``Arian Kazemi`` 181613217 7/21/2023

Database Design for Online Bookstore (UNF and 1NF)

UNF (Un-Normalized Form) Database Design:

We can start by identifying the primary entity, which is the "Book" entity, along with its attributes:

Book (ISBN, Title, Author(s), Publication Date, Genre, Publisher, Format, Price, Availability Status, Number of Pages, Customer Reviews, Categories, Editions)

In this UNF design, we have included all the attributes related to a book, including multiple formats, customer reviews, categories, and editions. However, this design is not in a normalized form, as it may contain repeating groups.

Handling Repeating Groups:

To handle the repeating groups, we will create separate entities for the attributes that can have multiple occurrences.

Format Entity:

Format (ISBN, FormatType, Price, Availability Status, Number of Pages)

In this entity, the "ISBN" will act as the primary key, and "FormatType" will indicate the type of format (e.g., hardcover, paperback, ebook).

Customer Review Entity:

CustomerReview (ISBN, Rating, Comments, Reviewer's Name)

Here, the "ISBN" will act as the primary key, and each book can have multiple customer reviews.

Category Entity:

Category (ISBN, CategoryName)

Again, the "ISBN" will act as the primary key, and each book can belong to multiple categories.

Edition Entity:

Edition (ISBN, EditionNumber, EditionDate, Description)

In this entity, the combination of "ISBN" and "EditionNumber" will form the primary key, allowing each book to have multiple editions with unique details.

1NF (First Normal Form) Database Design:

After handling the repeating groups, the database is now in 1NF. All attributes are atomic, and each entity has a primary key.

Entities in 1NF:

Book (ISBN, Title, Author(s), Publication Date, Genre, Publisher)

Format (ISBN, FormatType, Price, Availability Status, Number of Pages)

CustomerReview (ISBN, Rating, Comments, Reviewer's Name)

Category (ISBN, CategoryName)

Edition (ISBN, EditionNumber, EditionDate, Description)

Relationships:

Book can have multiple formats (1 to Many relationship with Format, using ISBN as a foreign key in Format).

Book can have multiple customer reviews (1 to Many relationship with CustomerReview, using ISBN as a foreign key in CustomerReview).

Book can belong to multiple categories (1 to Many relationship with Category, using ISBN as a foreign key in Category).

Book can have multiple editions (1 to Many relationship with Edition, using ISBN as a foreign key in Edition).

Final Deliverables:

In the DBDL format, the final 1NF solution will look like this:

-- Create Book table

CREATE TABLE Book (

ISBN VARCHAR(20) PRIMARY KEY,

Title VARCHAR(255),

Authors VARCHAR(255),

PublicationDate DATE,

Genre VARCHAR(100),

Publisher VARCHAR(100)

);

-- Create Format table

CREATE TABLE Format (

ISBN VARCHAR(20),

FormatType VARCHAR(50),

Price DECIMAL(10, 2),

AvailabilityStatus VARCHAR(50),

NumberOfPages INT,

PRIMARY KEY (ISBN, FormatType),

FOREIGN KEY (ISBN) REFERENCES Book(ISBN)

);

-- Create CustomerReview table

CREATE TABLE CustomerReview (

ISBN VARCHAR(20),

Rating INT,

Comments TEXT,

ReviewerName VARCHAR(100),

PRIMARY KEY (ISBN, Rating),

FOREIGN KEY (ISBN) REFERENCES Book(ISBN)

);

-- Create Category table

CREATE TABLE Category (

ISBN VARCHAR(20),

CategoryName VARCHAR(100),

PRIMARY KEY (ISBN, CategoryName),

FOREIGN KEY (ISBN) REFERENCES Book(ISBN)

);

-- Create Edition table

CREATE TABLE Edition (

ISBN VARCHAR(20),

EditionNumber INT,

EditionDate DATE,

Description TEXT,

PRIMARY KEY (ISBN, EditionNumber),

FOREIGN KEY (ISBN) REFERENCES Book(ISBN)

);

Explanations of Decisions:

The decisions made during the normalization process were aimed at eliminating repeating groups and ensuring that each attribute is atomic. We created separate entities for attributes that can have multiple occurrences, such as formats, customer reviews, categories, and editions. Foreign keys were used to establish relationships between these entities and the main "Book" entity, allowing for efficient data organization and retrieval.

This normalized design improves data integrity, reduces data duplication, and allows for more flexibility in handling data in the future. The use of primary keys and foreign keys ensures data consistency and enables proper indexing for faster query execution.

Overall, the 1NF design provides a solid foundation for building a scalable and well-structured database for the online bookstore, meeting the requirements of data organization and integrity.