UNIVERSIDAD OBERTA DE CATALUNYA

Ingeniería Informática

EVOLUCIÓN DEL ÍNDICE ESPACIAL

PARA LA EXTENSIÓN JASPA SOBRE H2

MANUAL DE USUARIO

Alumno/a: Santiago González Prieto

Dirigido por: Jesús Manuel de Diego Alarcón

CURSO 2011-12 (Febrero)

Índice de contenido

1. Introducción

Este documento describe el funcionamiento de las funcionalidades desarrolladas durante el proyecto Fin de Carrera (PFC) “Desarrollo de un índice espacial para la extensión JASPA sobre H2”.

Inicialmente se describirá la preparación y configuración del entorno de ejecución necesario y la creación de una serie de tablas con datos de prueba, para pasar posteriormente a describir cómo se deben ejecutar las distintas funcionalidades implementadas.

2. Preparación y configuración del entorno

Una vez tenemos el sistema con los elementos básicos necesarios como la máquina virtual Java, la base de datos H2 instalada y la librería JASPA instalada sobre dicha base de datos, procederemos a realizar la instalación y configuración del proyecto de cara a ser ejecutado en dicha máquina.

Como primer paso, desplegaremos los ficheros del proyecto en un directorio del disco duro de la máquina, por ejemplo en la ruta *c:\PFC*.

Una vez hecho esto, procedemos a configurar la base de datos para que pueda encontrar los nuevos ficheros de funcionalidades. Para ello, editaremos el fichero h2.bat que se encuentra dentro de la ruta c:\jaspa4h2\bin en caso de haber utilizado la configuración por defecto al instalar el software de JASPA.

En este fichero, debemos añadir la ruta de las nuevas librerías incluidas en el proyecto, modificando el siguiente elemento:

SET JARS=%JASPA%\lib;
c:\PFC\dist\lib\addons.jar;c:\PFC\dist\lib\log4j.jar;c:\PFC\dist\lib.jar;c:\PFC\dist\lib\trove.jar;c:\PFC\dist\PFC.jar;

Una vez guardados los cambios realizados en este fichero, podremos iniciar la base de datos H2 ejecutando este mismo fichero h2.bat, y contando desde este momento con el acceso a las nuevas funcionalidades.

A continuación se accederá a H2 a través de su consola web, utilizando el comando h2console.bat dentro de la ruta c:\jaspa4h2\bin, y entraremos en el sistema con los datos ya creados por defecto para la base de datos en la que vamos a realizar las operaciones.

Una vez dentro del sistema, ejecutaremos el comando:

CREATE ALIAS IF NOT EXISTS JASPA.ST_INIT_SPIDX FOR "FunctionIdx.st_init_IDX"

que asignará dentro del sistema un nombre para la función de inicio utilizada en el fichero "arranque.sql".

Con estos pasos el sistema estará disponible para comenzar a trabajar, siempre teniendo en cuenta que la cadena de conexión a utilizar en la pantalla de login de la consola de h2 deberá de ser:

jdbc:h2:tcp://localhost/~/myfirstjaspadb;INIT=RUNSCRIPT FROM 'c:/pfc/config/arranque.sql';SCHEMA_SEARCH_PATH=PUBLIC,JASPA

3. Carga de datos de prueba

Preparado ya el entorno de ejecución en la máquina, vamos a proceder a realizar la creación y carga de datos en unas nuevas tablas que contendrán los datos geométricos sobre los que aplicaremos las ejecuciones de prueba que veremos en el siguiente punto de este documento.

Para ello, habremos creado previamente una base de datos de prueba cuando se configuró la librería JASPA en H2 con el siguiente comando:

h2script -url jdbc:h2:tcp://localhost/~/myfirstjaspadb -user sa -password 123 -script c:\jaspa4h2\sql\jaspa4h2.sql -showResults

Contando con que los datos de configuración son los indicados en dicho script, procedemos a realizar la carga de los ficheros de script incluidos dentro de la carpeta test del PFC. Comenzamos con la carga de tabla RIVERS:

h2runscript -url jdbc:h2:tcp://localhost/~/myfirstjaspadb -user sa -password 123 -script c:\PFC\test\riversh2.sql -showResults

A continuación, procedemos a realizar la carga de la tabla USES:

h2runscript -url jdbc:h2:tcp://localhost/~/myfirstjaspadb -user sa -password 123 -script c:\PFC\test\usesh2.sql –showResults

La siguiente será la carga de la tabla SOILS:

h2runscript -url jdbc:h2:tcp://localhost/~/myfirstjaspadb -user sa -password 123 -script c:\PFC\test\soilsh2.sql -showResults

Y, finalmente, procedemos a cargar la tabla PAISES:

h2runscript -url jdbc:h2:tcp://localhost/~/myfirstjaspadb -user sa -password 123 -script c:\PFC\test\paises.sql -showResults

En este momento, contamos con el sistema configurado completamente para poder realizar las pruebas de las funcionalidades junto con datos incorporados a la base de datos y que servirán para poder proporcionar información que devuelva resultas para estas funciones. A continuación procederemos a describir las acciones que el usuario podrá llevar a cabo para poder utilizarlas.

4. Funcionalidades disponibles

En esta sección vamos a describir las distintas funcionalidades que el usuario tiene disponibles tras los desarrollos realizados en este PFC.

4.1. Creación de los índices espaciales

La carga de los índices espaciales para las tablas de prueba que se han incorporado previamente, se realiza en el mismo momento en que el usuario accede a la base de datos H2.

Para ello, y como hemos comentado en puntos anteriores, hemos sustituido la cadena de conexión a H2 que teníamos por defecto al configurar la librería JASPA en ella

jdbc:h2:tcp://localhost/~/myfirstjaspadb;SCHEMA_SEARCH_PATH=PUBLIC,JASPA

por esta otra:

jdbc:h2:tcp://localhost/~/myfirstjaspadb;INIT=RUNSCRIPT FROM 'd:/pfc/config/arranque.sql';SCHEMA_SEARCH_PATH=PUBLIC,JASPA

Al ejecutar el arranque del sistema H2 con los comandos h2.bat y h2console.bat dentro del directorio c:\jaspa4h2\bin, automáticamente arrancará un script interno dentro de H2 que generará los índices espaciales para todos los campos geométricos que se han incorporado en los scripts de carga de datos de prueba de USES, SOILS, RIVERS y PAISES.

Estos índices estarán creados tanto en memoria RAM (no visibles a simple vista por el usuario), como en tablas persistentes de la base de datos. Son precisamente estas últimas las que el usuario podrá comprobar su generación, ya que aparecerá una tabla por cada campo geométrico indexado con el prefijo “SPATIALDATAMBRTABLE_”, seguido del nombre de la tabla origen y del campo geométrico de dicha tabla.

En este momento, podemos comenzar a realizar distintas operaciones que utilizan dichos índices espaciales.

4.2. Consulta espacial de inclusión

La primera de las funcionalidades que vamos a presentar es la que permite determinar los elementos de una geometría perteneciente a la base de datos que están incluidos dentro de un rectángulo dado.

Para ello, utilizaremos el comando:

SELECT ST_CONTAIN_SPIDX(TablaGeometrica,CampoGeometrico,minX,MinY,MaxX,MaxY)

que nos devuelve los elementos pertenecientes a la tabla *TablaGeometrica* dentro del campo geométrico *CampoGeometrico* incluidos totalmente dentro del rectángulo formado por los puntos *minX*, *minY*, *maxX* y *maxY*, utilizando el índice espacial.

Un ejemplo de ejecución de esta sentencia podría ser:

SELECT ST_CONTAIN_SPIDX('SOILS','GEOM',1,1,9000,9000)

En este caso, tendríamos como resultado aquellos datos geométricos pertenecientes a la tabla “SOILS” y al campo “GEOM” dentro de ella, que se encuentran contenidos totalmente dentro de un rectángulo formado por las coordenadas minX,MinY (1,1) y maxX, MaxY (9000,9000)

4.3. Consulta espacial de intersección

La siguiente funcionalidad es la que permite determinar los elementos de una geometría perteneciente a la base de datos que intersectan con un rectángulo dado.

Para ello, utilizaremos el comando:

SELECT ST_INTERSECT_SPIDX(TablaGeometrica,CampoGeometrico,minX,MinY,MaxX,MaxY)

que nos devuelve los elementos pertenecientes a la tabla *TablaGeometrica* dentro del campo geométrico *CampoGeometrico* que tienen toda o parte de su geometría dentro del rectángulo formado por los puntos *minX*, *minY*, *maxX* y *maxY*, utilizando el índice espacial.

Un ejemplo de ejecución de esta sentencia podría ser:

SELECT ST_INTERSECT_SPIDX('RIVERS','GEOM',5700,4600,6200,6000);

En este caso, tendríamos como resultado aquellos datos geométricos pertenecientes a la tabla "RIVERS" y al campo "GEOM" dentro de ella, que tienen toda o parte de su geometría dentro de un rectángulo formado por las coordenadas minX,MinY (5700,4600) y maxX, MaxY (6200,6000)

4.4. Consulta espacial de intersección entre dos campos geométricos

La siguiente funcionalidad nos permite determinar cuáles son los elementos de una geometría perteneciente a la base de datos que intersectan con los de otra geometría (la misma que la primera u otra distinta) dentro la base de datos.

Para ello, utilizaremos el comando:

SELECT _st_intersects(geometriaA, TablaGeometricaA, CampoGeometricoA, geometriaB, TablaGeometricaB, CampoGeometricoB)

que nos devuelve el valor "true" para aquellos elementos pertenecientes a la tabla *TablaGeometricaA* dentro del campo geométrico *CampoGeometricoA* que tienen toda o parte de

su geometría dentro de los elementos pertenecientes a la tabla *TablaGeometricaB* dentro del campo geométrico *CampoGeometricoB*, o el valor “false” en caso contrario.

Para obtener los datos de ambas tablas y realizar su función de intersección, se utilizan los índices espaciales creados en el arranque de la base de datos.

Un ejemplo de ejecución de esta sentencia podría ser:

SELECT _ST_INTERSECTS(A.GEOM,'USES','GEOM',B.GEOM,'SOILS','GEOM') FROM USES A, SOILS B;

En este caso, tendríamos como resultado el valor “true” para aquellas combinaciones de elementos de USES que intersectan con los valores de “SOILS”, mientras que los que no cumplieran esta condición tendríamos como resultado el valor “false”.

Este tipo de funciones son muy útiles para ayudar a reducir, en caso necesario, el rendimiento de otras funciones espaciales de JASPA como ST_INTERSECTION, que devuelve como resultado la geometría resultante de la intersección de dos elementos geométricos.

Si incluimos nuestra función _st_intersects dentro de la consulta, eliminaremos previamente los elementos que no intersectan entre ellos, ejecutando y devolviendo sólo para aquellos casos en que sí se cumple el requisito.

Un ejemplo de este caso podría ser la siguiente sentencia:

SELECT ST_INTERSECTION (A.GEOM, B.GEOM) FROM USES A, SOILS B WHERE _ST_INTERSECTS(A.GEOM,'USES','GEOM',B.GEOM,'SOILS','GEOM');

donde la función ***ST_INTERSECTION (A.GEOM, B.GEOM)*** solo se ejecutará para aquellas combinaciones donde el resultado de ***_ST_INTERSECTS (A.GEOM,'USES','GEOM',B.GEOM,'SOILS','GEOM')*** devuelva el valor “true”, y evitando su ejecución para el resto de ellas.