**类型概述**

Mariadb支持的一些字段数据类型

|  |  |
| --- | --- |
| 类别 | 关键字 |
| 字符型 | 定长字符串CHAR，变长字符串varchar |
| 数值型 | 整形INT，浮点型FLOAT，双精度DOUBLE |
| 日期时间型 | 日期时间DATETIME,日期DATE，时间TIME |
| 枚举型 | 单选集ENUM，多选集SET |

**常用数据类型**

字符型

定长字符串CHAR

可指定为1-255字符

当字段的值不够指定长度时，会在右边用空格补齐

当字段的值超过指定长度时，将会被截断

变长字符串varchar

可指定为1-255字符，

当字段的值不够指定长度时，按实际字符数分配存储单元

当字段的值超过指定长度时，将会被截断

数值型

整数型INT

占4个字节，数值范围是-2的31次-2的31次-1

当存入的值与类型不匹配时，作为0处理

单精度浮点型FLOAT

占4个字节，数值范围是-3.40E+38~+3.40e+38

定义格式：float（总宽度，小数位数）

当存入的值与类型不匹配时，作为0处理

当存入的值超出范围时，仅保存最大/最小值

双精度浮点型DOUBLE

占用8个字节，数值范围为-1.7E+308~1.7E+308

定义格式：double（总宽度，小数位数）

当存入的值与类型不匹配时，作为0处理

当存入的值超出范围时，仅保留最大/最小值

日期时间型

日期时间DATETIME

占用8个字节，同时记录日期、时间

值范围：0000-01-01 00:00:00 ~9999-12-31 23:59:59

日期DATE

占用4个字节，只记录日期

值范围：0000-01-01~9999-12-31

时间TIME

占用3个字节，只记录时间

范围：00:00:00~23:59:59

**修改字段信息**

Alter table基本用法（alter table [库名.]表名 执行动作（modify修改字段类型、change修改字段名、add添加字段、drop删除字段））

修改字段类型：alter table 表名 modify 字段名 类型(宽度) [first|after 字段名]；（first作为表格的第一个字段，after 字段xx插入到xx字段后）



修改字段名称：alter table 表名 change 旧字段名 新字段名 类型（宽度）;



**添加/删除字段**

添加新字段：alter table 表名 add 字段名 类型(宽度) [first|alfter 字段名] default 默认值;



删除字段：alter table 表名 drop 字段名



修改表名称：alter table 表名 rename 新表名;



**字符集设置**

关于默认字符集

新表的默认字符集参照所在的库

新表的默认字符集参照mariadb服务的运行参数

取决于/etc/my.cnf中的character\_set\_server配置

默认为latin1（不支持中文字符）

表的字符集处理

新建表格时指定默认字符集：create table 表名(…) default charset 字符集名;

查看表格的默认字符集：show create table 表名;

修改现有表格的默认字符集：alter table 表名 default charset 字符集名;

转换现有表格中数据记录的字符集：alter table 表名 convert to charset 字符集名;

库的字符集处理

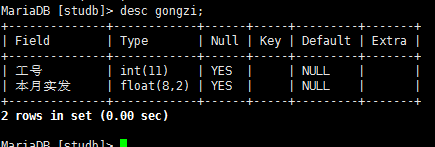
新建库时指定默认字符集：create database 库名 default charset 字符集名;

查看库的默认字符集：show create database 库名;

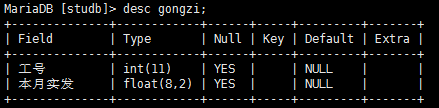
修改现有库的默认字符集：alter database 库名 default charset 字符集名；

**约束条件**

是否允许NULL值（适用于新建/修改表格的字段时，NULL允许为零值/未知值，缺省设置，not null不允许为空）。

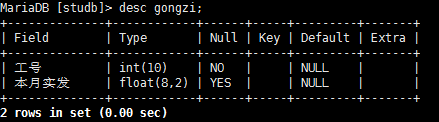


是否为字段设置默认值（适用于新建表或者为表增加字段时DEFAULT 设置默认值，默认为NULL）



修改：

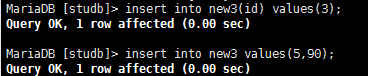


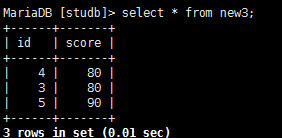


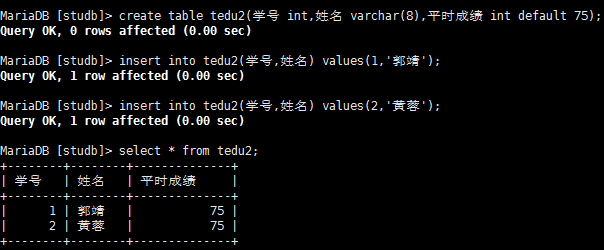
新增：











**主键索引**

什么索引：

指的是对表格记录的一些字段进行排序的方法

类似于书的目录

包括Btree、B+tree、hash等多种算法

索引的优点：通过创建唯一性索引，可以保证表中每一行数据的唯一性

可以加快数据查询的速度

Primary key主键

主键的特点及要求

每一张表只能有一个主键字段

通常把能够唯一标识一条记录的字段设为主键字段

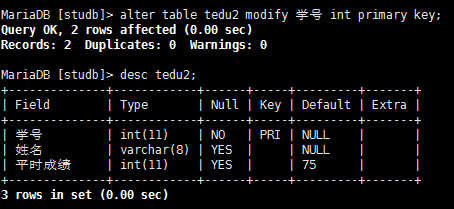
主键字段不允许出现重复的记录值，且不允许赋NULL值

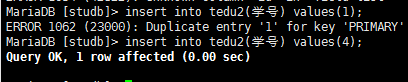
AUTO\_INCREMENT自动增长

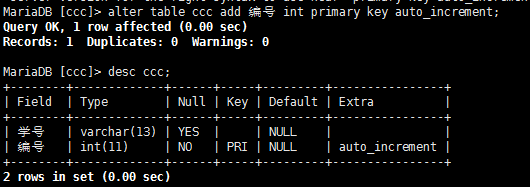
将字段设置值自动增加，无需手动赋值，适用于整数字段

通常与主键连用。

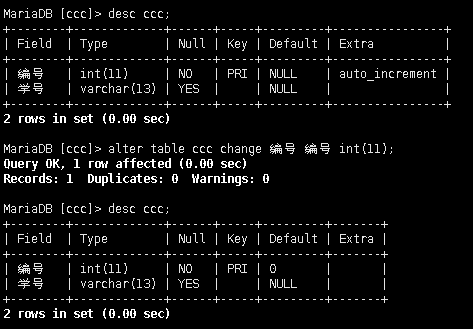
适用于新建/修改表格的字段时：primary key（字段xx）指定字段xx为主键字段



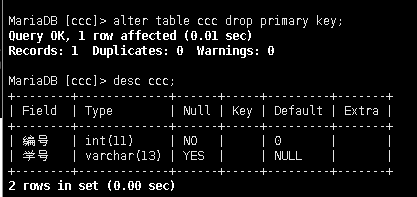




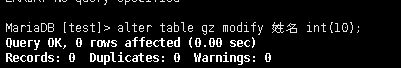
删除自动增加行



删除主键



删除自增长



**外键**

如果一个表中有一个非主键的字段指向了别一张表中的主键，就将该字段叫做外键，一张表中可以有多个外键。

默认作用有两点：1.对字表（外键所在的表）的作用：子表在进行写操作的时候，如果外键字段在父表中找不到对应的匹配，操作就会失败。

2.对父表的作用：对父表的主键在子表中被引用，操作就会失败。

外键的定制作用------三种约束模式：

district：严格模式（默认），父表不能删除或更新一个被子表引用的记录

cascade：级联模式，父表操作后，子表关联的数据也跟着一起操作

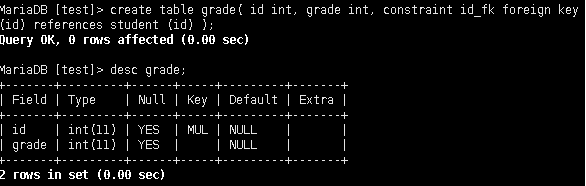
set null：置空模式，前提外键字段允许为null，父表操作后，子表对呀的字段被置空

使用外键的前提：

1. 存储引擎必须是innodb，否则创建的外键无约束效果
2. 外键的列类型必须与父表的主键类型完全一致
3. 外键的名字不能重复
4. 已经存在数据的字段被设为外键时，必须保证字段中的数据与父表的主键数据对应起来。

新增外键：

1. 在创建时：



foreign key(class)　references my\_tab2(主键字段名);

1. 在创建好的表中添加：

alter table my\_tab1 add [constraint 外键名] foreign key(外键字段名) references mytab2(主键字段名);

1. 删除外键：

alter table my\_tab drop foreign key 外键名字;

