

**数据库系统原理实践报告**

|  |  |
| --- | --- |
| 专 业： | 计算机科学与技术 |
| 班 级： | CS2006 |
| 学 号： | U202015471 |
| 姓 名： | 杨释钧 |
| 指导教师： | 左琼 |

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

20 年 月 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 4](#_Toc118022465)

[2.1 “任务标题” 4](#_Toc118022466)

[2.2 “任务标题” 5](#_Toc118022467)

[2.3 “任务标题” 5](#_Toc118022468)

[。。。 5](#_Toc118022469)

[2.9 并发控制与事务的隔离级别 5](#_Toc118022470)

[2.12 备份+日志：介质故障与数据库恢复 6](#_Toc118022471)

[2.13数据库设计与实现 6](#_Toc118022472)

[2.14 数据库应用开发(JAVA篇) 7](#_Toc118022473)

[2.15数据库的索引B+树实现 7](#_Toc118022474)

[3 课程总结 9](#_Toc118022475)

[附录 10](#_Toc118022476)

（注意目录在正式定稿时要依据正文标题结构和内容页码刷新同步）

# 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。

本课程相关资料网站：

MYSQL手册：<https://dev.mysql.com/doc/>

JAVA手册：<https://docs.oracle.com/javase/8/docs/api/index.html>

课程开放资源：<https://gitee.com/kylin8575543/db2022-spring>

**简要陈述介绍本实践课程的总体任务要求及其任务分解情况说明。（要求不超过1页篇幅）**

**注意：任务书中的红色和蓝色文字不要出现在实践报告中。**

**关于正文排版要求**

正文统一采用小四号宋体/Times New Roman和1.5倍行距，段前、段后均0磅间隔。

正文应论述清楚，文字简练通顺，插图简明，书写整洁。文中图、表按制图要求绘制。

**关于正文章节标题的排版要求：**

报告章标题称为一级标题，章内小节标题依次分为二级标题、三级标题等。一级标题的编号用数字1，2，…编制；二级标题的编号用1.1，1.2，…编制；三级标题的编号用1.1.1，1.2.1，… 编制；四级及以后各级标题可依此类推。建议标题不超过3级（如1.1.1），超出部分可根据需要使用(1)，①，A，a)，…等形式描述。

标题编号与标题文字之间均用空格隔开，如：“1 引言”、“2.1 需求分析”。报告正文的一级标题（章）须另起一页居中排版。

1级标题宋体小三号字加黑，单倍行距，段前、段后均12磅间隔。

2级标题宋体四号字加黑，1.25倍行距，段前6磅，段后0磅间隔。

3级标题宋体小四号字加黑，单倍行距，段前3磅、段后8磅间隔。

**关于正文中的图的排版要求：**

正文中所有插图要求图面整洁，布局合理，线条粗细均匀，圆弧连接光滑，尺寸标注规范。所有曲线、图表、线路图、流程图、程序框图、示意图等不准徒手画，必须按国家规定标准或工程要求采用计算机或手工绘制。

所有插图均应有图号和图名。图号按章编，如第2章的图为图2.1、图2.2、…，第3章的图为图3.1、图3.2、…等。图名是插图的名称，扼要概括图的内容，字数不宜太多。图号和图名写在图下方，并相对于图居中排版。少数图有图注，图注写在图下面且字号应比图号、图名的字小一号，图名和图注后面均不加标点符号。

所有插图均应在正文中予以引用。引用某插图时，一般写为“…见图x.y”或“图x.y是…”。正文中的插图均须安排在文中第一次引用到该图的正文下面，要求先见文，后见插图，且图一般不跨页绘制。

图中文字、图号和图名，统一采用小五号宋体。

**关于正文中的表的排版要求：**

表格由表号、表名、表头、表身等组成。表号按章编，如第2章的表为表2.1、表2.2、…，第3章的表为表3.1、表3.2、…等。表名是表格的名称，扼要概括表的内容，字数不宜太多。表号、表名放在表的正上方，相对于表体居中排版。表号及表名后不加标点。表头包括栏头、行头，与表身一起构成表格的主体。表中的竖称为栏，横格称为行。表身的内容，一般包括：数据、文字、公式和表图等。表内的数据对应位要对齐。少数表有表注，表注写在表下面且字号应比表号、表名的字小一号。

所有表格均应在正文中予以引用。引用某表格时，一般写为“…见表x.y”或“表x.y是…”。表格应尽量靠近正文的叙述，一般应先见文，后见表，表不跨节。表格允许转页。表格转页部分可以不写表号和表名，但要重复书写表头，并在表头右上角写“（续）”字标注。

表中文字、图号和图名，统一采用小五号宋体。

**关于正文中公式的排版要求：**

公式一般另行居中写，公式末不加标点。若公式前有文字，如例、解、证、假定等，文字顶格写，公式仍居中写。一行写不下时，公式允许转行。公式转行需处理得当，做到既意义正确，又使版面美观匀称。

公式要有编号，公式编号按章编，如第2章的公式为(2.1)、(2.2)、…，第3章的公式为(3.1)、(3.2)、…等。公式编号写在公式右侧行末顶边线，并加圆括号。

公式一般应在正文中予以引用，引用时以公式编号指示公式。正文中常有公式中表示量的符号说明，采用“式中”二字作为标志。一般可写成接排形式，如“式中，A指……；B指……”。

**关于实践报告的目录格式要求：**

目录是课程实验报告的纲要。正文的各级标题（一般最多取三级）、附录应编入目录，但目录本身不出现在其中。

目录要求层次清晰，含标题及对应的起始页号。目录的最后 “附录”无章节号。

课程实验报告正文、参考文献和附录页面，使用“1，2，3，… ”编连续页码。页码应标在页面的右下角。

目录中正文的各级标题名称、参考文献和附录及其对应的起始页号，务必与报告中正文的各级标题名称、附录及其对应的起始页号保持一致。

# 2 任务实施过程与分析

本章内容请选择实训中完成的任务中你觉得有代表性的关卡，对应任务书第2章的实践环节的2.1~2.15小节的具体关卡的任务阐述它们的完成过程，并给出必要的分析。请对各个2级标题任务分别在本章用对应的2级标题小节进行说明。

蓝色文字请替换成任务书中相应2级标题名称。

**关于报告篇幅要求的说明：**

**（1）报告篇幅并非越多越好，原则上报告除附录外能够在不超过32页的篇幅内主次分明、亮点特色突出、表述清晰的汇报本次实践任务完成情况即为好的报告。**

**（2）请在报告中保留任务书中所有的2级和3级标题的任务，但可在标题下的正文中注明“该任务关卡跳过”，或者“该关卡任务已完成，实施情况本报告略过”。每位同学可依据自己完成的任务关卡，从中选择自己认为最有代表性的原则上50%的关卡，参照模板要求阐述本次实践任务的主要工作。**

**实践报告的总体效果是要能较好的体现对数据库运行过程中的管理机制、SQL语法的掌握与灵活应用、数据库系统的设计与实现的方法和基本实践技能的掌握程度。**

**（3）本章内容中各个子任务的篇幅各人可依据完成情况相对灵活设定，原则上简单过程简洁明了即可，重点、难点部分则应有方法、思路以及过程的阐述。**

## 2.1 数据库、表与完整性约束的定义（Create）

本节需要我们掌握MySQL中创建数据库表，定义数据完整性约束，需要在实践中掌握MySQL中用Create创建数据库表以及增加如主码约束、外码约束等完整性约束的方法，这一部分在Educoder平台上给出了较为详细的知识介绍，难度较低。

（模板中的蓝色文字需替换成任务书中相应内容）

请用一段文字（原则上控制在5行以内）总体上概括本小节任务的内容与要求。

2.1.1 创建数据库

本关任务要求创建用于2022年北京冬奥会信息系统的数据库beijing2022，是对create命令的直接使用，比较简单，通过代码如下。

Create database beijing2022;

阐述子任务的完成过程，结合实际复杂程度进行必要的分析，例如代码思路、过程、代码注释等。

**每个3级子任务报告正文内容具体撰写要求如下：**

**（1）关于是否可以直接粘贴源代码。源代码合理排版后可以直接粘贴。**对于比较简单的解题SQL语句或者代码（参考标准：科学排版后代码约占4行）可直接粘贴源代码；但对于比较复杂的SQL语句或者相关代码，给出必要的注释说明，还可对相应代码的构建思路进行简要陈述。

**本章内容是要在必要的篇幅内高效、简洁阐述清楚任务的解决过程，简单代码不赘述，复杂代码阐述思路和关键点。**

（2）**关于过程分析。**若该子任务完成过程较复杂，其中遇到一些值得分析的问题或错误，可对问题或错误及其解决过程进行适当的分析。若认为过程比较简单，可合理略过本子任务的过程分析这部分内容。**原则上复杂过程的文字描述与分析控制在半页以内。**

（3）**关于结果贴图。**原则上直接粘贴源代码的子任务不贴结果截图，而比较复杂、陈述了代码思路或给出了注释的代码，可粘贴过程中的代表性错误或代表性结果图。**关于贴图的篇幅，请尽量遵循下述原则：一个3级子任务（第2.11节数据库设计与实现中数据库结构展示部分可突破图片版面要求的限制）不超过3张截图，单个图片版面尽量控制为相当于正文5~8行，一个3级子任务的所有截图版面总和尽量控制在半页以内。**

2.1.2 创建表及表的主码约束

本关要求在数据库TestDb中创建表t\_emp，要求字段id作为表的主码。本关需要我们了解主码约束的概念以及用CONSTRAINT关键字创建约束的方法。通过代码如下：

use TestDb;

create table t\_emp(

id INT,

name VARCHAR(32),

deptId INT,

salary FLOAT,

constraint PK\_t\_emp PRIMARY KEY (id)

);

2.1.3 创建外码约束

本关要求在数据库MyDb中新建两个表，并建立合适的外码约束。本官需要我们了解外码约束的概念以及用CONSTRAINT关键字创建约束的方法。核心代码如下：

constraint FK\_staff\_deptNo foreign key(deptNo) references dept(deptNo)

2.1.4 CHECK约束

本关要求在数据库MyDb中新建表，并且为其创建需要的CHECK约束，关键还是在于用CONSTRAINT关键字创建CHECK约束的语法。核心代码如下：

brand char(10) constraint CK\_products\_brand check(brand in ('A','B')),

price int constraint CK\_products\_price check(price >0)

2.1.5 DEFAULT约束

本关要求在数据库MyDb中新建表，并且为其创建需要的DEFAULT约束为字段指定默认值，关键在于default关键字的使用。核心代码如下：

mz char(16) default '汉族'

2.1.6 UNIQUE约束

本关要求在数据库MyDb中新建表，并且为其创建需要的UNIQUE约束以保证字段取值的唯一性，关键在于unique关键字的使用。核心代码如下：

ID char(18) unique

。。。。。

## 2.2 表结构与完整性约束的修改（ALTER）

在本节中，需要我们在实践中掌握MySQL中表结构与完整性约束的修改方法，重点是掌握关键字ALTER的使用，这一部分的相关知识在Educoder平台上有很完备的介绍，整体难度较低。

请用一段文字（原则上控制在5行以内）总体上概括本小节任务的内容与要求。

2.2.1 修改表名

本关要求我们使用ALTER关键字修改表名，主要是学习ALTER关键字中的rename方法的使用方法，核心代码如下所示。

alter table your\_table rename my\_table;

。。。。。

2.2.2 添加与删除字段

本关要求我们使用ALTER关键字对表中的字段进行增加或删除，重点是学习ALTER TABLE的语法，同时学习其ADD和DROP方法的使用方法，核心代码如下所示。

#语句1：删除表orderDetail中的列orderDate

alter table orderDetail drop orderDate;

#语句2：添加列unitPrice

alter table orderDetail add unitPrice numeric(10,2);

2.2.3 修改字段

本关要求我们使用ALTER关键字对表中的字段进行修改，这里仅仅要求我们修改一个列名以及修改一列的数据类型，主要要求我们掌握这两种动作的实现方法，关键在于modify关键字和rename关键字的相关语法，核心代码如下所示。

alter table addressBook modify QQ char(12),rename column weixin to wechat;

2.2.4 添加、删除与修改约束

本关要求我们掌握用ALTER TABLE语句对主码约束、外码约束、CHECK约束、UNIQUE约束的添加、删除与修改的方法，关键在于add和drop关键字的使用，核心代码如下所示。

alter table Staff add primary key(staffNo);

alter table Dept add constraint FK\_Dept\_mgrStaffNo foreign key(mgrStaffNo) references Staff(staffNo);

alter table Staff add constraint FK\_Staff\_dept foreign key(dept) references Dept(deptNo);

alter table Staff add constraint CK\_Staff\_gender check(gender in ('F','M'));

alter table Dept add constraint UN\_Dept\_deptName unique(deptName);

## 2.3 基于金融应用的数据查询（select）

在本节中，需要我们在实践中掌握MySQL中数据查询的方法，关键在于学习select的使用，这一部分中的一些题目在逻辑上比较复杂，可以说是本次实训中花费时间最多也最难的。这一部分主要涉及到数据查询中的单表查询、连接查询、嵌套查询、集合查询、基于派生表查询等常见的查询方法。

2.3.1 查询客户主要信息

本关是简单的单表查询，逻辑上比较简单，要求对最终结果进行排序，这就使用到了MySQL中的order by对结果进行排序，最终代码如下所示。

select c\_name,c\_phone,c\_mail from client order by c\_id;

2.3.2 邮箱为null的客户

本关是简单的单表查询，逻辑上比较简单要求查找邮箱字段值为null的客户，这就涉及到了where子句的用法，值得注意的是在判断邮箱字段值是否为null的时候需要使用is null。最终代码如下所示。

select c\_id,c\_name,c\_id\_card,c\_phone from client where c\_mail is null;

2.3.3 既买了保险又买了基金的客户

本关需要涉及到嵌套查询和多条件查询，由于我们需要查找同时具有保险和基金两种资产的客户，所以需要判断当前的客户在property表中是否同时具有这两种类型的资产。这里我使用的是exists关键字来判断存在性，最终需要用order by进行排序。最终代码如下所示。

select c\_name,c\_mail,c\_phone from client where

(exists(select 1 from property where c\_id=pro\_c\_id and pro\_type=2) and exists(select 1 from property where c\_id=pro\_c\_id and pro\_type=3)) order by c\_id;

2.3.4 办理了储蓄卡的客户信息

本关需要涉及到多表连接和条件查询。因为题目要求我们查找办理了储蓄卡的客户相关信息，我们需要将表client和表bank\_card按客户id进行等值连接，找出其中银行卡类型为储蓄卡的元组，最后用order by按照客户id进行排序即可，最终代码如下所示。

select c\_name,c\_phone,b\_number from client,bank\_card where c\_id=b\_c\_id and b\_type='储蓄卡' order by c\_id;

2.3.5 每份金额在30000~50000之间的理财产品

本关是单表查询，但是涉及到了between关键字和按多个字段升序降序排序的知识。我们需要对finances\_product进行单表查询，在where子句中用between and关键字查询金额在指定范围内的产品，然后根据题目中的要求按金额升序排序，金额相同时按理财年限降序排序，这就涉及到了order by子句的高级用法，也就是按多字段排序以及使用关键字desc进行降序排序，最终代码如下所示。

select p\_id,p\_amount,p\_year from finances\_product where p\_amount between 30000 and 50000 order by p\_amount,p\_year desc;

2.3.6 商品收益的众数

本关要求求众数，是一个比较典型的数据查询需求，涉及到子查询、分组统计、COUNT()函数、ALL关键字等用法。首先我们需要对表property用group by关键字进行分组统计，分组统计依据就是商品收益字段pro\_income，然后使用having子句筛选出各分组内元组个数最多的分组，这一过程需要使用关键字all、子查询、COUNT()函数来实现，最终代码如下所示。

select pro\_income,count(\*) as presence from property

group by pro\_income

having count(\*) >= all(select count(\*) from property group by pro\_income);

2.3.7 未购买任何理财产品的武汉居民

本关涉及到用LIKE进行模糊匹配、子查询、not exists谓词等知识的使用。首先我们需要使用关键字LIKE根据身份证号前四位匹配出武汉居民，然后通过子查询查找出购买了理财产品的客户id，也就是在property表中进行查找，最后用not exists谓词对该结果进行查找，也即得到了没有购买过理财产品的居民，最终代码如下所示。

select c\_name,c\_phone,c\_mail from client where

c\_id\_card like '4201%' and not exists(select 1 from property where

c\_id=pro\_c\_id and pro\_type = 1) order by c\_id;

2.3.8 持有两张以上信用卡的用户

本关涉及到子查询和分组统计的知识。首先通过一个子查询查找bank\_card表，根据该卡的持有者进行分组统计，也就是使用了关键字group by，同时也要用where查询出信用卡，通过COUNT()函数获取分组内信用卡的数量，大于等于2的时候就是我们需要的用户信息。最后在外层查询中使用关键字in判断当前的用户id是否在我们子查询结果中并用order by进行排序即可，最终代码如下所示。

select c\_name,c\_id\_card,c\_phone from client where

c\_id in (select b\_c\_id from bank\_card where b\_type='信用卡'

group by

b\_c\_id

having

count(\*) >=2) order by c\_id;

2.3.9 购买了货币型基金的客户信息

本关涉及到嵌套查询和exists关键字的用法，本身逻辑上比较简单，只需要根据用户的id查找到购买了基金并且同时购买了货币型基金的用户，将最终的结果用order by按照id排序即可，最终代码如下所示。

select c\_name,c\_phone,c\_mail from client

where exists(select 1 from property where c\_id=pro\_c\_id and pro\_type=3

and exists(select 1 from fund where f\_id=pro\_pif\_id and f\_type='货币型'))

order by c\_id;

2.3.10 投资总收益前三名的客户

本关的逻辑相对比较复杂，涉及到子查询、等值连接、分组统计、limit等知识。根据任务要求，总收益需要排除被冻结的资产，因此我们首先通过一个子查询得到一个客户id和客户总收益的表，需要排除掉冻结资产，用group by按照id进行分组统计，然后对新表和旧表进行等值连接，按照总资产进行降序排序，最终用limit 3获取投资总收益前三名的客户，这里也涉及到了用关键字as进行起别名的作用，最终代码如下所示。

select c\_name,c\_id\_card,total\_income as total\_income from client,

(select pro\_c\_id,sum(pro\_income) as total\_income from property where pro\_status!='冻结'

group by pro\_c\_id

) as newpro where c\_id=newpro.pro\_c\_id order by total\_income desc limit 3;

2.3.11 黄姓客户持卡数量

本关的逻辑相对复杂，涉及到外连接、分组统计、模糊匹配等知识。本关的一个坑点在于黄姓客户持卡数量可能是0，如果使用等值连接去查询的话这些客户信息全都会被舍弃掉，所以这里需要根据用户id使用左外连接连接表client和表bank\_card。此后需要通过like找出所有黄姓用户，然后根据用户id分组用COUNT()函数统计持卡数量，按照要求进行排序即可，这里也使用了ifnull关键字进行判定，最终代码如下所示。

select c\_id,c\_name,ifnull(count(b\_number),0) as number\_of\_cards from client left join bank\_card

on c\_id=b\_c\_id

where c\_name like '黄%'

group by c\_id

order by number\_of\_cards desc,c\_id;

2.3.12 客户理财、保险与基金投资总额

本关逻辑上比较复杂，涉及到了多个表之间的关系，涉及到分表合并、分组连接、外连接等知识

2.3.13 客户总资产

2.3.14 第N高问题

本关涉及到的主要知识有排序、去重、取第N高等，本身的逻辑比较简单，首先通过子查询找出第4高的保险金额，具体思路就是将表insurance查询，用distinct关键字去重之后用limit关键字找出第4高的保险金额，这里采用了limit+offset关键字的组合用法，在外层查询中再对表insurance进行查询即可，这层查询是为了找出保险金额等于内层子查询结果的元组，最终代码如下所示。

select i\_id,insurance.i\_amount from

(select distinct i\_amount from insurance

order by i\_amount desc

limit 1 offset 3) as temp,insurance where temp.i\_amount=insurance.i\_amount;

2.3.15 基金收益两种方式排名

该任务关卡跳过。

2.3.16 持有完全相同基金组合的客户

该任务关卡跳过。

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

该任务关卡跳过。

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。

## 2.4 数据查询(Select)之二

本节的核心内容仍然是MySQL中的数据查询相关知识，主要是对2.3节的补充，题目在逻辑上更加复杂。

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

本节主要需要我们在实践中学习MySQL中数据的插入、修改和删除的一些方法，重点在于对关键字Insert、Update、Delete的学习和使用，本身在逻辑上难度较低。

2.5.1 插入多条完整的客户信息

本任务需要我们用insert语句向表client中插入几条完整的数据，只需要按照表的格式来即可，难度较低，下面仅给出插入一条完整数据的代码。

insert into client

values(

1,

'林惠雯',

'960323053@qq.com',

'411014196712130323',

'15609032348',

'Mop5UPkl'

);

2.5.2 插入不完整的客户信息

相比于2.5.1，本任务需要我们仅仅插入一条客户信息中的部分字段，这个时候就需要指定插入的信息属于哪一列，仍然需要使用insert指令，最终代码如下所示。

insert into client(

c\_id,

c\_name,

c\_id\_card,

c\_phone,

c\_password

)

values(

33,

'蔡依婷',

'350972199204227621',

'18820762130',

'MKwEuc1sc6'

);

2.5.3 批量插入数据

本任务涉及到了insert较高级的用法，需要将另一个和表client格式一致的表中的数据批量导入到表client中，最终代码如下。

insert into client(

c\_id,

c\_name,

c\_mail,

c\_id\_card,

c\_phone,

c\_password

)

select \* from new\_client;

2.5.4 删除没有银行卡的客户信息

本任务涉及到了delete的用法，主要是通过not exists确定哪些客户没有银行卡，然后把这些用户删除即可，最终代码如下。

delete from client where

not exists(select 1 from bank\_card where b\_c\_id=c\_id);

2.5.5 冻结客户资产

本任务涉及到了update的用法，主要是用来更新字段的内容。本任务要求冻结指定手机号的客户的资产，所以我们需要先通过一个查询获取手机号，然后用关键字in判断指定的手机号是否在查询到的手机号中，最终代码如下。

update property

set pro\_status='冻结'

where '13686431238' in (select c\_phone from client where c\_id=pro\_c\_id);

2.5.6 连接更新

本任务需要根据client表中的身份证号信息更新property表中的身份证号信息，最终代码如下。

update property,client

set pro\_id\_card=c\_id\_card

where c\_id=pro\_c\_id;

## 2.6 视图

本节中开始涉及到视图相关的知识，合理的使用视图可以大大简化一些SQL操作。本节首先将介绍视图的创建方法，并且将基于视图进行数据查询，让我们在实践中感受到在一些场景下合理使用视图可以简化SQL操作的复杂性，从而写出更易读的SQL代码。

2.6.1 创建所有保险资产的详细记录视图

本任务涉及到用create语句创建视图的知识，需要创建包含所有保险资产记录的详细信息的视图v\_insurance\_detail，包括购买客户的名称、客户的身份证号、保险名称、保障项目、商品状态、商品数量、保险金额、保险年限、商品收益和购买时间，也就是说需要先用一个select语句得到这些信息，然后使用create view创建视图，最终代码如下所示。

create view v\_insurance\_detail as

select c\_name,c\_id\_card,i\_name,i\_project,pro\_status,pro\_quantity,i\_amount,i\_year,pro\_income,pro\_purchase\_time from property,insurance,client

where c\_id=pro\_c\_id AND pro\_type=2 AND i\_id=pro\_pif\_id;

2.6.2 基于视图的查询

本任务需要基于2.6.1中创建的视图v\_insurance\_detail做一些查询，重点是要计算保险投资总额和保险投资总收益这两个字段，也就是需要进行分组统计，需要根据身份证号这一字段进行分组，最终计算出这些信息之后再按照保险投资总额降序排列即可，最终代码如下所示。

select c\_name,c\_id\_card,sum(pro\_quantity\*i\_amount) as insurance\_total\_amount,sum(pro\_income) as insurance\_total\_revenue from v\_insurance\_detail

GROUP BY c\_id\_card

ORDER BY insurance\_total\_amount DESC;

## 2.7 存储过程与事务

本节主要涉及到了存储过程这一知识。存储过程是一种可编程的数据对象，可以让传统的SQL语言拥有更加强大的功能，使得SQL代码编写的过程更加规范，具有模块化的特点。本节将主要介绍用三种控制结构进行存储过程的构造，即流程控制语句、游标、事务。此外，由于这部分代码较长，所以仅仅介绍思路，而不直接粘贴代码。

2.7.1 使用流程控制语句的存储过程

本任务让我们用流程控制语句实现斐波那契数列的求解，也就是向表fibonacci中插入斐波那契数列的前n项，需要我们掌握使用流程控制语句的存储过程的编写。其实这里和常见的编程语言的编写已经很类似了，用递推公式来迭代计算斐波那契数列的下一项，每计算出一项就将它插入到表fibonacci中即可，当然在最开始需要加入一些特判。

2.7.2 使用游标的存储过程

本任务在逻辑上相对复杂一点，其实是一个用SQL的模拟过程，只需要将整个排班要求模拟出来即可，这里用一个循环变量i遍历从start\_date到end\_date之间的所有日期，用MySQL自带的dayofweek来获取这个日期具体是周几，注意dayofweek的返回和我们的常识有点区别。如果当前是周末并且轮到的医生是主任医生的话，就设置一个flag表示发生了这种特殊情况，然后读出下一个医生的信息，对其进行排班；如果当前是周一的话就看一下之前设置的flag，看一下有没有主任医生在之前进行了轮班，如果有的话就对其进行排班；否则就按最正常的进行排班即可。

2.7.3 使用事务的存储过程

本任务涉及到了事务的定义和应用相关的知识，用的是一个非常典型的银行转账的例子，事务定义的核心大概就是模拟，也就是按照任务中给出的注意事项来确定此时的转账能否发生，如果不能的话就rollback，否则就commit。

## 2.8 触发器

本节主要涉及到了触发器的相关知识，触发器是与某个表绑定的命名存储对象，与存储过程一样，它由一组语句组成，当这个表上发生某个操作(insert,delete,update)时，触发器被触发执行，这一部分编写的代码也比较长，所以仅仅叙述核心思路。

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

本任务需要我们用触发器实现一套完整性业务规则，在进行INSERT、DELETE、UPDATE等操作时激活触发器对数据进行检查，看看它是否符合业务规则，涉及到了MySQL中触发器的创建方法、抛出异常的方法、字符串拼接函数CONCAT()的使用。

主要的流程在任务书中已经写的比较明确了，也就是对new表中的pro\_type进行判断：如果不是我们预设的类型的话，用CONCAT()函数构造违法信息，用signal sqlstate抛出异常；如果是类型1的话，查看pro\_pif\_id是不是finances\_product中的某个主码值，如果不是就抛出异常，具体可以用关键字not exists进行判断；如果是类型2或类型3的话，只需要用类似的方法对约束进行检查即可，此处不再赘述。

## 2.9 用户自定义函数

本节中需要我们学习MySQL中创建并调用一个自定义函数的方法，自定义函数在创建之后就可以像库函数一样使用它，从而进一步提升了编写SQL语句的模块化程度，大大提升编程效率。

2.9.1 创建函数并在语句中使用它

本任务涉及到MySQL中函数的定义以及在select语句中应用自定义函数的方法，关键是利用create function语句创建函数的方法，这和我们以往学习的高级程序语言类似，创建的函数有参数也有返回值，只需要按照语法规则编写即可。我们这里需要编写一个计算所有储蓄卡余额的函数，只需要用select进行单表查询并用sum()进行计算即可，函数定义代码如下。

create function get\_deposit(client\_id int)

returns numeric(10,2)

begin

declare ans numeric(10,2);

select sum(b\_balance) into ans from bank\_card where b\_type='储蓄卡'

and b\_c\_id=client\_id;

return ans;

end$$

调用方法和调用库函数类似，调用部分的代码如下。

select c\_id\_card,c\_name,get\_deposit(c\_id) as total\_deposit

from client

where get\_deposit(c\_id)>=1000000

order by get\_deposit(c\_id) desc;

## 2.10 安全性控制

本节主要涉及到了MySQL中的安全性控制相关的知识，介绍了MySQL中的用户、角色、权限等方面的知识。在实践中我们可以掌握MySQL中用户、角色、权限的创建和回收，并且通过这些安全性控制手段防止数据泄露、更改或者被破坏。

2.10.1 用户和权限

本任务要求我们在金融应用场景数据库环境中，创建用户，并给用户授予指定权限，涉及到了MySQL中的安全性控制机制、CREATE USER语句的使用、GRANT和REVOKE语句的使用等知识。本任务的介绍中详细给出了这些知识的解释以及使用方法，所以在具体实现上难度并不算很大。最终代码如下。

#(1) 创建用户tom和jerry，初始密码均为'123456'；

create user 'tom' identified by '123456';

create user 'jerry' identified by '123456';

#(2) 授予用户tom查询客户的姓名，邮箱和电话的权限,且tom可转授权限；

grant select(c\_name,c\_mail,c\_phone) on client to 'tom' with grant option;

#(3) 授予用户jerry修改银行卡余额的权限；

grant update(b\_balance) on bank\_card to 'jerry';

#(4) 收回用户Cindy查询银行卡信息的权限。

revoke select on bank\_card from 'Cindy';

2.10.2 用户、角色与权限

本关我们需要创建角色，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。相关的知识在2.10.1节中已经介绍过，这里主要涉及到了用create role语句创建角色。最终代码如下。

# (1) 创建角色client\_manager和fund\_manager；

create role client\_manager;

create role fund\_manager;

# (2) 授予client\_manager对client表拥有select,insert,update的权限；

grant select,insert,update on client to client\_manager;

# (3) 授予client\_manager对bank\_card表拥有查询除银行卡余额外的select权限；

grant select(b\_type,b\_c\_id,b\_number) on bank\_card to client\_manager;

# (4) 授予fund\_manager对fund表的select,insert,update权限；

grant select,insert,update on fund to fund\_manager;

# (5) 将client\_manager的权限授予用户tom和jerry；

grant client\_manager to tom,jerry;

# (6) 将fund\_manager权限授予用户Cindy.

grant fund\_manager to Cindy;

## 2.11 并发控制与事务的隔离级别

本节主要涉及到了MySQL中的并发控制与事务隔离级别相关的知识。在并发操作中，很有可能出现数据不一致性，如丢失修改、读脏数据、不可重复读、幻读等数据不一致性。为解决上述不一致性问题，DBMS设计了专门的并发控制子系统，采用封锁机制进行并发控制，以保证事务的隔离性和一致性，本节我们将会学习到这些机制。

2.11.1 并发控制与事务隔离级别

本任务需要将事务的隔离级别设置成read uncommitted，根据任务中的知识介绍，只需要学会MySQL中事务隔离级别的设置方法即可。核心代码如下。

set session transaction isolation level read uncommitted;

2.11.2 读脏

脏数据实际上是一个事务读取到了另一个事务修改后的数据，而另一个事务后来撤销了本次修改。本任务要求我们设置合适的事务隔离级别，构造出读脏这一现象。实际上为了演示读脏现象，我们需要把事务隔离级别设置成最低级别READ UNCOMMITTED。为了控制事务执行的次序，我们需要借助sleep语句，也就是让事务1开始时sleep1秒，事务2对数据发生修改后sleep2秒再rollback，从而构造出读脏这种情况，代码此处略过。

2.11.3 不可重复读

不可重复读是指一个事务读取到某数据后，另一个事务修改了该数据，后来当第一个事务再次读取该数据的时候，发现两次读取的数据不一致。本任务要求我们设置合适的事务隔离级别，构造出不可重复读这一现象。根据分析，我们可以将事务隔离级别设置成READ UNCOMMITTED，然后设置合适的事务执行次序即可。最开始先让事务1读一次数据，事务2sleep1秒，事务1读过之后直接sleep2秒，事务2读一次之后再次sleep2秒，这个时候事务1会对数据进行修改，然后sleep3秒，然后事务2再读一次数据，发现数据不一致，产生了不可重复读现象。

2.11.4 幻读

幻读是在某一事物多次读取数据之间，另外一个事务对数据集进行了insert或delete。本任务要求在较高隔离级别repeatable read下重现幻读现象。本关实现起来比较简单，只需要让事务1在第一次查询之后sleep2秒即可，最终代码如下。

use testdb1;

select @@transaction\_isolation;

start transaction;

## 第1次查询余票超过300张的航班信息

select \* from ticket where tickets>300;

set @n=sleep(2);

-- 修改航班MU5111的执飞机型为A330-300：

update ticket set aircraft = 'A330-300' where flight\_no = 'MU5111';

-- 第2次查询余票超过300张的航班信息

select \* from ticket where tickets > 300;

commit;

2.11.5 主动加锁保证可重复读

本任务要求在事务隔离级别较低的read uncommitted情形下，通过主动加锁，保证事务的一致性。主要需要我们掌握MySQL中的锁机制，以及通过任务2.11.3中类似的方法控制并发顺序来尝试触发不可重复读，最终发现通过主动加锁的方式可以保证可重复读。代码略过。

2.11.6 可串行化

本任务要求选择除serializable(可串行化)以外的任何隔离级别，保证两个事务并发执行的结果是可串行化的。本关的实现较为简单，核心思路就是在合适的地方用sleep控制顺序以及在合适的地方加锁即可，代码此处略过。

## 2.12 备份+日志：介质故障与数据库恢复

本节主要涉及到了MySQL中的介质故障与恢复服务。为了保证数据库软件在实际生产环境中的高可用性，如今的数据库软件都提供了一套备份恢复机制，也就是在发生如存储介质故障这样的问题的时候可以利用数据库备份和日志文件来恢复数据库，解决这些问题，提高数据安全性和可靠性。

2.12.1 备份与恢复

本任务主要涉及了MySQL中提供的一些备份与恢复工具，这里主要使用了MySQL dump工具来对MySQL进行备份。整体逻辑比较简单，因为任务书中给出了这部分知识的用法，需要先用MySQL dump将数据库备份，然后再用备份文件重启数据库，最终代码如下。

# 对数据库residents作海量备份,备份至文件residents\_bak.sql:

mysqldump -h127.0.0.1 -uroot --databases residents > residents\_bak.sql

# 利用备份文件residents\_bak.sql还原数据库:

mysql -h127.0.0.1 -uroot < residents\_bak.sql

2.12.2 备份+日志：介质故障的发生与数据库的恢复

本任务模拟了介质故障的发生，介绍了利用备份和备份之后的日志恢复数据库的方法，用到的工具仍然是MySQL dump，不过增加了工具MySQL binlog的使用。在备份的时候，相比2.12.1需要使用—flush-logs刷新日志，代码如下所示。

mysqldump -h127.0.0.1 -uroot --flush-logs --databases train > train\_bak.sql

随后需要用备份的文件恢复数据库，代码如下所示。

mysql -h127.0.0.1 -uroot < train\_bak.sql

mysqlbinlog --no-defaults log/binlog.000018 | mysql -uroot

## 2.13 数据库设计与实现

本节主要涉及了数据库的设计与实现的过程，主要是对实际生产环境中的场景进行建模。

2.13.1 从概念模型到MySQL实现

本任务中给出了场景的E-R图，要求根据概念模型建立实际的MySQL表，只需要根据概念模型一步步来把整个数据库表建立完成即可，涉及到的代码较长，但是逻辑不复杂，此处省略。

2.13.2 从需求分析到逻辑模型

该任务关卡跳过

## 2.13数据库设计与实现

本节主要涉及了数据库的设计与实现的过程，主要是对实际生产环境中的场景进行建模。任务内容设计了从概念模型到MySQL实现、从需求分析到逻辑模型、建模工具的使用三个部分，囊括了将实际生产环境中的场景抽象成数据库表再到最终应用到MySQL数据库中的过程。

请用一段文字（原则上控制在5行以内）总体上概括本小节任务的内容与要求。

**注：本章内容的过程、绘图排版可依据实际情况展开，不受之前的三级子任务关于过程和图片的版面约束限制。**

2.13.1 从概念模型到MySQL实现

本任务中给出了场景的E-R图，要求根据概念模型建立实际的MySQL表，只需要根据概念模型一步步来把整个数据库表建立完成即可，涉及到的代码较长，但是逻辑不复杂，此处省略。

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

本任务需要学习数据库的设计与实现过程中常见的建模工具。任务文档中介绍了几种常用的建模工具，最终任务实现要求使用MySQL Workbench，使用该工具将任务文档中给出的模型文件导出成SQL脚本即可，整体过程较为简单。

2.13.4制约因素分析与设计

**（注意本小节为工程认证要求动作）**

**本小节内容篇幅各人可依据完成情况相对灵活设定，要求内容上有必要的响应，能够结合任务中的具体内容（例如业务流程的设计、数据结构的属性及信息之间的关联、数据的约束、对用户操作权限或者购买的权限的设计等问题的设计与实现中有哪些这方面的考虑），但不进行琐碎的赘述。建议本小节篇幅少则一段话，多则尽量控制不超过1页。**

阐述如何在任务的解决方案设计中考虑了社会、健康、安全、法律、文化及环境等**制约因素**。

可逐条归纳，配合适量的关联内容表格，也可配合适当的体现上述因素的方案图。

2.13.5工程师责任及其分析

**（注意本小节为工程认证要求动作）**

**本小节内容篇幅各人可依据完成情况相对灵活设定，要求内容上有必要的响应，能够结合任务中的具体内容，但不进行琐碎的赘述。建议本小节篇幅少则一段话，多则尽量控制不超过1页。**

分析和评价任务的解决过程和解决方案与社会、健康、安全、法律以及文化等**各因素之间的相互影响**，阐述对**工程师应承担的相关社会责任**的理解。

## 2.14 数据库应用开发(JAVA篇)

本节要求用java进行数据库应用开发，任务基于JDBC这组用于执行Java语句的Java API。本节需要我们掌握JDBC的体系结构和一些对数据库的简单操作，学习实际生产中后端的一些CRUD操作。本节中任务书给出的知识介绍非常详细，即使是对于没有Java基础的人来说想要完成本节的实验也比较轻松。

请用一段文字（原则上控制在5行以内）总体上概括本小节任务的内容与要求。

**注：本章内容的过程、流程图等绘图排版可依据实际情况展开，不受之前的三级子任务关于过程和图片的版面约束限制。**

2.14.1 JDBC体系结构和简单的查询

本任务需要我们了解JDBC体系结构并且对于用JDBC操作数据库的具体方法有简单的了解，具体要求我们查询client表中邮箱非空的客户，列出相关信息。利用JDBC开发的主要过程大概是首先创建数据库连接对象、创建Statement对象、用Statement对象执行SQL语句、借助ResultSet对结果进行输出。这里大部分步骤在任务书中都有示例代码，因此我们只需要

2.14.2 从需求分析到逻辑模型用户登录

。。。。。

。。。。。

## 2.15数据库的索引B+树实现

请用一段文字（原则上控制在5行以内）总体上概括本小节任务的内容与要求。

**注：本章内容的过程、流程图等绘图排版可依据实际情况展开，不受之前的三级子任务关于过程和图片的版面约束限制。**

2.15.1 BPlusTreePage的设计

。。。。。

2.15.2 BPlusTreeInternalPage的设计

。。。。。

。。。。。

# 3 课程总结

原则上可分为三个部分，第一部分先概述、总结此次课程实践的总体任务及其完成情况；第二部分可结合任务书第1章课程介绍中的各部分内容逐条归纳本次实践的主要工作（不要直接拷贝任务书内容，请归纳自己的主要工作内容和完成情况）；第三部分叙述此次课程实践的心得体会，展望此次课程实践的有待改进和完善的工作。**（本章原则上控制在一页以内）**

# 附录

（ 附录作为实践报告的可选项，内容可以是在报告正文中受篇幅所限不宜过多粘贴但有必要补充展示的结果表格、图片、复杂过程的流程图等等。个人可依据实际情况，若觉得没有必要，附录可忽略不写。

原则上附录篇幅请控制在5页左右，字体排版和正文相同）

**补充说明：**

**课程结束后，以班级为单位各班刻录一张光盘（或者按照学院的最新归档管理方式执行，可能可直接以班为单位提交一个归档压缩文件），每位同学在光盘中提交一个压缩文档，文件名为“学号-姓名.rar”，里面包含实践报告的doc和pdf版本、存储所有源代码的目录（目录中创建一个readme.txt文件对本代码目录的结构和内容做必要的说明，然后自行组织本代码目录中的子目录或文件来提交本此处实践课所完成的全部代码）。**