Database analyzes

Research and Results

Quantum Delta

Table of Contents

Glossary

Term	Definition
document-oriented database	A document database (also known as a document-oriented database or a document store) is a database that stores information in documents.
ORM framework	Is a piece of software designed to translate between the data representations used by databases and those used in object-oriented programming.

Introduction

This document will cover the decisions we will take regarding the database. So what framework are we going to use, how many databases, the design of the database, and which database server? These decisions will be based on research and best practices.

Purpose and Scope

This research document will cover the key decisions for the database. These decisions will cover the structure of the database, the choice of the appropriate frameworks, the required amount of databases, and the optimal choice for the database server.

To be concrete, it will cover the following:

- Framework Selection: Go over different database frameworks to choose the most appropriate one based on the needs of Quantum Delta.
- Database Quantity: Analyzing how much traffic the databases need to handle, looking at performance optimization, and maintenance.
- Database Design: an overview of how the design of the database will look, with an explanation of what the tables are used for.
- Database Server Selection: What different services are there to offer, that are free, have great performance, and have great security features?

Database Server

To store the data for the tournament management system, we need a database server to store the data. A database server is hardware that runs database software, which helps us manage the data. (Lombardi, 2023)

Management System

The first consideration we need to make is what kind of system we will use to make our database since there are different servers for each system type. Therefore we will look into Relational Database Management System (RDBMS) and Objected-Oriented Database Management (OODBMS) to see what can fit.

Relational Database Management System (RDBMS)

This management system is a database type that stores and provides access to tables that are related to each other. It is easy to view and represent the data in tables. They are logically structured, each row in a table represents a record, and each column is an attribute. The language SQL is used for querying data. (What is a Relational Database (RDBMS)?, n.d.)

Furthermore, the servers that use this system, have a mature ecosystem, meaning that they have great documentation, community support, and tooling which can be great for Chess Tournament Application (Enterprise Software). Examples are MySQL, PostgreSQL, SQL Server, and Oracle database.

Object-Oriented Database Management System (OODBMS)

This system will represent data in the form of objects and other Object-Oriented principles. A main usage of this can be when you have complex data and relationships that are difficult to map in a relationship. Relation wise can be a benefit, due to that you can immediately see the relations of objects in the database. Examples of databases are Object Database, ObjectStore, and ZODB. (Difference between RDBMS and OODBMS, 2023)

What Server Benefits This project?

This project will not have complex relationships. The most complex will be to add users to a tournament, which will be doable in an RDBMS. Also, the team is familiar with SQL and has no experience with OODBMS. Therefore, due to the project's needs and preferences, we need a server that supports RDBMS.

From our NFRs, we concluded that around 100 concurrent players can use the website without any performance downgrade. Furthermore, to also handle peek users, it is a good idea to keep onto the 10X Rule, this is to make sure the target is set higher than what you think needs to be needed. (CLEAR)

Now let's go over familiar services to work with, see if their service is good, and what the benefits and downsides are. This will be done by giving a SWOT analysis.

Oracle Database:

	Beneficial	Harmful
Internal	**Strengths** 1. Well established, since 1977 2. Scalability, suitable for large enterprise, adaptable architecture. 3. Robust Security Features 4. Wide range of administration tools. 5. Mult-platform support, and various operating systems.	**Weaknesses** 1. Expensive Licensing, limits the adoption of small development teams. 2. Learning curve is high, administration experience, and high-level SQL knowledge. 3. Hardware requirements are high
External	**Opportunities** 1. Cloud Adoption, offer cloud solutions. 2. Advanced features: Community growth, new customers, and keeping them. 3. Expansion into new markets: more diverse user base.	**Threads** 1. Competition. 2. Security risks: increasing cyber threads. 3. Keeping up to date with the regulations.

(What is Oracle Database: Guide to How This RDBMS Works, 2022)

MySQL:

	Beneficial	Harmful
Internal	**Strengths** 1. Open-Source: Customizable, free to use. 2. High performance: Speed, reliability, flexibility. 3. Popularity: Preferred choice, wide range of languages.	**Weaknesses** 1. GPL Licensing Conditions: Restrictions 2. Steep learning curve for beginners. 3. Limited Support options: Community-driven may lack assistance.
External	**Opportunities** 1. Cloud adoption 2. Continuous improvement to attract new users. 3. Industry Standard: further expansion to different industries	**Threads** 1. Competition from other RDBMS options. 2. Increase in importance regards security 3. Evolving technology environment.

(WALLIS, n.d.)

Microsoft SQL Server:

	Beneficial	Harmful
Internal	**Strengths** 1. High-performance 2. Advanced security 3. Supports on-premises and cloud-based deployments 4. Advanced business tools, integrated with BI tools for data analysis. 5. Integrated with Microsoft tools	**Weaknesses** 1. Platform Limitations: Only Windows and Linux Support 2. Not open-source, limited flexibility 3. Cost: Licensing costs can become high, especially for the Enterprise edition 4. Smaller Community: Compared to open-source databases.
External	**Opportunities** 1. Cloud Adoption 2. Continuous Improvement, new users, and evolving needs of the industry. 3. Expanding to different industry usage	**Threads** 1. Competition: Facing from other RDBMS options, open-source alternatives 2. Data Privacy Regulation: needs to be strict otherwise impacts MsSQL's reputation. 3. Vendor Lock-in: Customer can become dependable on MsSQL.

(SUNBUL, 2023)

PostgreSQL:

	Beneficial	Harmful
Internal	**Strengths** 1. Open Source: Community contribution, custom without vendor lock-in. 2. Highly Expandable: extendable with a lot of other types 3. Supports complex data types 4. Supports the languages we use.	**Weaknesses** 1. Can be limited to hosting platforms, limits the user base 2. Documentation: A lot of documentation, only in English. 3. Read heavy scenarios at, a lower reading speed compared to other database servers.
External	**Opportunities** 1. Flexible and open-source, get further adoption. 2. Continued development, increases solutions to performance issues.	**Threads** 1. Competition from other 2. Rapid technology changes in industry. 3. It is open-source, but it has certain features that an organization can rely on when facing vendor lock-in.

(PostgreSQL: a closer look at the object-relational database management system, n.d.)

MongoDB (NoSQL, document-oriented database):

	Beneficial	Harmful
Internal	**Strengths** **1. High performance: data stored in RAM, increased query execution.** **2. Simple: query syntax.** **3. Quick setup: installation, setup, and execution and fast and easy to set up.** **4. Scalable: efficient handling of large datasets.**	**Weaknesses** **1. Limited data size: documents max 16 MB, nesting depth is restricted to level 100.** **2. Duplicated data can lead to corruption and is hard to manage.** **3.**
External	**Opportunities** **1. Cloud adoption** **2. Industry Expansion** **3. Improvements in indexing, and optimizing query performance.**	**Threads** **1. Scalability: handling larger datasets, becomes more challenging.** **2. Vendor Lock-in: Too much depending on the features of MongoDB.** **3. Competition.**

(What is a Document Database?, n.d.)

So based on these SWOT Analyses, we can see they can have the same benefits and some threads such as the competition. Since the team doesn't have a budget, Oracle and Microsoft SQL are not a good option. Therefore in this following order, we wanted to prioritize the database usage, since we may want to use multiple databases.

1. MySQL: Scalable, community support, it has a high performance, and is adopted in multiple industries making it one of the options for this project with the limited budget and a need for reliability.
2. PostgreSQL: Great for this project, with great functionalities like those of MySQL.
3. MongoDB: Document-oriented, high performance, and efficient for semi-structured or unstructured data, which can be rapid in deployment, it uses a different approach to the traditional data structure.
4. Oracle Database: Longstanding, good features, but can become quite expensive.
5. Microsoft SQL Server: Great enterprise solution, but this also can become expensive.

Framework

For the database handling in the back-end, we want to keep everything to be consistent and easy to understand. The framework that can make this happen for Quantum Chess Tournament is going to use is Entity Framework. It is an open-source ORM framework for .NET that is supported by Microsoft. (What is Entity Framework?, n.d.)

Entity Framework usage

This framework will let the team work with data, using objects in c#, without the actual focus on the tables and columns inside of the database. This will let the developers have a higher abstraction when using the data. Also, this way we can write functions to the database that are easier to understand, and less error-prone, due to the nature of entity framework, that will translate your request to an SQL command. Therefore, SQL queries are no longer needed to write. To give a better overview, see the following picture. (What is Entity Framework?, n.d.)
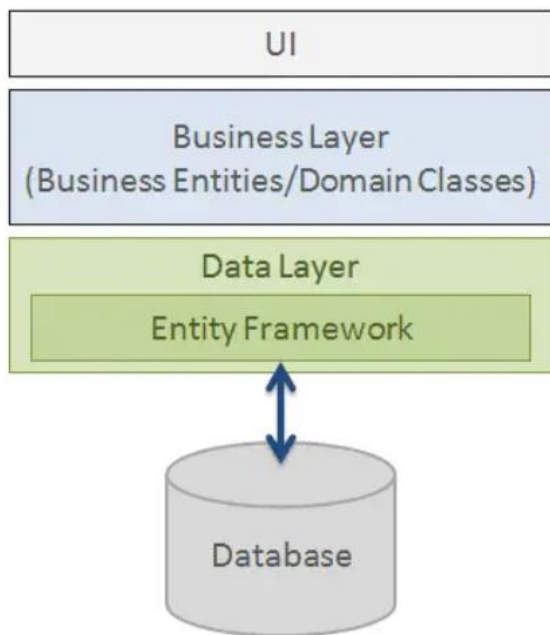

Figure 1: Entity framework way of working (What is Entity Framework?, n.d.)

To go more in-depth, see the following picture that shows how entity framework architecture gets the data.
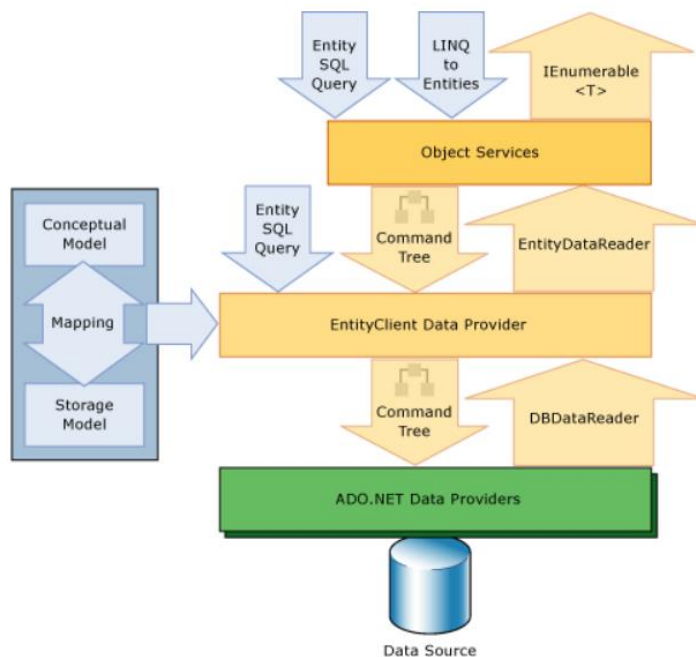

Figure 2: In-depth Entity Framework Retrieves Data. (Entity Framework overview, 2021)

So Entity Framework is a good way to allow the team to have fast development, with easy and quick operations to the database. The only thing is that you need to be careful when and know how it works to have great performance. With the right approach, also specified in (TLC Quantum, 2024, Coding Guidelines), to prevent down-performance.
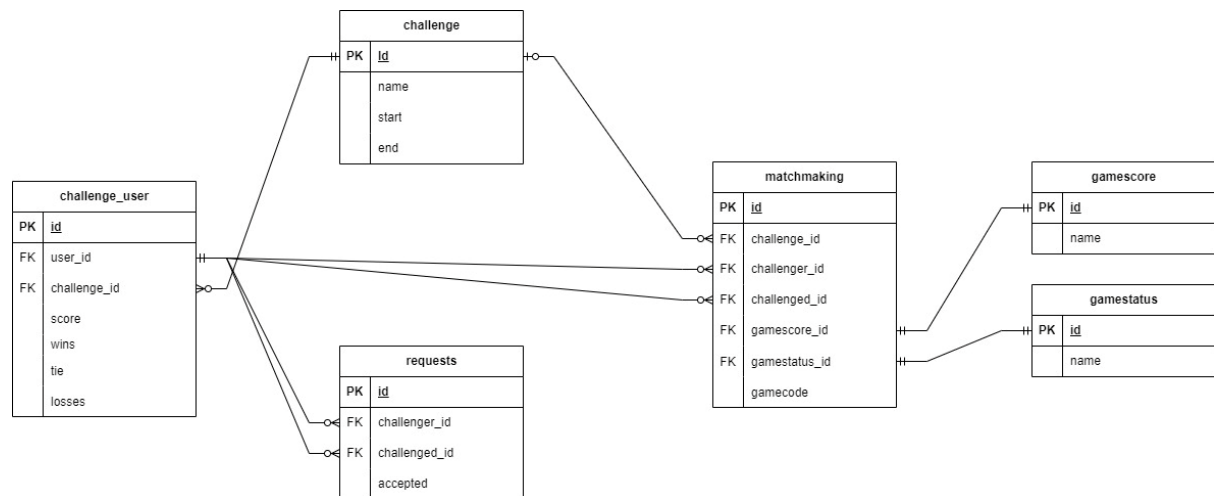
Database Design

Challenge Service Database

Figure 3: Challenge Service Database Diagram
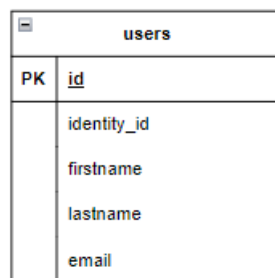
User Service Database

Figure 4 User Service Database

Description

The group got a request and it changed the plan of the project scope, therefore there is not yet an actual tournament database implementation. With the change of scope, we need to make a challenge system that can be used for tournament day on an event, therefore the priority is to make a challenge system. Moreover, this is also there is not yet a tournament database.

Keywords

Challenge: People can join a challenge. This challenge has a certain start and end date. People can challenge each other to a match.

Requests: After someone challenges a person, requests will be made, and the challenged person can either accept or reject it.

Challenge User: Once a user joins a challenge, a record will be created inside this table, listing all the user's data in that specific challenge.

Match Making: Once a user has accepted a game, a record is made with the needed information inside the match-making table. This also provides the game code that users use in order to play the game to each other.

Processes
Workflow:

- A challenge can be created for a certain time period. This is for now done manually.
- People can access these challenges by simply clicking a button.
- Users can view a list of all the users that are in the challenge, with the state if they are in a match or not.
- When a user challenges a user, a request is made, where the challenged user can accept or deny it.
- If accepted, a matchmaking is created, sending a game code to the users. This game can be entered and you will be redirected to the game. Once the game is over, the results are processed and the user stats are updated.

References

CLEAR, J. (n.d.). *The 10X Rule by Grant Cardone.*

Difference between RDBMS and OODBMS. (2023, 04 06). Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/difference-between-rdbms-and-oodbms/

Entity Framework overview. (2021, 09 15). Retrieved from microsoft: https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview

Lombardi, P. (2023, 09 25). *Database Server: Definition, Uses, Types and 7 Examples.* Retrieved from indeed: https://www.indeed.com/career-advice/career-development/database-server

PostgreSQL: a closer look at the object-relational database management system. (n.d.). Retrieved from ionos: https://www.ionos.com/digitalguide/server/know-how/postgresql/

SUNBUL. (2023, 11 26). *13 Key Differences Between MySQL vs MsSQL: Which Database To Choose in 2024?* Retrieved from redswitches: https://www.redswitches.com/blog/mysql-vs-mssql/#WhatIsMsSQL

WALLIS, J. (n.d.). *What Is MySQL? Definition, Comparision with SQL, Benefits & Features Explained.* Retrieved from intuji: https://intuji.com/what-is-mysql-definition-comparision-with-sql-benefits-features-explained/

What is a Document Database? (n.d.). Retrieved from mongodb: https://www.mongodb.com/document-databases

What is a Relational Database (RDBMS)? (n.d.). Retrieved from oracle: https://www.oracle.com/database/what-is-a-relational-database/#:~:text=A%20relational%20database%20is%20a,of%20representing%20data%20in%20tables.

What is Entity Framework? (n.d.). Retrieved from entityframeworktutorial: https://www.entityframeworktutorial.net/entityframework6/what-is-entityframework.aspx

What is Oracle Database: Guide to How This RDBMS Works. (2022, 07 25). Retrieved from sarasanalytics: https://sarasanalytics.com/blog/oracle-database/

TLC Quantum. (2024). Coding Guidelines (Unpublished manuscript), Fontys.