.

- Modificación de filas. Con esta orden se modifica alguno de los datos de las filas existentes en una tabla. La sintaxis de esta sentencia será:

  UPDATE NombreTabla

  SET NombreColumna=Expresión [, NombreColumna=Expresión …]

  [WHERE Condición];

  Donde:

  NombreTabla indica la tabla donde se encuentran las filas que deseamos modificar.

  NombreColumna indica el nombre de la columna cuyo valor deseamos modificar.

  Expresión, el resultado de esta expresión será el nuevo valor de la columna.

  Condición, especificaremos cual es la condición que debe cumplir las filas que deseamos modificar.

- Eliminación de filas. Nos permite borrar una o mas filas, la sintaxis de la sentencia sería la siguiente:

  DELETE FROM NombreTabla

  [WHERE Condición] ;

  Donde:

  NombreTabla especifica el nombre de la tabla de la cual se van a borrar las filas.

  Condición, especifica la condición que deben cumplir las filas a borrar.

14. CONSULTAS DE DATOS.

Las consultas consisten en recuperar datos almacenados en una base de datos, para ello se utiliza la sentencia SELECT.

14.1. Consulta de selección.

SELECT {* | [ALL/DISTINCT] ExpresionColumna [AliasColumna]

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

[, ExpresionColumna [AliasColumna] …] }

FROM NombreTabla [AliasTabla]

[WHERE CondicionSeleccion]

[GROUP BY ExpresiónColumnaAgrupacion | PosiciónAgrupacion [, ExpresiónColumnaAgrupacion | PosiciónAgrupacion …]

[HAVING CondicionSeleccionGrupos]]

[ORDER BY {ExpresionColumnaOrdenacion | PosiciónOrdenacion} [ASC |DESC]

[, {ExpresionColumnaOrdenacion | PosiciónOrdenacion} [ASC |DESC] ….]]

[LIMIT [M,] N];

Donde:

ALL obtiene los elementos seleccionados, aunque sean repetidos.

DISTINT obtiene los elementos seleccionados, eliminando los repetidos.

ExpresionColumna es un conjunto de nombres de columnas, literales o constantes, operadores, funciones y paréntesis.

AliasColumna se permite asignar otro nombre a la misma tabla dentro de la misma consulta.

AliasTabla su finalidad es implicar el nombre original de tabla.

CondicionSeleccion es una expresión cuyo resultado es VERDADERO/FALSO/NULO. Para cada fila se evalúa la condición de selección y si el resultado es VERDADERO se visualizan las expresiones indicadas.

GROUP BY permite agrupar las filas de una tabla, formando grupos según el contenido de algunas expresiones, y obtener salidas calculadas a partir de los grupos formados.

ExpresiónColumnaAgrupacion es una expresión que contiene columnas de la tabla, literales, operadores y/o funciones por la que se formarán los grupos.

PosiciónAgrupacion posición que ocupa la expresión por la que queremos agrupar en la lista de expresiones visualizadas.

CondicionSeleccionGrupos es una condición o expresión que devuelve el valor verdadero/falso/nulo, para seleccionar grupos.

ORDER BY se utiliza para obtener el resultado de la consulta clasificada por algún criterio.

ASC |DESC indica la forma de ordenación ascendente o descendente.

ExpresiónColumnaOrdenacion conjunto de nombres de columnas con literales, operadores y/o funciones.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

PosiciónOrdenacion se corresponde con el número de orden que ocupa en la lista las expresiones visualizadas en el SELECT.

LIMIT nos permite limitar el número de filas que se visualizan como resultado de la sentencia SELECT.

M es el número de fila por el que de comienza la visualización. Las filas se empiezan a numerar en el 0.

N indica el número de filas que se quieren visualizar.

14.2. Subconsultas.

Consiste en utilizar los resultados de una consulta dentro de otra.

El formato de las subconsultas sería:

(SELECT [ALL/DISTICT] ExpresionColumna [,ExpresionColumna …]

FROM NombreTabla [, NombreTabla]

[WHERE CondicionSelección]

[GROUP BY ExpresionColumnaAgrupacion [, ExpresionColumnaAgrupacion …]

[HAVING CondicionSeleccionGrupos]]);

En la siguiente tabla se muestra la relación entre el retorno de la subconsulta y el operador de comparación que hay que utilizar:

Retorno de la subconsulta	Operador comparativo
Valor simple	De tipo aritmético (=, <>, <, >, <=, >=)
Mas de un valor	De tipo lógico (IN, ANY, ALL, EXISTS)

14.3. Consultas multitablas.

Hasta ahora, las sentencias que hemos visto se basaban en una única tabla, a menudo es necesario utilizar datos procedentes de dos o mas tablas de una base de datos. La sintaxis de esta sentencia sería:

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

SELECT {* | [ALL/DISTINCT] ExpresionColumna [AliasColumna]

[, ExpresionColumna [AliasColumna] …] }

FROM NombreTabla [AliasTabla] [, NombreTabla [AliasTabla]

[WHERE CondicionSeleccion |CondicionComposicion }]

[GROUP BY ExpresiónColumnaAgrupacion | PosiciónAgrupacion [, ExpresiónColumnaAgrupacion | PosiciónAgrupacion …]

[HAVING CondicionSeleccionGrupos]]

[ORDER BY {ExpresionColumnaOrdenacion | PosiciónOrdenacion} [ASC |DESC]

[, {ExpresionColumnaOrdenacion | PosiciónOrdenacion} [ASC |DESC] ….]]

[LIMIT [m,] n];

14.4. Consultas dentro de otras instrucciones.

Vamos a ver como se crea una tabla utilizando la definición de otra ya creada:

CREATE TABLE [IF NOT EXISTS] NombreTabla

[(DefinicionColumna [, DefinicionColumna …])]

SentenciaSelect;

Donde:

NombreTabla sería el nombre de la nueva tabla que se va a crear.

SentenciaSelect es una sentencia de selección.

Podemos insertar en una tabla el resultado de una consulta sobre otra tabla, la sintaxis sería:

INSERT INTO NombreTabla [(NombreColumna [, NombreColumna …])]

SELECT FormatoSelect ;

También se pueden modificar las filas obtenidas como resultado de una consulta, la sintaxis es la siguiente:

UPDATE NombreTabla

SET NombreColumna=Expresión [, NombreColumna=Expresión …]

WHERE NombreColumna operador (SentenciaSelect);

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 35 OCTUBRE DE 2010

Y por último, en ocasiones puede interesar eliminar las filas obtenidas como resultado de una subconsulta de la siguiente manera:

DELETE FROM NombreTabla

WHERE NombreColumna operador (SentenciaSelect) ;

Autoría

- Nombre y Apellidos: Josefa Pérez Domínguez
- Centro, localidad, provincia: I.E.S La Arboleda, Lepe (Huelva)
- E-mail: mjperez_2001@hotmail.com