目录

[Mysql day06 2](#_Toc501121917)

[1 视图 2](#_Toc501121918)

[1.1 创建视图 2](#_Toc501121919)

[1.2 查看视图 3](#_Toc501121920)

[1.3 删除视图 3](#_Toc501121921)

[2 索引 3](#_Toc501121922)

[2.1 创建索引 4](#_Toc501121923)

[2.2 查看索引 4](#_Toc501121924)

[2.3 删除索引 4](#_Toc501121925)

[2.4 查询语句是否使用索引来查询 4](#_Toc501121926)

[3 数据库备份恢复 5](#_Toc501121927)

[3.1 备份 5](#_Toc501121928)

[3.2 恢复 6](#_Toc501121929)

[4 数据库编程 6](#_Toc501121930)

[5 变量 7](#_Toc501121931)

[5.1 自定义会话变量 7](#_Toc501121932)

[5.2 mysql系统环境变量 7](#_Toc501121933)

[5.3 代码块中的局部变量 7](#_Toc501121934)

[6 存储过程 8](#_Toc501121935)

[6.1 创建存储过程 8](#_Toc501121936)

[6.1.1 基本语法 8](#_Toc501121937)

[6.1.2 参数 8](#_Toc501121938)

[6.2 调用存储过程 9](#_Toc501121939)

[6.3 查看存储过程 9](#_Toc501121940)

[6.4 删除存储过程 9](#_Toc501121941)

[6.5 分支判断 10](#_Toc501121942)

[6.6 循环 11](#_Toc501121943)

[7 函数 12](#_Toc501121944)

[7.1 创建函数 12](#_Toc501121945)

[7.2 调用函数 12](#_Toc501121946)

[7.3 查看函数 13](#_Toc501121947)

[7.4 删除函数 13](#_Toc501121948)

[8 触发器 17](#_Toc501121949)

[8.1 创建触发器 18](#_Toc501121950)

[8.2 查看触发器 19](#_Toc501121951)

[8.3 删除触发器 19](#_Toc501121952)

[9 作业 19](#_Toc501121953)

Mysql day06

# 视图

将查询语句保存到数据库,起个名字,就是视图,以后可以简单的从视图查询

为什么要是用视图

* 简化查询
* 安全

可以向用户之暴露视图,而不暴露具体的表

## 创建视图

create or replace view v1

as

select ....;

视图测试

set names gbk;

use hr;

create or replace view v1

as

select

e1.employee\_id,e1.first\_name,

d.department\_name,

l.city,

e2.first\_name dept\_mgr,

e3.first\_name mgr\_name

from

employees e1

join departments d

on e1.department\_id=d.department\_id

join locations l

on d.location\_id=l.location\_id

join employees e2

on d.manager\_id=e2.employee\_id

join employees e3

on e1.manager\_id=e3.employee\_id;

select \* from v1;

## 查看视图

show tables;

desc v1;

show create table v1\G

## 删除视图

drop view v1;

# 索引

加速查询

索引为字段中数据,生成一份索引数据,通过索引数据,可以快速定位数据

mysql 的innodb和myisam数据表引擎,生成索引模式使用b-tree(平衡红黑树)数据结构

## 创建索引

create index index1 on tb1(c1);

where c1='sdfsdf'

where c1 like 'a%'

where c1 like '%a' 不能使用索引

create index index1 on tb1(c1, c2);

where c1='sdf' and c2='sdf'

where c1='sdf'

where c2='sdfsdf'

创建索引

create index index\_name

on employees(first\_name, last\_name);

## 查看索引

show create table employees\G

## 删除索引

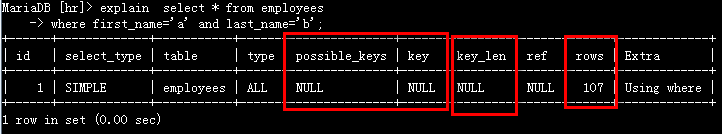
alter table employees

drop index index\_name;

## 查询语句是否使用索引来查询

**explain** select \* from employees

where first\_name='a' and last\_name='b';

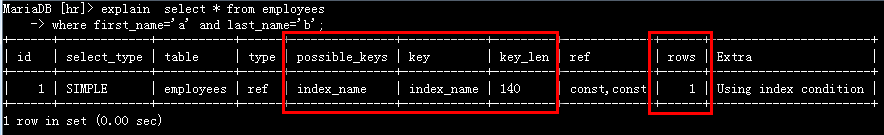


create index index\_name

on employees(first\_name, last\_name);

**explain** select \* from employees

where first\_name='a' and last\_name='b';



# 数据库备份恢复

在操作系统命令行执行

* mysqldump 命令备份数据
* mysql 命令恢复数据

## 备份

一行命令:

mysqldump -uroot -p

--default-character-set=utf8 //表中存的是什么编码

stu>/home/soft01/stu.sql //库>文件

备份hr库(一行命令)

mysqldump -uroot -p

--default-character-set=utf8

hr>d:\hr.sql

## 恢复

将数据恢复到一个指定的库

mysql -uroot -p

--default-character-set=utf8

stu2</home/soft01/stu.sql

恢复员工数据到hr2库

-- 新建hr2库

create database hr2 charset=utf8;

-- 数据恢复

mysql -uroot -p

--default-character-set=utf8

hr2<d:\hr.sql

-- 查看hr2库的数据

use hr2;

show tables;

select \* from employees;

# 数据库编程

* 存储过程
* 函数
* 触发器

# 变量

## 自定义会话变量

set @v1=43; -- 变量赋值必须set

select @v1;

set @v1=5;

select @v1;

## mysql系统环境变量

show variables;

show variables like 'tx\\_%';

show variables like '%char%';

show variables like '%auto%';

set autocommit=off;

set @@autocommit=off;

## 代码块中的局部变量

begin

declare i int;

declare i int default 0;

end;

# 存储过程

存储在数据库中的一段过程代码

## 创建存储过程

### 基本语法

-- 自定义结束符

delimiter //

drop procedure if exists p1;

//

create procedure p1()

begin

select now();

end;

//

### 参数

* in 输入参数
* out 输出参数
* inout 即可作为输入参数也可作为输出参数

p1(in a int, out b int)

call p1(5, @v1);

5作为入参传给a

@v1 变量作为输出参数,来获得存储过程的计算结果

存储过程参数

drop procedure if exists p2;

//

create procedure p2(in a int, out b int)

begin

set b = a\*a;

end;

//

call p2(5, @v1); //

select @v1; //

call p2(3, @v1); //

select @v1; //

## 调用存储过程

call p1();

//

## 查看存储过程

show procedure status\G

所有存储过程

show procedure status where db='test'\G

查看指定库中的存储过程

show create procedure p1\G

## 删除存储过程

drop procedure if exists p1;

## 分支判断

if 条件 then

...

end if;

if 条件 then

...

else

...

end if;

case

when 条件 then ...;

when 条件 then ...;

else

end case;

case 数据

when 值1 then ...;

when 值2 then ...;

else

end case;

分支测试

drop procedure if exists p3;

//

create procedure p3(in a int)

begin

if a=1 then

select 1;

end if;

case

when a=2 then select 2;

when a=3 then select 3;

else select 4;

end case;

end;

//

call p3(1); //

call p3(2); //

call p3(3); //

call p3(4); //

call p3(5); //

## 循环

while 条件 do

...

end while;

lp:loop -- 给循环起别名

...

if ... then

leave lp; -- 离开lp循环

end if;

end loop;

repeat

...

until 条件 end repeat;

循环测试

-- 创建一个测试表

create table t(

c int

);

-- 创建存储过程,循环向表中插入n行数据

create procedure p4(in n int)

begin

-- 定义一个控制循环次数的变量

-- declare 定义局部变量

declare i int default 0;

-- while

while i<5 do

insert into t values(i+1);

set i=i+1;

end while;

set i=0;

lp:loop

insert into t values(i+1);

set i=i+1;

if i>=5 then

leave lp;

end if;

end loop;

set i=0;

repeat

insert into t values(i+1);

set i=i+1;

until i>=5 end repeat;

end;

//

call p4(5);

//

# 函数

函数有返回值

## 创建函数

create or replace function fn1(a int)

returns varchar(255)

begin

declare r int;

set r = a\*a;

return r;

end;

## 调用函数

select fn1(3);

select employee\_id,fn1(department\_id)

from employees;

## 查看函数

show function status\G

show function status where db='test'\G

show create function f\G

## 删除函数

drop function if exists f;

函数测试

-- a的b次方

drop function if exists fn\_pow;

create function fn\_pow(a bigint, b bigint)

returns bigint

begin

declare r bigint;-- 用来存结果

declare i int; -- 用来控制循环此时

set r = a; -- 从第一个a开始,向后乘b-1次

set i = 0;

while i<b-1 do

set r = r\*a;

set i = i+1;

end while;

return r;

end;

//

select fn\_pow(2,3);

//

select c, fn\_pow(c,3) from t;

//

随机产生字符串的函数

\* 随机产生[a, b)长度的字符串

\* a+[0,b-a): floor(a+rand()\*(b-a))

\* 假设随机长度是 3,产生三个随机字符

\* 从一组字符串s0随机选取 'abcdefg....'

\* 随机定位j位置1+[0,char\_length(s0)-1)

\* 取一个字符substring(s0,j,1)

\* 重复3次取三个字符连接

drop function if exists fn\_randstr;

create function fn\_randstr(a int, b int)

returns varchar(255)

begin

declare len int;-- 随机的长度

declare s0 varchar(600) default ' abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ的一是在不了有和人这中大为上个国我以要他时来用们生到作地于出就分对成会可主发年动同工也能下过子说产种面而方后多定行学法所民得经十三之进着等部度家电力里如水化高自二理起小物现实加量都两体制机当使点从业本去把性好应开它合还因由其些然前外天政四日那社义事平形相全表间样与关各重新线内数正心反你明看原又么利比或但质气第向道命此变条只没结解问意建月公无系军很情者最立代想已通并提直题党程展五果料象员革位入常文总次品式活设及管特件长求老头基资边流路级少图山统接知较将组见计别她手角期根论运农指几九区强放决西被干做必战先回则任取据处队南给色光门即保治北造百规热领七海口东导器压志世金增争济阶油思术极交受联什认六共权收证改清己美再采转更单风切打白教速花带安场身车例真务具万每目至达走积示议声报斗完类八离华名确才科张信马节话米整空元况今集温传土许步群广石记需段研界拉林律叫且究观越织装影算低持音众书布复容儿须际商非验连断深难近矿千周委素技备半办青省列习响约支般史感劳便团往酸历市克何除消构府称太准精值号率族维划选标写存候毛亲快效斯院查江型眼王按格养易置派层片始却专状育厂京识适属圆包火住调满县局照参红细引听该铁价严龙飞';

declare j int; -- 从s0随机挑选的位置

-- 保存最终结果,从空串开始

declare s varchar(255) default '';

declare i int default 0;-- 控制次数

-- 算长度

set len = floor(a+rand()\*(b-a));

-- 循环len次挑选字符连接到s

while i<len do

-- 随机定位

set j =

floor(1+rand()\*(char\_length(s0)-1));

-- 拿出s0中j位置的一个字符连到s

set s=concat(s, substring(s0,j,1));

set i=i+1;

end while;

return s;

end;

//

select fn\_randstr(3,6);

//

set names gbk;

//

-- 如果数据库字符集是latin1,设置gbk也不管用

-- 1.换库 2.改当前库的字符集属性

alter database 库名 charset utf8;

产生大量数据

-- 创建一张**内存表**,大量暂时插入内存表中

-- 再将内存表数据,一次导入**磁盘表**

-- 内存表

create table tm(

c varchar(20)

)engine=**memory** charset=utf8;

//

-- 磁盘表

create table td(

id int primary key auto\_increment,

c varchar(20)

)engine=innodb charset=utf8;

//

-- 批量产生数据的存储过程

drop procedure if exists gen\_data;

//

create procedure gen\_data(in n int)

begin

declare i int default 0;-- 控制次数

-- 每次产生的字符串

declare s varchar(255);

-- 向内存表插入n条数据

while i<n do

set s = fn\_randstr(3,6);

insert into **tm** values(s);

set i=i+1;

end while;

-- 将内存表数据一次导入磁盘表

insert into td(c) select \* from **tm**;

-- 清空内存表

delete from **tm**;

end;

//

call gen\_data(50000);//

call gen\_data(50000);//

call gen\_data(50000);//

...

没有索引和添加索引查询对比

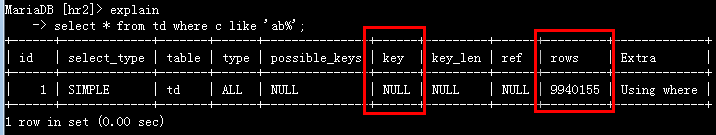
use hr2;

-- 1千万数据,需要3.5秒左右

select \* from td where c like 'ab%';

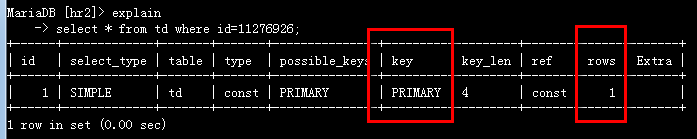
**explain**

select \* from td where c like 'ab%';



**explain**

select \* from td where id=11276926;

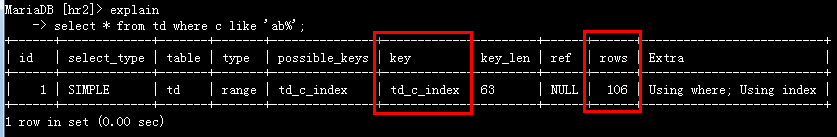


-- 创建c字段索引

create index td\_c\_index on td(c);

**explain**

select \* from td where c like 'ab%';



# 触发器

在插入、修改和删除时，可以自动触发一段代码执行

* 操作之前触发
* 操作之后触发
* 六种触发器:
  + 一张表中最多只能有6个触发器
  + before insert
  + before update
  + before delete
  + after insert
  + after update
  + after delete
* 两个隐含的对象
  + new - 新数据
  + old - 旧数据
  + insert
    - new 插入的新数据
    - old 没有
  + update
    - new 修改之后的新数据
    - old 修改之前的旧数据
  + delete
    - new 没有
    - old 被删除的数据
  + 访问新的数据行或旧的数据行:
    - new.employee\_id
    - old.first\_name

## 创建触发器

create trigger before\_td\_insert

before insert on td for each row

begin

...

end;

触发器测试

-- 新建tb1表

create table tb1(

id int primary key auto\_increment,

c varchar(20),

created datetime,

updated datetime

)engine=innodb charset=utf8;

//

-- 新建触发器,插入数据时

-- 自动为created和updated填入当前时间

create trigger tb1\_before\_insert

before insert on tb1 for each row

begin

set new.created=now();

set new.updated=now();

end

//

insert into tb1(c) values('a');

//

select \* from tb1;

//

## 查看触发器

use information\_schema

//

select \* from triggers\G

//

## 删除触发器

drop trigger 触发器名

//

# 作业

下载文件 jtds.sql

京淘电商数据库(含数据),

新建 jt 数据库,

用数据恢复将 jtds.sql导入jt库

商品表 tb\_item

有什么品牌

最贵商品的品牌、名称、价格

分页查看商品

商品名和商品描述

商品分类表 tb\_item\_cat

查询所有的顶层分类

查询 161 下的二层分类

查询 162 下的三层分类

用户表

订单表

订单编号是 20161001490698615071

查询没中类别的商品数量

查询 类别 163 的商品

查询商品价格不大于100的商品名称列表

查询品牌是联想,且价格在40000以上的商品名称和价格

查询品牌是三木,或价格在10以上的商品名称和价格

查询品牌是三木、广博、齐心的商品名称和价格

查询品牌不是联想、戴尔的商品名称和价格

查找品牌是联想且价格大于10000的电脑名称

查询联想或戴尔的电脑名称列表

查询联想、戴尔、三木的商品名称列表

查询不是戴尔的电脑名称列表

查询所有是记事本的商品品牌、名称和价格

查询品牌是末尾字符是'力'的商品的品牌、名称和价格

名称中有联想字样的商品名称

查询卖点含有'爆款'电脑名称

查询开头字母是A的电脑名称

将地址表中的城市、地区、详细地址组合到一起,

获取订单编号为 20161001490698615071 中的每种商品的标题、单价、数量、总价

统计商品表中各个品牌的商品数量

统计商品表中各个品牌的商品数量,并且按照数量从少到多排序

统计商品表中各个品牌的商品数量,去除品牌为NULL的行,并且按照数量从少到多排序

查询不同品牌中最贵的商品价格

将不同品牌中最贵的商品按照价格降序排序

找出不同品牌中最贵的商品的前三名

查询订购了10000028商品的客户姓名和联系方式(三层子查询)

首先查询订单表,根据10000028这个商品编号找到有该商品的订单编号

然后查询订单表,根据上面得到的订单编号找到订购此订单的用户编号

最后查询用户表,根据第二部中查到的用户编号,找到对应的用户名和联系方式

  查询没中类别的商品数量

    select cid,count(\*) from tb\_item group by cid

    查询 类别 163 的商品

    select id,title,price from tb\_item where cid=163;

    查询商品价格不大于100的商品名称列表

    select id,title,price from tb\_item where price<100;

    查询品牌是联想,且价格在40000以上的商品名称和价格

    select id,title,price from tb\_item where brand='联想' and price>40000;

    查询品牌是三木,或价格在50以下的商品名称和价格

    select id,brand,title,price from tb\_item where brand='三木' or price<50;

    查询品牌是三木、广博、齐心的商品名称和价格

    select id,brand,title,price from tb\_item where brand in('三木','广博','齐心');

    查询品牌不是联想、戴尔的商品名称和价格

    select id,brand,title,price from tb\_item where brand not in('联想','戴尔');

    查找品牌是联想且价格大于10000的电脑名称

    select id,brand,title,price from tb\_item where brand='联想' and price>10000;

    查询联想或戴尔的电脑名称列表

    select id,brand,title,price from tb\_item where brand='联想' or brand='戴尔';

    查询联想、戴尔、三木的商品名称列表

    select id,brand,title,price from tb\_item where brand in('联想','戴尔','三木');

    查询不是戴尔的电脑名称列表

    select id,brand,title,price from tb\_item where brand<>'戴尔';

    查询所有是记事本的商品品牌、名称和价格

    select id,brand,title,price from tb\_item where title like '%记事本%';

    查询品牌是末尾字符是'力'的商品的品牌、名称和价格

    select id,brand,title,price from tb\_item where brand like '%力';

    名称中有联想字样的商品名称

    select id,brand,title,price from tb\_item where title like '%联想%';

    查询卖点含有'爆款'电脑名称

    select id,brand,title,price from tb\_item where cell\_point like '%爆款%';

    查询开头字母是A的电脑名称

    select id,brand,title,price from tb\_item where title like 'A%';

    将地址表中的城市、地区、详细地址组合到一起,

    select concat(receiver\_city, receiver\_district,receiver\_address) addr from tb\_address;

    获取订单编号为 20161001490698615071 中的每种商品的标题、单价、数量、总价

    select title,price,num,total\_fee from tb\_order\_item where order\_id='20161001490698615071';

    统计商品表中各个品牌的商品数量

    select brand,count(\*) from tb\_item group by brand;

    统计商品表中各个品牌的商品数量,并且按照数量从少到多排序

    select brand,count(\*) c from tb\_item group by brand order by c;

    统计商品表中各个品牌的商品数量,去除品牌为NULL的行,并且按照数量从少到多排序

    select brand,count(\*) c from tb\_item where brand is not null group by brand order by c;

    查询不同品牌中最贵的商品价格

    select id,title,brand,price from tb\_item where (brand,price) in(

      select brand,max(price) m from tb\_item where brand is not null group by brand);

    将不同品牌中最贵的商品按照价格降序排序

    select id,title,brand,price from tb\_item where (brand,price) in(

      select brand,max(price) m from tb\_item where brand is not null group by brand)

    order by price desc;

    找出不同品牌中最贵的商品的前三名

    select id,title,brand,price from tb\_item where (brand,price) in(

      select brand,max(price) m from tb\_item where brand is not null group by brand)

    order by price desc

    limit 3;

    查询订购了10000028商品的客户姓名和联系方式

     首先查询订单表,根据10000028这个商品编号找到有该商品的订单编号

     然后查询订单表,根据上面得到的订单编号找到订购此订单的用户编号

     最后查询用户表,根据第二部中查到的用户编号,找到对应的用户名和联系方式

    select id, username, phone, email from tb\_user where id in

      (select user\_id from tb\_order where order\_id in

        (select order\_id from tb\_order\_item where item\_id=10000028))

    select

        distinct u.id, u.username, u.phone, u.email

    from

        tb\_user u join tb\_order o on u.id=o.user\_id

                  join tb\_order\_item i on o.order\_id=i.order\_id

    where

i.                     item\_id=10000028

  商品表 tb\_item

        select \* from tb\_item;

  有什么品牌

        select distinct brand from tb\_item;

  最贵商品的品牌、名称、价格

        select brand,title,price

        from tb\_item

        order by price desc

        limit 1;

  分页查看商品

        select id,brand,title,price

        from tb\_item

        order by price desc

        -- limit 0,10;

        -- limit 10,10;

        -- limit 20,10;

        limit 30,10;

  商品名和商品描述

        select

            title,item\_desc

        from

            tb\_item i join tb\_item\_desc d

            on i.id=d.item\_id

        limit 1;

  商品分类表 tb\_item\_cat

        select \* from tb\_item\_cat;

        select \* from tb\_item\_cat

        where name like '%电脑%' or

              name like '%笔记本%' or

              name like '%手机%';

  查询所有的顶层分类

        select \* from tb\_item\_cat

        where is\_parent=1 and parent\_id is null;

``查询 161 下的二层分类

        select \* from tb\_item\_cat

        where parent\_id=161 order by sort\_order;

  查询 162 下的三层分类

        select \* from tb\_item\_cat

        where parent\_id=162 order by sort\_order;

  用户表

        select \* from tb\_user;

  订单表

        select \* from tb\_order;

        用户 id 是14的订单

        select \* from tb\_order where user\_id=14;

  订单编号是 20161001490698615071

      查询这个订单的所有商品

        select \* from tb\_order\_item

        where order\_id=20161001490698615071;

周一java基础换老师,花倩

* java 知识量大
* 难度高
* 会有受挫感
* 非常中正常的
* 换老师,要尽快适应老师