Technologies for Information Systems

Part I (10 points)

prof. L. Tanca – June 27th, 2016

Available time: 25 minutes

Last Name ____

First Name ____

Student ID _______________ **Signature**______________________________________

1) Define the typical operations necessary in the multidimensional data model that is at the basis of data warehouses.

2) Consider these data mining problems: classification and clustering. Define them, discuss the differences between the two problems, and provide an application example for each of them.

Technologies for Information Systems

Part II (22 points)

prof. L. Tanca – June 27th, 2016

Available time: 2h 00m

Last Name ____	
First Name ____	
Student ID ________________	**Signature**________________________________

PoliTours is an agency organizing tours in specific cities of Lombardy, relying on a mobile application to promote the tours. A tour is composed of several places of interest to be visited, all located in the same city; each place of interest features a set of landmarks, like monuments or historical buildings. When a user is in a certain city, the PoliTours server delivers to the user's smartphone an XML document describing the tours available for that city, and the mobile application running on the device provides a user-friendly representation of the content of the document. The information system of PoliTours is very primitive: they just keep on their server the collection of XML documents associated with the cities.

UniTours is a similar agency, offering comparable services in Piedmont. A UniTours tour may envisage the visit of points of interest like monuments or historical buildings, but also the visit of companies producing typical local foods. UniTours relies on a better organized data store, constituted by a relational database.

PoliTours and UniTours have recently merged into a unique company named ***UniPoliTours***. The ownership of UniPoliTours wants now to build an integrated relational database storing all the data contained in the PoliTours XML documents and in the UniTours relational database. You must perform the integration assuring to lose the least possible amount of information.

The following is the DTD of the XML documents representing the cities in the PoliTours data store:

```
<!ELEMENT City (Tour+, ReferenceAgency)>
<!ELEMENT Tour (PlaceOfInterest+)>
<!ELEMENT PlaceOfInterest (Landmark+)>
<!ELEMENT ReferenceAgency EMPTY>
<!ATTLIST City    Name CDATA #REQUIRED
                  Province CDATA #REQURED>
<!ATTLIST Tour    Number CDATA #REQUIRED
                  Guide CDATA #REQUIRED
                  DurationInHours CDATA #REQUIRED
                  Price CDATA #REQUIRED>
<!ATTLIST PlaceOfInterest Name CDATA #REQUIRED>
<!ATTLIST Landmark        Name CDATA #REQUIRED
                          Description CDATA #REQUIRED
                          Type CDATA #REQUIRED>
<!ATTLIST ReferenceAgency        Name CDATA #REQUIRED
```

```
                                   Address CDATA #REQUIRED
                                   City CDATA #REQUIRED
                                   OpeningHours CDATA #IMPLIED>
```

- Each city is associated with a reference agency, which is not necessarily located in the city itself. A certain agency may be the reference agency for more than one city.
- The names of cities, landmarks, places of interest and reference agencies are unique.
- Each tour is associated with a progressive tour number, unique for each city.
- The guide is represented by a string in the form *Firstname%Lastname*. The company is small, and you can assume that there are no guides with the same firstnames and lastnames.

The following is a portion of the XML document associated with the city of Milan:

```
<City Name=”Milan” Province=”Milan”>
        <Tour Number=”1” DurationInHours=”5” Guide=”Carlo%Rossi” Price=”100”>
                <PlaceOfInterest Name=”Piazza del Duomo”>
                        <Landmark Name=”Duomo of Milan” Description=”…” Type=”Church”/>
                        <Landmark Name=”Royal Palace of Milan” Description=”…” Type=”Palace”/>
                </PlaceOfInterest>
                <PlaceOfInterest Name=”Piazza della Scala”>
                        <Landmark Name=”Teatro alla Scala” Description=”…” Type=”Theater”/>
                </PlaceOfInterest>
        </Tour>
        <Tour Number=”2” DurationInHours=”7” Guide=”Luca%Bianchi” Price=”50”>
                …
        </Tour>
        <ReferenceAgency Name=”Agency1” Address=”Via Roma, 5” City=”Sesto San Giovanni” OpeningHours=”9-
                18”/>
</City>
```

The schema of the relational database of UniTours is as follows:

CITY (<u>CityName</u>, Province, NumberOfInhabitants)
POINTOFINTEREST (<u>POIName</u>, Description, Type, HistoricalPeriod*, CityName) *// Type may assume values like ‘Theater’, ‘Monument’, ‘Palace’, ‘Church’, …*
FOODCOMPANY (<u>VATNumber</u>, Name, Address, Phone, Email, CompanyType, CityName) *// CompanyType may assume values like ‘Wine producer’, ‘Cheese factory’, …*
PRODUCT (<u>ProductName</u>, Year*, Description, FoodCompanyVATNumber) *// This table contains a selection of typical products. Year assumes a value for some kinds of products, like the wines.*
GUIDE (<u>SSN</u>, Firstname, Lastname, Address, BirthDate)
TOUR (<u>TourName</u>, SSNGuide, DurationInMinutes, Cost)
POINTOFINTERESTTOUR (<u>POIName</u>, <u>TourName</u>)
FOODCOMPANYTOUR (<u>FoodCompanyVATNumber</u>, <u>TourName</u>)

<u>Note</u>: You may assume that the names of places of interest, landmarks and points of interests in the two data sources are disjoint. E.g., you can assume that you will not find any point of interest named “Royal Palace of Milan” in Piedmont.

1. **Source schema reverse engineering**. Provide, for each input data source, the reverse engineering from the logical schema to the conceptual model (ER graph). For the XML source *PoliTours* provide also the relational translation of the ER schema. (5 points)

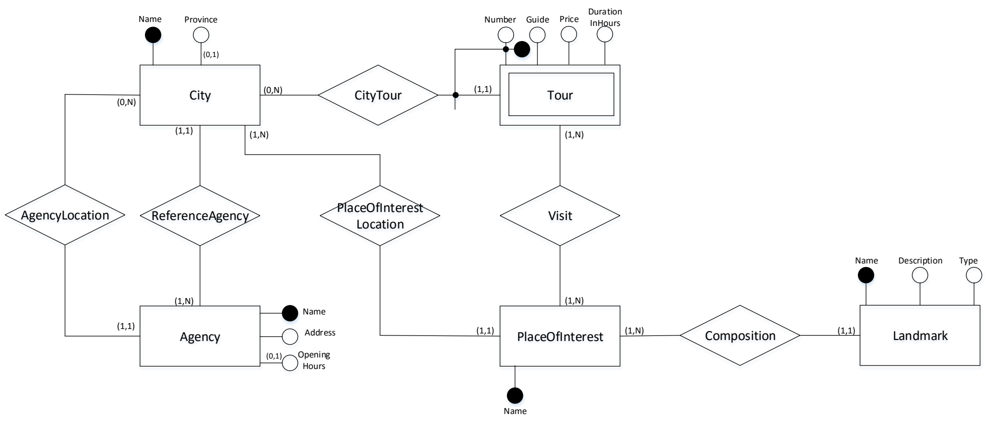
2. **Schema integration**. Design an integrated global conceptual schema (ER graph) for *UniPoliTours* capturing all the data coming from both *PoliTours* and *UniTours*, and provide the corresponding global logical (relational) schema. In more detail, follow these steps:
   a. *Related concept identification and conflict analysis and resolution*. Write a table as shown in the exercise sessions, using the following columns: "PoliTours concept", "UniTours concept", "Conflict", "Solution". (3.5 points)
   b. *Integrated conceptual schema* (ER graph). (3.5 points)
   c. *Conceptual to logical translation of the integrated schema*. (2 points)

3. **Query answering and mapping definition**. Consider the query Q: "Find firstname, lastname and (when available) birth date of the guides who supervise tours envisaging at least one visit in a theater".
   a. *Query formulation*. Consider query Q posed on the logical schema of *UniPoliTours* and write it either in SQL or in Datalog. (1.5 points)
   b. *Mapping definition*. Write the GAV mappings between the schema of *UniPoliTours* and the two sources either in SQL or in Datalog. For *PoliTours*, write the mappings between the *UniPoliTours* integrated relational schema and the *PoliTours* relational schema defined at point 1. Write the mappings only for the tables used to answer query Q**.** (4 points)
   c. *Query rewriting*. Show the rewriting of Q on the two data sources either in SQL or in Datalog. Again, for *PoliTours* consider the relational schema defined at point 1. (2.5 points)

SOLUTION

1. Source schema reverse engineering

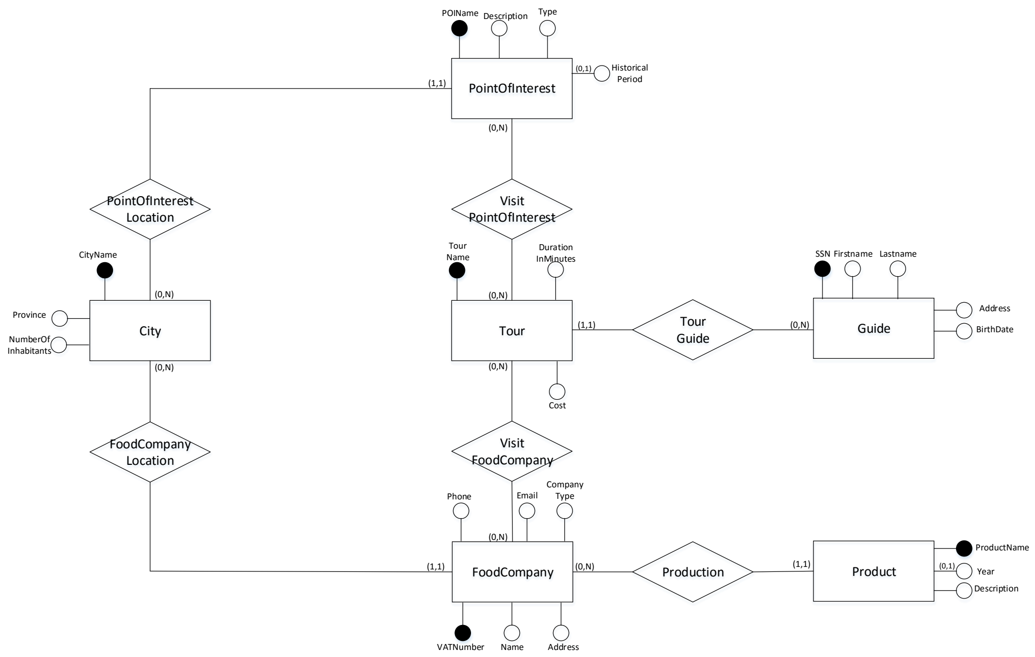
<u>PoliTours</u>



Relational schema
City (<u>Name</u>, Province*, ReferenceAgency)
Agency (<u>Name</u>, Address, OpeningHours*, CityName)
PlaceOfInterest (<u>Name</u>, CityName)
Tour (<u>CityName</u>, <u>Number</u>, Guide, Price, DurationInHours, CityName)
Landmark (<u>Name</u>, Description, Type, PlaceOfInterestName)
PlaceOfInterestTour (<u>PlaceOfInterestName</u>, <u>TourNumber</u>, <u>CityName</u>)

<u>NOTE</u>: In this course we do not study how to implement wrappers. Therefore, we suppose the existence of wrappers guaranteeing the correspondence between the XML source and its relational counterpart, without entering in details.

<u>UniTours</u>

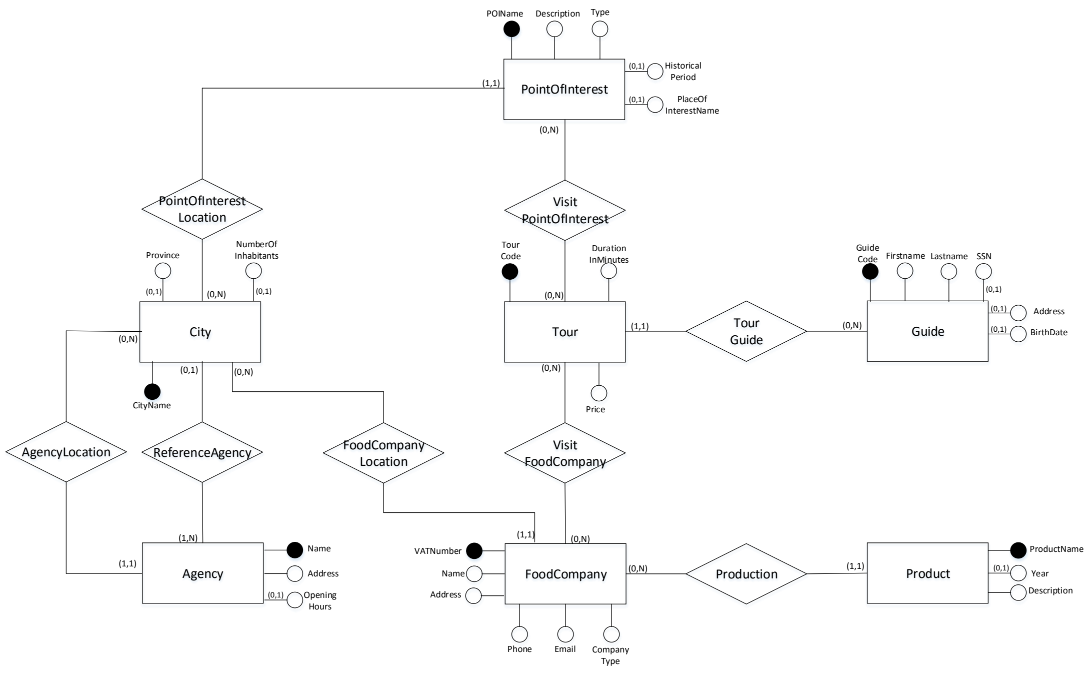


2. Schema integration

2a) Related concept identification + conflict analysis and resolution

PoliTours	UniTours	Conflict	Solution
City	City	Name conflicts	
		- Name → CityName	CityName
Landmark	PointOfInterest	Name conflicts	
		- Entity name	PointOfInterest
		- Name → POIName	POIName
Tour	Tour	Name conflicts	
		- Price → Cost	Price
		Key conflict	
		- CityName+Number → TourName	Generate a TourCode
		Data semantics conflicts	
		- DurationInHours → DurationInMinutes	DurationInMinutes
		Structure conflicts	
		- The tour is associated with places of interest, in their turn associated with landmarks → The tour is directly associated with points of interest (corresponding to landmarks)	Associate the tour directly with landmarks/points of of interest
		Cardinality conflicts	
		- The tour is associated with landmarks in just one city → A tour may be associated with points of interest located in more cities	A tour may be associated with landmarks/points of interest in more cities (the city of the tour will not be specified)
Guide (attribute)	Guide	Structure conflicts	
		- The guide is an attribute → The guide is an entity	The guide is an entity, with a new identifier since SSN is missing in PoliTours
		- Firstname and lastname are represented in the same attribute, separated by '%' → Firstname and lastname are distinct attributes	Firstname and lastname are distinct attributes

2b) Global conceptual schema



2c) Conceptual to logical translation

City (CityName, Province*, NumberOfInhabitants*, ReferenceAgency*)
Agency (Name, Address, OpeningHours*, CityName)
PointOfInterest (POIName, Description, Type, HistoricalPeriod*, PlaceOfInterestName*, CityName)
FoodCompany (VATNumber, Name, Address, Phone, Email, CompanyType, CityName)
Product (ProductName, Year*, Description, FoodCompanyVATNumber)
Guide (GuideCode, Firstname, Lastname, SSN*, Address*, BirthDate*)
Tour (TourCode, GuideCode, DurationInMinutes, Price)
PointOfInterestTour (POIName, TourCode)
FoodCompanyTour (FoodCompanyVATNumber, TourCode)

3. Query answering and mapping definition

3a) Query formulation

Find firstname, lastname and (when available) birth date of the guides who supervise tours envisaging at least one visit in a theater.

```
SELECT DISTINCT G.Firstname, G.Lastname, G.BirthDate
FROM Guide AS G, Tour AS T, PointOfInterestTour AS PT, PointOfInterest AS P
WHERE G.GuideCode=T.GuideCode AND T.TourCode=PT.TourCode AND PT.POIName=P.POIName AND
        P.Type='Theater'
```

3b) GAV mapping definition

The KeyGen(_, _) functions generate univocal identifiers.

```
CREATE VIEW UniPoliTours.Guide (GuideCode, Firstname, Lastname, SSN, Address, BirthDate) AS (
        SELECT KeyGenGuide(Guide, ‘PoliTours’), left(Guide, position(‘%’ in Guide)-1),
                right(Guide, length(Guide)-position(‘%’ in Guide)), null, null, null
        FROM PoliTours.Tour

        UNION

        SELECT KeyGenGuide(SSN, ‘UniTours’), Firstname, Lastname, SSN, Address, BirthDate
        FROM UniTours.Guide
)


CREATE VIEW UniPoliTours.Tour (TourCode, GuideCode, DurationInMinutes, Price) AS (
        SELECT KeyGenTour(CityName||Number, ‘PoliTours’), KeyGenGuide(Guide, ‘PoliTours’),
                DurationInHours*60, Price
        FROM PoliTours.Tour

        UNION

        SELECT KeyGenTour(TourName, ‘UniTours’), KeyGenGuide(SSNGuide, ‘UniTours’),
                DurationInMinutes, Cost
        FROM UniTours.Tour
)


CREATE VIEW UniPoliTours.PointOfInterestTour (POIName, TourCode) AS (
        SELECT L.Name, KeyGenTour(P.TourCity||P.TourNumber, ‘PoliTours’)
        FROM PoliTours.PlaceOfInterestTour AS P, PoliTours.Landmark AS L
        WHERE P.PlaceOfInterestName=L.PlaceOfInterestName

        UNION

        SELECT POIName, KeyGenTour(TourName, ‘UniTours’)
        FROM UniTours.PointOfInterestTour
)


CREATE VIEW UniPoliTours.PointOfInterest (POIName, Description, Type, HistoricalPeriod,
        PlaceOfInterest, CityName) AS (
        SELECT L.Name, L.Description, L.Type, null, P.Name, P.CityName
        FROM PoliTours.Landmark AS L, PoliTours.PlaceOfInterest AS P
        WHERE L.PlaceOfInterestName=P.Name

        UNION
        SELECT POIName, Description, Type, HistoricalPeriod, null, CityName
        FROM UniTours.PointOfInterest
)
```

3c) Query rewriting

```
SELECT left(Guide, position('%' in Guide)-1), right(Guide, length(Guide)-position('%' in Guide)), null
FROM PoliTours.Tour AS T, PoliTours.PlaceOfInterestTour AS PT, PoliTours.Landmark AS L
WHERE T.CityName=PT.CityName AND T.Number=PT.TourNumber AND
        PT.PlaceOfInterestName=L.PlaceOfInterestName AND L.Type='Theater'

UNION

SELECT G.Firstname, G.Lastname, G.BirthDate
FROM Guide AS G, Tour AS T, PointOfInterestTour AS PT, PointOfInterest AS P
WHERE G.SSN=T.SSNGuide AND T.TourName=PT.TourName AND PT.POIName=P.POIName AND
        P.Type='Theater'
```