目录

[（一）设计思路 3](#_Toc132834514)

[（二）适用范围 3](#_Toc132834515)

[（三）代码 3](#_Toc132834516)

[1、删除原有数据库 4](#_Toc132834517)

[2、创建数据库 4](#_Toc132834518)

[3、创建数据表 4](#_Toc132834519)

[用户权限表 4](#_Toc132834520)

[学生信息表 4](#_Toc132834521)

[学生成绩表 5](#_Toc132834522)

[用户权限级别表 5](#_Toc132834523)

[4、插入数据 5](#_Toc132834524)

[插入用户权限表数据 5](#_Toc132834525)

[插入学生信息表随机数据 6](#_Toc132834526)

[插入学生成绩表随机数据 6](#_Toc132834527)

[5、查询数据 7](#_Toc132834528)

[6、设置用户权限 7](#_Toc132834529)

[删除原有权限用户 7](#_Toc132834530)

[创建用户 7](#_Toc132834531)

[赋用户权限 8](#_Toc132834532)

[7、数据打包（备份与还原） 8](#_Toc132834533)

[（四）图表关系图 9](#_Toc132834534)

[（五）Navicat演示 10](#_Toc132834535)

[1、用户权限级别表 10](#_Toc132834536)

[2、学生信息表 11](#_Toc132834537)

[3、学生成绩表 12](#_Toc132834538)

[4、用户权限表 12](#_Toc132834539)

# （一）设计思路

MySQL 数据库的建表和初始化代码，主要分为以下几个部分：

1、创建一个名为 `student\_db` 的数据库，并在之后使用该数据库。

2、创建了四个表：`user\_auth`、`student\_info`、`student\_score`、`auth\_level`，

分别代表用户权限表、学生信息表、学生成绩表、用户权限级别表。

3、在 `auth\_level` 表中插入四个权限级别的名称，用于区分用户的权限。

4、在 `user\_auth` 表中插入六个用户的信息，包括用户名、密码和权限级别。

5、在 `student\_info` 表中插入五个学生的基本信息，包括姓名、性别、生日、入学年份和专业。

6、在 `student\_score` 表中插入五个学生的成绩信息，包括学生 ID、课程名称和成绩。

7、接下来是针对这些用户的权限控制。先删除了已经存在的用户，然后创建六个用户，每个用户对应插入数据。

8、对创建的用户进行授权，如 `admin` 用户被授予了所有权限，`dept\_admin` 用户被授予了 `student\_db` 数据库的所有权限，其它用户被授予了不同的权限。

# （二）适用范围

因为数据库较小，适用范围比较小，适用于一些小型的学生信息管理系统。

# （三）代码

以下是代码实现：

## 1、删除原有数据库

/\*删除原有数据库（如果存在）\*/

drop database if exists student\_db;

## 2、创建数据库

/\*创建数据库\*/

create database student\_db;

use student\_db;

## 3、创建数据表

### 用户权限表

/\*1. 用户权限表

/\*该表用于存储用户信息以及其所拥有的权限级别。\*/

CREATE TABLE `user\_auth` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`username` varchar(50) NOT NULL,

`password` varchar(50) NOT NULL,

`auth\_level` int(11) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

### 学生信息表

/\*2. 学生信息表

/\*该表用于存储学生的基本信息。\*/

CREATE TABLE `student\_info` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(50) NOT NULL,

`gender` enum('male','female') NOT NULL,

`birthday` date NOT NULL,

`enroll\_year` int(11) NOT NULL,

`major` varchar(50) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

### 学生成绩表

/\*3. 学生成绩表

/\*该表用于存储学生的成绩信息。\*/

CREATE TABLE `student\_score` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`student\_id` int(11) NOT NULL,

`course\_name` varchar(50) NOT NULL,

`score` decimal(5,2) NOT NULL,

PRIMARY KEY (`id`),

KEY `fk\_student\_id` (`student\_id`),

CONSTRAINT `fk\_student\_id` FOREIGN KEY (`student\_id`) REFERENCES `student\_info` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

### 用户权限级别表

/\*4. 用户权限级别表

/\*该表用于存储不同权限级别的信息。\*/

CREATE TABLE `auth\_level` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(50) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `auth\_level` (`name`) VALUES

('系统管理员'),

('部门管理员'),

('部门操作员'),

('部门员工');

## 4、插入数据

### 插入用户权限表数据

/\* 1. 插入用户权限表随机数据 \*/

INSERT INTO user\_auth (username, password, auth\_level)

VALUES

('admin', 'admin123', 1),

('dept\_admin', 'deptadmin123', 2),

('dept\_operator', 'deptoperator123', 3),

('dept\_employee', 'deptemployee123', 3),

('user1', 'user1123', 4),

('user2', 'user2123', 4);

### 插入学生信息表随机数据

/\* 2. 插入学生信息表随机数据 \*/

INSERT INTO student\_info (name, gender, birthday, enroll\_year, major)

VALUES

('张三', 'male', '2000-01-01', 2019, '计算机科学与技术'),

('李四', 'male', '1999-05-01', 2018, '物联网工程'),

('王五', 'male', '2001-08-01', 2020, '电子信息工程'),

('赵六', 'female', '2002-04-01', 2021, '通信工程'),

('钱七', 'female', '1998-11-01', 2017, '软件工程');

### 插入学生成绩表随机数据

/\* 3. 插入学生成绩表随机数据 \*/

INSERT INTO student\_score (student\_id, course\_name, score)

VALUES

(1, '数据结构', 88.5),

(1, '算法设计', 91),

(1, '操作系统', 86.5),

(2, '数据结构', 75),

(2, '算法设计', 81.5),

(2, '操作系统', 78),

(3, '数据结构', 92),

(3, '算法设计', 90),

(3, '操作系统', 94),

(4, '数据结构', 85.5),

(4, '算法设计', 83),

(4, '操作系统', 87),

(5, '数据结构', 78),

(5, '算法设计', 85.5),

(5, '操作系统', 79),

(1, '高等数学', 89),

(2, '高等数学', 77),

(3, '高等数学', 95),

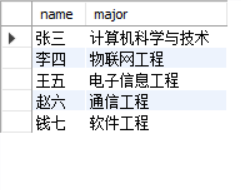
(4, '高等数学', 86),

(5, '高等数学', 80);

## 5、查询数据

/\*查询所有学生的姓名和所在专业：\*/

SELECT name, major FROM student\_info;



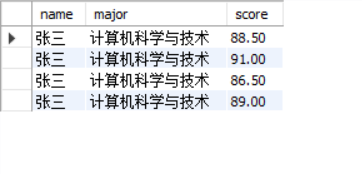
/\*查询某个学生的姓名、所在专业和成绩：\*/

SELECT si.name, si.major, ss.score

FROM student\_info si

LEFT JOIN student\_score ss ON si.id = ss.student\_id

WHERE si.id = 1;



## 6、设置用户权限

### 删除原有权限用户

/\*删除原有权限用户（如果存在）\*/

drop user if exists

`admin`@localhost,dept\_admin@localhost,

dept\_operator@localhost,dept\_employee@localhost,

user1@localhost,user2@localhost;

### 创建用户

/\*创建用户\*/

create user

'admin'@'localhost' identified by 'admin123',

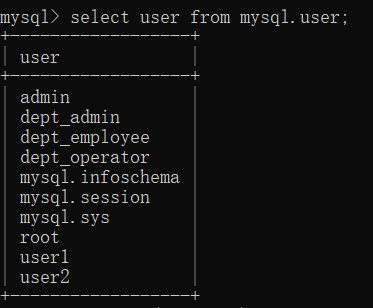
'dept\_admin'@'localhost' identified by 'deptadmin123',

'dept\_operator'@'localhost' identified by 'deptoperator123',

'dept\_employee'@'localhost' identified by 'deptemployee123',

'user1'@'localhost' identified by 'user1123',

'user2'@'localhost' identified by 'user2123';



### 赋用户权限

grant all on \* to `admin`@localhost;

grant all on student\_db.\* to dept\_admin@localhost;

grant create,alter,drop,select on student\_db.\* to dept\_operator@localhost;

grant create,alter,drop,select on student\_db.\* to dept\_employee@localhost;

grant select on student\_db.\* to user1@localhost;

grant select on student\_db.\* to user2@localhost;

create user ... 创建了六个用户，分别为 admin、dept\_admin、dept\_operator、dept\_employee、user1 和 user2。每个用户都有一个对应的密码（**密码为“账号名+123（有“\_”的删去掉）**“）。

赋予 admin 用户对所有数据库的所有权限。

赋予 dept\_admin 用户对 student\_db 数据库的所有权限。

赋予 dept\_operator 用户对 student\_db 数据库的创建、修改、删除和查询权限。

赋予 dept\_employee 用户对 student\_db 数据库的创建、修改、删除和查询权限。

赋予 user1 用户对 student\_db 数据库的查询权限。

赋予 user2 用户对 student\_db 数据库的查询权限。

## 7、数据打包（备份与还原）

/\*数据打包（备份）

/\*mysqldump -u [username] -p [database\_name] > [filename].sql\*/

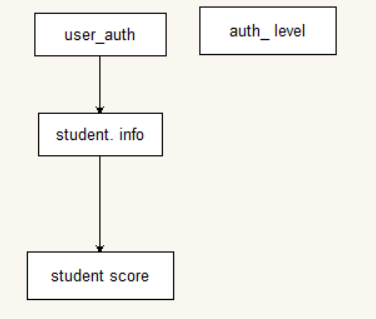
mysqldump -uadmin -padmin123 student\_db > C:/Users/PC/Desktop/学籍管理.sql

/\*数据还原\*/

mysql -uadmin -padmin123 student\_db < C:/Users/PC/Desktop/学籍管理.sql

注： 备份路径可自行更改

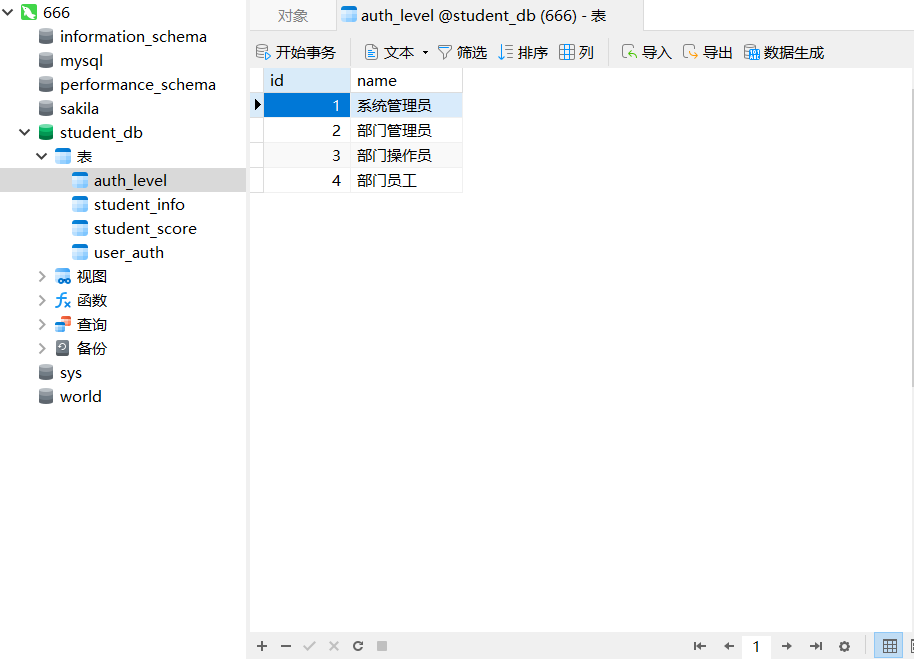
# （四）图表关系图



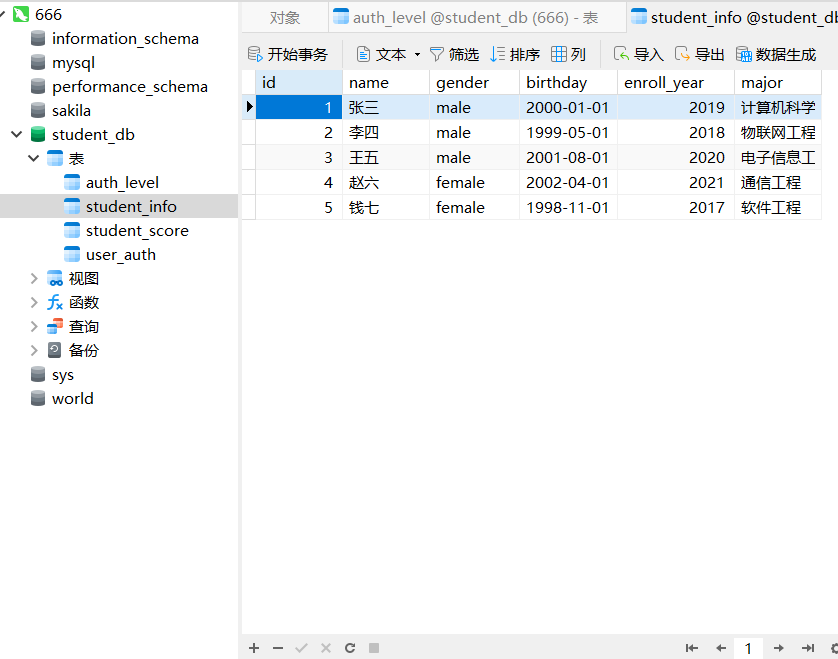
在这个数据库中，每个表都有一个主键（id），其用途是确保在表中的每一行都具有唯一的标识符。关系的表示方法包括在子表中的外键，其中子表的外键将引用父表的主键。例如，在student\_score表中，student\_id列是一个外键，引用student\_info表中的id。通过使用外键，我们可以轻松地将不同表中的信息联系在一起，并支持各种查询和操作，以便更好地组织和管理数据。

# （五）Navicat演示

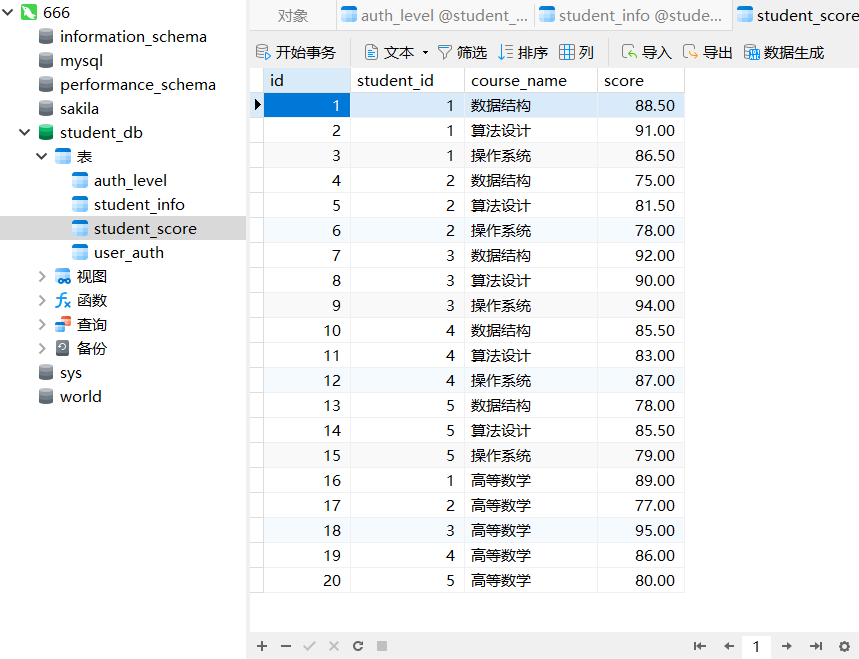
## 1、用户权限级别表



## 2、学生信息表



## 3、学生成绩表



## 4、用户权限表

