L.EC012: Bases de Dados

DB Project 21/22

This project aims to create and query a database in a context proposed by each group. To do so, it will be necessary to create the conceptual model for the selected theme, map this model to a relational schema, implement the schema in a SQLite database, populate the database and, finally, query the database.

Deliverables and dates

Composition of the working group [October 31, 11:55 PM]

Theme Definition [November 7, 11:55 PM]

Deliverable I [November 21, 11:55 PM]

i. **Task**: A
ii. **Deliverables**: Report in pdf format

with: cover (1 page), table of contents (1 page), context description (1 page) and UML diagram (1 page)

iii. **Evaluation:** 25% of the project grade

Deliverable II [December 12, 11:55 PM]

i. **Tasks**: B, C, D, E e F

ii. **Deliveries**:
   - **Report in pdf format delivered in I with the following additional sections**: revised UML diagram (1 page), relational schema (1 page), functional dependencies analysis and normal forms (1 to 2 pages); list and implementation form of constraints (1 to 2 pages)
   - Files: create.sql and populate.sql

i. **Evaluation**: 25% of the project grade

Deliverable III [January 30, 11:55 PM]

i. **Tasks**: G e H

ii. **Deliverables**:
   - **Report in pdf format delivered in II with the following additional sections**: listing of 10 queries in

natural language (1 page); listing of 3 triggers in natural language (1 page)

- Files: create.sql, populate.sql, intN.sql (x10), triggerN_add.sql (x3), triggerN_remove.sql (x3) and triggerN_verify.sql (x3)

i. **Evaluation**: 50% of the project grade

Composition of each group of students

The project will be done in groups of 3 students. Each group should be created in the "Groups for the project" activity in moodle. The first digit of the group identifier should be the class number. For example, group 101 should be composed by students from the class 2LEIC01.

Theme Definition

The theme should be proposed by the group to the lecturer of the theoretical-practical classes during the week beginning November 1st. After approval of the topic, the group must send, through Moodle, a description of its topic with a maximum of 100 words. This description should be similar to the statements of the exercises of the theoretical-practical classes, stating the expected entities, their attributes and how they relate to each other.

As an indication, relational schemas with 10 to 15 relations are expected. Some of these relations should have composite keys.

Tasks

A. **Conceptual Model Definition**

Before creating the conceptual model there should be a familiarization with the context associated with the topic of the work. In this step it is intended that students understand in detail the data that will have to be stored in the database.

A conceptual model must then be created in UML for the database to be created. In this model all necessary constraints and multiplicities of associations must be clearly indicated. If there are derived elements, these must be flagged appropriately.

The UML class diagram should be included in a report, where the context associated with the database should also be described. The context description must include all information that may be important for the evaluation of the conceptual model and *should not* be a description of the conceptual model.

A. **Relational Schema Definition**

Before the relational schema is defined, the conceptual model should be revised based on the evaluation of Deliverable I. The revised UML diagram should be added to the report that was previously delivered.

The revised conceptual model should be mapped to the relational schema, which should be added to the report in textual format using syntax:

R1 (atr1, atr2, atr3->R2)

A clear indication of the primary and foreign keys of each relation is expected.

A. **Functional Dependencies Analysis and Normal forms**

For each relation the associated set of functional dependencies and any violations of the Boyce-Codd Normal Form and 3rd Normal Form must be given. The non-existence of violations must be justified. Relations that are neither in the Boyce-Codd Normal Form nor in the 3rd Normal Form must be decomposed to one of these normal forms.

A. **Database creation in SQLite**

The next step involves creating the database in SQLite. SQLite allows you to read commands from a file. This feature should be used to (re)create the database whenever necessary.

You should create a file called create.sql that includes the SQL statements for creating all the tables mentioned in the relational schema resulting from step 4. Before the tables are created, make sure to delete previous tables with the same name. The file should look like this:

```
drop table if exists R1;

...

create table R1 ( .... );

...
```

A. Adding Constraints to the Database

When creating the database, you should include all convenient constraints for maintaining the integrity of the stored data. It is necessary to consider that the implementation of constraints in SQLite is not fully compliant with the SQL-99 (SQL2) standard.

The defined constraints must be listed, in an ordered manner and in natural language (for example: "no two students can have the same ID"), in the report. For each of the constraints you must also specify how it is implemented - key constraint (PRIMARY KEY or UNIQUE), referential integrity constraint (foreign key), CHECK constraint, NOT NULL?

Once you have identified how to implement each constraint, you need to implement them. To do this, changes must be made to the create.sql file. Constraints that require a trigger to be implemented should be left for Deliverable III of the project.

The create.sql file should be submitted in the 2nd deliverable.

A. **Data Loading**

After creating the database it is necessary to populate it. At this stage you must create a file called populate.sql that contains the SQL statements necessary for the introduction of data in the tables created. At the beginning of this file you must include the statement

PRAGMA foreign_keys = ON;

to ensure that the referential integrity check of the database is active.

The populate.sql file must be submitted in the 2nd deliverable.

A. **Database querying**

For this task a set of queries relevant to the database context must be defined. For example, a query that lists the existing countries in a library database is less relevant than a query that lists the most requested books in a given period. From this set, 10 queries should be selected that follow the criteria:

- are **different** from each other (for example, having one query that lists the name of the customers in the database and another that lists the name of the companies in the database is equivalent to having only 1 query);

- in their construction make use of the widest **diversity** of SQL operators;

- are of distinct **complexity.**

The 10 questions should be listed, in an ordered manner and in natural language, in the report.

As with the database creation, queries should first be tested interactively via the SQLite command line client.

Queries should be efficient. Whenever possible you should favor joins rather than queries with subqueries.

Each of the 10 queries should be written to its own file: int1.sql, int2.sql, ..., int10.sql. At the beginning of these files the following instructions should be included to make the result more readable:

```
.mode columns

.headers on

.nullvalue NULL
```

The file names should correspond to the order of the queries mentioned in the report.

A. Adding Triggers to the Database

Finally, 3 triggers should be defined that are useful for monitoring and maintaining the database. At least one of the triggers must implement a constraint. For each trigger 3 files must be created: triggerN_add.sql, triggerN_remove.sql and triggerN_check.sql, with N = 1, 2 or 3.

In triggerN_add.sql, you must include the SQL statement that allows you to create the trigger. If the constraint for which you are creating the trigger can be violated by more than one type of modification to the database, you must implement only one of the triggers and indicate in the report which additional type of trigger(s) would be needed to enforce the constraint. If the trigger finds that a constraint is being violated, it can modify the database to ensure that the violation is disabled or it can trigger an error. A SQLite trigger can trigger an error by:

```
SELECT raise(abort, ‘<error message>’);
```

When this instruction is executed, the action that triggered the trigger is undone and the desired error message is displayed.

In the file triggerN_remove.sql the statement that deletes the trigger from the database must be included.

In the file triggerN_verify.sql you should include the SQL statements that allow you to confirm that the trigger is well implemented. For example, if the trigger inserts a tuple into relation R2 whenever a tuple is inserted into relation R1, this file should have statements similar to:

```
SELECT * FROM TABLE R2;

INSERT INTO R1 VALUES (value1, value2, ...);

SELECT * FROM TABLE R2;
```

In the report you should briefly describe, in an ordered manner and in natural language, the 3 triggers implemented.

The file names must match the order of the interrogations mentioned in the report. In each of the files, the referential integrity check must be activated.

Evaluation of the participation of the various elements of the group

At the end of the report associated with each deliverable there should be a paragraph that qualitatively assesses the contribution of each member of the group to the final result associated with that deliverable This paragraph will allow anomalous cases to be detected and to act accordingly.

Delays

In order to be fair, late submissions will be penalized 1 value for each day of delay.

No submissions will be accepted later than 1 week from the submission date.

Authorship and Originality of the Work

All works will have their originality fully scrutinized. The authors of prevarications will be punished in case of similarities with third-party works (works not cited, other works by students of the curricular unit, etc.), that can range from the cancellation of the enrollment on the curricular unit to the initiation of disciplinary process to all elements of the group in question.