



DIGITALBUBBLE MARKETPLACE

Databases – LEI 2021/2022

MEET THE TEAM



ANDRÉ COLAÇO
andrecolaco@student.dei.uc.pt



SANCHO SIMÕES
sanchosimoes@student.dei.uc.pt



RODRIGO MACHADO
ramachado@student.dei.uc.pt

WHAT ARE WE CREATING?

We are developing a digital sales platform backend (database and REST API) specifically for electronic products, such as computers, televisions, smartphones, etc..., to support the commercialization of these goods.

The platform allows sellers to list their products, and buyers to shop through the full catalogue of products sold by all sellers.

The buyer can add items to their shopping cart with a desired quantity and finalize their order. Prior to buying, buyers can have all their questions answered in a questions section, with support for discussion threads. After buying, they are able to comment on / rate the purchased items.

Sellers are notified of purchase orders and questions on their stock, and buyers with any updates regarding their orders. All users in a question thread are notified with updates to the respective thread.

Administrators moderate the platform, register sellers/buyers, and create limited time discount campaigns with a limited amount of coupons which are applicable platform-wide.

Any changes/updates made to product listings are saved in a history, for later auditing of the platform and price tracking.

DATABASE OPERATIONS

The database main purpose is to hold all data relevant to the operation of the digital marketplace, such as users, products, shopping carts, orders, campaigns and product comments.

Triggers keep the products' history and stock up-to-date with updates and purchases, as well as delivery of notifications to users.

DATABASE TRANSACTIONS AND CONCURRENCY CONFLICTS

Purchases can only be finalized if all products in the cart have enough stock, and all stock must be updated before finalizing the order. To prevent inconsistencies, this operation should adhere to the ACID properties of database transactions.

If two buyers intend to buy all remaining stock of a product, one of these purchases must prevail, while the other fails. This means that the product's stock should be accessed / altered in an exclusive fashion.

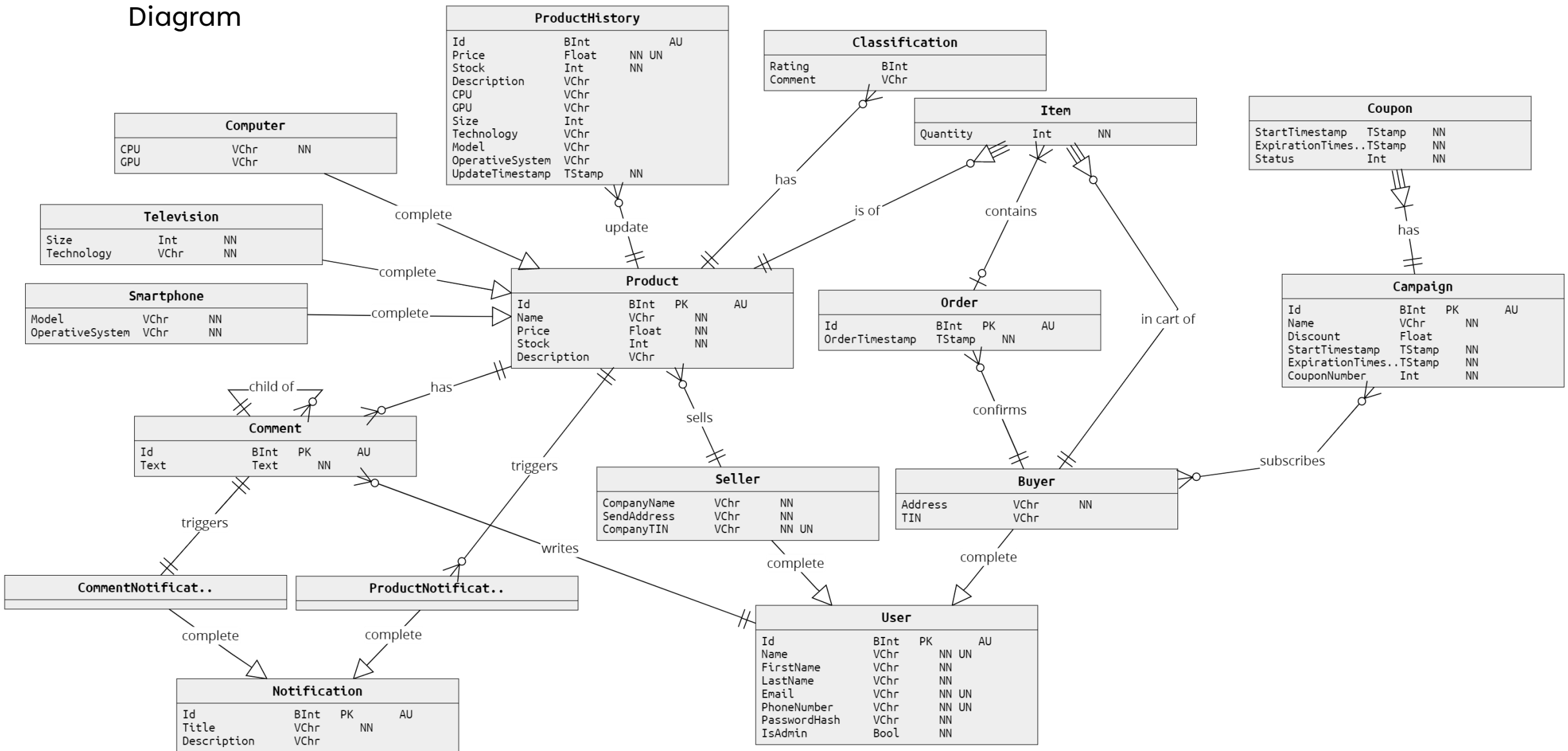
Campaign coupons exist in a limited number, and should be handed-out in a first-come first-served fashion. Buyers can only get a single coupon per campaign, which is available to them after subscription to the campaign.

These coupons suffer from the same concurrency conflict as purchases, meaning that the transaction must be done in an exclusive fashion.

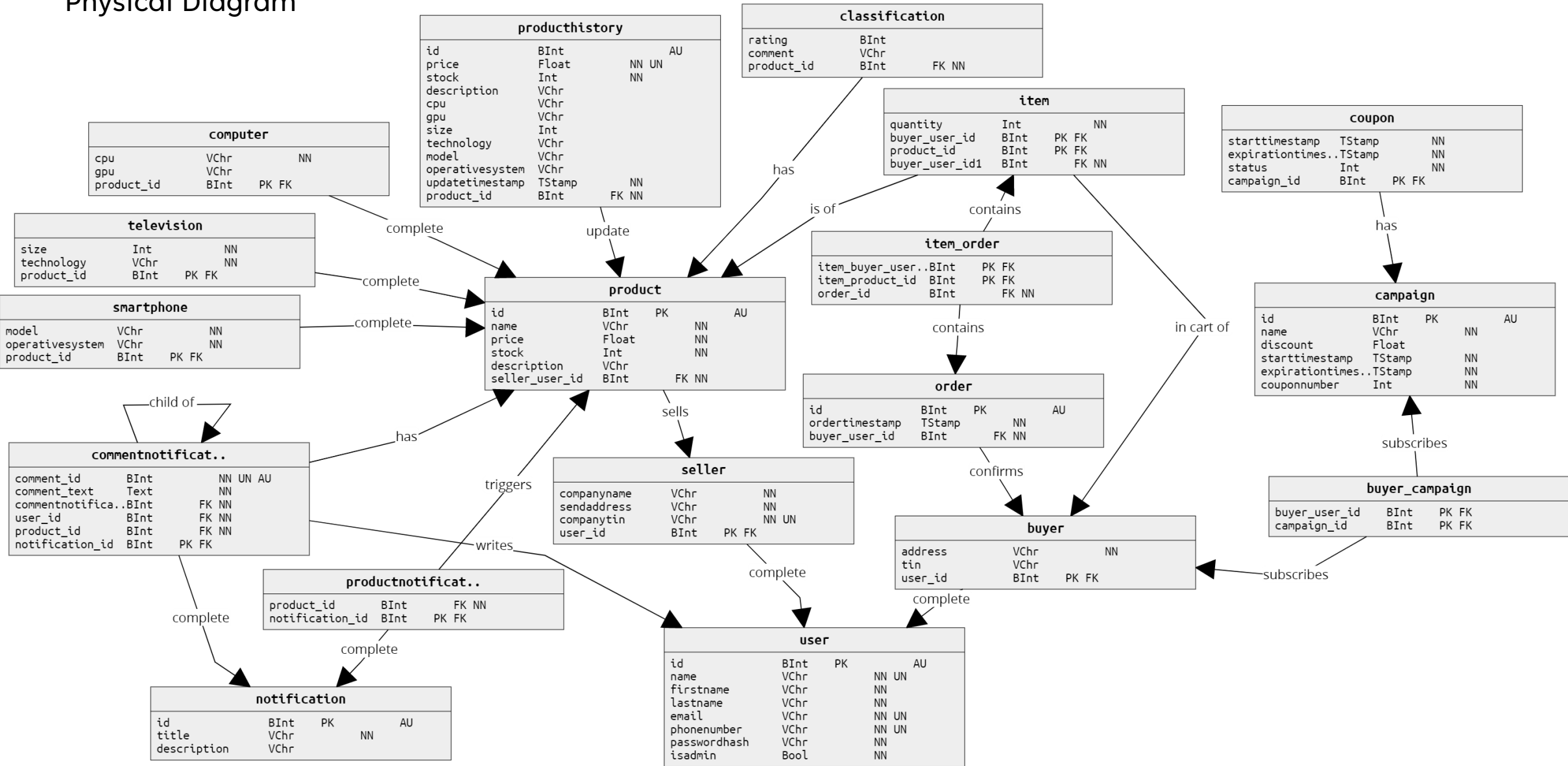


DATABASE DIAGRAMS

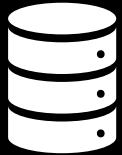
Entity-Relationship (ER) Diagram



Physical Diagram



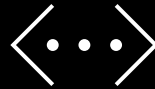
TECHNOLOGIES USED



DBMS

PostgreSQL 14.2

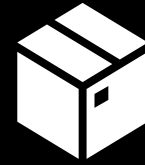
- Object-relational
- FOSS



REST API

Python 3 + Flask Microframework

- Database connection made with **Psycopg2**
- FOSS



CONTAINERS

Docker + Compose

- Host-agnostic
- Isolated environment

DEVELOPMENT PLAN

MARCH

- Draft database ER
- Develop REST API skeleton

APRIL

- ER tuning
- DB triggers and procedures development
- API authentication and authorization
- Transaction development
- API endpoints and DB query development
- DB indices creation

MAY

- Documentation and report
- Final testing and touch-ups
- Benchmarking
- Creation of API requests examples
- Final delivery

Work will be divided equally amongst team members along the project timeline.