**数据库实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验人：** | **凌国明** | | **学号：** | **21307077** | **日 期：** | **2023年12月8日** |
| **院（系）：** | **计算机学院** | | | **专业（班级）：** | **计算机科学与技术** | |
| **实验题目：** | | **实验5：数据库设计实验** | | | | |

* + - 1. **实验目的**

掌握数据库设计基本方法及数据库设计工具。

* + - 1. **实验内容和要求**

掌握数据库设计基本步骤，包括数据库概念结构设计、逻辑结构设计，物理结构设计，数据库模式SQL语句生成。能够使用数据库设计工具进行数据库设计。

* + - 1. **实验重点和难点**

实验重点：概念结构设计、逻辑结构设计。

实验难点：逻辑结构设计。逻辑结构设计虽然可以按照一定的规则从概念结构转换而来，但是由于概念结构通常比较抽象，较少考虑更多细节，因此转换而成的逻辑结构还需要进一步调整和优化。逻辑结构承接概念结构和物理结构，处于核心地位，因而是数据库设计的重点，也是难点。

* + - 1. **实验工具**
* Navicat （参阅：<http://www.yaotu.net/biancheng/774.html> ）
* PowerDesigner、ERwin、Office Visio、亿图图示、starUML、Visual Studio Code等数据库设计工具
* MySQL、SQL Server
  + - 1. **实验过程**

设计一个应用数据库。请使用Navicat、PowerDesigner或者ERwin等数据库设计工具设计该数据库。

(1)数据库概念结构设计

先识别出系统中的实体。然后根据实际语义，分析实体之间的联系，确定实体之间一对一，一对多和多对多联系。据此，绘制实体-联系图(E-R图)。

设计如下（参阅实验教材）：

设计一个学校图书借阅管理系统，根据图书借阅相关信息，设计实体如下：



图书(Book):属性包括书号、书名、作者、出版社等。

类别(Category):独立实体，属性包括类别ID、类别名称等。

读者(Reader): 包含读者ID、姓名、联系方式等属性。

借阅证(Borrowing Card): 包含借阅证ID、有效期等属性。

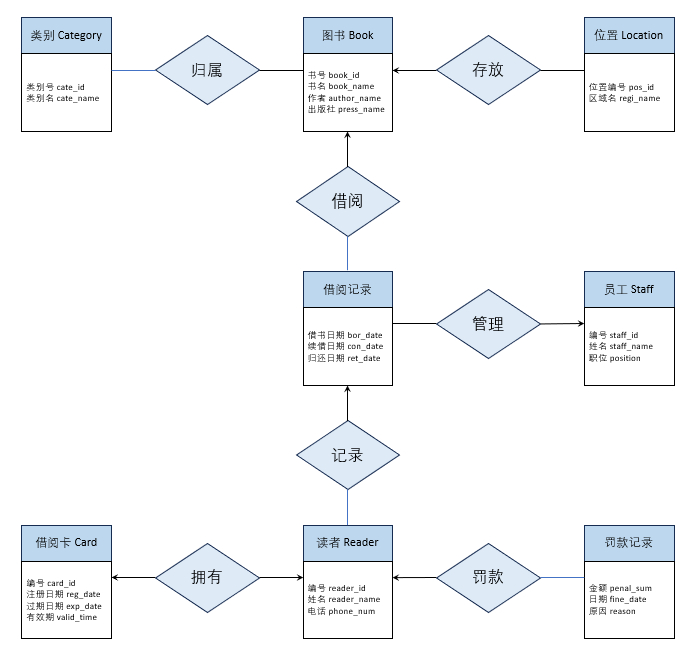
借阅记录(Borrowing Record): 包含借书日期、续借日期、归还日期等属性。

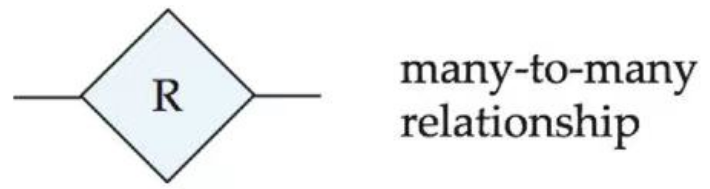
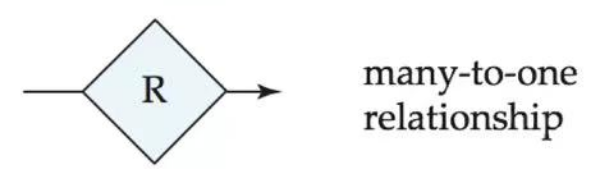
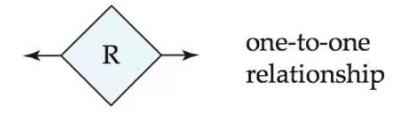
罚款记录(Fine Record): 记录罚款的详细信息。

员工(Staff): 包含员工ID、姓名、职位等属性。

位置(Location): 图书的具体存放位置，包含位置ID、区域等属性。

根据现实生活中各实体之间的联系，我设计了一组联系集来关联实体集。



归属(Book - Category): 一个图书可以对应多个类别，一个类别可以包含多个图书。关系类型：M:N。

借阅(Borrowing Record - Book): 一个借阅记录对应一个图书，一个图书可以有多个借阅记录。关系类型：1:N。

拥有(Reader - Borrowing Card): 一个读者拥有一个借阅证，一个借阅证对应一个读者。关系类型：1:1。

罚款(Reader - Fine Record): 一个读者可以有多个罚款记录，一个罚款记录对应一个读者。关系类型：1:N。

记录(Borrowing Card – B.. Record): 一个借阅证对应多个借阅记录，一个借阅记录对应一个借阅证。关系类型：1:N。

管理(Staff - Borrowing Record): 一个员工可以管理多个借阅记录，一个借阅记录由一个员工管理。关系类型：1:N。

存放(Book - Location): 一个图书存放在一个位置，一个位置可以有多个图书。关系类型：1:N。

这里为了方便向逻辑结构转换，统一采用二元关系去表达实体之间的联系。

(2)数据库逻辑结构设计

按照数据库设计原理中概念结构转化成逻辑结构的规则，每个实体转换成一个关系，多对多的联系也转换成一个关系。从而，根据上述E-R图设计数据库逻辑结构。

设计如下（参阅实验教材）：

|  |  |  |
| --- | --- | --- |
| 图书(Book) 表：  书号 (主键)  书名  作者  出版社  是否在馆 | 类别(Category) 表：  类别ID (主键)  类别名称 | 读者(Reader) 表：  读者ID (主键)  姓名  联系方式 |
| 借阅证(Borrowing Card) ：  借阅证ID (主键)  有效期  读者ID (外键，引用读者表的读者ID) | 位置(Location) 表：  位置ID (主键)  区域  书号 (外键，引用图书表的书号) | 罚款记录(Fine Record) 表：  罚款记录ID (主键)  详细信息  读者ID (外键，引用读者表的读者ID) |
| 员工(Staff) 表：  员工ID (主键)  姓名  职位 | 借阅记录(Borrowing Record) 表：  借阅记录ID (主键)  借书日期  续借日期  归还日期  书号 (外键，引用图书表的书号)  借阅证ID (外键，引用借阅证表的借阅证ID)  员工ID (外键，引用员工表的员工ID) | 图书与类别关系(处理M:N关系)：  书号 (外键，引用图书表的书号)  类别ID (外键，引用类别表的类别ID) |

(3)数据库物理结构设计

数据库物理结构首先根据逻辑结构自动转换生成，然后根据应用需求设计数据库的索引结构、存储结构。

设计如下（参阅实验教材）：

以下是索引结构的设计：

|  |  |  |
| --- | --- | --- |
| 图书(Book)表：  对书号建立主键索引，因为它是唯一标识。  对书名、作者和出版社建立非聚集索引，以加快搜索和查询速度。 | 类别(Category)表：  对类别ID建立主键索引。  对类别名称建立非聚集索引。 | 读者(Reader)表：  对读者ID建立主键索引。  对姓名建立非聚集索引，以便快速检索读者信息。 |
| 借阅证(Borrowing Card)表：  对借阅证ID建立主键索引。  对读者ID建立外键索引。 | 借阅记录(Borrowing Record)表：  对借阅记录ID建立主键索引。  对书号、借阅证ID和员工ID建立外键索引，以加快关联查询。 | 罚款记录(Fine Record)表：  对罚款记录ID建立主键索引。  对读者ID建立外键索引。 |
| 员工(Staff)表：  对员工ID建立主键索引。 | 位置(Location)表：  对位置ID建立主键索引。  对书号建立外键索引。 | 图书与类别关系(Book-Category)表：  对书号和类别ID建立复合索引。 |

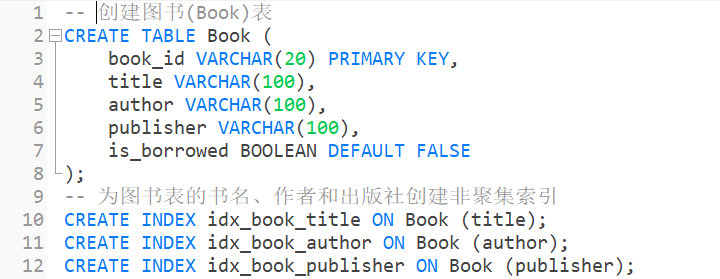
以下是存储结构的设计

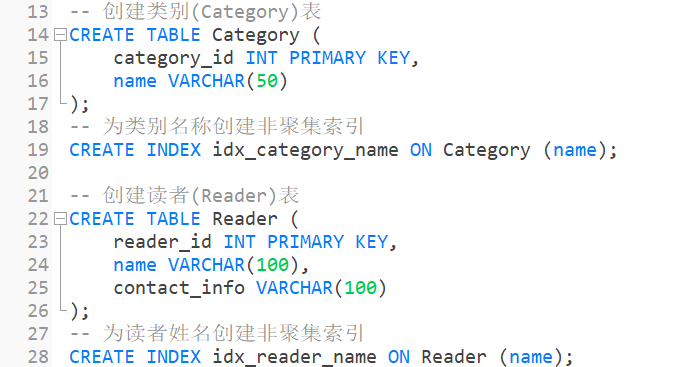
|  |  |
| --- | --- |
| 图书(Book)表：  使用B-tree结构存储书号索引，以优化查询和范围搜索。  书名、作者和出版社字段可使用全文索引以优化文本搜索。 | 类别(Category)和读者(Reader)表：  类别ID和读者ID使用B-tree索引。  类别名称和读者姓名可使用散列索引以加快等值查询。 |
| 借阅记录(Borrowing Record)表：  使用B-tree索引来存储借阅记录ID，书号和借阅证ID。  考虑到借书和还书操作频繁，可以使用聚集索引来优化插入和更新性能。 | 位置(Location)表：  使用B-tree索引来存储位置ID。  书号索引应考虑空间效率，因为一个位置可能存放多本书。 |

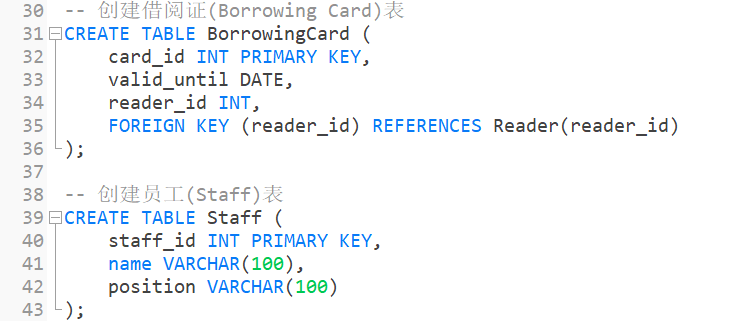
(4) 数据库模式SQL语句生成

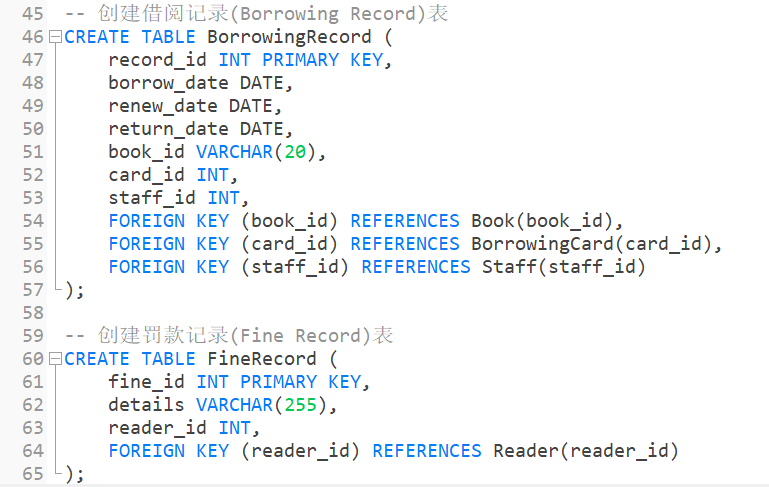
生成MySQL或SQL Server数据库管理系统的SQL语句。

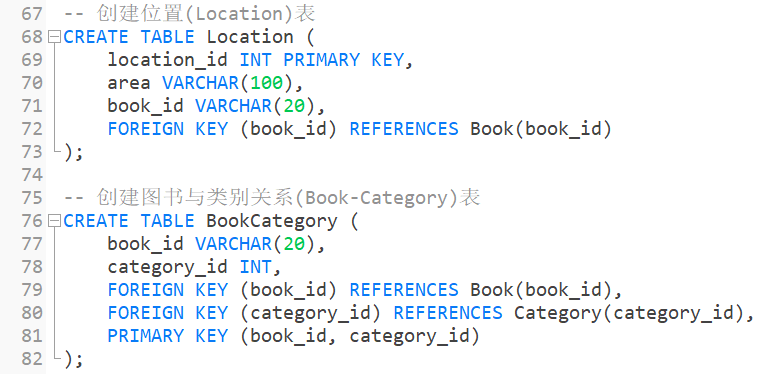
设计如下（参阅实验教材）：



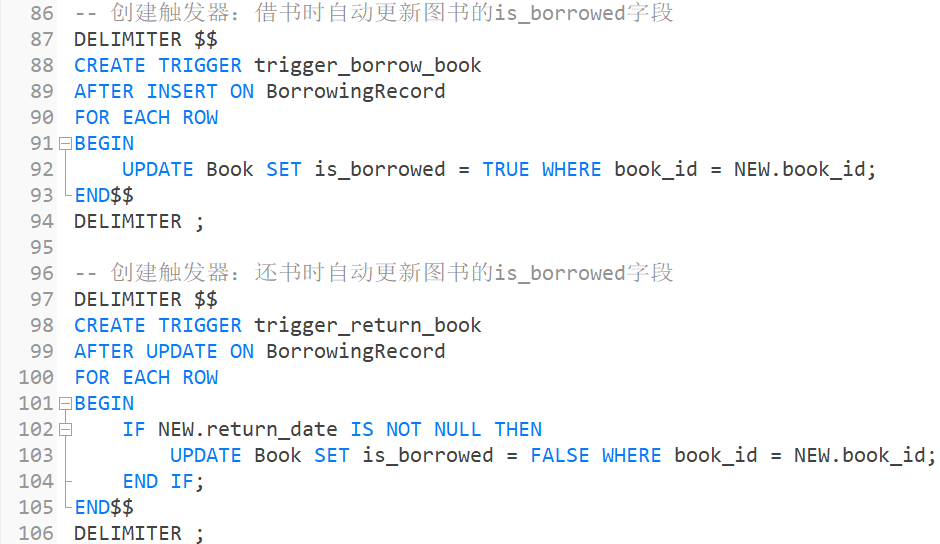




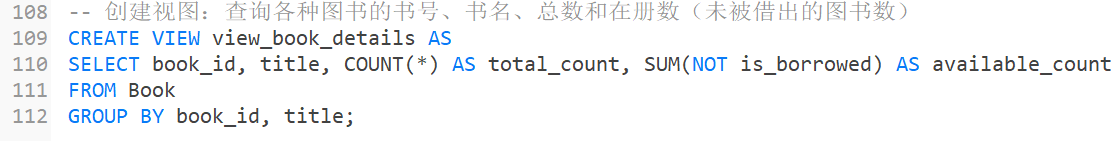




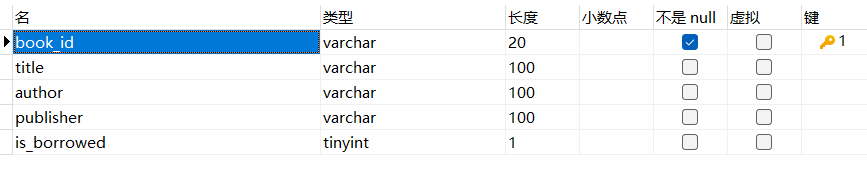
然后创建触发器，在修改借阅记录时，更新图书的状态



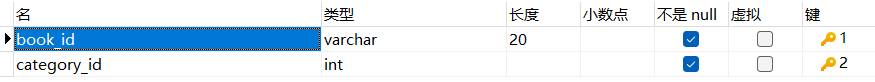
最后创建视图，查询各种图书的书号、书名、总数和在册数（未被借出的图书数）



Book表：



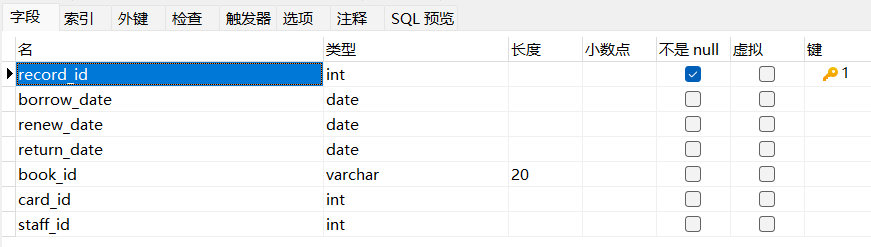
BookCategory表：



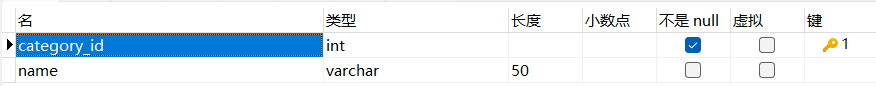
BorrowingCard表：



BorrowingRecord表



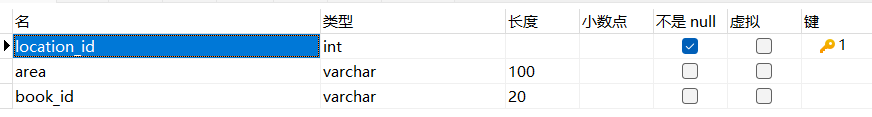
Category表：



FineRecord表：



Location表：



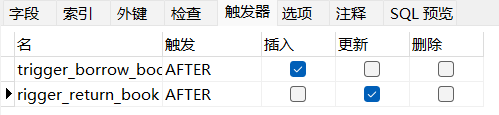
Reader表：



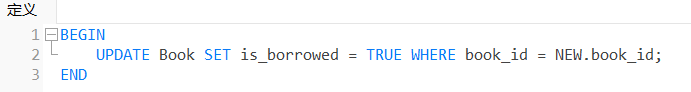
Staff表：



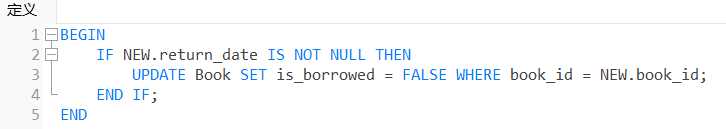
BorrowingRecord表中的触发器：



trigger\_borrow\_book：



trigger\_return\_book：



* + - 1. **与实验结果相关的文件**

init.sql：SQL语句初始化数据库（建表，创建触发器，创建视图）

library.sql：数据库的转储（library数据库的结构和数据）

* + - 1. **实验总结**

在这次的数据库设计实验中，我从ER图的分析开始，逐步深入到逻辑结构的设计，再到物理结构（包括索引和存储结构）的实现，最后通过编写SQL语句完成了整个数据库的构建。这个过程不仅加深了我对数据库设计原理的理解，也提升了我的实际操作能力。

首先，ER图为我提供了一个清晰的视角来观察和理解实体间的关系。通过将每个实体转化为一个关系表，并为多对多关系建立单独的关联表，我得以在逻辑层面上确立了数据的结构。这个步骤对于后续的物理设计至关重要，因为它直接影响了数据的存储和检索效率。

进入物理设计阶段，我特别关注于索引结构和存储结构的优化。通过为主键和频繁查询的字段创建索引，我能显著提升查询速度。特别是在面对大量数据和复杂查询时，合理的索引策略显得尤为重要。此外，考虑到数据的访问模式和存储效率，我选择了适合各种场景的索引类型，如B-tree和全文索引。

编写SQL语句是实践中最具挑战性的部分。我不仅需要确保语句的正确性和高效性，还要考虑数据完整性和安全性。创建触发器和视图使我能够自动化一些常规任务，如更新图书的借阅状态，同时也使得复杂的查询变得简单。通过这个过程，我深刻体会到了SQL语言的强大和灵活性。

总的来说，这次实验是一次宝贵的学习经历。这次实验我不仅更深入地理解了数据库设计的理论知识，而且我还通过实践学会了如何将理论应用于实际的数据库构建中。