Material Imprimible

Curso de SQL Server

Módulo 3: Manipulación de datos

Delete

A veces podemos desear deshacernos de los registros de una tabla. Para ello, utilizamos el comando DELETE FROM. La sintaxis para esto es:

```
DELETE FROM "nombre_tabla"
WHERE "condición";
```

Es más fácil utilizar un ejemplo. Por ejemplo, digamos que actualmente tenemos la siguiente tabla:

Tabla *Store_Information*

Store_Name	Sales	Txn_Date
Los Angeles	1500	05-Jan-1999
San Diego	250	07-Jan-1999
Los Angeles	300	08-Jan-1999
Boston	700	08-Jan-1999

y decidimos no mantener ninguna información sobre Los Ángeles en esta tabla. Para lograrlo, ingresamos el siguiente SQL:

```
DELETE FROM Store_Information
WHERE Store_Name = 'Los Angeles';
```

Ahora el contenido de la tabla se vería:

Tabla *Store_Information*

Store_Name	Sales	Txn_Date
San Diego	250	07-Jan-1999
Boston	700	08-Jan-1999

Update

La sentencia UPDATE se utiliza para modificar valores en una tabla.

La sintaxis de SQL UPDATE es:

```
UPDATE nombre_tabla
SET columna1 = valor1, columna2 = valor2
WHERE columna3 = valor3
```

La cláusula SET establece los nuevos valores para las columnas indicadas.

La cláusula WHERE sirve para seleccionar las filas que queremos modificar.

Ojo: Si omitimos la cláusula WHERE, por defecto, modificará los valores en todas las filas de la tabla.

Ejemplo del uso de SQL UPDATE

nombre	**apellido1**	**apellido2**
ANTONIO	PEREZ	GOMEZ
LUIS	LOPEZ	PEREZ
ANTONIO	GARCIA	BENITO
PEDRO	RUIZ	GONZALEZ

Si queremos cambiar el apellido2 'BENITO' por 'RODRIGUEZ' ejecutaremos:

```
UPDATE personas
SET apellido2 = 'RODRIGUEZ'
WHERE nombre = 'ANTONIO'
```

```
AND apellido1 = 'GARCIA'
AND apellido2 = 'BENITO'
```

Ahora la tabla 'personas' quedará así:

nombre	**apellido1**	**apellido2**
ANTONIO	PEREZ	GOMEZ
LUIS	LOPEZ	PEREZ
ANTONIO	GARCIA	**RODRIGUEZ**
PEDRO	RUIZ	GONZALEZ

Una vez que hay datos en la tabla, podríamos tener la necesidad de modificar los mismos. Para hacerlo, utilizamos el comando UPDATE. La sintaxis para esto es.

```
UPDATE "nombre_tabla"
SET "columna_1" = [nuevo valor]
WHERE "condición";
```

Por ejemplo, digamos que actualmente tenemos la tabla a continuación:

Tabla *Store_Information*

Store_Name	Sales	Txn_Date
Los Ángeles	1500	05-Jan-1999
San Diego	250	07-Jan-1999
Los Ángeles	300	08-Jan-1999
Boston	700	08-Jan-1999

y notamos que las ventas para Los Ángeles el 08/01/1999 es realmente de 500€ en vez de 300€ dólares estadounidenses, y que esa entrada en particular necesita actualizarse. Para hacerlo, utilizamos el siguiente SQL:

```
UPDATE Store_Information
SET Sales = 500
WHERE Store_Name = 'Los Angeles'
AND Txn_Date = '08-Jan-1999';
```

La tabla resultante sería:

Tabla *Store_Information*

Store_Name	Sales	Txn_Date
Los Angeles	1500	05-Jan-1999
San Diego	250	07-Jan-1999
Los Angeles	500	08-Jan-1999
Boston	700	08-Jan-1999

En este caso, hay sólo una fila que satisface la condición en la cláusula WHERE. Si hay múltiples filas que satisfacen la condición, todas ellas se modificarán.

También es posible UPDATE múltiples columnas al mismo tiempo. La sintaxis en este caso se vería como la siguiente:

```
UPDATE "nombre_tabla"
SET colonne 1 = [[valor1], colonne 2 = [valor2]
WHERE "condición";
```

Funciones de agregado

Una *función de agregación* es una función que resume las filas de un grupo en un solo valor. COUNT, MIN y MAX son ejemplos de funciones de agregación.

COUNT

Regresa la cuneta de todos los valores del SELECT. También se puede usar la opción DISTINCT para solamente contar los valores distintos.

AVG

Regresa el promedio de todos los valores del SELECT. También se puede usar la opción DISTINCT para calcular el promedio de los valores distintos.

MIN

Regresa el valor mínimo de todos los valores del SELECT.

MAX

Regresa el Valor máximo de todos los valores del SELECT.

SUM

Regresa la suma de todos los valores del SELECT.

Agrupamiento

La cláusula GROUP BY **te permite organizar las filas de una consulta en grupos**. Los grupos están determinados por las columnas que se especifican en la cláusula GROUP BY.

En la práctica, la cláusula GROUP BY a menudo se usa con funciones agregadas para generar informes resumidos.

Una función agregada realiza un cálculo en un grupo y devuelve un valor único por grupo. Por ejemplo, **COUNT() devuelve el número de filas en cada grupo**. Otras funciones agregadas comúnmente utilizadas son SUM(), **AVG() (promedio)**, **MIN() (mínimo)**, **MAX() (máximo)**.

La cláusula GROUP BY organiza las filas en grupos y una función agregada devuelve el resumen (El total de registros, el valor mínimo, el valor máximo, el promedio, la suma, etc.) para cada grupo.

Por ejemplo, la siguiente consulta devuelve el número de pedidos realizados por el cliente por año:

```
SELECT customer_id, YEAR (order_date) order_year, COUNT (order_id) order_placed
FROM sales.orders
WHERE customer_id IN (1, 2)
GROUP BY customer_id, YEAR (order_date)
ORDER BY customer_id;
```

Filtros

En bases de datos con muchos objetos, puede usar el filtrado para buscar tablas, vistas y otros elementos específicos. En esta sección se describe cómo filtrar las tablas, pero puede seguir estos pasos en cualquier otro nodo del Explorador de objetos:

1. Conéctese a su servidor de SQL Server.
2. Expanda Bases de datos > Elija su Base de Datos > Tablas. Aparecerán todas las tablas de la base de datos.
3. Haga clic con el botón derecho en Tablas y, después, seleccione Filtro > Configuración del filtro
4. En la ventana **Configuración del filtro** puede modificar algunas de las siguientes opciones de filtro: Filtro por Nombre o Filtro por Esquema.
5. Para borrar el filtro, haga clic con el botón derecho en **Tablas** y, después, seleccione **Quitar filtro**.

Fuentes: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/533_update.html
https://estradawebgroup.com/Post/-Como-agrupar-registros-en-SQL-Server-con-la-clausula-GROUP-BY-/20372