# Technologies for Information Systems

# Part I (10 points)

prof. L. Tanca – September 8th, 2017

Available time: 25 minutes

**Last Name** _____

**First Name** _____

**Student ID** _____ **Signature**_____

1. Describe the GAV (global as view) and LAV (local as view) approaches used to define the mapping between the global logical schema and the single source schemata in the data integration context. Discuss the main differences between the two approaches and describe under which conditions GAV is more appropriate than LAV, and vice versa. (5 points)

2. Briefly define pervasive data management and the main problems that must be solved. (5 points)

- **During this part of the exam, students are not allowed to consult books or notes.**
- **Students should answer the theoretical questions using their own words, in order for the teachers to be able to assess their real level of understanding.**

# Technologies for Information Systems

# Part II (22 points)

prof. L. Tanca – September 8th, 2017

Available time: 2h 00m

**Last Name** _____

**First Name** _____

**Student ID** _____ **Signature**_____

*PoliConferences* is a company organizing scientific conferences. The conferences are grouped in series, and for each series there is a conference in each year. Users may register to participate in conferences, and each conference defines some registration categories, each associated with its own price. The users register to the conferences selecting a specific registration category, and pay an amount depending on the registration category and on the discount that may be applied.
The management of PoliConferences has now hired you to design a data warehouse to analyze the registrations.

The following is the schema of the operational database used by PoliConferences:

COUNTRY (CountryName, Continent)

CONFERENCE (Series, Year, Country, StartDate, DurationInDays, Capacity) *// Capacity is the maximum number of registrants that can be accepted for this conference. An example of a conference series is: "Very Large DataBases"; every year there is a different conference of the series "Very Large DataBases".*

USER (UserCode, Name, Surname, CountryName)

REGISTRATIONCATEGORY (Series, Year, CategoryName, Price) *// Example of registration category names: "Early" registration, "Regular" registration, "Late" registration. All the prices are expressed in US dollars.*

REGISTRATION (UserCode, Series, Year, CategoryName, RegistrationDate, PercDiscount) *// The discount is a percentage expressed as a number between 0 and 100; each registration may be associated with a different discount.*

1. (3 points) Perform the reverse engineering of the given logical schema into a conceptual schema (Entity-Relationship model).

2. With respect to the produced ER diagram, discover the fact(s) that are useful specifically for answering the queries reported below. For each of these facts:
    a. (3 points) Produce the attribute tree (with pruning and grafting).
    b. (3 points) Produce the conceptual schema (fact schema).

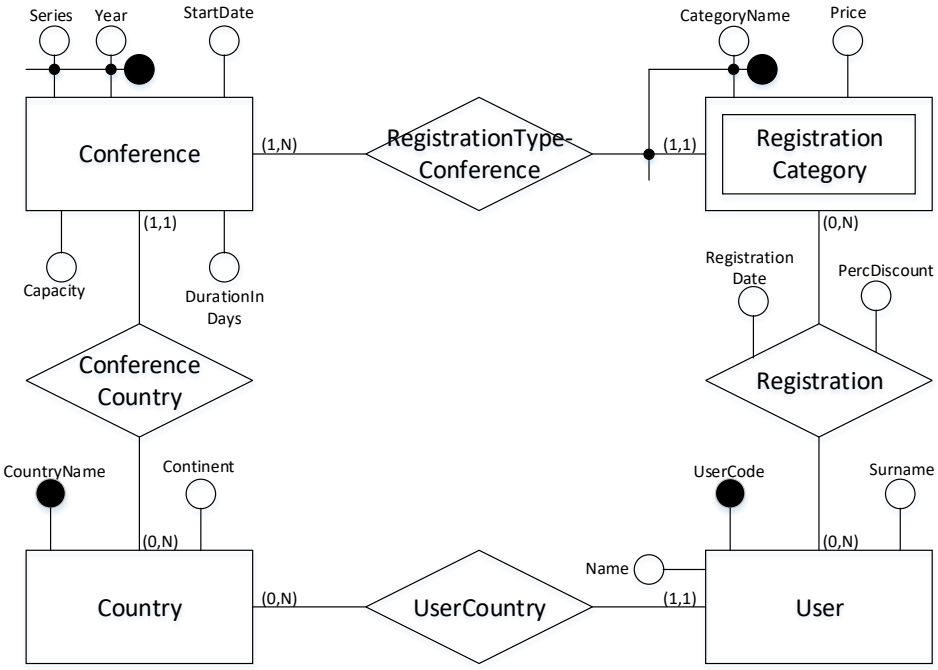
c. (2 points) Produce the glossary.

3. (3 points) Produce a logical schema consistent with the conceptual schema.

4. Write in SQL the following queries against the designed logical schema:
   a. (2 points) Considering only the conferences located in Europe, compute the average discount for each continent of the user and day of the week.
   b. (2 points) Compute the total revenue for each conference country, registration date and user country. Include in the answer also the aggregations computed using only one and two of the three attributes.
   c. (2 points) For each series, compute the conference(s) (specify the year(s)) with the greatest number of registrants.
   d. (2 points) Find the conferences (specify series and year) that in October 2016 have collected from the registrations a revenue greater of more than 20% than that collected in September 2016.

# SOLUTION

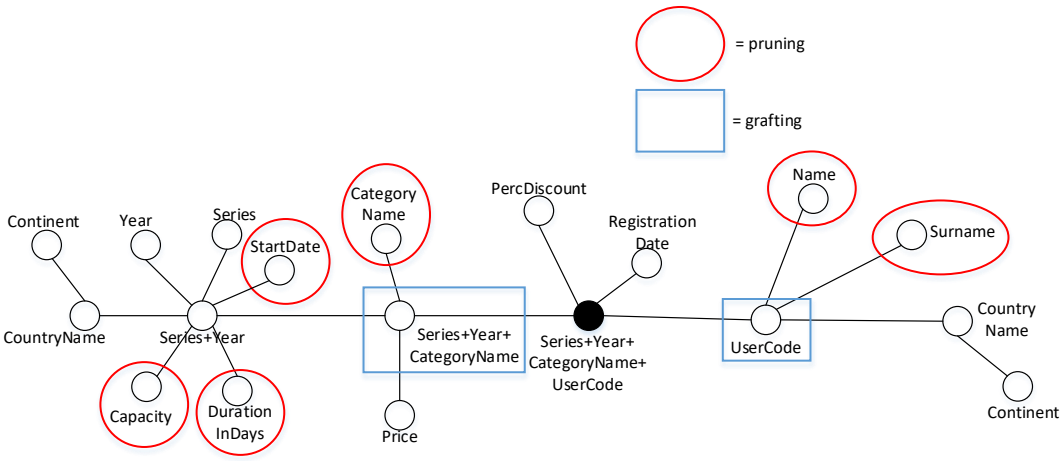
## 1. Reverse engineering



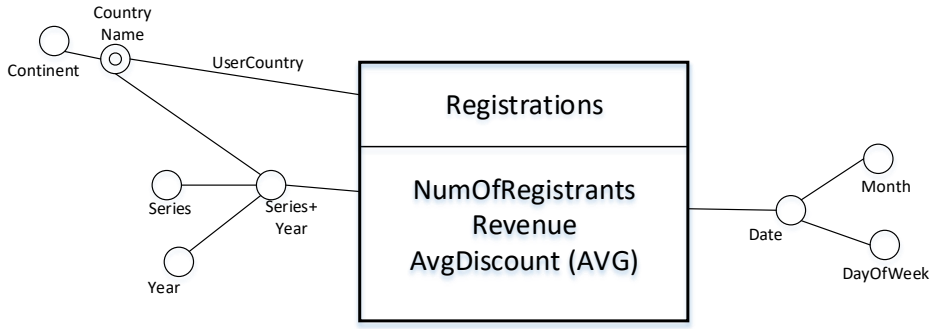

## 2. Conceptual design

Fact: Registrations (Registration relationship)

### 2a) Attribute tree



## 2b) Fact schema




## 2c) Glossary

NumOfRegistrations

**SELECT** U.CountryName, R.Series, R.Year, R.RegistrationDate, **COUNT**(*)
**FROM** Registration **AS** R, User **AS** U
**WHERE** R.UserCode=U.UserCode
**GROUP BY** U.CountryName, R.Series, R.Year, R.RegistrationDate

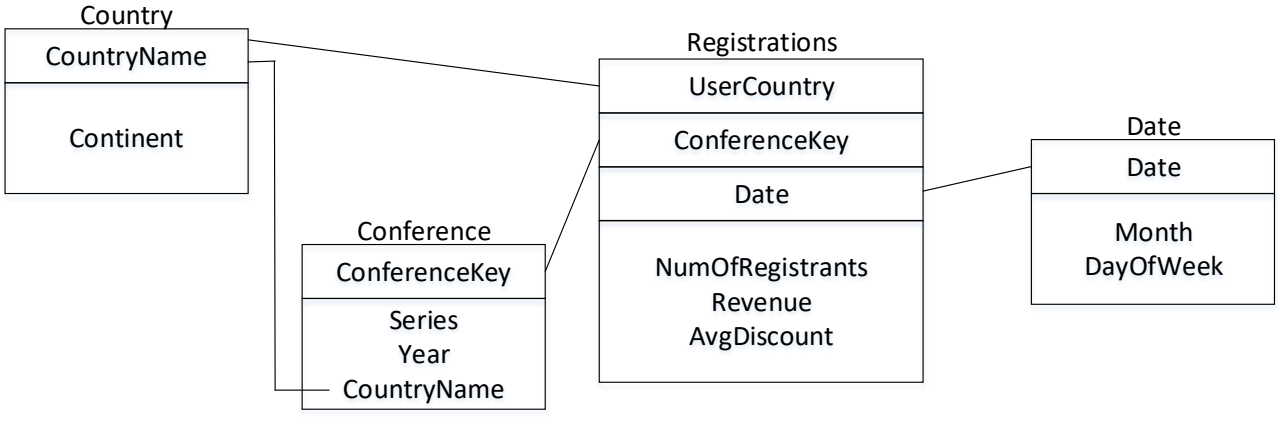
Revenue

**SELECT** U.CountryName, R.Series, R.Year, R.RegistrationDate,
    **SUM**(RC.Price - RC.Price*R.PercDiscount*0.01)
**FROM** Registration **AS** R, User **AS** U, RegistrationCategory **AS** RC
**WHERE** R.UserCode=U.UserCode **AND** R.Series=RC.Series **AND** R.Year=RC.Year **AND**
    R.CategoryName=RC.CategoryName
**GROUP BY** U.CountryName, R.Series, R.Year, R.RegistrationDate

AvgDiscount

**SELECT** U.CountryName, R.Series, R.Year, R.RegistrationDate, **AVG**(R.PercDiscount)
**FROM** Registration **AS** R, User **AS** U
**WHERE** R.UserCode=U.UserCode
**GROUP BY** U.CountryName, R.Series, R.Year, R.RegistrationDate

# 3. Logical design



## 4. Query answering

### 4a) Considering only the conferences located in Europe, compute the average discount for each continent of the user and day of the week.

```
SELECT C-U.Continent, D.DayOfWeek, SUM(R.NumOfRegistrants*R.AvgDiscount)/SUM(R.NumOfRegistants)
FROM Registrations AS R, Conference AS C, Country AS C-C, Country AS C-U
WHERE R.ConferenceKey=C.ConferenceKey AND C.CountryName=C-C.CountryName AND
        R.UserCountry=C-U.CountryName AND C-C.Continent='Europe'
GROUP BY C-U.Continent, D.DayOfWeek
```

### 4b) Compute the total revenue for each conference country, registration date and user country. Include in the answer also the aggregations computed using only one and two of the three attributes.

```
SELECT C.CountryName, R.Date, R.UserCountry, SUM(R.Revenue)
FROM Registrations AS R, Conference AS C
WHERE R.ConferenceKey=C.ConferenceKey
GROUP BY C.CountryName, R.Date, R.UserCountry WITH CUBE
```

### 4c) For each series, compute the conference(s) (specify the year(s)) with the greatest number of registrants.

```
CREATE VIEW ConferenceNumRegistrants (ConferenceKey, Series, Year, Num) AS (
        SELECT C.ConferenceKey, C.Series, C.Year, SUM(R.NumOfRegistrants)
        FROM Registrations AS R, Conference AS C
        WHERE R.ConferenceKey=C.ConferenceKey
        GROUP BY C.ConferenceKey, C.Series, C.Year
)
SELECT C.Series, C.Year
FROM ConferenceNumRegistrants AS C
WHERE C.Num =   (       SELECT MAX(C2.Num)
                        FROM ConferenceNumRegistrants AS C2
                        WHERE C2.Series=C.Series
                )
```

### 4d) Find the conferences (specify series and year) that in October 2016 have collected from the registrations a revenue greater of more than 20% than that collected in September 2016.

```
SELECT C.Series, C.Year
FROM Registrations AS R, Conference AS C, Date AS D
WHERE R.ConferenceKey=C.ConferenceKey AND R.Date=D.Date AND D.Month='Oct-2016'
GROUP BY C.ConferenceKey, C.Series, C.Year
HAVING SUM(R.Revenue) > 1.2 * (
                                        SELECT SUM(R2.Revenue)
```

```
        FROM Registrations AS R2, Date AS D2
        WHERE R2.Date=D2.Date AND D2.Month='Sep-2016' AND
              R2.ConferenceKey=R.ConferenceKey
)
```