Data Management for Big Data

July 10, 2020

2019–2020, 2nd summer 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 departments and physicians:

$DEPARTMENT(Name, Building, Floor, Chief)$;

$PHYSICIAN(PhysicianId, Name, Surname, Specialization, Gender, BirthDate, Department, HomeCity)$;

$BELONGS\_TO(City, Region)$.

Let every department be univocally identified by a name and characterized by its location (building and floor) and chief. Let us assume that a physician can be the chief of at most one department (the department he/she belongs to). We do not exclude the possibility for two distinct departments to be located at the same floor of the same building.

Let every physician be univocally identified by a code and characterized by a name, a surname, a specialization (we assume to record exactly one specialization for each physician), a gender, a birth date, a department (each physician is assigned to one and only one department), and a home city.

Let every city be univocally identified by its name and characterized by the region it belongs to.

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 departments such that (i) all their physicians reside in the region *Piemonte* and (ii) at least one of them resides in the city *Torino*.

Exercise 2: Let us sinthesize the ER conceptual schema of a database for the management of the study program of a university on the basis of the following set of requirements.

- Each course is univocally identified by its numerical code and characterized by a name, a set of prerequisites (a set of other courses that must be alredy passed in order to take the exam), and the professor that teaches it (each course is taught by one professor only).
- Each student is univocally identified by an ID number and characterized by a first name, a last name, an email, an address, and a birth date. Each student is enrolled in one or more courses and has passed a number of them (possibly no one yet). For each course that the student has passed, we record the date of the exam and the obtained grade.
- Professors are partitioned in full and associate professors. Each professor is univocally identified by his/her fiscal code and characterized by a first name, a last name, a mobile number, the department to which he/she belongs, and the courses that he/she teaches (one or more). Each department is univocally identified by its name and characterized by an address. For each full professor, we record the day he/she was named full professor.

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 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 a key and there are n tuples satisfying $A > X$, spread over b blocks.)

   ☐ $(h + n) * (t_T + t_S)$

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

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

   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.

   ☐ The Catalog stores results of queries that are executed frequently

   ☐ The Catalog stores indices to be used to optimize query executions

   ☐ The Catalog stores statistical information useful for cost estimations

3. Consider a Distributed DB System over 2 sites, Trieste and Dublin, and consider the frequently executed query

   SELECT `name`, `surname`, `location`
   FROM PROJECTS, EMPLOYEES
   WHERE `title` = `p_title`

over relation schemas

PROJECTS(<u>`title`</u>, `location`, `budget`)
EMPLOYEES(<u>`code`</u>, `name`, `surname`, `p_title`)
foreign key: `p_title` REFERENCES PROJECTS(<u>`title`</u>)

(Attribute `location` can only assume "Trieste" or "Dublin" as values.)

Define a suitable fragmentation strategy (that fragments one or both relations) for efficient executions of the above query. In particular, which kinds of fragmentation (primary horizontal, vertical, derived horizontal) are employed and which fragments are produced? Argument briefly your choice.

4. Consider the 2 transactions T_1 (over operations $R_1(y), R_1(x), W_1(x)$) and T_2 (over operations $W_2(x), W_2(y)$) formalized through the 2 following partial orders, respectively:

$$T_1 = \{R_1(x) \prec W_1(x), R_1(y) \prec W_1(x)\}$$
$$T_2 = \{W_2(y) \prec W_2(x)\}.$$

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

1. Explain the main differences between OLTP and OLAP
2. Name the *3 Vs of Big Data*, and briefly explain one of them
3. In the context of text analytics, what is the benefit of relying on *n-grams* with respect to single words?
4. Which of the time series in Figure 1 would you decompose using an *additive* model? And which using a *multiplicative* model? Why?

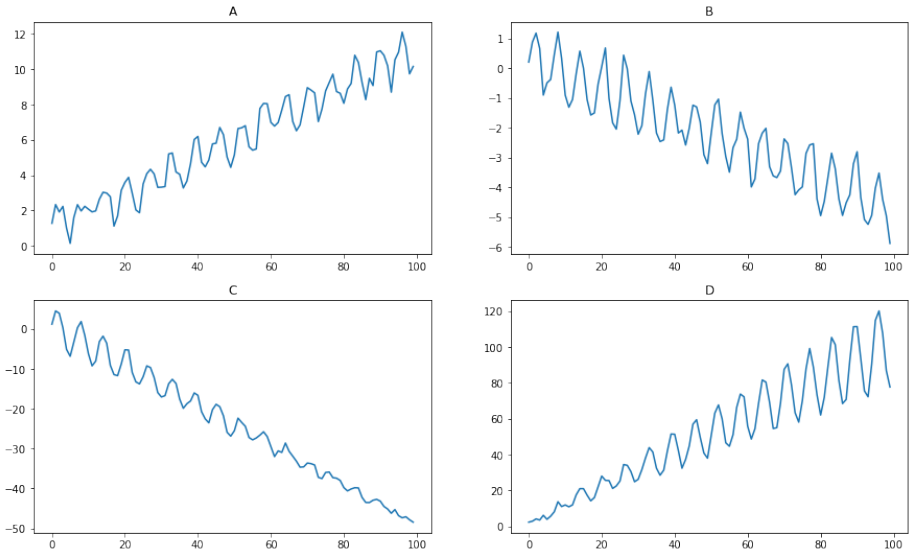

Figure 1: Time series

5. What are the advantages of using Apache Pig, with respect to plain MapReduce?