



MySQL® 数据库设计与应用

第3章 数据库、数据表、 记录、索引



第3章内容

● 数据库操作

● 记录操作-插入、修改、删除记录

● 数据类型

● 索引概述及操作

● 数据表操作-创建、查看等

● course管理系统实例

● 数据表操作-修改表



MySQL数据库设计与应用

数据表操作之

创建、查看、复制、改名及删除操作

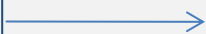


设计数据表结构

在数据库中，表是最重要、最基本的对象，是存储数据的基本单位。
数据表从哪里来呢？数据表由关系模式转换而来。但不是简单的转换。

(实体) 关系

关系名 (属性1,
属性2,)



表名

字段名1	字段名2	字段名n
值1	值1	值1
值2	值2	值2
.....
值n	值n	值n



teacher

Teacher_id	Teacher_name	age	Enter_time	professional
10101	潘多拉	35	1998-7-2	副教授
10102	吉米	27	2013-7-5	讲师
10103	郝本	45	1995-8-10	副教授



在设计表结构时要考虑下面几个方面：

- 字段名要通俗易懂且具有代表性，字段名不允许重复，为了编程方便，尽量使用英文名字。
- 字段类型选择的原则是：根据需求选择合适的字段类型，在满足需求的情况下字段类型尽可能小。不参加数学计算的数字定义为字符串型，如学号、电话号码等。
- 字段长度是指能容纳该字段值的最大数量，如学号为12位数字，长度设置为12；课程名可能包含10个汉字，长度就要设置为10。
- 字符集编码选择：含有中文字符时统一采用utf8或gbk类型，以避免乱码情况的发生。



Department

字段名	数据类型	长度	是否空值	是否主键外键	默认值	备注
Department_id	char	3	no	主键		学院编号
Department_name	varchar	10	no			学院名称



teacher

字段名	数据类型	长度	是否空值	是否主键外键	默认值	备注
Teacher_id	char	5	no	主键		教师编号
teacher_name	varchar	4	no			教师姓名
Department_id	char	3	no	外键		学院编号
gender	char	1	no		男	性别



course

字段名	数据类型	长度	是否空值	是否主键外键	默认值	备注
Course_id	char	4	no	主键		课程号
course_name	varchar	10	no			课程名
capacity	tinyint	4	no		60	人数上限
Teacher_id	char	5	no	外键		教师编号



Classes

字段名	数据类型	长度	是否空值	是否主键外键	默认值	备注
class_id	char	8	no	主键		班级编号
class_name	varchar	8	no			班级名
Year	smallint	4	no			年度
Department_id	char	3	no	外键		学院编号



students

字段名	数据类型	长度	是否空值	是否主键外键	默认值	备注
students_id	char	12	no	主键		学号
students_name	varchar	4	no			姓名
class_id	char	4	no	外键		班级编号
phone	varchar	18				电话



choose

字段名	数据类型	长度	是否空值	是否主键外键	默认值	备注
Choose_id	int	11	no	主键		auto_increment
student_id	char	12	no	外键		学号
Course_id	char	4	no	外键		课程号
choose_time	datetime		no			选课时间
report	int				0	成绩



»» 创建数据表

创建数据库表使用create table 语句。

语法格式如下：

```
create table 表名(  
    字段名1 数据类型 [约束条件],  
    字段名2 数据类型 [约束条件],  
    ...  
    [其他约束条件],  
    [其他约束条件]  
)其他选项;
```



在course数据库中创建department表

```
use course;
```

```
CREATE TABLE department(  
  Department_id char(3) NOT NULL PRIMARY KEY COMMENT '学院编号',  
  Department_name varchar(10) NOT NULL COMMENT '学院名称'  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



创建teacher表SQL命令

```
CREATE TABLE teacher(  
  Teacher_id char(5) NOT NULL PRIMARY KEY COMMENT '教师编号',  
  Teacher_Name varchar(4) NOT NULL COMMENT '教师姓名',  
  Department_id char(3) NOT NULL COMMENT '学院编号',  
  Gender char(1) NOT NULL DEFAULT '男' COMMENT '性别',  
  CONSTRAINT teacher_department FOREIGN KEY (Department_id)  
  REFERENCES department(Department_id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



创建classes表SQL命令

```
CREATE TABLE classes(  
  Classe_id char(8) NOT NULL PRIMARY KEY COMMENT '班级编号',  
  Class_name varchar(8) NOT NULL COMMENT '班级名',  
  Year smallint(4) NOT NULL COMMENT '年度',  
  Department_id char(3) NOT NULL COMMENT '学院编号',  
  CONSTRAINT class_department FOREIGN KEY (Department_id)  
  REFERENCES department (Department_id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```




创建student表SQL命令

```
CREATE TABLE student(  
  Student_id char(12) NOT NULL PRIMARY KEY COMMENT '学号',  
  Student_name varchar(4) NOT NULL COMMENT '姓名',  
  Class_id char(4) NOT NULL COMMENT '班级编号',  
  Phone varchar(18) NOT NULL COMMENT '电话',  
  CONSTRAINT student_class FOREIGN KEY (Class_id) REFERENCES  
  classes (Class_id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



创建course表SQL命令

```
CREATE TABLE course(  
  Course_id char(4) NOT NULL PRIMARY KEY COMMENT '课程号',  
  Course_name varchar(10) NOT NULL COMMENT '课程名',  
  Capacity tinyint(4) NOT NULL DEFAULT 60 COMMENT '人数上限',  
  Teacher_id char(5) NOT NULL COMMENT '教师编号',  
  CONSTRAINT course_teacher FOREIGN KEY (Teacher_id)  
  REFERENCES teacher (Teacher_id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



创建choose表SQL命令

```
CREATE TABLE choose(  
  Choose_id int(11) NOT NULL PRIMARY KEY  
  AUTO_INCREMENT,  
  Student_id char(12) NOT NULL COMMENT '学号',  
  Course_id char(4) NOT NULL COMMENT '课程号',  
  Choose_time datetime NOT NULL COMMENT '选课时间',  
  report int DEFAULT 0 COMMENT '成绩',  
  CONSTRAINT choose_course FOREIGN KEY (Course_id)  
  REFERENCES course (Course_id),  
  CONSTRAINT choose_student FOREIGN KEY (Student_id)  
  REFERENCES students (Student_id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



SQL语句中关键字做一些说明：

1. PRIMARY KEY用于定义主键。也可以使用多字段来定义主键。
2. COMMENT注释该字段的含义。
3. NOT NULL是非空约束。
4. DEFAULT为该字段加默认值，可以减少输入次数。
5. AUTO_INCREMENT为自增型属性，一般用作主键，数值会自动加1。
6. 当表中有外键字段时用CONSTRAINT设置外键。
7. ENGINE=InnoDB是设置该表的存储引擎，DEFAULT CHARSET=utf8是设置该表的默认字符集。



» 显示表结构，查看数据表

显示表结构的命令：

方式一：DESCRIBE/DESC 表名

- 使用DESCRIBE table_name可显示指定表的结构
- DESCRIBE也可用DESC替代

方式二：SHOW CREATE TABLE 表名 查看表的详细信息

查看数据列表命令：show tables;



»» 复制表结构

复制一个表结构有两种实现方法:

方法一：在create table语句的末尾添加like子句，可以将源表的表结构复制到新表中，语法格式如下：

create table 新表名 like 源表

方法二、在create table语句的末尾添加一个select语句，可以实现表结构和表记录的复制。语法格式如下：

create table 新表名 select * from 源表



»» 修改表名

修改表名的语法格式较为简单，语法格式如下：

`rename table 旧表名 to 新表名`

或者：

`alter table 旧表名 rename 新表名`



删除表

删除表的SQL语法格式为:

drop table 表名;

如果表之间存在外键约束关系，应先删除外键约束条件，再删除表，或者先删子表，再删父表。

注意：删除数据表的操作应该谨慎使用。一旦删除了数据表，表中的数据将会全部清除，没有备份则无法恢复。



»» 总结与思考

本节介绍了数据表的创建、查看、重命名、复制和删除操作的相关知识。在数据库开发中，数据库表占相当重的分量，合理的表结构会减少数据冗余，提高数据库的性能。

思考：你能为自己的联系人设计一个数据表结构吗？



信息时代，
数据织就未来

