

**数据库系统原理实践报告**

|  |  |
| --- | --- |
| 专 业： | 计算机科学与技术 |
| 班 级： | 计卓2001班 |
| 学 号： | U201914858 |
| 姓 名： | 王怡彬 |
| 指导教师： | 谢美意 |

|  |  |
| --- | --- |
| 分数 |  |
| 教师签名 |  |

2023 年 1 月 2 日

**教师评分页**

|  |  |
| --- | --- |
| 子目标 | 子目标评分 |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
|  |  |

|  |  |
| --- | --- |
| 总分 |  |

目 录

[1 课程任务概述 1](#_Toc123658845)

[2 任务实施过程与分析 2](#_Toc123658846)

[2.1数据库、表与完整性约束的定义(Create) 2](#_Toc123658847)

[2.2表结构与完整性约束的修改(ALTER) 2](#_Toc123658848)

[2.4 数据查询(Select)之二 7](#_Toc123658849)

[2.5 数据的插入、修改与删除(Insert,Update,Delete) 8](#_Toc123658850)

[2.6 视图 9](#_Toc123658851)

[2.7 存储过程与事务 9](#_Toc123658852)

[2.8 触发器 11](#_Toc123658853)

[2.9 用户自定义函数 11](#_Toc123658854)

[2.10 安全性控制 11](#_Toc123658855)

[2.11 并发控制与事务的隔离级别 11](#_Toc123658856)

[2.12 备份+日志：介质故障与数据库恢复 12](#_Toc123658857)

[2.13数据库设计与实现 12](#_Toc123658858)

[2.14 数据库应用开发(JAVA篇) 13](#_Toc123658859)

[2.15数据库的索引B+树实现 14](#_Toc123658860)

[3 课程总结 15](#_Toc123658861)

[附录 16](#_Toc123658862)

# 1 课程任务概述

该课程是为了配合"数据库系统原理"课程而开设的实践课，旨在帮助学生在实践中加深对数据库系统原理的理解。该课程以MySQL语言为例，并重视理论与实践的结合。课程内容涵盖了MySQL的数据对象管理与编程、数据处理任务、数据库的系统内核（如安全性控制、完整性控制、恢复机制、并发控制机制）、数据库的设计与实现、以及数据库应用系统的开发。

# 2 任务实施过程与分析

## 数据库、表与完整性约束的定义(Create)

使用MySQL语言实现数据库、表与完整性约束的定义。掌握create及其相关语句的使用。

* + 1. **创建数据库**

该关卡任务已完成，实施情况本报告略过

* + 1. **创建表及表的主码约束**

该关卡任务已完成，实施情况本报告略过

* + 1. **创建外码约束(foreign key)**

该关卡任务已完成，实施情况本报告略过

* + 1. **CHECK约束**

该关卡任务已完成，实施情况本报告略过

* + 1. **DEFAULT约束**

该关卡任务已完成，实施情况本报告略过

* + 1. **UNIQUE约束**

该关卡任务已完成，实施情况本报告略过

## 表结构与完整性约束的修改(ALTER)

使用MySQL语言实现表结构与完整性约束的修改。掌握alter及其相关语句的使用。

* + 1. **修改表名**

该关卡任务已完成，实施情况本报告略过

* + 1. **添加与删除字段**

该关卡任务已完成，实施情况本报告略过

* + 1. **修改字段**

该关卡任务已完成，实施情况本报告略过

* + 1. **添加、删除与修改约束**

该关卡任务已完成，实施情况本报告略过

1. **数据查询(Select)之一**

使用MySQL语言实现数据查询。掌握select及其相关语句的使用。

* + 1. **金融应用场景介绍,查询客户主要信息**

该关卡任务已完成，实施情况本报告略过

* + 1. **邮箱为null的客户**

该关卡任务已完成，实施情况本报告略过

* + 1. **既买了保险又买了基金的客户**

使用嵌套的子查询完成多表多条件的条目筛选。第二个子查询查询出所有具有pro\_type=2和在上一个子查询查询出的pro\_c\_id列表中的客户的pro\_c\_id。最后，使用最外层的SELECT语句查询出所有满足条件的客户的对应信息，并按照c\_id排序，代码如图2.1所示。

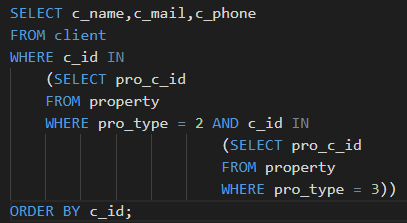


图2.1 任务2.3.3代码

* + 1. **办理了储蓄卡的客户信息**

在select语句的from子句中查询两个表client和bank\_card以实现多表连接，并在where子句中使用and操作对多条件进行查询，并按照c\_id排序，代码如图2.2所示。

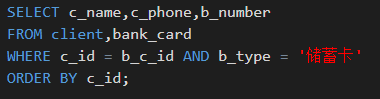


图2.2 任务2.3.4代码

* + 1. **每份金额在30000～50000之间的理财产品**

该任务较为简单，此处不再赘述，代码如图2.3所示。

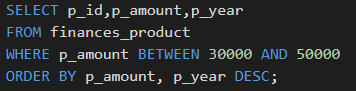


图2.3 任务2.3.5代码

* + 1. **商品收益的众数**

具体来说，首先使用GROUP BY子句按照pro\_income列分组，然后使用COUNT函数统计每组的行数。接下来使用HAVING子句过滤出出现次数最多的组，这里使用了ALL关键字和一个子查询，以实现只保留每组的行数都大于等于子查询中每组的行数的各组，即众数的筛选。最后，使用SELECT语句显示pro\_income列和每组的行数，代码如图2.4所示。

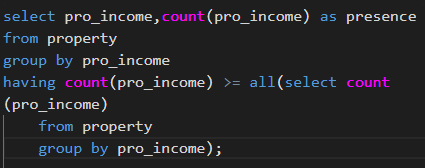


图2.4 任务2.3.6代码

* + 1. **未购买任何理财产品的武汉居民**

使用like子句和not exists谓词排除掉存在pro\_type=1的记录的客户，代码如图2.5所示。

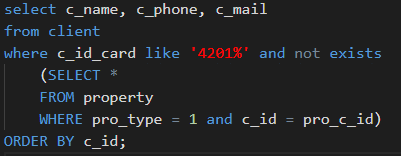


图2.5 任务2.3.7代码

* + 1. **持有两张信用卡的用户**

首先使用一个子查询查询出拥有至少两张信用卡的客户，然后使用WHERE子句和IN运算符将查询结果用作条件，查询出满足条件的客户的对应信息，代码如图2.6所示。

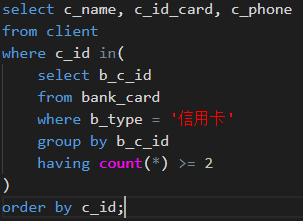


图2.6 任务2.3.8代码

* + 1. **购买了货币型基金的客户信息**

与上一关类似，使用子查询查询出拥有货币型基金且pro\_type=3的客户，然后使用WHERE子句和IN运算符将查询结果用作条件，查询出满足条件的客户的对应信息。代码如图2.7所示。

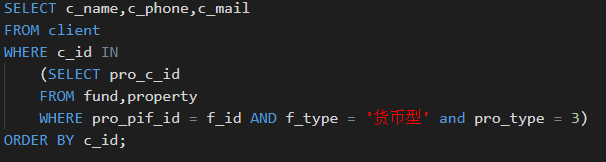


图2.7 任务2.3.9代码

* + 1. **投资总收益前三名的客户**

首先使用子查询和派生表查询出查询出每个客户的id和投资总收益，并将结果进行排序，最后使用LIMIT子句限制只显示前三条记录。代码如图2.8所示。

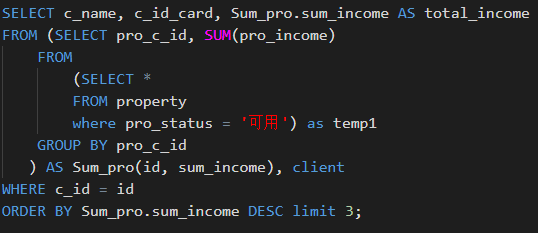


图2.8 任务2.3.10代码

* + 1. **黄姓客户持卡数量**

使用外连接left outer join实现在两表连接时将无记录项置为0的操作(没有办卡的卡数量计为0)，使用like语句对客户名进行筛选以找出黄姓客户，并利用group by子句对其进行分组统计。代码如图2.9所示。

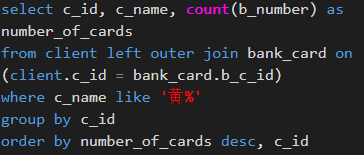


图2.9 任务2.3.11代码

* + 1. **客户理财、保险与基金投资总额**

使用外连接left outer join和ifnull函数实现在两表连接时将无记录项置为0的操作，并多次使用外连接和子查询统计各类(理财,保险,基金)资产投资金额，最后在select语句中对各项投资金额进行加和以计算投资总额。代码如图2.10所示。

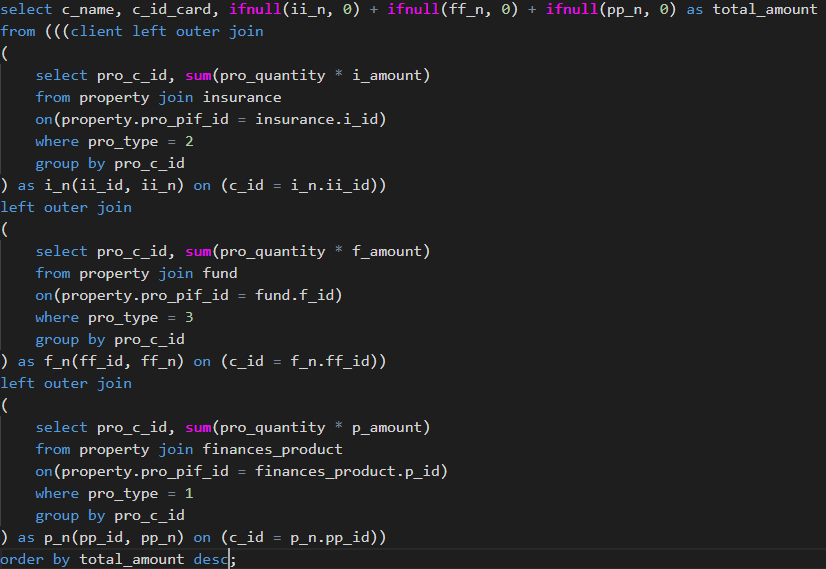


图2.10 任务2.3.12代码

* + 1. **客户总资产**

与上一关类似，使用外连接left outer join和ifnull函数实现在两表连接时将无记录项置为0的操作，并多次使用外连接和子查询统计多表项的数据和，最后在select语句中对多项数据进行加减操作运算以计算得出客户总资产。由于代码过长且内容与上一关大体重复，此处不再展示代码。

* + 1. **第N高问题**

首先使用子查询查询出保险账户金额排名第四的保险的i\_amount。然后使用WHERE子句将查询结果用作条件，查询出满足条件的数据项。

在子查询中使用GROUP BY子句以实现消除重复数据项；使用ORDER BY子句进行排序；使用LIMIT子句，并将其参数设为3，1。以跳过前三条记录，取第四条记录。代码如图2.11所示。

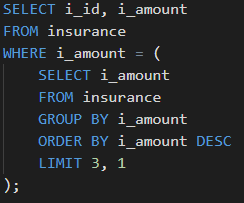


图2.11 任务2.3.14代码

* + 1. **基金收益两种方式排名**

第一部分根据总收益进行倒序排序并使用rank函数为每个id分配一个名次（rank），利用rank函数特性实现总收益相同时名次也相同但不连续。第二部分与第一部分主体相同，但利用dense\_rank函数特性实现总收益相同时名次也相同且连续。代码如图2.12所示。

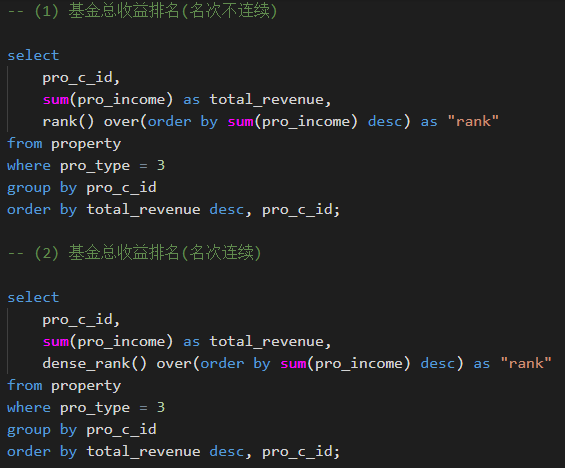


图2.12 任务2.3.15代码

* + 1. **持有完全相同基金组合的客户**

该任务关卡跳过

* + 1. **购买基金的高峰期**

该任务关卡跳过

* + 1. **至少有一张信用卡余额超过5000元的客户信用卡总余额**

该任务关卡跳过

* + 1. **以日历表格式显示每日基金购买总金额**

该任务关卡跳过

## 2.4 数据查询(Select)之二

使用MySQL语言实现数据查询。掌握select及其相关语句的使用。

* + 1. **查询销售总额前三的理财产品**

该任务关卡跳过

* + 1. **投资积极且偏好理财类产品的客户**

该关卡任务已完成，实施情况本报告略过

* + 1. **查询购买了所有畅销理财产品的客户**

该任务关卡跳过

* + 1. **查找相似的理财产品**

该任务关卡跳过

* + 1. **查询任意两个客户的相同理财产品数**

该任务关卡跳过

* + 1. **查找相似的理财客户**

该任务关卡跳过

## 2.5 数据的插入、修改与删除(Insert,Update,Delete)

使用MySQL语言实现表数据的插入、修改与删除。掌握Insert,Update,Delete及其相关语句的使用。

* + 1. **插入多条完整的客户信息**

该关卡任务已完成，实施情况本报告略过

* + 1. **插入不完整的客户信息**

该任务较为简单，此处不再赘述，代码如图2.13所示。

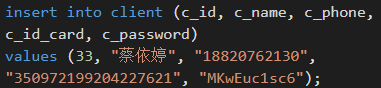


图2.13 任务2.5.2代码

* + 1. **批量插入数据**

该关卡任务已完成，实施情况本报告略过

* + 1. **删除没有银行卡的客户信息**

该任务较为简单，此处不再赘述，代码如图2.14所示。

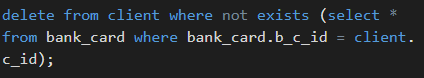


图2.14 任务2.5.4代码

* + 1. **冻结客户资产**

该关卡任务已完成，实施情况本报告略过

* + 1. **连接更新**

该任务较为简单，此处不再赘述，代码如图2.15所示。

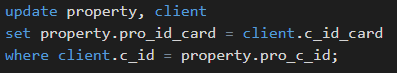


图2.15 任务2.5.6代码

## 2.6 视图

使用MySQL语言实现视图view的创建和基于视图的查询。掌握create view及其相关语句的使用。

* + 1. **创建所有保险资产的详细记录视图**

该关卡任务已完成，实施情况本报告略过

* + 1. **基于视图的查询**

该任务较为简单，此处不再赘述，代码如图2.16所示。

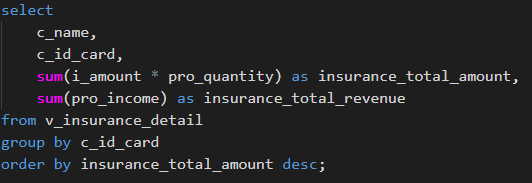


图2.16 任务2.6.2代码

## 2.7 存储过程与事务

使用MySQL语言实现存储过程的创建和使用。掌握使用流程控制语句、游标和事务三种不同的存储过程。

* + 1. **使用流程控制语句的存储过程**

在存储过程中，使用了一个WITH RECURSIVE子句和CTE（递归查询）来生成斐波拉契数列。首先定义了CTE的初始值为(0, 0, 0)，三个数值分别表示数列索引、当前值和前一个值，然后使用UNION ALL语句将当前的CTE和其他行结合起来。每一行的值都通过使用IF函数来计算：如果id小于2，则当前值为1；否则，当前值等于前一个值（pre）加上当前值（cur）。最后进行更新，将当前值赋值给前一个值。代码如图2.17所示。

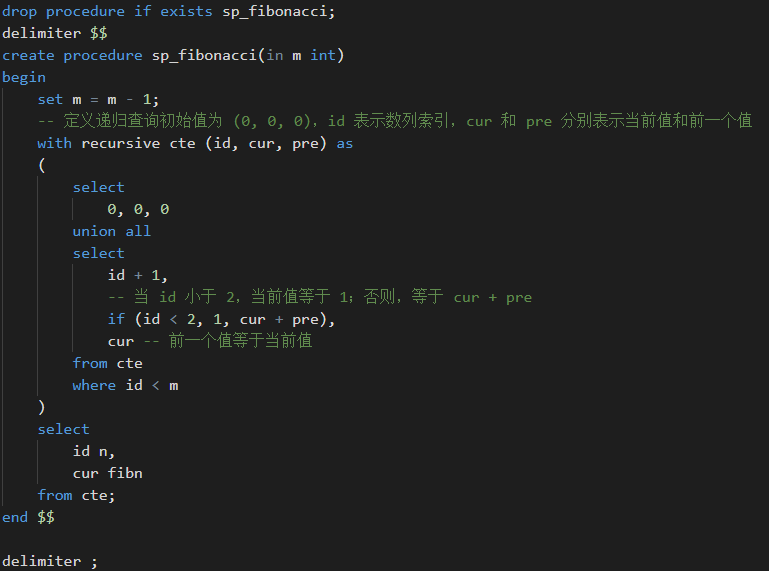


图2.17 任务2.7.1代码

* + 1. **使用游标的存储过程**

在循环中，它将每天的值班医生、护士分别从员工表中按照类型分类查询，并将每天的值班信息插入到 night\_shift\_schedule 表中。循环会继续直到开始日期超过结束日期为止。

首先定义变量：光标 cur1 和 cur2、医生 doctor、护士 nurse1 和 nurse2、标志变量 done 和 typ、以及一个字符变量 h。

然后分别打开光标 cur1 和 cur2。接下来，使用 while 循环，循环条件是 start\_date 小于等于 end\_date，每次循环都将 start\_date 增加 1 天。在循环中，首先从 cur1 中获取两个护士的信息，并将这些信息赋值给 nurse1 和 nurse2。然后计算当前日期的星期数，如果是周日且 h 不为空，则将 h 的值赋给 doctor，并将 h 置空；否则，从 cur2 中获取医生的信息，并将这些信息赋值给 doctor 和 typ。如果当前日期是周五、周六、周日，并且 typ 等于 1，则将 doctor 的值赋给 h，并再次从 cur2 中获取医生的信息，然后将这些信息赋值给 doctor 和 typ。最后，将某天的日期、doctor、nurse1、nurse2 的信息插入 night\_shift\_schedule 表中。在循环结束后，关闭光标 cur1 和 cur2。

由于代码较长，此处不贴出代码。

* + 1. **使用事务的存储过程**

该关卡任务已完成，实施情况本报告略过

## 2.8 触发器

**2.8.1 为投资表 property 实现业务约束规则-根据投资类别分别引用不同表的主码**

为了避免插入数据不合法导致的错误，我们需要在插入事件执行之前触发触发器，并对每一条要插入的数据进行检查。为了更好地反馈出错信息，可以先定义用于输出错误信息的字符串，然后依次检查数据。

## 2.9 用户自定义函数

掌握MySQL语言中函数的定义并调用函数。

**2.9.1 创建函数并在语句中使用它**

要计算某个客户所有储蓄卡的存款总数，需要先把客户表和银行卡表连接起来。连接的条件是客户的 id 和银行卡的 id 相同，并且银行卡类型是储蓄卡。然后用 sum 聚集函数统计存款总数并返回。调用较为简单，此处不再赘述，函数体代码如图2.18所示。

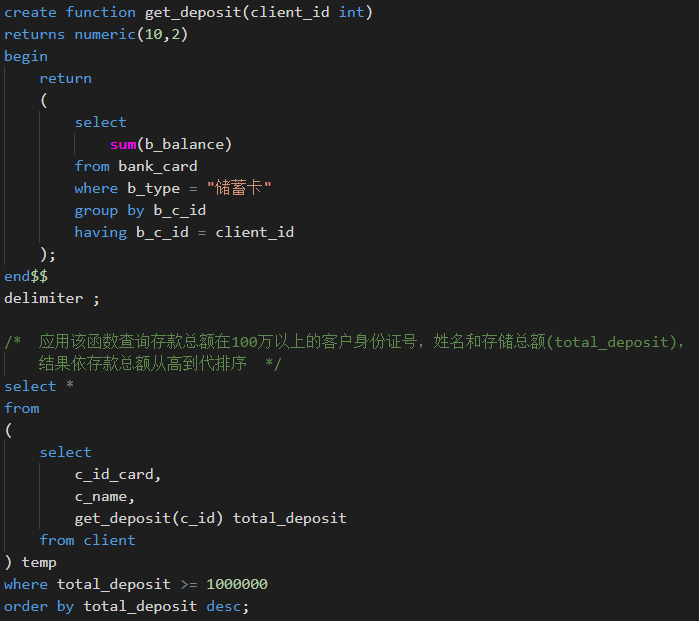


图2.18 任务2.9.1代码

## 2.10 安全性控制

**2.10.1 用户和权限**

该关卡任务已完成，实施情况本报告略过

**2.10.2 用户、角色与权限**

该关卡任务已完成，实施情况本报告略过

## 2.11 并发控制与事务的隔离级别

通过设置不同隔离级别和加锁来避免丢失修改、读脏、不可重复读。

**2.11.1 并发控制与事务的隔离级别**

该关卡任务已完成，实施情况本报告略过

**2.11.2 读脏**

将事务的隔离级别设为 READ COMMITTED以实现读脏。在事务 2 修改数据之后，使用sleep函数等待指定量的时间以确保事务 1 能够在事务 2 将修改撤销之前读取到修改后的数据。

**2.11.3 不可重复读**

为了在事务之间实现不可重复读，需要设置事务的隔离级别为 READ COMMITTED，然后先在事务 1 中修改数据并使用sleep函数等待一段时间，使得事务 2 在修改之前能够读到一次数据；提交事务1，之后令事务 2 使用sleep函数等待指定量的时间后再次读取该数据，即可使得事务 2 的两次读取到的数据不一致。

**2.11.4 幻读**

该关卡任务已完成，实施情况本报告略过

**2.11.5 主动加锁保证可重复读**

该关卡任务已完成，实施情况本报告略过

**2.11.6 可串行化**

该关卡任务已完成，实施情况本报告略过

## 2.12 备份+日志：介质故障与数据库恢复

使用备份与日志功能完成介质故障与数据库的恢复，掌握 MySQL 的恢复机制和MySQL备份与恢复工具。

**2.12.1备份与恢复**

该关卡任务已完成，实施情况本报告略过

**2.12.2备份+日志：介质故障的发生与数据库的恢复**

该关卡任务已完成，实施情况本报告略过

## 2.13数据库设计与实现

基于实际场景，完成数据库的设计与实现。

**2.13.1 从概念模型到MySQL实现**

根据任务要求，按需对各个主体创建表，并按需指定各个主体的主码、外码和其他码的缺省值、不可空性、不可重性等。由于代码过长且任务较为简单，此处不再赘述，也不贴出代码。

**2.13.2 从需求分析到逻辑模型**

该任务关卡跳过

**2.13.3建模工具的使用**

该任务关卡跳过

**2.13.4制约因素分析与设计**

在解决方案设计中，应当考虑以下制约因素：

社会因素：用户需求；

健康因素：任务对用户健康无影响；

安全因素：数据安全和网络安全；

法律因素：符合本地法律法规，遵守用户隐私权；

文化因素：不同文化背景下的用户使用习惯；

环境因素：不会对环境造成负面影响。

**2.13.5工程师责任及其分析**

解决问题时，应当全面考虑各种可能的影响因素，以确保解决方案全面、可持续，能够满足用户需求，并保护用户健康、安全和隐私。工程师应当努力满足社会需求，并致力于解决环境问题和保护用户健康、安全和隐私。同时也应学习相关法律法规，并在解决任务时遵守法律法规，杜绝违法违纪行为。

## 2.14 数据库应用开发(JAVA篇)

使用java语言完成简单的数据库应用系统的开发。

**2.14.1 JDBC体系结构和简单的查询**

该关卡任务已完成，实施情况本报告略过

**2.12.2 用户登录**

该关卡任务已完成，实施情况本报告略过

**2.12.3 添加新客户**

该关卡任务已完成，实施情况本报告略过

**2.12.4 银行卡销户**

该关卡任务已完成，实施情况本报告略过

**2.12.5 客户修改密码**

该关卡任务已完成，实施情况本报告略过

**2.12.6 事务与转账操作**

首先使用 PreparedStatement 对象执行一个 SQL 查询，检查转出账号是否存在，以及该账号是否是信用卡或者账户余额是否足够支付转账金额。如果满足任意一种情况，则会返回 false；然后，程序再次使用 Prepared Statement 对象执行一个 SQL 查询，检查转入账号是否存在。如果不存在，则会返回 false；最后，判断转入账号的类型是否为信用卡，如果是，则将转入账号的余额减去转账金额；如果不是，则将转入账号的余额加上转账金额。最后，将转出账号的余额减去转账金额并更新数据库中的信息。由于代码较长，此处不贴出代码。

**2.12.7 把稀疏表格转为键值对存储** 先通过 JDBC 驱动连接到 MySQL 数据库，然后执行一个查询语句，获取 entrance\_exam 表中的数据。接下来遍历结果集，对于每一行数据，检查其中的每一个成绩列，如果该成绩不为空，就执行一个插入操作，将该成绩插入到另一张表 sc 中。由于代码较长，此处不贴出代码。

## 2.15数据库的索引B+树实现

完成B+树索引的数据结构，并实现它的内部节点和叶子节点类型；然后分别实现索引的插入和删除功能，以及B+树迭代器。

**2.15.1 BPlusTreePage的设计**

该任务关卡跳过

**2.15.2 BPlusTreeInternalPage的设计**

该任务关卡跳过

# 3 课程总结

在本次课程实践任务涵盖了MySQL的数据对象管理与编程、数据处理任务、数据库的系统内核（如安全性控制、完整性控制、恢复机制、并发控制机制）、数据库的设计与实现、以及数据库应用系统的开发。我根据课程任务要求按需完成了对应的任务量，达到了本次课程实践任务的要求。

各个部分的主要工作情况如下：

1. 数据库、表、索引、视图、约束、存储过程、函数、触发器、游标等数据对象的管理与编程

该部分是数据库管理的基础，任务较为简单但重要，全部完成。

2. 数据查询，数据插入、删除与修改等数据处理相关任务

数据的增删改查是数据库管理的核心，除数据查询少量任务外其他均全部完成。

3. 数据库的安全性控制，完整性控制，恢复机制，并发控制机制等系统内核的实验

除个别关卡跳过外，基本上全部完成

4. 数据库的设计与实现

根据任务要求初步完成了从概念模型到MySQL的实现，跳过了从需求分析到逻辑模型以及建模工具的使用。

5. 数据库应用系统的开发

全部完成，实现了用java语言完成一个基础数据库应用系统的开发。

6. 数据库的索引B+树实现

该部分任务选择了跳过。

在本次课程实践任务中，我通过实验将课本上的理论知识转化为了实践结果，在实践中进一步增进了对理论知识的认知。例如，在数据查询这一任务系列中，通过不断练习数据查询这个最常用的功能，也不断深化了我对MySQL数据查询逻辑的认知；而通过使用java语言开发一个基础的数据库应用系统，也进一步加深了我对数据库应用系统的理解。

在这次实践中，我认为不足的地方是实验没有为MySQL一些便捷的函数（例如ifnull()）提供简明扼要的介绍，使得在一些关卡中花费了大量时间查阅资料以学习MySQL语言在课本和实验指导中都尚未提及的语法和函数。但总体上本次实验任务内容灵活，可以主动选择自己感兴趣的方向进行学习和实践，这一点我非常感谢。

# 附录