Sistemas de Informação e Bases de Dados
Assignment 3 - Web Application

Using Views for a Dashboard

Create views over the tables in the database model, corresponding to the following relational schema.

```
dim_date(date, day, month, year)
   IC: date corresponds to a date existing in consultations

dim_client(VAT, gender, age)
  VAT: FK(client)

dim_location(zip, city)
   IC: zip corresponds to a zip code existing in clients

facts_consultations(VAT, date, zip, num_diagnostic_codes, num_procedures)
  VAT: FK(dim_client)
 date: FK(dim_date)
  zip: FK(dim_location)
```

Present the SQL code for creating each of the views corresponding to the tables in the previous model, so that the views feature the corresponding records in the database (i.e., information on all the clients that had consultations, together with the associated number of procedures, number of diagnostic codes).

Indexes

Suggest indexes that could improve the performance of the following query:

```
SELECT
    VAT,
    COUNT(*) AS num_consultations,
    SUM(num_procedures) AS total_procedures
FROM
    facts_consultations
GROUP BY
    VAT
ORDER BY
    total_procedures;
```

Provide SQL instructions for implementing the most appropriate indexes. Justify your choice and provide the corresponding query plan(s).

A Web Application Using the Database

Develop a simple Web-based application leveraging the database created and populated in Part 2 to perform the following tasks:

- A client comes to the clinic asking for an appointment. We need to check if the client already exists in the database, and check if there is a doctor available for the desired date/time. You should therefore create Web pages to support these verifications and the search for matching clients based on different information elements: given the VAT, a (part of the) name for the client, and/or the (parts of the) address, you should display the records of matching clients. On the displayed result, include the possibility of registering a new appointment for the listed clients. Include also options for adding new clients to the database, and for listing the doctors that are available for consultations at a given date/time (you can consider that doctors can give consultations on any day, and that all consultations last for one hour, starting at exact hours between 9AM and 5PM).

- Create a set of Web pages to support the access and registry of information associated with an appointment/consultation. Selecting (or clicking on) a given client in the results page of the previous interaction should lead to another page listing, chronologically, all previous appointments and consultations for that client. Selecting (or clicking on) an appointment/consult from this list should lead to a page presenting all information on the appointment/consult, including the SOAP notes, existing diagnostic codes, and any existing prescriptions. There should also be an option for adding information for a new consultation (i.e., add information to an appointment created to the interaction from the previous point), including the assisting nurse(s), SOAP notes, diagnostic codes, and prescriptions.

- Create a Dashboard Web page that uses the facts_consultations View and OLAP queries.

You do not need to code all the functionalities of a full application, instead focusing only on the ones that were asked. If some other operation is needed (e.g., inserting new doctors, inserting new diagnostic codes, etc.), you can perform it manually through the SQL command line prompt. Consider using simple HTML forms and tables to illustrate the results, avoiding complex Web design choices. Whenever appropriate, use prepared statements in the code, thus increasing performance and/or avoiding security problems with SQL injection.

Submission Notes

A report for the 3rd assignment should be submitted to Fenix as a single PDF file, readable with a standard program such as Adobe Reader. The report should have one separate section for each of the aforementioned tasks. Please include the code together with a screenshot of each Web page. Bonus points if you also include the URL for a working version of the Web App (i.e., use db.tecnico.ulisboa.pt from IST to load the database and deploy the web application using the instructions on the README.md, as shown on the lab classes, and provide the corresponding URL in your report), together with brief instructions for how it should be used. The document cover page should mention the names, student numbers, and group number of its authors. Provide notes explaining the rationale behind non-trivial implementation decisions.