Modeling and Databases: Project Rules

A) About groups. The project can be developed either alone, or in a group of two. When the project is developed in a group of two, the numbers given below as upper bounds should be considered as lower bounds.

B) Draft of specification and models. The project is based on modeling a simple business case scenario through the design of a set of interconnected models (ER, BPMN) and a database. All the designs should follow a specification of the requirements that the students provide on their own, and **that must be discussed and agreed upon with the lecturers**. As a starting point for the discussion, the student should ideally provide:

- an initial *written description* of the application domain (the specification of the requirements),
- a draft of the *ER schema* that is as complete as possible (with respect to the specification), and
- the *high-level map with actions*.

By "draft of the ER schema", we intend the schema with the entities and relationships (with their cardinality constraints attached), and just the main attributes (*especially*, the identifiers). The diagram can be drawn by hand and does not need to be fancy, just readable. Typically, in the initial discussion and also in the initial design phases, the models get revised, so initially it is not worth spending too much time on the layout or graphical appearance. These initial requirements are also considered a **draft** at this stage, and are finalized for the following Point C. However, the more complete they are now, the better we can give you feedback on your project idea.

C) Final version of specification and models. After the above, students are **required** to confirm with the lecturer also the final specification (of roughly a page), the final versions of the ER diagram and of the high-level map with actions, before they start working on the subsequent phases of the design of their application. *The modality is the same as for the draft (point B)*. Students are also encouraged to use the office hours to clarify any doubts they might have about the various design phases. This helps to avoid that serious problems with their design get discovered only when the project is discussed as part of the exam. Roughly speaking, the specification of the requirements should satisfy the following criteria:

C.1 ER Model

1. It should be based on a domain containing between 6 and 10 main conceptual entities (i.e., without counting sub-entities that appear in ISAs or generalizations). 10 for groups of two.
2. There should be some structure in the set of entities, i.e., the ER schema should in addition contain a few ISAs and/or generalizations.
3. There should be sufficient structure in the relationships, which usually means that the representation of the ER schema as a graph (where the nodes are given by the entities and relationships, and the edges are given by the participation of entities in relationships) should contain some cycles.
4. The schema should contain cardinality constraints on the participation of entities to relationships that are different from the default $(0, n)$.
5. The schema should contain some identifiers made of multiple attributes, and at least one external identification for some entity.
6. The schema should contain at least one optional attribute and at least one multi-valued attribute.
7. There should be some external constraint that cannot be represented in the ER model.
8. The specification should include an indication about the volumes for the various entities and relationships (pay attention to the coherence between the volumes and the cardinality constraints of the ER schema). These volumes can be given at the end of the specification, with a brief paragraph or directly as a table.

9. The specification should include the frequency of at least 5 operations. Consider the following point when doing this: choose operations that are needed to implement activities in the process.

C.2 Case-specific objects and process

1. The process diagram should contain from 5 to 10 top-level activities that are linked to the high-level map.
2. At least two of the top-level activities should be considered as sub-processes that are then subject to child-level expansions, that is, specifying the inner subprocess structure. In the child-level diagrams, tasks can also be represented as collapsed subprocesses, which are not further expanded.
3. The process diagram should contain at least three different types of not None events (for example, start message, intermediate interrupting timer and end error).
4. There should be at least two resources participating in the process, preferably with duly added message flows. The consistency of added flows across different levels of the process diagrams should be respected.
5. The process should refer to (a subset of) the ER-diagram modeled before. If refinements of the ER-diagrama are needed, they can be documented in a separate diagram. But this is not mandatory.
6. Relevant case data objects as well as the (persistent) storage should be included in the process diagrams.
7. The process model should contain 1-2 DMN decisions with fully described decision tables.

D) Complete design of the database. The student should carry out the design of the database according to the requirements, following the methodology presented in the course. There are 5 phases:

1. *Conceptual design*, producing the following documents:
   - structured and organized requirements;
   - glossary of terms (typically optional, but required for this project);
   - diagram of the conceptual schema;
   - data dictionary of the conceptual schema (only for external constraints - the data dictionary for the entities, relationships, and the attributes can be omitted);
   - table of volumes and table of operations (those chosen before) according to the foreseen application load. Complete these 5 phases assuming the following simplification: these operations **are all the operations that need to be considered.**

2. *Restructuring of the conceptual schema*, producing the restructured conceptual schema (diagram and data dictionary), the corresponding table of volumes, and the access tables to assess the cost of the various operations.

3. *Direct translation* to the relational model, producing the relational schema with external constraints and the specification of the application load in terms of the relational schema.

4. *Restructuring of the relational schema*, also taking into account the application load, producing the restructured relational schema with external constraints and the specification of the application load in terms of the restructured relational schema. The reasons for the restructuring steps on the relational schema must be stated explicitly, by referring to the operations and the corresponding access tables.

5. *Specification of the database in SQL*, by defining relations with suitable constraints (capturing keys, foreign keys, and tuple constraints) and stored procedures with triggers to implement additional constraints (inclusions, external constraints). In case the students opt for using a DBMS other than PostgreSQL, remember that there are differences in how these are implemented. Refer to the material provided.

E) Design of the process. Analogously, the design of the BPMN diagram and of the DMN tables should be carried out according to the requirements and should respect the modeling conventions introduced in the lectures and labs.

F) Implementation. The student should also develop a simple application (typically in Java, or web pages in PHP) that implements **the operations as in the previous point** through the interaction with the database via a textual user interface (in the case of Java) or via the webpages themselves. The application does not need to be good-looking

and, *within reason*, implementation details not concerned with the content of this course will not be evaluated. The application should:

- allow the user to answer the queries that are part of the workload (typically by instantiating some parameters with values provided as input);
- allow the user to perform some basic insertions and/or updates of (some of) the data in the database, asking the user for input;
- catch the exceptions that might be generated so that the application does not terminate prematurely with an error.

G) Documentation. The documentation for the project should include:

- A single pdf file containing a textual description of the requirements and the documents for the phases above: *Conceptual design*, *Restructuring of the conceptual schema*, *Direct translation*, *Specification of the database in SQL*. Remember to include at the beginning of the pdf file a header or a title page with your name, student number, the title of your project, and the date;
- One or more SQL files containing the specification of the database (phase 5 above), including the insertion queries that populate it;
- Source files.

H) Zip file. All documents have to be bundled in a single ZIP file and have to be uploaded to MS Teams within the "Introduction to Databases Project" assignment.

I) Deadline. The deadline for submitting the project is, unless announced otherwise, at 23:59 two days before the day set for the discussion of the projects (the date will be communicated later).