Database Systems: NoSQL (DATA.DB.300) Project

Spring 2026

Project report

Teemu Salonen (student number: K444339)

Tuukka Salonen (student number: 151125973)

Contents

Business domain description 1
Chosen DBMSs 2
Use cases 2
Primary use cases (PostgreSQL) 2
Additional use cases (MongoDB) 4
Database design 5
Database distribution 6
PostgreSQL distribution 6
MongoDB distribution 7

Business domain description

The business domain is an online digital game distribution platform for selling, distributing, and managing digital video games. The platform allows users to create accounts, browse a catalogue of games, purchase games and view owned games. Game publishers can publish new games, manage game data, set prices, and control regional availability. The system maintains records of users, games, purchases, and ownership, along with essential game-related data, including interactions and top trending games.

The service is international and intended for users worldwide, with a primary focus on Europe. Games may have country-specific availability and have country-specific pricing.

The main end-users of the platform are players, who browse and purchase games, and publishers, who publish and manage their game offerings. The platform is expected to support continuous user registrations, frequent game browsing, and regular purchase transactions, with peak concurrent users reaching tens of thousands during high-traffic periods, such as new game releases.

From a data access perspective, the domain is read-heavy, as users frequently browse game listings and view game details. Approximately 70–80% of operations are read operations, while 20–30% are write operations, consisting mainly of user registrations, game reviews, game publications or updates, and purchase transactions.

Chosen DBMSs

Primary DBMS: PostgreSQL

Reasoning:

- Strong transactional guarantees (ACID)
- Referential integrity between users, purchases, and games
- Structured data with clear relationships
- Supports strong indexing
- Ideal for core business data (users, games, purchases, pricing)

Secondary DBMS for the additional features/use cases: MongoDB

Reasoning:

- Well-suited for high-volume, write-heavy interaction data (game interactions)
- Flexible schema allows easy evolution of user interaction models
- Scales horizontally through sharding

Use cases

The queries for the use cases for PostgreSQL and MongoDB can be found in the ‘queries/’ directories of their respective folders within the repository.

Primary use cases (PostgreSQL)

UC1: User registration

- Write operation
    - Create a new user account

UC2: User login / account lookup

- Read operation
    - Retrieve user account information

UC3: Browse available games

- Read operation
    - List games currently available for buying (for the user's country)
    - Can be filtered by genre, price, or release date

UC4: Search games

- Read operation
    - Search games by partial title, publisher or description
    - Using combination of pg_trgm and fuzzystrmatch for similarity search
        - First uses trigram matching for the initial candidate search
        - Then uses weighted scoring with similarity, matching title, publisher and description. Levenshtein is used for title score. The scores are then combined for a total relevance score.
    - Returns 25 most relevant search results

UC5: View game details

- Read operation
    - Retrieve detailed information for a single game based on its id
    - Uses user id to only return available games for the requesting user

UC6: Publisher adds a new game

- Write transaction
    - Insert a new game into the platform
    - Assign publisher, pricing, availability, hardware requirements and other data
    - Insert data into multiple tables using a transaction

UC7: Publisher updates game

- Write transaction
    - Update pricing and/or other data for a game

    - Use transaction when updating multiple tables

UC8: Purchase a game

- Write transaction
    - Create a purchase record
    - Assign game ownership to the user
    - Insert data into multiple tables using a transaction

UC9: View owned games

- Read operation
    - List all games owned by a user

UC10: View purchase history

- Read operation
    - List purchased game titles for a user
    - Sort by purchase timestamp

Additional use cases (MongoDB)

UC11: User likes or reviews a game

- Write operation
    - First check from PostgreSQL that user owns the game
    - Create an interaction record for the game (like or review)

UC12: View trending games

- Read operation
    - Retrieve the most popular games for a selected time range (day, week, month, or year) based on user interactions
    - Retrieve detailed game data from PostgreSQL based on resulting ids
    - Trending data would not be computed in real time from user actions but instead, a scheduled background job (like a cron task) would periodically aggregate interaction data from 'game_interactions' collection and update the 'game_trending_stats' collection

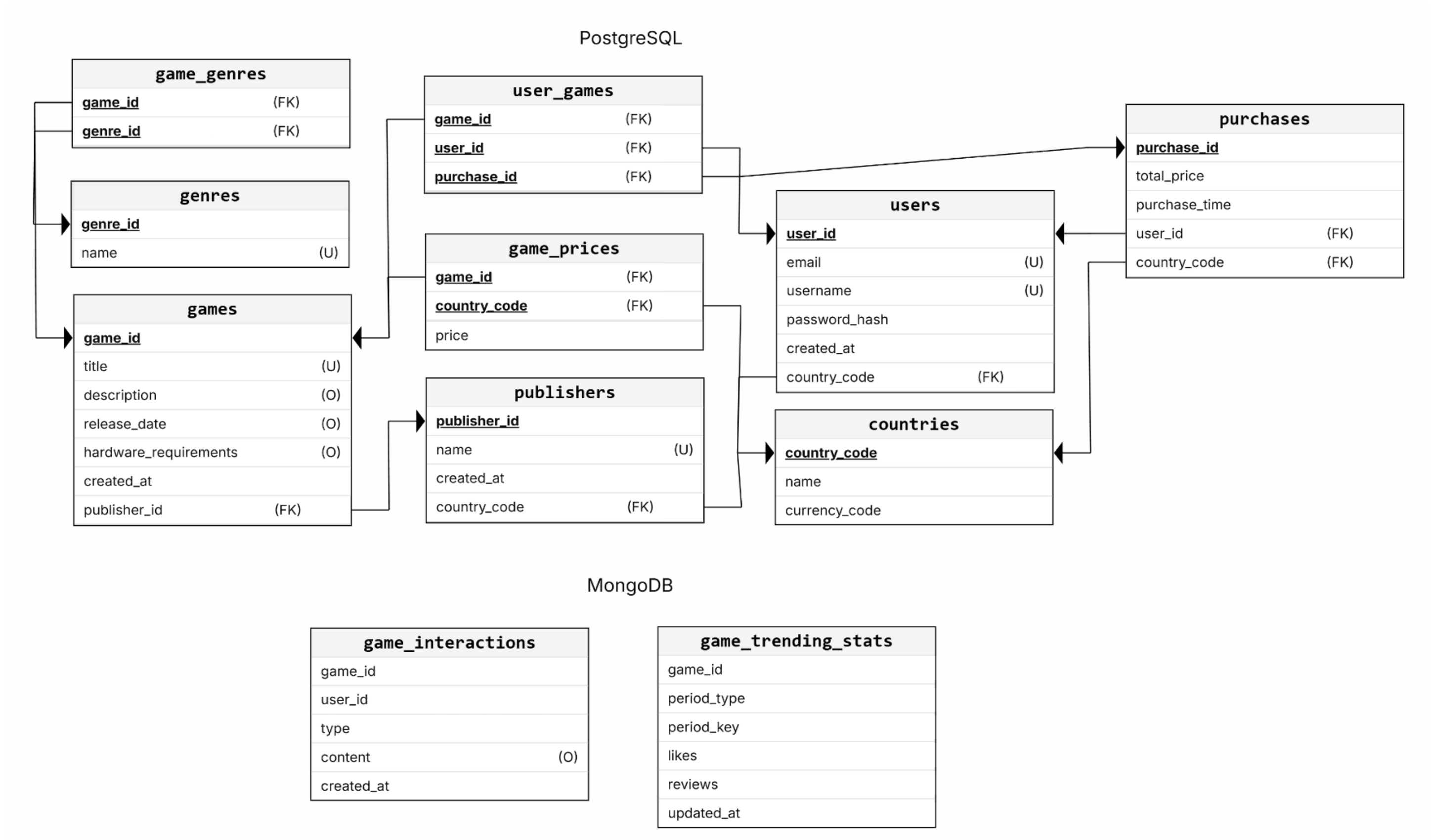
    - The trending statistics follow a hybrid aggregation model: daily records store per-day interaction counts and rolling aggregate documents (weekly, monthly, yearly) are derived from daily data for fast querying

UC13: Retrieve games with highest engagement

- Read operation
    - Retrieves top games based on most interactions
    - Considers time since last interaction to the total engagement score for the game. If more than a day/week has passed since the last interaction for the game, time weight score is dropped affecting the total engagement score
    - Retrieve detailed game data from PostgreSQL based on resulting ids

Database design

The relational schema of the database is presented below. The initialization scripts for PostgreSQL and MongoDB can be found in the 'init/' directories of their respective folders within the repository. The diagram is available in the 'documentation/' folder.



Database distribution

The distributed database will have 20 overall database nodes in 5 data centres, 5 being PostgreSQL nodes and 15 being MongoDB nodes.

The data centres are:

- Germany
- Netherlands
- US East Coast
- Xi’an, China
- New Zealand

PostgreSQL distribution

The goal of the PostgreSQL distribution is to maintain strong ACID guarantees. There will be one primary node that handles writes, which is in Germany. The 4 other nodes, located in Netherlands, Xian (China), New Zealand and US East Coast will have read replicas of the primary node serving read operations for the users closest to them.

This design aims to keep the read operations fast for the users while keeping the data consistent between the nodes. Also, centring the write operations to one data centre, eliminates the need for cross-datacentre transactional operations.

Distribution

Germany (primary):

- Node 1: PostgreSQL master

Netherlands:

- Node 2: PostgreSQL replica

US East Coast:

- Node 3: PostgreSQL replica

Xi’an, China:

- Node 4: PostgreSQL replica

New Zealand:

- Node 5: PostgreSQL replica

MongoDB distribution

The goal of the MongoDB distribution is to keep low latency for high volume of game interaction and trending game aggregation operations. The distribution will include 5 shards, with each shard including 3 nodes (primary + 2 secondaries) for high availability. ‘game_trending_stats’ collection will be replicated to all nodes with no sharding due to its relatively low size. Hashed ‘user_id’ will be used as the sharding key for the ‘game_interactions’ collection, as most operations are performed on a per-user basis and this approach provides a reasonable load balance. If users’ region data were to be added to the ‘game_interactions’ collection, sharding could be optimized further by using a compound shard key combining region and ‘user_id’, allowing data to be distributed more effectively based on users’ actual locations.

Distribution

Shard 1:

- Primary node: Germany
- Secondary nodes: Netherlands and Xi’an, China

Shard 2:

- Primary node: Netherlands
- Secondary nodes: Germany and New Zealand

Shard 3:

- Primary node: US East Coast
- Secondary nodes: Xi’an, China and New Zealand

Shard 4:

- Primary node: Xi’an, China
- Secondary nodes: US East Coast and Germany

Shard 5:

- Primary node: New Zealand
- Secondary nodes: Netherlands, US East Coast