Project Documentation

Table of Contents

[Introduction 2](#_Toc100612139)

[Create a Database - BookRentalDetails 2](#_Toc100612140)

[Database Diagram 3](#_Toc100612141)

[Normalization Techniques 4](#_Toc100612142)

[First Normal Form (1NF) 4](#_Toc100612143)

[Second Normal Form (2NF) 4](#_Toc100612144)

[Third Normal Form (3NF) 5](#_Toc100612145)

[Constraints 6](#_Toc100612146)

[Table 1: Student\_UserDetails 6](#_Toc100612147)

[Table 2: GenderInfo 7](#_Toc100612148)

[Table 3: Books 7](#_Toc100612149)

[Views 8](#_Toc100612150)

[View 1: [dbo].[BookView\_Info] 8](#_Toc100612151)

[Output of [dbo].[BookView\_Info] 9](#_Toc100612152)

[View 2: [dbo].[GenderView\_Info] 9](#_Toc100612153)

[Output of [dbo].[GenderView\_Info] 9](#_Toc100612154)

[Stored procedures 10](#_Toc100612155)

[Stored Procedure 1: [dbo].[SP\_BookSearch] 10](#_Toc100612156)

[Output of [dbo].[SP\_BookSearch] 10](#_Toc100612157)

[Stored Procedure 2: [dbo].[SP\_InsertBookRecord] 11](#_Toc100612158)

[Output of[dbo].[SP\_InsertBookRecord] 12](#_Toc100612159)

[Functions 12](#_Toc100612160)

[Md5\_GeneratorFunction 12](#_Toc100612161)

[Triggers 13](#_Toc100612162)

[Trigger 1: [dbo].[alert] 13](#_Toc100612163)

[Trigger 2: [insert\_into\_library] 14](#_Toc100612164)

# Introduction

The main objective of this project is to design and implement database of students’ details and books they rent. This Data model is designed using database diagrams. This project contains key constraints, entity integrity, referential integrity constraints that are implemented on tables. The SQL objects that are created are views, stored procedures, functions, and triggers.

# Create a Database - BookRentalDetails

Graphical user interface, application, Word

Description automatically generated

**Figure 1BookRentalDetails Database**

Here, created a database named BookRentalDetails.

Graphical user interface

Description automatically generated with low confidence

# Database Diagram

Diagram

Description automatically generated with medium confidence

**Figure 2 Entity-Relationship: BookRentalDetails**

The above Entity Relation - Diagram has 3 entities. Such as Student\_UserDetails, Books, GenderInfo. Every entity has its own attributes.

**Student\_UserDetails** entity has the following attributes.

* GSU\_ID
* FullName
* PhysicalAddress
* Gender\_ID

**Books** entity has the following attributes

* GSU\_ID
* BookRented

**GenderInfo** has the following attributes

* Gender\_ID
* Gender

**Relationship between** **Student\_UserDetails and Books**

The two entities Student\_UserDetailsandBooksshare one to many relationships.

**Relationship between** **Student\_UserDetails and GenderInfo**

The two entities Student\_UserDetailsand GenderInfoshare one to many relationships.

# Normalization Techniques

Normalization is a process of reducing the data redundancy and it also maintains consistency in the database.

|  |  |  |  |
| --- | --- | --- | --- |
| **FullName** | **PhysicalAddress** | **BookRented** | **Gender** |
| Joe Root | 1st street, Park Forest | Pride and Prejudice | Male |
| Ariel Chappel | 3rd Ave, Richton Park | The Alchemist | Female |
| Tia Star | 5th street, Chicago Heights | Gone Girl | Female |
| Marcus Warne | 7th street, Oak Forest | The Song of Achilles | Male |
| Keri Cook | 9th street, Tinley Park | The Fault in Our Stars | Prefer Not to Say |

**Unnormalized Data**

## First Normal Form (1NF)

In the first normal form, each cell should have single value and every record should be unique.

|  |  |  |  |
| --- | --- | --- | --- |
| FullName | PhysicalAddress | BookRented | Gender |
| Joe Root | 1st street, Park Forest | Pride and Prejudice | Male |
| Joe Root | 1st street, Park Forest | Gone Girl | Male |
| Ariel Chappel | 3rd Ave, Richton Park | The Alchemist | Female |
| Ariel Chappel | 3rd Ave, Richton Park | The Fault in Our Stars | Female |
| Tia Star | 5th street, Chicago Heights | Gone Girl | Female |
| Tia Star | 5th street, Chicago Heights | The Alchemist | Female |
| Marcus Warne | 7th street, Oak Forest | The Song of Achilles | Male |
| Keri Cook | 9th street, Tinley Park | The Fault in Our Stars | Prefer Not to Say |

## Second Normal Form (2NF)

2NF should be in 1NF and columns functionally or partially dependent on primary key.

Primary key

|  |  |  |  |
| --- | --- | --- | --- |
| GSU\_ID | FullName | PhysicalAddress | Gender |
| 1001 | Joe Root | 1st street, Park Forest | Male |
| 1002 | Ariel Chappel | 3rd Ave, Richton Park | Female |
| 1003 | Tia Star | 5th street, Chicago Heights | Female |
| 1004 | Marcus Warne | 7th street, Oak Forest | Male |
| 1005 | Keri Cook | 9th street, Tinley Park | Prefer Not to Say |

***Table 1: Student Information***

Primary Key

|  |  |
| --- | --- |
| GSU\_ID | BookRented |
| 1001 | Pride and Prejudice |
| 1001 | Gone Girl |
| 1002 | The Alchemist |
| 1002 | The Fault in Our Stars |
| 1003 | Gone Girl |
| 1003 | The Alchemist |
| 1004 | The Song of Achilles |
| 1005 | The Fault in Our Stars |

***Table 2: Books Rented***

Here, the first normal form (1NF) table is divided into two tables.

Table 1: Student Information

Table 2: Books rented

GSU\_ID is a new column which acts as the ‘primary key’.

## Third Normal Form (3NF)

3NF should be in 2NF and it should not contain any transitive dependencies.

|  |  |  |  |
| --- | --- | --- | --- |
| GSU\_ID | FullName | PhysicalAddress | Gender\_ID |
| 1001 | Joe Root | 1st street, Park Forest | 1 |
| 1002 | Ariel Chappel | 3rd Ave, Richton Park | 2 |
| 1003 | Tia Star | 5th street, Chicago Heights | 2 |
| 1004 | Marcus Warne | 7th street, Oak Forest | 1 |
| 1005 | Keri Cook | 9th street, Tinley Park | 3 |

***Table 1: Student\_UserDetails***

|  |  |
| --- | --- |
| GSU\_ID | BookRented |
| 1001 | Pride and Prejudice |
| 1001 | Gone Girl |
| 1002 | The Alchemist |
| 1002 | The Fault in Our Stars |
| 1003 | Gone Girl |
| 1003 | The Alchemist |
| 1004 | The Song of Achilles |
| 1005 | The Fault in Our Stars |

***Table 2: Book***

**Primary key**

|  |  |
| --- | --- |
| Gender\_ID | Gender |
| 1 | Male |
| 2 | Female |
| 3 | Prefer Not to Say |

***Table 3: GenderDetails***

The 3NF table consists of three tables.

Table 1: It contains student details.

Table 2: It has information about the rented books.

Table 3: It contains gender details along with gender ID that acts as primary key**.**

# Constraints

Constraints specify rules for data in a table. Primary keys, foreign keys, check constraints, and not null are few important constraints.

## Table 1: Student\_UserDetails

|  |  |  |  |
| --- | --- | --- | --- |
| GSU\_ID | FullName | PhysicalAddress | Gender\_ID |
| 1001 | Joe Root | 1st street, Park Forest | 1 |
| 1002 | Ariel Chappel | 3rd Ave, Richton Park | 2 |
| 1003 | Tia Star | 5th street, Chicago Heights | 2 |
| 1004 | Marcus Warne | 7th street, Oak Forest | 1 |
| 1005 | Keri Cook | 9th street, Tinley Park | 3 |

**Table 1: Student\_UserDetails**

Graphical user interface, text

Description automatically generated

This Table consists of following Constraints

1. Key Constraints**:** GSU\_ID(PK), FullName, PhysicalAddress
2. Entity Integrity Constraint: GSU\_ID(PK) IS NOT NULL and Unique
3. Referential Integrity**:**
4. Check Constraints: FullName should only contain alphabets.

## Table 2: GenderInfo

|  |  |
| --- | --- |
| Gender\_ID | Gender |
| 1 | Male |
| 2 | Female |
| 3 | Prefer Not to Say |

**Table 2: GenderInfo**

Graphical user interface, text, application

Description automatically generated

This Table consists of following Constraints

1. Key Constraint: Gender\_ID (PK)[Gender]
2. Entity Integrity Constraint: Gender\_ID (PK) IS NOT NULL and Unique
3. Referential Integrity:

## Table 3: Books

|  |  |
| --- | --- |
| GSU\_ID | BookRented |
| 1001 | Pride and Prejudice |
| 1001 | Gone Girl |
| 1002 | The Alchemist |
| 1002 | The Fault in Our Stars |
| 1003 | Gone Girl |
| 1003 | The Alchemist |
| 1004 | The Song of Achilles |
| 1005 | The Fault in Our Stars |

**Table 3: Books**

Text

Description automatically generated

This Table consists of following Constraints

1. Key Constraint: GSU\_ID (PK) [BookRented]
2. Entity Integrity Constraint: GSU\_ID(PK) IS NOT NULL and Unique
3. Referential Integrity:

# Views

A view is a virtual table based on the result set of an SQL statement.

In this project, two views have been created.

They are [dbo].[BookView\_Info] and [dbo].[GenderView\_Info].

## View 1: [dbo].[BookView\_Info]

Graphical user interface, text, application

Description automatically generated

**Figure 3 BookView\_Info**



### Output of [dbo].[BookView\_Info]

Text, table

Description automatically generated

**Figure 4 Output of BookView\_Info**

The BookView\_Info view describes the full name and the books rented by the student.

## View 2: [dbo].[GenderView\_Info]

Graphical user interface, text, application

Description automatically generated

**Figure 5 GenderView\_Info**



### Output of [dbo].[GenderView\_Info]

Table

Description automatically generated

**Figure 6 Output of GenderView\_Info**

The GenderView\_Info view illustrates the number of books rented by each gender.

# Stored procedures

A stored procedure is an SQL code, which can be reused any number of times.

In this project, two stored procedures named [dbo].[SP\_BookSearch] and [dbo].[SP\_InsertBookRecord] have been created.

## Stored Procedure 1: [dbo].[SP\_BookSearch]

Graphical user interface, text, application, email

Description automatically generated

**Figure 7 SP\_BookSearch**

### Output of [dbo].[SP\_BookSearch]

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, application, table

Description automatically generated

**Figure 8 Output of SP\_BookSearch**

Stored Procedure 2: [dbo].[SP\_InsertBookRecord]

Graphical user interface, text, application

Description automatically generated

**Figure 9 SP\_InsertBookRecord**

### Output of[dbo].[SP\_InsertBookRecord]

Graphical user interface, text, application

Description automatically generated



Table

Description automatically generated

**Figure 10 Output of InsertBookRecord**

Here, new book named ‘Intelligent Investor’ has been updated for the ID 1001.

# Functions

To use sql code repeatedly, function can be used. In this project, one function named [dbo].[Md5\_GeneratorFunction] has been created.

Graphical user interface, text, application

Description automatically generated

**Figure 11 Md5\_GeneratorFunction**

### Md5\_GeneratorFunction

A picture containing text

Description automatically generated

Text

Description automatically generated

**Figure 12 Output of Md5\_GeneratorFunction**

# Triggers

A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server. Triggers fire when any valid event fires.

In this project two triggers [dbo].[insert\_into\_library] and [dbo].[alert] have been implemented.

The [alert] trigger raises a message when a new record is inserted or updated.

## Trigger 1: [dbo].[alert]

Graphical user interface, text

Description automatically generated

**Figure 13 trigger\_alert**

Graphical user interface, application

Description automatically generated

## Trigger 2: [insert\_into\_library]

Graphical user interface, text, application

Description automatically generated

**Figure 14 trigger\_insert\_into\_library**

Whenever a new record (GSU\_ID, FullName) is updated in [dbo].[Student\_UserDetails] table, the [dbo].[Library] will be triggered.

Text

Description automatically generated with medium confidence

**Figure 15 [dbo].[Library]**

Graphical user interface, table

Description automatically generated

Updated [dbo].[Library]