

**数据库系统原理实践报告**

|  |  |
| --- | --- |
| 专 业： | 计算机科学与技术 |
| 班 级： |  |
| 学 号： |  |
| 姓 名： |  |
| 指导教师： | 赵小松 |

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

2023年06月26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](#_Toc138703288)

[2 任务实施过程与分析 2](#_Toc138703289)

[2.1 数据库、表与完整性约束的定义 (Create) 2](#_Toc138703290)

[2.2表结构与完整性约束的修改 (ALTER) 3](#_Toc138703291)

[2.3数据查询 (Select) 5](#_Toc138703292)

[2.4 数据的插入、修改与删除 14](#_Toc138703293)

[2.5 视图 15](#_Toc138703294)

[2.6 存储过程与事务 16](#_Toc138703295)

[2.7 触发器 18](#_Toc138703296)

[2.8 用户自定义函数 19](#_Toc138703297)

[2.9 安全性控制 20](#_Toc138703298)

[2.10 并发控制与事务的隔离级别 21](#_Toc138703299)

[2.11 数据库应用开发（JAVA 篇） 24](#_Toc138703300)

[2.12 备份 + 日志：介质故障与数据库恢复 29](#_Toc138703301)

[2.13 数据库设计与实现 29](#_Toc138703302)

[2.14 数据库的索引 B+树实现 31](#_Toc138703303)

[3 课程总结 36](#_Toc138703304)

# 1 课程任务概述

《数据库系统原理实践》是一个与《数据库系统原理》课程配套的实践课程，旨在将理论与实践相结合。本课程以OpenGauss为例，通过系统性的设计一系列实训任务，涵盖以下几个主要内容：

1. 数据对象的管理与编程：包括数据库、表、索引、视图、约束、存储过程、函数、触发器、游标等数据对象的创建、修改和管理。
2. 数据处理相关任务：重点关注数据查询以及数据插入、删除和修改等数据处理操作，通过实践任务来加深对这些操作的理解和掌握。
3. 数据库的安全性控制、完整性控制、恢复机制和并发控制机制等系统内核的实验：通过实践任务，学习数据库的安全性控制方法、数据完整性保护策略、故障恢复机制以及并发控制机制等核心概念和技术。
4. 数据库的设计与实现：通过实践任务，学习数据库设计的基本原则和方法，了解关系数据库模型的设计过程，并进行实际的数据库设计和实现。
5. 数据库应用系统的开发（JAVA篇）：在数据库应用开发环节，使用JAVA 1.8进行实践任务，掌握基于数据库的应用系统开发方法和技巧。
6. 数据库内核实验（B+树）：通过实践任务，深入了解数据库内部的数据结构和算法，重点研究B+树索引的原理、实现和优化方法。

本课程借助头歌实践教学平台进行教学，相关课程的URL和邀请码将由教师发布。实验环境为基于Linux操作系统的OpenGauss 2.1版本，而在数据库应用开发环节，使用JAVA 1.8作为开发语言。

通过《数据库系统原理实践》课程的学习，我将深入理解数据库系统的原理和实践，掌握数据库的设计、管理和应用开发的技能，为将来在数据库领域的工作和研究打下坚实的基础。

# 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头歌平台上的全部关卡，下面将重点针对其中的部分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。

**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 创建表及主码约束**

本关任务：在指定的数据库中创建一个表，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 2.1.1所示。

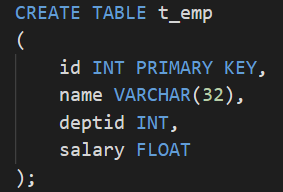


图2.1.1 创建表和主码约束

**2.1.3 创建外码约束**

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，代码如图2.1.2所示。

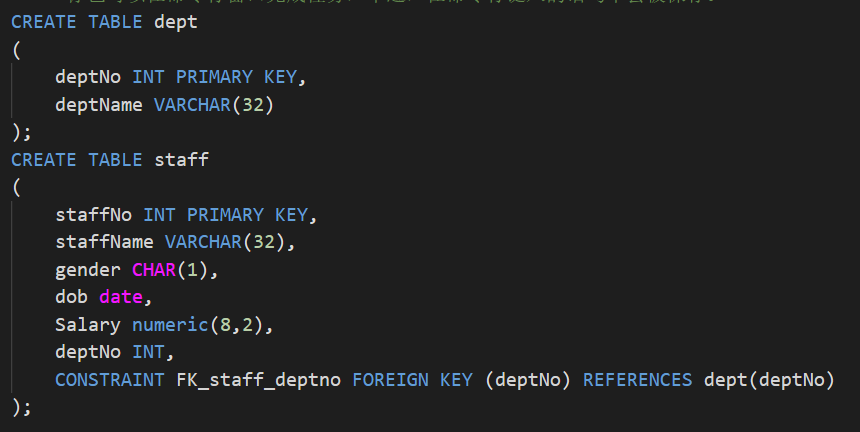


图2.1.2 外码约束

代码中的第 146行创建了表 staff 的 deptNo 到表 dept 的 deptNo 的外码约束，约束名为 FK\_staff\_deptno。

**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 2.1.3所示。代码的 10、11两行对 price和 brand的取值进行了约束。

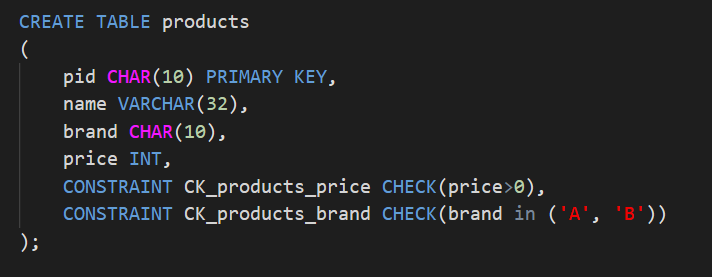


图 2.1.3 CHECK约束

**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。

**2.2.1 修改表名**

本关任务：修改表的名称，将表名 your\_table 更改为 my\_table 。 ALTER TABLE 的简化的语法为：

ALTER TABLE 表名[修改事项 [, 修改事项] ...]

根据任务要求选择“RENAME [TO|AS] 新表名” 修改事项，代码如图 2.2.1 所示。



图 2.2.1 修改表名

**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**

该关卡任务已完成，实施情况本报告略过。

**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。

对每个要求修选取合适的修改事项即可，代码如图 2.2.2 所示。 图 2.2.2 添加、修改与删除约束



图 2.2.2 添加、修改与删除约束

## 2.3数据查询 (Select)

本节的 25 个关卡由浅到深围绕 select 语句在不同场景下的应用展开。

**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 字句即可完成该条件查询，代码如图 2.3.1 所示。

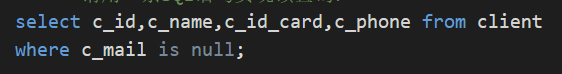


图 2.3.1 邮箱为 null

**2.3.3 既买了保险又买了基金的客户**

本关任务：查询既买了保险又买了基金的客户的名称和邮箱。使用 IN 子查 询和 and 来同时满足两个条件即可，代码如图 2.3.2 所示。

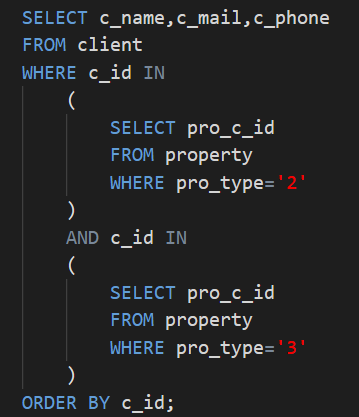


图 2.3.2 既买了保险又买了基金

**2.3.4 办理了储蓄卡的客户信息**

该关卡任务已完成，实施情况本报告略过。

**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 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 2.3.3 所示。

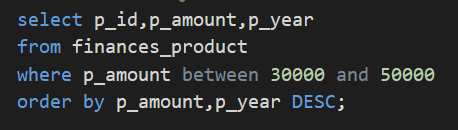


图 2.3.3 理财产品金额在 30000~5000

**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，如 果某个分组数量大于等于所有分组或者数量最大的分组，则为众数，此处用到嵌 套查询和 group by 子句。代码如图 2.3.4 所示。

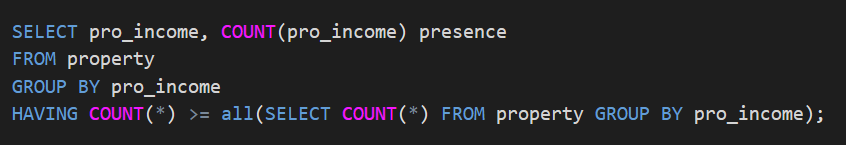


图 2.3.4 商品收益众数

**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“join” 对用户表 client 和资产表 property 做自然连接，然后按 c\_id 分组统计收益和并降序排序，利用“limit 3” 输出前 3 名用户即可。注 意 sum（）的使用和限制资产状态不为“冻结”。

代码如图 2.3.5 所示。

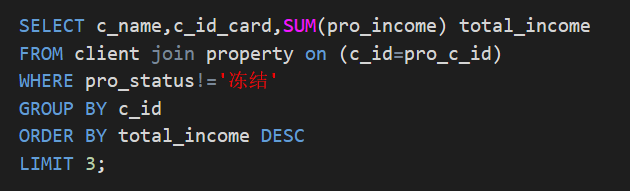


图 2.3.5 总收益前三名

**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，可以用一条 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 做自然连接，最后再按 c\_id 分组，统计资产和并降序排序即可。值得注意的一个常识：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。还要学习 coalesce（）的使用用来处理空值。最终代码如图 2.3.6 所示。

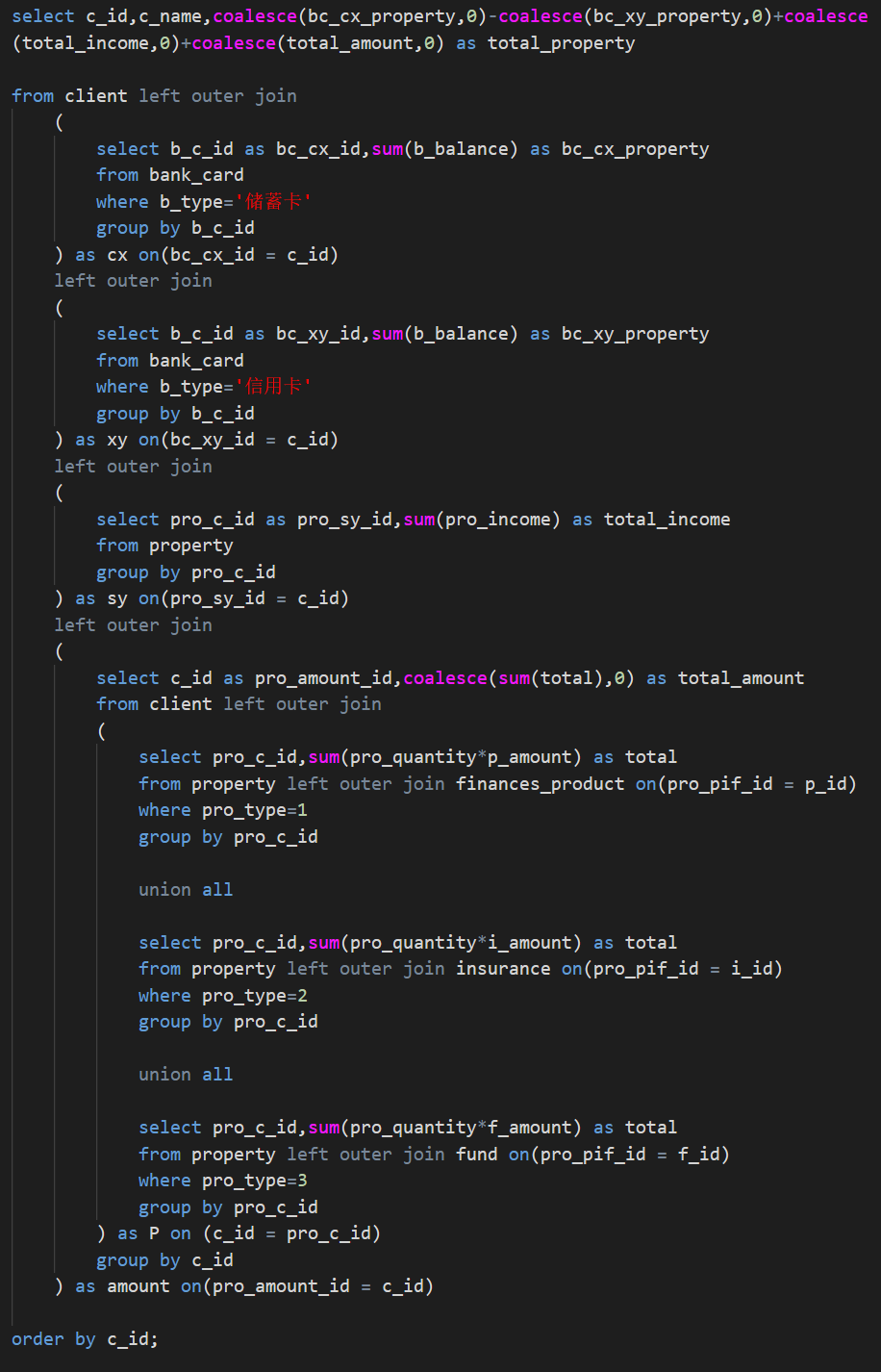


图 2.3.6 客户总资产

**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 dense\_rank（）选对不同的保险金额进行排序， 再利用 “rank=4”判断 即可获取 select 结果第 4 高的保险金额。最后在外 部查询中判断保险金额是否等于该金额即可，代码如图 2.3.7 所示。

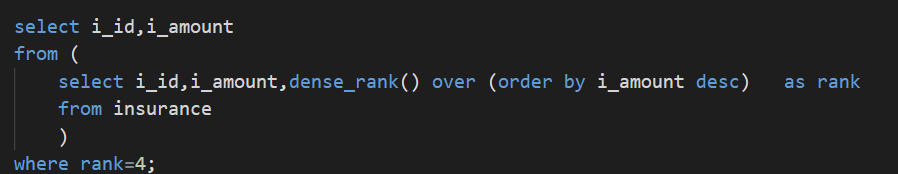


图 2.3.7 第 N 高问题

**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 2.3.8，图 2.3.9 所示：

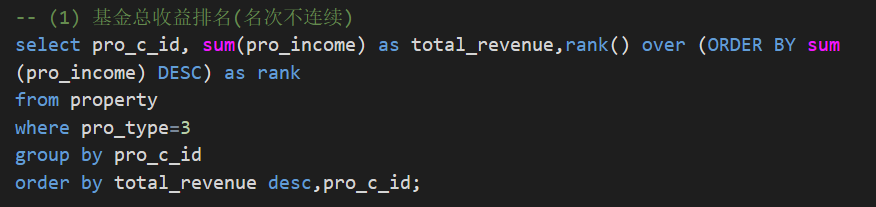


图 2.3.8 名次不连续

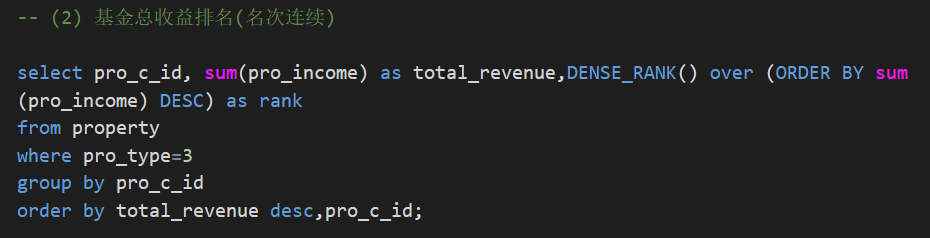


图 2.3.9 名次连续

**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。

首先最内层两个相同的查询，把客户 c\_id 和其购买的基金 f\_id 的结果保存 到表中，之后进行自然连接得到若干三元组，存储着两个不同客户购买同一个基 金的信息，根据客户 c\_id 进行分组，与另一个查询每个客户购买基金数量的表 进行对比，确认购买同一基金组合的两个客户均满足：这个基金组合是二人购买 10 的全部基金。代码如图 2.3.10 所示。

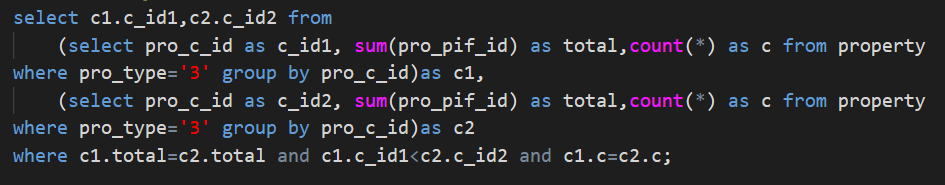


图 2.3.10 完全相同基金组合

**2.3.17 购买基金的高峰期**

本关任务：查询 2022 年 2 月购买基金的高峰期，如果连续三个交易日， 投资者购买基金的总金额超过 100 万，则称这连续的几日为投资者购买基金的 高峰期。

在最外层查询的 where 条件中，已经查询了所有当日购买金额大于 1000000 的日期，只需查询以下三种情况，当前日期分别为三个连续金额超过 1000000 的交易日的第一天、第二天和第三天。代码如图 2.3.11 所示。

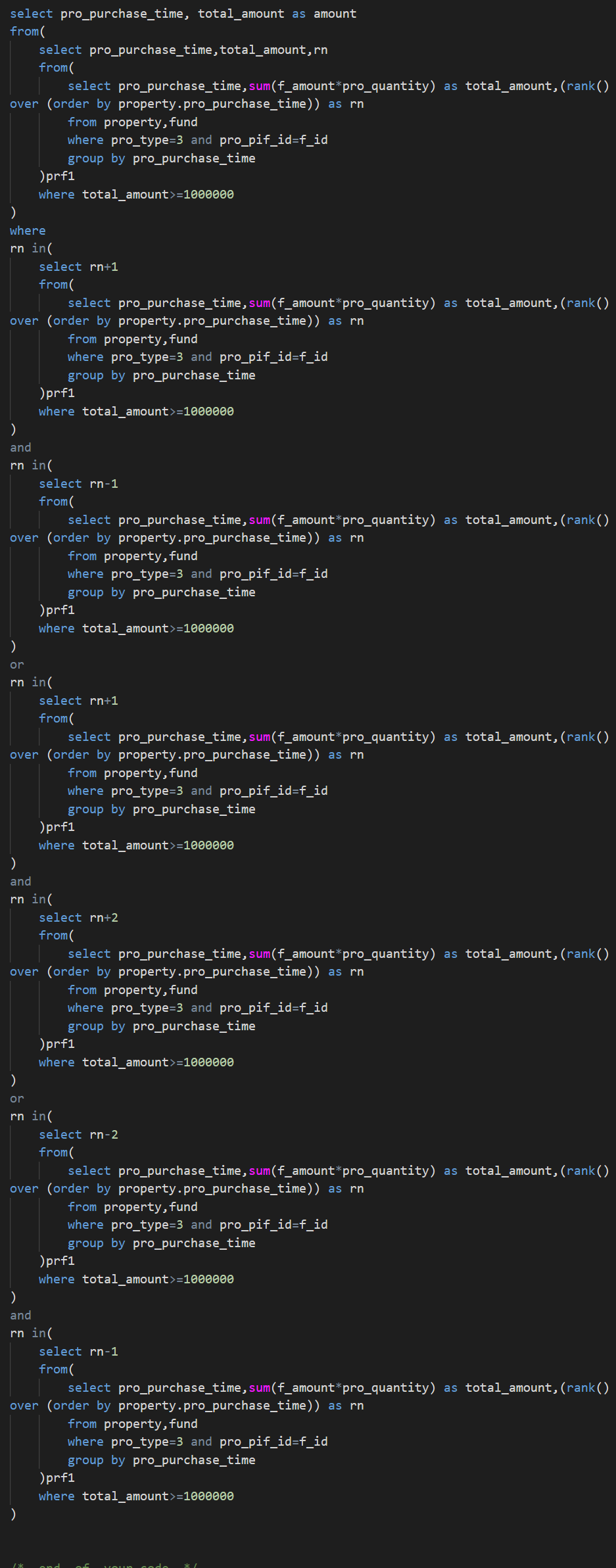


图 2.3.11 购买基金高峰期

**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**

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 left join 到一起。代码如图 2.3.12 所示。

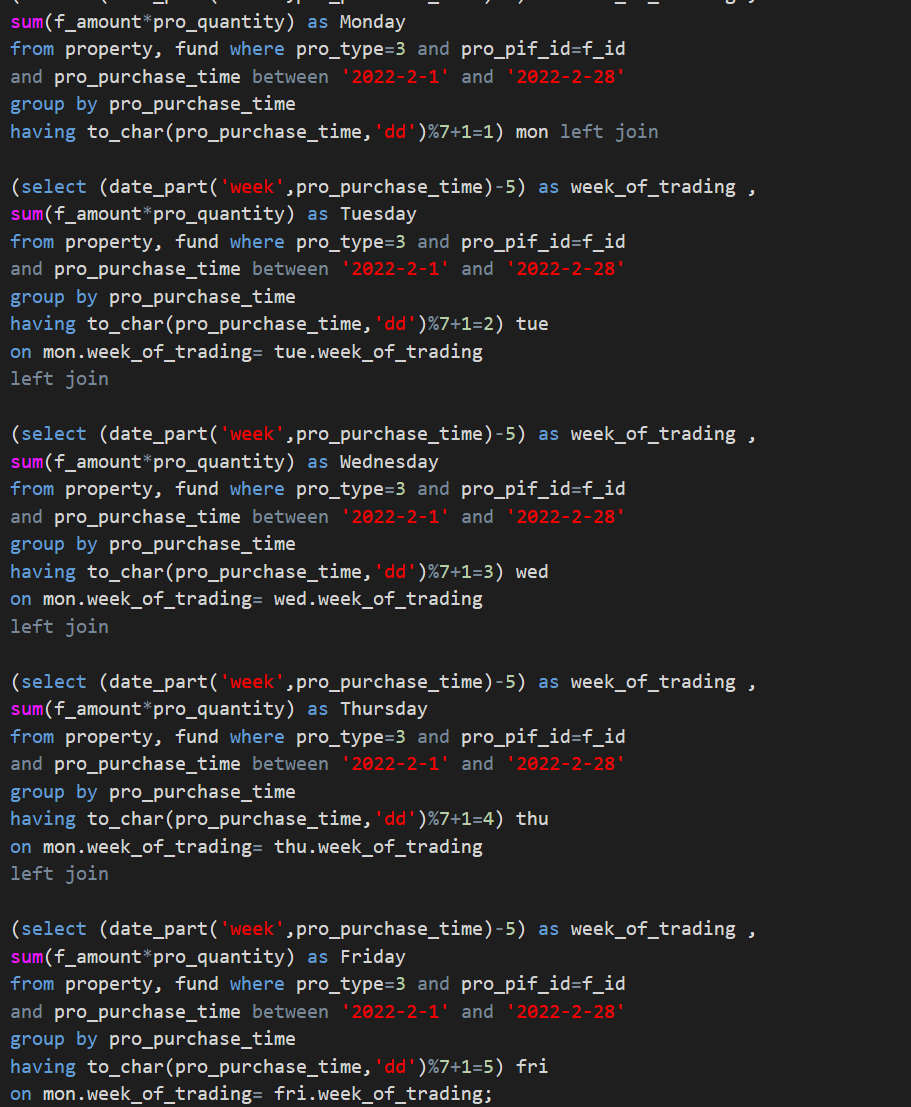


图 2.3.12 日历表格式

**2.3.20 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。

**2.3.21 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，用客 户编号相等连接两个表，找出理财产品数目大于基金数目的客户即可。代码如图 2.3.13 所示。

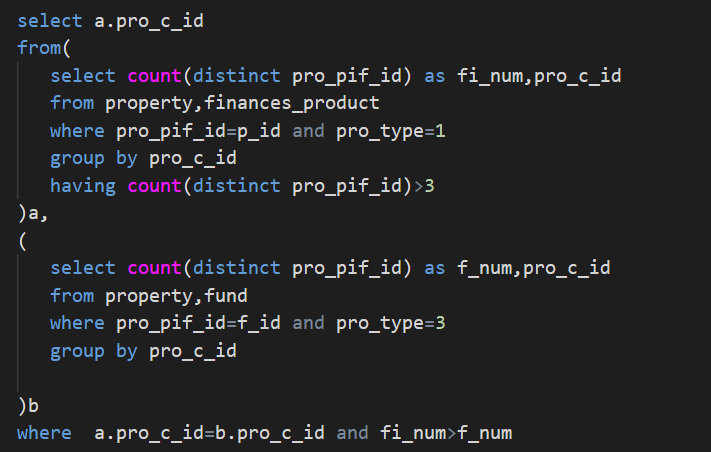


图 2.3.13 投资积极且偏好理财的客户

**2.3.22 查询购买了所有畅销理财产品的客户**

该关卡任务已完成，实施情况本报告略过。

**2.3.23 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。

**2.3.24 查询任意两个客户的相同理财产品数**

该关卡任务已完成，实施情况本报告略过。

**2.3.25 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。

## 2.4 数据的插入、修改与删除

本节的 6 个关卡围绕 Insert, Update, Delete 语句在不同场景下的应用 展开。

**2.4.1 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。

**2.4.2 插入不完整的客户信息**

本关任务：向客户表 client 插入一条数据不全的记录。

插入不完整的数据记录需要在 values 前指定插入的列或者把对应空值的 列设置为 NULL，代码如图 2.4.1 所示。

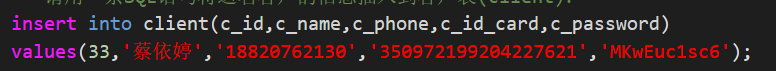


图 2.4.1 插入不完整的客户信息

**2.4.3 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，代码如图 2.4.2 所示。

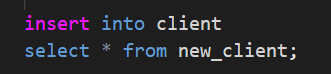


图 2.4.2 批量插入

**2.4.4 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。

**2.4.5 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exists 子查询即可实现。代码如图2.4.3所示。

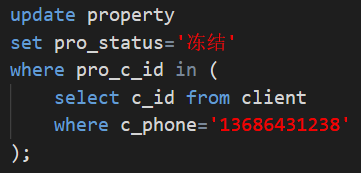


图 2.4.3 冻结客户资产

**2.4.6 连接更新**

该关卡任务已完成，实施情况本报告略过。

## 2.5 视图

本节的 2 个关卡围绕视图的创建与使用展开。

**2.5.1 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，代码如图 2.5.1 所示。

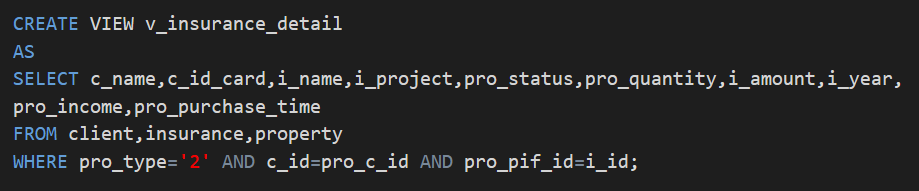


图 2.5.1 创建视图

**2.5.2 基于视图的查询**

该关卡任务已完成，实施情况本报告略过。

## 2.6 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。

**2.6.1 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只是语法不太一样，代码如图 2.6.1 所示。

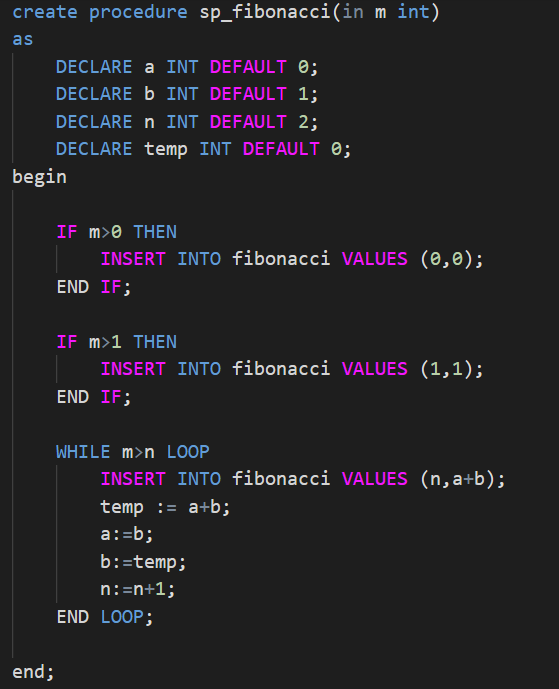


图 2.6.1 流程控制

**2.6.2 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。

**2.6.3 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置为 0 ，并且执行 rollback 恢复对数据库的修改。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 commit 提交更改。代码如图 2.6.2 所示。

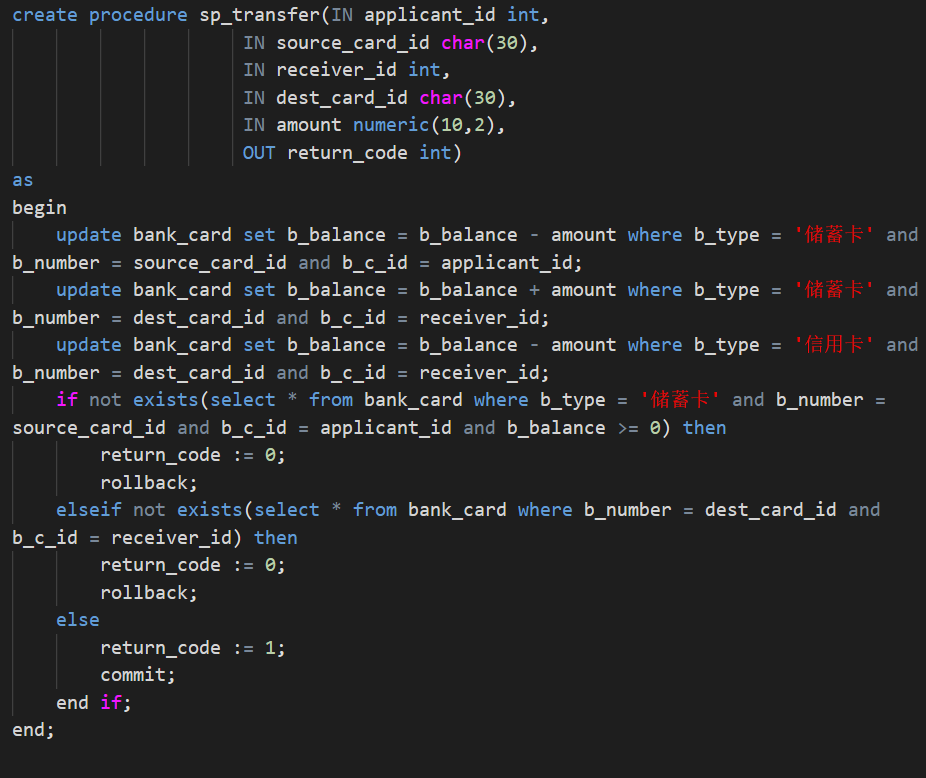


图 2.6.2 事务

## 2.7 触发器

本节的 1 个关卡涉及触发器的使用。

**2.7.1 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， 具体用触发器函数 TRI\_INSERT\_FUNC（）实现功能，如果执行过程中发现有不满 足条件的情况则利用 concat（）函数编辑报错信息 msg，并通过“raise exception ‘%’,msg”抛出报错信息。代码如图 2.7.1 所示。

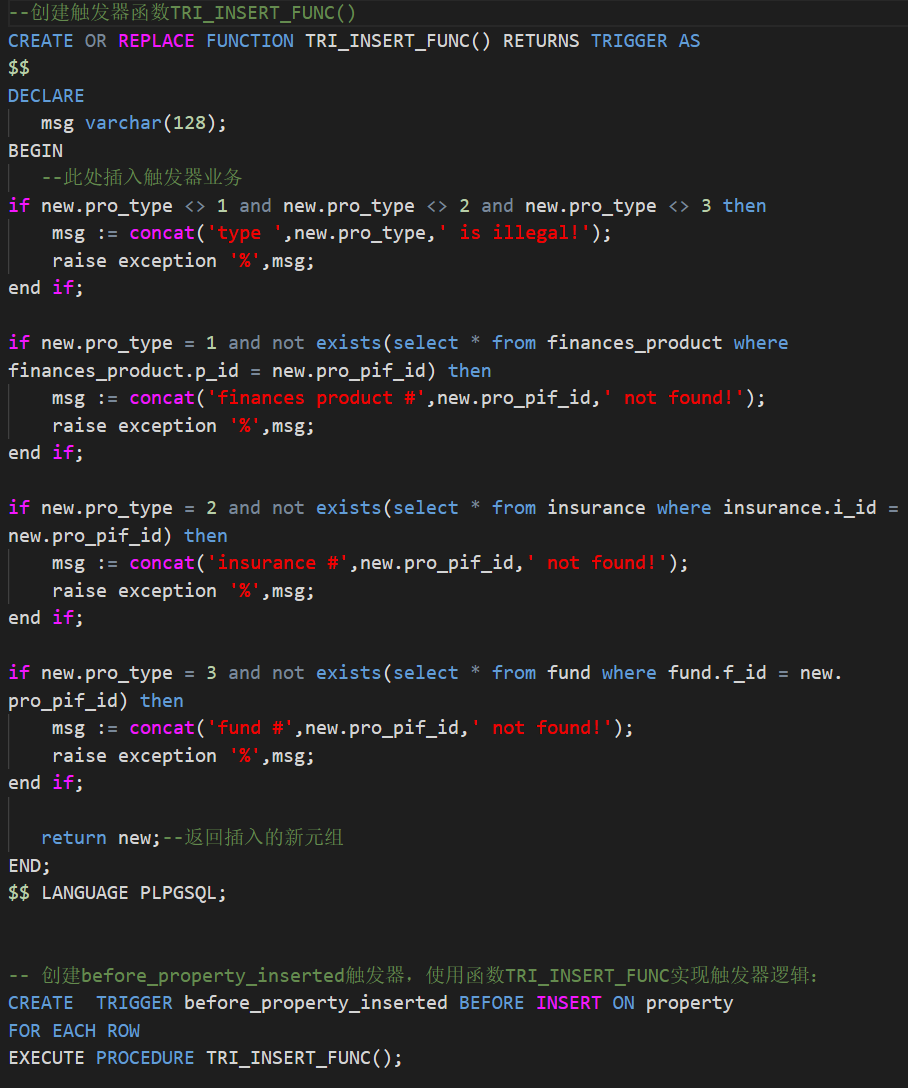


图 2.7.1 触发器

## 2.8 用户自定义函数

本节的 1 个关卡涉及用户自定义函数的定义和使用。

**2.8.1 创建函数并在语句中使用它**

本关任务：编写一个依据客户编号计算其在本金融机构的存储总额的函数， 并在 SELECT 语句使用这个函数。

根据任务要求和实验指导编写代码，函数的语法格式是固定的很简单。代码 如图 2.8.1 所示。

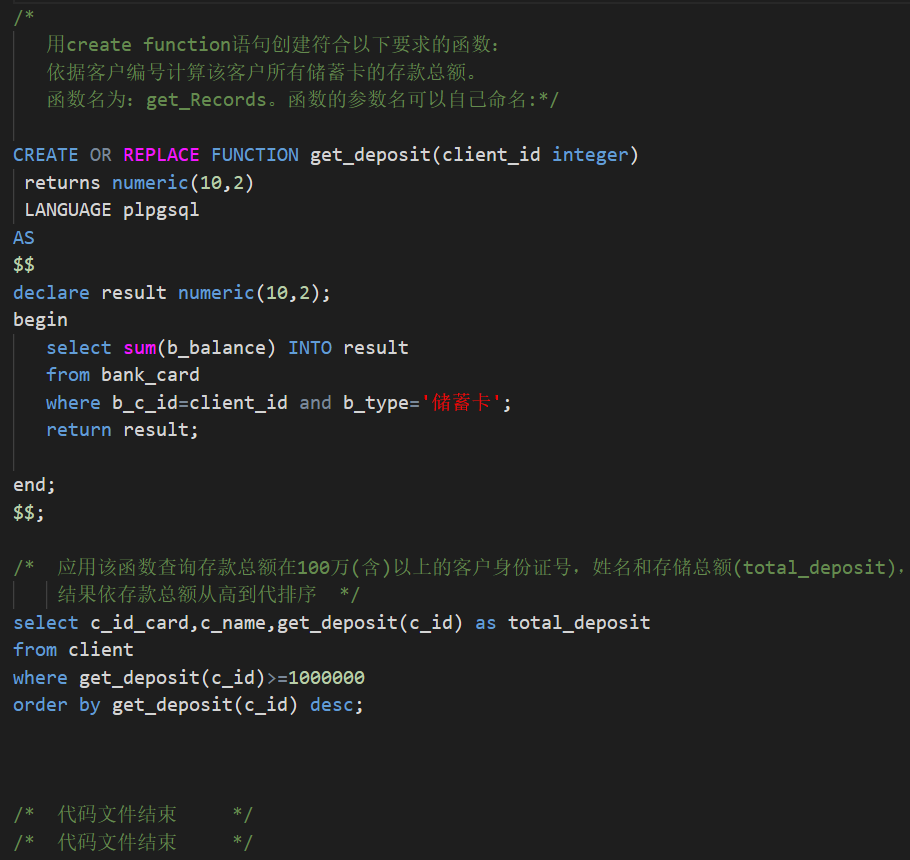


图 2.8.1 自定义函数

## 2.9 安全性控制

本节的 2 个关卡涉及数据库中的用户、角色和权限等内容。

**2.9.1 用户和权限**

该关卡任务已完成，实施情况本报告略过。

**2.9.2 用户、角色与权限**

本关任务：创建角色，授予角色一组权限，并将角色代表的权限授予指定的 一组用户。

根据任务要求合理使用 create...with password, grant...on...to 语句编写指令即可，代码如图 2.9.1 所示。

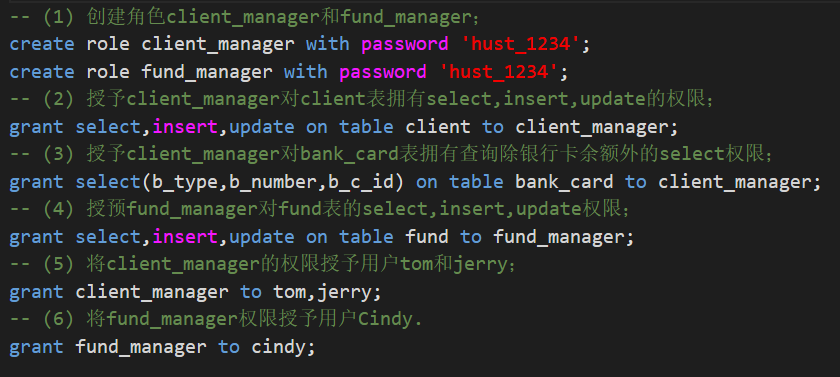


图 2.9.1 安全性控制

## 2.10 并发控制与事务的隔离级别

本节的 4 个关卡涉及数据库中并发控制与事务的隔离级别相关内容，包括隔 离级别的设置等，还通过添加等待代码实现了不可重复读、幻读等出错场景。

**2.10.1 不可重复读**

本关任务：设置事务的隔离级别。

数据库是共享资源，允许多个用户同时访问同一数据库，特别是在互联网应 用成为主流的当下，高可用性、高并发是所有应用追求的目标。但并发操作不加 控制，便会产生数据的不一致性。 并发操作可能带来的数据不一致性包括：

1) 丢失修改(lost update)

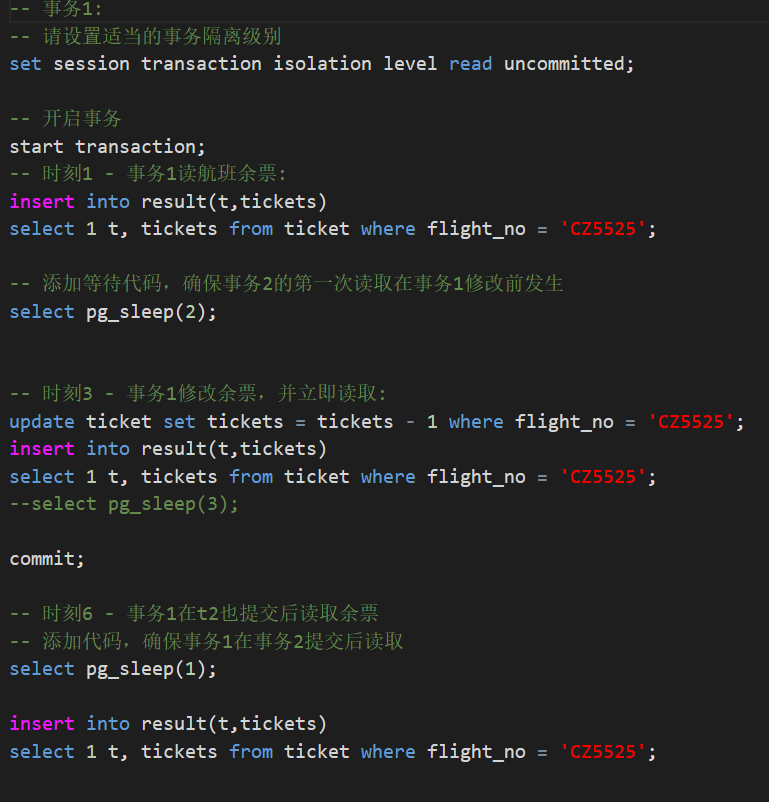
2) 读脏数据(dirty read)

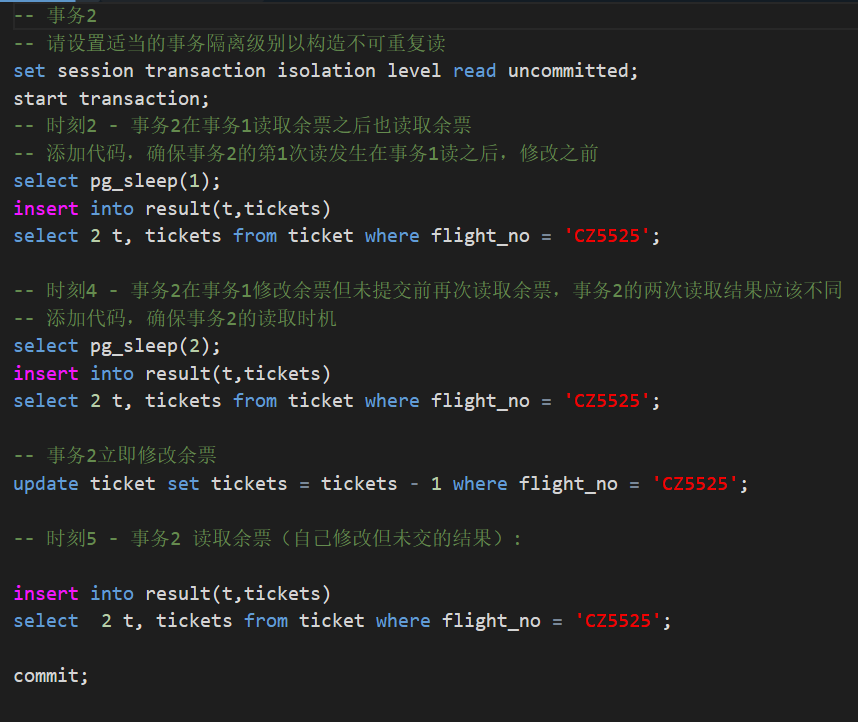
3) 不可重复读(non-repeatable read)

4) 幻读(phantom read)

最低的隔离级别不能避免读脏、不可重复读和幻读，而最高的隔离级别，可 保证多个并发事务的任何调度，都不会产生数据的不一致性，但其代价是并发度 最低。

根据任务要求将事务的隔离级别设置为 read uncommitted，并且根据任务 要求的执行顺序利用 pg\_sleep（）函数实现程序内部的等待，t1.sql 和 t2.sql 的代码分别如图 2.10.1 和图 2.10.2 所示。

图 2.10.1

图 2.10.2

**2.10.2 幻读**

本关任务：本关要求大家在较高隔离级别，即仅次于 serializable 的 repeatable read 隔离级别下重现“幻读”现象，这样，可更好地体验不可重复 读与幻读的区别。

两次查询余票超过 300 张的航班信息(第 2 次查询已替你写好)；在第 1 次查 询之后，事务 t2 插入了一条航班信息并提交(t2.sql 已替你写下好)；第 2 次查 询的记录数增多,发生“幻读”。代码如图 2.10.3 所示。

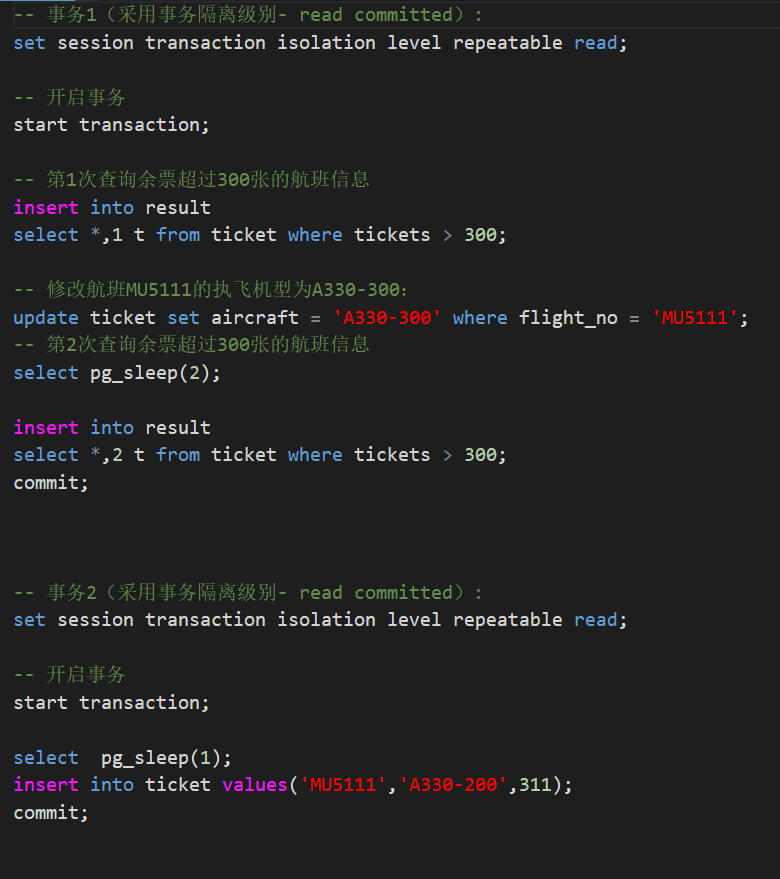


图 2.10.3 幻读

**2.10.3 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。

**2.10.4 可串行化**

该关卡任务已完成，实施情况本报告略过。

## 2.11 数据库应用开发（JAVA 篇）

本实训 7 个关卡要求结合 java 和数据库语言实现一个简单的数据库应用。 2.11.1 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, 26 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 createState‐ ment() 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，其中 有三个执行方法：

1.boolean execute(String SQL) ：如果可以检索到 ResultSet 象，则返回一个 布尔值 true ；否则返回 false 。使用此方法执行 SQL DDL 语句或需要使用真正 的动态 SQL 时。

2.int executeUpdate(String SQL) ：返回受 SQL 语句执行影响的行数。使用 此方法执行预期会影响多个行的 SQL 语句，例如 INSERT, UPDATE 或 DELETE 语句。

3.ResultSet executeQuery(String SQL) ：返回一个 ResultSet 对象。当希望获 得结果集时，使用此方法，就像使用 SELECT 语句一样。

本关需要选用第 3 个执行方法，使用 resultSet.next() 遍历 ResultSet 输出 相应的信息即可，代码如图 2.11.1 所示。

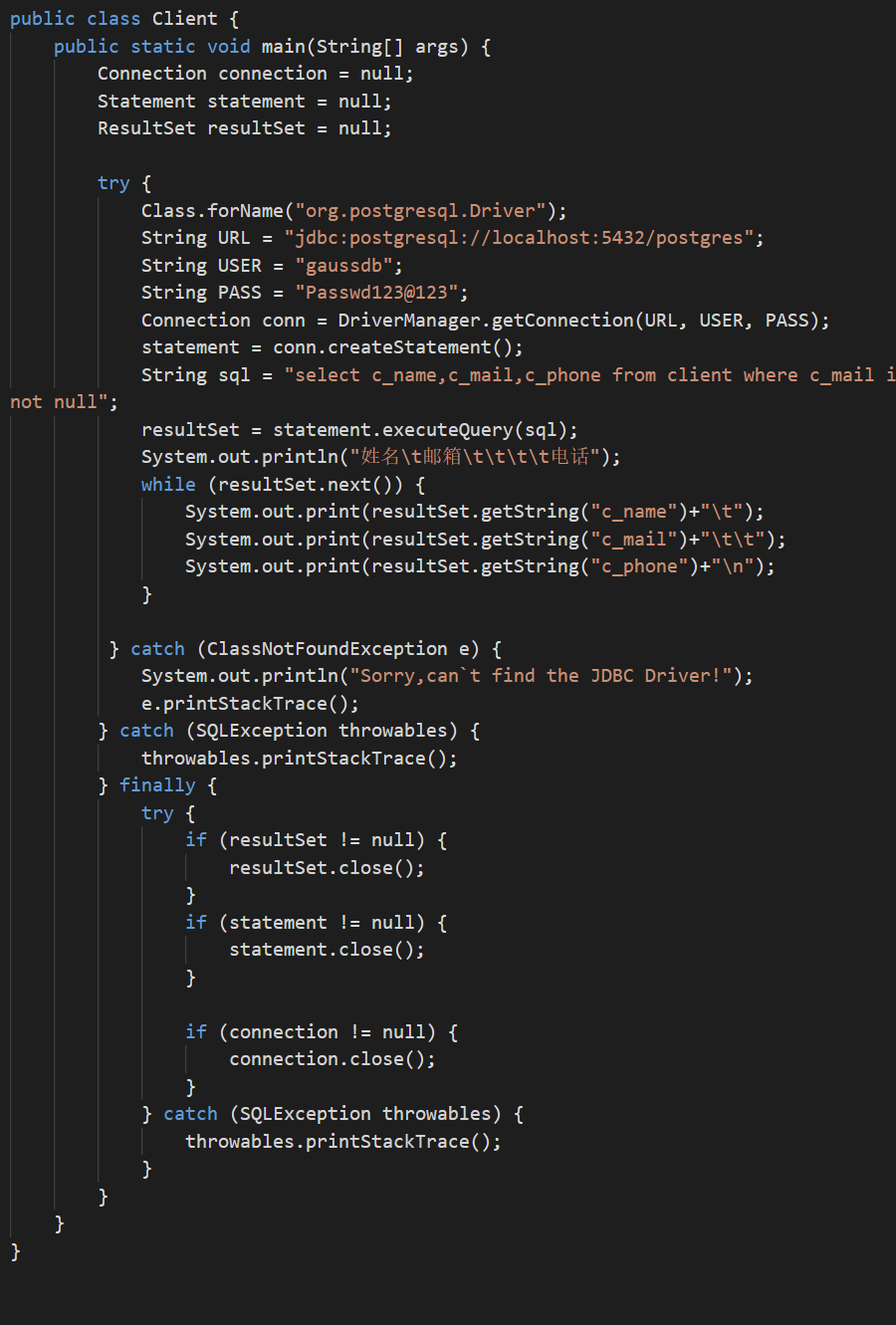


图 2.11.1 JDBC体系结构和简单查询

**2.11.2 用户登录**

该关卡任务已完成，实施情况本报告略过。

**2.11.3 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 2.11.2 所示。



图 2.11.2 添加新客户

**2.11.4 银行卡销户**

该关卡任务已完成，实施情况本报告略过。

**2.11.5 客户修改密码**

该关卡任务已完成，实施情况本报告略过。

**2.11.6 转账与事务操作**

本关任务：编写一个银行卡转账的方法。根据题意编写判断条件，如果不合法则直接 return false ，然后判断收款方卡 类型是否为信用卡决定是否要把收款金额设置为负数。代码如图 2.11.3 所示。（main 函数不做修改已省略）



图 2.11.3 转账与事务操作

**2.11.7 把稀疏表格转化为键值对存储**

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

本节的 1 个关卡涉及数据库的备份与恢复方法。

**2.12.1 备份与恢复**

本关任务：备份数据库，然后再恢复它。

使用 gs\_dump 将数据库备份到文件中，代码如图 2.12.1 所示。

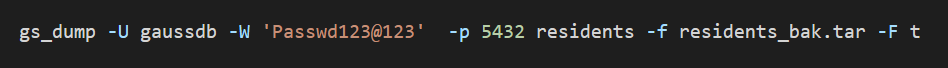


图 2.12.1 备份

使用 gs\_restore 从文件中恢复数据库，代码如图 2.12.2 所示。



图 2.12.2 恢复

## 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。

**2.13.1 从概念模型到 OpenGauss 实现**

该关卡任务已完成，实施情况本报告略过。

**2.13.2 从需求分析到逻辑模型**

本关任务：根据应用场景业务需求描述，完成 E‐R 图，并转换成关系模式。 根据任务要求描述转换得到的 E‐R 图如图 2.13.1，代码如图 2.13.2 所示。

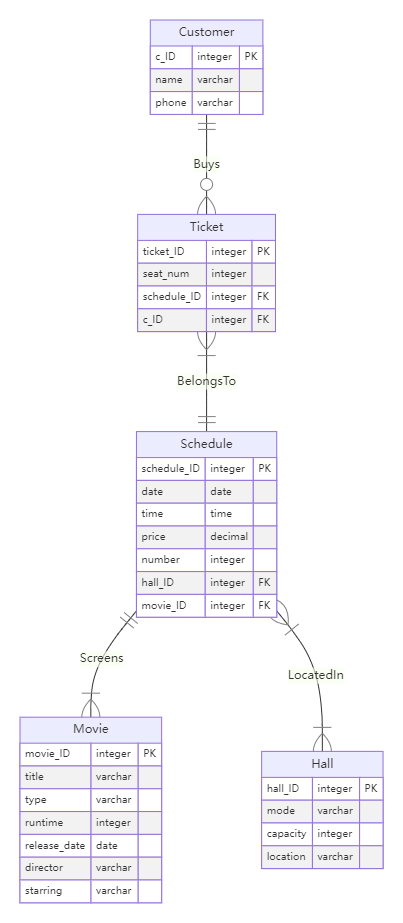


图 2.13.1 E-R 图

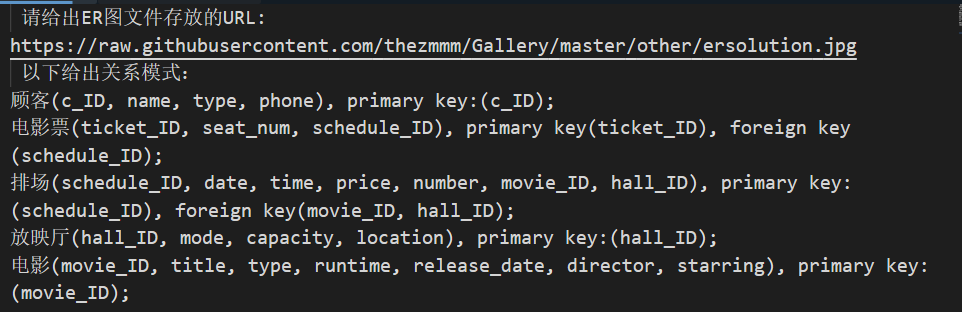


图 2.13.2 逻辑模型

**2.13.3 建模工具的使用**

该关卡任务已完成，实施情况本报告略过。

## 2.14 数据库的索引 B+树实现

本实训的 5 个关卡目的在于进行 B+数基本数据结构的实现和简单操作比如 insert 和 remove 的实现。

**2.14.1 BPlusTreePage 的设计**

本关任务：作为 B+树索引结点类型的数据结构设计的第一部分：实现 BPlusTreePage 类，该类是 B+树叶结点类型和内部结点类型的父类，提供 B+树 结点的基本功能。

BPlusTreePage 作为 BPlusTreeInternalPage 与 BPlusTreeInternalPage 父类，包 含了 B+树结点的基本信息和功能，比如结点类型，包含元素存储最大值以及现 存元素个数，父结点 id，当前结点 id，如图 2.14.1 所示。



图 2.14.1

**2.14.2 BPlusTreeInternalPage 的设计**

该关卡任务已完成，实施情况本报告略过。

**2.14.3 BPlusTreeLeafPage 的设计**

该关卡任务已完成，实施情况本报告略过。

**2.14.4 B+树索引：Insert**

该关卡任务已完成，实施情况本报告略过。

**2.14.5 B+树索引：Remove**

本关任务：实现 B+树索引的删除操作。 一颗完整的 B+树由内部节点和叶子节点组成，随着记录的删除，节点不断 合并减少，但仍需保持 B+树的完整性和一致性。当删除特定记录时，首先从根 节点进入 B+树，通过内部节点的索引信息不断向下遍历，直至对应的叶子节点， 之后删除该记录，删除后判断内部记录是否已小于最小值，是则开始进行树的调 整。首先判断是否可以与相邻兄弟节点合并，如果可以合并兄弟节点，则合并后 删除其在父节点中的索引，即递归进行元素删除操作，直至不再有合并发生；否 则可以进行元素的调动，移动兄弟节点中一个元素补充到叶子节点中，并维持 B+树的索引正确。

此处进行几个关键函数的展示：

1. void BPLUSTREE\_TYPE::Remove(const KeyType &key)

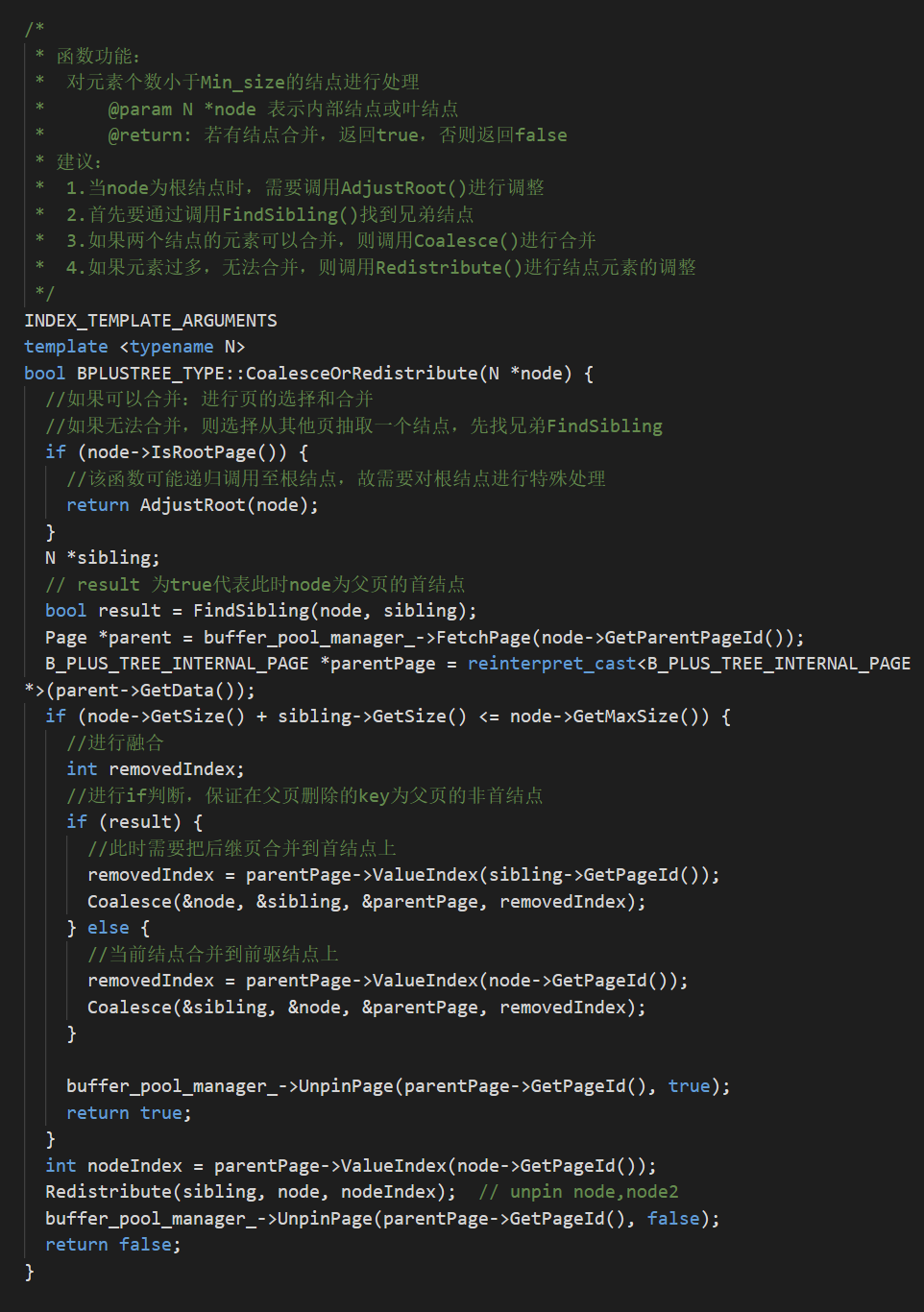
删除功能的主函数，主要依照删除逻辑设计，当 B+树为空时直接返回；不 为空时首先调用 FindLeafPage（）函数找到删除值所在的页；删除记录后，根据 不同情况，要么与兄弟节点合并，要么进行元素调动。代码如图 2.14.2 所示。



图 2.14.2 Remove（）函数

2. bool BPLUSTREE\_TYPE::CoalesceOrRedistribute(N \*node)

判断是否可以合并，可以的话执行操作，如不行则改为调用相应函数进行元 素调动。代码如图 2.14.3 所示。

图2.14.3 CoalesceOrRedistribute（）函数

3. bool BPLUSTREE\_TYPE::Coalesce(N \*\*neighbor\_node, N \*\*node, BPlusTreeInternalPage \*\*parent， int index)

本函数具体实现合并功能，并且在执行过程中，如果父节点也变得不满足 min\_size\_的条件，则还需递归调用 CoalesceOrRedistribute（）函数，直到调整 好所有结点。代码如图 2.14.4 所示。



图 2.14.4 Coalesce（）函数

# 3 课程总结

在本次课程中，我完成了14个实训实验，涵盖了数据库的多个方面，包括数据库和表的定义、完整性约束的创建和修改、数据查询、数据的插入、修改和删除、视图的创建和使用、存储过程与事务、触发器、用户自定义函数、安全性控制、并发控制与事务隔离级别、数据库应用开发、数据库的备份与日志、数据库的设计与实现以及B+树实现，总计涉及70个关卡。

在实施课程实验的过程中，我合理利用搜索引擎查阅相关资料，学习了OpenGauss的一些复杂语法和一些经典问题的简单处理方法，从而简化编程过程。我们能够熟练地使用OpenGauss实现一些复杂的查询，并且深刻感受到OpenGauss的灵活性。在许多关卡中，实现的方法并不是固定的，熟悉内置函数有时会带来意想不到的简化。编程的关键在于逻辑设计，首先要在脑海中构建一个完整的查询框架，不能一步一步进行思考。

在基于JDBC的数据库应用开发中，我深刻体会到了代码安全的重要性。如果不遵守规范，就可能会面临SQL注入攻击带来的严重后果。本次课程实验内容充实而完整，具有很强的引导性，完成实验后收获颇丰。尤其是最后的B+树实现，难度较大，具有很大的挑战性，同时也让我学习了一种新的数据结构，对于学习操作系统等其他课程也有很大的帮助。

最后，我衷心感谢课程组的老师们对实验内容的精心打造，以及在实验过程中给予的悉心指导。他们的辛勤工作和指导让我们能够深入理解数据库系统的原理和实践，并获得宝贵的经验和知识。