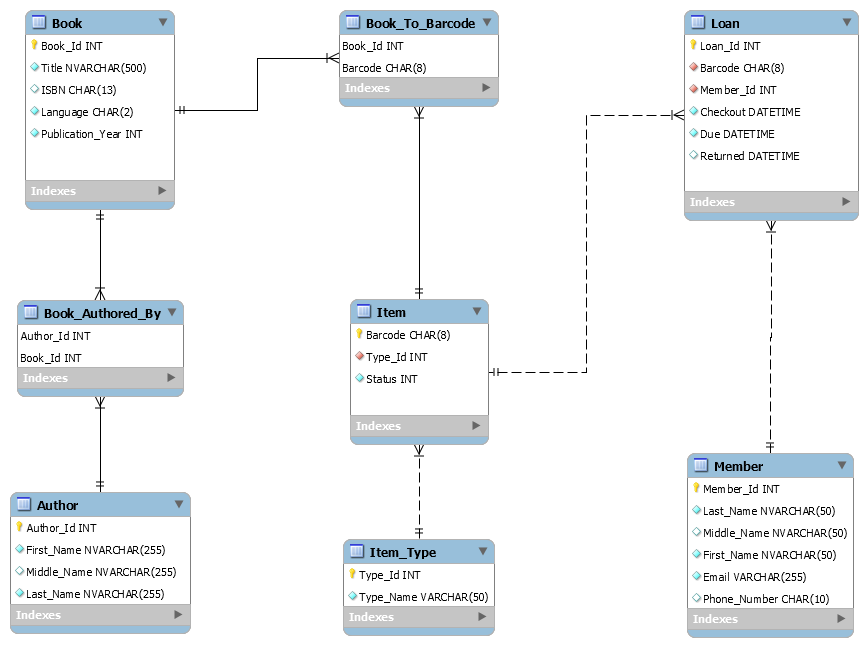
**FUBC Library Database Design Documentation**

**Purpose**

This document is written to help current and future database administrators become familiar with the design of the FUBC Library Database to facilitate database maintenance and any changes to the database that are necessary, as well as record design decision that were made along the way.

**ER Diagram**



**General Description**

The FUBC Library Database is meant to record the items the FUBC library has and allow library members to loan those items for a limited time. At the moment the only items being kept track of are of the type *Book* but the library anticipates to have items of other types such as CD’s. Each individual item will identified by an 8 digit barcode assigned to it at the time of addition.

Members could be identified by their member id’s assigned to them at sign up, or by their unique email address. Members will be able to loan arbitrary number of books for a period of 2 months each.

Only the basic required information is recorded about the books of which a great majority are written in either Ukrainian, Russian or are translated to those languages from English.

In the near future the library anticipates adding the ability to keep track of books by category.

**Tables**

**\*On delete or update if table depends on a foreign key that is being changed the change will be rejected(default behavior.) Can be changed if needed.**

**PK:** primary key **FK:** foreign key **NN:** non null **UQ:** unique **UN:** unsigned **AI:** autoincremented

*ITEM*:

The central table of the database meant to keep track of each individual item and it’s availability for checkout. The *Status* field is set to 1 by default and changed to 0 by a creation of a new entry in the **LOAN** table (that is when a member loans an item) or to 1 when the **LOAN** table is updated with a return date (that is when a member returns an item). The foreign key constrain Type\_Id ensures that only items of existing types can be added.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field\_Name** | **Datatype** | **PK** | **FK** | **NN** | **UQ** | **UN** | **AI** | **Default Expression** | **Additional Info** |
| *Barcode* | CHAR(8) | ✔ |  | ✔ | ✔ |  |  |  | Unique identifier of a single item. |
| *Type\_Id* | VARCHAR(50) |  | ✔ | ✔ | ✔ | ✔ |  |  | See Table ITEM\_TYPE |
| *Status* | INT |  |  | ✔ |  | ✔ |  | 1 | Indicates if an item is available (1) for checkout or not (0). |

*ITEM\_TYPE*:

Keeps track of different types of items in the library.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field\_Name** | **Datatype** | **PK** | **FK** | **NN** | **UQ** | **UN** | **AI** | **Default Expression** | **Additional Info** |
| *Type\_Id* | INT | ✔ |  | ✔ | ✔ | ✔ | ✔ |  | Unique identifier for the item types |
| *Type\_Name* | VARCHAR(50) |  |  | ✔ | ✔ |  |  |  |  |

*BOOK*:

Keeps track of book information. The ISBN field can be Null because many books in the library do not have an ISBN number. In the future a Category\_Id will field will be added to keep track of category a book belongs to and it can only belong to one.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field\_Name** | **Datatype** | **PK** | **FK** | **NN** | **UQ** | **UN** | **AI** | **Default Expression** | **Additional Info** |
| *Book\_Id* | INT | ✔ |  | ✔ | ✔ | ✔ | ✔ |  | Starts at 0 |
| *Title* | NVARCHAR(500) |  |  | ✔ | ✔ |  |  |  | NVARCHAR |
| *ISBN* | CHAR(13) |  |  |  |  |  |  |  |  |
| *Language* | CHAR(2) |  |  | ✔ |  |  |  |  | Language the book is written. |
| *Publication\_Year* | INT |  |  | ✔ |  | ✔ |  |  |  |

*BOOK\_TO\_BARCODE*:

Since the library can have multiple copies of the same book and we want to keep track of each individual copy it was necessary to create this table which ties each books barcode to its corresponding book id. (Book\_Id, Barcode) is the unique primary key pair.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field\_Name** | **Datatype** | **PK** | **FK** | **NN** | **UQ** | **UN** | **AI** | **Default Expression** | **Additional Info** |
| *Book\_Id* | INT | ✔ | ✔ | ✔ | ✔ | ✔ |  |  | See BOOK table. |
| *Barcode* | CHAR(8) | ✔ | ✔ | ✔ | ✔ |  |  |  | See ITEM table. |

*AUTHOR*:

Records the names of the authors in the database.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field\_Name** | **Datatype** | **PK** | **FK** | **NN** | **UQ** | **UN** | **AI** | **Default Expression** | **Additional Info** |
| *Author\_Id* | INT | ✔ |  | ✔ | ✔ | ✔ | ✔ |  |  |
| *First\_Name* | NVARCHAR(255) |  |  | ✔ |  |  |  |  | NVARCHAR |
| *Middle\_Name* | NVARCHAR(255) |  |  |  |  |  |  |  | NVARCHAR |
| *Last\_Name* | NVARCHAR(255) |  |  | ✔ |  |  |  |  | NVARCHAR |

*BOOK\_AUTHORED\_BY*:

Since a book can be authored by multiple authors and an author can write multiple books this table was created to break up the many to many relationship. (Author\_Id, Book\_Id) is the unique primary key pair.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field\_Name** | **Datatype** | **PK** | **FK** | **NN** | **UQ** | **UN** | **AI** | **Default Expression** | **Additional Info** |
| *Author\_Id* | INT | ✔ | ✔ | ✔ | ✔ | ✔ |  |  | See AUTHOR table. |
| *Book\_Id* | INT | ✔ | ✔ | ✔ | ✔ | ✔ |  |  | See BOOK table. |

*MEMBER*:

This table records various information about library members. Members could be identified by their unique id (starting at 0) assigned at registration or their unique email address. There could be duplicate Last\_Name, First\_Name combinations.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field\_Name** | **Datatype** | **PK** | **FK** | **NN** | **UQ** | **UN** | **AI** | **Default Expression** | **Additional Info** |
| *Member\_Id* | INT | ✔ |  | ✔ | ✔ | ✔ | ✔ |  | Starts at 0 |
| *Last\_Name* | NVARCHAR(50) |  |  | ✔ |  |  |  |  | NVARCHAR |
| *First\_Name* | NVARCHAR(50) |  |  | ✔ |  |  |  |  | NVARCHAR |
| *Email* | VARCHAR(255) |  |  | ✔ | ✔ |  |  |  |  |
| *Phone\_Number* | CHAR(10) |  |  |  |  |  |  |  |  |

*LOAN*:

Records which items are loaned to which members. Each item has a two months loan period. The *returned* field is empty by default but gets updated once the member returns the item on loan. The FK Barcode ensures that only existing items can be checked out. The FK Member\_Id ensure that only existing members can check out items.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Field\_Name** | **Datatype** | **PK** | **FK** | **NN** | **UQ** | **UN** | **AI** | **Default Expression** | **Additional Info** |
| *Loan\_Id* | INT | ✔ |  | ✔ | ✔ | ✔ | ✔ |  | Starts at 0 |
| *Barcode* | CHAR(8) |  | ✔ | ✔ |  |  |  |  | See ITEM table |
| *Member\_Id* | INT |  | ✔ | ✔ |  |  |  |  | See MEMBER table |
| *Checkout* | DATETIME |  |  | ✔ |  |  |  | Current Time | Set with a trigger that grabs the current time at checkout. |
| *Due* | DATETIME |  |  | ✔ |  |  |  | Current time + 2 months | Set with a trigger. |
| *Returned* | DATETIME |  |  |  |  |  |  |  | Set with a trigger that grabs the current time at return. |

**Assumptions**

Books

* We have multiple copies of the same book.
* There are at least 3 languages that the books in the library are written in.
* Majority of books are written in Russian or Ukrainian.
* A book could have multiple authors.
* Some books don’t have an ISBN number.
* Some books will be thrown away.
* There are books that are too old to be added.
* There is a possibility that two different books have the same title.
* A book must have an author.
* Book titles could be recorded in different languages.

Authors

* An author must have at least 1 book to be in the database.
* Authors could have a middle name.
* There could be multiple authors with the same name.
* Author names could be recorded in different languages.

Barcodes

* Barcodes are unique
* 8 digits long
* No check digit
* Start at 1

Items

* Items are of different types (books, cd’s…)
* Items are identified by a barcode.
* The library has an item or the item is checked out.
* If an item is removed from the library it is deleted.

Members

* Members have email addresses.
* If they don’t make one for them.
* Members could have the same names.
* Some members don’t have numbers.
* Members could have middle names.

Loans

* Members can loan items.
* Different items will have different loan times.
* Members return items that they loan.
* If they don’t and the loan period is expired, they should be contacted by email or phone.
* The loan and return dates are recorded.
* Items of type “Book” are loaned for periods of 2 months.

**Constraints**

Not Enforced by the Database (must be enforced programmatically)

* A book must have at least one author.
* An author must write at least one book.
* An item must have a status of 1 (available for checkout) to be checked out.
* A book does not have to have a barcode to be kept track of. There is a possibility that we want to keep track of books but not let members check them out.

Enforced by the Database

* An item must be of an existing type to be added to a database.
* An item can be of only one type.
* For an item to be loaned it must have a valid barcode.
* Items can only be loaned to existing members.

**Triggers**

set\_due\_date:

Before an insertion into the **LOAN** table the due date is set to current date + 2 months. Since the only item type the library cares about right now is “Book”, this is ok but it will have to be fixed in the future.

unavailable\_status

After insertion into the **LOAN** table the *Status* field in the **Item** table is set to 0 to recorded that the item is not available for check out.

available\_status

Does the opposite of the above trigger but on update to the **LOAN** table when a return date is added, that is when an items is returned it becomes available for checkout again.

**Edge Cases**

* Need to check that the book\_title + author combination is unique
  + Right now that is being done inside add\_book() method which checks that both the author and book\_title don’t exist. This needs to be improved.