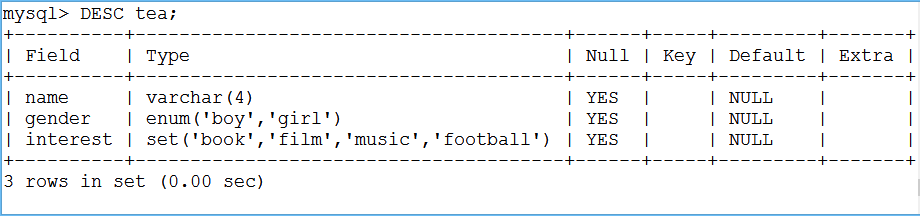
# DB约束条件

**约束条件**

|  |  |
| --- | --- |
| 值 | 解释 |
| Null | 允许为空，默认设置 |
| Not null | 不允许为空 |
| key | 索引类型 |
| default | 设置默认，缺省为null |



create table t8(name char(10) not null default "tom",

age tinyint(2) not null default 18,

likes set("eat","sleep","move") not null default "move"

);

**修改表结构**

语法结构

|  |  |
| --- | --- |
| 选项 | 作用 |
| add | 添加字段,添加主键 |
| modify | 修改字段类型，及约束 |
| change | 修改字段名 |
| drop | 修改字段 |
| rename | 修改表名 |

修改时如果表下有数据，这是不能修改表的数据类型，宽度和约束条件，不然会产生冲突，修改不成功，宽度可以调大。

增加字段并放在表的第一列

mysql> alter table t9 add sex set("man","wman") default "man" after name;

增加字段并放在哪个字段之后

修改字段名

alter table 表明 change 源字段名 新字段名 类型(宽度) 约束条件;

alter table t9 change sex ssex set("man","wman","zhong") not null default "zhong" ;

mysql> alter table t9 add sex set("man","wman") default "man" after name;

修改字段的位置

alter table t9 modify ssex set('man','wman','zhong') not null default 'zhong' after age;

select \* from t9

**MySql 索引**

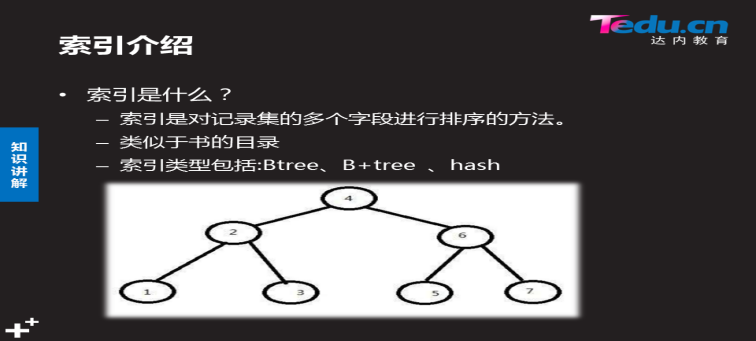
是什么

- 索引时对记录集的多个字段进行排序的方法

- 类似于数的目录

- 索引算法包括：Btree(二叉树)、B+tree、hash，mysql默认Btree，可修改

- 索引的标志是MUL



索引优点

- 通过创建为以性索引，可以保证数据库表中每一行数据的唯一性

- 可以加**快数据的检索速度**

索引优点

- 当对表中的数据进行数据进行增加、删除和修改的时候，索引叶要动态的维护，降低了数据的维护速度

- 索引需要占用物理空间

**键值类型**

Foreign key

Primary key

Fulltext

Unique

Index

Index 使用说明

- 一个表中可以有多个index字段

- 字段的值允许有重复，且可以赋null值

- 经常把做查询的条件的字段设置为index字段

- index字段的key标志是MUL

建表的时候指定索引字段

- index(字段1),index(字段2)...

create table ta( id tinyint not null

Name char(10) not null,

Index(id),index(name)

);

在已有的表中设置index字段

- create index 索引名 on 表名(字段名);

- create index name on ta(name);

删除表中的索引字段

- drop index 索引名 on 表名;

- drop index name on ta(name);

查看表的索引信息

- show index from 表名;

**primary key主键**

注意

- 一个表中只能有一个primary key字段

- 对应的字段值不允许有重复，且不允许赋NULL值

- 如果有多个字段都做为primary key，称为复合主键，必须一起创建,一起删除

- 主键字段的key标志是PRI

- 通常与AUTO\_INCREMENT连用(auto\_increment)

- 经常把表中能够为以表示记录的字段设置为主键字段 [记录编号字段]

添加主键的创建

- create table ta1(id tinyint primary key, name char(10), stu tinyint, index(stu));

- create table ta1(id tinyint, name char(10), stu tinyint, primary key(id), index(stu));

复合主键

把多个字段合并在一起充当主键

多个字段不能同时相同.

创建

create table ta3(cip char(15),

port tinyint,

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

primary key(cip,port)

);

desc ta3; 查询的时候cip和port的key标示是PRI

删除主键

alter table ta3 drop primary key;

添加主键

alter table ta3 add primary key(cip);

添加复合主键

alter table ta3 add primary key(cip,port);

在修改主键的时候不需要加上主键的设置信息，因为主键有专门删除的语句

auto\_incremen 自动增长的字段

create table ta3(id int(3) primary key auto\_increment,name char(10);

删除有自增长的主键时，需要将主键删除后才能删除主键

alter table ta3 modify id int(3) not null;

alter table ta3 drop primary key;

**Foreign key外键**

什么是外键

让当前表字段的值在另一个表中字段值的范围内选择

使用外键的条件

表的存储引擎必须是 innodb

字段类型要一致

被参照字段必须要是索引类型的一种 (primary key)

基本用法

– FOREIGN KEY( 表 A 的字段名 )

References 表 B( 字段名 )

ON UPDATE CASCADE

ON DELETE CASCADE

新建员工表

create table ygtab(

yg\_id int(2) primary key auto\_increment,

name char(10)

)engine=innodb;

指定只添加name的字段，yg\_id自动增长

insert into ygtab(name) values("fangfang");

新建工资表

create table gztab(

gz\_id int(2),

pay float(7,2),

foreign key(gz\_id) references ygtab(yg\_id)

on update cascade on delete cascade

)engine=innodb;

查看建表命令

Show create table 表名;

当表被外键关联着，此表无法删除