CT80A0000 Data-Intensive Systems

Assignment 3 Documentation and explanation of some design choices

Räkköläinen Aino

Simple Yarn E-shop web application

This distributed database system is a simple e-commerce platform for selling yarn for knitting and crocheting. Knitting needles and crocheting hooks are sold as well. Some talented designers have provided their patterns available for the shop and some of those are free of charge and some with small fees.

There are three different databases: one in UK, one in Italy and one in Finland. Each country has some unique products but some of them are also replicated. For instance, products from Italy that are available for shipping to Finland, can also be shown when customer selects Finland.

Knitting and crocheting patterns are usually available fully online and thus, that table will be replicated to all countries. Then, when it comes to yarns, I decided to have those tables partially replicated meaning that some of data items are available at each country, but some can be unique for only for specific country. I used some real yarn and knitting product brand names such as Novita, KnitPro, Adriafil and took inspiration from some existing e-shops (LoopKnitting, 2024; Novita, 2024 and YarnStreet, 2024). Otherwise, all data is dummy data and not really accurate. For instance, it is not sure that Novita yarns are actually sold/shipped to Italy or UK.

Data Fragmentation is implemented within Yarn table because it is likely to be the biggest one and thus, it uses horizontal fragmentation based on the country of origin. For demonstration, it uses only the three countries which databases are defined but it could be scalable to have other countries of origin as well. In my opinion, it could help with replication of yarn table too because then the updates can be written only to partition that corresponds to the origin of the yarn. Of course, there might be some imbalance between countries, and this might not be the most efficient partition method, but it was useful for the purpose of this assignment.

Partitioning is demonstrated in UI in a way that user can filter only the yarns originated to currently selected country. It means that in the backend, the query is done to that particular partition instead of the whole yarn table. By default, the user interface shows all values of yarns table from all three countries.

Lazy distributed data replication protocol could fit for this because each shop functions independently and has mostly their own products. Except the few ones that are sold in all places. The data replication was not automatized for this assignment, instead it

was done by hand by copying the wanted data to other databases. However, that is one future improvement to make this more realistic product if certain tables are replicated automatically.

References:

Data:

UK Dummy Data:

LoopKnitting. 2024. British Yarns. [website]. [referenced 6th November 2024]. Available at: https://loopknitting.com/collections/british-yarns

Finnish Dummy Data:

Novita. 2024. Yarns. [online document]. [referenced 6th November 2024]. Available at: https://novita.com/collections/langat

Italian DB Dummy Data:

YarnStreet. 2024. Yarn from Italy. [website]. [referenced 6th November]. Available at: https://yarnstreet.com/news/yarns-from-italy

Code (links to tutorials which were followed while coding or otherwise checked:

Connection to Postgresql was done by following this tutorial: https://www.youtube.com/watch?v=O4bNwkC1ZxA

Materialize library was used for UI and this tutorial was used to setup it: https://www.youtube.com/watch?v=ETulDkWxVew

Inserting multiple values in SQL: https://www.naukri.com/code360/library/sql-insert-multiple-rows

Parameters with GET queries in Express: https://stackoverflow.com/questions/6912584/how-to-get-get-query-string-variables-in-express-js-on-node-js