**DBI202 - Assignment**

**Group 4 – Store Management**

**Department of Software Engineering**

**FPT University**

Major Professor: *Trần Lâm Quân*

**Member:**

|  |  |
| --- | --- |
| **Name** | **ID** |
| Khuất Quang Hưng | HE190895 |
| Vũ Hải Nam |  |
| Minh |  |

1. **Presentation of problem:**

Managing the operations of a bookstore can be challenging, especially for chain stores handling large inventories and multiple locations. To address these challenges, our group has developed a Bookstore Management System to help streamline the operations of a local bookstore chain.

The system was designed to efficiently manage inventory, sales, discounts, author collaborations, and employee roles, ensuring smooth day-to-day operations.

1. **Entities and Their Functions**

- Publishers: Represents the entities providing books to the bookstore.  
- Titles (Books): Books sourced from specific publishers.  
- Authors: Represents the writers of the books; books can have multiple authors.  
- TitleAuthor: Junction table for many-to-many relationships between authors and titles.  
- Stores: Represents individual stores where books are sold.  
- Sales: Tracks sales transactions.  
- Discounts: Stores promotional discounts on book titles.  
- Jobs: Defines employee roles in the bookstore.  
- Employees: Represents employees and their store assignments.

1. **ER model of the system:**
2. **Relational model (moving from ER model):**

The database is normalized to the Third Normal Form (3NF) to ensure no redundancy and data integrity.   
Functional dependencies between attributes are identified, and all relations are structured to eliminate partial and transitive dependencies.

1. **Data requirements specification (data dictionary) and List of data constrain:**

**Publishers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Data Type** | **Length** | **Value** |
| |  | | --- | | pub\_id |  |  | | --- | |  | | |  | | --- | | The publisher identifier (PK) |  |  | | --- | |  | | |  | | --- | | INT |  |  | | --- | |  | | - | |  | | --- | | NOT NULL, UNIQUE |  |  | | --- | |  | |
| |  | | --- | | pub\_name |  |  | | --- | |  | | |  | | --- | | The name of the publisher |  |  | | --- | |  | | |  | | --- | | NVARCHAR |  |  | | --- | |  | | 100 | NOT NULL |

**Authors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Data Type** | **Length** | **Value** |
| |  | | --- | | au\_id |  |  | | --- | |  | | |  | | --- | | The author identifier (PK) |  |  | | --- | |  | | |  | | --- | | INT |  |  | | --- | |  | | - | NOT NULL  UNIQUE |
| |  | | --- | | au\_name |  |  | | --- | |  | | |  | | --- | | The name of the author |  |  | | --- | |  | | NVARCHAR | 100 | NOT NULL |

**TitleAuthor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Data Type** | **Length** | **Value** |
| |  | | --- | | au\_id |  |  | | --- | |  | | |  | | --- | | The author identifier (FK) |  |  | | --- | |  | | INT | - | |  | | --- | | FOREIGN KEY, NOT NULL |  |  | | --- | |  | |
| |  | | --- | | title\_id |  |  | | --- | |  | | |  | | --- | | The book identifier (FK) |  |  | | --- | |  | | INT | - | |  | | --- | | FOREIGN KEY, NOT NULL |  |  | | --- | |  | |
| |  | | --- | | Primary Key |  |  | | --- | |  | | |  | | --- | | Composite key of au\_id and title\_id |  |  | | --- | |  | |  | - |  |

**Titles (Books)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Data Type** | **Length** | **Value** |
| |  | | --- | | title\_id |  |  | | --- | |  | | |  | | --- | | The book identifier (PK) |  |  | | --- | |  | | VARCHAR | - | NOT NULL  UNIQUE |
| |  | | --- | | title\_name |  |  | | --- | |  | | |  | | --- | | The name of the book |  |  | | --- | |  | | NVARCHAR | 200 | |  | | --- | | NOT NULL |  |  | | --- | |  | |
| |  | | --- | | pub\_id |  |  | | --- | |  | | |  | | --- | | The associated publisher (FK) |  |  | | --- | |  | | INT | - | |  | | --- | | FOREIGN KEY, NOT NULL | |

**Stores**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Data Type** | **Length** | **Value** |
| |  | | --- | | stor\_id |  |  | | --- | |  |      |  | | --- | |  | | |  | | --- | | The store identifier (PK) |  |  | | --- | |  | | INT | - | NOT NULL  UNIQUE |
| |  | | --- | | stor\_name |  |  | | --- | |  | | |  | | --- | | The name of the store |  |  | | --- | |  | | |  | | --- | | NVARCHAR |  |  | | --- | |  | | |  | | --- | | 100 |  |  |  | | --- | --- | |  |  | | NOT NULL |

**Sales**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Data Type** | **Length** | **Value** |
| |  | | --- | | sale\_id |  |  | | --- | |  | | |  | | --- | | The sale transaction identifier (PK) |  |  | | --- | |  | | INT | - | |  | | --- | | NOT NULL, UNIQUE |  |  | | --- | |  | |
| |  | | --- | | stor\_id |  |  | | --- | |  | | |  | | --- | | The store where the sale occurred (FK) |  |  | | --- | |  | | INT | - | |  | | --- | | FOREIGN KEY, NOT NULL |  |  | | --- | |  | |
| |  | | --- | | title\_id |  |  | | --- | |  | | |  | | --- | | The book sold (FK) |  |  | | --- | |  | | INT | - | |  | | --- | | FOREIGN KEY, NOT NULL |  |  | | --- | |  | |
| sale\_date | |  | | --- | | DATE |  |  | | --- | |  | |  | - | |  | | --- | | NOT NULL |  |  | | --- | |  | |
| |  | | --- | | quantity\_sold |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | INT |  |  | | --- | |  | |  |  | | --- | |  | |  | - | NOT NULL,  CHECK > 0 |

**Discounts**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Data Type** | **Length** | **Value** |
| discount\_id | The discount identifier (PK) | INT | - | NOT NULL, UNIQUE |
| stor\_id | The store offering the discount (FK) | INT | - | FOREIGN KEY, NOT NULL |
| discount\_value | The value of the discount | DECIMAL | 5, 2 | NOT NULL, CHECK > 0 |

**Jobs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Data Type** | **Length** | **Value** |
| job\_id | The job role identifier (PK) | INT | - | NOT NULL, UNIQUE |
| job\_title | The title of the job role | NVARCHAR | 100 | NOT NULL |

**Employee**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Element** | **Description** | **Data Type** | **Length** | **Value** |
| emp\_id | The employee identifier (PK) | INT | - | NOT NULL, UNIQUE |
| emp\_name | The name of the employee | NVARCHAR | 100 | NOT NULL |
| emp\_phone | The employee's phone number | VARCHAR | 10 | NOT NULL, UNIQUE |
| emp\_address | The address of the employee | NVARCHAR | 200 | CAN NULL |
| job\_id | The job role of the employee (FK) | INT | - | FOREIGN KEY, NOT NULL |
| stor\_id | The store the employee works at (FK) | INT | - | FOREIGN KEY, NOT NULL |

1. **SQL Implementation**

**4.** **SQL Implementation**

The database was created using SQL Server with the following components:

[createDB.sql](https://github.com/hygef-v4/DBI202/blob/main/dbi_ass/createDB.sql)

**4.1 Alter Table Constraints**

- **Mandatory Job Assignment for Employees**  
Ensures that every employee has a defined job by making the job\_id field mandatory (non-null).

**- Unique Book Titles**  
Enforces uniqueness in the title\_name column so that each book title is distinct, preventing duplicate entries.

- **Discount Value Limit**  
Restricts discount values to a maximum of 100%, ensuring that discounts do not exceed allowable limits.

[constraints.sql](https://github.com/hygef-v4/DBI202/blob/main/dbi_ass/constraints.sql)

**4.2 Data Insertion**

At least five records were inserted for each table to test the database functionality.

[insert.sql](https://github.com/hygef-v4/DBI202/blob/main/dbi_ass/insert.sql)

**4.3 Queries**  
The following types of queries were implemented:  
a. Inner Join  
b. Outer Join  
c. Subquery in WHERE clause  
d. Subquery in FROM clause  
e. Group By with aggregate functions  
[queries.sql](https://github.com/hygef-v4/DBI202/blob/main/dbi_ass/queries.sql)

**4.4 Transactions, Triggers, and Procedures**

- A sample transaction was implemented with rollback functionality.  
- A trigger was set up to enforce specific business rules upon data modification.  
- Stored procedures were created to automate common database operations.  
[transaction.sql](https://github.com/hygef-v4/DBI202/blob/main/dbi_ass/transaction.sql)

[trigger.sql](https://github.com/hygef-v4/DBI202/blob/main/dbi_ass/trigger.sql)

[procedure.sql](https://github.com/hygef-v4/DBI202/blob/main/dbi_ass/procedure.sql)

1. **Conclusion Implementation**

The Bookstore Management System efficiently manages core bookstore operations.   
This project demonstrates the practical application of database design, normalization, and SQL implementation.