Technologies for Information Systems

Part I (10 points)

prof. L. Tanca – September 6th, 2016

Available time: 25 minutes

Last Name _____

First Name _____

Student ID _____ **Signature**_____

1. Consider the following data mining problem: Sequential Pattern Discovery. Define and illustrate it clearly with an application example.
2. Define the concepts of valid time and transaction time in temporal databases also describing their advantages and disadvantages.

- **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 6th, 2016

Available time: 2h 00m

Last Name _____

First Name _____

Student ID _____ **Signature**_____

Polimusic.com is a website allowing its customers to listen to songs by paying a fee for each listening. Each song has a specific price. 60% of the income derived by a listening is retained by Polimusic.com, while the remaining 40% is paid to the artists featuring the song. A song may feature more than one artist, and in this case the money is divided among the featuring artists on the basis of a weight.
The management of *Polimusic.com* has now hired you to design a data warehouse to analyze the listenings.

The following is the schema of the *Polimusic.com* operational database:

ARTIST (ArtistCode, Name, RecordLabel) *// An artist may be either a single person or a band.*
SONG (SongCode, Title, Genre, Language, DurationInSeconds, Price)
SONGARTIST (SongCode, ArtistCode, WeightForProfits) *// The weights assigned to the artists featuring a certain song sum to 1.*
USER (UserCode, Surname, GivenName, BirthDate, HomeCountry, RegistrationDate)
COUNTRY (CountryName, Continent)
LISTENING (UserCode, SongCode, StartDate, StartTime, HighResolution) *// HighResolution is a boolean attribute which is true if the listening was performed using high-resolution audio.*

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

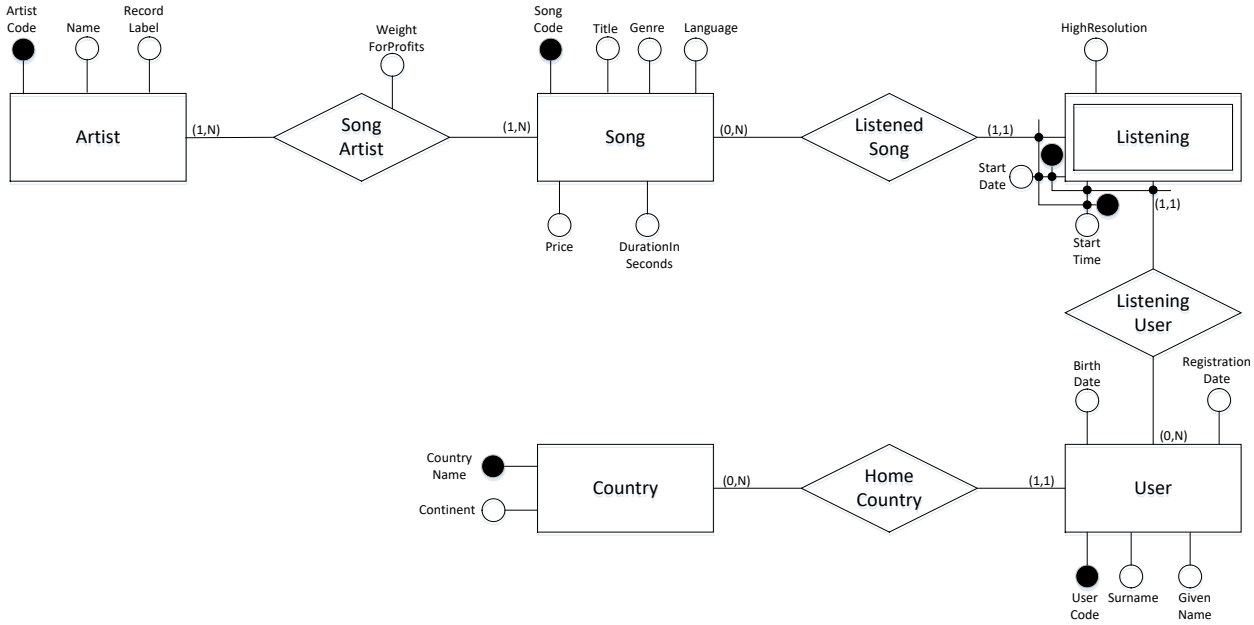
2. With respect to the produced ER diagram, discover the fact(s) that are useful specifically for answering the queries reported below. For each of these facts:
    a. (3 points) Produce the attribute tree (with pruning and grafting).
    b. (3 points) Produce the conceptual schema (fact schema).
    c. (2 points) Identify the measure(s) and 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. (1.5 points) Considering only the listenings performed on Sundays by users born in 1990, find the total income realized by the website (therefore excluding the quota for the artists) grouped by month, user (specify code, surname and given name), type of listening (high resolution or not) and genre of the song.
    b. (1.5 points) For each artist, find the name and the number of listenings realized on 3rd July 2014.
    c. (2.5 points) Find the record label whose artists obtained the greatest total income in 2015.
    d. (2.5 points) Select for each continent the code and the title of the song with the greatest total listening time (note: all the listenings to a song can be considered as complete).

SOLUTION

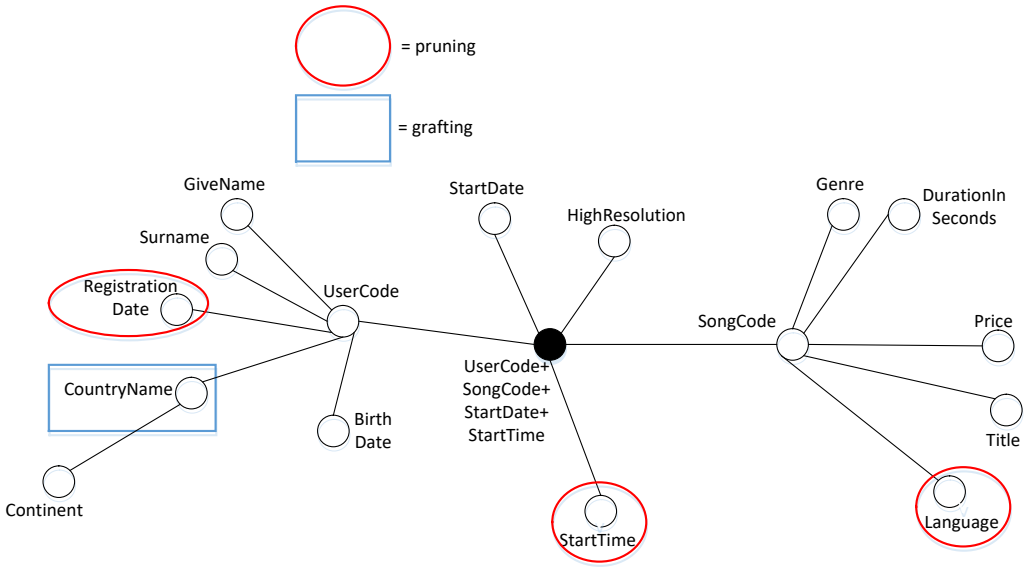
1. Reverse engineering



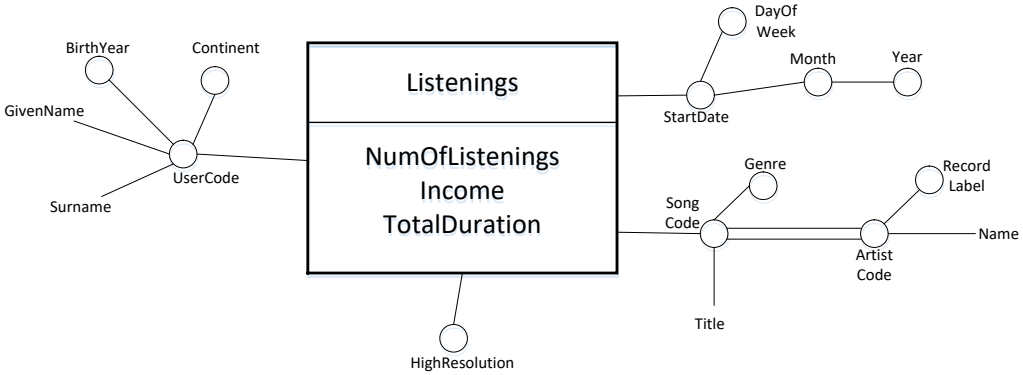
2. Conceptual design

Fact: Listenings (Listening entity)

2a) Attribute tree



2b) Fact schema



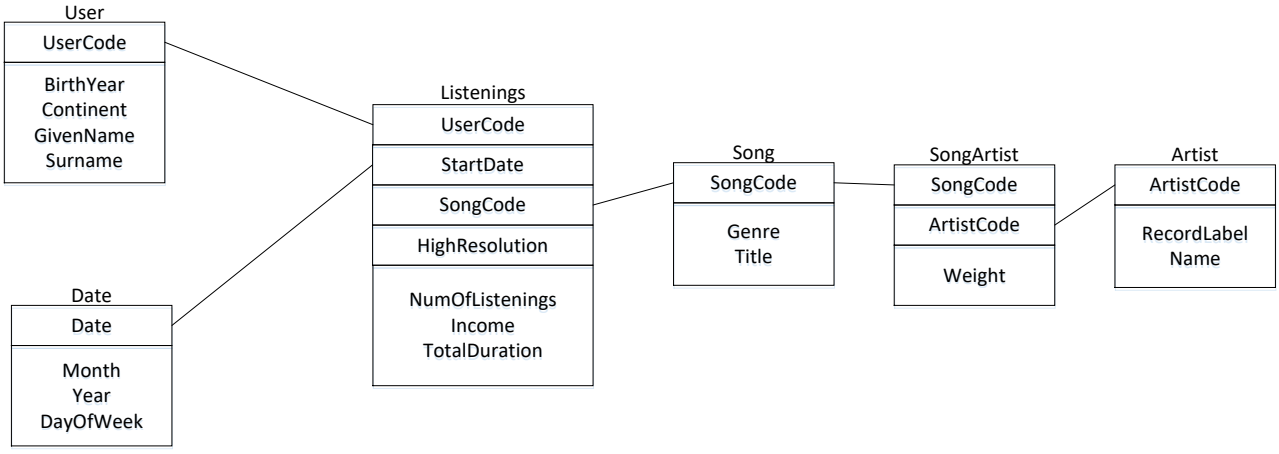
2c) Glossary

NumOfListenings
SELECT L.UserCode, L.StartDate, L.SongCode, L.HighResolution, **COUNT**(*)
FROM Listening **AS** L
GROUP BY L.UserCode, L.StartDate, L.SongCode, L.HighResolution

Income
SELECT L.UserCode, L.StartDate, L.SongCode, L.HighResolution, **SUM**(S.Price)
FROM Listening **AS** L, Song **AS** S
WHERE L.SongCode=S.SongCode
GROUP BY L.UserCode, L.StartDate, L.SongCode, L.HighResolution

TotalDuration
SELECT L.UserCode, L.StartDate, L.SongCode, L.HighResolution, **SUM**(S.DurationInSeconds)
FROM Listening **AS** L, Song **AS** S
WHERE L.SongCode=S.SongCode
GROUP BY L.UserCode, L.StartDate, L.SongCode, L.HighResolution

3. Logical design



4. Query answering

4a) Considering only the listenings performed on Sundays by users born in 1990, find the total income realized by the website (therefore excluding the quota for the artists) grouped by month, user (specify code, surname and given name), type of listening (high resolution or not) and genre of the song.

```
SELECT D.Month, U.UserCode, U.Surname, U.GivenName, L.HighResolution, S.Genre,
        SUM(0.6*L.Income)
FROM Listenings AS L, Date AS D, User AS U, Song AS S
WHERE L.StartDate=D.Date AND L.UserCode=U.UserCode AND L.SongCode=S.SongCode AND
        D.DayOfWeek='Sunday' AND U.BirthYear='1990'
GROUP BY D.Month, U.UserCode, U.Surname, U.GivenName, L.HighResolution, S.Genre
```

4b) For each artist, find the name and the number of listenings realized on 3rd July 2014.

```
SELECT A.ArtistCode, A.Name, SUM(L.NumOfListenings)
FROM Listening AS L, SongArtist AS SA, Artist AS A
WHERE L.SongCode=SA.SongCode AND SA.ArtistCode=A.ArtistCode AND L.StartDate='2014-07-03'
GROUP BY A.ArtistCode, A.Name
```

4c) Find the record label(s) whose artists obtained the greatest total income in 2015.

```
CREATE VIEW LabelIncome (RecordLabel, Income) AS (
        SELECT A.RecordLabel, SUM(0.4*SA.Weight*L.Income)
        FROM Listenings AS L, Date AS D, SongArtist AS SA, Artist AS A
        WHERE L.StartDate=D.Date AND L.SongCode=SA.SongCode AND SA.ArtistCode=A.ArtistCode AND
                D.Year='2015'
        GROUP BY A.RecordLabel
)

SELECT RecordLabel
FROM LabelIncome
WHERE Income = (
                        SELECT MAX(Income)
                        FROM LabelIncome
                )
```

4d) Select for each continent the code and the title of the song(s) with the greatest total listening time (note: all the listenings to a song can be considered as complete).

```
CREATE VIEW ContinentSongDuration (Continent, SongCode, Title, TotalDuration) AS (
        SELECT U.Continent, S.SongCode, S.Title, SUM(L.TotalDuration)
        FROM Listenings AS L, User AS U, Song AS S
        WHERE L.UserCode=U.UserCode AND L.SongCode=S.SongCode
        GROUP BY U.Continent, S.SongCode, S.Title
)

SELECT C.Continent, C.SongCode, C.Title
FROM ContinentSongDuration AS C
WHERE C.TotalDuration = (
                                SELECT MAX(C2.TotalDuration)
                                FROM ContinentSongDuration AS C2
                                WHERE C.Continent=C2.Continent
                        )
```