Database Management Systems

February 22^{nd} 2011

1. (6 Points) The following relations are given (primary keys are underlined):

   AUTHOR(AId, AName, City, BirthDate)
   PUBLISHER(PId, PName, Address, PCity)
   BOOK(ISBN, Title, AId, PId, Type, Price)
   SALES(ISBN, Date, SoldCopyNumber)

   Assume the following cardinalities:

   - card(AUTHOR)= 10^5 tuples,
     MIN(BirthDate) = 1-1-1949, MAX(BirthDate) = 31-12-1978,
     number of City $\simeq 10$,
   - card(PUBLISHER)= 10^3 tuples,
     number of PCity $\simeq 10$,
   - card(BOOK)= 10^7 tuples,
     MIN(Price) = 5, MAX(Price) = 44,
     number of Type $\simeq 20$,
   - card(SALES)= 10^9 tuples,
     MIN(Date) = 01-01-2010, MAX(Date) = 31-12-2010,

   Furthermore, assume the following reduction factor for the group by condition:

   - having sum(SoldCopyNumber)$\geq$1.000 $\simeq \frac{1}{10}$.

   Consider the following SQL query:

```
select B1.ISBN, B1.Title, A.Aname
from BOOK B1, AUTHOR A
where B1.AId=A.AId and A.City='Milan'
      and B1.Type<>'Historical novel'
      and B1.ISBN NOT IN (select S.ISBN
                          from SALES S, BOOK B
                          where S.ISBN=B.ISBN and B.Price > 40
                          and S.Date ≥ 01/04/2010 and S.Date ≤ 30/04/2010
                          group by S.ISBN
                          having sum(SoldCopyNumber)≥1.000)
```

   For the SQL query:

   (a) Report the corresponding algebraic expression and specify the cardinality of each node (representing an intermediate result or a leaf). If necessary, assume a data distribution. Also analyze the group by anticipation.

   (b) Select one or more secondary physical structures to increase query performance. Justify your choice and report the corresponding execution plan (join orders, access methods, etc.).

2. (7 Points) The following relations are given (primary keys are underlined, optional attributes are denoted with *):

```
JURYMAN(JuryManCode, Name, City)
JURYMAN_EVALUATION(JuryManCode, SongCode, Score)
TOTAL_SCORE_FOR_EACH_SONG(SongCode, NumberOfJurymen, TotalScore, IsWinner)
```

Write the triggers managing the following activities for a song competition. During the competition, 14 different songs are presented. These songs are evaluated by a jury, which votes the winning song.

(1) *Integrity constraint on the composition of the jury.* The `JURYMAN` table contains the composition of the jury. The jury must include at most 300 jurymen. All modification operations on the `JURYMAN` table violating the integrity constraint must not be executed. Write the trigger enforcing this integrity constraint.

(2) *Selection of the winning song.* Each juryman assigns a score to every song. For each song, the `TOTAL_SCORE_FOR_EACH_SONG` table contains the number of jurymen who have assigned a score to the song (attribute `NumberOfJurymen`) and the total score assigned by them (attribute `TotalScore`). The evaluation process is concluded when, for all 14 songs, all jurymen have assigned their score. At this point, the song with the highest total score is selected. This is the winning song. Write the trigger managing the following activities.

When a juryman assigns a score to a song (a new record is inserted in `JURYMAN_EVALUATION`), the trigger propagates the modifications on the `JURYMAN_EVALUATION` table to the `TOTAL_SCORE_FOR_EACH_SONG` table. Then, the trigger checks if the evaluation process is concluded. In this case, for the winning song, the `IsWinner` attribute is set to 'Yes'.

3. Data Warehouse design

Problem specifications

A market analysis company is interested in analyzing the activities of social network users.

The company wants to evaluate some statistics about registered users (both newly registered and total users), and some specific activities. In particular, the activities of interest are the addition of another user to the own contact list (friends or followers).

The company has been allowed access to the databases of the most popular social networking sites. These databases should be integrated in a data warehouse that allows the following analyses to be efficiently performed.

The company would like to analyze the number of total users, the number of newly registered users and the average number of contacts (friends or followers) for each user, according to:

- the day of the year (1-366), the day of the week (Monday-Sunday), the day of the month (1-31), working days or holidays, week of the year (1-52);
- the date, the month, the 2-month period, the trimester and the year;
- the social networking site, the year of its foundation, the nation in which the website was founded;
- the birth year, the gender and the home nation of the users.

The data warehouse will store information about years 2006-2010. Moreover, the following statistics are known (the candidate may estimate missing information that she deems relevant):

- 20 social networking sites founded in 10 different nations between 2004 and 2006 are considered;
- the users belong to roughly 200 different nations and are aged between 15 and 65 years;
- unsubscriptions are not considered, thus a user can stop using a website but she can not cancel her subscription (the total number of registered users is monotonically increasing);

The following are some of the frequent analyses the company is interested in:

(a) for each social network, select the monthly total number of newly registered users and the monthly growth percentage of the social network. The monthly growth percentage is defined as the ratio between the new users registered in the considered month and the total number of subscribed users in the considered month.

(b) For each social network and for each month, select the daily average of new users, separately for female and male users.

(c) Select the total number of newly registered users for each day of the week and the percentage of newly registered users for each day of the week respect with the total number of newly registered users, separately for female and male users. Assign a decreasing rank to the number of newly registered users for each pair (gender, day of the week).

Design

(a) (7 points) Design the data warehouse to address the described issues. In particular, the designed data warehouse must allow efficient execution of all the queries described in the specifications.

(b) (4 points) Write frequent query (b) of the "problem specifications" using the extended SQL language.

(c) (*Optional:* 5 points) Write frequent query (a) of the "problem specifications" using the extended SQL language.