2024年数据库系统原理实践任务书

# 课程介绍

“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，基础内容涉及以下几个部分，并可结合实际对DBMS原理的掌握情况向内核设计延伸：

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。

在实践进行过程中，大部分时间同学们会面对着这样的界面：

同学们可以尽情在实践环境中大展身手，以提升个人能力，当然也要尽最大努力去完成本次实践的所有关卡。如果学习过程中遇到问题，首先学会自己查资料解决，然后求助老师。

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>

实践中遇到平台故障或其他问题，请及时联系各班老师及助教。

# 实践环节

## 2.1 任务组成及选题要求

**登录头歌上的本课程平台后，将会看到总体任务由一系列实训任务构成，而进入每个实训后将会看到其由若干关卡组成，关卡依据其难易程度和工作量会有不同的分值。依据每个关卡布置的任务，在头歌界面输入相应答题代码后运行头歌的测评试功能，系统会自动依据运行结果评判该关卡是否通过，并记录相应分值。**

**本次课程，若所有关卡全部完成，将会获得头歌平台总分超过100分，然而课程考核并不要求所有关卡全部完成，大部分关卡同学们可依据自身兴趣和能力选择完成或者跳过不做（但对有些实训或者关卡则有具体的必做要求，本任务书接下来将会具体说明），最终程序检查满分只计头哥平台中的100分。**

**不允许整体跳过的实训（至少要完成该实训中的一个关卡任务）包括：前5个实训、第13个实训（MySQL-数据库设计与实现）。**

**此外，实训14“数据库应用开发(JAVA篇)”则规定最多跳过其中一个关卡。**

## 实训任务概览

### 实训1 数据库、表与完整性约束的定义(Create)

本实训主要考察对数据库对象或约束条件的创建语法的掌握情况，包括创建数据库、表基表的主码约束、外码约束、CHECK约束DEFAULT约束和UNIQUE约束，具体任务见任务内各个关卡。

注意头歌平台内每个关卡实践过程中包含了自测运行环境和测评环境，测评环境的重置在头歌平台内可能需要一定的时间。

### 实训2表结构与完整性约束的修改(ALTER)

本实训要求用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)。

同学们需要在每个关卡中在代码文件中完整给出代码。在代码文件中，最好不要使用use指定数据库，以免评测出现问题。

### 实训3 基于金融应用的数据查询(Select)

本实训采用的是某银行的一个金融场景应用的模拟数据库，测试库中有已有相应测试数据，请依据关卡任务需求完成相应查询动作。数据库中表，表结构以及所有字段的说明如下：

表1 client(客户表)

| **字段名称** | **数据类型** | **约束** | **说明** |
| --- | --- | --- | --- |
| c\_id | INTEGER | PRIMARY KEY | 客户编号 |
| c\_name | VARCHAR(100) | NOT NULL | 客户名称 |
| c\_mail | CHAR(30) | UNIQUE | 客户邮箱 |
| c\_id\_card | CHAR(20) | UNIQUE NOT NULL | 客户身份证 |
| c\_phone | CHAR(20) | UNIQUE NOT NULL | 客户手机号 |
| c\_password | CHAR(20) | NOT NULL | 客户登录密码 |

表2 bank\_card(银行卡)

| **字段名称** | **数据类型** | **约束** | **说明** |
| --- | --- | --- | --- |
| b\_number | CHAR(30) | PRIMARY KEY | 银行卡号 |
| b\_type | CHAR(20) | 无 | 银行卡类型(储蓄卡/信用卡) |
| b\_c\_id | INTEGER | NOT NULL FOREIGN KEY | 所属客户编号,引用自client表的c\_id字段。 |
| b\_balance | NUMERIC(10,2) | NOT NULL | 余额,信用卡余额系指已透支的金额 |

说明：银行卡类型只有”储蓄卡”或”信用卡”两种取值。对于b\_balance列，如果b\_type, b\_balance取值为(”储蓄卡”,10000)，表示这张储蓄卡内有10000的储蓄余额；而b\_type, b\_balance取值为(”信用卡”,10000)，表示这张信用卡已经透支10000元。

表3 finances\_product(理财产品表)

| **字段名称** | **数据类型** | **约束** | **说明** |
| --- | --- | --- | --- |
| p\_name | VARCHAR(100) | NOT NULL | 产品名称 |
| p\_id | INTEGER | PRIMARY KEY | 产品编号 |
| p\_description | VARCHAR(4000) | 无 | 产品描述 |
| p\_amount | INTEGER | 无 | 购买金额 |
| p\_year | INTEGER | 无 | 理财年限 |

表4 insurance(保险表)

| **字段名称** | **数据类型** | **约束** | **说明** |
| --- | --- | --- | --- |
| i\_name | VARCHAR(100) | NOT NULL | 保险名称 |
| i\_id | INTEGER | PRIMARY KEY | 保险编号 |
| i\_amount | INTEGER | 无 | 保险金额 |
| i\_person | CHAR(20) | 无 | 适用人群 |
| i\_year | INTEGER | 无 | 保险年限 |
| i\_project | VARCHAR(200) | 无 | 保障项目 |

表5 fund(基金表)

| **字段名称** | **数据类型** | **约束** | **说明** |
| --- | --- | --- | --- |
| f\_name | VARCHAR(100) | NOT NULL | 基金名称 |
| f\_id | INTEGER | PRIMARY KEY | 基金编号 |
| f\_type | CHAR(20) | 无 | 基金类型 |
| f\_amount | INTEGER | 无 | 基金金额 |
| risk\_level | CHAR(20) | NOT NULL | 风险等级 |
| f\_manager | INTEGER | NOT NULL | 基金管理者 |

说明：以上3张表中的金额都指每购入一份所要花费的金额。

表6 property(资产表)

| **字段名称** | **数据类型** | **约束** | **说明** |
| --- | --- | --- | --- |
| pro\_id | INTEGER | PRIMARY KEY | 资产编号 |
| pro\_c\_id | INTEGER | NOT NULL | 客户编号  FOREIGN KEY |
| pro\_pif\_id | INTEGER | NOT NULL | 业务约束 |
| pro\_type | INTEGER | NOT NULL | 商品类型:1表示理财产品;2表示保险;3表示基金 |
| pro\_status | CHAR(20) | 无 | 商品状态 |
| pro\_quantity | INTEGER | 无 | 商品数量 |
| pro\_income | INTEGER | 无 | 商品收益 |
| pro\_purchase\_time | DATE | 无 | 购买时间 |

说明：1.商品状态只有”可用”或”冻结”两种取值。2.商品收益指的是本条资产记录所记录商品的总收益，例如f\_id, f\_amount为(1031, 10000), pro\_pif\_id, pro\_type, pro\_quantity, pro\_income为(1031,3,10,27000)表示这条资产记录购入10份1037号基金，总花费10\*10000=100000元，总收益为27000元。

### 实训4 数据查询(Select)-新增

本小节子任务仍然以第2.3子任务的数据库内容为背景，但内容与统计、相似性推荐相关。

### 实训5 数据的插入、修改与删除(Insert,Update,Delete)

### 实训6 视图

### 实训7 存储过程与事务

### 实训8 触发器

### 实训9 用户自定义函数

### 实训10 安全性控制

### 实训11 并发控制与事务的隔离级别

背景数据库有表ticket记录了航班余票数，其结构如下表所示：

| **列** | **类型** | **说明** |
| --- | --- | --- |
| flight\_no | char(6) | primary key |
| tickets | int | 余票数 |

有两个涉及该表的并发事务t1和t2，分别定义在t1.sql和t2.sql代码文件中。平台会让两个事务并发执行，请同学们通过修改代码文件来达到题目预期的并发执行效果。

### 实训12 备份+日志：介质故障与数据库恢复

设有居民人口登记数据库residents,请为该数据库做一次静态的(个人独享服务器)海量逻辑备份，备份文件命名为residents\_bak.sql。 然后再用该逻辑备份文件恢复数据库。

注意：请仔细阅读平台上给出的注意部分，以免命令行操作中出现问题。

### 实训13数据库设计与实现

#### 从概念模型到MySQL实现

任务要求详见平台编程要求，此处给出E-R图方便翻阅。



#### 从需求分析到逻辑模型

设计一个影院管理系统。影院对当前的放映厅和电影进行排片，顾客到来后，可以购买任一排场的电影票，进入对应放映厅观看。系统中有以下实体集：

电影(movie)：属性有标识号(movie\_ID)、电影名(title)、类型(type)、时长(runtime)、首映日期(release\_date)、导演姓名(director)、主演姓名(starring)。

顾客(customer)：属性有标识号(c\_ID)、姓名(name)、手机号(phone)。

放映厅(hall)：属性有标识号(hall\_ID)、放映模式(mode)、容纳人数(capacity)、位置(location)。

排场(schedule)：属性有标识号(schedule\_ID)、日期(date)、时间(time)、票价(price)、票数(number)。

电影票(ticket)：属性有标识号(ticket\_ID)、座位号(seat\_num)。

实体间的关系描述如下：

①. 顾客和电影票有一对多的购买关系。每位顾客可以买多张电影票，每张电影票被一位顾客购买。

②. 电影票和排场有多对一的属于关系。一张电影票只属于一个排场，一个排场有多张电影票。

③. 排场和电影有一对多的放映关系。每个排场放一部电影，每部电影可以在多个排场放映。

④. 排场和放映厅有一对多的位于关系。每个排场位于一个放映厅，每个放映厅可以安排多个排场。

请按照上述要求完成：

1.画出E-R图。

2.给出对应的关系模式。

在平台文本编辑器内给出E-R图url和关系模式。

强烈建议同学们在提交前检查一次自己提交的url是否可用，有些图片链接点击可用是由于在本地上有正确的cookie。同学们可以将链接发送到手机上，在手机上尝试是否可以查看图片来确认。

互联网上图片都以图链形式传送，在web中右键点击看到的图片，菜单中大部分时间都会有复制图片链接的选项，在本关中需要同学们来提供一个这样的url链接，在网络上定位到E-R图图片。

以下给出4种创建图链的方式，当然还有更多其他方式可以达到目的，同学们可以自己选择其一。

1. git操作示意

①登录gitee或注册一个gitee账号。

②如果账号下还没有仓库，按下边操作创建一个可用的仓库，如果已有跳转到3.



或在右上角



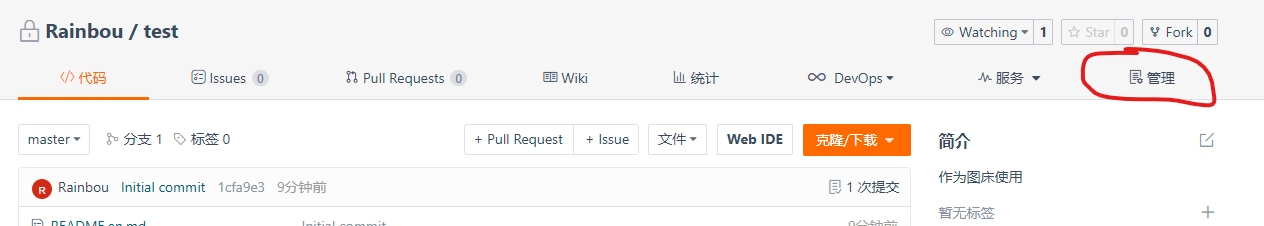
在创建仓库时可以不指定初始化仓库，勾选设置模板中的Readme文件即可。



如果没有勾选设置模板，面对一个原始的仓库，也可以点击这里来达到同样目的。



点击管理来把仓库设置为开源（只需要在第一个设置页下拉即可）。

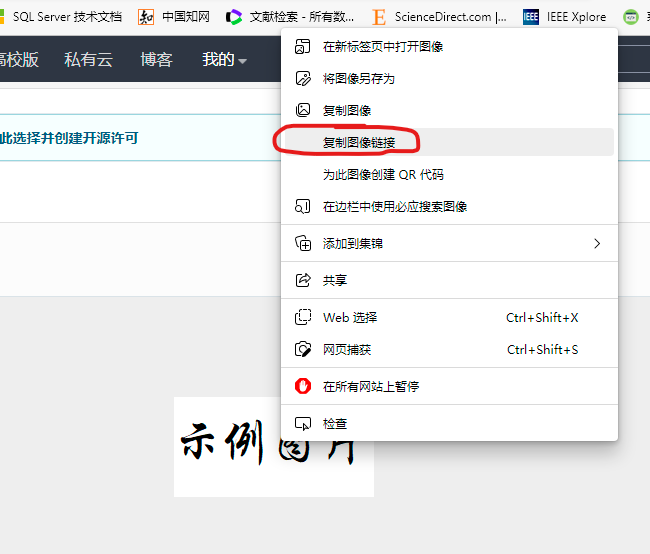




③回到仓库首页，来上传E-R图图形文件。



④点击图形文件查看，并在图片上右键选择复制链接，接下来就可以把url粘贴到平台上进行提交。



例如：<https://gitee.com/rainbou_1/test/raw/master/2.png>

1. 路过图床

①<https://imgtu.com/> 点击链接进入图床站。

②上传图片。

注：支持上传jpg、png、gif格式图片。

③下划到图片链接区域，点击复制url链接，在平台提交。



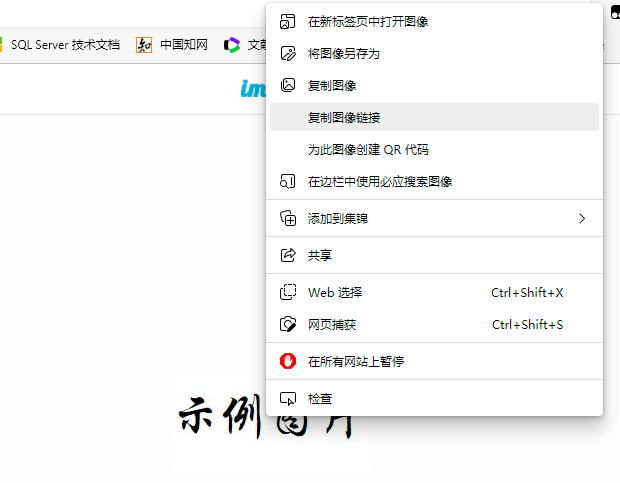
例如：<https://s1.ax1x.com/2022/03/29/qsLNDg.jpg>

1. imgbb图床

①<https://imgbb.com/> 点击链接进入图床站.

②上传图片，上传成功后点击图片进入查看大图界面。

③图片上右键选择复制链接，接下来就可以把url粘贴到平台上进行提交。

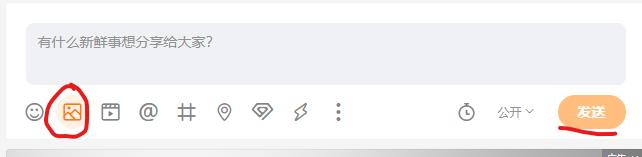


例如：<https://i.ibb.co/KjvsVC2/2.jpg>

1. 新浪微博

①登录微博 <https://weibo.com/>

②创建一条图片微博。



③点击发出的微博查看大图，并右键复制链接，粘贴到平台上提交。



例如：<https://wx4.sinaimg.cn/mw2000/0077zLsRly1h0qiguzgwqj305k02s3ye.jpg>

#### 建模工具的使用

下载资料包中的rbac.mwb，利用MySQL Workbench建模模块中的Forward engineering功能，自动导出SQL脚本，把脚本粘贴到代码文件中。

### 实训14 数据库应用开发(JAVA篇)（最多跳过一题）

#### JDBC体系结构和简单的查询

正确使用JDBC，查询金融应用场景数据库finance的client表(客户表)中邮箱不为空的客户信息，列出客户姓名，邮箱和电话.

输出格式要求：

姓名 邮箱 电话

夏雅惠 [57433144004@qq.com](mailto:57433144004@qq.com) 18962433158

钟庭玮 [59434199077@163.com](mailto:59434199077@163.com) 18110434192

其中，标题以及字段值之间用制表符隔开。第1列和第2列间用一个制表符，第2列和第3列间，标题用4个制表符，字段值用两个制表符隔开。

#### 用户登录

编程体验客户登录功能.程序先后提示客户输用户名和密码:

请输入用户名：

请输入密码：

客户的邮箱(c\_mail)充当用户名,而不是编号(c\_id).通常邮箱更容易记住.

根据客户的输入,输出以下两类信息之一:

登录成功。

用户名或密码错误！

#### 添加新客户

补充代码，实现向client表插入客户信息的方法，返回插入的行数。

其他要求详见平台。

#### 银行卡销户

补充代码，实现向银行卡销号的方法，只要客户编号和银行卡号匹配，即从bank\_card表中删除该银行卡，返回被删除的行数。

#### 客户修改密码

补充代码，实现修改密码的方法。客户修改密码通常需要确认客户身份，即客户需提供用户名(以邮箱为用户名)和密码，方法返回一个整数：1 - 密码修改成功；2 - 用户不存在；3 - 密码不正确；-1 - 程序异常(如没能连接到数据库等）。

#### 事务与转账操作

补充代码，实现一个银行卡转账的方法，方法返回boolean值，true表示转帐成功，false表示转账失败，并不需要细分或解释失败的原因。

下列任一情形都不可转账(转账失败的原因)：

转出或转入帐号不存在

转出账号是信用卡

转出帐号余额不足

#### 把稀疏表格转为键值对存储

一个表有很多列，但只有少数列存储有值，这样的表称为稀疏表。举个例子，大学里有几千名学生和几百门课程，为记录学生的成绩，设计这样一张表：每门课程占一列，每个学生占一行。但是，绝大多数学生只会选修这几百门课程中的一小部分，因此，整个表只会稀疏地存储了一些数据，这将产生大量的闲置空间。

一种更有效的方法是以(学号, 课程，成绩)这样的三元组形式存储。

设有高考成绩登记表entrance\_exam,其结构如下：

| **列名** | **类型** | **说明** |
| --- | --- | --- |
| sno | int | 学号，主码 |
| chinese | int | 语文 |
| math | int | 数学 |
| English | int | 英语 |
| physics | int | 物理 |
| chemistry | int | 化学 |
| biology | int | 生物 |
| history | int | 历史 |
| geography | int | 地理 |
| politics | int | 政治 |

转存表sc结构如下：

| **列名** | **类型** | **说明** |
| --- | --- | --- |
| sno | int | 学号 |
| col\_name | varchar(50) | 列名 |
| col\_value | varchar(50) | 列值 |

补充代码，完成转存任务。sc表初始为空表, 程序依前述规则将entrance\_exam表的值转写到sc表。对每一行，请从左至右依次考察每一列，转存非空列。

### 实训15 存储管理(Storage Manager)

数据在磁盘⽂件中是按照⻚⾯（Page）形式组织的。为避免直接访问磁盘数据⻚⾯⽽造成⾼昂的I/O开销，存储⼦系统在内存中创建缓冲池（Buffer Pool）来缓存部分磁盘数据⻚⾯。缓冲池维护固定数量的内存⻚⾯，每个内存⻚⾯称为“帧”（Frame），⼀般情况下，每⼀帧的⼤⼩与磁盘数据⻚⾯的⼤⼩保持⼀致。受制于内存容量，缓冲池只能缓存部分数据⻚⾯。因此，缓冲池管理的⽬标，就是在受限缓冲池⼤⼩的前提下，设计合适的内外存⻚⾯调度策略，尽可能将经常访问的磁盘数据⻚⾯维护在缓冲池中，从⽽减少磁盘I/O开销。 本实验涉及缓冲池管理的重要内容，学⽣需要实现数据库存储系统中的缓冲池管理器，即BufferPoolManager类。它的数据结构包括Page 、DiskManager 、Replacer 类的对象等。

数据库表中的⼀⾏数据，称为元组（Tuple）或者记录（Record），每条记录由多个字段（Field）组成。DBMS存储系统负责将其解释成属性类型和值。记录可以按行存储，亦可按列存储，本实验采用行存储记录。

本实验有四个⼦任务：Disk Manager 、Page Replacer 、Buffer Pool Manager和Record Manger。

### 实训16 索引管理（Indexing）

索引是实现快速存取的手段。本实验要求学⽣需要实现存储系统中的索引管理器，它主要由IxManager 类、IxIndexHandle类、IxNodeHandle 类、IxScan 类组成。本实验将索引的底层数据结构选取为B+树。 IxManager 类提供了创建/打开/关闭/删除索引⽂件的接⼝，其内部实现调⽤了实验⼀实现的DiskManager和BufferPoolManager 类的接⼝。IxIndexHandle 类⽤于实现B+树的基本功能，且⽀持并发。每个IxIndexHandle 对应⼀个索引⽂件，当IxManager 执⾏打开⽂件操作时，便会创建⼀个指向IxIndexHandle 的指针。 IxNodeHandle 类⽤于实现B+树的单个结点的基本功能，⽅便IxIndexHandle 类进⾏调⽤。 IxScan 类⽤于遍历叶⼦结点。 其中，同学们只要实现IxIndexHandle 、IxNodeHandle 中的接⼝。已提供其他类的完整源码。

# 撰写课程实践报告

格式参见实践报告模板。

# 成绩占比说明

**实践课综合成绩**

**=实践课程序头歌成绩（占70%）+检查时现场陈述与回答问题（占15%）+实践报告（占15%）**。