UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences Appliquées

January 21, 2025

INFO-H-415 – Advanced databases

First session examination

Name: **Last name:**

A hotel chain has put together a database to keep track of various aspects of their business.

The database is based on the following relational schema:

- **City** (CityID, Name, CityGeo, HistoricCenterGeo)
  - CityGeo is a geometrical POLYGON
  - HistoricCenterGeo is a geometrical POLYGON

- **Hotel** (HotelID, HotelName, HotelGeo, FromDate, ToDate, CityID)
  - HotelGeo is a geometrical POINT corresponding to the address of the hotel
  - FromDate is the date of creation
  - ToDate is the date of removal (null if still exists)
  - CityID references City(CityID)

- **HotelRoom** (HotelID, RoomID, MaxPersons)
  - HotelID references Hotel(HotelID)

- **HotelRoomPrice** (HotelID, RoomID, PricePerNight, FromDate, ToDate)
  - (HotelID, RoomID) references HotelRoom(HotelID, RoomID)
  - FromDate is the date of setting the price
  - ToDate is the date of changing the price (maximum date for the current row)

- **RoomReservation** (ReservationID, HotelID, RoomID, ReservationDate, NoOfPersons, FromDate, ToDate)
  - (HotelID,RoomID) references HotelRoom((HotelID,RoomID)))
  - ReservationDate is the date of the reservation
  - FromDate date of check-in
  - ToDate date of check-out

- **TouristCircuit** (CircuitID, Name, CircuitGeo, CityID)
  - CircuitGeo is a geometrical LINESTRING
  - CityID references City(CityID)

- **PointOfInterest** (POIID, Name, POIGeo, CityID)
  - POIGeo is a geometrical POINT
  - CityID references City(CityID)

Name: **Last name:**

1 Spatial Databases

For the following questions, suppose you are using a PostgreSQL database with the POSTGIS extension added. We also suppose working in 4326 (WGS84) spatial reference system.

1. For each city, give the currently most expensive hotel in its historical centre (Justify your computation of ”most expensive”).
2. For each city, rank their tourist circuit by the number of points of interest. A point of interest is part of the tourist circuit if it is less than 100m away from the circuit.
3. For each tourist circuit, compute the time it will take to complete. Suppose a walking pace of 5km/h and each point of interest on the circuit takes 20min.
4. For each hotel, list the tourist circuit that have their starting and ending points less than 1km away from the hotel.

1 Spatial Databases

2 Temporal Databases

For the following questions, suppose you are using a PostgreSQL database (No extensions apart from PostGIS were added!).

1. For each hotel, give the history of the number of unused rooms.
2. For each hotel room, give the time intervals when the number of people was less than the maximum.
3. For each hotel compute their total revenues in 2024. Suppose that the price of a reservation is computed at check-out based on the price per night at that time.
4. For each hotel give the last room booked as well as its total cost to the client.

2 Temporal Databases

3 MobilityDB Databases

For the following questions, suppose you are using a PostgreSQL database with both PostGIS and MobilityDB extensions added. Furthermore, suppose that there are two additional tables, one for municipalities and another that register the trips of tourist guides in charge of the tourist circuits as follows:

- **Municipalities**(MunicipalityId, MunicipalityName, MunicipalityGeo)
  - MunicipalityGeo is a geometry
- **Trips**(TripID, GuideID, CircuitID, StartDate, StartTime, NoPersons, Trip)
  - CircuitID references TouristCircuit.CircuitID
  - StartDate is the date of departure
  - StartTime is the time of departure
  - Trip is a tgeompoint

1. Average duration for each tourist circuit.
2. Total duration spent by tourist circuits in each municipality in December 2024.
3. Average duration spent by tourist circuits visiting each POI. For this, assume that a circuit ”visit” a POI while it is within 500 m of the POI.
4. For each pair of distinct tourist circuits, determine the periods of time when the were less than 500 meters from each other.

3 MobilityDB Databases

4 Active Databases

For the following, suppose you are using a Microsoft SQL Server database with no extension added.

1. A room cannot have two reservations for the same day.
2. The reservations of a room must be done when the hotel is in operation.
3. Define a derived attribute TotalPrice that computes the price of a RoomReservation.
4. Define a derived attribute NoPOIs that counts the number of POIs visited by a TouristCircuit.

4 Active Databases

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

ST_DumpPoints(geometry) Returns a set of all points that make up a geometry

ST_EndPoint(geometry) Returns the last point of a Linestring or CircularLinestring geometry as a POINT.

ST_StartPoint(geometry) Returns the first point of a Linestring or CircularLinestring geometry as a POINT.

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.

ST_Within(geomA, geomB) Returns TRUE if geometry A is within geometry B

You can use the following MobilityDB functions:

- Return the lower or upper bound
  - lower(spans) $\rightarrow$ base
  - upper(spans) $\rightarrow$ base
- Return the value or time span ignoring the potential gaps
  - valueSpan(tnumber) $\rightarrow$ numspan
  - timeSpan(ttype) $\rightarrow$ tstzspan
- Return the trajectory
  - trajectory(tpoint) $\rightarrow$ geo
- Return the start, end, or n-th timestamp
  - startTimestamp(ttype) $\rightarrow$ timestamptz
  - endTimestamp(ttype) $\rightarrow$ timestamptz
  - timestampN(ttype,integer) $\rightarrow$ timestamptz
- Restrict to (the complement of) a set of values

  - atValues(ttype,values) → ttype
  - minusValues(ttype,values) → ttype

- Return the duration
  - duration({datespan,tstzspan}) → interval
  - duration({datespanset,tstzspanset},boundspan bool=false) → interval

- Return the smallest distance ever
  - {geo,tpoint} —=— {geo,tpoint} → float

- Return the temporal distance
  - {point,tpoint} ¡-¿ {point,tpoint} → tfloat

- Restrict to (the complement of) a geometry and a Z span
  - atGeometry(tgeompoint,geometry,zspan=NULL) → tgeompoint
  - minusGeometry(tgeompoint,geometry,zspan=NULL) → tgeompoint