UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences Appliquées

January 24, 2023

INFO-H-415 – Advanced databases

First session examination

Consider a social platform where users can post messages. This platform has the particularity of allowing users to post messages with an expiration time, that is, the messages are not shown anymore after this time. To see the message of other users, a user should first subscribe to these users. Messages are stored on servers. Each server stores the messages posted in some regions during some particular time periods.

The social platform is based on the following relational model:

- **User** (ID, Name, Start, End)
  - Start is the moment where the user registers on the platform
  - End is the moment when the user unregisters
- **UserLocation** (UID, Start, point, End)
  - UID references User.ID
  - point is a geometrical POINT. It is the location of the user. To simplify, the platform will suppose that the user is located at "point" from Start to End.
- **UserSubscription** (SID, ID, Start, End)
  - SID and ID both reference User.ID. SID is the identification of the user who subscribes to the messages of the user identified with ID.
  - UID references User.ID
- **Message** (ID, text, UID, Start, End, topic)
  - UID references User.ID
  - Start is the moment where the message is posted
  - End is the expiry date of the message
- **Server** (ID, point)
  - point is a geometrical point. It is the location of the server.
- **ServerStore** (SID, RID)
  - RID references Region.ID.
  - SID references Server.ID
- **Region** (ID, region)
  - region is a geometrical multipolygon.

"Start" and "End" are always timestamps. A same server might store the messages of different regions (leading to several rows with the same SID in the ServerStore table).

1 Temporal and Spatial Databases

For the following questions, suppose you are using a Microsoft SQL Server database.

(1) Give the periods during which two users with ID "1" and "2" are located in the same region.
(2) By using table "R1(Start, End)" containing the results of the first query, give the periods during which the user with ID "1" is subscribed to user with ID "2" but is not located in the same region.
(3) Give the number of subscriptions each user has. Do not coalesce the results.
(4) With table "R3(Name, Count, Start, End)" containing the result of the answer of Question 3, coalesce R3.
(5) Give the server on which the highest number of messages were posted and are still active on "01/01/2023". To know on which server a message is stored, use the location of the user at the time of posting the message.
(6) Give, for each user, the server on which is stored the last active message posted by the user.
(7) By using a table "R5(UID, SID) containing the results of the previous query, give the ID of the user who is currently located the farthest of the server hosting her last active message.
(8) Give, for each server, its distance with the centroid of all the regions for which it is hosting messages.

You can access the current time by using the function "getTime()". You can use the following PostGIS functions:

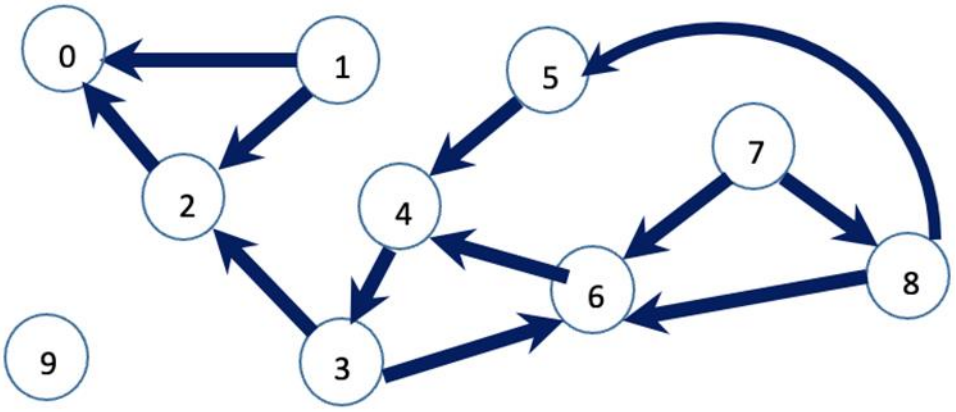
ST_Centroid(geometry) Returns the geometric center of a geometry
ST_Distance(geomA, geomB) Returns the 2D Cartesian distance between two geometries if these are points
ST_DumpPoints(geometry) Returns a set of all points that make up a geometry
ST_Intersection(geomA, geomB) Returns a geometry that represents the shared portion of geomA and geomB.
ST_Intersects(geomA, geomB) Returns TRUE if the Geometries share any portion of space and FALSE if they do not.
ST_Length(geometry) Returns the length of the geometry if it is a Line or MultiLine.
ST_Area(geometry) Returns the area of the geometry if it is a Polygone or Multipolygone.
ST_Union(geometry) Aggregating function, returns a geometry that represents the point set union of the Geometries.
ST_Segmentize(geometry) Return a modified geometry having no segment longer than the given distance
ST_Value(geometry, raster) Returns the value of a given band of a raster at a given geometry point.

2 Active Databases

(1) The location of users is only kept during the time they are registered on the platform.
(2) At each point in time a user has a single location.
(3) The location of a server is contained in its region.
(4) There are at most 3 servers located in a region.

3 Graph Databases

Conside the graph depicted below.



Express in Cypher the following queries

(1) List all n-hops in the graph that do not contain repeated nodes. The result must be of the form: start, length, path, end. There are 49 in the answer. You cannot use the functions in the APOC library. The query must work for any graph instance, not only for the one in the figure. The following is a subset of the answer.

start	length	lista	end
1	1	[0,1]	0
1	1	[2,1]	2
1	2	[0,2,1]	0
2	1	[0,2]	0
3	1	[2,3]	2
3	1	[6,3]	6
3	2	[0,2,3]	0
3	2	[4,6,3]	4
...	...	...	...

(2) Express the same query but to compute all the paths that contain nodes repeated exactly two times (e.g., the path [6, 4, 3, 6] will be in the answer). Again, the solution must work for any graph. For example, some solution may work for the graph in the figure, but will not work if we add an edge from 8 to 7.

Consider that, in the above database, each node is a segment of a path, and the node number is the length of that segment. For example, node 8 is a segment that is followed by segment 5. In that case, the length will be 13 (8 + 5). Express in Cypher the following queries (see next page).

(3) Compute the LONGEST path (not in number of hops, but in length), from node 8 to each of the other nodes. For example, the maximum path from node 8 to 4 has length 30 and is achieved with 5 hops: 8, 5, 4, 3, 6, 4. The Cypher query should work correctly for every graph. If it were executed with the above graph instance, the result should be as follows.

id	longitud	nrhops	r
6	27.0	4	[[6,14.0,1],[6,27.0,4],[6,26.0,4]]
5	13.0	1	[[5,13.0,1]]
4	30.0	5	[4,18.0,2],[4,17.0,2],[4,30.0,5]]
3	21.0	3	[[3,21.0,3],[3,20.0,3]]
2	23.0	4	[2,23.0,4],[2,22.0,4]]
0	23.0	5	[0,23.0,5],[0,22.0,5]]

In this answer, id is the identifier of the final node in the path, and r contains the three possibilities of reaching this node, with lengths 14, 27 and 26, and 1, 4 and 4 hops.

(4) Compute the same as in question (3) but considering only paths without repeated nodes. The answer for the graph in the figure would be as follows.

id	longitud	nrhops
6	26.0	4
5	13.0	1
4	18.0	2
3	21.0	3
2	23.0	4
0	23.0	5