Database 2

EnchantedEars Final Assignment

NHL Stenden, Database 2, Information Technology

▼ Entities

Song: It has a songID, a title, an artist, a genre, a release date, duration, an album and a playlist.

Playlist: It has a playlistID, a name, a description, and a collection of songs.

Relationships: Many-to-One with Playlist, because a song can be in multiple playlist and each playlist can have multiple songs.

▼ Operations to Test

1. **Create:** Add new songs to the database.
2. **Read:** Retrieve information about songs, possibly with different filtering options (by artist, genre, album).
3. **Update:** Modify details of existing songs (change the genre or update the release date).
4. **Delete:** Remove songs from the database.

⛸️ Old Database 2

Following the module book: make the assignments and the final assignment.

▼ Considerations

Indexes: Explore how well each technology handles indexing, especially as the number of songs increases.

Complexity: Keep the schema and relationships simple to focus on the CRUD operations.

Introduction

EnchantedEars was founded in 2024 by Virag Szabo for educational purposes and her own improvement in the Netherlands (Haarlem). She wanted to create something unique besides Netflix or Spotify and something that is not to complicated to build, but hard enough to create. The student is going to work hard in the upcoming weeks on this final assignment, following the steps from the subject. This student needs to make a positive contribution to the implementation of this music streaming system.

Objectives

- Evaluate ADO NET, Entity Framework, and NoSQL (MongoDB) for CRUD operations.
- Test the speeds for varying numbers of songs (1, 1000, 100 000, 1 000 000).
- Generate a comprehensive report for decision-making.

Methodology

- Brief explanation of ADO.NET, Entity Framework (Code-First), and MongoDB.
- Description of CRUD operations on the "Song" entity.
- Use Stopwatch for measuring execution times.

Implementation

ADO NET

- Implementing a new database using ADO NET approach for Song and Playlist.
- Perform CRUD operations on the "Song" entity.

Entity Framework

- Implementing a new database using ADO NET approach for Song and Playlist.
- Perform CRUD operations on the "Song" and “Playlist” entity.

NoSQL (MongoDB)

- Implementing a new database using ADO NET approach for Song and Playlist.
- Perform CRUD operations on the "Song" entity.

Speed Test Results

- **Present the speeds in a matrix (table)**
  - Rows: 1, 1000, 100,000, 1,000,000.
  - Columns: ADO.NET, Entity Framework, NoSQL (MongoDB).
- **Create graphs to visually represent the speed comparisons**
  - Bar graphs for each operation (Create, Read, Update, Delete).
  - X-axis: Number of songs, Y-axis: Execution time.

Explanatory Text

For each comparison (ADO.NET vs. Entity Framework, Entity Framework vs. NoSQL, ADO.NET vs. NoSQL), provide explanatory text:

- Discuss any notable results or patterns observed.
- Analyse performance differences and possible reasons.

Reliability Measures

- Describe measures taken to ensure the reliability of the information.
- Specify the specifications of the device used for testing.