

INFO-H-415 – Advanced databases

First session examination

The exam is divided in four sections. All sub-questions are worth approximately the same amount of points. However, some of these will only take you a minute, some require a bit more thinking, and a couple would require weeks to answer perfectly. Make the best use of your time.

For the three first questions (active, temporal, and spatial databases) we consider the case of an insurance company that uses the following database:

- Client(ClientNo, FirstName, LastName, Location, TelNo)
- Car(PlateNo, Model, Year, ClientNo, FromDate, ToDate)
ClientNo references Client.ClientNo
- InsuranceOffice(OfficeNo, Name, Location, FromDate, ToDate)
- Policy(PolicyNo, PolicyType, PlateNo, InsuranceOfficeNo, Amount, FromDate, ToDate)
PlateNo references Car.PlateNo
InsuranceOfficeNo references InsuranceOffice.OfficeNo
- Accident(AccNo, PlateNo, Location, Description, DateTime)
PlateNo references Car.PlateNo
- Garage(GarageNo, Location, FromDate, ToDate)
- Repair(AccNo, Description, GarageNo, Cost, FromDate, ToDate)
AccNo references Accident.AccNo
GarageNo references Garage.GarageNo

where

- Table Client contains the persons that have a car covered by an insurance policy. The location is a geographic point that represents the client's address.
- Table Car contains information describing cars. The temporality describes the period during which a client owns a car.
- Table InsuranceOffice contains the offices of the insurance representatives who are in charge of managing insurance policies. The location is a geographic point that represents the office's address. The CoveredRegion field is a geographic area that represents the region that the office is covering, i.e., determining which client and which accident it is supposed to manage.
- Table Policy contains information about car insurance policies. The temporality describes the period during which a policy covers a car.
- Table Accident contains the accidents occurred on cars. The temporality describes the instant at which an accident happened.
- Table Garage contains the garages that are available for repairing cars after an accident. The temporality describes the the period during which a garage was open. The location is a geographic point that represents the garage's address.
- Table Repair contains the repairs made on cars after an accident. The temporality describes the period of time during which the repair was made. Notice that a repair may be performed several months after an accident, even if the insurance policy has expired meanwhile.

1 Active Databases (5 pt)

Write the code ensuring the following **integrity constraints**. Whenever multiple triggers are needed to enforce a single integrity constraint, list all of them, but write the code in full for only one of them. Throughout the entire question you should provide at least one example for each of ON INSERT, ON UPDATE, and ON DELETE triggers.

1. A car can only be repaired during its lifespan.
2. An accident can be registered in the database only in the period it is covered by an insurance policy.
3. It is not possible to have a car repaired for an accident if another accident has happened in between. (*Example: If I had an accident two weeks ago, I may not repair the car for that accident today if I had another accident two days ago.*)
4. A car can only be repaired if the cost is less than the amount of the insurance policy.
5. All policies signed by a client must be managed by the same insurance office.

2 Temporal Databases (5 pt)

Consider the same database as for the preceding question. Write the following SQL queries.

1. Give the name and address of clients such that all the cars they currently own are covered by an insurance policy.
2. For each car, give the periods of time in which it was not covered by any insurance policy.
3. For each client, give the number of cars owned by him/her and the corresponding periods.
4. Give the client number of clients who never had an accident for any of their cars during the time the cars were covered by an insurance policy. (It is supposed that an insurance policy is related to exactly one client, i.e., it stops when a car changes its owner.)
5. For each accident that occurred in 2014, give the average time clients waited for the corresponding repairs. (The duration is calculated from the date of the accident and the end date the last repair.)

3 Spatial Databases (5 pt)

Consider the same database as for the preceding question. Write in SQL the following queries, only returning the results that are valid today (based on the temporality of the involved tables).

1. Give the name of the insurance offices that are located outside the region they manage.
2. List the accidents for which the insurance office that is responsible for managing the accident (based on the region each office is managing) is not the same as the one that manages the policy of the car that was involved in the accident.
3. Display the pairs of insurance offices for which the regions they cover partially overlap.
4. For all accidents, give the insurance office that should manage the accident, i.e., that is the closest to the accident.
5. Give the list of all insurance offices that were active in 2014 and, for each, provide the total number of clients and the total number of accidents managed for that same year.

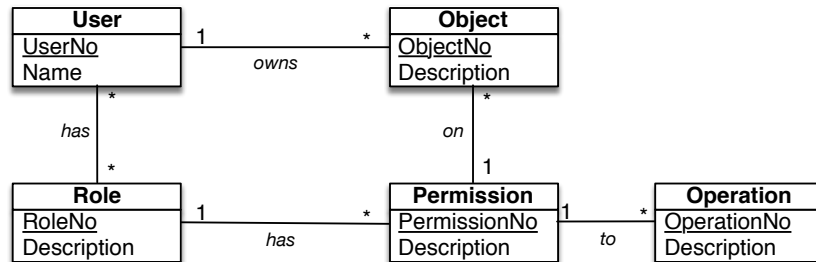
Note. You might need some of the following PostGIS functions:

- `ST_Area(geomA)` — Returns the area of the surface of A (if it is a polygon or multi-polygon).
- `ST_Centroid(geomA)` – Returns the geometric center of a geometry.
- `ST_Contains(geomA,geomB)` – Returns TRUE if and only if no points of B lie in the exterior of A, and at least one point of the interior of B lies in the interior of A.

- ST_Covers(geomA,geomB)– Returns TRUE if no point in Geometry B is outside Geometry A.
- ST_Crosses(geomA,geomB) – Returns TRUE if the supplied geometries have some, but not all, interior points in common.
- ST_Distance(geomA,geomB) – Returns the distance between the two geometries.

4 Object Databases (5 pt)

Consider the simple role-based access control diagram given below



Here is part of the associated types in Oracle:

```

CREATE TYPE TUser;
CREATE TYPE TRole;
CREATE TYPE TObjct;
CREATE TYPE TPermission;
CREATE TYPE TOperation;

CREATE TYPE TSetRefRoles AS TABLE OF REF TRole;
CREATE TYPE TSetRefObjects AS TABLE OF REF TObjct;

CREATE OR REPLACE TUser AS OBJECT(
    UserNo INTEGER,
    Name VARCHAR2(50),
    owns TSetRefObjects,
    hasRoles TSetRefRoles );
    
```

We impose that each relationship can be traversed by queries in **both** directions. Note that, a user can have several roles and that more than one role can contain a permission on a given object. In that case, the union of all these permissions is the user's final permission on that object.

Questions

1. Write the **type definitions** for TPermission, TRole, TObjct, and TOperation.
2. Write the queries that return:
 - (a) the users that have "read" permission (the operation description is "read") on *all* objects
 - (b) the roles that give "write" permission to the user with UserNo = 123 on the object with ObjectNo = 456.
 - (c) the objects that are dangling, i.e. that are not related to any role
 - (d) the objects that are owned by a user but that this very user has not the right to change. (We assume that to be able to change an object, a user must have at least one permission with the description "write" granted on that object.)