

Business Data Management

Term Project

Instructor: Periklis A. Papakonstantinou (periklis.research@gmail.com)

Teaching assistant: Hafiz Asif, Jithesh

Due date: May 2nd, 11:59pm

The project consists of four parts and also has an additional question, which is of general interest.

Grading: this project will be marked out of 100 points. However, the sum of marks is 110. This means that you can get up to 10 bonus points. Note that by skipping the simple Data-Base Administration task your total marks sum up to 95.

Focus: light project in designing and implementing a database. Much of the work is in studying and understanding how the today's "hot" web-services integrate with SQL servers.

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 2 constraints more (in addition) to the ones 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 queries in Relational Algebra:

- Find the customer (or customers) who paid the highest room rate in 2017 and is also related to at least one more non-primary customer.

- Find the room (or rooms) that was booked for the most consecutive number of days in a single booking by a customer in 2017.

Implement the database in SQL. Specify the tables together with all constraints. Realize the above two queries in SQL. In addition, you should realize the following query:

- For every event type find the name of the group that had the biggest number of participants.

What to report (same for (i) and (ii)) The Relational Algebra queries. 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 (20 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 page where the user can add customers, rooms, and room reservations. Also make a simple HTML page where the user can search for reservations based on any of their features. 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).

Part 4: Basic DBA task (15 points)

Write a Python script (just a small Python program) that populate your database with many entries. The entries don't have to make sense to humans, e.g. the customer names can be A, AA, AAA, or 1, 2, 3 etc – we just do this to artificially create many entries so that we can test *performance*. Then, (II) write Python scripts to test the query performance of the database. Your queries must involve various SQL joins. In your Python program you must use Python commands that delay (e.g. make the Python program wait at some line for 0.05 seconds) and other Python commands that measure time; e.g. you record the time before a loop that generates many queries and then you record the time immediately after the loop is done – by subtracting the values you time the loop. (III) experiment with these queries and report graphs showing the performance (turnaround time) for executing batches of queries. Then, (IV) learn how to make indexes in MySQL and make indexes that will improve the performance. (V) Report the new graphs after you did appropriate indexing. Finally, (VI) write an essay of at most 200 words explaining your findings.

Remark: do not use the ready-to-use MYSQL tools for profiling/performance evaluation.

Integrating RDBMS's and Web services (35 points)

Research the literature and discuss in 4,000 words or more, and also include graphs, diagrams, and whatever else you found to justify your exposition on the following topic:

As a technology CGI is not widely used anymore. Furthermore, it is not the simplest to use. We used it in class because it is the cleanest to comprehend when first approaching the issue of integrating web-services and RDBMSs. Other alternatives exist, e.g. FastCGI. Furthermore, various programming languages have built their own interfaces for communicating with Apache. Why is that? Read about Java Servlets, also discuss Apache Tomcat, PHP, and search for other technologies that act as intermediates between an SQL Server (e.g. MYSQL) and web interfaces.

In the first half of your report (2000 words) you should compare these technologies and give at least one example where one would be preferable to another. Also, discuss why Python is also very relevant to certain such applications. In your comparison you should come up with very brief and high-level fictional settings that involve how web-services integrate with RDBMSs – when you discuss the role of Python you should also explain how web services and Business Analytics performed over structured databases work (as in e.g. over a relational database system such as MYSQL). Finally, discuss Javascript **when it is used in the web-browser side (client-side)** to access databases and explain why this should be never used. Do not confuse the use of Javascript in creating a user-interface (here we are only referring to using Javascript when accessing directly from the browser a database). In your overall exposition (overall means: PHP, Servlets, FastCGI, CGI, Javascript) you should explain carefully where each program is running (which things run on a Web Browser and which run on the server side) and why is this very important to know and carefully design.

In the second half of your report (last 2000 words) you should read and understand the high-level organization of the following “platforms” or interfaces/protocols:

- NodeJS (what if Javascript were on the server-side)
- Django
- WSGI (explain the connection to Python)

Much of the above can be used without the Apache interfering or “interfering much” if you like. Compare the advantages of these technologies to what you collectively discussed in the first part of your report. Come up with a fictional case-study about a “Database problem story” where these technologies would be preferable to the ones discussed in the first part of your essay.

Remark: Your essay should focus on applications regarding Databases. It is not about Web-technologies in general. **You must focus on how RDBMSs integrate with Web-services.** Any generic, web-service focused exposition will not receive marks.