

Thanks for submitting your entry for the GIS Solutions Challenge, we've got it safe and sound.

Good luck!



Submission

Return to editing



Robin Rowe's team

Title

GeoTrue SQL Queries

Short description

SQL queries that understand geodesic and route distances and may return either text data or maps.

What problem does your solution solve for? Please check all that apply:

- What is the geodesic distance between two features?
- What is the network distance between two features?
- Is a point inside or outside a polygon?

A primary goal of the GIS Solutions Challenge is to create a tool that is useful to a non-data scientist in decision making (i.e., produces a business insight). Who would use your tool and what would it tell them? What problem does your solution solve for the decision-maker?

Database users would like to be able to run geo queries, but it's difficult or impossible to do so with existing SQL databases. This proposal is in addition to our GeoTrue proposal to write a generic library for geo calculations. It uses the GeoTrue API to extend SQL to enable geo queries to run from a conventional SQL interface. Instead of learning a new tool or needing developers to write an app to talk to the GeoTrue API, users only need to know a few new SQL commands.

What is the output of your solution? Can this output be interpreted by a non-data scientist?

Returns text or maps. An ordinary user of SQL writes geo queries easily that would otherwise need the expertise of a data scientist. A SQL SELECT geo query returns text data in the format a SQL user would expect. When the answer is a map instead of text, it generates an image file with the name specified in the query.

What input files are required? Are any additional formats accepted as an input?

MySQL or Sqlite databases. CSV spreadsheet data files. Maps database.

Please describe a sample workflow for your solution (i.e. steps to achieve an output):

Find every company not Starbucks within 100 feet walking distance of a Starbucks:

```
SET DISTANCE ROUTE FEET;
SELECT * FROM Companies
WHERE Name != 'Starbucks' AND DISTANCE (SELECT FROM Companies WHERE Name = 'Starbucks') < 100;
```

Find all the parking lots within the 1,000 meter range of a mapping drone:

```
SET DISTANCE GEODESIC METERS;
SELECT * FROM Companies
WHERE Name = 'Launch Point #1' AND DISTANCE (SELECT FROM Companies WHERE Purpose = 'Parking') < 1000;
```

More geo queries:

- 1. Parking: Find the nearest parking lot, parking garage or parking meter. Given real-time IoT data, query for the nearest open parking place.
- 2. Transportation: Find the nearest bus stop, ride share location, train station or airport. Given real-time IoT data, get the time the bus will arrive.
- 3. Food: Find the nearest restaurant by type of food served or cost. Find the nearest convenience stores. May restrict results to only those businesses open at a specific time.
- 4. Safety: Find the nearest police or fire station. Return a map showing how to get there from specified location.

Please upload a diagram of your sample workflow

GeoTrueSqlWorkflow.jpeg

Preview

Download

Please describe what language/s your solution involves



Please list any existing packages your solution involves

MySQL or Sqlite.

What existing open source tools/programs would be able to use your solution?

Any system that uses MySQL or Sqlite.

How did you arrive at your solution? What research, or existing tools did you use? How feasible do you think your solution is to implement based on modern open source GIS capabilities?

Researched popularity of SQL databases. For web servers, MySQL is the most popular. For embedded systems, Sqlite is the most popular. Both are open source. We will extend to support geo queries. We will fork them and, assuming they want geo enhancements, push our code enhancements back to the original open source database projects.

We asked BizOps.ai, a business intelligence startup that calculates the value of companies for buyout purchasers, if geo queries would be useful to them. A loud yes! Having geo query capability would be tremendously useful to price the market value of shopping centers based on their locations and what's near them.

What challenges do you think you will face in the implementation of your solution?

Integration with existing codebases always presents interesting challenges. It's a matter of dovetailing our enhancements into a system that didn't expect it. Maybe not that difficult, but potentially time consuming to find the right spot in the code, make the right changes, and perform the relevant QA.

How will your solution handle large amounts of data?

Yes, intended for big data applications. Also for embedded systems, to handle small data.

How is your solution different from existing tools? (i.e. This can be in regards to speed, accuracy, scalability, innovate approach, new design, etc.)

Innovative approach to bring geo capabilities to widely used database programs. Broad compatibility with existing systems that have built upon popular SQL databases.

What's special about our proposal is the potential for worldwide impact by making SQL geo query capabilities available in the most popular databases. Bringing new powerful geo query capabilities to existing systems is game changing. We will change the world.

Return to editing

Was this page helpful?

☒ Yes ☐ No