
Team ORCHID

Library Database Project Physical Database Design

Table of Contents

Introduction.....	3
Project Overview.....	3
Scope	3
Platform	3
Physical Schema.....	3
Picklists	3
Account Type.....	4
Contribution Role.....	4
Genre.....	4
Item Type	4
Rating	4
Entity Tables	5
Account	5
Contributor	6
Item	6
Relationship Tables.....	7
Contribution	7
Loan	7
Reservation	8
Database Sample.....	8

Introduction

Document-specific summary here.

Project Overview

The Library Database Project aims to simplify data management and improve daily operations for a single library. This system helps librarians easily track inventory, manage user accounts, keep records of borrowed items, and create useful reports. Library staff and visitors will have appropriate access based on their roles, ensuring easy and organized interactions with the library's resources.

Scope

This database is designed for a single library, handling its catalog, user accounts, and loan history. It is not intended for multi-branch systems but provides essential features to support daily operations. The system enables structured and accessible interactions for all users, from staff to guests.

Platform

We chose to use the University of Kansas' cycle servers with MariaDB through phpMyAdmin, as provided by the course. This choice was based on a few factors.

1. *Requirements met.* While other database management systems met the requirements, MariaDB is open source and efficient, and as a relatively small project, we did not require the additional features that MySQL offered.
2. *Simplicity.* Since the database was already set up by the course and phpMyAdmin has a graphical user interface, this meant that team members could quickly understand the tool and how to use it.
3. *Ease of access.* Since we are hosted on the University of Kansas servers, we can easily access it on-campus as well as off-campus by using the KU Anywhere VPN.

Physical Schema

This section shows each of the data definition language (DDL) statements for the tables, as we did not have to create the DDL statements for the database itself.

Picklists

This section contains the DDL statements for all picklists.

Account Type

```
CREATE TABLE account_type(  
  account_type_id int(11) NOT NULL AUTO_INCREMENT,  
  account_type_name varchar(10) NOT NULL,  
  PRIMARY KEY (account_type_id));
```

Contribution Role

```
CREATE TABLE contribution_role(  
  role_id int(11) NOT NULL AUTO_INCREMENT,  
  role_name varchar(15) NOT NULL,  
  PRIMARY KEY (role_id),  
  UNIQUE KEY role_name (role_name));
```

Genre

```
CREATE TABLE genre(  
  genre_id int(11) NOT NULL AUTO_INCREMENT,  
  genre_name varchar(15) NOT NULL,  
  PRIMARY KEY (genre_id),  
  UNIQUE KEY genre_name (genre_name));
```

Item Type

```
CREATE TABLE item_type(  
  item_type_id int(11) NOT NULL AUTO_INCREMENT,  
  item_type_name varchar(10) NOT NULL,  
  PRIMARY KEY (item_type_id));
```

Rating

```
CREATE TABLE rating(  
  rating_id int(11) NOT NULL AUTO_INCREMENT,
```

```
rating_name varchar(5) NOT NULL,
PRIMARY KEY (rating_id),
UNIQUE KEY rating_name (rating_name));
```

Entity Tables

This section contains the DDL statements that define standalone entities, i.e. accounts, items, and contributors.

Account

```
CREATE TABLE account (
account_id int(11) NOT NULL AUTO_INCREMENT,
first_name varchar(35) NOT NULL,
last_name varchar(35) NOT NULL,
middle_initial char(1) DEFAULT NULL,
card_number int(13) NOT NULL,
account_type_id int(11) NOT NULL DEFAULT 1,
restricted tinyint(1) NOT NULL DEFAULT 0,
address varchar(50) DEFAULT NULL,
birthdate date NOT NULL,
email varchar(62) NOT NULL,
phone_number varchar(15) DEFAULT NULL,
parent_id int(11) DEFAULT NULL,
PRIMARY KEY (account_id),
UNIQUE KEY card_number (card_number),
UNIQUE KEY email (email),
KEY account_type_id (account_type_id),
KEY parent_id (parent_id),
CONSTRAINT valid_parent FOREIGN KEY (parent_id) REFERENCES account
(account_id) ON UPDATE CASCADE,
```

```
CONSTRAINT valid_type FOREIGN KEY (account_type_id) REFERENCES
account_type (account_type_id) ON UPDATE CASCADE )
```

Contributor

```
CREATE TABLE Contributor (
contributor_id INT PRIMARY KEY,
first_name VARCHAR(35),
last_name VARCHAR(35),
middle_initial CHAR(1));
```

Item

```
CREATE TABLE item (
item_id int(11) NOT NULL AUTO_INCREMENT,
item_type_id int(11) NOT NULL DEFAULT 1,
title varchar(50) NOT NULL,
description varchar(255) NOT NULL,
genre_id int(11) NOT NULL,
ISBN varchar(17) DEFAULT NULL,
publication_year int(11) NOT NULL,
publication_date date NOT NULL DEFAULT current_timestamp(),
publisher varchar(50) NOT NULL,
issue_number int(11) DEFAULT NULL,
explicit tinyint(1) DEFAULT NULL,
rating_id int(11) DEFAULT NULL,
total_quantity int(11) NOT NULL,
quantity_available int(11) NOT NULL,
reservation_amount int(11) NOT NULL,
PRIMARY KEY (item_id), KEY genre_id (genre_id),
```

```

KEY rating_id (rating_id),
KEY item_type_id (item_type_id),
CONSTRAINT item_ibfk_1 FOREIGN KEY (genre_id) REFERENCES genre
(genre_id),
CONSTRAINT item_ibfk_2 FOREIGN KEY (item_type_id) REFERENCES item_type
(item_type_id),
CONSTRAINT item_ibfk_3 FOREIGN KEY (rating_id) REFERENCES rating
(rating_id));

```

Relationship Tables

This section contains the DDL statements that define relationships between standalone entities, i.e. accounts, items, and contributors.

Contribution

```

CREATE TABLE Contribution (
contribution_id INT PRIMARY KEY,
item_id INT, contributor_id INT,
role_id INT,
FOREIGN KEY (item_id) REFERENCES Item(item_id),
FOREIGN KEY (contributor_id) REFERENCES Contributor(contributor_id),
FOREIGN KEY (role_id) REFERENCES Role(role_id));

```

Loan

```

CREATE TABLE loan (
loan_id int(11) NOT NULL AUTO_INCREMENT,
item_id int(11) NOT NULL,
account_id int(11) NOT NULL,
loan_out_date date NOT NULL DEFAULT current_timestamp(),
due_date date NOT NULL,

```

```

return_date date DEFAULT NULL,
PRIMARY KEY (loan_id),
KEY account_id (account_id),
KEY item_id (item_id),
CONSTRAINT loan_ibfk_1 FOREIGN KEY (account_id) REFERENCES account
(account_id),
CONSTRAINT loan_ibfk_2 FOREIGN KEY (item_id) REFERENCES item
(item_id) )

```

Reservation

```

CREATE TABLE reservation (
reservation_id int(11) NOT NULL AUTO_INCREMENT,
item_id int(11) NOT NULL,
account_id int(11) NOT NULL,
reservation_date date NOT NULL,
reservation_end_date date NOT NULL,
PRIMARY KEY (reservation_id),
KEY account_id (account_id),
KEY item_id (item_id) USING BTREE,
CONSTRAINT valid_account FOREIGN KEY (account_id) REFERENCES account
(account_id) ON UPDATE CASCADE)

```

Database Sample

The database sample tables are included in the Github, as including it within this document would make it less readable. They will be located within the *sample_tables* folder.