

**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2110班 |
| 学 号： | U202115610 |
| 姓 名： | 张锦深 |
| 指导教师： | 胡贯荣 |

|  |  |
| --- | --- |
| 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)

# 1 课程任务概述

本实验中要求将课程所学数据库理论知识转化具体实践，完成SQL语句的编写并用于解决具体实际任务，最终达到增强SQL语句编写能力，熟悉数据库CRUD操作，提高工程创新实践水平的目的。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)

# 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2.1，2.2，2.3，2.4，2.5，2.6，2.7，2.8，2.9，2.10，2.11，2.12，2.13，2.14，2.15子任务，下面将重点针对其中的任务阐述其完成过程中的具体工作。

## 2.1 数据查询(SELECT)之一

本小节任务主要包括对表的简易查询操作。要求能够熟悉掌握SELECT的语句的结构并能够用SQL语句实现简易查询逻辑。实验环境为某一银行的金融场景应用的模拟数据库finance，其中包含client,bank\_card,finances\_product, insurance,fund,property六个表。

本任务已完成查询客户主要信息，邮箱为空的客户，既买了保险又买了基金的客户，办理了储蓄卡的客户信息，每份金额在30000-50000之间的理财产品，商品收益的众数，未购买任何理财产品的武汉居民，持有两张信用卡的用户，购买了货币型基金的客户信息，投资总收益前三名的客户，黄姓客户持卡数量，客户理财，保险与基金投资金额，客户总资产，第N高，基金收益两种方式排名，持有完全相同基金组合的客户，购买基金的高峰期，至少有一张信用卡余额超过5000元的客户信用卡总额，以日历表显示每日基金购买总金额关卡，共19关。

**2.1.1 客户理财、保险与基金投资总额**

关卡任务主要是查询所有客户及其投资的总金额。投资产品种类包含fund,finance\_product,insurance三种，涉及其对应的三个表。要计算投资金额，还涉及表property中的pro\_quantity属性，表property包含外码pro\_pif\_id，其引用f\_id,p\_id,i\_id。此外，还需要设计

实验步骤思想主要是分而治之。将投资总金额分解为基金投资加理财投资加保险投资。转化为SQL逻辑为先按类型查询利用各用户的基金投资总额，理财产品投资总额，保险投资总额，再将用户信息依次与他们进行左外连接，计算得到总金额和用户信息。具体步骤如下

按类型分别查询用户各用户的基金投资总额，理财产品投资总额，保险投资总额。如图2.1所示，图2.1是查询投资理财产品的用户及其投资总额。主要利用了内连接和聚集函数实现理财产品的投资总金额查询，基金以及保险的查询类似，更换连接属性以及筛选条件的值即可。

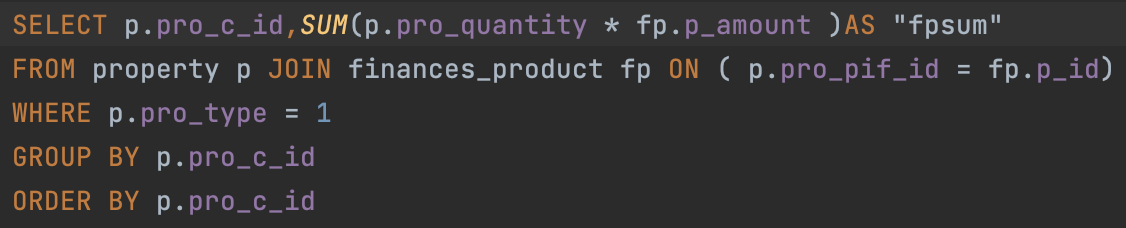


图2.1 查询投资理财产品的用户及其投资总额

综合用户信息和各投资表，利用左外连接来保证所有客户均被考虑到，无论他是否进行了投资。同时，查询对象为用户ID及其投资总和，注意，左外连接会产生空值问题，为了避免，添加条件判断IFNULL来解决该问题，如图2.2所示。

截屏2023-05-30 16.12.48

图2.2 查询用户信息及所有投资总额。

**2.1.2 购买基金的高峰期**

关卡任务是查询2022年2月购买基金的高峰期，其中，购买基金高峰期要求连续三天每日的基金交易总额均超过100万元。实验具体步骤如下。

先利用WITH...AS公用子表达式来封装2月基金交易额大于100万交易日的日期，当日基金交易总额及其属于2月的第几个工作日信息的查询结果。公用表达式中查询语句利用左外连接将日期表和金额表进行连接，并使用RANK...OVER窗口函数按照交易日日期进行排序，从而得到2月每个交易日日期dat、当日基金交易总额total\_amount、2月第几个交易日rk共三行信息，最后利用WHERE进行筛选得到交易额符合大于100万要求的交易日。

上一个步骤查询了所有大于100万交易日的信息，现在判断是否为高峰期，要求三个交易日连续，判断依据为是否有三个以上的连续的rk出现，若有，则为高峰期，筛选高峰期SQL语句如下图，其中temp为上一步骤得到的公用表达式。

该方法在实施初期遇到了问题，考虑的不够健全。错误原因是一开始没有与日期表进行左外连接，导致2.17该日因为当日没有进行基金交易而被忽略，导致rk计算错误，最后通过与日期表左外连接来保证每个日期都有。

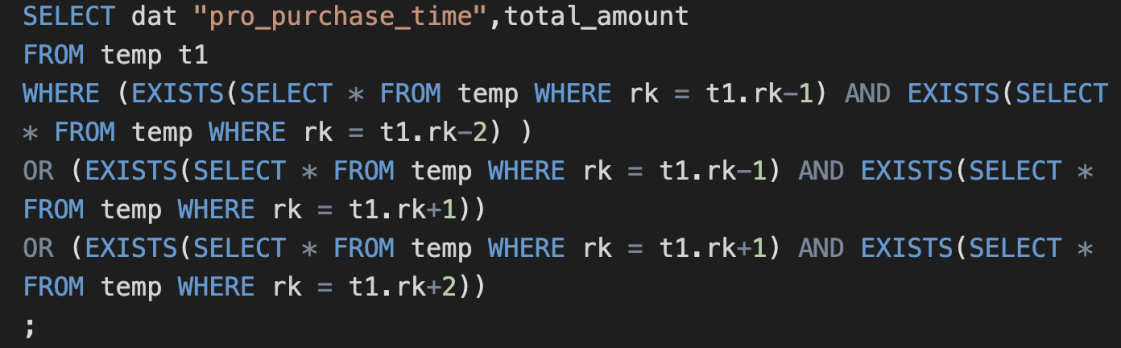


图2.3 查询基金高峰期的SQL语句代码

**2.1.3 以日历表的形式显示每日基金购买总金额**

关卡任务是以日历表的形式显示2022年2月每日基金购买总金额，简单来说就是行转列。

主要步骤是先查询二月每个交易日相关信息，包括所在交易周，星期几，以及当天的交易金额。此时得到的是行格式。如图2.3所示

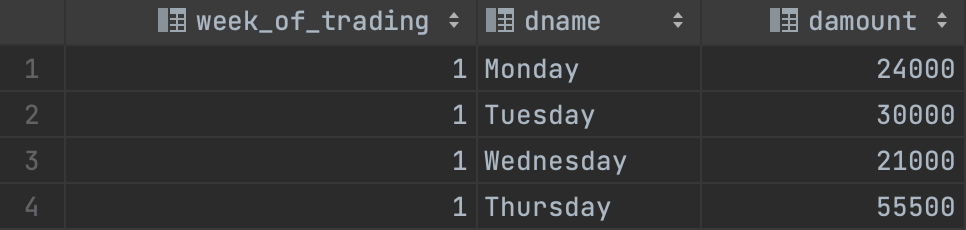


图2.3 2022年2月每个交易日相关信息查询结果

然后将行转化为列。第一步通过查询加条件判断语句先进行列扩展，根据每条信息所在的周和星期进行扩展，日历表的每一列，列扩展结果如图2.4所示.

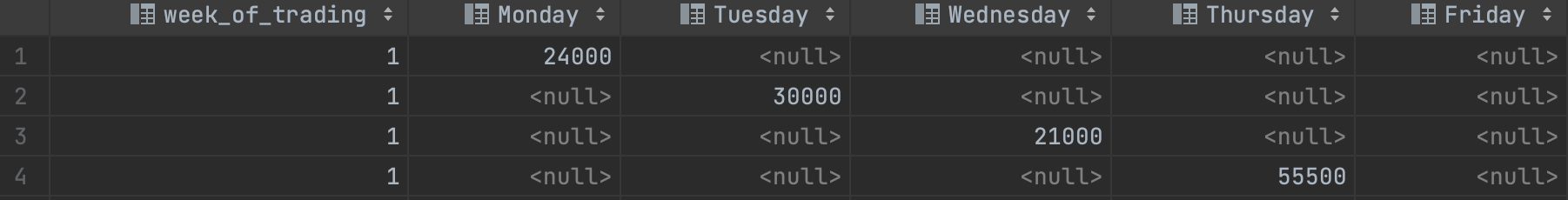


图2.4 列扩展结果

在根据分组查询，将同一交易周的合并，利用聚集函数MAX来解决空值问题，最终合并结果如图2.5所示，行转列成功实现。

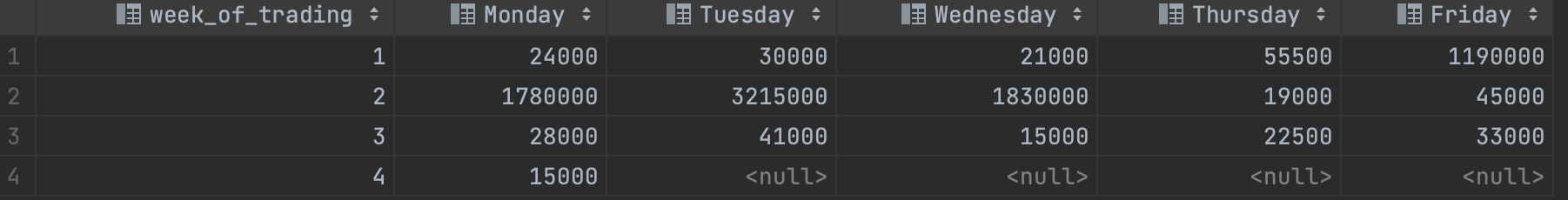


图2.5 行合并结果

## **2.2 数据查询(SELECT)之二**

本小节任务依旧为查询，是2.3小节的扩展，融合了了窗口函数以及推荐算法相关知识。实验环境一致，查询逻辑难度上升。要求熟悉掌握了查询语句及能够根据逻辑组织查询语句。

本任务已完成查询销售总额前三的理财产品，查询投资积极且偏好理财类产品的用户，查询购买了所有畅销理财产品的用户，查询相似的理财产品，查询任意两个用户的相同理财产品数以及查询相似理财客户，共6个关卡。

**2.2.1 查询投资积极且偏好理财产品类的客户**

该任务是条件查询。查询条件之一为投资积极，即购买理财产品种类大于三种，另一条件为偏好理财产品，要求持有基金种类数量小于持有的理财产品数。主要步骤如下。

先查询投资积极的客户，利用GROUP BY...HAVING 语句和COUNT聚集函数以及DISTINCT对持有投资理财产品种类大于3的客户进行过滤，得到投资积极的客户id及其拥有理财产品数量。

同上面的逻辑，查询每个客户的id及其拥有的基金产品数量，只是不进行过滤。

将投资积极客户id与持有基金产品客户id进行左外连接，注意，这里必须左外连接，避免空值问题引起的信息丢失，再WHERE筛选得到持有理财产品种类数目大于持有基金种类数目的客户id即可。

**2.2.2 查找相似的理财产品**

该任务与推荐算法相关，相似产品的定义是该产品的拥有量前三的客户所拥有的其他产品，而产品相似度是指相似产品被全体客户持有总人数。该任务要求查询14号理财产品的相似产品及其与14号理财产品的相似度，实验具体步骤如下。

先查询每个顾客的14号理财产品购买量并利用DENSE\_RANK...OVER按照购买量进行排序。

然后查询筛选购买量排名前三的客户id，再查询购买14号产品前三客户购买的其他理财产品id，即相似产品id。

最后根据相似产品id查询购买该产品的用户数量，得到相似产品pro\_pif\_id及其相似度cc，并利用DENSE\_RANK...OVER()进行排序得到rk然后筛选相似度前三的产品信息即可。

## 2.3 存储过程与事务

本小节任务内容主要包括存储过程与事务，要求掌握存储过程、变量的定义和使用，流程控制语句的使用，游标的定义与使用以及事务的定义与使用。

本小节任务已完成使用流程控制语句的存储过程，使用游标的存储过程以及使用事务的存储过程三个关卡。下面将重点针对其中的使用游标的存储过程任务进行阐述。

**2.3.1 使用游标的存储过程**

任务主要内容是医院某科室排班，要求按照工号每天安排1名医生值夜班，2名护士，其中，主任医师不排周末夜班，主任医师轮空，其他医生递补直至下一周一。该任务涉及到了存储过程的定义，游标、异常处理器的定义和使用以及流程控制语句的使用知识点，综合性较强，具体实验步骤如下。

定义相关变量，包括当前日期c\_date，用于循环控制；医生名字doctor\_name、两个护士名字nurse1\_name,nurse2\_name，用来保存游标读入的信息；周末以及星期一标志位is\_weekend,is\_Monday，用于标志是否为周末或周一；游标出错标志位doc\_flag,nurse\_flag，用于标志doc\_cursor,nurse\_cursor是否出现取空异常，方便异常处理；标志为doc\_selected,nurse\_selected，用于标志是否完成医生，护士排班；标志位doctor\_weekend，用于标志出现主任医生按顺序排班周末的情况。

定义游标doc\_cursor,nurse\_cursor分别用于便利employee表中医生和护士名字。

定义异常处理器用于解决cursor带来的NOT FOUND 异常，类型为CONTINUE，在处理器中，判断doc\_selected，若doc\_selected ==FALSE，说明为doc\_cursor出现了NOT FOUND 异常，将doc\_flag设置为TRUE，并关闭doc\_cursor然后重新开启；否则说明nurse\_cursor出现了NOT FOUND 异常，将nurse\_flag设置为TRUE，并关闭nurse\_cursor然后重新开启。

变量定义并初始化之后，进入循环判断语句，循环控制条件为c\_date<=end\_data，每次循环末尾修改c\_date++。在每一轮循环中，先排班医生。医生排班逻辑如下，先判断是否为周一is\_Monday且是否存在主任医师排班周末轮空情况doctor\_wee kend，若都符合，则说明为主任医师调班周一，则该天排班主任医师即可，医生选择结束，设置doc\_selected为TRUE；若doc\_selected=FALSE，即未出现上述情况，则正常排班，利用doc\_cursor取排班医生姓名，进行特殊判断，若该天为周末且取到的医生为主任医师，则设置doc\_weekend为TRUE,并继续取，选择完后设置doc\_selected=TRUE。注意，过程中每次使用游标取得信息之后，均需要对doc\_flag进行判断，若doc\_flag为TRUE，即存在NOT FOUND异常，则进行异常处理，重新利用游标取得护士信息，完成后设置doc\_flag为FALSE。

之后就是取护士，取护士逻辑简单，利用游标分别取得两次护士信息并保存，注意，每次利用游标取信息后对nurse\_flag进行判断，如果nurse\_flag为TRUE，则进行异常处理，重新用游标取护士信息，完成设置nurse\_flag为FALSE。最后设置nurse\_selected为1；

依次取得该天值班的医生和护士信息，将其插入night\_shift\_schedule表中。最后修改循环控制条件，cdate = cdate + 1day，结束该轮循环。

## 2.4 触发器

本小节任务主要包括触发器的创建与使用，要求掌握触发器创建的语法，时机以及流程控制语句。

本小节任务已完成为投资表property实现业务约束规则1个关卡。下面将重点针对其中的为投资表property实现业务约束规则任务进行阐述。

**2.4.1 为投资表property实现业务约束规则**

该任务要求实现property表业务约束规则，在插入前设置一个触发器检查数据是否符合约束条件。主要涉及到触发器的创建与使用，SQL抛出异常和错误以及流程控制知识点，具体步骤如下。

创建行触发器befor\_property\_inserted，选择触发时机为插入property表前。在触发器中，首先判断插入新元组的pro\_type是否符合完整性约束，即是否在(1,2,3)中，若不在，抛出用户自定义异常SQLSTATE ‘45000’，异常信息为”type \* is illegal”。然后根据pro\_type查询相应表中是否存在一样的pro\_pif\_id，即pro\_pif\_id是否符合完整性约束，若不符合，抛出用户自定义异常，异常消息为”class #\* not found”。如此，就完成了一个实现property表业务约束规则的触发器。

## 2.5并发控制与事务的隔离级别

本小节任务主要包括事务的隔离级别设置以及并发控制。要求熟悉掌握并发操作带来的问题以及解决方法。

本任务已完成并发控制与事务的隔离级别，读脏，不可重复读，幻读，主动加锁保证可重复读，可串形化六个个关卡。下面将重点针对其中的为主动加锁保证可重复读关卡进行阐述

**2.5.1 主动加锁保证可重复读**

本任务中包含两个事务T1,T2，T1依次对MU2455航班余票进行两次查询，而T2事务中对MU2455航班进行了出票操作，若不加以控制，则出现事务T1两次查询的余票不同，即不可重复读现象。先要求T1事务中两次查询的结果应保持一致，这就要求我们加以并发控制。该关卡主要涉及并发控制技术中的加锁技术。具体实验步骤如下。

首先设置隔离级别为read uncommitted，来保证会出现不可重复读现象。然后为事务T1中两次读操作，即查询MU2455航班余票操作，加上共享锁，来保证读的过程中事务T2的写操作不能并发进行，即事务T2等待。直至事务T1两次查询结束后，提交事务，释放tickets表锁，此时T2进行写操作。如此，通过加锁保证了可重复读。

添加一个时间表。

## 2.6 数据库设计与实现

本小节任务主要是数据库的设计与实现，涉及到数据库的需求分析，概念设计，逻辑设计以及物理设计。要求掌握数据库设计实现的步骤方法以及正逆向建模工具的使用。

**2.6.1 从概念模型到MySQL实现**

本任务主要内容为了解数据库设计的阶段和以及每个阶段的任务，最终将图所示的ER模型转化为具体的关系模式，实现从概念设计到逻辑设计的过程。具体实验步骤如下。

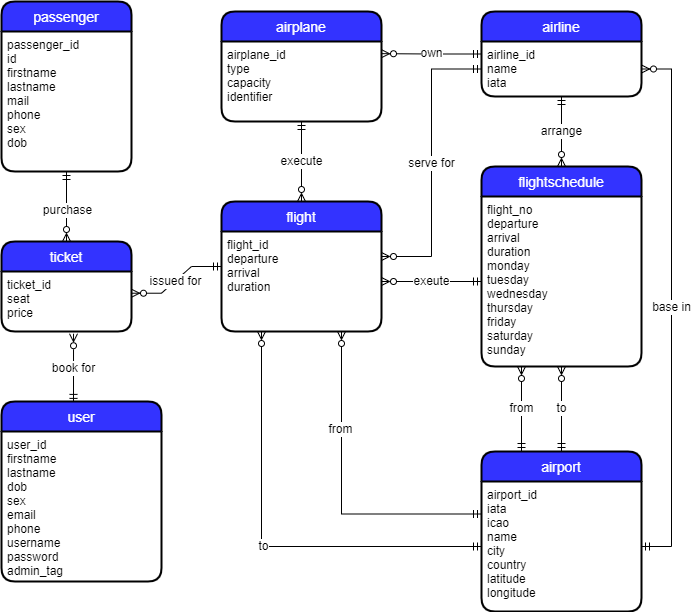


图 任务ER模型

该数据库包含8个实体型，则至少包含8个关系模式，分别代表一个实体型，再根据实体型之间的关系进行合并。以下是从实体型的角度一一进行分析，也可以从关系角度逐一进行分析。

对于passenger,user,airport实体型，不存在多对一以及多对多的对外关系，则无其他属性需要合并增加。根据固有属性定义关系模式，建立主码约束以及完整性约束即可。接下来就是找到多对一的关系并进行合并。

对于airline实体型，存在与airport多对一的基地关系，即一个机场是多个航空公司的母港。在airline关系模式中增加属性airport\_id来建立这段关系，并为其建立外码约束。

对于airplane实体型，存在与airline多对一的被拥有关系，即一个航空公司拥有多架飞机。在airplane关系模式中添加airline\_id来保存这段关系，并为其建立外码约束。

对于flightschedule实体型，分别存在与airline,airport多对一的排班，出发地与目的地关系。在flightschedule关系模式中增加airline\_id,from,to三个属性来建立这三段关系，并为其进行外码约束。注意，from，to是语法关键词，需要用” ` ”加以环绕将属性与关键词进行区分。

对于flight实体型，分别存在与airplane,airline,flightschedule的执飞，提供和安排关系。在flight关系模式中增加airplance\_id,airline\_id,filght\_no属性来建立这三段关系，并为其建立外码约束。

对于ticket实体型，分别存在与passenger，user，flight多对一的购买，订票和被发行关系，将关系向多边实体型，即ticket，靠。在ticket增加属性passenger\_id,user\_id,filght\_id来建立这三段关系，并分别为这三个属性建立外码约束。

最终建立的关系模式如下。

1. user(user\_id,firstname,lastname,dob,sex,email,phone,username,password,admin\_tag) 主码(user\_id)
2. password(passenger\_id,id,firstname,lastname,mail,phone,seat,dob) 主码(passenger\_id)
3. airport(airport\_id,iata,icao,name,city,country,latitude,longitude) 主码(airport\_id)
4. airline(airline\_id,name,iata,airport\_id) 主码(airline\_id) 外码(airport\_id)
5. airplane(airplane\_id,type,capacity,identifier,airline\_id) 主码(airplane\_id) 外码(airline\_id)
6. flightschedule(flight\_no,departure,arrival,duration,monday,tuesday,wednesday,thursday,friday,saturday,sunday,airline\_id,from,to) 主码(flight\_no) 外码(airline\_id,from,to)
7. flight(flight\_id,departure,arrival,duration,airline\_id,airplane\_id.flight\_no,from,to) 主码(flight\_id) 外码(airline\_id,airplane\_id,flight\_no,from,to)
8. ticket(ticket\_id,seat,price,flight\_id,passenger\_id,user\_id) 主码(ticket\_id) 外码(passenger\_id,user\_id)

**2.6.2 从需求分析到逻辑模型**

本任务主要内容是设计一个影院管理系统，要求根据需求进行分析，将需求转化为概念ER模型，最终转化为关系模式，主要涉及数据库设计的方法步骤知识点。具体实验步骤如下。

分析需求，共五个实体集，分别为movie,customer,hall,schedule,ticket,他们之间存在以下关系。顾客与电影票的一对多的购买关系；电影票和排场的多对一的属于关系；排场和电影的一对多的放映关系；排场和放映厅的一对多的位于关系。根据以上关系，建立陈氏ER模型，如图所示，到此，实现了从需求分析到概念设计。

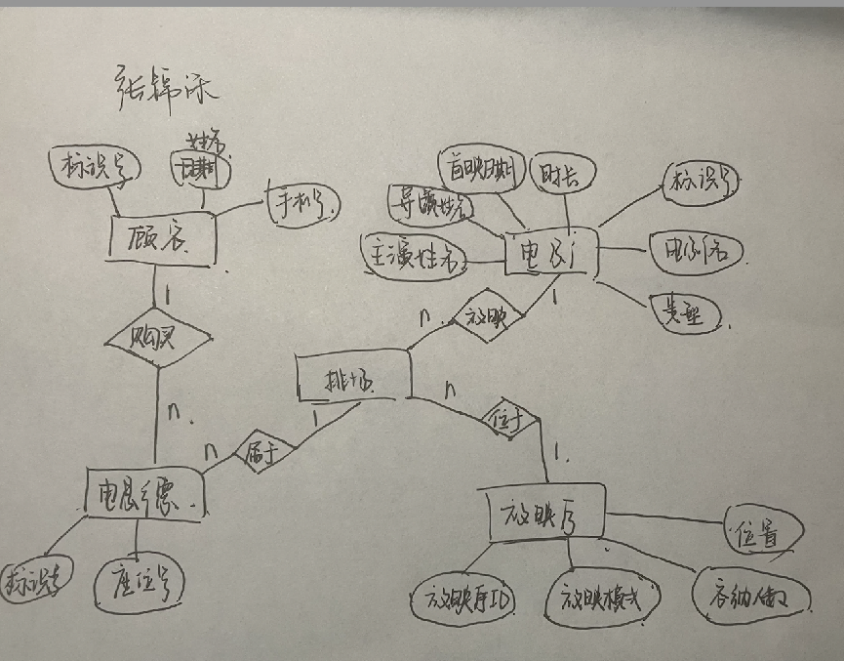


图 影院管理系统陈氏ER模型

将陈氏ER图转化为具体关系模式，方法与2.1类似，这里不做赘述。关系模式最终转化结果如下。

1. movie(movie\_ID, title, type, runtime, release\_date, director, starring), 主码(movie\_ID)
2. customer(c\_ID, name, phone), 主码(c\_ID)
3. hall(hall\_ID, mode, capacity, location), 主码(hall\_ID)
4. schedule(schedule\_ID, date, time, price, number, movie\_ID, hall\_ID), 主码(schedule\_ID), 外码(movie\_ID, hall\_ID)
5. ticket(ticket\_ID, seat\_num, c\_ID, schedule\_ID), 主码(ticket\_ID), 外码(c\_ID, schedule\_ID)

**2.6.3** **制约因素分析与设计**

在从实际问题的需求分析到数据库的概念模型和逻辑模型的构建方案设计中，需要考虑若干制约因素，如社会、健康 等。例如，在任务一的机票订票系统中，系统需要考虑到旅客的实际情况，例如，购票人与乘坐人不是同一个人的情况，所以包含两个实体型user以及passenger，从而机票信息同时与user和passenger相关联。此外，对于系统的权限也存在若干要求，例如实体集user就刻意根据根据权限分成两类：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。同时，在订票系统中，需要考虑对于每个机场，都有唯一的IATA编码和ICAO编码。

**2.6.4 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需求分析，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，并理解应承担的社会责任，做到工程造福于人民；安全方面，工程师应该尽可能考虑系统中潜在的安全问题并及时填补漏洞，确保工程的安全可靠性；科学创新方面，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，必须通过实践得到真理。法律方面，工程师在工程建造时应遵纪守法，坚守底线。

## 2.7 数据库应用开发(JAVA篇)

本小节任务主要内容是JDBC，即嵌入式SQL(Java)。要求能够通过Java语言实现对数据库的CRUD操作，实现数据库应用开发。

本任务已完成并发控制与事务的隔离级别，读脏，不可重复读，幻读，主动加锁保证可重复读，可串形化六个个关卡。下面将重点针对其中的事务与转账操作关卡进行阐述

**2.7.1 JDBC的体系结构和简单的查询**

本关卡任务是查询client表中邮箱非空的客户信息，要求掌握JDBC的体系结构、核心组件以及其JDBC执行sql语句的使用方法。具体实验步骤如下。

首先，需要建立与数据库的连接。连接数据库分为两步。先导入需要的包，主要是java.sql。然后调用Class类的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，调用 DriverManager.getConnection(String url, String user,String password)方法建立与数据库的连接。方法参数包括数据库网址url,数据库登陆用户user及其密码password，在该实验中显式的给出，实际开发中应该将其放入资源文件并从资源文件读入相关信息。

完成与数据库的连接后，创建Statement(PrepareStatement)接口实例封装需要执行的SQL语句。调用connection.createStatement()方法创建Statement接口实例对象并执行相应的sql语句。这里只需要查询邮箱非空客户，查询语句是固定静态的。若碰到带有参数的动态SQL语句，应该创建prepareStatament接口实例，并对sql语句进行动态注入并编译。

执行SQL语句。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对象，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。

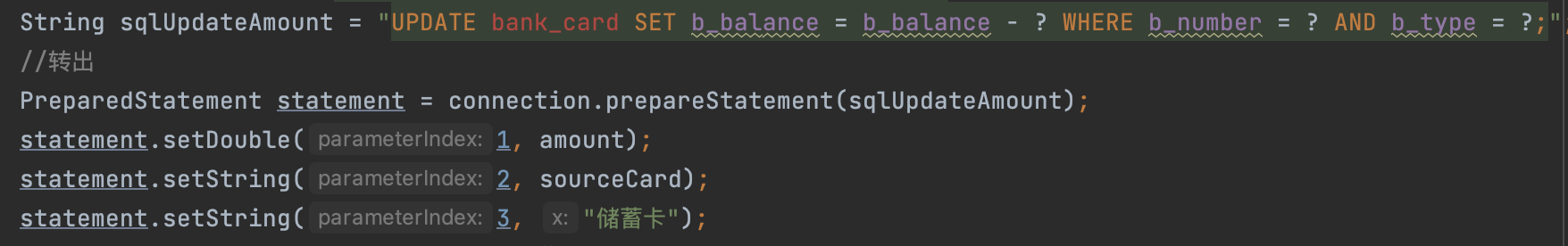
执行SQL语句后，从SQL语句执行结果集中提取数据，利用resultSet.getXXX()方法获取相应的资源。本关卡中分别利用getString()方法获取查询客户信息中的”c\_name”,”c\_mail”,”c\_phone”并逐一打印输出。

完成对数据库的操作后，需要关闭所有的资源，比如数据库连接connection, Statement(PreparedStatement)实例对象，调用close()方法关闭资源。

**2.7.1 事务与转账操作**

本关卡要求编写银行转账的方法transferBalance()，在方法中要求实现转账操作，若操作失败，则事务回退rollback，否则事务提交commit。该关卡主要涉及JDBC实现数据查询更新操作以及JDBC实现事务处理知识点。具体实验步骤如下。

首先，利用executeUpdate执行update的动态SQL语句实现账户转入转出操作。



紧接着判断，查询bank\_card表中是否存在source\_card，若不存在，connection.rollback()实现事务回退，撤销操作，若source\_card存在,判断查询结果中source\_card转出后余额是否大于0，若小于0，则转账失败，事务回退，撤销转账操作，若余额大于0，继续查询bank\_card表中是否存在source\_card，若存在，则事务正常提交connect.commit，否则事务回退，撤销转账操作。

## 2.8 数据库的索引B+树实现

本小节任务主要内容为数据库中B+树的简单实现，要求了解数据库管理系统内核以及物理存储结构设计。

**2.8.1 BPlusTreePage**的设计

本关卡任务主要内容是设计BPlusTreePage类，其该类是B+树的叶子节点和内节点的父类，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。这些成员均为私有，先任务要求编写getter以及setter方法，对外提供访问变量以及修改变量的接口。这一关只是简单的getter和setter，填空即可，不做赘述。

**2.8.2 BPlusTreeInternalPage**的设计

本关卡任务主要是设计BPlusTreeInternalPage类，该类作为B+树的内部结点类型，其继承于BPlusTreePage提供B+树内部结点的功能。对于B+树的内部页，它不存储任何数据，而只保存键以及子页索引指针，共M个键，M+1个指针。任务主要内容是实现内部结点的初始化、查找、分裂，合并以及重分配等算法，方法较多，下列对各算法中各挑选一代表方法进行着重阐述。

1. 初始化

函数名:Init

传入参数：page\_id 页号，parent\_id 父页号，max\_size页元素最大值

返回值：void

函数功能：对内部页类 BPlusTreeInternalPage实例对象进行初始化。

函数设计思想：调用父类BPlusTreePage中的Setter方法分别对page\_id,parent\_id, max\_size成员变量进行赋值。并调用SetPageType方法将该对象的页类型设置为内部页INTERNAL\_PAGE。

1. 查找

函数名:Lookup

传入参数：key 查找目标键值，comparator 键值比较器

返回值：ValueType

函数功能：在key值有序排列的数组中，找到特定key值对应的value。若没找到，返回-1；

函数设计思想：采用由于页内已按照键值大小有序排列，只需要进行二分查找即可。二分查找思想已经滚瓜烂熟，这里不做赘述，注意每次键值比较通过比较器comparator实现。

1. 内部结点分裂

函数名:MoveHalfTo

传入参数： recipient 新生成的结点,buffer\_pool\_manager 缓冲区对象

返回值：void

函数功能：移动this结点中一半（向上取整）的元素到一个新生成的recipient结点中.一般用于当前结点中元素溢出时进行分裂，将多余元素移入新结点中。

函数设计思想：调用CopyNFrom将this结点中后半部分移动到recipient结点中，并更新this结点的大小。

1. 内部结点合并

函数名:MoveAllTo

传入参数： recipient 合并结点,buffer\_pool\_manager 缓冲区对象

返回值：void

函数功能：合并this结点的元素至recipient结点中，即将this结点中的全部元素移至recipient结点的尾部。用于删除后size很小的情况下，不宜重分配而采用合并的情形。

函数设计思想：获取this的父节点，并清除父节点中指向该节点的键值对；然后调用CopyNForm方法将this节点合并到recipient节点中；最后更新子节点的父节点信息。

1. 重分配

函数名:MoveFirstToEndOf

传入参数：recipient 合并结点,buffer\_pool\_manager 缓冲区对象

返回值：void

函数功能：移动this结点的首元素至recipient结点的末尾。用于合并后size很满，不宜merge只能重分配的情形。

函数设计思想：获取this的父节点，并清除父节点中指向该节点的键值对；然后调用CopyNForm方法将this节点合并到recipient节点中；调用memmove使得数组整体前移，使得头元素发生变化。此时父结点指向该结点的元素key值需要调整，为新的无效元素的key值上移至该元素

**2.8.3 BPlusTreeLeafPage**的设计

本关卡任务主要是设计BPlusTreeLeafPage类，该类作为B+树的叶子结点类型，其继承于BPlusTreePage，提供B+树树叶结点的功能。对于叶子结点，它有序存储了m个键条目(key)和m个值条目(value)。此外，叶结点成员将在父类BPlusTree Page的基础上额外增加一个数组，用于存储数据记录信息以及next\_page\_id\_存储相邻叶结点的索引指针。任务主要内容是实现树叶结点的初始化、查找、分裂，合并以及重分配等算法，方法较多，大多数与内部结点类BPlusInternalPage类类似，唯一不同是该结点还需要考虑数据记录以及与兄弟结点之间链表关系的维护，这里较上面并无太大新意，不做赘述。

2.8.4 B+树索引：Insert

本关卡任务主要是实现B+树的插入功能以及B+树的迭代器。具体函数设计如下

。。。。。

。。。。。

# 3 课程总结

该课程实践的总体任务是在MySQL环境下，完成一系列SQL语言实训任务。共15个实训任务，所有任务中每一关卡均已完成，任务完成度100%。

实训任务1，2主要完成了数据库、表、完整性约束的创建与修改，即DDL语句的训练；实训任务3，4完成了利用SQL语句基于金融数据库场景下的查询操作，要求掌握SQL中DML查询语句的使用，培养了数据库的场景应用和实践能力。实训任务5则完成了数据更新操作，涉及SQL中DML语句，包括数据的修改，插入，删除。实训任务6主要涉及到视图，了解视图机制，学习视图的创建与使用。实训任务7主要涉及存储过程与事务，学习变量、存储过程、事务的定义与使用，以及流程控制语句。实训任务8主要涵盖触发器知识点，进一步了解数据完整性。实训任务9主要设计函数的创建与使用。实训任务10主要知识点是数据库安全性控制，了解自主存取控制方法操作。实训任务11主要涉及并发带来的问题以及并发控制技术。实训任务12主要内容是数据库恢复技术，即备份与日志技术。实训任务13内容是数据库的设计与实现，从需求分析到概念设计到逻辑设计再到物理设计，体验数据库工程的设计过程与步骤。额实训任务14内容是数据库的应用开发，利用嵌入式SQL，这里是Java，即JDBC，对数据库进行应用开发。实训任务15则深入数据库内核，编写C++程序来实现简易B+树的存储结构。

通过此次课程，真的是收获颇丰。纸上学来终觉浅，在理论课上的学到的知识，偏抽象。而这们实践课能够将与理论课堂学习的知识联系，并从多维度，多方位，全知识点覆盖，使得学生能够将知识付诸实践，对知识的理解也更为深刻。至少我个人是这样的，在实践课前，对数据库理论知识比较模糊懵懂，当自己上手实操，开云见日，对数据库理解更深了。对于该门课程，我有一点小小的建议，就是B+树那一块，虽然每个函数注释都很详细，但是仍有一点难理解，需要一定的能力。个人建议可以添加视频或者图解来解释B+树的创建过程，结点的关系以及操作时的结构变换。还可以添加UML模型图，来交代每个类的信息。

# 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。

建议附录篇幅控制在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）。**

**上述光盘资料是按照学院的管理要求待课程结束后提交学院资料室归档保存的资料，以备将来学校或者教育部的学科检查、抽查使用，因此请同学们做好相关的提交工作。**

**插入**

**（1）通过查找确定叶子结点L；**

**（2）将排序后的关键字插入结点L ；**

**（3.1）如果空间充足（即结点L含有的关键字数目小于阶数），则插入结束；**

**（3.2）否则将结点L分裂为左右两个叶子结点（L和L2）：**

**重新分配关键字需要使其平衡，因此将结点中间的关键字进位到父结点中；**

**将指向结点L2的指针插入L的父结点中。**

**考虑重复键问题**

**删除**

**（1）从根结点出发找到该关键字所在结点L，删除该关键字；**

**（2.1）如果结点L的关键字数目不少于M/2，则删除完成；**

**（2.2）如果结点L仅有M/2-1个关键字数目，向兄弟结点借一个关键字（兄弟结点指与L有相同父结点的相邻结点）；**

**（2.3）否则将结点L与其兄弟结点合并，合并时需要删除父结点中的关键字（指向L或其兄弟结点）。**