Database Interface - CRUD Operations with .NET and MySQL

Manta Mădălin-Ștefan, 432Fa  
Faculty of Electronics, Telecommunications and Information Technology

**This document presents the Music Management Application, a desktop-based system designed to provide a simple and intuitive interface for interacting with a relational database. The application implements CRUD (Create, Read, Update, Delete) operations for managing musical data, including albums, musicians, and their relationships. Developed using .NET Framework with Windows Forms as the frontend and MySQL as the database, the project focuses on usability and functionality. The interface enables users to seamlessly interact with the database through a user-friendly graphical interface. This report details the database structure, application design, and implementation of core functionalities, highlighting the effectiveness of the approach to database management.**

1. Introduction

The Music Management Application was developed to provide an intuitive and effective interface for managing musical data stored in a relational database designed for educational purposes. The application allows users to interact with the database through CRUD operations on entities such as albums, musicians, and their relationships.

Built with .NET Framework and Windows Forms for the graphical user interface and MySQL for the database, the project focuses on demonstrating the fundamentals of desktop application development and database management. The application is structured to ensure ease of use, making it accessible for both beginners and advanced users looking to explore database integration in desktop applications. Additionally, the interface prioritizes clarity and usability for a seamless user experience.

1. DATABASE STRUCTURE

The relational database is implemented in MySQL and consists of the following tables:

Albums: Stores details about albums, including their titles and release years.

Musicians: Contains information about musicians, including their names and music genres.

Albums\_Musicians: A junction table representing the many-to-many relationship between albums and musicians.

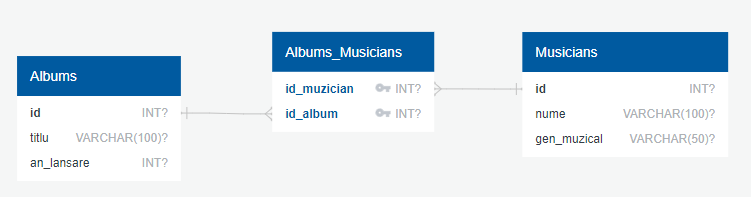


Fig. 1. Database diagram

1. Implementation

The application is structured around .NET Framework with Windows Forms for the user interface and MySQL for the backend database. CRUD functionalities are implemented via event handlers in the Windows Forms application, connecting user actions to database operations.

* 1. *Functionalities*

The Music Management Application provides a simple and intuitive graphical interface for managing musical data stored in the database. Users can perform the following operations:

**1. Musicians Management**

* View the list of musicians.
* Add new musicians.
* Edit musician details.
* Delete musicians, ensuring database integrity.

**2. Albums Management**

* View the list of albums.
* Add new albums.
* Edit album details.
* Delete albums, ensuring database integrity.

**3. Relationships Management**

* View the relationships between musicians and albums.
* Add a relationship between a musician and an album.
* Delete relationships.
  1. *Code Structure*

In .NET, event-driven programming links user interface components to backend logic. Key functionalities include:

1. Database Connection: The MySqlConnection class is used to connect to the database.
2. Event Handlers: Each button click triggers specific CRUD operations. For example:

* Adding a musician involves inserting data into the Musicians table.
* Deleting a musician removes related entries in the Albums\_Musicians table to maintain referential integrity.

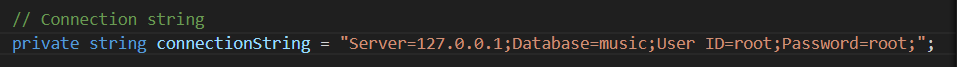


Fig. 2. Connection string

* 1. *User Interface*

The graphical interface is built using Windows Forms, providing:

1. ListBox components to display data from tables
2. TextBox components for user input.
3. Buttons for triggering CRUD operations.
4. Clear labels and placeholders to guide users.
5. Results

The application was tested for all CRUD operations, ensuring the correctness of database interactions and the integrity of data. Key test cases include:

**1. Adding a Musician**

* **Input:** Name = "Eminem", Genre = "Rap".
* **Steps:** Enter details in the TextBox and click "Add Musician".
* **Result:** The new musician is added to the list and the database.

**2. Deleting an Album**

* **Input:** ID = 3 (Album "Revival").
* **Steps:** Select the album and click "Delete Album"
* **Result:** The album is removed from the list and the database.

**3. Creating a Relationship**

* **Input:** Musician = "Eminem", Album = "The Marshall Mathers LP".
* **Steps:** Select the musician and album, then click "Create Relationship".
* **Result:** The relationship is added to the Albums\_Musicians table.

**4. Modifying a Musician**

* **Input:** Name = "El Nino", Genre = "Hip Hop"
* **Steps:** Select the wanted artist, enter the new details in the TextBox and click "Modify Musician".
* **Result:** Our desired musician has now the changed attributes.

All test cases were executed successfully, and the application performed as expected, maintaining database integrity and providing a seamless user experience.

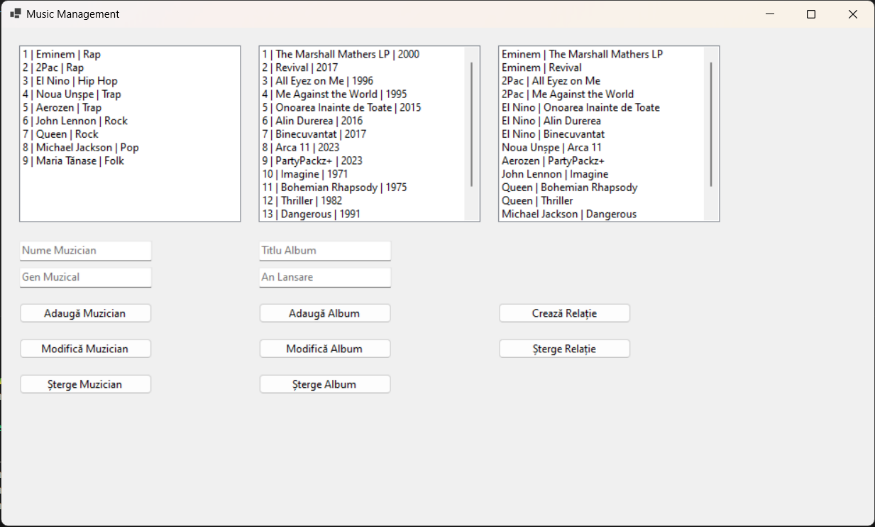


Fig. 3. Main interface

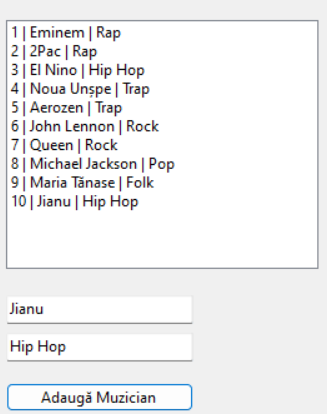
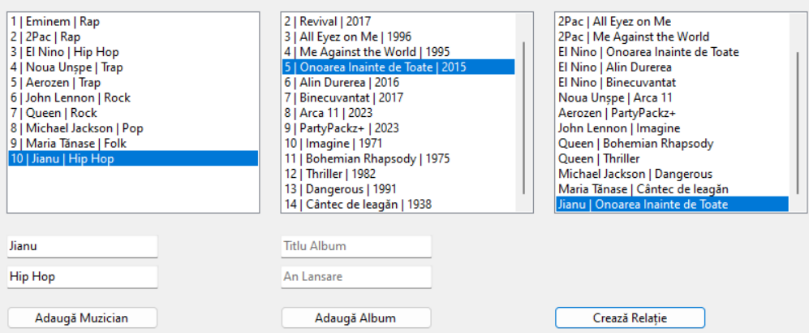


Fig. 4. Adding a Musician

Fig. 5. Creating a Relationship

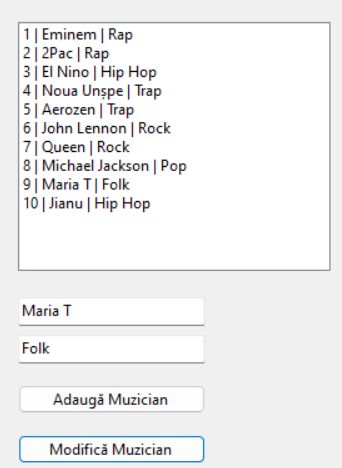


Fig. 6. Modifying a Musician

1. Conclusion

The Music Management Application successfully demonstrates CRUD operations on a MySQL database through a .NET Framework desktop application. The interface is user-friendly and provides essential functionalities such as adding, viewing, editing, and deleting data. While the project meets its initial objectives, future enhancements could include:

* Advanced filtering and search capabilities.
* Enhanced UI styling with modern design principles.
* Additional features like export and reporting tools.

References

[Fig. 1] <https://app.quickdatabasediagrams.com/#/>

MySQL Documentation, "MySQL 8.0 Reference Manual," 2023. [Online]. Available: <https://dev.mysql.com/doc/>.

.NET Documentation, "Windows Forms Overview," 2023. [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/>.

MySql.Data Library, "MySQL Connector for .NET," 2023. [Online]. Available: <https://dev.mysql.com/downloads/connector/net/>.

Microsoft Documentation, "Getting Started with Windows Forms in Visual Studio," 2023. [Online]. Available: <https://learn.microsoft.com/en-us/visualstudio/ide/getting-started-windows-forms>.