Data Management for Big Data

June 17, 2018

2018–2018, 1st summer 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 films and actors:

$FILM(FilmCode, Title, Filmmaker, Year)$;

$ACTOR(ActorCode, Surname, Name, Sex, BirthDate, Nationality)$;

$INTERPREATATION(Film, Actor, Character)$

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 film to be directed by a single filmmaker and each filmmaker to be univocally identified by his/her surname. Let us consider the possibility that distinct films have the same title (this is the case, for instance, with remakes). 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. For the sake of simplicity, let us assume that, in a given film, each actor may play one role only.

Preliminarily define primary keys, other candidate keys (if any), and foreign keys (if any). Then, formulate an SQL query to compute the following data (without using the operator CONTAINS and exploiting aggregate functions only if they are strictly necessary):

- the set of actors that act in all films directed by the filmmaker Loach.

Exercise 2: Let us sinthesize an ER conceptual schema of a database for the management of data about the employees of a given company, and their abilities, the projects they work on, and the departments they belong to on the basis of the following set of requirements.

- Each employee has a code, that univocally identifies him/her, which is assigned by the company. We would like to keep track of the name, the surname, the birth date, and the recruitment date of each employee. If an employee is married with another employee of the company, we would like to record the spouse and the wedding date. Each employee has a qualification, e.g., administrative staff, commercial staff, programmer, analyst, designer, and so one.
- For all employees with a higher-education qualification (degree and, possibly, PhD), we would like to know the degree class (computer science, mathematics, physics, etc.) and the graduation date, and, in case, the PhD class (computer science, mathematics, physics, etc.) and the date at which they defended their dissertation. For the sake of simplicity, we assume that (i) each employee has at most one degree, (ii) each employee has at most one doctorate, and (iii) to get a PhD, an employee must have a degree (the vice versa is obviously not true: there are employees with a degree, but no doctorate).
- The work of the company is structured in a number of projects. Each project is identified by a code, defined by the company, and it is characterized by a budget and a duration (number of months). Each project involves one or more employees. Each employee has a number of abilities (one or more) and he/she works on one or more projects. In each project he/she works on, an employee exploits one or more of his/her abilities, not necessarily all of them. We would like to record the competences that an employee uses in each project he/she works on.
- The company has a number of departments. Each department is identified by a name and it is characterized by a phone number and an email address. Each employee belongs to a single department. Each department sources its goods from various suppliers and each supplier may supply more than one department. We would like to record the name and the address of each supplier. Moreover, we would like to keep track of the last purchase of each departement (date and supplier).

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.

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 5 tuples satisfying $A = X$, all of them being stored in the same block.)

   □ $h * (t_T + t_S) + t_S + t_T$

   □ $h * (t_T + t_S) + t_S + 5 * t_T$

   □ $h * (t_T + t_S) + 5 * t_S + 5 * t_T$

   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. What is a Query Execution Plan and which is a convenient way to formally represent it?

   ____

   ____

   ____

   ____

   ____

   Hint: how does a Query Execution Plan relates to the query (in relational algebra) and its representation?

3. Select the statement that better describes the query optimization process.

□ The query optimization process estimates costs (execution time) for different query execution plans and chooses the fastest

□ The query optimization process aims at producing a result (a query) that minimizes space consumption

□ The query optimization process always executes *selections* and *projections* before *joins*

Short explanation (optional): ______________________________

____

4. 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): ______________________________

____

5. 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): ______________________________

____

6. Explain the notion of fragmentation in the context of Distributed DB Systems, by mentioning (and briefly explaining) the different kinds of fragmentation you know of. (You can use pictures if you think it would help.)

____

____

____

____

____

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. With respect to the Data Warehouse context, briefly define the ETL process phase.
2. List two common operations that can be performed over an OLAP cube.
3. List the four main families of NoSQL databases.
4. Briefly explain the difference between *replication* and *sharding*.
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 JSON document containing a nested object ready 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?
   - How many documents updates will typically occur in one hour?
   - 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?

6. Give an example of a computation that can be performed using the MapReduce paradigm on data split across HDFS nodes (not the code, only the computation).
7. Explain why dbRef(s) (also named references) across documents in MongoDB differ from foreign keys used by relational databases.