**Localization Documentation**

**Location Radius**

Retrieve location with a certain radius:

0.5 km

Source: http://www.cs.nyu.edu/visual/home/proj/tiger/gisfaq.html

For short distances such as calculating the radius within 1 km, a flat-earth approximation is sufficient. Flat-earth formulas for calculating distance between two points show noticeable errors when distance is more than 20 km (12 miles).

More computationally accurate results are produced by the Haversine Formula, which gives mathematically exact results. However, as it is computationally heavy for SQL queries, this was not used.

Polar Coordinate Flat-Earth Formula:

where

Combining this equation gives us:

6367\*SQRT(SQUARE(PI()/2-currentLat) + SQUARE(PI()/2-latitude) –

2\*(PI()/2-currentLat)\*(PI()/2-latitude)\*COS(lontigude-currentLon))

SELECT \* FROM kd268.merchantLocations WHERE

6367\*SQRT((PI()/2 - currentLat)^2 + (PI()/2-latitude)^2 - 2\*(PI()/2 - currentLat)\*(PI()/2-latitude)\*COS(longitude - currentLon))

https://developers.google.com/maps/articles/phpsqlsearch\_v3#findnearsql

SELECT id, ( 3959 \* acos( cos( radians(37) ) \* cos( radians( lat ) ) \* cos( radians( lng ) - radians(-122) ) + sin( radians(37) ) \* sin( radians( lat ) ) ) ) AS distance FROM markers HAVING distance < 25 ORDER BY distance LIMIT 0 , 20;

Latitude and Longitude was in degrees, but the formula needed radians!!! DOH

**Web Service Scripts**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Input** | **Output** | **Description** | **Main Querie(s)** |
| PHP Add User Transaction | merchant  customer  productIndex  cost | Result from inserting into table | Add a transaction to the database table “customerTransactions.” If the user has an Android, it attempts to send a push notification via Google Cloud Messaging to notify the user of a new charge. | INSERT INTO kd268.customerTransactions (merchant, customer, productIndex, cost, paid) VALUES ('$merchant', '$customer', '$productIndex', '$cost' , '0') |
| PHP Get All Merchants | userName | userName and merchantName of all merchants | Retrieves a list of all merchants from the table “merchants.” | SELECT userName, merchantName FROM kd268.merchants |
| PHP Get All Subscribed Merchants | userName | Merchants user is subscribed to | Retrieves all merchants in which the user is subscribed to. This information is pulled from the “subscribeForDeals” table. | SELECT merchant FROM kd268.subscribeForDeals WHERE customer = '$userName' |
| PHP Get Deals For Customers | userName | dealIndex of all deals from subscribed merchants | Retrieve all deals from subscribed merchants for a specific user. The queries are made to the “customerDeals” table and the “deals” table. | SELECT dealIndex FROM kd268.customerDeals WHERE userName = '$userName' AND redeemed='0' AND sent='0'  SELECT \* FROM kd268.deals WHERE dealIndex = '$dealIndex' AND enabled = TRUE AND accepted = TRUE ORDER BY merchant DESC |
| PHP Get Deals for Merchants | merchant | All deals for specific merchant | Retrieve all deals for the specific merchant from the “deals” table. | SELECT \* from kd268.deals WHERE merchant = '$merchantName' |
| PHP Get Merchant Transactions | merchant | All transactions of merchant | Retrieve a list of transactions made by the merchant from the “customerTransactions” table. | SELECT \* FROM kd268.customerTransactions WHERE merchant = '$merchant' ORDER BY PurchaseTime DESC |
| PHP Get Nearby Customers | userName | Nearby customers within 25 m radius | Retrieve nearby customers within a radius of 25 m. This query uses  the table “merchantLocations” and “customers.” | SELECT latitude, longitude FROM kd268.merchantLocations WHERE  merchantUserName = '$merchantName'  SELECT \* ,($R \* ACOS(COS(RADIANS($merchantLat)) \* COS(RADIANS(currentLat/$mil)) \* COS(RADIANS(currentLon/$mil) - RADIANS($merchantLon)) + SIN(RADIANS($merchantLat)) \* SIN(RADIANS(currentLat/$mil)))) AS distance FROM kd268.customers HAVING distance < $maxDist ORDER BY distance |
| PHP Get Nearby Merchants | userName | Nearby merchants within 50 m radius | Retrieve nearby merchants within a distance of 50 m. This query uses customer location from “customers” table and “merchantLocations” table. | SELECT currentLat, currentLon FROM kd268.customers WHERE userName = '$currentUser'  SELECT \* ,($R \* ACOS(COS(RADIANS($currentLat)) \* COS(RADIANS(latitude/$mil)) \* COS(RADIANS(longitude/$mil) - RADIANS($currentLon)) + SIN(RADIANS($currentLat)) \* SIN(RADIANS(latitude/$mil)))) AS distance FROM kd268.merchantLocations HAVING distance < $maxDist ORDER BY distance |
| PHP Get Pending Charges | userName | List of pending charges | Retrieve a list of pending charges for the specific user from the “customerTransactions” table. | SELECT \* FROM kd268.customerTransactions WHERE merchant = '$merchantName' and paid = 0 ORDER BY PurchaseTime DESC |
| PHP Get Subscribed Deals For Customers | userName |  | For the specific user, pull information about all the user's subscribed merchants. Then, for each subscribed merchant, put all of the merchant's deals into the “customerDeals” table. | SELECT \* FROM kd268.subscribeForDeals WHERE customer = '$currentUser'  SELECT dealIndex FROM kd268.deals WHERE merchant = '$merchant' AND enabled = TRUE  INSERT INTO kd268.customerDeals (userName, dealIndex) VALUES ('$currentUser', '$dealIndex') |
| PHP Get Targeted Deals For Customers | userName | Result of inserting all targeted deals into db | Put all targeted deals for the specific user in the “customerDeals” table. This script should be called after the update location script. Queries tables “customers”, “userTransactions”, “merchantProductType”, “deals”, and “customerDeals.” | SELECT \* FROM kd268.customers WHERE userName='$userName'  SELECT merchant FROM kd268.userTransactions WHERE customer='  SELECT type FROM kd268.merchantProductType WHERE merchant='$merchant'  SELECT dealIndex ,($R \* ACOS(COS(RADIANS($userLat)) \* COS(RADIANS(targetLat/$mil)) \* COS(RADIANS(targetLon/$mil) - RADIANS($userLon)) + SIN(RADIANS($userLat)) \* SIN(RADIANS(targetLat/$mil)))) AS distance FROM kd268.deals WHERE targetGender='$gender' OR targetGender='B' AND student='$studentStatus' AND minAge<= '$age' AND maxAge>='$age' AND accepted=TRUE AND enabled=TRUE HAVING distance < $maxDist ORDER BY distance  INSERT INTO kd268.customerDeals (dealIndex, userName) VALUES('$dealIndex','$userName') |
| PHP Get User Transactions | userName | All transactions by user | Retrieve all transaction history for the specific user. Queries the table “customerTransactions.” | SELECT \* FROM kd268.customerTransactions WHERE customer = '$userName' ORDER BY PurchaseTime DESC |
| PHP Opt In All Deals | userName |  | User subscribes to all merchants. Queries “merchant” table and “subscribeForDeals” table. | SELECT \* FROM kd268.merchants  INSERT INTO kd268.subscribeForDeals (customer, merchant) VALUES ('$userName', '$merchant') |
| PHP Opt In Merchant | userName  merchant  optIn |  | User subscribes to a specific merchant. Queries “subscribeForDeals” table. | INSERT INTO kd268.subscribeForDeals (customer, merchant) VALUES ('$userName', '$merchant')  DELETE FROM kd268.subscribeForDeals WHERE customer = '$userName' AND merchant = '$merchant' |
| PHP Opt Out All Deals | userName |  | Unsubscribe from all merchants. Queries “subscribeForDeals” table. | DELETE FROM kd268.subscribeForDeals WHERE customer = '$userName' |
| PHP Register New User | name  email  regId | Result of inserting new user into db | Register a new user in the table “gcm\_users.” | INSERT INTO kd268.gcm\_users(name, email, gcm\_regid, created\_at) VALUES('$name', '$email', '$regId', NOW()) |
| PHP Update Merchant Location | latitude  longitude  userName | Result of update query | Update the merchant location in the database. This script can be used for debugging purposes. Queries “merchantLocations” table. | UPDATE `kd268`.`merchantLocations` SET latitude = '$lat' ,longitude = '$lon' WHERE merchantUserName = '$username' |
| PHP Update Payment | userName  paid  cancelled  transactionIndex | Result of update query | Update the specific transaction to paid or cancelled. Queries the table “customerTransactions.” | UPDATE kd268.customerTransactions SET paid = '$paid' , cancelled = '$cancelled' WHERE customer = '$userName' AND transactionIndex = '$transactionIndex' |
| PHP Update User Location | userName  latitude  longitude |  | Updates the user's current location in the database table “customers.” | UPDATE kd268.customers SET currentLat = '$latitude' ,currentLon = '$longitude', timestamp = now() WHERE userName = '$userName' |
| PHP Validate Login | userName  pwd  android | 1 if success, 0 if fail login | Validates login for the user based on the username/password combination. Queries “customers” or “merchants” table. | SELECT \* FROM kd268.customers WHERE userName = '$userName' AND pwd = '$pwd'  SELECT \* FROM kd268.merchants WHERE userName = '$userName' AND pwd = '$pwd'  UPDATE kd268.customers SET android = '$android' WHERE userName = '$userName' |
| PHP Toggle Deal Status | enabled  dealIndex |  | Enable or disable specific deals in the “deals” table. | UPDATE kd268.deals SET enabled = '$enabled' WHERE dealIndex = '$dealIndex' |