**数据库实验报告**

姓名：桑龙龙

学号：20030540015

目录

[实验1-3大学及公司练习 3](#_Toc59825021)

[实验4-旅游模式的设计 3](#_Toc59825022)

[1、概述 3](#_Toc59825023)

[2、数据库设计 3](#_Toc59825024)

[2.1概念设计 3](#_Toc59825025)

[2.2逻辑设计 5](#_Toc59825026)

[2.3 事务设计 5](#_Toc59825027)

[3、编程实现 7](#_Toc59825028)

[3.1开发环境 7](#_Toc59825029)

[3.2、实现技术 7](#_Toc59825030)

[3.2.1连接数据库 7](#_Toc59825031)

[3.2.2查询调用 8](#_Toc59825032)

[3.2.3前端示范 9](#_Toc59825033)

[4、附录 9](#_Toc59825034)

[实验5-学生学籍管理 9](#_Toc59825035)

[1、概述 9](#_Toc59825036)

[2、数据库设计 10](#_Toc59825037)

[2.1、概念设计 10](#_Toc59825038)

[2.2、逻辑设计 11](#_Toc59825039)

[2.1、事务设计 11](#_Toc59825040)

[3、附录 12](#_Toc59825041)

# 实验1-3大学及公司练习

实验1、2、3进行了书本中大学和公司两个习题的练习，Mysql代码呈现在附录中。

**附录**

/university/build.sql创建大学的数据库源码

/university/sqlQuery.sql书上大学相关习题的解答源码

/company/build.sql 创建公司的数据库源码

/company/sqlQuery.sql 书上公司相关习题的解答源码

# 实验4-旅游模式的设计

## 1、概述

本文基于实验要求，需要设计并实现一个简单的旅行预订系统。该系统涉及的信息有航班、大巴班车、宾馆房间和客户数据等信息。

在本次实验中，完成了数据库的概念设计、逻辑设计、事务设计和编程实现。

## 2、数据库设计

本文涉及数据库相关部分的设计，完成了从概念设计、逻辑设计到事务设计的数据库系统设计全过程。

### 2.1概念设计

概念设计是指通过对用户需求进行综合、归纳与抽象，形成一个独立于具体DBMS的概念模型。

本次设计中有4种实体，顾客、航班、宾馆和巴士，其中顾客实体的属性有顾客的独有id、姓名、年龄、密码和性别这4个属性，航班实体有航班id、价格、座位数、可提供座位数、起始城市和目的城市6个属性，宾馆实体有所在城市、房间数、可提供房间数3个属性，巴士实体有所在城市、巴士数、可提供座位数这3个属性，其局部E-R图和整体E-R图如下。

|  |  |
| --- | --- |
| C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\5U~RM}967U2[$8@8`_193MU.png  Customer的E-R图 | C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\Q_$SL(OFK4@TCY6ZC4GR0}F.png  Bus的E-R图 |
| C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\S8MT~W`NJ5)YIE46RS~XWXR.png  Flight的E-R图 | C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\RGI%]ZO@B~NUMC[R@AVZQB5.png  Hotel的E-R图 |
| 整体E-R图 | |

### 2.2逻辑设计

逻辑结构设计是将概念结构转换为某个DBMS所支持的数据模型，并将进行优化。

根据图中的E-R图设计，根据转换规则，得到如下关系模式。

Flights (flight\_id, price, num\_seat, num\_avail, from\_city, to\_city)

hotels (city, price, num\_room, num\_avail)

bus (city, price,num\_bus, num\_avail)

customers (customer\_id, name, password, age, gender)

reservation (reser\_id , customer\_id, reser\_typ, from\_city, to\_city)

### 2.3 事务设计

针对本次实验数据库模式及业务需求，设计了如下事务：

1. **航班信息入库**

create procedure insertIntoFlights(in a int,in b double,in c int,in d int,in e int,in f int)

a，b，c，d，e，f分别对应flights表中的flight\_id，price，num\_seat，num\_avail，from\_city，to\_city 6个属性。通过存储过程insertIntoFlights可将该行插入到flights表中

1. **航班信息更新**

create procedure updateFlights(in a int,in b double,in c int,in d int,in e int,in f int)

表示将flights表中的flight\_id=a的行中的数据（price，num\_seat，num\_avail，from\_city，to\_city）这5个属性更新为（b，c，d，e，f）。

1. **巴士信息入库**

create procedure insertIntoBus(in a int,in b double,in c int,in d int)

a，b，c，d分别对应flights表中的city，price，num\_bus，num\_avail 4个属性。通过存储过程insertIntoBus可将该行插入到bus表中。

1. **巴士信息更新**

create procedure updateBus(in a int,in b double,in c int,in d int)

表示将bus表中的city=a的行中的数据(price，num\_bus，num\_avail)这3个属性更新为（b，c，d）。

1. **宾馆信息入库**

create procedure insertIntoHotel(in a int,in b double,in c int,in d int)

a，b，c，d分别对应hotels表中的city，price，num\_room，num\_avail 4个属性。通过存储过程insertIntoHotel可将该行插入到hotels表中。

1. **宾馆信息更新**

create procedure updateHotel(in a int,in b double,in c int,in d int)

表示将hotels表中的city=a的行中的数据(price，num\_room，num\_avail)这3个属性更新为（b，c，d）。

1. **客户信息注册（入库）**

create procedure insertIntoCustomers(in a int,in b varchar(32),in c char(32),in d int,in e char(1))

a，b，c，d，e分别对应customers表中的customer\_id，name，password，age，gender 5个属性。通过存储过程insertIntoCustomers可将该行插入到customers表中

1. **客户信息更新**

create procedure updateCustomers(in a int,in b varchar(32),in c char(32),in d int,in e char(1))

表示将customers表中的customer\_id=a的行中的数据(name，password，age，gender)这4个属性更新为（b，c，d，e）。

1. **预定航班**

create procedure bookFlight(in id int,in fromCity int,in toCity int)

id为想要预定航班的客户的customer\_id，fromCity，toCity表明id客户想要预定从fromCity到toCity的航班。

1. **预订房间**

create procedure bookHotel(in id int,in tarCity int)

id为想要预定房间的客户的customer\_id，tarCity表明id客户想要预定从tarCity的房间。

1. **预定巴士**

create procedure bookBus(in id int,in tarCity int)

id为想要预定巴士的客户的customer\_id，tarCity表明id客户想要预定从tarCity的巴士。

1. **查询航班**

create procedure queryFlight(in city int)

city为用户查询的航班起始城市或者航班终点，将会展示出所有的可能航班，按照航班剩余的座位数量降序排列。

1. **查询房间**

create procedure queryHotel(in tarCity int)

tarCity为用户查询的城市，将会展示城市tarCity的宾馆信息

1. **查询巴士**

create procedure queryBus(in tarCity int)

tarCity为用户查询的城市，将会展示城市tarCity的巴士信息

1. **查询客户预订信息**

create procedure queryReservation(in id int)

id为待查询客户的customer\_id，将会返回客户id的所有预订信息，包含航班，房间和巴士。

1. **查询客户路线**

create procedure queryRoute(in id int)

id为待查询客户的customer\_id，将会返回客户id的途径城市

## 3、编程实现

针对本次实验的旅游模式数据库模式及业务需求，编程实现了部分前台功能，可以进行数据插入以及数据查询。

### 3.1开发环境

后台的数据库开发基于Mysql，版本为Server version: 5.7.32-0ubuntu0.18.04.1 (Ubuntu)，操作系统的版本为Ubuntu 7.3.0-16ubuntu3。MySQL是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。

前端应用通过C++进行编写，命令行模式进行交互，实现基础的数据的插入，数据的展示，通过Mysql提供的mysql/mysql.h用于连接及操控数据库。

### 3.2、实现技术

#### 3.2.1连接数据库

以下代码展示如何连接数据库。

1. **const** **char**\* user="root";//your mysql user name
2. **const** **char**\* password="xxx";//your mysql password
3. **const** **char**\* host="localhost";//your host
4. **const** **char**\* database="tour";//your database name
5. MYSQL my\_connect;//connect interaface
6. **bool** connect(){
7. //function to connect to your database
8. mysql\_init(&my\_connext);
9. **if** (mysql\_real\_connect(&my\_connect, host, user, pwd, database, port, NULL, 0)){
10. printf("Connect\n");
11. **return** **true**;
12. }**else**{
13. printf("Connext failed\n");
14. **return** **false**;
15. }
16. }

#### 3.2.2查询调用

以下代码展示如何通过C++进行一次查询，command是要运行的Mysql的命令。

1. **void** query(**const** **char**\* command){
2. //command:mysql query sentence
3. MYSQL\_RES\* res;
4. MYSQL\_FIELD\* col;
5. MYSQL\_ROW row;
6. **int** r,c,res\_num;
7. res\_num=mysql\_query(&my\_connect,(**const** **char**\*)command);
8. //running the query sentence of command
9. **if**(res\_num!=0){
10. //no fetch any result
11. printf("This query failed, maybe not exisit any data\n");
12. **return**;
13. }**else**{
14. res=mysql\_store\_result(&my\_connect);
15. //get the query result
16. **if**(res){
17. //print result
18. **while**(row=mysql\_fetch\_row(res)){
19. cout<<row[1]<<endl;
20. }
21. }
22. }
23. }

#### 3.2.3前端示范

|  |  |
| --- | --- |
| C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\S}(E8D`NESH1%8NPB4BE`~B.png  主界面 | C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\5Z9NWLFL0(_VK3E25HEET7V.png  数据存储界面 |
| C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\J$FG_BC37XUVDUM8}TD)6EL.png  数据插入界面 | C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\8H%(TWA@8@5`$1T[KY@EN[S.png  用户信息查询界面 |

## 4、附录

/tour/build.sql 为创建本实验即tour数据库的sql语句

/tour/sqlQuery.sql 为本实验中所用到的存储过程

/tour/manipulate.cpp 为操纵数据库的前端源文件，该文件在Linux系统下编写，Windows系统不可使用。

/tour/E-R.png 为本实验所用到的整体E-R图

/tour/entity.pdf 为本实验中实体的E-R图

# 实验5-学生学籍管理

## 1、概述

本文基于实验要求，需要设计并实现一个简单的学生学籍管理系统，学籍系统需要记录学生上过的课程信息，以及课程的相关信息，最终需要能够进行一些简单的查询。

在本次实验中，完成了数据库的概念设计、逻辑设计、事务设计。

## 2、数据库设计

本文涉及数据库相关部分的设计，完成了从概念设计、逻辑设计到事务设计的数据库系统设计全过程。

### 2.1、概念设计

本实验中共有6个实体，其中课程实体具有课程id、课程名、专业id、类别、学年、学分、学期这7个属性，学生实体具有学生id、姓名、年龄、性别这4个实体，学生实体通过学习课程与课程实体产生关系，教师实体有教师id、姓名、年龄等数个属性，教师实体需要具体的教授某门课程从而与课程实体产生关系，从而也间接的与学生之间存在关系，除此之外还有专业、班级、系这三个实体。局部E-R图与整体E-R图如下。

|  |  |
| --- | --- |
| C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\CB)IEEB8C}CGL252]27@IHX.png  课程E-R图 | C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\OIU}4)NUMR8AKQC9UV0S])D.png  学生E-R图 |
| C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\4_KLT3(3QWFY50XV7_`$PPA.png  教师E-R图 | C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\QN}U`B6MH[FG)DX`T3MWIX8.png  专业E-R图 |
| C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\4C2F02J2EZ_]85NO76%@OKE.png  班级E-R图 | C:\Users\Administrator\AppData\Roaming\Tencent\Users\630987814\QQ\WinTemp\RichOle\$7CLDDHBDB[NVIYYQN@S[JK.png  系E-R图 |
| 整体E-R图 | |

### 2.2、逻辑设计

逻辑结构设计是将概念结构转换为某个DBMS所支持的数据模型，并将进行优化。

根据图中的E-R图设计，根据转换规则，得到如下关系模式。

instructor (instructor\_id, age, name, gender, specialty\_name)

student (student\_id, name, age, gender, class\_id)

major (major\_name, specialty\_name)

class (class\_id, major\_name)

specialty (specialty\_name, builing, budget)

course (course\_id, instructor\_id, name, type, year, credit, semester)

takes (student\_id, grade, make\_up\_exam, course\_id, year, semester)

### 2.1、事务设计

1. **查询学生的课程以及平均成绩**

create procedure queryCouseLearned(in id int)

id为待查询的学生的student\_id，将会查询出该id的所有所上课程的得分情况，并且统计出必修课平均成绩和选修课平均成绩

1. **查询教授过某同学的所有老师**

create procedure queryInstructorTeached(in id int)

id为待查询学生的student\_id，将会查询过所有教授过该同学课程的老师

1. **查询待开除的学生**

create procedure queryShouldExpelStudent()

本次查询将会列出理应被开除的所有学生，如果一学期不及格的必修课学分超过10个，或者不及格必修课学分累计超过30个；或者不及格选修课学分累计超过20个。都被视为应当开除。

## 3、附录

/studentManagement/build.sql 创建本实验数据库的源文件

/studentManagement/sqlQuery.sql 本实验所用到的存储过程

/tour/E-R.jpg 为本实验所用到的整体E-R图

/tour/entity.pdf 为本实验中实体的E-R图