

**数据库系统原理实践报告**

|  |  |
| --- | --- |
| 专 业： | 计算机科学与技术学院 |
| 班 级： | CS2103 |
| 学 号： | U202115387 |
| 姓 名： | 杨至卓 |
| 指导教师： | 赵小松 |

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

2023 年 6 月 24 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](#_Toc118022464)

[2 任务实施过程与分析 2](#_Toc118022465)

[2.1 数据库、表与完整性约束的定义(Create) . 2](#_Toc118022466)

[2.1.1创建数据库 2](#_Toc118022466)

[2.1.2 创建表及主码约束 2](#_Toc118022466)

[2.1.3 创建外码约束 2](#_Toc118022466)

[2.1.4 CHECK 约束 3](#_Toc118022466)

[2.1.5 DEFAULT 约束 3](#_Toc118022466)

[2.1.6 UNIQUE 约束 3](#_Toc118022466)

[2.2 表结构与完整性约束的修改 (ALTER) 3](#_Toc118022466)

[2.2.1修改表名 4](#_Toc118022466)

[2.2.2添加与删除字段 4](#_Toc118022466)

[2.2.3修改字段 4](#_Toc118022466)

[2.2.4添加、删除与修改约束 4](#_Toc118022466)

[2.3 数据查询 (Select)之一 4](#_Toc118022467)

[2.3.1查询客户主要信息 5](#_Toc118022466)

[2.3.2邮箱为 null 的客户 5](#_Toc118022466)

[2.3.3既买了保险又买了基金的客户 5](#_Toc118022466)

[2.3.4办理了储蓄卡的客户信息 5](#_Toc118022466)

[2.3.5每份金额在 30000∼50000 之间的理财产品 5](#_Toc118022466)

[2.3.6商品收益的众数 5](#_Toc118022466)

[2.3.7未购买任何理财产品的武汉居民 6](#_Toc118022466)

[2.3.8持有两张信用卡的用户 6](#_Toc118022466)

[2.3.9购买了货币型基金的客户信息 6](#_Toc118022466)

[2.3.10投资总收益前三名的客户 6](#_Toc118022466)

[2.3.11黄姓客户持卡数量 7](#_Toc118022466)

[2.3.12客户理财、保险与基金投资总额 7](#_Toc118022466)

[2.3.13客户总资产 7](#_Toc118022466)

[2.3.14第 N 高问题 8](#_Toc118022466)

[2.3.15基金收益两种方式排名 8](#_Toc118022466)

[2.3.16持有完全相同基金组合的客户 9](#_Toc118022466)

[2.3.17购买基金的高峰期 10](#_Toc118022466)

[2.3.18至少有一张信用卡余额超过 5000 元的客户信用卡总金额 10](#_Toc118022466)

[2.3.19以日历表格式显示每日基金购买总金额 10](#_Toc118022466)

[2.4 数据查询 (Select)之二 11](#_Toc118022468)

[2.4.1查询销售总额前三的理财产品 11](#_Toc118022466)

[2.4.2投资积极且偏好理财类产品的客户 11](#_Toc118022466)

[2.4.3查询购买了所有畅销理财产品的客户 11](#_Toc118022466)

[2.4.4查找相似的理财产品 12](#_Toc118022466)

[2.4.5查询任意两个客户的相同理财产品数 12](#_Toc118022466)

[2.4.6查找相似的理财客户 12](#_Toc118022466)

[2.5 数据的插入、修改与删除(Insert,Update,Delete) 13](#_Toc118022466)

[2.5.1插入多条完整的客户信息 13](#_Toc118022466)

[2.5.2插入不完整的客户信息 13](#_Toc118022466)

[2.5.3批量插入数据 13](#_Toc118022466)

[2.5.4删除没有银行卡的客户信息 13](#_Toc118022466)

[2.5.5冻结客户资产 13](#_Toc118022466)

[2.5.6连接更新 13](#_Toc118022466)

[2.6 视图 14](#_Toc118022466)

[2.6.1创建所有保险资产的详细记录视图 14](#_Toc118022466)

[2.6.2基于视图的查询 14](#_Toc118022466)

[2.7 存储过程与事务 14](#_Toc118022469)

[2.7.1使用流程控制语句的存储过程 14](#_Toc118022466)

[2.7.2使用游标的存储过程 15](#_Toc118022466)

[2.7.3使用事务的存储过程 15](#_Toc118022466)

[2.8 触发器 16](#_Toc118022470)

[2.8.1 为投资表 property 实现业务约束规则 16](#_Toc118022471)

[2.13数据库设计与实现 17](#_Toc118022472)

[2.13.1从概念模型到 MySQL 实现 17](#_Toc118022466)

[2.13.2制约因素分析与设计 21](#_Toc118022466)

[2.13.3工程师责任及其分析 21](#_Toc118022466)

[2.14 数据库应用开发(JAVA篇) 22](#_Toc118022473)

[2.14.1 JDBC体系结构和简单的查询 22](#_Toc118022474)

[2.14.2用户登录 22](#_Toc118022466)

[2.14.3添加新用户 22](#_Toc118022466)

[2.14.4银行卡销户 22](#_Toc118022466)

[2.14.5客户修改密码 22](#_Toc118022466)

[2.14.6事务与转账操作 23](#_Toc118022466)

[2.14.7把稀疏表格转为键值对存储 23](#_Toc118022466)

[3 课程总结 25](#_Toc118022475)

# 1 课程任务概述

“数据库系统原理实践”是配合“数据库系统原理”课程独立开设的实践课，注重理论与实践相结合。本课程以MySQL为例，系统性地设计了一系列的实训任务，内容涉及以下几个部分：

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；
2. 数据查询，数据插入、删除与修改等数据处理相关任务；
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；
4. 数据库的设计与实现；
5. 数据库应用系统的开发(JAVA 篇)。

课程任务依托头歌实践教学平台。实验环境为Linux操作系统下的MySQL 8.0.28（主要为8.028版本，部分关卡使用8.022版本，使用中基本无差别）。在数据库应用开发环节，使用JAVA 1.8。

# 2 任务实施过程与分析

本次实践课程在头歌平台进行，实践任务均在平台上提交代码，所有完成的任务、关卡均通过了自动测评。本次实践最终完成了任务书中的2.1~2.8、2.10、2.13~2.14子任务，下面将重点针对其中的2.1~2.8、2.13~2.14任务阐述其完成过程中的具体工作。

## 2.1 数据库、表与完整性约束的定义(Create)

本节的六个关卡围绕数据库和表的创建展开，分别完成了数据库的创建、表

的创建以及表中主码、外码、CHECK、DEFAUT 和UNIQUE 等约束的建立。本任务已完成1-6关卡。

**2.1.1 创建数据库**

该关卡任务已完成，实施情况本报告略过。

**2.1.2 创建表及表的主码约束**

本关任务：在指定的数据库TestDb中创建一个表，并为表指定主码。

使用create table命令创建表t\_emp，并对主码id附加primary key约束。如图2.1所示。

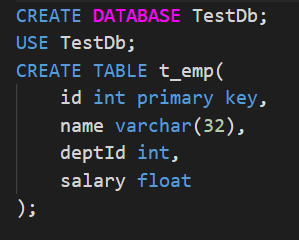


图2.1 t\_emp表的主码约束

**2.1.3 创建外码约束(foreign key)**

本关任务：创建外码约束（参照完整性约束）。

按关卡要求创建dept表和staff表，并给表staff创建外码，并给外码约束命名。代码中使用constraint语句创建了表staff的deptNo到表dept的deptNo的外码约束，将约束命名为FK\_staff\_deptno。如图2.2所示。

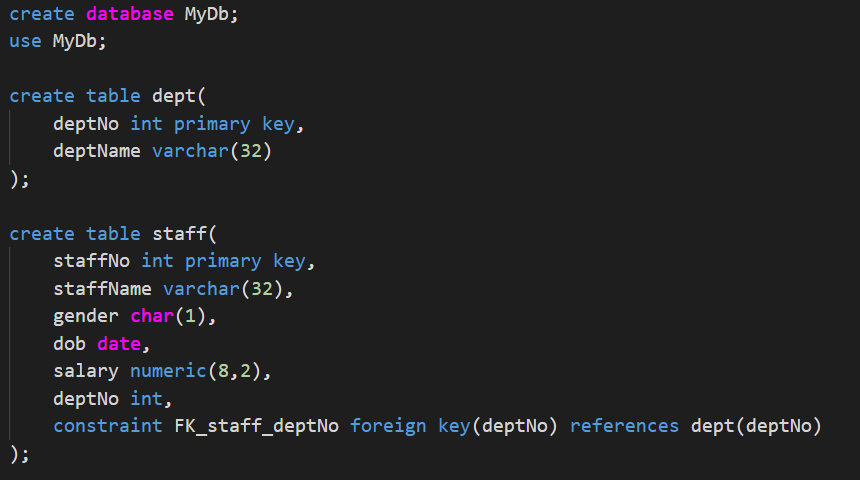


图2.2 staff表创建外码约束

**2.1.4 CHECK约束**

本关任务：为表创建 CHECK 约束。

根据任务要求，创建表products并添加check约束，代码如图2.3所示。 代码对price和brand的取值进行了check约束，并命名为CK\_products\_price，CK\_products\_brand。

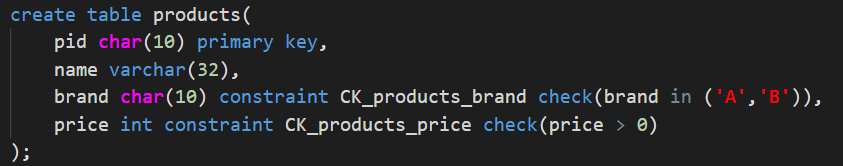


图2.3 products表创建check约束

**2.1.5 DEFAULT约束**

该关卡任务已完成，实施情况本报告略过。

**2.1.6 UNIQUE约束**

该关卡任务已完成，实施情况本报告略过。

## 2.2 表结构与完整性约束的修改(ALTER)

本节的四个关卡围绕数据库中表的基本修改操作展开，设计用alter语句

对表的定义进行修改，如更换/修改表名、列名、列的类型、列约束、表约束；

添加或删除列、约束等。本关已完成1-4关卡。

**2.2.1 修改表名**

该关卡任务已完成，实施情况本报告略过。

**2.2.2 添加与删除字段**

该关卡任务已完成，实施情况本报告略过。

**2.2.3 修改字段**

本关任务：修改字段。

根据任务要求，使用alter语句修改addressBook表的字段。首先使用本实验介绍中讲解的modify语句来修改QQ字段的数据类型，然后使用rename to语句来修改weixin字段的名称为wechat。如图2.4所示。

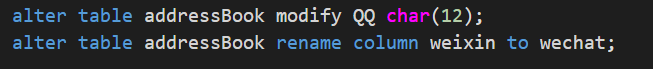


图2.4 修改字段类型和名称

**2.2.4 添加、删除与修改约束**

本关任务：添加、删除与修改约束。

使用alert + 表 + add + 完整性约束条件语句，对每个要求修选取合适的修改事项即可，代码如图 2.5 所示。

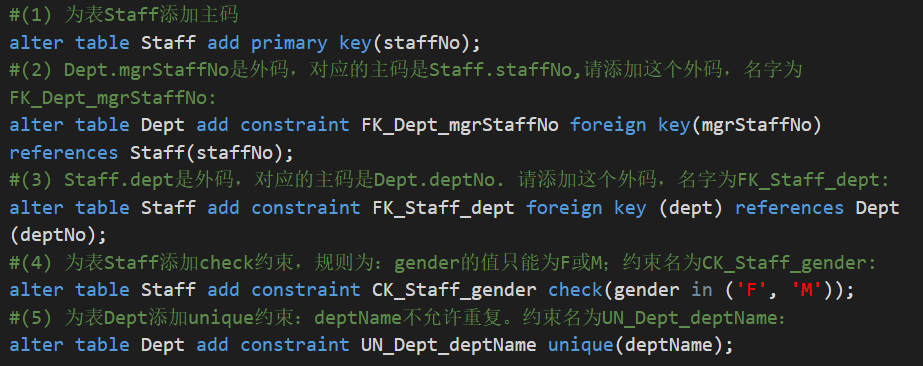


图2.5 为表添加完整性约束条件

## 2.3 数据查询(Select)之一

本节的19个关卡由浅到深围绕select语句在不同场景下的应用展开，并详细涵盖了where、group by、order by、exists等等书上已有的知识和用法，帮助我们理解和灵活运用select。本任务已完成1-19关卡。

**2.3.1 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。

**2.3.2 邮箱为null的客户**

本关任务：查询客户表(client)中没有填写邮箱的客户的编号、名称、身份证号、手机号。

使用where字句查找邮箱c\_mail = null的用户即可完成该条件查询，代码如图2.6所示。

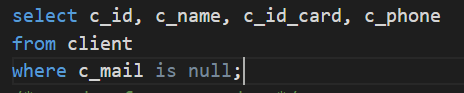


图2.6 邮箱为null的用户信息

**2.3.3 既买了保险又买了基金的客户**

该关卡任务已完成，实施情况本报告略过。

**2.3.4 办理了储蓄卡的客户信息**

本关任务：查询办理了储蓄卡的客户名称、手机号、银行卡号。

使用where语句和两个表的联合查询，条件为卡类型bank\_card.b\_type为储蓄卡且两个表中的c\_id相同即可。如图2.7所示。

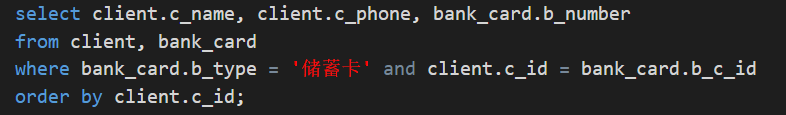


图2.7 办理了储蓄卡的用户信息

**2.3.5 每份金额在30000～50000之间的理财产品**

本关任务：查询金额在30000和50000之间的理财产品。

使用between … and语句即可实现条件查询，还需要掌握order语句进行结果排序，用desc关键字使结果降序排序，代码如图2.8所示。

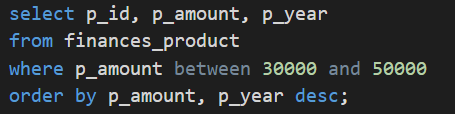


图2.8 金额在30000-50000的理财产品

**2.3.6 商品收益的众数**

本关任务：查询资产表中所有资产记录里商品收益的众数和它出现的次数。

在property表中基于pro\_income分组，利用count函数统计数量为presence，然后利用all来将每个分组的数量和所有分组比较，如果某个分组数量大于等于所有分组的数量，则为众数，此处用到嵌套查询和group by子句。代码如图2.9所示。

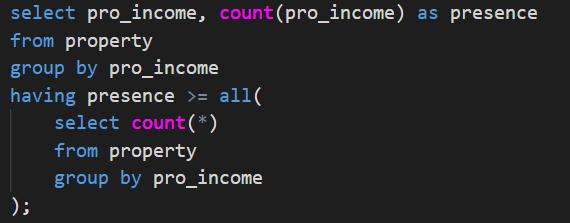


图2.9

**2.3.7 未购买任何理财产品的武汉居民**

该关卡任务已完成，实施情况本报告略过。

**2.3.8 持有两张信用卡的用户**

该关卡任务已完成，实施情况本报告略过。

**2.3.9 购买了货币型基金的客户信息**

该关卡任务已完成，实施情况本报告略过。

**2.3.10 投资总收益前三名的客户**

本关任务：查询投资总收益前三名的客户。

利用子查询对资产表property找出其中状态为“可用”的资产的pro\_c\_id和pro\_income，命名为新表pro，然后用where语句将client和pro中c\_id相同的元组连接，然后按c\_id分组和sum函数统计收益和total\_income并且按照total\_income降序排序，利用“limit 3”输出前3名用户即可。代码如图2.10所示。

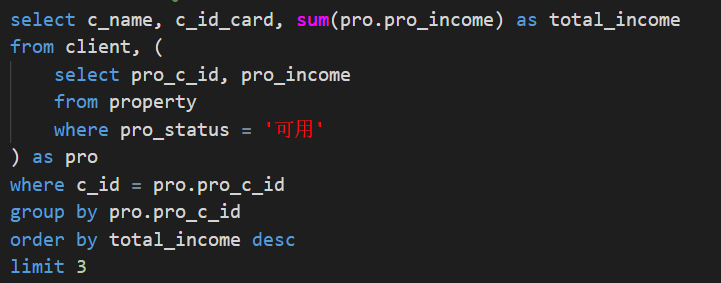
****

图2.10 投资总收益前三

**2.3.11 黄姓客户持卡数量**

该关卡任务已完成，实施情况本报告略过。

**2.3.12 客户理财、保险与基金投资总额**

该关卡任务已完成，实施情况本报告略过。

**2.3.13 客户总资产**

本关任务：查询客户在本行的总资产。

对于每个客户每种类型的资产总和，我们可以令where条件为p\_id = pro\_pif\_id和pro\_type为所需类型，各用一条select语句获取，将全部的结果利用 union all 取可重并集，再与用户表client以条件为c\_id = pro.id做左连接，最后再按c\_id分组，统计资产和total\_property并且按照total\_property降序排序即可。值得注意的是，题目条件告诉我们，如果银行卡类型的信用卡，那么总资产要减去信用卡的b\_balance，即透支金额。最后，我们还要处理空值，这里我使用了网上查到的ifnull（）语句来处理空值，如果值为空，则值修改为0。最终代码如图2.11所示。

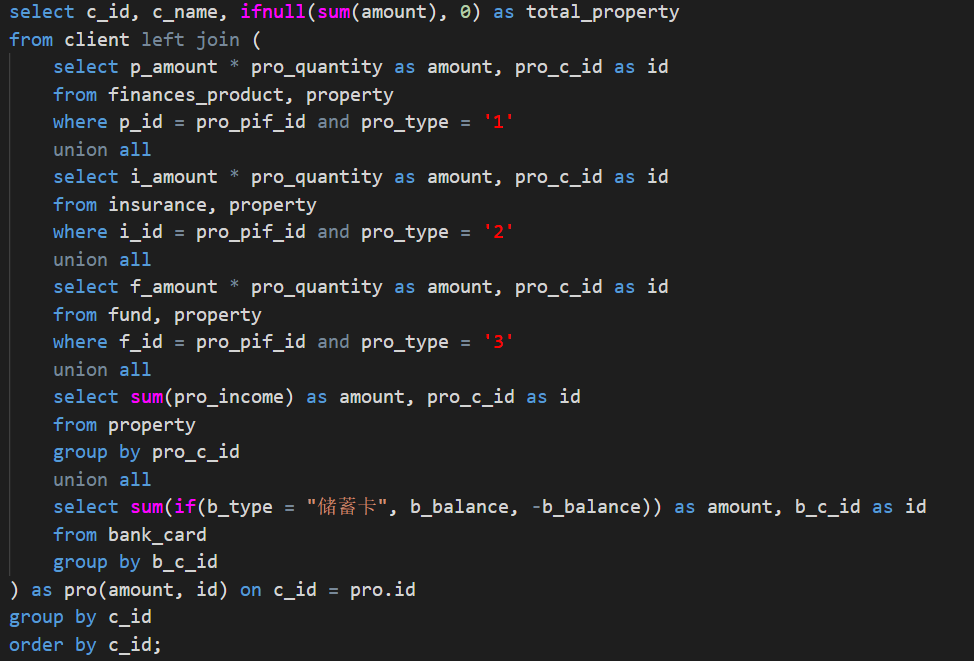


图2.11 查询客户总资产

**2.3.14 第N高问题**

本关任务：查询每份保险金额第4高保险产品的编号和保险金额。

在保险表insurance中利用网上查到的limit n, m找出第四高的保险金额，并用ifnull（）使保险数少于四时第四高金额为null，然后在外部查询中找出保险金额等于该金额的保险编号和金额即可，代码如图2.12所示。

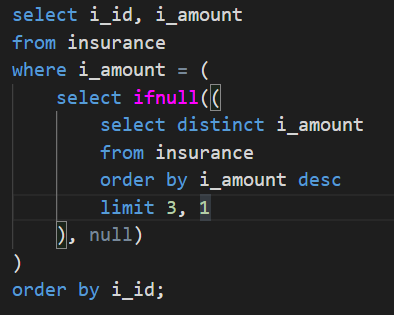


图2.12 查找第4高金额的保险信息

**2.3.15 基金收益两种方式排名**

本关任务：对客户基金投资收益实现两种方式的排名次。

以降序的基金投资收益和为关键字，直接调用库函数即可：函数rank()是名次不连续的排名，函数dense\_rank()是名次连续的排名。名次不连续代码和名次连续代码分别如图2.13、图2.14所示：

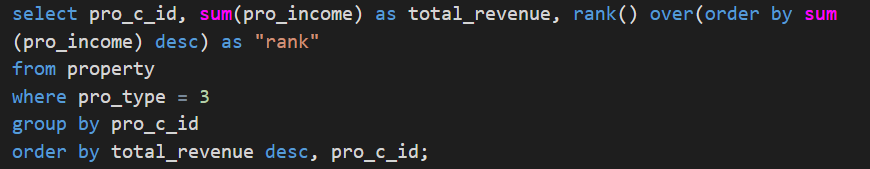


图2.13 不连续排名

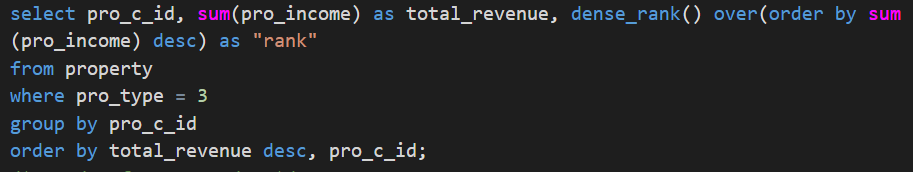


图2.14 连续排名

**2.3.16 持有完全相同基金组合的客户**

本关任务：查询持有完全相同基金组合的客户。

本关可以用离散数学的理论，将问题变成查找client中，不存在基金a，用户c拥有而用户d不拥有，也不存在基金b，用户c不拥有而用户d拥有，并且c和d至少拥有一个基金的用户组c、d，这样就保证c、d拥有完全相同的基金组。使用not exists和exists语句来实现这两个查找。代码如图 2.15 所示。

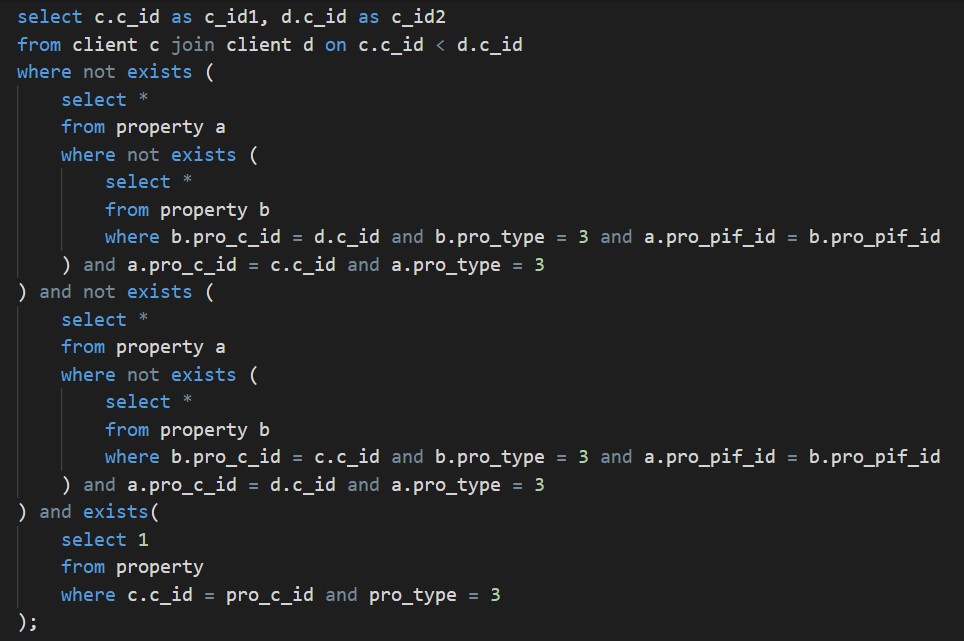


图2.15 持有完全相同基金的客户

**2.3.17 购买基金的高峰期**

该关卡任务已完成，实施情况本报告略过。

**2.3.18 至少有一张信用卡余额超过5000元的客户信用卡总余额**

该关卡任务已完成，实施情况本报告略过。

**2.3.19 以日历表格式显示每日基金购买总金额**

本关任务：以日历表格式显示2022年2月每周每日基金购买总金额。

事先已经得知2022.2.7是第六周，因此对date\_part取得“week”结果减5 就可以得到在2月的周次。对资产表property和基金表fund做自然连接，并按交易时间是星期几分别计算出当日购买总金额，再按照周次相等group到一起，用聚集函数sum计算每日基金购买总金额。代码如图2.16所示

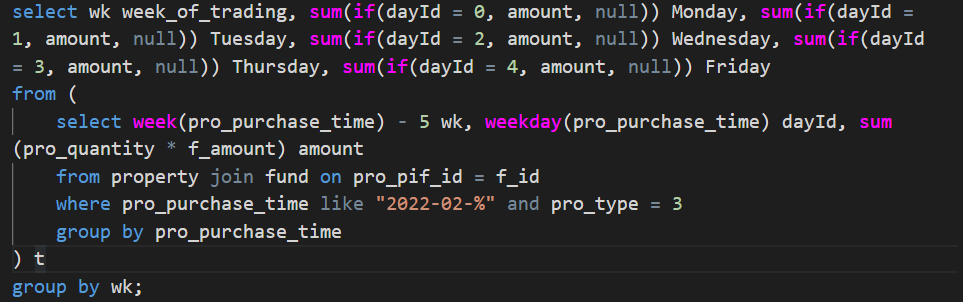


图2.16 日历表格式查看每日基金购买总金额

## 2.4 数据查询(Select)之二

本节的6个关卡在“数据查询(Select)之一”的基础上，讲解了一些函数的运用和更复杂的select语句的用法。本任务已完成1-6关卡。

**2.4.1 查询销售总额前三的理财产品**

该关卡任务已完成，实施情况本报告略过。

**2.4.2 投资积极且偏好理财类产品的客户**

本关任务：找到投资积极且偏好理财类产品的客户。

利用嵌套的两个子查询分别查找用户购买的理财产品数目和基金数目，和用户编号一起形成两个表t1和t2，然后用客户编号pro\_c\_id相等连接两个表，找出理财产品数目t1.cnt1大于基金数目t2.cnt2的客户即可。代码如图2.17所示。

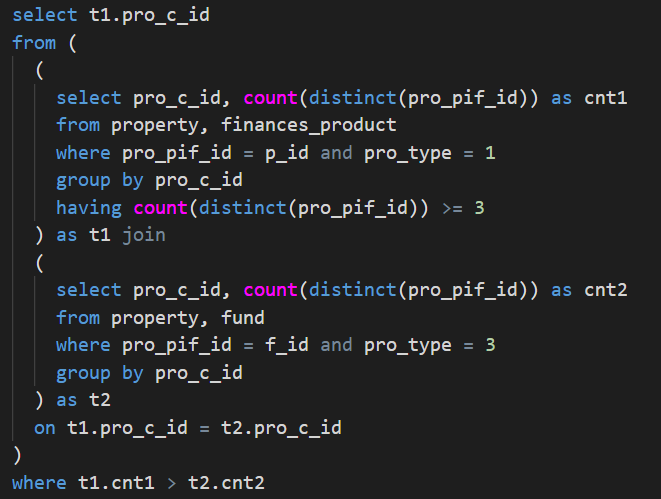


图2.17 投资积极且偏好理财产品的客户

**2.4.3 查询购买了所有畅销理财产品的客户**

本关任务：查询购买了所有畅销理财产品的客户。

本关使用离散数学语言可以转化为：查找没有一个畅销理财产品是他没有买的的用户。这样我们使用两个资产表a、c。首先找出所有畅销理财产品，取出他们的产品编号作为表b，然后用两个not exists语句，将a、b、c表通过b和c的产品编号相同以及a和c的客户编号相同还有c的产品类型为理财产品这三个条件结合，查找出没有一个畅销理财产品是他没有买的的用户。代码如图2.18所示。

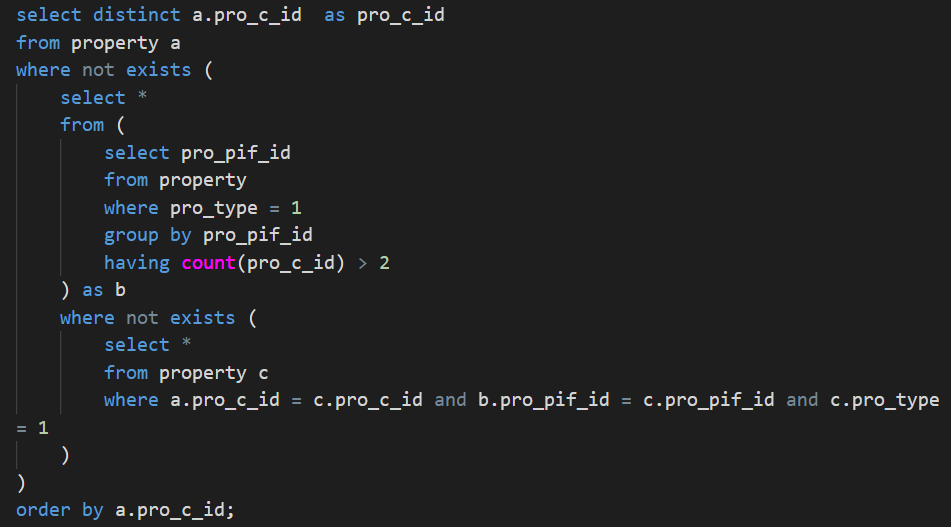


图2.18 购买了所有畅销理财产品的客户

**2.4.4 查找相似的理财产品**

该关卡任务已完成，实施情况本报告略过。

**2.4.5 查询任意两个客户的相同理财产品数**

本关任务：查询任意两个客户的相同理财产品数。

将两个资产表property p1和p2根据产品编号pif\_id相同连接起来，然后选择其中p1和p2类型pro\_type相同且客户编号c\_id不同的元组，然后通过group分组，利用count函数计算客户相同的理财产品数total\_count。代码如图2.19所示。

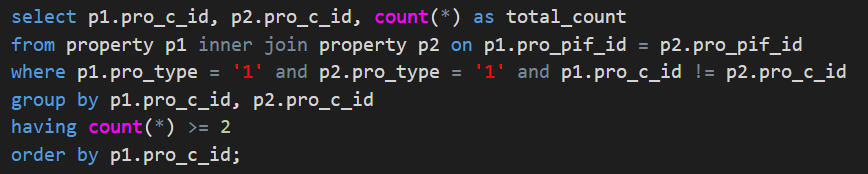


图2.19 任意两个客户的相同理财产品数

**2.4.6 查找相似的理财客户**

该关卡任务已完成，实施情况本报告略过。

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

本节的6个关卡围绕Insert, Update, Delete语句，在不同场景下展开应用，对表中的数据进行插入、修改和删除操作。本任务已完成1-6关卡。

**2.5.1 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。

**2.5.2 插入不完整的客户信息**

该关卡任务已完成，实施情况本报告略过。

**2.5.3 批量插入数据**

本关任务：向客户表client批量插入数据。

将insert语句中的value字段更换为select查询语句即可，因为两个表结构完全相同，这样就可以实现批量插入数据。代码如图2.20所示。

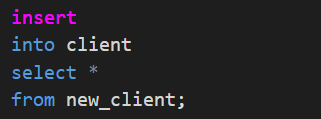


图2.20 插入select数据

**2.5.4 删除没有银行卡的客户信息**

该关卡任务已完成，实施情况本报告略过。

**2.5.5 冻结客户资产**

本关任务：冻结客户的投资资产。

使用一条update语句结合where条件语句，判断该客户是否在电话号码为“13686431238”子查询客户中即可实现。代码如图2.21所示。

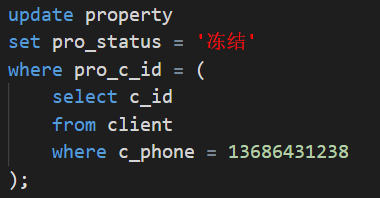


图2.21 冻结客户资产

**2.5.6 连接更新**

本关任务：根据客户表的内容修改资产表的内容。

本关仍使用update语句，使用property表和client表，用select语句找出两个表中客户编号c\_id相同的元组，取出client表中的客户身份证号c\_id\_card，并更新到property表中的元组中。代码如图2.22所示。

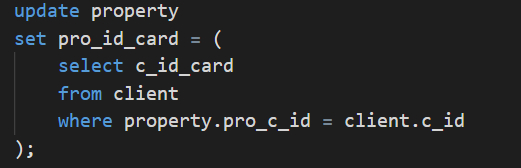


图2.22 连接更新

## 2.6 视图

本节的2个关卡围绕视图的创建与使用展开。在select语句的基础上创建视图。本任务已完成1-2关卡。

**2.6.1 创建所有保险资产的详细记录视图**

本关任务：创建所有保险资产的详细记录视图。

根据任务要求编写select语句，并在前面加上“create view ... as”即可完成创建视图的工作，代码如图2.23所示。

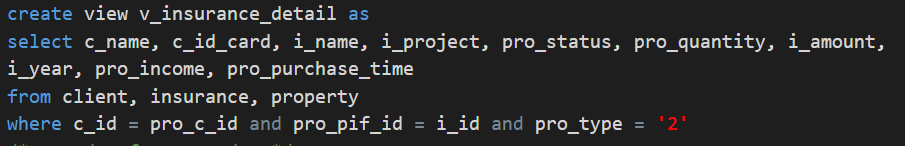


图2.23 视图v\_insurance\_detail的创建

**2.6.2 基于视图的查询**

该关卡任务已完成，实施情况本报告略过。

## 2.7 存储过程与事务

本节的3个关卡分别涉及使用流程控制语句的存储过程、使用游标的存储过程和使用事务的存储过程。本任务已完成1-3关卡。

**2.7.1 使用流程控制语句的存储过程**

本关任务：创建一个存储过程，向表fibonacci插入斐波拉契数列的前n项。

利用“create procedure ... as...begin...end”定义一个存储过程，在as和begin之间定义变量并且赋初值，begin和end之间是流程语句，算法的思想是共通的，代码如图2.24所示。

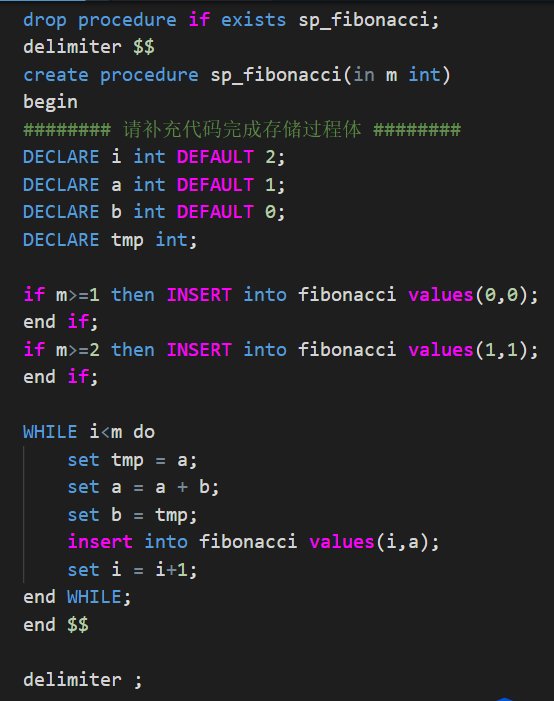


图2.24 流程控制语句

**2.7.2 使用游标的存储过程**

该关卡任务已完成，实施情况本报告略过。

**2.7.3 使用事务的存储过程**

本关任务：编写实现转账功能的存储过程。

根据题意编写判断条件，如果不合法则将返回值设置为0，并且跳出pro代码段部分。如果合法，则先判断付款方和收款方的卡类型，如果是信用卡则要把转账金额设置为负数，最后使用update更新即可，然后将返回值设置为1。代码如图2.25所示。

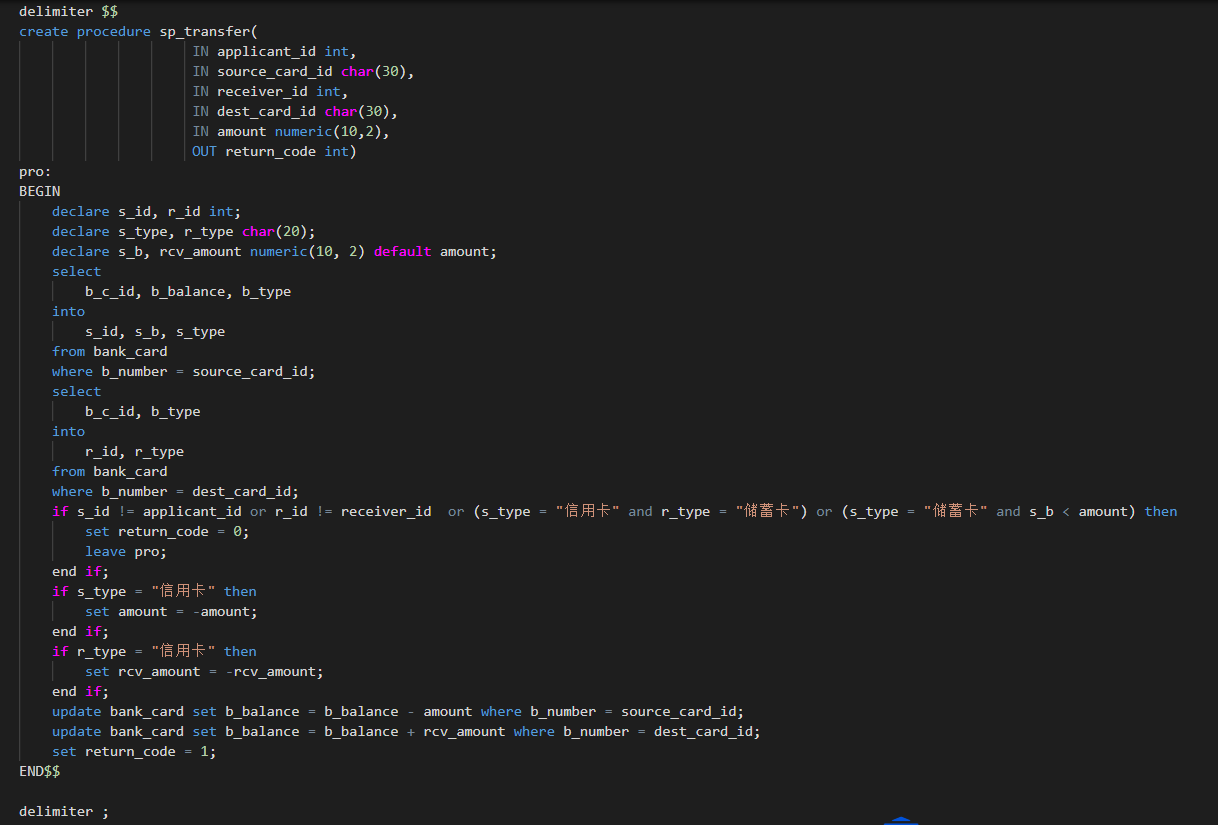


图2.25 使用事务的存储过程

## 2.8 触发器

本节的1个关卡涉及触发器的使用。本任务已完成1关卡。

**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**

本关任务：为资产表property编写一个触发器，以实现任务所要求的完整性业务规则。

根据任务要求声明“BEFORE INSERT ON property”类型的触发器，如果执行过程中发现有投资的类型和它的id不匹配，即在对应类型的表的select子查询中无法找到该id，则说明违反完整性业务规则，利用concat（）函数编辑报错信息msg；当数据不合法时设置出错信息：SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = msg。代码如图2.26所示。

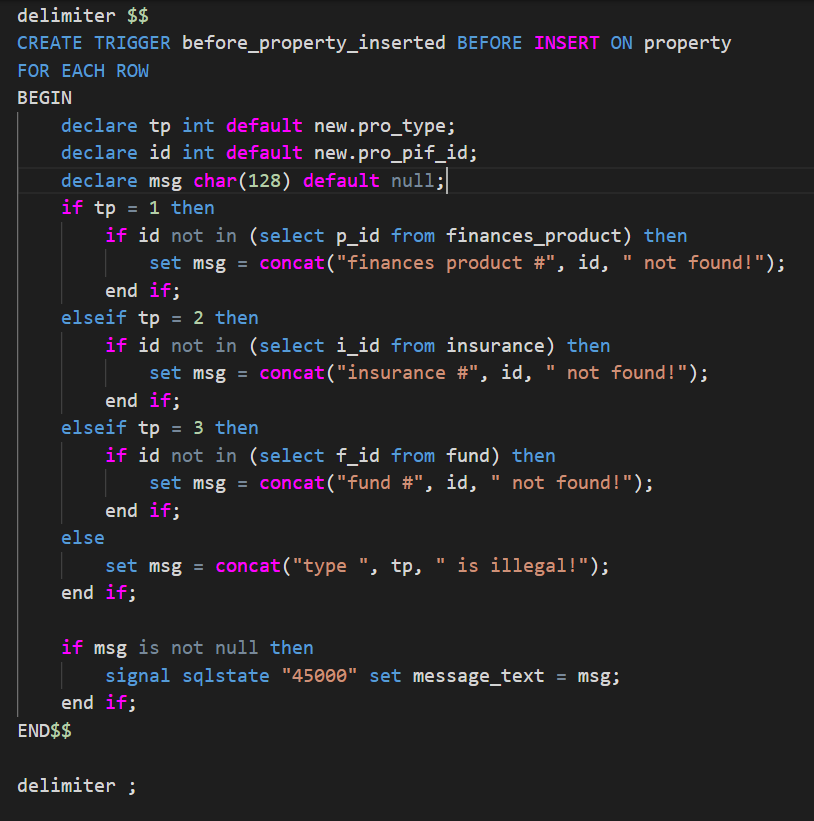


图2.26 创建触发器before\_property\_inserted

## 2.13 数据库设计与实现

本节的3个关卡涉及数据库设计与实现相关内容，包括从概念模型到OpenGauss实现、E‐R图的构建、建模工具的使用等。本任务完成关卡1。

**2.13.1 从概念模型到MySQL实现**

本关任务：将已建好的概念模型，变成MySQL物理实现。

使用create table，根据题目提供的概念模型建立数据库即可。代码如下所示。

CREATE TABLE user(

    user\_id INT AUTO\_INCREMENT PRIMARY KEY,

    firstname VARCHAR(50) NOT NULL,

    lastname VARCHAR(50) NOT NULL,

    dob date NOT NULL,

    sex CHAR(1) NOT NULL,

    email VARCHAR(50),

    phone VARCHAR(30),

    username VARCHAR(20) NOT NULL UNIQUE,

    password CHAR(32) NOT NULL,

    admin\_tag tinyint DEFAULT 0 NOT NULL

);

CREATE TABLE passenger(

     passenger\_id INT AUTO\_INCREMENT PRIMARY KEY,

     id CHAR(18)NOT NULL UNIQUE,

     firstname VARCHAR(50) NOT NULL,

     lastname VARCHAR(50) NOT NULL,

     mail VARCHAR(50),

     phone VARCHAR(20) NOT NULL,

     sex CHAR(1) NOT NULL,

     dob DATE

);

CREATE TABLE airport (

     airport\_id int AUTO\_INCREMENT PRIMARY KEY,

     iata CHAR(3) NOT NULL UNIQUE,

     icao CHAR(4) NOT NULL UNIQUE,

     name VARCHAR(50) NOT NULL,

     city VARCHAR(50),

     country VARCHAR(50),

     latitude NUMERIC(11, 8),

     longitude NUMERIC(11, 8),

     INDEX(name)

);

CREATE TABLE airline (

     airline\_id int AUTO\_INCREMENT PRIMARY KEY,

     name VARCHAR(30) NOT NULL,

     iata CHAR(2) NOT NULL UNIQUE,

     airport\_id INTEGER NOT NULL,

     FOREIGN KEY(airport\_id) REFERENCES airport(airport\_id)

);

CREATE TABLE airplane (

     airplane\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,

     type VARCHAR(50) NOT NULL,

     capacity SMALLINT NOT NULL,

     identifier VARCHAR(50) NOT NULL,

     airline\_id INTEGER NOT NULL,

     FOREIGN KEY(airline\_id) REFERENCES airline(airline\_id)

);

create table flightschedule (

     flight\_no char(8) primary key,

     departure time not null,

     arrival time not null,

     duration smallint not null,

     monday tinyint default 0,

     tuesday tinyint default 0,

     wednesday tinyint default 0,

     thursday tinyint default 0,

     friday tinyint default 0,

     saturday tinyint default 0,

     sunday tinyint default 0,

     airline\_id INTEGER not null,

     `from` INTEGER not null,

     `to` INTEGER not null,

     foreign key(airline\_id) references airline(airline\_id),

     foreign key(`from`) references airport(airport\_id),

     foreign key(`to`) references airport(airport\_id)

);

CREATE TABLE flight (

     flight\_id INTEGER AUTO\_INCREMENT PRIMARY KEY,

     departure DATETIME NOT NULL,

     arrival DATETIME NOT NULL,

     duration SMALLINT NOT NULL,

    airline\_id INTEGER NOT NULL,

    airplane\_id INTEGER NOT NULL,

    flight\_no CHAR(8) NOT NULL,

    `from` int NOT NULL,

     `to` INTEGER NOT NULL,

**2.13.2** **制约因素分析与设计**

在从实际问题的建模到数据库的概念模型和逻辑模型的构建过程中，需要考 虑若干制约因素。以机票订票系统为例，系统需要考虑到旅客的实际情况，旅客 可以多次乘坐飞机，一张机票肯定是某个用户为某个特定的旅客购买的特定航班 的机票，所以机票信息不仅跟乘坐人有关，同时还需要记录购买人信息(虽然两 者有时是同一人)。此外，对于系统的权限也存在若干要求，例如实体“用户”就可以根据权限分成两类，用户分两类：普通用户可以订票，管理用户有权限维护和管理整个系统的运营。

**2.13.3 工程师责任及其分析**

社会方面，工程师应该能够基于工程相关背景知识进行合理分析，评价专业工程实践和复杂工程问题解决方案对社会、健康、安全、法律以及文化的影响，并理解应承担的责任。安全方面，工程师应该尽可能考虑系统中存在的安全漏洞，安全性是所有系统用户关心的重要命题；科学发展方面，工程师应该能够基于科学原理并采用科学方法对复杂工程问题进行研究，包括设计实验、分析与解释数据、并通过信息综合得出合理有效的结论。

## 2.14 数据库应用开发(JAVA篇)

本节的7个关卡从JDBC体系结构出发，涉及java开发数据库应用的基本知识。本任务已完成1-7关卡。

**2.14.1 JDBC体系结构和简单的查询**

本关任务：查询client表中邮箱非空的客户信息，列出客户姓名，邮箱和电话。

使用Java的Class.forName()方法，将驱动程序的类文件动态加载到内存中，并将其自动注册。加载驱动程序后，使用DriverManager.getConnection(String url, String user,String password)方法建立连接。在使用Statement对象执行SQL语句之前，需要使用Connection对象的createStatement()方法创建Statement的一个实例。创建Statement对象后，可以使用它来执行一个SQL语句，使用resultSet.next()遍历ResultSet输出相应的信息即可，代码如图2.27所示。

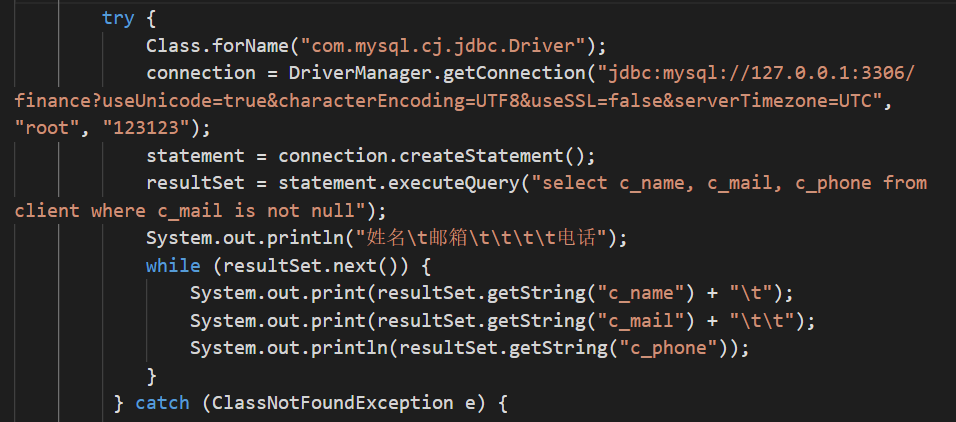


图2.27 JDBC体系结构和查询

**2.14.2 用户登录**

该关卡任务已完成，实施情况本报告略过。

**2.14.3 添加新客户**

本关任务：编程完成向客户表client插入记录的方法。

编写 SQL 语句，用PreparedStatement类执行即可，代码如图2.28所示。

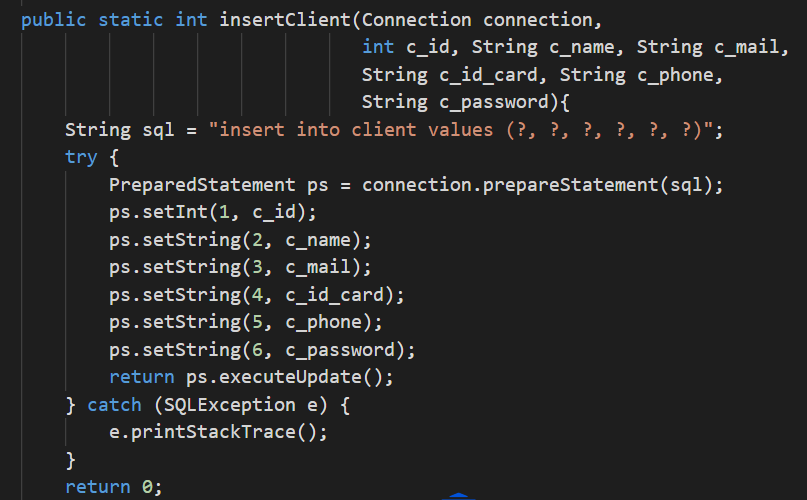


图2.28 添加新客户

**2.14.4 银行卡销户**

该关卡任务已完成，实施情况本报告略过。

**2.14.5 客户修改密码**

本关任务：编写修改客户登录密码的方法。

先判断用户是否存在，再判断密码是否正确，最后利用update语句修改密码。代码如图2.29所示。

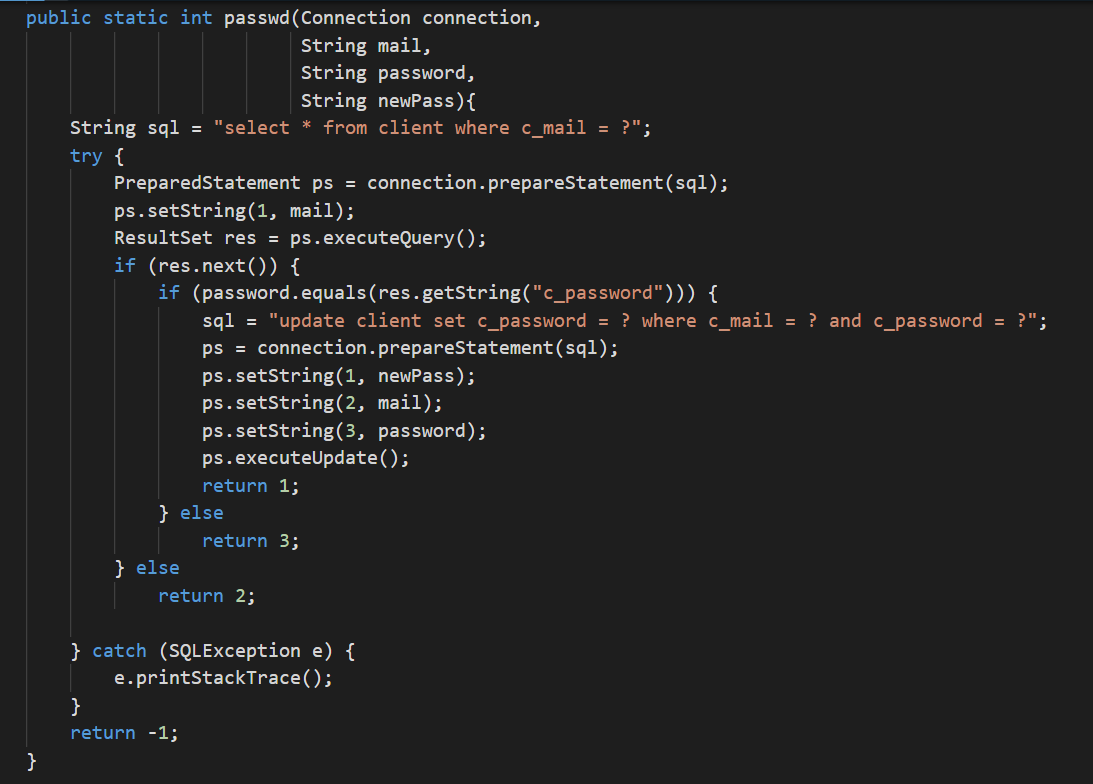


图2.29 客户修改密码

**2.14.6 事务与转账操作**

该关卡任务已完成，实施情况本报告略过。

**2.14.7 把稀疏表格转为键值对存储**

本关任务：将一个稀疏的表中有保存数据的列值，以键值对 “(列名, 列值)” 的形式转存到另一个表中，这样可以直接丢失没有值列。

选出entrance\_exam表中所有数据，对每个学生枚举所有学科，如果该学生存在该学科的成绩，就将其插入sc表中。代码如图2.30所示。

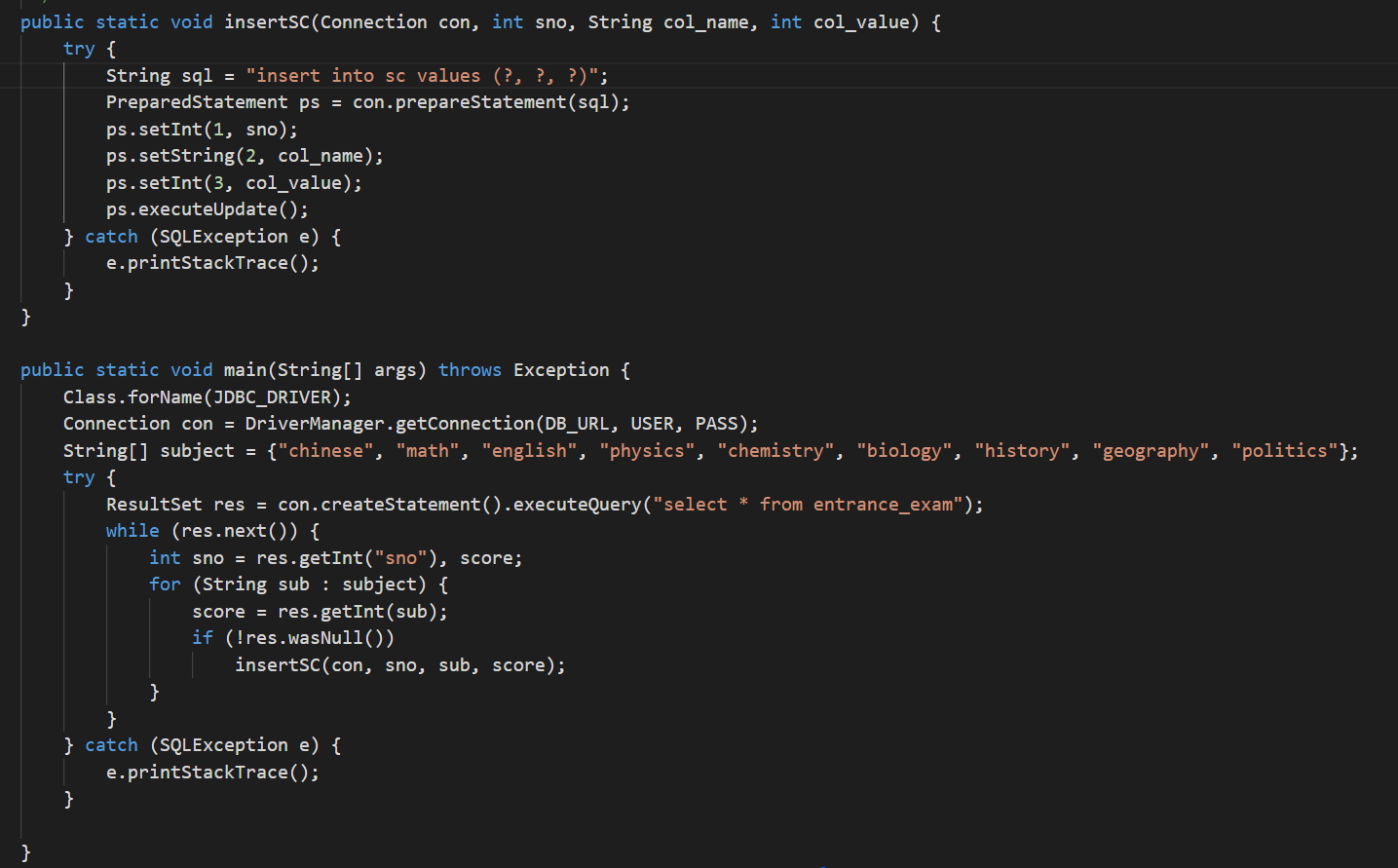


图2.30 稀疏表格转化为键值对

# 3 课程总结

本次课程实验从数据库、表的定义出发，分别完成了表的完整性约束的创建和修改、数据 查询、数据的插入、修改与删除、视图的创建与使用、存储过程与事务、触发器、用户自定义 函数、安全性控制、并发控制与事务的隔离级别、数据库的备份与日志、数据库的设计与实现和数据库应用开发等15个实训实验。

在课程实验的实施过程中，通过合理使用搜索引擎查阅相关资料，学习MySQL的一些复杂语法和一些经典问题的简单处理方法简化编程，让我能够熟练使用MySQL实现一些复杂的查询，同时也感受到了MySQL的灵活性。在基于JDBC的数据库应用开发中，我切实体会到了开发过程中代码安全的重要性，如果不遵守规范，则有可能会因为SQL注入攻击造成非常严重的后果。

此次课程实验内容充实完整，引导性强，完成实验的收获也很大，能够让学生更加深入地理解数据库是如何运作的。希望在之后的课程中老师们能够讲解一些java，这样我们在完成java部分实验的过程中会更加理解，也能减轻同学们的压力。最后，衷心地感谢课程组老师们对实验内容的精心打造以及实验过程中的悉心指导！