Technologies for Information Systems

Part I (10 points)

prof. L. Tanca – July 3rd, 2017

Available time: 25 minutes

Last Name _____

First Name _____

Student ID _____ **Signature**_____

1. Describe the Box-Plot method for representing statistical data, using an example to illustrate it clearly.

2. Describe the main differences between materialized and virtual data integration, explaining which of these two approaches is used in the case of Data Warehousing and why.

- **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 – July 3rd, 2017

Available time: 2h 00m

Last Name _____

First Name _____

Student ID _____ **Signature**_____

PoliMultiplex is a chain of multiplex movie theaters operating in Italy. In order to watch movies the customers must necessarily obtain a fidelity card, and each ticket issued by PoliMultiplex is associated with the customer that has purchased it and stored in the operational database of the company. The operational database contains also information about the theaters of the chain, the showings, and the programmed movies; each movie is also associated with the main actors that it features.
The management of PoliMultiplex has now hired you to design a data warehouse to analyze the issued tickets.
The following is the schema of the operational database used by PoliMultiplex:

CITY (CityName, Region)
CUSTOMER (CustomerId, Name, HomeCityName, BirthYear)
THEATER (TheaterId, TheaterName, CityName)
SHOWINGROOM (TheaterId, RoomNr, NrSeats)
MOVIE (MovieId, Title, Genre, DurationInMinutes, ProductionYear)
SHOWING (ShowingId, Date, Time, TheaterId, RoomNr, MovieId, Price) *// Each Showing may have a different price.*
ACTOR (ActorId, ActorName, Gender, BirthYear, HomeCountry) *// The attribute Gender may take the values 'M' or 'F'.*
STARRING (MovieId, ActorId)
TICKET (CustomerId, ShowingId)

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

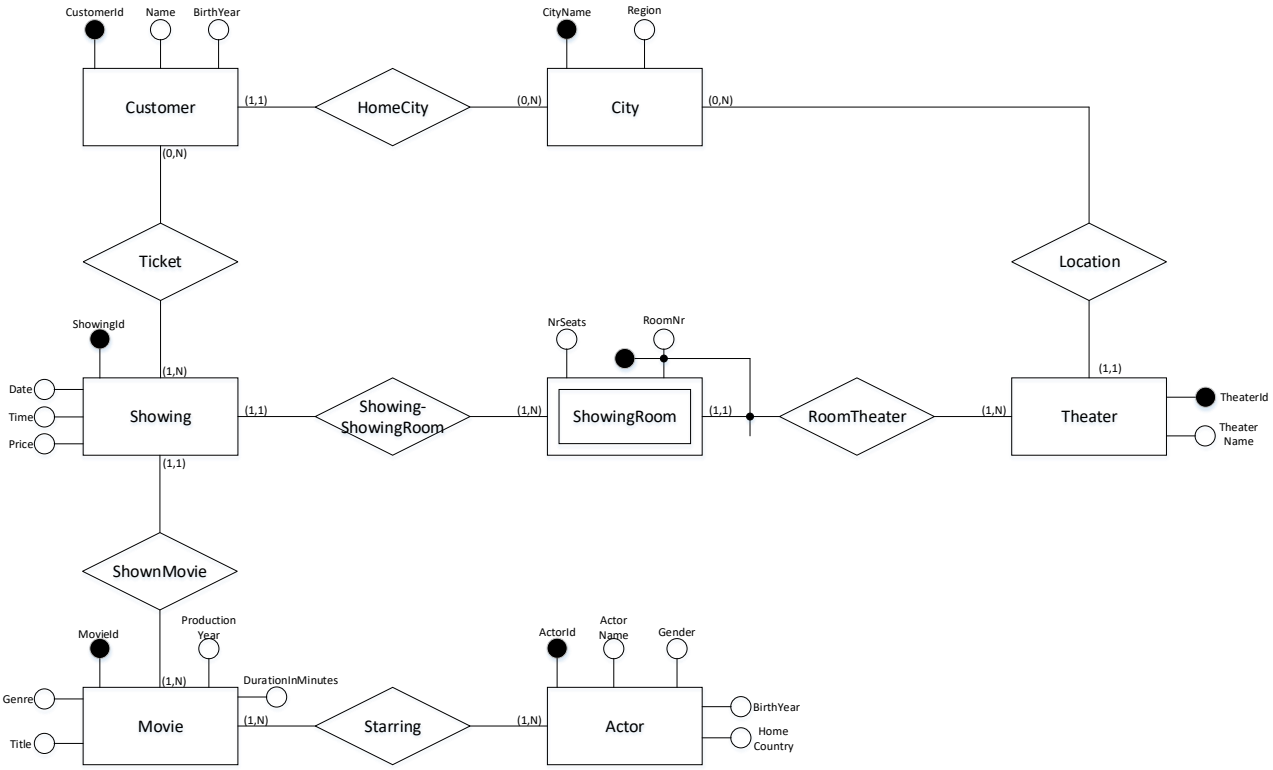
2. With respect to the produced ER diagram, discover the fact(s) that are useful specifically for answering the queries reported below. For each of these facts:
    a. (3 points) Produce the attribute tree (with pruning and grafting).
    b. (3 points) Produce the conceptual schema (fact schema).
    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 customers from Rome, aggregate the total income by date, month and year (include in the answer the aggregations computed only by date, only by month and only by year).
    b. (2 points) Compute the total number of tickets for each actor (specify id and name), customer home region, customer birth year and region of the theater.
    c. (2 points) Compute the total income realized by each movie (specify id and title), but considering only the movies starring at least one actress.
    d. (2 points) Considering only the theaters located in Milan, compute the genre(s) associated with the greatest number of tickets for each day of the week and theater (specify id and name).

SOLUTION

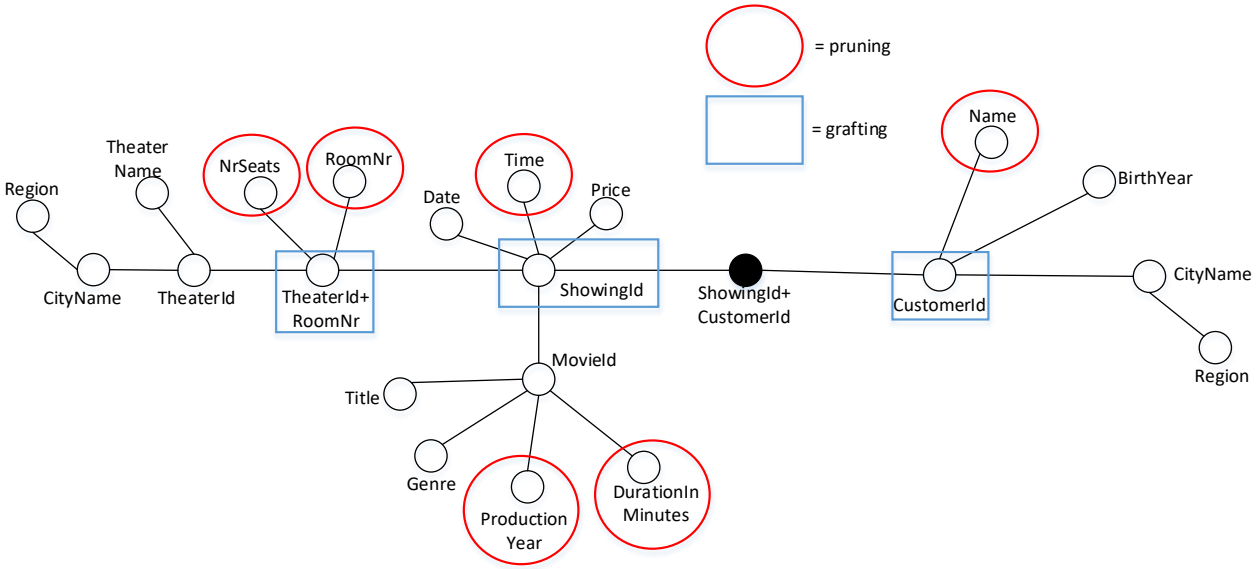
1. Reverse engineering



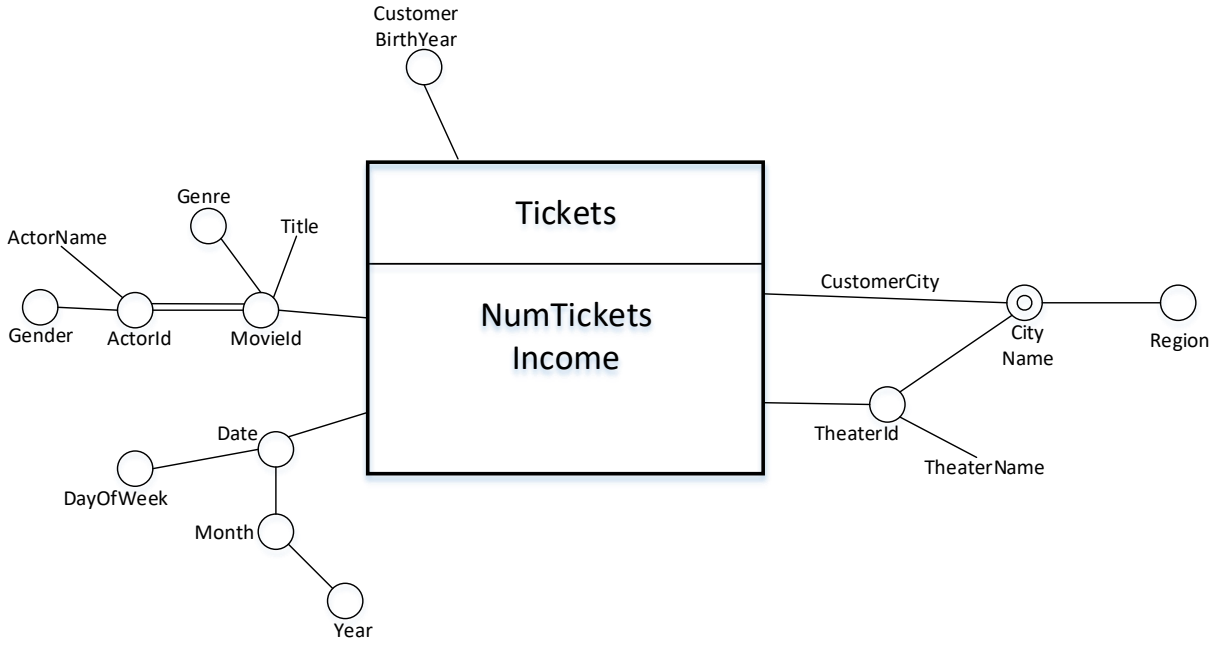
2. Conceptual design

Fact: Tickets (Ticket relationship)

2a) Attribute tree



2b) Fact schema



2c) Glossary

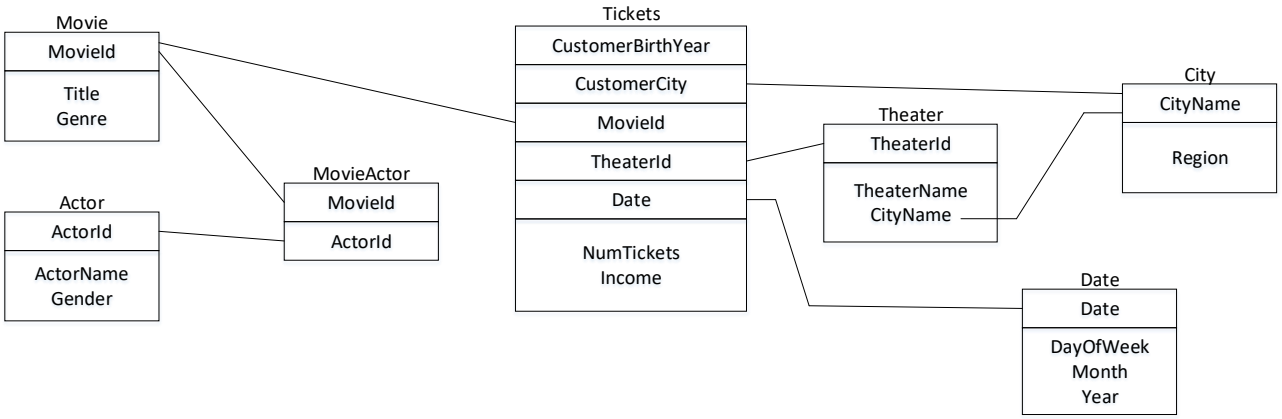
NumTickets

SELECT S.MovieId, S.Date, S.TheaterId, C.BirthYear, C.HomeCityName, **COUNT**(*)
FROM Ticket **AS** T, Showing **AS** S, Customer **AS** C
WHERE T.ShowingId=S.ShowingId **AND** T.CustomerId=C.CustomerId
GROUP BY S.MovieId, S.Date, S.TheaterId, C.BirthYear, C.HomeCityName

Income

SELECT S.MovieId, S.Date, S.TheaterId, C.BirthYear, C.HomeCityName, **SUM**(S.Price)
FROM Ticket **AS** T, Showing **AS** S, Customer **AS** C
WHERE T.ShowingId=S.ShowingId **AND** T.CustomerId=C.CustomerId
GROUP BY S.MovieId, S.Date, S.TheaterId, C.BirthYear, C.HomeCityName

3. Logical design



4. Query answering

4a) Considering only the customers from Rome, aggregate the total income by date, month and year (include in the answer the aggregations computed only by date, only by month and only by year).

```
SELECT D.Year, D.Month, D.Date, SUM(T.Income)
FROM Tickets AS T, Date AS D
WHERE T.Date=D.Date AND T.CustomerCity='Rome'
GROUP BY D.Year, D.Month, D.Date WITH ROLLUP
```

4b) Compute the total number of tickets for each actor (specify id and name), customer home region, customer birth year and region of the theater.

```
SELECT A.ActorId, A.ActorName, C-C.Region, T.CustomerBirthYear, C-T.Region, SUM(T.NumTickets)
FROM Tickets AS T, City AS C-C, Theater AS Th, City AS C-T, MovieActor AS MA, Actor AS A
WHERE T.CustomerCity=C-C.CityName AND T.TheaterId=Th.TheaterId AND Th.CityName=C-T.CityName
        AND T.MovieId=MA.MovieId AND MA.ActorId=A.ActorId
GROUP BY A.ActorId, A.ActorName, C-C.Region, T.CustomerBirthYear, C-T.Region
```

4c) Compute the total income realized by each movie (specify id and title), but considering only the movies starring at least one actress.

```
SELECT M.MovieId, M.Title, SUM(T.Income)
FROM Tickets AS T, Movie AS M
WHERE T.MovieId=M.MovieId AND M.MovieId IN (
                SELECT MA.MovieId
                FROM MovieActor AS MA, Actor AS A
                WHERE MA.ActorId=A.ActorId AND A.Gender='F'
                )
GROUP BY M.MovieId, M.Title
```

4d) Considering only the theaters located in Milan, compute the genre(s) associated with the greatest number of tickets for each day of the week and theater (specify id and name).

```
CREATE VIEW DowTheaterGenreTickets (DayOfWeek, TheaterId, TheaterName, Genre, NumTickets) AS (
        SELECT D.DayOfWeek, Th.TheaterId, Th.TheaterName, M.Genre, SUM(T.NumTickets)
        FROM Tickets AS T, Date AS D, Movie AS M, Theater AS Th
        WHERE T.Date=D.Date AND T.MovieId=M.MovieId AND T.TheaterId=Th.TheaterId AND
                Th.CityName='Milan'
        GROUP BY D.DayOfWeek, Th.TheaterId, Th.TheaterName, M.Genre
)

SELECT DayOfWeek, TheaterId, TheaterName, Genre
FROM DowGenreTickets AS D
WHERE D.NumTickets = (
                                SELECT MAX(D2.NumTickets)
                                FROM DowGenreTickets AS D2
                                WHERE D.DayOfWeek=D2.DayOfWeek AND D.TheaterId=D2.TheaterId
                        )
```