Tasks for the F-Klub case in DIS

Purpose

As you probably already know, the F-klub sells different products (e.g., drinks and events) to students and staff, mainly from the CS department. To be able buy something from the F-klub, you have to be a member of the F-klub and make an advance payment to get a positive balance on your account. The F-klub uses a home-grown software system to keep track of each member's balance and the products the member has bought. All data is available in a single database. This works very well for the F-klub. However, it is difficult for the F-klub to analyze the data and get an overview. In this case study, we assume that you have been hired by the F-klub as consultants. Your job is to **design and implement a business intelligence solution that can help the F-klub**. The solution should be easy to use since we have been told that the F-klub wants to hire an administrator that knows a lot about how to run a social club, but not much about computers or software. For now, **you work on a proof-of-concept** meaning that you don't have to make a solution where all implementation details have been taken care of. The solution should, however, be showable and usable for basic analysis.

You are given access to the F-klub's data. As in any BI project, you ***cannot*** rest assured that the data from the source system is clean, complete, consistent, etc. Investigate the quality of the source data carefully! If you want to, you can "extend" the data in meaningful ways. For example, you can add cost prices (the F-klub would be able to tell you those prices if you asked for them). However, you should not add something like fictitious sales data (the F-klub does not benefit at all from a result that tells them that some data generator generates many fictitious sales of chocolate milk on Tuesday afternoons). Consider carefully if an extension adds value to the solution.

You should **go through the tasks listed below**. Note that **there are assignments to hand in**.

Required Software

You can work on the case in small groups, and it is up to you if you want to run your solution on a single PC or on all your PCs.

You will need *PostgreSQL* and either the GUI-based ETL tool *Apache Hop* (requires Java 17) or the code-based *pygrametl* (requires Python). If you want to use pygrametl, you can install it with `pip3 install pygrametl` and you should also install psycopg2 to be able to connect to PostgreSQL from Python code: `pip3 install psycopg2`.

If you use Windows, you will also be able to run *TARGIT*[1] on your local system which is recommended. If you don't use Windows, you can make a virtual machine such as Oracle VirtualBox with Windows[2] if you want to try TARGIT. If you don't want to use TARGIT/Windows, you can instead use *Mondrian* as OLAP engine and *Saiku* as OLAP client[3]. Note that this is much more difficult to set up and gives a less good user-experience.

[1] Installation instructions for TARGIT will follow later
[2] You can get Windows for educational purposes from https://www.ekstranet.its.aau.dk/software/microsoft
[3] https://github.com/project-a/mondrian-server provides a self-contained .war with Mondrian and Saiku bundled

Timeline

Note that a total number of ~33 hours is expected for the case work. Only 12 of those hours are scheduled.

- Finish Task A *before* case session 1
- Finish B, and C in case session 1 – 19/9. Spend most time on Task C.
    - Hand in *Assignment 1* no later than 12:30 on the 25/9. This assignment can be mailed to me as a PDF file. Make sure to ***clearly show the full name(!!!) and AAU email address of each co-author on the frontpage*** and ***CC every co-autho***r when you send the PDF.
- Work on Task D in case session 2 – 26/9
    - Hand in *Assignment 2* when you are done, but no later than 12:30 on the 2/10. This assignment can also be mailed to me as a PDF file. Remember to show full names and AAU email addresses on the frontpage and CC every co-author.
- Work on Task E and F in case session 3 – 3/10
    - Hand in ***Assignment 3*** **(in a single document also including the two previous assignments)** no later than 8:00 on the 20/10. This assignment can also be mailed to me as a PDF file (show full names and AAU email addresses and CC every co-author).

If you have time left in the ordinary exercise sessions, you can also work on the case. Talk to me if you for some reason need some extra time. Do not hand in work your group did not do itself!

Task A – Prepare your system

Get and install PostgreSQL and the ETL tool of your choice. Get the source data here.

You do not have to document this task.

Task B – Choose business process(es)

Explore the source data. Consider which business processes you *can* model. Choose which business process you *want* to model. It is absolutely fine if you **pick just one business process. Do not pick more than two business processes**. Choose the granularity for your business process(es). Consider if your choices are reasonable for a paying customer. Give examples of the kinds of questions you want to be able to answer with the new BI solution.

You document this task and the following task in the same assignment.

Task C – Dimensional modeling

Design the dimensions and choose the measures. You can choose to have SCDs, but it is also fine if you have no SCDs in this proof-of-concept. Remember to make analysis easy.

Document your design by using the diagramming technique used in the course book. Also show the relational schema that implements the model.

- *Advice 1:* Use only lowercase letters for all table names and column names.
- *Advice 2:* Create proper surrogate keys for dimension tables.
- *Advice 3:* Task E will be easier if you create a star schema instead of a snowflake schema.
- *Advice 4:* Pay attention to which user owns the tables you create – dwuser should be the owner

Insert manually a few rows of test data.

Assignment 1: A short description of the business process and the granularity including arguments for your key choices. Diagrams showing your dimensional modeling and a relational schema that implements the model. Add explanations/arguments for your key choices. Max. six A4 pages per case group.

Task D – Design and implement an ETL flow

Make a high-level description of your ETL flow for doing an initial load of the relational data warehouse. Implement the ETL flow using Apache Hop or pygrametl. If you use pygrametl/Python, you can use psycopg2 to connect to PostgreSQL.

If you want to, you can make an ETL solution that performs incremental loads. In this project, it is, however, also perfectly acceptable if you choose just to make a solution that TRUNCATEs all target tables and then does a full load.

If you need to do some processing that is not available in the ETL tool (or that you cannot figure out how to use), you are free to use other approaches. For example, you can write a program that does preprocessing of the data before the ETL tool is used. If you use pygrametl, you can of course also use programming to solve problems.

You do not have to set up scheduling of when to refresh the DW.

Assignment 2: A description (i.e., a figure and explanations/arguments for your key choices) of your ETL flow. Screenshots from the GUI-based ETL tool showing your flow or code (snippets) from your pygrametl program, including explanations of non-obvious things. Max. six A4 pages per group.

Task E – Create your cube(s)

This task differs depending on your used software. We have relatively small amounts of data, so you do not have to make any aggregates.

TARGIT (recommended!)

Define your structure as described in the TARGIT documentation and shown in the demonstration 27/9.

Mondrian (*not* recommended!)

Read Section 1 – 4 in http://mondrian.pentaho.com/documentation/schema.php. Based on this, you can define your multidimensional cube by editing the Mondrian cube definitions file which is specified in your mondrian-server.properties file. Run Mondrian as documented on https://github.com/project-a/mondrian-server.

To browse the data, start the web browser and go to http://localhost:8080. Note that Mondrian is very sensitive to even small errors in the XML. The error messages and debugging facilities are unfortunately not very good, but look in the debug information Mondrian prints to the screen. To find the cause of the problem, you might have to comment out parts of your XML and then enable them one by one.

You document this task together with Task F.

Task F – Create reports and analyze the data

Use a graphical OLAP client like TARGIT (recommended) or Saiku and analyze the data. Prepare some analyses that you can show to the F-klub when you deliver the new BI solution. Be prepared that they want to drill-down into the data and slice-and-dice it ☺

Assignment 3: Show screenshots of analyses made on your cube. Show the answers to (some of) the queries you considered in Task A. Max. five pages (+ the pages from the previous assignments - this final assignment should also include all the previous parts). It is OK to update something, e.g., your dimensional design or ETL flow, from an earlier assignment.