**北京邮电大学软件学院**

**2019-2020学年第2学期实验报告**

**课程名称： 数据库系统原理课程设计\_\_\_\_\_\_**

**实验名称：**  实验四 数据库模式的设计

**实验完成人：**

**姓名： 王衔飞 学号：**

**指导教师：**\_\_\_\_\_\_\_\_\_ **刁婷** \_\_\_\_\_\_\_\_\_\_\_

**日 期： 2019 年 5 月 20 日**

1. **实验目的**

1) 了解 E-R 图的基本概念和根据数据需求描述抽象出 E-R 图并将其转换为数据 库逻辑模式进而实现数据库中的表和视图。

2) 通过进行数据库表的建立操作，熟悉并掌握 Power Designer 数据库表的建立方法，理解关系数据库表的结构，巩固 SQL 标准中关于数据库表的建立语句。

3) 通过对 Power Designer 中建立、维护视图的实验，熟悉 Power Designer 中对视图的操作方法和途径，理解和掌握视图的概念。

1. **实验内容**

针对以下需求信息，尽可能全面地给出各个实体的属性和实体之间的联系。

在线考试系统需求信息如下：

在线考试系统是关于一门课程的授课教师安排自己的学生在线参加各种考试的应用，如阶段性考试，期中考试和期末考试等。在线考试系统要求有用户的 登录和登出。在线考试系统主要包括用户管理、试题管理、试卷管理和考试管理功能。需要实现教师输入试题，从试题生成试卷；学生参加考试获取试卷，提交答案和给出考试成绩等主要逻辑功能。

系统的用户包括教师、学生角色，一个用户有且只有一种角色。

鉴于在线考试的客观条件限制，试题完全采用单项选择形式。试题有所属知识点、内容、分值、备选答案项和唯一正确答案等属性组成。课程的知识点是确 定的，可以扩展，一道试题只能考察一个知识点。

教师录入各种试题构成题库，并根据考察的知识点不同生成试卷，相同知识

点的试题只能在一张试卷中出现一次，试卷由试卷标题和一定数量（即知识点的数量）的试题组成。试卷生成后，教师指定某次考试使用的试卷（唯一），学生 参加考试使用统一的试卷（试卷可能被不同时间的考试重复使用），考试信息还 包含考试标题、任教老师（唯一）、考试时间。

学生登录后，可以参加考试并在提交答案后立刻得到自己的考试成绩，也可以查看自己的考试历史记录。教师登录后可以查看学生的成绩。

1) 画出 E-R 图；

2) 将 E-R 图输入 Power Designer 行成概念模型；

3) 将概念模型转化为数据库逻辑模型；

4) 成功后，查看生成的表和视图的情况；

5) 尝试导出数据库的表和视图的脚本（.sql）。

1. **实验环境**

MySQL 8.0.19 PowerDesigner16.5

1. **实验过程描述**

确定需求、实体及属性：

* + - 1. **Question**：问题

**属性：**questionNo问题号，knowledgeId知识点号，content题目，point分数, allAnswers所有答案，rightAnswers正确答案

* + - 1. **Knowledge**:知识点

**属性：**knowledgeId知识点号,knowledgeContent知识点内容

* + - 1. User：用户

属性：userID用户号，type类别，name姓名，pw密码

* + - 1. Paper：试卷

属性：paperID试卷号，title标题

* + - 1. Exam：考试

属性：examID考试号，title考试标题，teacherUID任课教师，time考试时间

画出ER图：

电脑萤幕的截图

描述已自动生成

生成逻辑模型：

电脑萤幕的截图

描述已自动生成

生成物理模型并导出SQL

电脑萤幕的截图

描述已自动生成

导出的SQL：

/\*==============================================================\*/

/\* DBMS name: MySQL 5.0 \*/

/\* Created on: 2020/6/4 21:13:24 \*/

/\*==============================================================\*/

drop table if exists Exam;

drop table if exists Knowledge;

drop table if exists Paper;

drop table if exists Questions;

drop table if exists User;

drop table if exists attend;

drop table if exists has;

/\*==============================================================\*/

/\* Table: Exam \*/

/\*==============================================================\*/

create table Exam

(

examID int not null,

teacherUID int,

time datetime,

examTitle varchar(1024)

);

/\*==============================================================\*/

/\* Table: Knowledge \*/

/\*==============================================================\*/

create table Knowledge

(

knowledgeId int not null,

knowledgeContent varchar(1024)

);

/\*==============================================================\*/

/\* Table: Paper \*/

/\*==============================================================\*/

create table Paper

(

paperID int not null,

paperTitle varchar(1024)

);

/\*==============================================================\*/

/\* Table: Questions \*/

/\*==============================================================\*/

create table Questions

(

questionNo int not null,

knowledgeId int,

content varchar(1024),

rightAnswers varchar(1024),

allAnswers varchar(1024),

score int

);

/\*==============================================================\*/

/\* Table: User \*/

/\*==============================================================\*/

create table User

(

userID int not null,

type bool,

name varchar(1024),

pw varchar(1024)

);

/\*==============================================================\*/

/\* Table: attend \*/

/\*==============================================================\*/

create table attend

(

userID int not null,

examID int not null

);

/\*==============================================================\*/

/\* Table: has \*/

/\*==============================================================\*/

create table has

(

questionNo int not null,

paperID int not null

);

alter table Exam add constraint FK\_use foreign key ()

references Paper on delete restrict on update restrict;

alter table Questions add constraint FK\_ask foreign key ()

references Knowledge on delete restrict on update restrict;

alter table attend add constraint FK\_attend foreign key ()

references Exam on delete restrict on update restrict;

alter table attend add constraint FK\_attend2 foreign key ()

references User on delete restrict on update restrict;

alter table has add constraint FK\_has foreign key ()

references Questions on delete restrict on update restrict;

alter table has add constraint FK\_has2 foreign key ()

references Paper on delete restrict on update restrict;

1. **实验结果**

完成试验任务。

心得：通过本次实验，学习到了如何绘制er模型，然后如何使用Power Designer一步一步绘制概念模型物理模型及生成SQL语句。