Mysql Day2

**数据类型的关键字 存储范围 赋值方式 合理赋值**

数值类型的宽度，是显示宽度，不能够给字段赋值的大小，字段的由类型决定

Create database db2;

Use db2;

Create table t1(name char(3),level int(3) **zerofill**); 用0补空格位

1. 字段约束条件
2. 作用：限制如何给字段赋值的
3. 包括内容有：

是否给字段赋NULL值，

null 默认赋值NULL值

Not null 不允许赋null值

Zerofill 值不够位数，用0补齐

Create table t4(name char(4) **not null**,level int(3) **zerofill**); #name不允许空

Select \* from t4 where name is null； #查询空值

1. 1 修改表结构

Key 键值类型：普通索引 唯一索引 全文索引 主键 外键

Default 默认值作用：当不被字段赋值时，使用默认值给字段赋值

不设置默认值是 系统定义的是null

Default 值

修改表结构的默认值null

Extra 额外设置，字段是否设置为自动增加，默认没有自动增长功能

mysql> create table t5(classroom char(7) **default** "nsd1801",name char(5) **not** **null**,age tinyint(2) **unsigned** zerofill **default** 18, sex enum("boy", "girl") **not null default** "boy");

mysql> insert into t5(name) values("lucy"); #没有赋值默认赋默认值

mysql> insert into t5 values("nsd1802","alice",21,"girl"); #有赋值，即用所给赋值

mysql> select \* from t5;

二、2修改表结构

添加新字段的类型宽度及约束条件

1修改表结构的命令

Mysql> alter table 库.名 执行动作；

添加新字段: alter table 库.表 add 字段名 类型（宽度） 【约束条件】

默认不指定在后面加，First指定在第一列. after跟字段可以指定在哪个字段后面添加

mysql> use nbdb;

mysql> alter table t1 **add** addrhome varchar(50) default "guangdong";

mysql> alter table t1 add name char(15) **first**;

mysql> alter table t1 add sex enum("boy","girl") not null default "boy" **after** name;

mysql> select \* from t1; 查看检验

删除已有字段: alter table 库.表 drop 字段名；

Alter table stuble t1 drop sex;

Alter table stuble t1 drop sex,drop name;

**修改已有字段的类型宽度及约束条件。修改时不能与已经存储的数据矛盾，**

默认不指定在后面加，First指定在第一列. after跟字段可以指定在哪个字段后面添加

修改已有字段：alter table 库.表 Modify 字段名 执行动作；

mysql> alter table nbdb.t1 modify addrhome char(15) default "guangdong";

mysql> alter table t1 modify level smallint(6) not null;

mysql> alter table t1 modify age tinyint(4) first;

mysql> select \* from t1;

mysql> alter table t1 modify age tinyint(4) after name;

mysql> select \* from t1;

修改字段名：alter table 库.名 change 原字段 新字段 执行动作；

mysql> alter table t1 change addrhome home char(15) default "guangdong";

修改库名：alter table 库名 rename 新库名；

mysql> alter table t1 rename dogperson;

1. Mysql键值

设置在表中字段上的，作用时约束如何给字段赋值。同时会给字段做索引。

索引介绍：树形目录结构 类似与书的“目录”

优点：加快查询表记录的速度

缺点：会减慢编辑表记录的速度，且占用磁盘的物理存储空间

(delete insert update)

1、键值类型：普通索引 唯一索引 全文索引 **主键 外键**

\* \*

1）普通索引使用 (index)

使用规则

查看

Desc 表名；

Show index from 表名；

创建索引

在已有表创建

**Table:t2im**

**Key\_name:aaa**

**Column\_name:age**

**Index\_type:BTREE(二叉树）Hash**

二叉数：数值越多，分插越多

Create index 索引名 on 库名.表名（字段名)

mysql> create index aaa on nbdb.t2(age);

mysql> desc nbdb.t2;

mysql> show index from nbdb.t2;

建表时创建

Create table 表名（

字段列表，

Index（字段名）；

Index（字段名）；

）；

mysql> create table t21(

-> classroom char(7) default "nsd1801",

-> name char(5) not null,

-> age tinyint(2) unsigned zerofill default 18,

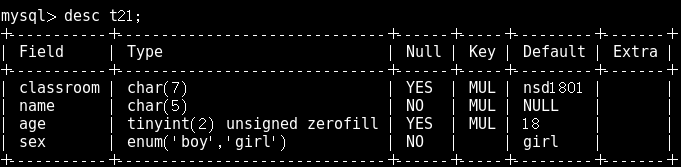
-> sex enum("boy","girl") not null default "girl",

-> index(name),

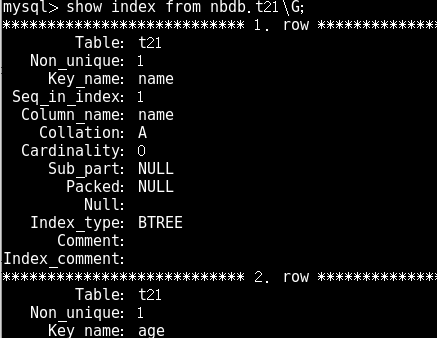
-> index(age),

-> index(classroom));

mysql> desc t21;



mysql> show index from nbdb.t21\G;



删除

drop index 索引名 on 库名.表名

mysql> drop index aaa on t2;

mysql> desc t2;

mysql> show index from t2;

1. 主健

使用规则

限制不能重复**相同一个字符串**和**空值**

删除**主键**时不会自动删除**不能写入空值**

查看

Desc 表名； key ----------->PRI

创建

在已有表创建： alter table 表 add primary key（字段名）；

mysql> alter table t21 add stu\_id char(9) first;

mysql> alter table t21 add **primary key(stu\_id)**;

mysql> insert into t21 values(null,"nsd1801","ssh",18,"boy"); 写入空值报错

mysql> insert into t21 values("**nsd180101**","nsd1801","ssh",18,"boy"); 不是空值可以写入

mysql> insert into t21 values("**nsd180101**","nsd1858","ssg",18,"boy"); 写入重复报错

mysql> insert into t21 values("**nsd180118**","nsd1801","ssg",18,"boy"); 不重复可以写入

在创建表时创建

Create table 表名（

字段列表，

……

Primary key（字段名）

）；

mysql> create table nbdb.t22(person\_id char(18), name char(10), age tinyint(2) unsigned, **primary key(person\_id))**; #设置字段的主键

mysql> desc nbdb.t22;

mysql> insert into nbdb.t22 values("**xx**","ben",11); #写入可以

mysql> insert into nbdb.t22 values("**xx**","bob",10); #写入报错

mysql> insert into nbdb.t22 values("**fx**","bob",10); #写入可以

Create table 表名（

字段 类型（宽度） primary key，

……

）；

复合主键

创建复合主键的使用：多个字段一起做主键，插入记录时，只要做主键字段的值不同时重复，旧可以插入记录。

主键primary key 通常和auto\_increment连用

让字段的值自动增长 1++

数值类型

创建复合主键：

Create table 表名（

字段列表，

Primary key（字段名，字段名……）

）；

mysql> create table nbdb.t23(

-> clientip char(15),

-> port smallint(2),

-> status enum("allow","deny"),

-> **primary key**(clientip,port));

mysql> insert into nbdb.t23 values(**"1.1.1.1",22,"**deny"); 正常写入

mysql> insert into t23 values("**1.1.1.1",22,"**allow"); 重复的不能写入，报错

mysql> insert into t23 values(**"1.1.1.1",80,**"allow"); 正常写入

mysql> insert into t23 values(**"2.1.1.1",80,**"allow"); 正常写入

主键primary key 通常和auto\_increment连用

(auto\_increment)值自动增长，类似i++

mysql> create table nbdb.t24(

-> stu\_id int(2) zerofill primary key auto\_increment,

-> name char(10),

-> age tinyint(2) unsigned,

-> sex enum("boy","girl"));

mysql> insert into t24(name,age,sex)values("bob",19,"girl");

mysql> insert into t24(name,age,sex)values("bob",19,"girl");

mysql> insert into t24(name,age,sex)values("bob",19,"girl");

mysql> select \* from t24;

mysql> alter table dogperson add id int(2) primary key

-> auto\_increment first; 在创建好的表中加入主键

删除主键

alter table 表名 drop primary key;

mysql> alter table t21 drop primary key;

1. 外键

作用：限制如何给字段赋值

-- 给当前表中字段赋值时，值只能在其他表的指定字段值的范围里选择。

使用规则

被参照必须得要存在

被参照的表的值必须时唯一的

查看

创建外键

**创建外键 foreign key 的命令格式：**

Create table 表（

字段名列表，

Foreign key（字段名） references 库名.表名（字段名） #设置被参照更改自动更改

On update cascade on delete cascade #设置被参照的表的数据删除自动删除

）；

在已有表中创建

Alter table Foreign key（字段名） references 库名.表名（字段名） #

On update cascade on delete cascade #被参照的表的数据删除自动删除

）

建表时创键外键

mysql> create table jfb(

-> jfb\_id int(2) **primary key auto\_increment**,设置主键自动赋值

-> name char(15),

-> pay float(7,2))**engine=innodb**;

mysql> insert jfb (name,pay) values("bob",26800);

mysql> insert jfb (name,pay) values("tom",26000);

mysql> select \* from jfb;

//创建被参照表并且写入内容

mysql> create table bjb(

-> bjb\_id int(2),

-> name char(15),

-> **foreign key**(bjb\_id) **references jfb**(jfb\_id)

#指定外键字段 #指定被参照表字段

-> **on update cascade on delete cascade**);

#同步被参照删除之后自动删除

//创建外键表

写入验证：

mysql> insert into bjb values(9,"bob"); 失败

mysql> insert into bjb values(1,"bob"); 成功

mysql> insert into bjb values(2,"tom"); 成功

mysql> insert into bjb values(3,"lucy"); 失败

同步更新验证

mysql> update jfb set jfb\_id =9 where name="bob"; 修改bob的jfb\_id的值

mysql> select \* from bjb;

同步删除验证

mysql> delete from jfb where jfb\_id=2; #删除id等于2的数据

mysql> select \* from bjb;

3、删除外键

查看

Show create table 表名；

mysql> show create table bjb; #查看创建表的命令

删除

Alter table 表名 drop foreign key 外键名；

mysql> alter table bjb drop foreign key bjb\_ibfk\_1;