Data Management for Big Data

January 27, 2020

2018–2019, 2nd winter session
Teachers: Angelo Montanari, Dario Della Monica, and Paolo Gallo

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.

Part I: Fundamentals of database systems

Exercise 1:

Let us consider the following relational schema about actors and films:
$FILM(FilmCode, Title, Filmmaker, Year)$;
$ACTOR(ActorCode, Surname, Name, Sex, BirthDate, Nationality)$;
$INTERPREATATION(Film, Actor, Role)$

Let every film be univocally identified by a code and characterized by a title, a filmmaker, and the year when it has been released. For the sake of simplicity, let us assume each filmmaker to be univocally identified by his/her surname and each film to be directed by a single filmmaker. Let us consider the possibility that distinct films have the same title (this is the case, for instance, with remakes), but exclude the possibility for two films with the same title to be released the same year. Let every actor be univocally identified by a code and characterized by a name, a surname, a sex, a birth date, and a nationality. Let us assume that more than one actor may act in a film and that the same actor may act in more than one film. Finally, let us assume that, in a given film, each involved actor plays only one role.

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 actors that acted together in at most one film directed by the filmmaker Quentin Tarantino.

Exercise 2:

Let us sinthesize the ER conceptual schema of a database of professional training courses organized by a given company for its employees on the basis of the following set of requirements.

- Each course is characterized by a name and by a (possibly empty) set of preparatory courses. Each course is univocally identified by its name. A given course can be attended only if all its preparatory courses have been already attended.
- The company may offer more than one edition of each course at different times and venues. Each edition of a course is characterized by a starting date, an ending date, a duration, a venue, a set of teachers (one or more), and a set of attendees (employees of the company), and it is univocally identified by its starting date and venue, that is, different editions of a course may start at the same date, but at different venues, or may be organized at the same venue, but starting at different dates. Each teacher is characterized by a set of skills, a forwarding address, and one or more phone numbers. Each attendee is characterized by an initial level of readiness.
- For each employee of the company, we record his/her personal data, task, education qualification, and previous work experiences. For each previous work experience, we keep track of the type of work done and of its duration.

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.
- Short explanations 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.
- You can use additional sheets 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 primary 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 not a key and there are n tuples satisfying $A = X$, spread over b blocks.)

   ☐ $(h + n) * (t_T + t_S)$

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

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

   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. Select the correct statement.

   ☐ Distributed DB Systems have become a necessity for several reasons (independently from their benefits and drawbacks)

   ☐ Distributed DB Systems have been chosen by most companies because they are better than Centralized DB Systems in every respect

   ☐ Distributed DB Systems have become mandatory due to regulations on data privacy and security

   Short explanation (optional): _____

   _____

3. Which sentence matches better the notion of transparency?

- ☐ Transparency means that it is known to the user where (which node of the Distributed DB System) data is stored and where queries are executed
- ☐ Transparency concerns the ability of the user to choose the best query execution plan to execute a query
- ☐ Transparency is about the separation between the higher level (semantics) of a system and the lower level (implementation)

Short explanation (optional): _____

_____

4. Which are the three properties certifying fragmentation correctness? Add a very short explanation for each of them.

_____

_____

_____

_____

_____

_____

_____

_____

_____

5. Let R be a relation whose primary key is the attribute *key*, M be a partition of attributes of R, and S be a relation such that there is a link L with *owner(L)* $= R$ and *member(L)* $= S$. In other word, one of the attributes of S is a foreign key referring to attribute *key* of relation R.

   Which is the vertical fragmentation over relation R induced by M?

   - ☐ $\{R_i \mid R_i = \sigma_{m_i}(R), m_i \in M\}$
   - ☐ $\{R_i \mid R_i = \Pi_{m_i \cup \{key\}}(R), m_i \in M\}$
   - ☐ $\{R_i \mid R_i = R \ltimes S_{m_i}, m_i \in M\}$

   Short explanation (optional): _____

   _____

6. Consider the 2 transactions T_1 (over operations $W_1(x), W_1(y)$) and T_2 (over operations $R_2(y), R_2(x), W_2(x)$) formalized through the 2 following partial orders, respectively:

   $$T_1 = \{W_1(y) \prec W_1(x)\}$$
   $$T_2 = \{R_2(x) \prec W_2(x), R_2(x) \prec R_2(y)\}.$$

   Is there a history over $\{T_1, T_2\}$ that is serializable but not serial? If yes, write down one such history.

   Is there a history over $\{T_1, T_2\}$ that is serial but not serializable? If yes, write down one such history.

Is there a history over $\{T_1, T_2\}$ that is neither serializable nor serial? If yes, write down one such history.

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

7. Explain the difference between checking that an XML document is well-formed and validating it.

Hint: which is the difference between an XML parser and an XML validator?

Part III: Data analysis and big data

1. Point out some differences between Data Lake and Data Warehouse.

2. Briefly describe the main characteristics of MapReduce.

3. Briefly describe what is a *column database* and some fields of applications.

4. Briefly introduce the tiers that compose a modern general data warehouse architecture.

Hint: There are four tiers to manage the data from different sources. Name each of them and the task performed.

5. Consider a time series from a single sensor which is transmitting a pressure value every minute. Using a document database you can store a single JSON document for each reading:

```
{
  pressID: "I8008",
  pressure: 1024,
  ts: ISODate("2018-11-10T22:56:00.00-0500")
 }
```

alternatively, you can store a single JSON document containing a nested object to store one hour of observations and update it as soon as a new measure comes (every minute):

```
{
 pressID: "I8008",
 pressure: {0:1003, 1:1023, 2:1012, ..., 59:1002},
 ts: ISODate("2018-11-10T22:00:00.00-0500")
}
```

For each of the two models above write down:

- How many document writes will typically occur in **one hour**?
  First:_____ Second: _____
- How many documents updates will typically occur in **one hour**?
  First:_____ Second: _____
- Suppose that each document will include a 100 bytes long field for indexing purposes, what will be the space used only by those fields for storing an **entire day of readings**?
  First:_____ Second: _____

6. Briefly describe what is the stemming process in the context of text mining.

7. Explain why dbRef(s) (also named references) across documents in MongoDB differ from foreign keys used by relational databases.

Hint: Please keep in mind that some properties are guaranteed by default by relational databases.