哈爾濱工業大學

数 据 库 实 验 一 报 告

实验一：数据库设计与应用开发

姓　　名 c y c l e k e

学　　号 x x x x x x x x x x x

老　　师 _____

专　　业 计　　算　　机

日　　期 2020 年 04 月 22 日

目录

第1章 实验目的

在本次实验中，我需要构思一个规模合理且功能较完善的数据密集型应用场景，之后根据应用需求设计一个关系数据库并开发数据库系统应用，从而能够

1. 学会正确运用概念数据库设计方法，正确使用实体——联系图（ER 图）表示概念数据模型；

2. 学会正确运用逻辑数据库设计方法，在概念数据模型的基础上，设计合理的关系数据库模式；

3. 学会正确运用物理数据库设计方法，根据工作负载，合理设计数据库的存取方法与存储结构；

4. 掌握一种关系数据库管理系统（RDBMS）的使用方法，使用 SQL 创建、更新和查询关系数据库；

5. 掌握数据库系统应用开发方法。

第 2 章　数据库设计

2.1　需求分析

在本次实验中，我计划设计一个书评管理系统。在本书评系统中，我们需要管理若干书籍以及书籍对应的评论（包括书籍和评论的相关信息）。在该系统中，具体的数据包括各种书籍信息（包括书名、ISBN 码、书籍领域、作者和出版社），出版社信息（包括出版社名称），作者信息（包括姓名、性别），评论信息（包括书籍、评论的读者和评论的内容），读者信息（包括姓名）。

我们需要约束书籍的 ISBN 码各不相同，书籍的领域在给定的若干个领域中，其作者和出版社也必须是数据库中存在的实体。作者的性别也必然是给定给定性别中的一种。自然，评论的读者和数据也要在数据库中出现。

在使用书评系统时，我们需要支持随时添加合法的书籍、出版社、作者、读者和评论，同时支持对于上述内容的搜索。在搜索时，不同的内容支持不同的搜索方式。如对于书籍，我们需要支持按照唯一的 ISBN 码来进行搜索，对于评论、出版社等内容，我们需要支持按照评论内容是否包含给定的关键字来进行搜索。为了便于管理，我们可以给每个实体设置一个编号，自然书评系统需要支持按照编号进行搜索。

2.2　概念数据库设计

在书评系统中，概念数据库的实体型包括出版社、书籍、作者和读者，联系型包括出版关系、写作关系和评论关系。其包含的属性值和关联实体如下：

- 出版社：编号，名称；
- 书籍：编号，ISBN 码，书名，领域；
- 作者：编号，姓名，性别；
- 读者：编号，姓名；
- 出版关系：出版社，书籍；
- 写作关系：作者，书籍；
- 评论关系：编号，评论内容，读者，书籍。

根据上述的关系，我们可以画出对应的 ER 图，之后分析实体之间的关系，是否有必要建立多元关系，再确定基数。最终构造的 ER 图如图2.1。

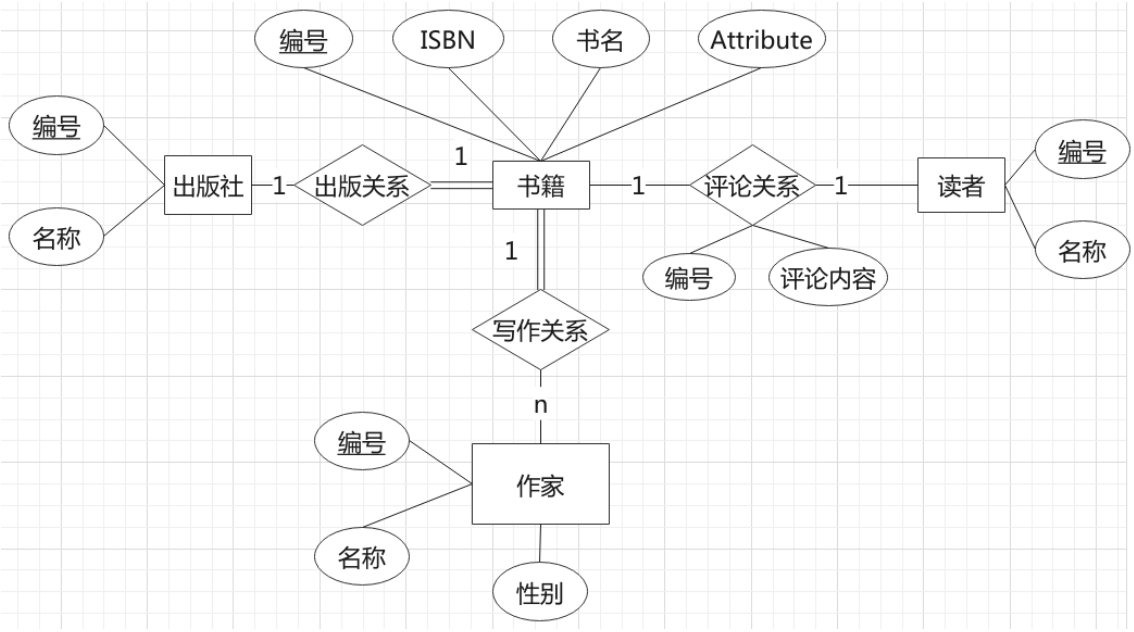

图 2.1: 概念数据库对应的 ER 图

2.3 逻辑数据库设计

由于出版关系仅与出版社和书籍相关，且一个书籍仅与一个出版社存在该关系，所以我将出版关系存储在书籍类中。实体型转换的结果如下：

- 出版社：Publisher (Id, Name)
- 作者：Author (Id, Name, Sex)
- 书籍：Book (Id, ISBN, Name, Category, PublisherId)
- 读者：Reader (Id, Name)

关系型转换的结果如下：

- 写作关系：WriteBook (AuthorId, BookId)
- 评论关系：Comment (Id, ReaderId, BookId, Content)

2.4 物理数据库设计

对于绝大部分关系，我使用其主键建立主索引（写作关系的主键为 AuthorId 和 BookId 形成的二元组）。在 MySQL 声明主键时，系统会自动地建立主索引。特别地，我对于书籍上的 ISBN 还建立了一个唯一索引。

2.5 数据库建立

在本次实验中，我使用 MySQL 来建立数据库。其对应的 MySQL 代码为：

```
CREATE TABLE Publisher (
  Id INT AUTO_INCREMENT PRIMARY KEY,
  Name VARCHAR(256) NOT NULL
);

CREATE TABLE Author (
  Id INT AUTO_INCREMENT PRIMARY KEY,
  Name VARCHAR(256) NOT NULL,
  Sex ENUM('男','女') NOT NULL
);

CREATE TABLE Book (
  Id INT AUTO_INCREMENT PRIMARY KEY,
  ISBN VARCHAR(17) NOT NULL UNIQUE,
  Name VARCHAR(20) NOT NULL,
  Category ENUM('社会科学', '自然科学', '文学', '技术', '军事', '宗教', '综合'),
  PublisherId INT NOT NULL,
  FOREIGN KEY (PublisherId) REFERENCES Publisher(Id),
  UNIQUE INDEX(ISBN)
);

CREATE TABLE Reader (
  Id INT AUTO_INCREMENT PRIMARY KEY,
  Name VARCHAR(256) NOT NULL
);

CREATE TABLE Comment (
  Id INT AUTO_INCREMENT PRIMARY KEY,
  ReaderId INT NOT NULL,
  BookId INT NOT NULL,
  Content VARCHAR(256) NOT NULL,
  FOREIGN KEY (ReaderId) REFERENCES Reader(Id),
  FOREIGN KEY (BookId) REFERENCES Book(Id)
);

CREATE TABLE WriteBook (
  AuthorId INT NOT NULL,
  BookId INT NOT NULL,
  PRIMARY KEY (AuthorId, BookId),
  FOREIGN KEY (AuthorId) REFERENCES Author(Id),
  FOREIGN KEY (BookId) REFERENCES Book(Id)
);
```

注意其中“关系型”对应的 MySQL 表中声明了外键，这样能够保证数据的一致性。

第 3 章 数据库应用开发

3.1 数据库应用设计

在本次实验中，我将数据库管理同用户界面隔离，这样便于后续数据库逻辑更改后应用的更新。源代码的目录结构如下：

```
src
|-- app.py
|-- data_manager
|   |-- author_manager.py
|   |-- book_manager.py
|   |-- comment_manager.py
|   |-- mysql_connect.py
|   |-- publisher_manager.py
|   |-- reader_manager.py
|   `-- write_book_manager.py
`-- ui
    |-- mainwindow.ui
    `-- ui_mainwindow.py

2 directories, 10 files
```

在开发应用的过程中，我利用 PyMySQL 库来建立应用和 MySQL 数据库的连接，使用 PyQt5 来实现图形化界面。

我将所有负责数据管理的类放置在 data_manager 目录下，其中 MySQLConnect 类负责管理同 MySQL 数据库的连接，对一些重复率高的 MySQL 语句设置模版（如创建表、插入和查询语句）。如对于插入数据，我使用格式化字符串来实现。

```
cursor.execute("INSERT INTO {} ({}) VALUES ({});".format(
    table, keys, values))
```

而 AuthorManager、BookManager、CommentManager、PublisherManager、ReaderManager 和 WriteBookManager 分别管理作者表、书籍表、评论表、出版社表、读者表和写作关系表。他们通过 MySQLConnect 类来连接连接数据库，之后通过 MySQLConnect 类提供的 API 来更新和查询自己负责的数据表，为前端程序提供数据的查询和更新。

同 GUI 界面相关的配置文件和界面配置则存放在 ui 目录下。它们负责将数据直观地展示在用户面前，同时为用户提供相应的输入和输出方式，让用户能够通

过一个简洁、友好的界面来对数据进行管理。

最后由 App 类负责将图形化界面同数据管理联系起来，将相关的按钮和函数关联起来，实现让用户进行数据管理。

3.2 实现结果

在命令行中输入下面的指令即可运行该书评系统，其初始界面如图3.1所示。

```
1 python src/app.py
```

点击在出版社标签页中点击 ‘‘列出所有’’ 按钮，应用会从数据库中获取所有的出版社信息并在表格中列出。在其他的标签页中，应用也提供相似的按钮来查看所有的相关信息。

图 3.1: 应用初始界面

图 3.2: 列出所有出版社

切换到评论页中，我们可以使用关键字搜索相关评论。点击 ‘‘按关键字搜索评论’’，应用会弹出输入关键字的搜索框（图3.3），我们在其中输入 ‘‘的’’ 字，表格中就会列出包含 ‘‘的’’ 字的评论（图3.4）。对于其他数据，应用还支持通过 ID 编号，ISBN，作者和读者等信息来进行搜索。

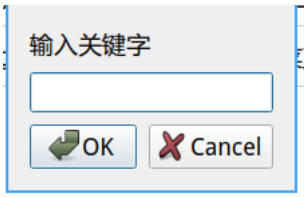

图 3.3: 输入关键字

图 3.4: 列出包含 ‘‘的’’ 字的评论

第4章 实验总结

在本次实验中，我学会了正确运用概念数据库设计方法，将概念数据库通过ER图形式化地表示出来，并逐步转换为逻辑书籍库和物理数据库，从而能够设计一个高效的关系数据库。之后我利用设计初的数据库实现一个用户友好的数据库系统应用，掌握数据库系统应用开发方法，同时还提高了自己对于MySQL语言的熟悉程度。