# New York City - Traffic Collision Mapping Tutorial, by HP Haven OnDemand

In this tutorial, we will be making an app to plot accidents in New York City on its Map. Accident data is being stored in a Vertica Server. Technology stack required for the tutorial is as follows:

1. Vertica Server (<https://my.vertica.com/docs/7.1.x/HTML/index.htm#Authoring/GettingStartedGuide/Other/GettingStartedGuide.htm%3FTocPath%3DGetting%20Started%20Guide%7C_____0> )
2. Java JDK
3. Tomcat Server (Preferably 7)
4. Eclipse Java EE version

The tutorial will assume that environment for the app has been already created. You should be able to make hello world web application (<http://geronimo.apache.org/GMOxDOC30/developing-a-hello-world-web-application.html> ).

Complete code and documentation can found at <https://github.com/gvir/NYAccidentCollisionApp> .

Tutorial is being divided into three main parts:

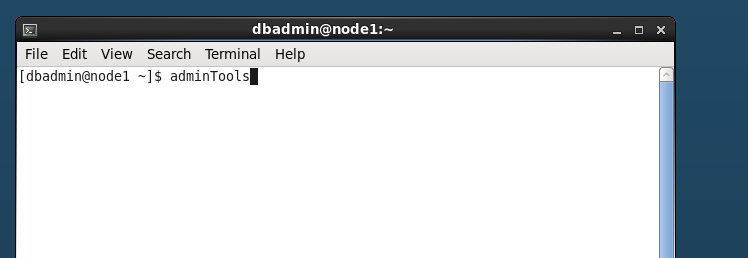
1. Creating table and uploading data in Vertica Server – Data used will be loaded in the Vertica Server that can used for further process.
2. Making connection in Java – User will be able to execute query through java and get the data from Vertica
3. Creating Servlets : They will process requests from client.
4. Creating Client Side of the Application: User will create query which can be executed to get the data from Server and will be plotted on the map.

Overall flow is as follows:

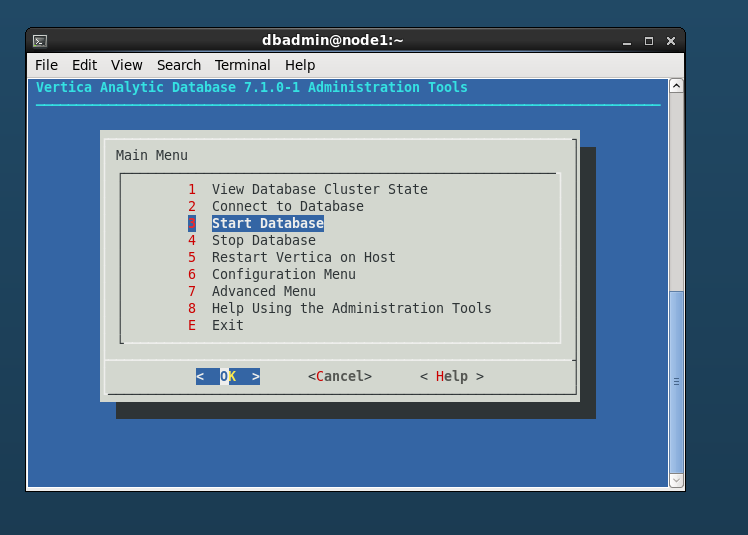
1. User when submit, JavaScript processes the inputs and convert it into vSQL query and calls the server through AJAX request.
2. Server calls the Vertica and executes the query and passes the response in JSON format to client.
3. Client gets the response and plots the data on Map using Google MAP API.

# Creating table and uploading data in Vertica Server

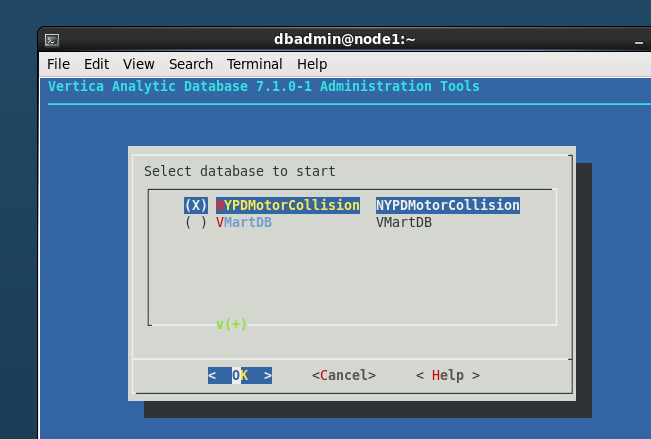
* 1. Get <https://github.com/gvir/NYAccidentCollisionApp/tree/master/data> from github and clone it on the Vertica Server.
  2. Go to console and enter adminTools



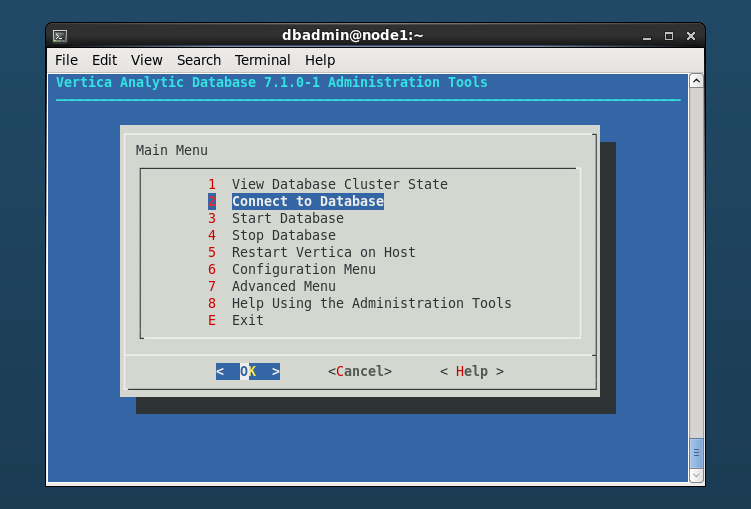
* 1. Click on start database



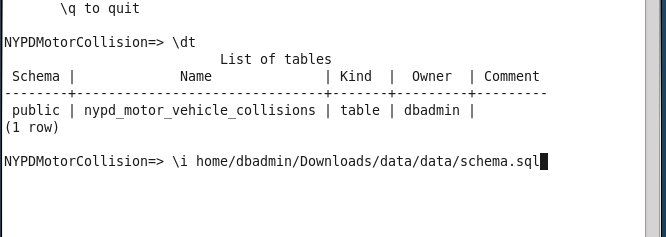
* 1. Start the database after selecting database and then entering password in the next screen.



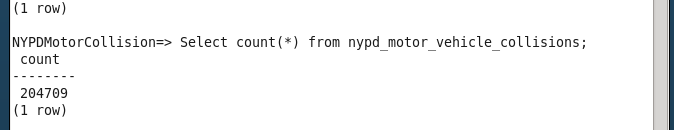
* 1. After database is started, click on start database



* 1. Now you can execute queries. Query to execute a script is \i {path to the sql file}.
  2. Execute schema.sql in data folder to create the table and [data](https://github.com/gvir/NYAccidentCollisionApp/tree/master/data)/[sample\_data](https://github.com/gvir/NYAccidentCollisionApp/tree/master/data/sample_data)/data\_nypd\_motor\_vehicle\_collisions.sql to load data in data folder downloaded in step a.



* 1. Run “Select count(\*) from nypd\_motor\_vehicle\_collisions;” in vsql console. You should be able to see count as 204709 as shown.



* 1. Now you have completed part 1.

# Making Connection in Java

* 1. Here we will be making DAL layer for our application.
  2. Download <https://github.com/gvir/NYAccidentCollisionApp/blob/master/source_code/NYCAccidentsVisualizer/vertica-jdbc-7.1.1-0.jar> and add it in the external libs.
  3. Below snippet of code loads drover class and create a connection to the Vertica server. ClassNotFoundException is thrown if driver class is not found and SQLException is throen if server cannot be contacted. Here sample variable values can be :
     1. Connection string jdbc:vertica://192.168.1.3:5433/NYPDMotorCollision
     2. username = dbadmin
     3. password = password



* 1. Below snippet of code shows how to execute the query using connection



* 1. Complete class can be seen at <https://github.com/gvir/NYAccidentCollisionApp/blob/master/source_code/NYCAccidentsVisualizer/src/Web/VerticaDAL.java>
  2. We have completed the part 2.

# Creating Servlets

* 1. Get Servlet : Return the index.jsp page



* 1. Post Servlet: It gets the query from the client and execute it and then return the data in JSON format. It uses <https://github.com/gvir/NYAccidentCollisionApp/blob/master/source_code/NYCAccidentsVisualizer/gson-2.3.1.jar> for JSON. Import Jar file in the project.



* 1. Whole class can be accessed at <https://github.com/gvir/NYAccidentCollisionApp/blob/master/source_code/NYCAccidentsVisualizer/build/classes/Web/Index.class>

# Creating client side

* 1. UI Filters look like :



* 1. We have made a query generator which will parse the data from the elements and convert it into a vSQL query. Code can be checked at <https://github.com/gvir/NYAccidentCollisionApp/blob/master/source_code/NYCAccidentsVisualizer/WebContent/js/script.js> . Engine class creates the query which can directly executed at vSQL console. Sample queries are :

Select \* from nypd\_motor\_vehicle\_collisions where date >= '2014-03-15 00:00' and date < '2015-03-15 19:00' and number\_of\_persons\_killed > 0 and number\_of\_pedestrians\_injured > 0 and number\_of\_motorist\_killed > 0 and (vehicle\_type\_code\_1 in ('VAN') or vehicle\_type\_code\_2 in ('VAN') or vehicle\_type\_code\_3 in ('VAN') or vehicle\_type\_code\_4 in ('VAN') or vehicle\_type\_code\_5 in ('VAN')) and longitude is not null and latitude is not null LIMIT 400

Select \* from nypd\_motor\_vehicle\_collisions where date >= '2014-03-15 00:00' and date < '2015-03-15 19:00' and number\_of\_persons\_killed > 0 and number\_of\_motorist\_killed > 0 and (contributing\_factor\_vehicle\_1 in ('Backing Unsafely','Traffic Control Disregarded') or contributing\_factor\_vehicle\_2 in ('Backing Unsafely','Traffic Control Disregarded') or contributing\_factor\_vehicle\_3 in ('Backing Unsafely','Traffic Control Disregarded') or contributing\_factor\_vehicle\_4 in ('Backing Unsafely','Traffic Control Disregarded') or contributing\_factor\_vehicle\_5 in ('Backing Unsafely','Traffic Control Disregarded')) and longitude is not null and latitude is not null LIMIT 400

* 1. Now created query has to be passed to the server to get the data. We already know post method in Servlet class takes the query parameter and returns the data in JSON format.



* 1. We have used Google Map to plot the data. As it can be seen we make an AJAX request. Data from the server is then processed to plot the data on the map.
  2. Please refer <https://developers.google.com/maps/documentation/javascript/tutorial#HelloWorld> to start with MAP API.
  3. Please refer <https://developers.google.com/maps/documentation/javascript/markers> to know more about plotting and deleting markers in the graph.
  4. Script code can be found at <https://github.com/gvir/NYAccidentCollisionApp/blob/master/source_code/NYCAccidentsVisualizer/WebContent/js/script.js>
  5. HTML Code can be found at <https://github.com/gvir/NYAccidentCollisionApp/blob/master/source_code/NYCAccidentsVisualizer/WebContent/pages/index.jsp>