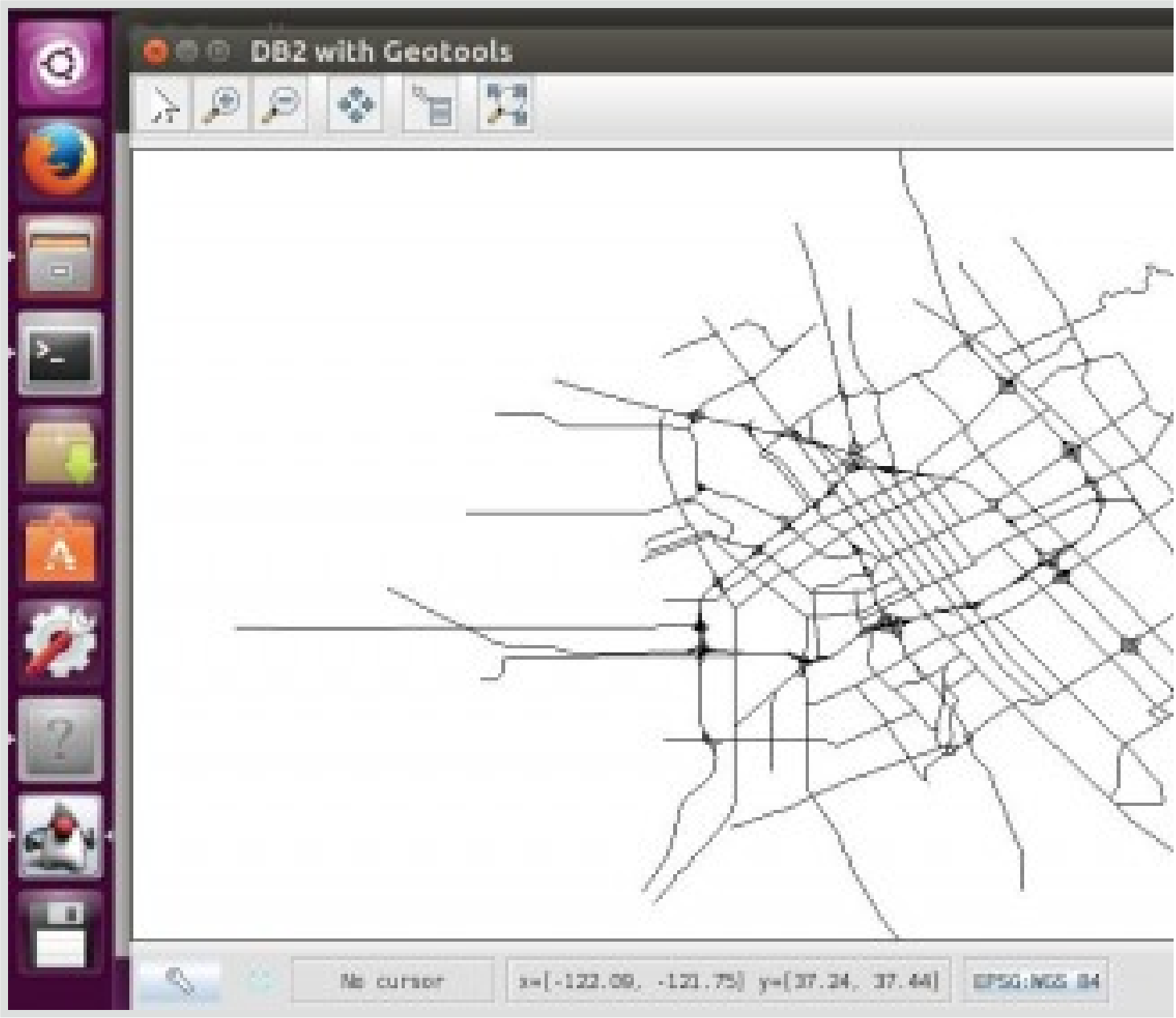
Manage Db2 Spatial Data with open sourc

David Adler

Published on January 12, 2017 / *Updated on February 9, 2021*



Contents

Conclusion

Overview

Skill Level: Any Skill Level

GeoTools is an open source toolkit that provides a Java API to work with spatial data. This recipe tal
steps to visualize and access spatial data in a Db2 database.

Ingredients

- Db2 client and server(Db2 Developer client & server download)
- GeoTools install packages (download)
- GeoTools web site and documentation
- GeoTools Quickstart for Eclipse
- Sample data (download)
- Db2 spatial documentation as appropriate for your environment:
  - Db2 for z/OS Knowledge Center
  - Db2 LUW Knowledge Center

With this recipe you will populate Db2 tables with spatial data, display a simple map and use the GeoT
spatial data from Db2.

This recipe uses Eclipse on Ubuntu Linux but any Java development environment can be used on Wind
platforms.

Step-by-step

1 Setup Db2 and sample data

1. Note: this recipe uses GeoTools on Ubuntu Linux connecting to a local DB2 database. Char connection values in the examples to your database name and userid.
2. Download and unzip the sample data to a convenient directory.
3. Open a Db2 command window where you can execute SQL statements. If you are using a r sure it is cataloged locally. You can use the scripts **catalog-luw.sql** or **catalog-zos.sql** as a changes for the remote server location. (Use **catalog-luw.sql** for DashDB)
   You can execute the script with a command like:
   db2 -tvf catalog-luw.sql
4. Import sample spatial for banks, customers and San Jose streets.
   – Modify the scripts **import-luw.sql** or **import-zos.sql** as appropriate to change the userid procedure. (The scripts contain descriptions of each of the steps and any modifications ne
   – Execute the script with a command like:
   db2 -tvf import-luw.sql
   – Check that there were no processing errors
5. Verify the imported data with the SQL statement (change "**osuser**" to your connection use **db2 select name, street, varchar(db2gse.st_astext(geom),32) from osuser.banks**
   which should return the name, street address and location of the **Meridian** and **San Carlos**

2 Setup development environment

1. If you don't already have a Java 8 JDK (or later) and/or Eclipse environment installed, use
2. Ensure that Java 8 JDK is installed. In a command window, run 'javac -version'. If this does higher, install the Java 8 JDK with the command:
   sudo apt install openjdk-8-jdk
3. From eclipse.org, download the Eclipse Java IDE, **eclipse-java-neon-1a-linux-gtk.tar.gz**, a Archive manager.

3 Display a shapefile using GeoTools

Before attempting to access spatial data in it is a good idea to first go through the excellent Ge

Instead of downloading and opening the shapefile listed in the tutorial, select the file **sjstreets recipe-data** that was downloaded earlier in setting up the database.

This should result in a display similar to the following:

4 Display a map accessing Db2

1. In Eclipse, open the properties for the **tutorial** project to select **Java Build Path > Add Ext** navigate to the DB2 installation directory and select the **db2jcc4.jar** file which in a typical

will be in **/opt/ibm/db2/V11.5/java**.

2. Edit the **pom.xml**, add the following dependencies before **</dependencies>** and save the

```
<dependency>
<groupId>org.geotools.jdbc</groupId>
<artifactId>gt-jdbc-db2</artifactId>
<version>${geotools.version}</version>
</dependency>
<dependency>
<groupId>org.geotools</groupId>
<artifactId>gt-process</artifactId>
<version>${geotools.version}</version>
</dependency>
<dependency>
<groupId>org.geotools</groupId>
<artifactId>gt-wms</artifactId>
```

```
<version>${geotools.version}</version>
</dependency>
<dependency>
<groupId>org.geotools</groupId>
<artifactId>gt-epsg-hsql</artifactId>
<version>${geotools.version}</version>
</dependency>
```

3. Create a new class **DB2TestMap,** replace the generated code with the source code **DB2Te**
recipe-data and save the source file. Comments in the source code describe the various st

connection dialog, connect to DB2, access a particular table (FeatureSource) and display t

4. Run the application which will display a connection dialog and enter the connection values
   Note that the schema must be specified in uppercase.

5. This should display a similar map as above:

5 Use the GeoTools API

Many applications don’t need to visualize a map but simply access the spatial data directly in c analysis.

Create a new class DB2TestAPI, replace the generated code with the source code DB2TestAPI data and save the source file.

Comments in the source code describe the various steps to:

1. Specify the DB2 connection parameters (hard-coded)

2. Connect to the DB2 DataStore

3. Access the BANKS table (FeatureSource) and display the bank name and its location using v

4. Access the CUSTOMERS table (FeatureSource) to get the bounds of all the customers, a cou
within the bounds and a count of a subset of the customers in a smaller rectangular window.

Run the application. The output should look like the following:

6 Conclusion

GeoTools is a widely-used Java-based GIS toolkit that can be used in Java environments to bu
applications that don’t necessarily require visualization of the data.

---

by David Adler

Join The Discussion

Enter your comments...