Curso Introducción a SQL Espacial sobre PostGIS

Structured Query Language (SQL)

Comandos SQL de manipulación

formaSIG

La plataforma de aprendizaje en SIG Libre

sigte SERVEI DE SISTEMES D'INFORMACIÓ GEOGRÀFICA I TELEDETECCIÓ Universitat de Girona

UdGFormació FUNDACIÓ UNIVERSITAT DE GIRONA: INNOVACIÓ I FORMACIÓ

Edita: Servicio de SIG y Teledetección (SIGTE) de la Universitat de Girona

Año: 2014

Contenidos elaborados por: Toni Hernández Vallès

Comandos SQL de manipulación

A diferencia de los comandos SQL de estrutura que, como hemos visto anteriormente, definen el esqueleto de nuestra base de datos, los comandos SQL de manipulación de datos se utilizan para introducir, modificar y eliminar los datos propiamente.

Para poder ejecutar los comandos SQL de manipulación es necesario que esos comandos respeten la estructura de la base de datos. Por ejemplo, si intentamos introducir valores nulos para una columna definida para que no acepte valores nulos, entonces obtendremos un error en tiempo de ejecución.

INSERT INTO

El comando INSERT INTO se utiliza para insertar datos (filas) en una tabla.

Sintanxis:

```
INSERT INTO <nombre_tabla>
[<lista_columnas>]
VALUES (<lista_valores>);
```

Para insertar una nueva fila en la tabla vendedor que hemos visto anteriormente utilizaremos el siguiente comando:

```
INSERT INTO vendedor (vendedorId, nombre,primer_apellido, segundo_apellido,
fecha_nacimiento) VALUES (1, 'Esteban', 'Hernández', 'Rivera', '972342345');
```

Los valores pertenecientes a las columnas de tipo alfanumérico (nombre, apellidos, telefono) deben ir indicados entre comillas sencillas ('). Los valores numéricos, por su parte, no requieren de comillas.

Si nos fijamos veremos que la lista_columnas aparece entre corchetes [], lo que significa que es opcional. Únicamente si conocemos de antemano el número de columnas de una tabla así como su orden dentro de la tabla podemos simplificar el comando obviando la lista de columnas.

El ejemplo anterior quedaría entonces:

```
INSERT INTO vendedor VALUES (1, 'Esteban', 'Hernández', 'Rivera', '972342345');
```

COPY

El comando COPY permite tanto importar los datos de un fichero a una tabla, como exportar datos de una tabla a un fichero. Los ficheros de importación o exportación pueden ser tanto ficheros de texto plano como ficheros binarios.

Sintaxis de importación para ficheros de texto:

COPY <nombre_tabla> **FROM** 'ruta_al_fichero'
[**WITH DELIMITER AS** <caracter>];

Donde <caracter> es el carácter utilizado dentro del fichero para indicar la separación entre columnas. En caso de no utilizar la parte opcional WITH DELIMITER AS...., se utilizará la tabulación como carácter de separación. Esto se aplica tanto para la importación como para la exportación.

Aspecto que presenta un fichero donde las columnas están separadas por tabulaciones.

1 Esteban Hernández Rivera 972342345
2 Jorge Ferrer Comas 555-785-156

Aspecto del mismo fichero donde las columnas están separadas por comas.

1,Esteban,Hernández,Rivera, 972342345
2,Jorge,Ferrer,Comas,555-785-156

Sintaxis de importación para ficheros binarios:

En este caso no es preciso indicar como debe llevarse a cabo la separación entre columnas.

COPY <nombre_tabla> **FROM** 'ruta_al_fichero' [**AS BINARY**];

Ejemplo:

COPY vendedor **FROM** 'c:\usuario\alumno\vendedor.txt' **AS BINARY**;

Sintaxis para la exportación de ficheros de texto:

COPY <nombre_tabla> **TO** 'ruta_al_fichero';

DELETE FROM

Utilizaremos este comando para eliminar datos (filas) de una tabla. El comando DELETE FROM permite eliminar todos los datos de una tabla o bien únicamente los datos que cumplan cierta condición.

Sintanxis:

DELETE FROM <nombre_tabla>
[**WHERE** <condicion>];

Si queremos eliminar todos les vendedores de la tabla vendedor deberemos utilizar el siguiente comando:

```
DELETE FROM vendedor;
```

En cambio si queremos eliminar únicamente los vendedores que cumplan una condición utilizaremos, por ejemplo, el siguiente comando:

```
DELETE FROM vendedor WHERE nombre='Esteban';
```

En este caso se eliminarán todas las filas de los vendedores cuyo nombre sea 'Esteban'. Esta es una
condición sencilla pero podemos definir condiciones mucho más complejas y específicas. Veremos
más sobre las condiciones cuando veamos el comando de selección SELECT.

UPDATE

Comando de modificación de datos existentes.

Sintanxis:

UPDATE <nombre_tabla>

```
SET <nombre_columna> = <expresión>
[,<nombre_columna> = <expresión>]...
[WHERE <condición>];
```

Si queremos modificar el nombre y el primer apellido del vendedor cuyo vendedorId es igual a 1, utilizaremos:

```
UPDATE vendedor
SET nombre='Luis', primer_apellido='Morera'
WHERE vendedorId=1;
```

Igual que con el comando DELETE FROM , es muy importante definir bien las condiciones para evitar modificaciones no deseadas.

sigte
SERVEI DE SISTEMES D'INFORMACIÓ GEOGRÀFICA I TELEDETECCIÓ
Universitat de Girona

UdGFormació
FUNDACIÓ UNIVERSITAT DE GIRONA: INNOVACIÓ I FORMACIÓ

www.sigte.udg.edu/formasig

formasig@sigte.org