Methods and tools for the development of Design database for a bank app

1st Alejandro Gómez Moyano
Faculty of Engineering
Universidad Distrital Fransisco José de Caldas
Bogotá ,Colombia

2nd Catalina Ariza Ardila
Faculty of Engineering
Universidad Distrital Fransisco José de Caldas
Bogotá ,Colombia

***Abstract*—the banks have a thirty terabytes of information that grows fifteen percent per year, and this data have to be secure, easy to found and show. that´s the reason to have a simple, fast and scalable database, protecting the documentation, and thinking in the future with the reduction of redundancy for less cost in vane for this type of archive**

***Index Terms*—ER model, ontology method, user stories, CRUD,relational algebra**

I. Introduction

since the creation of the databases in 1884, there have been many advances to keep the data safe and clean, this is more necessary in the record of money and the moves of it,the loss or wrong modification of counts are a catastrophe,Also in a era of the internet that increments the need of immediacy and different situations is required limits of thousandths of seconds to manipulate our finance. in other types of model in characteristics like the provision of data description tables in recently developed information systems represents a major advance toward the goal of data independence . Such tables facilitate changing certain characteristics of the data representation stored in a data bank. However, the variety of data representation characteristics which can be changed without logically impairing some application programs is still quite limited. [3] . An understanding of RA can be used to improve query performance The query-processing component of a database engine translates SQL code into a query plan that includes RA operations. The query optimizer attempts to speed up query execution by reducing the processing time of each operation [2] .the ER-model is a good way to keep the order of the data,prioritizing the security and the accuracy of the moves and changes using json archives isn´t neccesary think in the lenguage of the backend or frontend, its like a universal lenguage to communicate with everything

II. Methods and Materials

A. materials

1) Draw.io : The principal material for the fist part of modeling, is a software to diagramming ,draw any type of mental maps, concept maps, diagrams or different graphic representations, such as hierarchy or set diagrams.

Identify applicable funding agency here. If none, delete this.

2) python: is a programming lenguage but in specific the library faker to create a dummy database ,like a proof for the attributes and the limits.

3) Mysql: it´s a open source sql databse management system.

4) Postman: it´s an Api fro buildings and using API´s

B. methods

1) analyze user´s stories:: use the user stories for begin to found a pattern between the entities and functions or actions, see various names of objects , persons o entity that repeat and identify her attributes. in this case was necessary to create it because there aren´t client, thinking in the security was important use a transaction code to verify the identity of the user. Also will be possible see the client who owe a lot of money to have a report of debt.

2) know the limits:: thinking and making an exam of the requirements extract the limits of the database, thinking in only the internal system, or if this system can be connected to other banks, do actions like transaction to another bank.

3) create the M.E.R:: using the 10 steps ontology to create the MER it´s easy to identify the relations, attributes and needs of the database with the base of the user stories

4) perform tests: with the operators of RA, its a way to test the results and functionality of the diagram

5) create the database in SQL: the diagram ans the user stories make it more easy to do this task, and with the triggers and functions

III. Experiments and Results

create a system, see modules in the real life and extract connections its not a task that will be perfect,but with the practice there are ways to taste the design or found the way to write the query to show determined information ,recognize one or more entity to have a relation between 2 objects say that this relation is not direct, or with the relation 1 to 1 , if this relation is not optional so that means that is something that no is a entity and should be part of a table.

in the search of taste the model ,the attributes and the constraints, it is necessary to have a dummy database, it is hard to found a data in internet that satisfy the same characteristic, so use the library faker of python was a good idea, this presents advantages because have an option of generate data fake of a specific country,with this its possible found a average in the names, address, phone number,e-mail and the password.

the result throws a simple diagram, with a semantic and closer vocabulary of the final client, this is important for a new revision, it could be with a stakeholder who knows the additional data to recollect,new function or a use for get information with new queries or a analysis. with Relational algebra ,something universal and near of the managements its a way to recognize what it is the purpose of their requirements .

A. examples of Relational Algebra

1.rename the table to ”bank cards” and project bank cards with associated accounts and their expiration’s dates:

$$P_{bank\ cards}(\Pi_{card\ number,account,expiry\ date}(Banking\ cards))$$

2.project the date and account of movements of user whose name is Pedro ,before April 9,2024:

$$\Pi_{date,amount}(\sigma_{date} <\text{"}April\ 9,2024\text{"}(Movement\ history) >)$$
$$\cap(\sigma_{name} = \text{"}Pedro\text{"}(user))$$

3.project user x movements before x date :

$$\Pi_{id,name,amount}(\sigma_{userfk=date} <\text{"}X(Movement\ history) >)$$

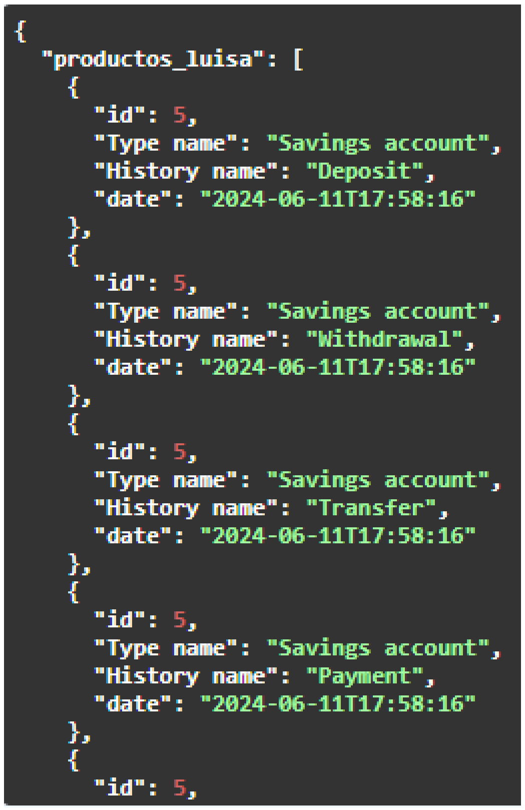
most of this queries are necessary to see by the client like the record of movements with the date in an order chronological that’s the reason of automate and insert into a trigger, and this activate when there are a new movement and update hourly.

CÓDIGO SQL

```
CREATE VIEW productos_Luisa AS
SELECT fProduct.id, fpType.name AS 'Type name',
    mHistory.name AS 'History name', mHistory.date
FROM financial_product fProduct
INNER JOIN users usr ON usr.id_number = fProduct.
    user_fk
INNER JOIN financial_product_type fpType ON fpType.
    id = fProduct.status_fk
INNER JOIN movement_history mHistory ON mHistory.
    financial_product_fk
WHERE usr.name = 'Luisa'
GROUP BY fProduct.id, fpType.name, mHistory.name,
    mHistory.date;

SELECT * FROM productos_Luisa;
```

In the process of writing queries, it was possible to find tasks necessary to obtain a report or update depending on specific actions, for this reason it was decided to implement triggers and automate with functions, to obtain reports. Another decision was to implement the use of fast APIs to make it easier to enter additional data, thinking about the scalability of the database, making it more efficient to have multiple requests without losing performance and implementing a connection with any programming language.



```
{
  "productos_luisa": [
    {
      "id": 5,
      "Type name": "Savings account",
      "History name": "Deposit",
      "date": "2024-06-11T17:58:16"
    },
    {
      "id": 5,
      "Type name": "Savings account",
      "History name": "Withdrawal",
      "date": "2024-06-11T17:58:16"
    },
    {
      "id": 5,
      "Type name": "Savings account",
      "History name": "Transfer",
      "date": "2024-06-11T17:58:16"
    },
    {
      "id": 5,
      "Type name": "Savings account",
      "History name": "Payment",
      "date": "2024-06-11T17:58:16"
    },
    {
      "id": 5,
```

Fig. 1. output del query de view productos luisa

IV. CONCLUSION

In this work, we have explored methods and tools for developing a design database for a banking application. We began by discussing the importance of the entity-relationship model and the ontology method for representing entities and relationships among the involved parties. These approaches allow us to effectively capture the needs and requirements of stakeholders, establish meaningful connections between entities, and define a clear structure for our database.

Additionally, we analyzed the modeling process step by step, from identifying entities and attributes to creating the entity-relationship model and normalizing relationships. We emphasized the importance of continuous validation and evaluation of the model, as well as the use of tools like Draw.io and libraries like Faker in Python to generate test data and facilitate development.

We underscored the utility of relational algebra for querying and verifying the effectiveness of our design. We recognize that developing a database is a dynamic process that requires continuous attention to the changing needs of users and application functions.Finally the use of create a Api it´s important to have various request and more easy to use the functions.

The methods and tools presented in this work provide a solid framework for the successful development of a design database for a banking application, ensuring its security, efficiency, and usability. We are confident that these approaches will significantly contribute to the project's success and meet the needs of the stakeholders involved.

References

[1] López Rodríguez, Yoan Antoni, Hidalgo Delgado, Yusnie, & Silega Martínez, Nemury. (2021). "Escenarios de vinculación de las bases de datos relacionales y las ontologías: un mapeo sistemático". Enfoque UTE, 12(4), 58-75.

[2] McMaster, K. and Sambasivam, Samuel and Hadfield, S. (2012). Relational algebra and SQL: Better together. Proceedings of the Information Systems Education Conference, ISECON. 29.

[3] Codd, E.F. A Relational Model of Data for Large Shared Data Banks. In: Broy, M., Denert, E. (eds) Pioneers and Their Contributions to Software Engineering. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-48354-7-4.