UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences Appliquées

January 21, 2020

INFO-H-415 – Advanced databases

First session examination

The examination is divided in three sections, which are graded by different persons. Please answer to these different sections on different pages.

Schema

Suppose a country that is divided in different regions which themselves are divided in communes. In this country, a smartphone application allows the rental of cars from a float of cars in free service. Citizens wanting to use these cars must register to the system. The history of the users' address should be kept. To be able to use a car in a given commune at a given instant, users must have a subscription for this commune valid at the moment of the rental. Users can have different subscriptions and one subsciption can give acces to more than one commune.

Here is an excerpt of the relational model used:

- **User** (UID, UserName, RegistrationTime)
- **UserAddress** (UID, StartDate, Address, Point, EndDate)
  - Point is a geometrical POINT. It is the location of user's address.
  - UID references User.UID
- **Subscription** (SID, UID, StartDate, EndDate)
  - UID references User.UID
- **CommuneSubscription** (SID, CommuneName)
  - SID references Subscription.SID
  - CommuneName references Commune.CommuneName
- **Country** (CountryName, Altitude, Geom)
  - Altitude is a PostGIS raster containing only one band
  - Geom is a geometrical MULTIPOLYGON
- **Region** (RegionName, Capital, Country)
  - Capital references Commune.CommuneName
  - Country references Country.CountryName
- **Commune** (CommuneName, Geom, RegionName)
  - Geom is a geometrical MULTIPOLYGON
  - RegionName references Region.RegionName
- **CarTrip** (CID, Start, UID, End, Itinerary)
  - Itinerary is a geometrical MULTILINE
  - UID references User.UID
  - CID references the CID of the concerned car (the corresponding table is not represented here).

"StartDate" and "EndDate" are of type Date, while "Start" and "End" are of type Datetime.

1 Temporal and Spatial Databases

For the following questions, suppose you are using a Microsoft SQL Server database. When you need spatial functions, assume the availability of spatial functions such as those provided by PostGIS, shown below.

(1) Give the period during which user with UID 1 was living in the commune called Ixelles but did not have any subscription valid for this commune.
(2) Give the history of the communes for which there was the most subscriptions. Do not coalesce the results.
(3) For each User, provide for each of the Commune the coalesced history for which the user had an active subscription in this Commune.
(4) Give for each region, the number of users which were living in this region during the complete $[01/01/2000, 01/01/2001]$ period. User that moved from on address to another (both address in the same region) should be counted. You should assume that the different addresses of any user form a continuous and uninterrupted lifecycle (if a user has a registred address at a time instant A and another or the same registered address at a later time instant B, then he also has a registered address for each time instant C such that $C \in [A, B]$). Furthermore, a user can not have two different addresses at the same time. We suppose you can compare time points to strings of the form `"dd/mm/yyyy"` using standard ordinal comparators (>, <, =, ...).
(5) List the different regions with their total area.
(6) For each commune, provide the distance between its centroid and the centroid of the capital of the region they are in.
(7) Give the car trip (CID and StartDate time) which has performed the longest segment in the commune of Ixelles.
(8) List for each trip the altitude of the lowest and highest point of the trip.

2 Active Databases

For the following questions, suppose you are using a Microsoft SQL Server database. When you need spatial functions, assume the availability of spatial functions such as those provided by PostGIS, shown below. If needed, you can create a trigger that can be applied on several tables.

(9) Ensure with a trigger that at any instant a user has a single address.
(10) Ensure with a trigger that at any instant a user has a single subscription to a commune.
(11) Ensure with a trigger that the communes in table Commune does not overlap.
(12) Ensure with a trigger that the geometry of a Country is equal to the union of its composing communes.

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.