Introduction to Databases – Guidelines for the Exam Project

The final mark of the *Introduction to Databases* exam is computed as a weighted average of the *written exam mark* (70%) and the *project mark* (30%). Moreover, to be admitted to the written exam, the student must have discussed the project, and the project must have been evaluated positively. In other words, **without having passed the project, the written exam cannot be taken**.

Interaction with the Lecturer/TA Regarding the Project

All interactions with the lecturer/TA that concern the project have to take place in one or both of the following ways:

- in person, after the lectures/labs or during the office hours;
- in a group chat on MS Teams that involves the lecturer, the TA, and the student(s) that develop(s) the project. For that, please create in MS Teams the group chat and rename it to:
  "IDB Project – ⟨*Surname St1*⟩ ⟨*First Name St1*⟩[, ⟨*Surname St2*⟩ ⟨*First Name St2*⟩]" .

Due to the overload of emails that we receive, there is unfortunately no guarantee that emails sent to the lecturer/TA regarding the project will be read and answered. In other words, email is not an appropriate communication mechanism for any matter that concerns the project (and more in general the course), and MS Teams should be used instead.

Rules for the Project

a) The project can be developed either alone, or in a group of two.

b) The project is based on the design of a simple database, following a **specification of the requirements** that the students provide on their own, and that **must be discussed and agreed upon with the lecturer/TA**. 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) and a draft of the ER schema that is as complete as possible (with respect to the specification). By "draft" we intend the schema with the entities and relationships and just the main attributes, typically the identifiers. The diagram can be drawn by hand and does not need to be fancy, but should be readable. Typically, in the initial discussion and also in the initial design phases, the diagram gets revised, so initially it is not worth spending too much time on the layout or graphical appearance. Alternatively, for the discussion, the students can also present just the draft of the ER diagram, and describe the application domain in the meeting. However, trying to write up the specification of the application domain is a good exercise to better understand which information the database should store. And this specification is in any case required as part of the project documentation.

c) The **ER diagram** must be specified **in the notation that is adopted in the course**. This is not for aesthetic reasons, but because alternative notations that are sometimes used (e.g., adopted in textbooks or supported by specific tools), often do not allow one to represent important information about the schema. Small variations of the notation are acceptable only if they allow one to represent (in a straightforward way, without requiring additional explanations) all of the following elements: possibly multiple identifiers consisting of multiple attributes and/or roles; external identifiers of entities; identifiers of relationships; primary identifiers; minimum and maximum cardinalities on roles and on attributes; attributes both of entities and of relationships; optional and multivalued attributes; composite attributes; complete and non-complete generalization hierarchies. If a student intends to use an alternative notation in their project, they have to carefully evaluate how each of the above elements is represented in their notation. If these elements are not supported, the notation has to be discarded.

   In addition, to ensure readability, the diagram must be drawn on a plain white background, avoiding inverted colors, squared or lined paper, and any elements that reduce the contrast. If drawn on a computer, the size of the fonts must be chosen so that the text stays readable when the whole diagram is shown on the screen of a laptop, and if drawn by hand, the handwriting must be legible.

d) Students are also **required to confirm with the lecturer/TA the final version of the ER schema** (diagram and constraints), before they start working on the subsequent phases of the design of their database. They 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. This could cause the project to be evaluated negatively, which would prevent the student from taking the written exam.

e) Roughly speaking, the specification of the requirements and the corresponding ER schema should satisfy the following criteria:

1. It should be based on a domain containing at least 5 main conceptual entities (i.e., without counting sub-entities that appear in ISAs or generalizations). When the project is developed by a single student, there should be at most 10 main conceptual entities, and when it is developed in a group of two, there should be at least 10 main conceptual entities.
2. There should be some structure in the set of entities, i.e., the ER schema should contain **at least one ISA** and **at least one generalization**.
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 all of the following elements:
   - at least **one n:n relationship**, i.e., a relationship where the maximum cardinality on two different roles is n;
   - cardinality constraints on the participation of entities to relationships that are different from the default $(0, n)$. In particular, **all of the cardinalities** $(0, n)$**,** $(1, n)$**,** $(0, 1)$**, and** $(1, 1)$ **on a role** should be present in the schema;
   - **more than one primary identifier made of multiple attributes**, and in addition at least one **external identifier**, i.e., an identifier of some entity that involves both attributes and roles;
   - at least **one optional attribute**, at least **one multi-valued attribute**, and at least **one composite attribute**.
5. There should be **some external constraints**, i.e., constraints that cannot be represented in the ER model. Remember that a proper external constraint should impose some *condition on the information represented in the database*, i.e., one should be able to provide a database instance that would violate the constraint, and the constraint then rules out such instances.
6. 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).
7. The specification should include a workload of the most common **queries and operations** (between 6 and 10) that are of interest in the modeled domain, with an indication of their frequency. **At least 3** of the operations should be **queries**, and **at least 3** should be **updates** on the database (insertions, deletions, or updates). When the project is developed by a **group of two**, these **bounds are doubled**.

f) The student should carry out the design of the database according to the requirements, following the methodology presented in the course, and consisting of the following phases:

1. *Conceptual design*, producing the following documentation:
   a) structured and organized requirements;
   b) glossary of terms (typically optional, but required for this project);
   c) diagram of the conceptual schema;
   d) data dictionary of the conceptual schema (listing the entities, relationships, and external constraints – the data dictionary for the attributes can be omitted);
   e) table of volumes;
   f) table of operations, according to the specified application load.

2. *Restructuring of the conceptual schema*, producing the following documentation:
   a) restructured conceptual schema (diagram and data dictionary);
   b) updated table of volumes (only for the entities and relationships that have changed);
   c) access tables to assess the cost of the operations in the workload.
3. *Direct translation* to the relational model, producing the relational schema with external constraints.
4. *Restructuring of the relational schema* taking into account the application load. **The reasons for the restructuring steps on the relational schema, the tables and constraints affected by the change, and the resulting tables and constraints, should be stated explicitly**, by referring to the operations. The specification of the complete restructured relational schema with external constraints resulting from the restructuring is optional, but students are advised to specify it, since having such complete schema eases the implementation in SQL.
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 (e.g., inclusions, disjointness, external constraints).

g) The student should also develop a simple application (typically in Java) that implements some interaction with the database via a textual or graphical user interface. The application should provide the following functionalities:

- 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 the update operations (insertions, deletions, updates) that are part of the workload, asking the user for the necessary input, and guaranteeing that the consistency of the data in the database is preserved;
- catch the exceptions that might be generated by JDBC due to possible mistakes in the interaction with the database, so that the application does not terminate prematurely with an error.

Rules for Handing-in the Project

a) The documentation produced for the project should include:

1. A single pdf document consisting of a textual description of the requirements (of roughly one page) and the documentation for Phases 1–4 above.
   The document should have a title page containing at least the name and student number of the student, the title of the project, and the date of completion.
   The pages of the pdf document should be numbered (the page numbers are important for the evaluation, so that one can refer to them for comments).
2. One or more SQL files containing the specification of the database (Phase 5 above); if your project includes more than one SQL files, place all SQL files in a sub-folder called "`sql`".
3. One or more files (typically in Java) containing the application that interacts with the database; if your project includes more than one Java files, place all Java files in a sub-folder called "`java`".

b) The student should **not** include in the project folder any other files than the ones above! In particular, they should not include any library files, jar files, pom files, IDE configuration files, or any other files or folders generated by the IDE. The purpose of what is handed-in is not to provide an executable version of the project, but to provide the source files for a manual check of what has been implemented. The functionality of the application will be reviewed when the students discuss their project.

c) All documents should be placed inside a folder named
"⟨*SurnameSt1*⟩-⟨*FirstNameSt1*⟩-⟨*IDSt1*⟩[+⟨*SurnameSt2*⟩-⟨*FirstNameSt2*⟩-⟨*IDSt2*⟩]_⟨*ProjectName*⟩"
and the folder should be compressed in a single ZIP file with the same name as the folder (and extension `.zip`).

d) The ZIP file has to be uploaded to MS Teams within the "Introduction to Databases Project" assignment.

e) 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 (see below).

Rules for the Project Discussion before the Exam

a) The discussion of the IDB projects typically takes place one or two days before the date set for the written exam, and will be announced on MS Teams before each exam session.

b) One or two days before the project discussion, we will post in MS Teams a schedule that assigns a specific time slot to the discussion of each project. Specifically, the discussion of the projects will be scheduled in the order in which they have been submitted to MS Teams (considering the last modification time), where projects submitted first will be discussed first. The discussion of each project will last between 15 and 30 minutes.

c) The purpose of the project discussion is two-fold:

   1. To discuss the various phases of the database design carried out in the project, and verify the correctness of the corresponding documentation.
   2. To verify the functionality of the developed application that interacts with the database, and show how the update operations executed in the application affect the data stored in the database.

d) In the case of a face-to-face discussion (this is the default modality that will be applied, unless there are specific reasons to do differently), students should bring a laptop with the project documentation, a running version of the application they have developed for the project, and a running version of pgAdmin connected to the database they have developed.

   In the case of an online discussion via MS Teams (this modality will be applied only in exceptional circumstances), students should be ready to share on their screen the project documentation and display in the shared screen a running version of the application they have developed for the project, and a running version of pgAdmin connected to the database they have developed.

e) Students should be ready to display on their laptop (or share on screen via MS Teams) the different parts of the project documentation:

   - the requirements,
   - the conceptual schema (diagram and constraints),
   - the restructured conceptual schema (diagram and constraints),
   - the relational schema (with constraints) resulting from the direct translation,
   - the restructured relational schema (with constraints).

   This can simply be the respective pages of their project document, if the information therein is readable well on screen, or separate pdf or image files (e.g., when the resolution in the project document is too low).

f) In order to discuss the various phases of the development of the database design, and illustrate how the output of one phase is used to generate the output of the next phase, it is convenient if the two outputs can be visualized simultaneously on the screen. Therefore, for the purpose of the discussion, there should be two separate pdf files with the same content, so that two different parts can be visualized simultaneously (e.g., the restructured ER schema and the relational schema resulting from the direct translation).

g) Notice that when the discussion of the projects takes place in the office of the lecturer, the project documentation will be visualized on the screen of the lecturer, using the version that has been handed in.

   Instead, the functionalities of the developed application and their effects on the database have to be shown in any case on the laptop of the student (or shared on screen via MS Teams).