



华中科技大学

数据库系统原理实践报告

专 业： 计算机科学与技术

班 级： 计科 2009

学 号： U202015557

姓 名： 周帅君

指导教师： 丁晓锋

分数	
教师签名	

2022 年 12 月 25 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
2 任务实施过程与分析	2
2.1 数据库、表与完整性约束的定义(CREATE).....	2
2.2 表结构与完整性约束的修改(ALTER).....	2
2.3 数据查询(SELECT)	3
2.4 数据的插入、修改与删除(INSERT,UPDATE,DELETE)	5
2.5 视图.....	6
2.6 存储过程与事务.....	6
2.7 触发器.....	7
2.8 用户自定义函数.....	7
2.9 安全性控制.....	7
2.10 并发控制与事务的隔离级别.....	7
2.11 数据库应用开发(JAVA 篇)	8
2.12 备份+日志：介质故障与数据库恢复.....	8
2.13 数据库设计与实现.....	9
2.14 数据库的索引 B+树实现.....	10
3 课程总结	12

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 OpenGauss 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)；
- 6) 数据库内核实验(B+树)。

课程依托头歌实践教学平台，实践课程 url 见相关课堂教师发布链接及其邀请码。实验环境为 Linux 操作系统下的 OpenGauss2.1。在数据库应用开发环节，使用 JAVA 1.8。

总共有 70 个关卡，全部完成则获得头哥平台总分 155 分，不要求所有关卡都完成，最终程序检查满分只计头哥平台中的 100 分，即有些关卡可以自行选择跳过不做，但是实践环节的 2 级标题子任务的前 5 个以及第 13 个子任务（2.1 至 2.5 子任务、2.13 子任务）不能整体跳过（至少要完成其中每个二级子任务中的一个关卡）。

2 任务实施过程与分析

2.1 数据库、表与完整性约束的定义(Create)

本章要求使用一句 SQL 语句进行数据库与表的创建，并为表增加相应约束。

2.1.1 创建数据库

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.1.2 创建表及表的主码约束

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.1.3 创建外码约束(foreign key)

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.1.4 CHECK 约束

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.1.5 DEFAULT 约束

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.1.6 UNIQUE 约束

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.2 表结构与完整性约束的修改(ALTER)

本章要求使用一句 SQL 语句对表的内容、结构与约束进行添加、删除与修改。

2.2.1 修改表名

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.2.2 添加与删除字段

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.2.3 修改字段

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.2.4 添加、删除与修改约束

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.3 数据查询(Select)

本章要求使用一句 SQL 语句按照题目要求完成对表格的查询，其中数据库使用某银行的一个金融场景应用的模拟数据库，其中包含多张表格。

2.3.1 查询客户主要信息

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.3.2 邮箱为 null 的客户

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.3.3 既买了保险又买了基金的客户

本关卡应使用嵌套查询，分别对结果进行限制。首先，应先查找出购买了保险的客户（`pro_type = 2`）的 `pro_c_id`，在此基础上，查找出购买了基金的用户（`pro_type = 3`）的 `pro_c_id`，最后根据以上查找结果在 `client` 表中查找出符合要求的用户的名称、邮箱和电话。

2.3.4 办理了储蓄卡的客户信息

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.3.5 每份金额在 30000~50000 之间的理财产品

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.3.6 商品收益的众数

本关卡需要能够使用统计函数解决问题。首先，使用 `count()` 函数对商品收益值（`pro_income`）进行数量统计，然后使用 `all` 运算符限制查询出收益值出现次数大于等于任一收益值出现次数，即为众数，并且 `count(*)` 可统计出数目。

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，由于较为简单，实施情况本报告略过，不再赘述。

2.3.8 持有两张信用卡的用户

本关卡应使用嵌套查询，分别对结果进行限制。首先查询出信用卡记录（`b_type = '信用卡'`），然后使用聚集函数 `count()` 计数 `b_c_id` 大于等于二的用户，最后据此在 `client` 表中选出这些客户的名称、身份证号和手机号。

2.3.9 购买了货币型基金的客户信息

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.3.10 投资总收益前三名的客户

本关卡涉及多表多限制，应分布嵌套查询。首先从 `property` 表中查询出未冻结的记录 (`pro_status != '冻结'`)，然后依 `pro_c_id` 对收益 (`pro_income`) 进行 `sum()` 求和，并以 `order by total_income desc limit 3` 选出金额最高的 3 项，最后依照 `pro_c_id` 在 `client` 表中查询出剩下的信息。

2.3.11 黄姓客户持卡数量

本关首先通过 `left join` 将 `client` 与 `bank_card` 两表进行外连接，使得能够根据 `c_id` 使用 `count(b_number)` 来计数用户的银行卡数量，将此作为子表，借助字符串匹配 `like '黄%'` 选出姓黄的用户们的银行卡数量。

2.3.12 客户理财、保险与基金投资总额

本关设计较多表的联合查询。首先，将 `property` 表分别与 `finances_product`、`insurance` 和 `fund` 表用 `left join` 进行外连接，以此将商品数量与相应产品的金额进行对应，之后利用 `sum()` 查询出它们的乘积，即某种商品总金额，与相应的 `pro_c_id` 结合组成子表，再与 `client` 表进行 `left join` 外连接，从而总金额和需要的信息。其中可使用 `coalesce()` 函数将可能存在的空值转换为 0。

2.3.13 客户总资产

客户总资产为储蓄卡余额，投资总额，投资总收益的和，再扣除信用卡透支的金额，应分表查取。储蓄卡余额可通过 `client left join bank_card` 连接后，使用 `sum()` 获取相应 `c_id` 的总余额。信用卡透支额与投资总收益也通过同样的方式获取，投资金额与关卡 2.3.12 查询方法相同。将以上查询作为子表，依 `c_id` 计算相应的储蓄卡余额加上投资金额加上投资总收益减去信用卡透支额即得总资产。在这个过程中也可使用 `coalesce()` 函数将可能存在的空值转换为 0。

2.3.14 第 N 高问题

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。但其中值得注意的是 `limit` 子句的参数，第一个参数表示偏移，即确定开始筛选的位置，第二个参数表示按序挑出的数量。

2.3.15 基金收益两种方式排名

本关需要掌握 `opengauss` 几种排序函数的用法。对于名次不连续的排序，使用 `rank()` 函数，而名词连续的排序需要使用 `dense_rank()`。

2.3.16 持有完全相同基金组合的客户

本关需要掌握 opengauss 的列表聚集函数。使用 listagg()函数依 pro_c_id 将持有基金种类(pro_pif_id)聚集到一个列表中,这样就可以统一比较某一 pro_c_id 的所有基金种类。

2.3.17 购买基金的高峰期

该任务关卡跳过。

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, 由于较为简单, 实施情况本报告略过, 不再赘述。

2.3.19 以日历表格式显示每日基金购买总金额

该任务关卡跳过。

2.3.20 查询销售总额前三的理财产品

该任务关卡跳过。

2.3.21 投资积极且偏好理财类产品的客户

该任务关卡跳过。

2.3.22 查询购买了所有畅销理财产品的客户

该任务关卡跳过。

2.3.23 查找相似的理财产品

该任务关卡跳过。

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, 由于较为简单, 实施情况本报告略过, 不再赘述。

2.3.25 查找相似的理财客户

该任务关卡跳过。

2.4 数据的插入、修改与删除(Insert,Update,Delete)

本章要求使用一句 SQL 语句对表进行插入、修改与删除操作。

2.4.1 插入多条完整的客户信息

该关卡任务已完成, 由于较为简单, 实施情况本报告略过, 不再赘述。

2.4.2 插入不完整的客户信息

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.4.3 批量插入数据

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.4.4 删除没有银行卡的客户信息

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.4.5 冻结客户资产

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.4.6 连接更新

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.5 视图

本章要求能够创建表的视图，并对视图进行查询。

2.5.1 创建所有保险资产的详细记录视图

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.5.2 基于视图的查询

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.6 存储过程与事务

本章要求能够掌握一些基本的存储过程语句，使用不同语句完成存储过程与事务的设计。

2.6.1 使用流程控制语句的存储过程

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.6.2 使用游标的存储过程

该任务关卡跳过。

2.6.3 使用事务的存储过程

本关卡首先进行转账，进行三类转账：源储蓄卡转出、目标储蓄卡转入和目标信用卡转入，此三类更新涵盖了所有可能的转账情况。转账后，需要检查转出储蓄卡余额是否为负值，以及转入账户及卡号是否对应，若是则可以提交，否则进行回滚。

2.7 触发器

本章要求掌握基本的触发器函数语句，设计简单的触发器函数。

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

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.8 用户自定义函数

本章要求能够自定义函数并在查询中使用。

2.8.1 创建函数并在语句中使用它

该任务关卡跳过。

2.9 安全性控制

本章要求掌握基本的安全性语句，进行权限的给予与收回。

2.9.1 用户和权限

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.9.2 用户、角色与权限

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.10 并发控制与事务的隔离级别

本章任务关卡跳过。

2.10.1 不可重复读

该任务关卡跳过。

2.10.2 幻读

该任务关卡跳过。

2.10.3 主动加锁保证可重复读

该任务关卡跳过。

2.10.4 可串行化

该任务关卡跳过。

2.11 数据库应用开发(JAVA 篇)

本章要求掌握 java 利用数据库进行开发的基本方式，在 java 程序中链接数据库并对数据库进行操作。

2.11.1 JDBC 体系结构和简单的查询

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.11.2 用户登录

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.11.3 添加新客户

本关需要掌握向 SQL 语句中插入参数的方法，即使用“?”占参数位，然后使用 PreparedStatement 的 set 方法簇设定参数。

2.11.4 银行卡销户

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.11.5 客户修改密码

本关重在逻辑处理。首先应检查用户的用户名（邮箱）是否存在，再检查用户旧密码输入是否错误，最后才能更新。过程中根据错误情况设置返回信号值，最终返回该信号值。

2.11.6 事务与转账操作

该关卡与 2.6.3 关基本相同，重在将语句与 java 程序逻辑结合。

2.11.7 把稀疏表格转为键值对存储

本关卡中，insertSC 只需根据传入参数执行简单的插入语句即可。在 main 函数中，遍历待转换数据库，对每门课的成绩数据进行判断，若不为 0，则调用 insertSC 函数插入转存表。

2.12 备份+日志：介质故障与数据库恢复

本章要求掌握数据库备份的方式，能够使用 opengauss 导出导入语句进行数据库备份。

2.12.1 备份与恢复

只需在 test1_1.sh 文件中导出 residents 数据库，然后在 test1_2.sh 中导入即

可。

2.13 数据库设计与实现

本章要求能够进行数据库模型的设计，并根据设计的模型构建数据库。

2.13.1 从概念模型到 OpenGauss 实现

本关根据题图进行基本的表构建后，还需注意一些细节：opengauss 中自增约束需要通过定义序列来实现；对于索引的唯一性、属性的非空约束和初值需要结合题目确定。

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

本关使用 PowerDesigner “生成数据库” 功能即可生成相应 SQL 语句。

2.13.4 制约因素分析与设计

- 1) 社会、健康及文化等方面，需要充分考虑数据库设计所处的社会文化情境，依此做出合理的设计。如在进行表设计时，应考虑数据库用户的可能的姓名长度、ID 卡号长度等将这些信息定义为合适的数据类型。
- 2) 安全方面，在数据库应用时，应同时设计对应的权限视图，使得某一用户只能操作对应权限内的数据。
- 3) 法律方面，对于数据库存储的信息，可以通过设置列约束限制数据库内信息，保证数据库的安全合法。
- 4) 环境方面，数据库设计时应采取高效简洁的设计思路，设计出资源占用更少的数据库结构，间接为环境保护做出贡献。

2.13.5 工程师责任及其分析

在数据库设计过程中，各种制约因素相互影响、相互作用，缺一不可。法律为安全提供支撑，安全为法律提供保障，法律与安全为社会、健康及文化提供土壤，而环境是我们最终追求的目标。

作为一名工程师，除了技术上的设计，还需注意考虑任务本身的社会因素，平衡各方面的制约因素，设计出综合水平较高的产品。

2.14 数据库的索引 B+树实现

本章要求掌握数据库的内部索引机制——B+树，并补充设计各种方法，完成 B+树索引的实现。

2.14.1 BPlusTreePage 的设计

该关卡任务已完成，由于较为简单，实施情况本报告略过，不再赘述。

2.14.2 BPlusTreeInternalPage 的设计

本关需设计实现 B+树内部页面类的方法。其中，Lookup()方法需要使用二分查找法。CopyNFrom()方法：从当前结点尾部开始，循环处理每个待拷贝元素，在每次循环中，首先将键对值拷贝至 array_[i]中，然后利用缓冲区，先将页面（array_[i].second）固定在缓冲区中，再将该部分页面的父结点设为当前结点，最后将页面从缓冲区中解锁。MoveHalfTo()方法：利用 CopyNFrom()方法，将结点中一半（向上取整）的元素移动到另一结点中。MoveAllTo()方法：首先将父结点在缓冲区中固定，将父结点 0 位键设为当前页面在父结点中的键后再解锁，然后遍历当前结点，将当前结点元素移动至目标结点后，并通过类似的缓冲区方式将当前页面（array_[i].second）父结点设为目标结点。CopyLastFrom()方法：将键对值拷贝至 array_尾部，并将页面（pair.second）父结点设为当前结点。MoveFirstToEndOf()方法：首先将父结点在缓冲区中固定，将父结点 0 位键设为当前页面在父结点中的键后再解锁，然后借助 memmove()方法将 array_空间整体向前移动一个单位，再借助 CopyLastFrom()方法将键对值复制至目标结点尾部，并将父结点指向该结点的元素键更改为新的当前结点无效元素键。CopyFirstFrom()方法：首先将父结点在缓冲区中固定，将父结点 0 位键设为当前页面在父结点中的键后再解锁，然后借助 memmove()方法将 array_整体后移一位，将 0 位拷贝为键对值 pair，再将该键对值页面的父结点设为当前结点，并将父结点指向该结点的元素键更改为新的当前结点无效元素键。MoveLastToFrontOf()方法：借助 CopyFirstFrom()方法，将尾部键对值移至首部。

2.14.3 BPlusTreeLeafPage 的设计

该任务关卡跳过。

2.14.4 B+树索引：Insert

该任务关卡跳过。

2.14.5 B+树索引：Remove

该任务关卡跳过。

3 课程总结

本次实验进行了数据库从使用、应用到原理的操作实践，以闯关的方式完成各级子任务。完成度基本达标，完成范围基本覆盖知识掌握要求。

本次实验的主要工作有：

- 1) 完成了数据库、表、索引、视图、约束、存储过程、触发器数据对象的管理与编程，进行了相应的增、删、查、改操作，并对其安全性控制进行了操作；
- 2) 借助工具软件进行了数据库的设计，并根据设计完成了数据库的实现；
- 3) 对进行数据库进行了应用；
- 4) 实现了部分数据库内核检索机制。

通过本次实验，我提高了数据库操作技巧，加深了对数据库的认识，同时接触了更多数据库的控制操作、应用操作以及内核实现，让我对数据库的机制有了更加充分的理解。但本次实验任存在不足，主要为对陌生的高级控制操作不够熟练，对困难的数据库内核内容不够熟悉，在未来应加强对这两方面的学习。