**Shelf Masters**

**Library Management System**

**Version 1.0**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 4/27/2025 | 1.0 | Physical Database Design | Abhiroop Goel |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. **Introduction 4** 
   * 1. Purpose
     2. Scope
     3. Overview

**2.Physical Schema**

**3.Table Contents**

**Physical Database Design**

# Introduction

The Database Requirements Specification serves as a structured blueprint delineating the anticipated functionalities, requirements, and expectations of the database application. This document offers a meticulous overview of the program's scope and intricacies, facilitating a comprehensive understanding prior to the commencement of development activities.

## Purpose

## The Library Management System (LMS) is built to streamline library operations including cataloging media, managing members and staff, processing borrow/return transactions, handling item reservations, and administering overdue fines. The system enhances operational efficiency and user experience by automating key processes like fine calculation and report generation.

## Scope

The system supports cataloging of physical and digital items, user and staff management, borrowing and return tracking, reservation management, and fine enforcement. Excluded features include e-book lending, external system integrations, and recommendation algorithms.

## Definitions, Acronyms, and Abbreviations

* DDL (Data Definition Language): SQL commands used to define database structures.
* DML (Data Manipulation Language): SQL commands used to manage data (insert, update, delete).
* MariaDB: A community-developed relational database management system, forked from MySQL.
* Transaction Integrity: Ensuring that borrowing and returning operations are complete, consistent, isolated, and durable.

## Physical Schema

-- Table: auth\_user

CREATE TABLE auth\_user (

id INT PRIMARY KEY,

password VARCHAR(128),

last\_login TIMESTAMP,

is\_superuser BOOLEAN,

first\_name VARCHAR(150),

last\_name VARCHAR(150),

email VARCHAR(254),

is\_staff BOOLEAN,

is\_active BOOLEAN,

date\_joined TIMESTAMP

);

-- Table: accounts\_user

CREATE TABLE accounts\_user (

id INT PRIMARY KEY,

user\_id INT,

profile\_pic VARCHAR(255),

bio TEXT,

FOREIGN KEY (user\_id) REFERENCES auth\_user(id)

);

-- Table: catalog\_item

CREATE TABLE catalog\_item (

id INT PRIMARY KEY,

item\_type VARCHAR(50),

isbn VARCHAR(20),

issue\_number INT,

publication\_date DATE,

title VARCHAR(200),

publication\_year INT,

genre VARCHAR(50),

creator VARCHAR(100),

item\_format VARCHAR(50),

availability BOOLEAN

);

-- Table: reports\_report

CREATE TABLE reports\_report (

id INT PRIMARY KEY,

created\_by\_id INT,

title VARCHAR(100),

description TEXT,

created\_at TIMESTAMP,

FOREIGN KEY (created\_by\_id) REFERENCES auth\_user(id)

);

-- Table: loans\_reservation

CREATE TABLE loans\_reservation (

id INT PRIMARY KEY,

member\_id INT,

item\_id INT,

reservation\_id INT,

request\_date DATE,

due\_date DATE,

status VARCHAR(20),

FOREIGN KEY (member\_id) REFERENCES auth\_user(id),

FOREIGN KEY (item\_id) REFERENCES catalog\_item(id),

FOREIGN KEY (reservation\_id) REFERENCES reservations\_reservation(id)

);

-- Table: loans\_borrowingtransaction

CREATE TABLE loans\_borrowingtransaction (

id INT PRIMARY KEY,

member\_id INT,

reservation\_id INT,

borrow\_date DATE,

return\_date DATE,

FOREIGN KEY (member\_id) REFERENCES auth\_user(id),

FOREIGN KEY (reservation\_id) REFERENCES loans\_reservation(id)

);

-- Table: reservations\_reservation

CREATE TABLE reservations\_reservation (

id INT PRIMARY KEY,

content\_type\_id INT,

object\_id INT,

timestamp TIMESTAMP,

FOREIGN KEY (content\_type\_id) REFERENCES django\_content\_type(id)

-- Note: object\_id is a polymorphic reference and not a standard foreign key.

);

-- Table: notifications\_notification

CREATE TABLE notifications\_notification (

id INT PRIMARY KEY,

recipient\_id INT,

message TEXT,

created\_at TIMESTAMP,

is\_read BOOLEAN,

FOREIGN KEY (recipient\_id) REFERENCES auth\_user(id)

);

-- Table: core\_auditing

CREATE TABLE core\_auditing (

id INT PRIMARY KEY,

user\_id INT,

action VARCHAR(100),

timestamp TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES auth\_user(id)

);

-- Table: core\_systemconfig

CREATE TABLE core\_systemconfig (

id INT PRIMARY KEY,

config\_key VARCHAR(100),

config\_value TEXT

);