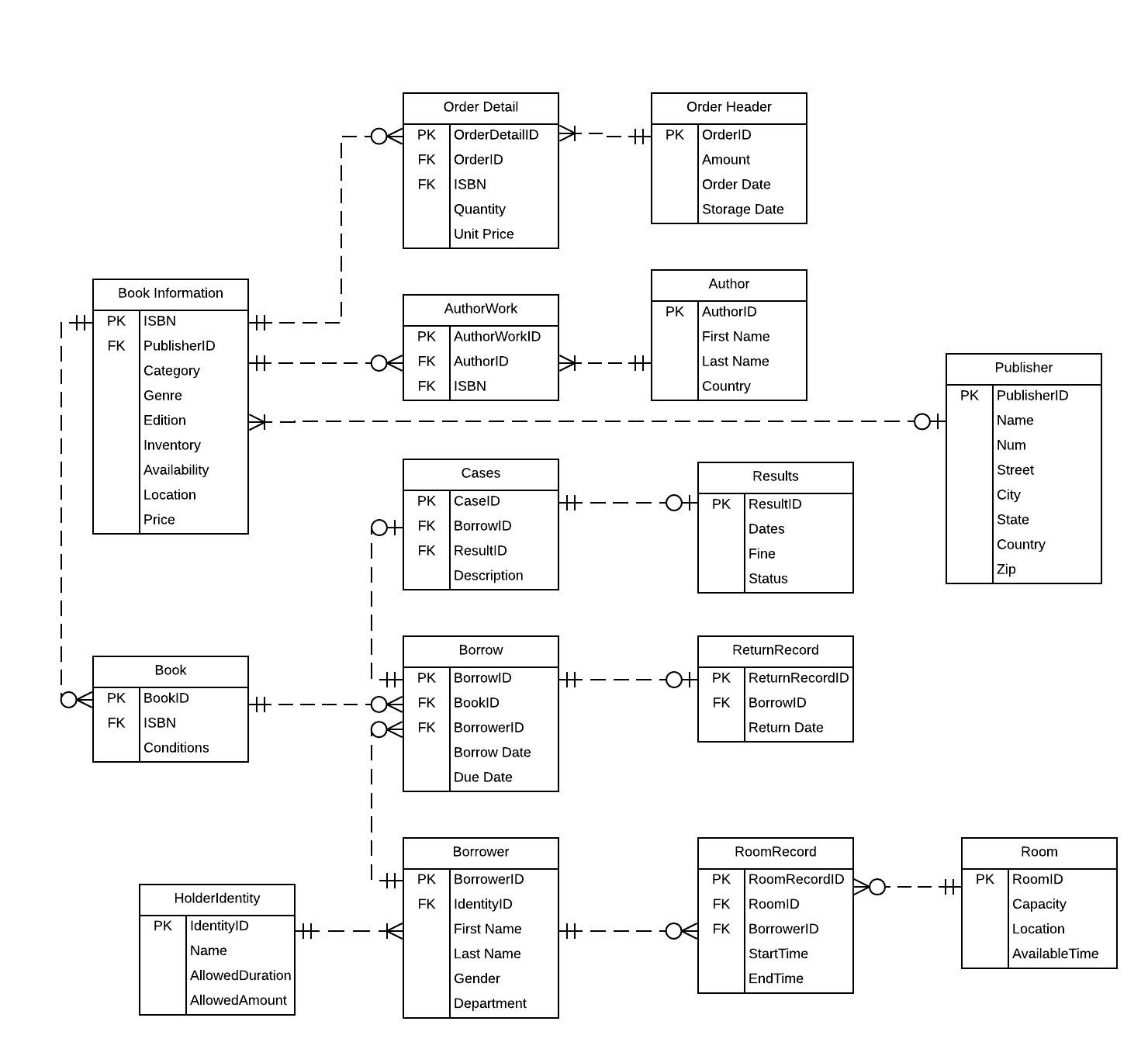
**University Library Management Database**



**Database Purpose:**

The purpose of this database is to maintain the data used to track and report the books lending situation, purchase information of the books, library rooms reservation, borrowers’ individual information and constraints. The database will be used by the library administrator, students and teachers.

**Business Problem Addressed:**

1. Allow the library administrator to generate descriptive reports about book information.
2. Allow the library administrator to browse, manage, maintain and modify the book information.
3. Permit students and teachers to check and modify individual information and library records.
4. Provide an orderly schedule for book check-out and room reservation to avoid the clash.
5. Permit students to check whether they should pay penalties for beyond return of the book.

**Business Rules:**

1. Each borrow record can only related to one particular book
2. Each room record can only relate to one particular booking.
3. Only person who has borrowed books will be tracked in the system as a borrower
4. Only authors who have works in the database can be added as an author.
5. Borrowers who have unpaid cases can not create new borrow requests.

**Design Decisions:**

|  |  |  |
| --- | --- | --- |
| **Entity Name** | **Why Entity Included** | **How Entity is Related to Other Entities** |
| Book Information | One of the basic purposes of the database is to store the information of the books in the library stock. This entity is used to store the information of each book with its’ unique ISBN, so that both readers and admins can search the book in the library system by any of the attributes, such as title, category, or author. The Availability, location, and inventory information are also collected and updated. | Because there are multiple books with the same information in library stock, so the book information entity is related to the book entity(real books in stock) by its primary key ISBN. And it is also related to the purchase order entity by ISBN too, so that the system can update the inventory of each book. And it is also related to author and publisher information by foreign key author ID and publisher ID. |
| OrderHeader | OrderHeader and OrderDetail entities are used to collect purchase order information. When the library buys new books, a purchase order data is added into the database, so that the admin can track the order status, cost and update the book inventory. | OrderHeader is related to OrderDetail by primary key OrderID. |
| OrderDetail | One OrderDetail record contains only one book, and one OrderHeader can have many OrderDetail. | This is related to OrderHeader by OrderID and related to Book Information by ISBN. |
| Author | Author Entity is used to store information of authors. One author can relate to multiple AuthorWork in the database, and admins can update author information, such as first name, last name, country and introduction. | Author is related to AuthorWork by primary key AuthorID. |
| AuthorWork | This entity stores the information of each author’s work on one book. In case of the books that have many authors, each author has a single AuthorWork record related to this book. | It is related to the Author by AuthorID and related to Book Information by ISBN. |
| Publisher | This entity stores the information of publishers. One publisher can publish multiple books, while one book can only have one publisher. The name and address of each publisher will be recorded and able to be updated as needed. | It is related to the Book Information by primary key PublisherID |
| Book | This entity records every single book itself as a property of the library. It is different from the Book Information that each record of Book Information records all the books with the same ISBN since the content of those books would be the same, while Book entity focuses on each single book itself and its physical condition. | It is related to the Book Information by ISBN, which indicates what book it is. It is also related to the Borrow entity by BookID, which allows the Borrow entity to track the specific book borrowed, and check the book condition upon return. |
| Borrow | This entity stores the information of borrow records. It tracks the borrower, the specific book borrowed, the borrow date, and due date. Each borrow record can only contain one book and one borrower. While returning, a return record will be created based on the borrow record, and if the book condition is not right upon return, a case will be filed using the borrow record. | It is related to the Borrower by BorrowerID, and to the Book by BookID, which indicates who borrowed which book. It is also related to the Return and Case by BorrowID during the returning process. |
| Cases | The case entity is to describe a case when an accident occurs, so that it is able to give a clear record about what is the event and who is involved in. And a case entity has a foreign key, which is related to the result of the case. | It is related to the Borrow entity by BorrowID, in order to track the accident for a specific borrowing activity. And it is related to the result entity by resultID, so that the users can check the result of the specified case. |
| Results | This entity stores the information of case results. Each result is only related to one particular case. It tracks the information of the Date it was filed, the Fine should be paid, and the processing status. | It is only related to the Case by primary key CaseID. There will be a result record only if there is a case filed. |
| Return | The Return entity is to record the return date for the book which is borrowed. If the return date of this book is given, the book will be set available. Then the other candidates for this book have access in order. | It is only related to Borrow entity by BorrowID, because only the borrower borrows the book, the return activity will occur. And the return record is always used to set the status of the book so that it can be borrowed next time. |
| Borrower | The Borrower entity is to collect demographic information of individuals, which are identified by their Borrower ID. When borrowing the books, the borrower’s information should be used for a piece of library record. And it is also used for room reservation records, so that the library administrator is able to browse and modify the records if necessary. | The Borrower entity is related to the Borrow entity as a crucial factor, since the records must include the borrower’s information so that the library administrator is able to inquiry and modify the schedule for the specific person. And the Borrower ID is unique so that the Borrow entity and the Room Record entity can use it to identify the specific individual. |
| HolderIdentity | The HolderIdentity entity is to give the constraint to the person who wants to borrow books or reserve rooms from the library. It has a primary key which is Identity ID. And it limits the duration and amount of the books for the borrower with its FKs. Then students and teachers should borrow books within the constraints. | Because the Borrower has the Identity ID in his individual information for specific constraint on this account. The Identity entity is able to distribute the constraint to the person by its Identity ID. |
| Room Record | The room record records every booking of the room. When booking the room, the information related should be recorded. When it starts and when it ends so that the library administrator can manage the room efficiently. Each record includes only one room and one borrower. | It is related to the foreign key roomID and borrowerID. With borrowerID, the administrator can check who makes the booking and tell them to come on time. With roomID, people can see the information of the room they have reserved before. |
| Room | The room entity provides the basic decomposition of the reservation. People could find if the room is meeting their demands, like individual study or group discussions. Also, It could help people find where the room is and if it is available at that time. | It is related to the Room Record by primary key Room ID. So that people can check if the room is available and find the room they want according to their needs. |