Business Data Management
Term Project

Instructor: Periklis A. Papakonstantinou (periklis.research@gmail.com)
Teaching assistant: Nikolas Melissaris-Papanikolaou

Final Report Due date: Wednesday, May 1st at 11.59pm.

Presentation dates: the week of April 22 – 26 (during class time).

The project has two main parts. (1) **50%** of the project grade is to design from scratch a web-based database application. (2) **50%** of the project grade is to choose an advanced topic (from a provided list) and make a slides-based 5-8 minutes long presentation + at most 2-3 minutes for question answering. You should decide and notify the instructor in advance: who among the team members will deliver the presentation and who will be responsible for answering questions.

The day of your presentation: you must send at least 48 hours in advance your presentation to the instructor (periklis.research@gmail.com) in order to be eligible to present.

PART 1

Here is a list of presentation topics. You should email your preferences to the instructor by April 2nd at 1.30pm (one email for each group). For example, if in your email you list 3,2,1,4,5,6,7,9 this means that topic # 3 has the highest priority in the preference of your group, then it is topic # 2 and so on. **Preferences will be honored in a first-come-first served order. The time is determined by the timestamp the instructor received each email.**

Presentations

List of presentation topics: please, use this numbering when stating your group's preference list.

1. What is MongoDB and how does it compare to relational databases? Give one or more concrete examples. **You must provide an example that contasts this technology with what you learned in this class.** Explain the advantage of using MongoDB. Conclude by explaining in which situation/situations you would prefer using an RDBMS instead of MongoDB.

2. Overview of what is query planning and query optimization. How is query optimization done. Give a concrete example. **During your presentation you must explain how is this material related to what you learned in this class.**

3. What is OLAP and how does it compare to relational databases? Give one or more concrete examples. Explain the advantage of using OLAP. **You must provide an example that contasts this technology with what you learned in this class.** Conclude by explaining the limitations of OLAP and what is in the modern times the system that potentially replaces it.

4. CGI is an "ancient" technology. Is CGI obsolete? What are the possible alternatives to CGI? At the very least you should explain what is what is FastCGI, what is PHP, what is the difference between PHP and CGI. You should also explain clearly what is Javascript, Java Servlets, JSP, what exactly are their differences and emphasize on explaining server-side and browser-side applications. **You must provide examples that contast these technologies with what you learned in this class.**

5. Study the (high-level) differences between the following SQL servers: ORACLE, MS SQL server, MySQL, DB2, MariaDB. For each of them give an example where one would have been preferred instead of all the others. **You must present things in a way that the features of these products are related to what you learned in this class.**

6. Read, understand in depth, and present the research paper "How to Win a Hot Dog Eating Contest: Distributed Incremental View Maintenance with Batch Updates" (`https://infoscience.epfl.ch/record/218203/files/sigmod2016-cr.pdf`). This topic is a little more demanding that the rest because in addition to this paper who should find and read the related work (it is impossible for a student to access this content by just reading the 6 pages of this paper). **During your presentation you must explain how is this material related to what you learned in this class.**

7. Overview of distributed RDBMSs: what is the main problem in distributed DBMSs? What is two and three phase commit? List the most popular concurrency control mechanisms. Give an example or two. **During your presentation you must explain how is this material related to what you learned in this class.**

8. What is the programming language Ruby? Give examples comparing it with Python. How is Ruby used in Web applications? Give a concrete example. We use Ruby instead of what? What is Ruby, Rails, and their relationship? **You must provide**

an example that explains how is Ruby related to what you learned in this class.

9. Read, understand in depth, and present the research paper "Squall: Scalable Real-time Analytics" (`http://www.vldb.org/pvldb/vol9/p1553-vitorovic.pdf`). This topic is a little more demanding that the rest because in addition to this paper who should find and read the related work (it is impossible for a student to access this content by just reading the 6 pages of this paper). **During your presentation you must explain how is this material related to what you learned in this class.**

10. Overview of Differential Privacy in statistical databases. What is the purpose of statistical databases. Give an example of how differential privacy is achieved in statistical databases. Which products of Microsoft, Apple, and Google are differentially private. **You must give an example explaining how is this material related to what learned in class.**

11. Using UML (Unified Modeling Language) instead of ER. Briefly explain what is UML, what are the commonalities and differences of UML and ER diagrams. Give at least one illustrative example. **During your presentation you must explain how is this material related to what you learned in this class.**

PART 2

General description

Go through the typical phases for building a database system for a small boutique hotel business. This hotel hosts only two types of customers (visitors): (i) individual customers and (ii) customers that belong to groups of people for moderate-scale events (e.g. reunions, small-size weddings etc). Each hotel room is characterized by its type/size (e.g. single occupancy, double, queen or king size bed and number of beds, and other specific to the boutique hotel – improvise!), where is it located (floor and number), various prices and for various seasons and allowed discounts, other amenities (improvise!), room photos, and possibly other characteristics. The customers are identified by their name, ID (various types), age, and general preferences about room types. An event is characterized by the event name, the event type (fix some types), approximate number of participants, payment details, various notes about the event, whether any special room is required to host the event and details about the arrangements needed for this room, and anything else you can add. An event has one customer that is billed for it. An event is related to customers that participate in this event. There are two types of customers. The primary ones, i.e. those that are responsible for the occupancy of the room, responsible for the bill etc, and non-primary, i.e. those who are occupying the room together with the primary one (i.e. these are dependents). A primary customer is related to a room, together with the rate she/he got, payment method, dates of occupancy, etc. Every person that stays in a room must appear as a customer (although in reality this doesn't happen), although only the primary is responsible for the room. Obvious constraints apply. At minimum you should consider the fact that one room cannot be occupied by two different primary customers on the same date. Same thing for event rooms.

Simplifications: For simplicity we allow events to be attended only by hotel customers. Also, you should not consider other hotel amenities, such as room service, restaurants, parking etc. Also, you should only consider simplified payment methods (e.g. do not consider holds on credit cards, special payment methods when big lump sums are required etc). Finally, we significantly simplify things by not considering in the database the employees of the hotel.

Constraints you should be adding: You should model all constraints and you should write down in plain language all related assumptions you made. The first tasks you have to do, is to devise any constraints that make the above description complete. All your data modeling must incorporate those constraints.

Part 1 (20 points)

1. Describe in at most one page (preferably less than half a page) any assumptions and constraints you made. To get full marks, in addition to any constraints mentioned in "General description" you should describe at least one new constraint in addition to the constraints specified above. All key constraints should be determined in the ER below.
2. Provide a complete ER diagram that models the "General description" and includes all constraints and assumptions you wrote in (1) above.

Part 2 (20 points)

Implement the ER diagram as a Relational Model. Specify the tables you are using together with the domains (datatypes).

Remark: the relational model is not going to encode all of the constraints of the ER model.

Realize the above database and express the following query in Relational Algebra and in SQL:

Find the customer that did the maximum number of reservations in 2019.

When you implement the database in SQL specify the tables together with all constraints.

What to report (same for (i) and (ii)) The Relational Algebra query. The schema of your relational DB, the SQL queries that create the relations (**careful to include all ICs**). We cannot provide data for DB tables because part of the question is you to design the DB schema (i.e. not everyone will have the same tables). Therefore, you should make up entries to your database and list the contents of the relation instances (tables) in a similar way that they were listed at the end of the Lab Exercise you did a few weeks ago. Make sure that the entries you make are indicative for the SQL queries you should present. Finally, you should report the SQL output on the example entries you made.

Part 3 (10 points)

Integrate with Apache and CGI. The interface should be in HTML (very basic HTML is okay for getting full marks). For the CGI you can use Python or any other programming language you feel comfortable with.

Make three simple HTML pages where the user can add customers, rooms, and room reservations. Also make a forth HTML page where the user can search for reservations based on any feature you want (make it as simple as you want it to be – searching using a single feature gets full). The results of the queries should be returned in tabular form (no need to use the specifics for HTML table – the result should be human readable).