Data Management for Big Data

February 1, 2021

2019–2020, 1st winter session
Teachers: Angelo Montanari, Dario Della Monica, Andrea Brunello

Surname and name: _____

Student ID (matricola): _____ email: _____

Each part of the exam will contribute equally (one third) to determine the final score. Thus, total score of each part might be normalized so that you get at most 10 points from each part. Teachers can assign extra bonus points at their own discretion.

Be careful: use a neat handwriting. It is particularly important in this emergency situation, since the test will be scanned and emailed to the teachers.

Part I: Fundamentals of database systems

Exercise 1:

Let us consider the following relational schema about skiers and ski races:
$SKIER(CodS, Name, Surname, Nationality, BirthYear)$;
$RACE(CodG, Place, Date, Discipline)$;
$PARTECIPATION(Race, Skier, Position)$.

Let us assume each skier to be univocally identified by a code, and characterized by a name, a surname, a nationality, and a birth year. We also assume each race to be univocally identified by a code, and characterized by a place, a date, and a discipline (giant slalom, downhill, ..). We assume that more than one race of the same discipline may take place at the same place, but at different dates, and that more than one race may take place at the same place and date, but of different disciplines. Finally, let us assume that all the skiers that participate in a race have been classified, that is, got a position. We admit the possibility of ex aequo in the final ranking of a race.

Define preliminarily primary keys, other candidate keys (if any), and foreign keys (if any). Then, formulate an SQL query to compute the following data (exploiting aggregate functions only if they are strictly necessary):

- the ski races with a German skier among the top 3 and devoid of ex aequo.

Exercise 2:

Let us sinthesize the ER conceptual schema of a database for the testing of its prototypes by a car manufacturer.

- Each prototype is univocally identified by a code and it is characterized by a name, a cost of production, and a development time (measured in months). The company may run one or more tests on every prototype.
- Each test involves one and only one prototype, and it is characterized by a code, that univocally identified the test among those run on the prototype (we do not exclude the possibility of associating the same code to different tests on different prototypes), the name of the test driver, the execution date, the start time (hour), and the duration of the test (measured in minutes)
- We distinguish between two types of testing: track testing and bench testing. For each track testing, we record the track where it has been executed and the external temperature during the test. For each bench testing, we record the used test bench. In addition, each bench test is characterized by a numerical code, that univocally identifies it among the tests executed on that bench.
- Each test bench is univocally identified by a code and it is characterized by a degree of quality. Moreover, for each test bench, we record the countries where it has been approved (no one, one, more than one). Each country is univocally identified by its name, and it is characterized by its capital and by the address and the phone number of the office of reference for the approval of the test benches.

Build an ER schema that describes the above requirements, clearly explaining any assumption you made. In particular, for each entity, identify its possible keys, and carefully specify the constraints associated with each relation.

Part II: Advanced database models, languages, and systems

Instructions for multiple-choice questions.

- A right answer is worth +1.
- A wrong answer is worth -0.33.
- It is possible to give a short explanations for multiple-choice questions. It should be **very** short (no more than 2 lines) and should be written in a neat handwriting. They are optional and used **at teacher's discretion**, mainly to increase the score of a wrong answer; seldom, to lower the score of a right one.

Instructions for open questions.

- The score assigned to an open question can range from 0 to 3 points.
- No negative score is assigned to open questions.
- Be careful: use a neat handwriting.
- Try to be succinct also for open questions.

1. Let t_S = "time for one seek", t_T = "time for one-block transfer", and h be the height of a secondary index (a tree) over attribute A of relation R. Which is the estimated cost for accessing all tuples where $A = X$? (Assume that A is a key.)

   ☐ $h * (t_T + t_S) + b * (t_S + t_T)$

   ☐ $h * (t_T + t_S) + t_S + b * t_T$

   ☐ $h * (t_T + t_S) + t_S + t_T$

   Short explanation (optional): _____

   _____

   Hint: recall that
   - *a primary index is defined over the attribute(s) used to physically order the file in the filesystem;*
   - *a secondary index is defined over any (subset) of the other attributes.*

2. Consider a relational algebra expression of the form

   $$\sigma_\tau(R_1) \bowtie R_2,$$

   i.e., where a *selection* occurs as sub-expression of a *natural join*. Briefly explain why the number of tuples of R_1 matching the condition τ of the *selection* affects the overall execution time of the whole expression and clarify whether or not (argument your answer) it is always more efficient to execute selections before joins.

   _____

   _____

   _____

   _____

   _____

3. If a relation R is fragmented, according to vertical fragmentation, into $\{R_1, \ldots, R_n\}$, then we have:

   ☐ $R = R_1 \bigcup R_2 \bigcup \ldots \bigcup R_n$

   ☐ $R = R_1 \bowtie R_2 \bowtie \ldots \bowtie R_n$

   ☐ $R = R_1 \ltimes R_2 \ltimes \ldots \ltimes R_n$

   Short explanation (optional): _____

4. Consider the 2 transactions T_1 (over operations $R_1(y), W_1(y), R_1(x), W_1(x)$) and T_2 (over operations $R_2(x), W_2(y), W_2(x)$) formalized through the 2 following partial orders, respectively:
   $T_1 = \{W_1(x) \prec R_1(x), R_1(x) \prec R_1(y), W_1(x) \prec W_1(y), W_1(y) \prec R_1(y)\}$
   $T_2 = \{W_2(x) \prec R_2(x), W_2(y) \prec R_2(x)\}$.
   Is there a history over $\{T_1, T_2\}$ that is serializable but not serial? If yes, write down one such history. If not, write down a history that is both serializable and serial.

   Is there a history over $\{T_1, T_2\}$ that is serial but not serializable? If yes, write down one such history. If not, write down a history that is neither serializable nor serial.

   Hint: recall that serial histories are always concurrent transaction executions, that is, they are formalized through linear orders.

Part III: Data analysis and big data

- Describe the differences between *descriptive*, *predictive* and *prescriptive* analytics, providing also an example for each case.
- Name two common operations that can be performed over an OLAP cube, and provide some examples.
- In the context of NoSQL, briefly describe the BASE properties.
- In the context of text indexing in text analytics, briefly describe *stemming*. Provide an example showing the usefulness of such an operation.
- Briefly describe the decomposition of time series using the *multiplicative model*. When should it be preferred with respect to an *additive model*?