数据库2

**约束条件：**

null(允许为空，默认设置)

not null(不允许为空)

default(设置默认值)

key(索引类型：普通索引，)

mysql> create table db2.t2(class char(10),name char(3),age tinyint default 19,likes set("a","b","c","d") not null);

#爱好不能为空

#############################################

**修改表结构**

alter table 表名 add（添加新的字段）

mysql> alter table db1.t5 add stu\_num char(9) first,add pay float(7,2) default 20000 after name;

#添加字段，在原有字段前面加，也可以在原有字段后面加

修改字段时，不能与原有数据冲突，修改之后的字段对新增数据有效

alter table 表名 modify（修改字段的类型，还可修改字段位置）

mysql> alter table db1.t5 modify sex enum('boy','girl','no') after name;

#修改字段类型,

alter table 表名 change 旧字段名 新字段名 类型

mysql> alter table db1.t5 change stu\_num stu\_id char(9) not null;

alter table 表名 drop 字段名（删除字段）

mysql> alter table db1.t5 drop qq;

#删除某个字段

alter table 表名 rename 新表名(修改表名)

mysql> alter table db1.t5 rename infor;

###########################################

**mysql键值**

普通索引 index(使用索引之后 key状态显示为MULL)

唯一索引 unique

主键 primary key(使用主键之后key状态显示为PRI)

索引在一个表中可以有多个

一个表中只能有一个主键

创建新表时，添加索引

mysql> create table db2.t3(name char(5),sex enum("w","m"),age tinyint,index(age),index(name));

查看索引

mysql> show index from db2.t3\G;

在原有表上添加索引

create index 索引名 on 表名(字段名) 索引名可与字段名相同或者不同

mysql> create index sex on db2.t3(sex);

删除索引

drop index 索引名 on 表名

mysql> drop index name on t3;

使用主键（主键对应的字段具有唯一性，不能重复，且不允许赋空值）

primary key

创建新表时，添加主键

方法1：

mysql> create table db2.t4(class char(10),name char(5),age tinyint,stu\_id char(20),primary key(stu\_id));

方法2：

mysql> create table db2.t4(class char(10),name char(5),age tinyint,stu\_id char(20) primary key);

在已有表中设置主键：

mysql>alter table db2.t1 add primary key(字段名);注意 该字段里面的数据不能有空值或者重复值

复合主键

mysql> create table db2.t6(class char(10),name char(5),pay enum("yes","no") default "no",primary key(class,name,pay));

#使用复合主键时，字段赋值不能完全相同，有一个字段赋值不同即可成功赋值。

mysql> insert into db2.t6 values("nsd1903","ll","yes");

删除复合主键

mysql> alter table db2.t6 drop primary key;

添加复合主键

mysql> alter table db2.t6 add primary key(class,name,pay);

主键与auto\_increment连用

数值类型字段的不赋值自增长（类似i++）

mysql> create table db2.t7(id int primary key auto\_increment,name char(5),sex enum("w","m"),class char(10));

mysql> insert into db2.t7(name,sex,class) values("lucy","w","nsd1903");

Query OK, 1 row affected (0.02 sec)

mysql> select \* from db2.t7;

+----+-------+------+---------+

| id | name | sex | class |

+----+-------+------+---------+

| 1 | tom | m | nsd1902 |

| 2 | jerry | m | nsd1902 |

| 3 | lucy | w | nsd1903 |

+----+-------+------+---------+

如果给字段赋值n，再一次添加数据之后，字段数值为n+1

mysql> insert into db2.t7 values(8,"harry","m","nsd190204");

Query OK, 1 row affected (0.02 sec)

mysql> insert into db2.t7(name,sex,class) values("jack","m","nsd190202");

Query OK, 1 row affected (0.03 sec)

mysql> select \* from db2.t7;

+----+-------+------+-----------+

| id | name | sex | class |

+----+-------+------+-----------+

| 1 | tom | m | nsd1902 |

| 2 | jerry | m | nsd1902 |

| 3 | lucy | w | nsd1903 |

| 8 | harry | m | nsd190204 |

| 9 | jack | m | nsd190202 |

+----+-------+------+-----------+

5 rows in set (0.00 sec)

外键（foreign key）保证信息一致，表被约束限制

一张表参考另一张表的字段（该字段必须为索引类型），字段类型要一致，表的搜索引擎必须是innodb。

mysql> create table db1.yg(yg\_id int primary key auto\_increment,name char(10))engine=innodb;

#新建yg表作为参考表

mysql> create table gz(gz\_id int,gz float(7,2) default 20000,foreign key(gz\_id) references yg(yg\_id) on update cascade on delete cascade)engine=innodb;

#新建gz表，gz\_id参照yg\_id，同步删除，同步更新

mysql> update yg set yg\_id=7 where yg\_id=3;

#yg表更新，gz表随之更新

mysql> select \* from db1.gz;

+-------+----------+

| gz\_id | gz |

+-------+----------+

| 1 | 26000.00 |

| 2 | 27000.00 |

| 7 | 20000.00 |

+-------+----------+

3 rows in set (0.00 sec)

mysql> delete from yg where yg\_id=7;

Query OK, 1 row affected (0.03 sec)

mysql> select \* from db1.gz;

+-------+----------+

| gz\_id | gz |

+-------+----------+

| 1 | 26000.00 |

| 2 | 27000.00 |

+-------+----------+

2 rows in set (0.00 sec)

mysql> alter table db1.gz drop foreign key gz\_ibfk\_1;

#删除外键，gz\_ibfk\_1为外键名，可通过show create table db1.gz\G查询

mysql> alter table db1.xf add foreign key(xf\_id) references stu(stu\_id) on update cascade on delete cascade;

#在已有表添加外键，保持字段内（使用外键）数据一致