Laboratory #3

An heterogenous DB system

In the third laboratory of the subject, you need to create a simple application that uses **both** a relational and a non-relational database.

First of all, you will be assigned a type of non-relational database system[1]. Check in Campus Virtual which NoSQL DB type you have been assigned. Take your time to study the DB type.

You'll need to choose a specific DBMS for that **NoSQL** DB type (find the most popular ones using the specific rankings in [1]). Using a programming language of your choice, you'll need to learn to connect to the DBMS from code.

In parallel, you will need to create a relational DB using the **Relational** DBMS that you prefer. You will need to connect from code to this relational DB too.

Next, you will need to design and implement a (component of a) toy application that *requires* a DB system that uses **both** a NoSQL DB and a Relational DB.

Example 1

Imagine that you have been assigned the NoSQL DB type: *Graph databases*. You select Neo4j and MySQL as DBMS for NoSQL and Relational subsystems, respectively. You envision a social network application that saves the relationships between people and posts in the Graph DB, and the details of people and posts in a Relational DB.

Your application needs to use **key** attributes that can be used to establish relationships between data from both DBs. Then, implement CRUD operations (insertions/deletions/updates) of your application, and make sure that they are **atomic** operations across your heterogeneous (NoSQL + relational) DB system; that is, a modification which can affect both DB systems needs to be carried out completely, or not at all.

[1] For more information on popular tools of the different DB types: https://db-engines.com/

Example 2

How are you going to know the details of a post (RDBMS) that was liked by (GraphDB) a given user with certain characteristics (RDBMS)?

What happens if a user removes a post which was liked by someone else?

What happens if a user removes its previous "like" to a post?

What happens if a user removes its account?

Apart from the Lab, you will need to prepare a small **presentation** for our Theory lessons explaining:

1. the characteristics of the **NoSQL DB model** assigned to you, and
2. a simple explanation on **how to use the DBMS** that you chose for it.

You'll need to present it in front of your classmates. Check in Campus Virtual which (theory) slot you have been assigned.

Laboratory #3

An heterogenous DB system

What to do:

1. Check which NoSQL database type you have been assigned.
2. Choose a specific DBMS for that NoSQL DB type, and install it.
3. Learn about how to use it.
   a. How do you create a DB? How do you modify data (insert/delete/update)?
   b. How do you make queries?
4. Think of a toy application that needs both a NoSQL DB and a relational DB.
5. Design the (implicit) schemas, i.e., explain your data, and create the DBs.
   a. Which **keys** are you going to use to join the data from both DBs?
   b. Populate the DBs.
6. Connect to both DBMSs from your app's code.
   a. You can choose the programming language that you prefer.
7. Implement CRUD operations (insert/update/delete) for your app.; you need to implement at least one that affects both the NoSQL DB and the relational DB at the same time. Thus, the operation needs to be atomic; that is, the modifications in both DBs are carried out, or none.

   Is your DBMS append-only? How would you *represent* a deletion? That is, how would you differentiate valid data from invalid (*deprecated*) data?

Extra. Prepare a presentation for your classmates with all what you have learnt about your NoSQL DB type and how to use the DBMS that you chose for it.

Laboratory #3

An heterogenous DB system

What to deliver:

- Explain to me the NoSQL DB type assigned and the chosen DBMS for it.
  Labs slot **November 18th (19:00)**
- Explain to me your application (follow Example 1 above)
  Labs slot **November 25th (19:00)**
- Explain to me the CRUD operations that you are considering (follow Example 2 above)
  Labs slot **December 9th (19:00)**
- Delivery of the whole final application
  Defense @ Labs slot / Code via CV **December 16th (19:00)**
- Presentation[2] of your NoSQL DB type to your classmates.
  Theory slot **December 14th/21st (17:00)**

How:

- In pairs
- Only one delivery per pair

[2] https://fs.blog/2021/02/feynman-learning-technique/