



# SQL QUERY EXECUTION TIME BETWEEN TWO SINGIDUNUM UNIVERSITY LOCATIONS

## PRIMENA GEOGRAFSKIH INFORMACIONIH SISTEMA U GEO-MODELOVANJU DRUŠTVENIH MREŽA

Verka Jovanović, Vladimir Lazović, Nikola Minić  
Singidunum University, Danijelova 32, Belgrade, Serbia

### Abstract:

The paper presents the creation of SQL queries using mathematical functions that provide the distance between the two points in the shortest time possible and at a highest level of accuracy. The research shows that one of the three most commonly used formulae demonstrated the highest level of accuracy. Haversin formula, which has showed satisfactory accuracy, has been converted into a SQL query where the query execution time was tested. In order to get the information about the location as quickly as possible, the query was expanded with a request to search for the locations within the predefined radius starting from the given coordinates. The aim of this research was to use the results obtained for the purpose of modern mobile phone applications development.

### Key words:

Vincenty's formulae, great circle formula, haversine formula, geolocation, location determination.

### Apstrakt:

Rad prikazuje kreiranje SQL upita na osnovu matematičkih funkcija koje određuju rastojanje između dve tačke u najkraćem vremenskom roku i sa najvećom mogućom preciznošću. Istraživanje je pokazalo da je jedna od tri najčešće korišćene formule ostvarila najveću preciznost. Naime, harvesine formula, koja je jednostavna za korišćenje, takođe je pokazala zadovoljavajuću preciznost, i ubačena je u SQL upit kako bi se ispitalo vreme izvršavanja. Kako bi se u što kraćem vremenskom roku dobile informacije o lokaciji, upit je proširen za zahtevom da pretražuje lokacije u prethodno definisanom radijusu koristeći zadate koordinate. Cilj ovog istraživanja jeste da tako dobijeni rezultati budu iskorišćeni za razvoj savremenih mobilnih aplikacija.

### Ključne reči:

Vinsentijeve formule, formula velikog kruga, 'harvesine' formula, geolokacija, određivanje lokacije.

## 1. INTRODUCTION

Due to the need to constantly move from one place to another, humans have used various methods to calculate the distance between two points with varying degrees of accuracy. The ability to accurately determine the exact position increased along with the technological advancement. The term geolocation can be best described as a geopositioning of people, places or objects. In modern era, this implies the use of devices that have the Internet access (computers, tablets, *etc.*), smartphones or GPS devices. The aim of this paper is to present various formulae for calculating the distance between the two points and to create SQL query which can be included in the smartphone application.

## 2. GEOLOCATION

The position of a person on a map of the world is a single point on that map, which comprises two components, latitude and longitude informing GPS software of one's location (Holdener III, 2011). Once pointed, the information is used by GPS software to get more information for the user, such as the information concerning nearby businesses, traffic jams, or other people. Since it has a point, the application will use a process of reverse geocoding to get this information about the area surrounding the user (Holdener III, 2011). However, the position does not necessarily have to come from a GPS system.

Besides GPS, there are numerous methods for modern computing devices to gain location information, and not all of them rely on GPS satellites in doing so. The following is a list of some of the ways location is processed (Holdener III, 2011, p. 7):

- ♦ IP Address;
- ♦ GSM/CDMA Cell IDs;
- ♦ Wi-Fi and Bluetooth MAC Address;
- ♦ User Input.

IP Address location usage is available for any device connected to a network or the Internet (desktops, printers, routers, *etc.*). This number can be viewed in the same way as a postal address. In most cases, IP Address is provided by the Internet service provider (ISP) in regional formed blocks. Therefore, a country, a region or even a city can be easily identified by an IP Address.

Mobile ID is a unique number that identifies every mobile device in a certain area, similar to IP. There are two most widely used networks (Holdener III, 2011, p. 10):

- ♦ Global System for Mobile Communications (GSM);
- ♦ Code Division Multiple Access (CDMA).

Wi-Fi and Bluetooth MAC Address location usage is available on the devices that use wireless technologies. User Input is available on any device and is software on a device requesting location, such as zip code, from the user via some input method, typically a textbox (Holdener III, 2011).



### 3. DISTANCE CALCULATION METHODS

Accuracy of geolocation is the distance measured from the actual location. The most common way to use precision in GIS (Geographic Information Systems) is to say “point accuracy of 20 meters” which means that the actual location of the point is no more than 20 meters from the location measured for that point.

One of main factors is the accuracy of the manner in which the location is measured. IP address is less accurate than Cell ID, which is less accurate than GPS. The most obvious reason is that the location of the IP address takes the location of the router or firewall that may be miles away from the location from which the request was sent to geolocation. This situation is most common in large corporate environments. Cell ID is more accurate than the IP address geolocation as the extent of triangulation network of towers that are located nearby. GPS is more accurate than Cell ID as the geolocation is determined based on complex calculations of more satellites.

Of course, hardware problems, radio interference, weather conditions, *etc.* affect the signal strength and reduce the accuracy of geolocation request. Therefore, it is important to collect accurate information whenever you apply for geolocation, so that the user can be aware of errors in the application. Accuracy of geolocation information will improve as technology becomes more sophisticated, but even the best solutions remain beyond the reach of the manufacturer.

There are many ways to calculate the distance between the two points on the earth's surface, defined by their latitude and longitude. The methods vary in complexity and accuracy. The simpler the way, the less accurate it is.

#### 3.1 VINCENTY'S FORMULAS

The first (direct) method calculates the location of the point that has a distance and direction of the second point. The second (inverse) method calculates geographical distance and direction between the two given points. These methods are widely used in geodesy because they are accurate to 0,5 mm on the Earth ellipsoid.

Between two nearly antipodal points, the iterative formula may fail to converge.

The most accurate and widely used globally-applicable model for the earth ellipsoid is WGS-84. Other ellipsoids offering a better fit to the local geoid include Airy (1830) in the UK, International 1924 in much of Europe, Clarke (1880) in Africa, GRS-67 in South America, and many others. America (NAD83) and Australia (GDA) use GRS-80, functionally equivalent to the WGS-84 ellipsoid.

#### 3.2 GREAT CIRCLE FORMULA

A great circle is a section of a sphere that contains a diameter of the sphere (Weisstein, 2010). Small circles are sections of

a sphere that do not contain a diameter. A great circle becomes a straight line in a gnomonic projection (Steinhaus, 1999, pp. 220-221).

An orthodome, which presents the shortest path between two points on a sphere, is a segment of a great circle. In order to find the great circle (or geodesic) distance between two points located at latitude  $\delta$  and longitude  $\lambda$  of  $(\delta_1, \lambda_1)$  and  $(\delta_2, \lambda_2)$  on a sphere of radius  $a$ , one must convert spherical coordinates to Cartesian coordinates using

$$\mathbf{r}_i = \begin{bmatrix} \cos \lambda_i \cos \delta_i \\ \sin \lambda_i \cos \delta_i \\ \sin \delta_i \end{bmatrix} \quad (1)$$

Therefore, the great circle distance is then

$$d = a \cos^{-1}[\cos \delta_1 \cos \delta_2 \cos(\lambda_1 - \lambda_2) + \sin \delta_1 \sin \delta_2] \quad (2)$$

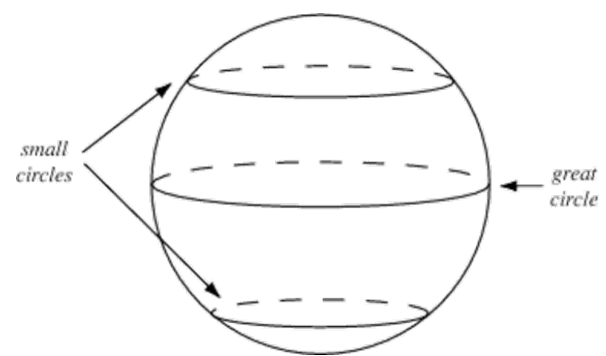


Figure 1. Sections of the sphere(Weisstein, 2010)

In case of the Earth, the equatorial radius is a  $\approx 6.378$  km. However, the Earth's flattening cannot be taken into account in this derivation, because the problem is considerably more complicated for a spheroid or ellipsoid (since both have a radius which is a function of latitude). This leads to extremely complicated expressions for oblate spheroid geodesics and geodesics on other ellipsoids (Weisstein, 2010).

#### 3.3 HAVERSINE FORMULA

The haversine formula is an equation important in navigation, giving great-circle distances between two points on a sphere from their longitudes and latitudes (Sinnott, 1984). It is a special case of a more general formula in spherical trigonometry, the law of haversines, relating the sides and angles of spherical triangles.

Where  $d$  represents the distance between two points with longitude and latitude  $(\phi, \lambda)$  and  $r$  is the radius of the Earth (6.378 km).

Ellipsoid	a	b	f
WGS-84	6.378.137 m ( $\pm 2$ m)	$\approx 6.356.752,314245$ m	$\approx 1 / 298,257223563$
GRS-80	6.378.137 m	$\approx 6.356.752,314140$ m	$1 / 298,257222101$
Airy 1830	6.377.563,396 m	6.356.256,910 m	$\approx 1 / 299,3249646$
International 1924	6.378.388 m	$\approx 6.356.911,946$ m	$1 / 297$
Clarke 1880	6.378.249,145 m	$\approx 6.356.514,86955$ m	$1 / 293,465$
GRS-67	6.378.160 m	$\approx 6.356.774,719$ m	$1 / 298,247167$

Table 1. Latitude/longitude points on different types of ellipsoids (Veness, 2014)



$$d = 2r \arcsin(\sqrt{\text{haversin}(\phi_2 - \phi_1) + \cos(\phi_1) \cos(\phi_2) \text{haversin}(\lambda_2 - \lambda_1)}) = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1) \cos(\phi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \quad (3)$$

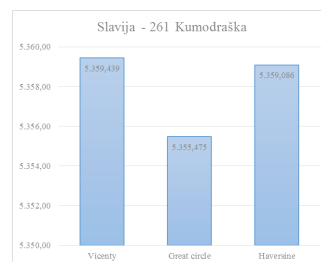
#### 4. THE PROPOSED SOLUTION AND EXPERIMENTAL RESULTS

##### 4.1 COMPARISON OF THE ACCURACY OF THE FORMULAS

The accuracy of these formulas was tested on several examples that included the distance between three Singidunum University's locations in Belgrade and Slavija square. All distances are expressed in meters.

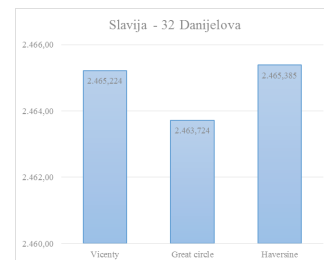
Example 1: Distance in meters between Slavija square and Singidunum University in 261 Kumodraška.

Slavija		261 Kumodraška
44,8025144	Latitude	44,75934854
20,46637058	Longitude	20,49657059
Formulas	Calculated distance	
Vicenty	5.359,439	
Great circle	5.355,475	
Haversine	5.359,086	



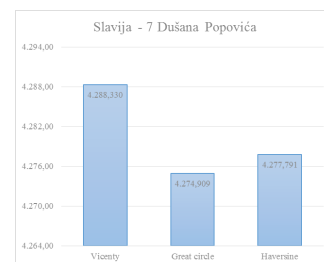
Example 2: Distance in meters between Slavija square and Singidunum University in 32 Danijelova.

Slavija		32 Danijelova
44,8025144	Latitude	44,78217773
20,46637058	Longitude	20,47881603
Formulas	Calculated distance	
Vicenty	2.465,224	
Great circle	2.463,724	
Haversine	2.465,385	



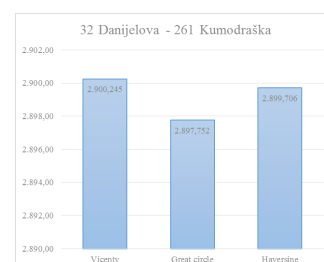
Example 3: Distance in meters between Slavija square and Singidunum University in 7 Dušana Popovića.

Slavija		7 Dušana Popovića
44,8025144	Latitude	44,79060716
20,46637058	Longitude	20,51792264
Formulas	Calculated distance	
Vicenty	4.288,330	
Great circle	4.274,909	
Haversine	4.277,791	



Example 4: Distance in meters between Singidunum University's buildings in 32 Danijelova and 261 Kumodraška.

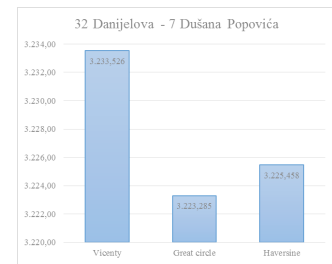
32 Danijelova		261 Kumodraška
44,78217773	Latitude	44,75934854
20,47881603	Longitude	20,49657059
Formulas	Calculated distance	
Vicenty	2.900,245	
Great circle	2.897,752	
Haversine	2.899,706	





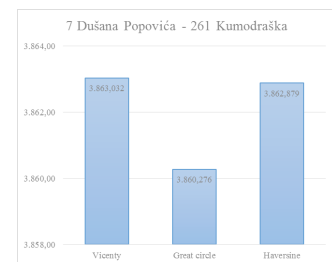
Example 5: Distance in meters between Singidunum University's buildings in 32 Danijelova and 7 Dušana Popovića.

32 Danijelova		7 Dušana Popovića
44,78217773	Latitude	44,79060716
20,47881603	Longitude	20,51792264
Formulas	Calculated distance	
Vicenty	3.233,526	
Great circle	3.223,285	
Haversine	3.225,458	



Example 6: Distance in meters between Singidunum University's buildings in 7 Dušana Popovića and 261 Kumodraška.

261 Kumodraška		7 Dušana Popovića
44,75934854	Latitude	44,79060716
20,49657059	Longitude	20,51792264
Formulas	Calculated distance	
Vicenty	3.863,032	
Great circle	3.860,276	
Haversine	3.862,879	



## 4.2 COMPARISON OF SQL QUERY EXECUTION TIME

Based on the previous testing, we have selected haversine formula, as it proved accuracy high enough for the purpose of this paper.

Based on the chosen formula, SQL query was created in MySQL DBMS, an execution that proved not fast enough, after which the original request was modified. Thus, query execution time was reduced for about half a second, which did not constitute a sufficient performance improvement. The original request was included in the procedure, where the query execution time was reduced for an additional 0,4 seconds, and at the end of the procedure, it included a modified request with achieved execution speed of 0,6 seconds in the standings than 5.000 records.

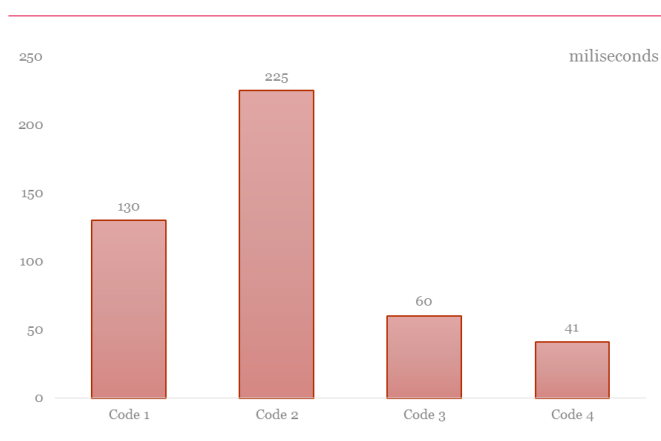


Figure 2. Query execution time

Code 1 shows haversine formulae in the SQL query form used in the paper (Figure 3).

```
set @orig_lat =44.8024516277;
set @orig_lon =20.4662723541;
set @dist =1;
```

```
SELECT stores.*,
6371*2*ASIN(SQRT(POWER(SIN((@
orig_lat -abs(stores.
latitude))*pi()/180/2),2)+COS(@
orig_lat *pi()/180)*COS(abs(stores.
latitude)*pi()/180)*POWER(SIN((@orig_lon
- stores.longitude)*pi()/180/2),2)))as
distance
FROM stores
HAVING distance <= @dist
ORDERBY distance ASC;
```

Code 2 shows modified SQL query, in which coordinates calculation was added in order to avoid the search of the entire table. Query filters objects between two coordinates and then calculates the distance between the objects.

```
set @orig_lat =44.8024516277;
set @orig_lon =20.4662723541;
set @dist =1;

set @lon1 = @orig_lon-@dist/
abs(cos(radians(@orig_lat))*111.22176);
set @lon2 = @orig_lon+@dist/
abs(cos(radians(@orig_lat))*111.22176);

set @lat1 = @orig_lat-(@dist/111.22176);
set @lat2 = @orig_lat+(@dist/111.22176);

SELECT stores.*,
6371*2*ASIN(SQRT(POWER(SIN((@
orig_lat -abs(stores.
latitude))*pi()/180/2),2)+COS(@
orig_lat *pi()/180)*COS(abs(stores.
latitude)*pi()/180)*POWER(SIN((@orig_lon
- stores.longitude)*pi()/180/2),2)))as
distance
FROM stores
WHERE
stores.longitude BETWEEN @lon1 AND @lon2
AND
stores.latitude BETWEEN @lat1 AND @lat2
HAVING distance <= @dist
ORDERBY distance ASC;
```

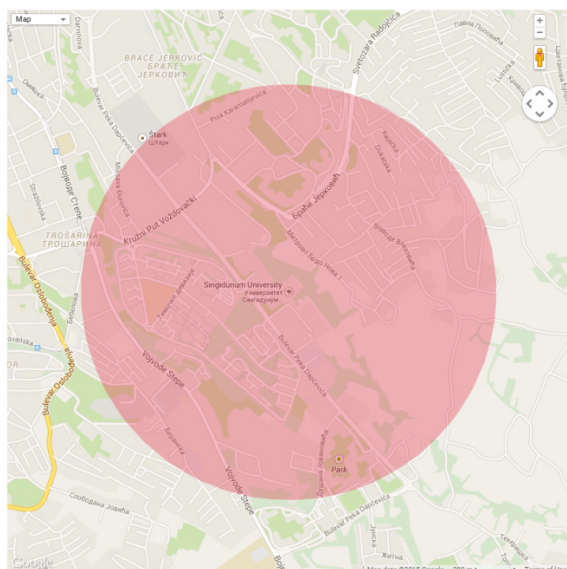


Figure 3.

Code 3 shows implementation of Code 1 into stored procedure.

```
CREATEPROCEDURE `distance`(IN `orig_lat` DOUBLE(12,10),IN `orig_lon` DOUBLE(12,10),IN `dist` FLOAT(4))
BEGIN
SELECT stores.*,
6371*2*ASIN(SQRT(POWER(SIN((orig_lat - stores.latitude)*pi()/180/2),2)+COS(or
ig_lat *pi()/180)*COS(stores.latitude
*pi()/180)*POWER(SIN((orig_lon - stores.
longitude)*pi()/180/2),2)))as distance,
FROM stores
HAVING distance <= dist
ORDERBY distance ASC;
END
```

Code 4 shows implementation of Code 2 into stored procedure.

```
CREATEPROCEDURE `distance`(IN `orig_lat` DOUBLE(12,10),IN `orig_lon` DOUBLE(12,10),IN `dist` FLOAT(4))
BEGIN
set @lon1 = orig_lon-dist/
abs(cos(radians(orig_lat))*111.22176);
set @lon2 = orig_lon+dist/
abs(cos(radians(orig_lat))*111.22176);

set @lat1 = orig_lat-(dist/111.22176);
set @lat2 = orig_lat+(dist/111.22176);
```

```
SELECT stores.*,
6371*2*ASIN(SQRT(POWER(SIN((orig_lat - stores.latitude)*pi()/180/2),2)+COS(or
ig_lat *pi()/180)*COS(stores.latitude
*pi()/180)*POWER(SIN((orig_lon - stores.
longitude)*pi()/180/2),2)))as distance,
FROM stores
WHERE
stores.longitude BETWEEN @lon1 AND @lon2
AND
stores.latitude BETWEEN @lat1 AND @lat2
HAVING distance <= dist
ORDERBY distance ASC;
END
```

## 5. SUMMARY

In this paper, we have proposed the use of the ‘haversine’ formula, as the most accurate methods of calculating the distance between two points. The experimental results obtained by means of this method proved to be accurate by more than one meter than the other two used in the experiment. As haversine formula proved to be the most accurate, it was implemented in SQL query, which was further tested for query execution time with four SQL queries. Query execution time that proved to be sufficiently fast enough will be included in the smartphone application development. The research was conducted on the server with limited resources. However, query execution time can be shortened if dedicated DBMS server is used.

## REFERENCES

- Holdener III, A. T. (2011). *HTML5 Geolocation*. Sebastopol, CA: O’Reilly Media, Inc.
- Sinnott, R. W. (1984). Virtues of the Haversine. *Sky and Telescope*, 68(2), 159.
- Steinhaus, H. (1999). *Mathematical Snapshots 3rd ed.* New York: Dover Recreational Math.
- Veness, C. (2014). *Vincenty solutions of geodesics on the ellipsoid*. Retrieved October 15, 2014, from <http://www.movable-type.co.uk/scripts/latlong-vincenty.html>
- Vincenty, T. (1975). Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations. *Survey Review*, 23(176), 88-93.
- Weisstein, E. W. (2010). *Great Circle*. MathWorld – A Wolfram Web Resource. Retrieved October 15, 2014, from <http://mathworld.wolfram.com/GreatCircle.html>