燕十八mysql系统教程笔记

char定长字符串，无论够不够指定长度都占满长度，如果不够长度会自动用空格补到指定长度

ecshop 建表

CREATE TABLE `ecs\_goods` (

`goods\_id` mediumint(8) unsigned NOT NULL AUTO\_INCREMENT,

`cat\_id` smallint(5) unsigned NOT NULL DEFAULT '0',

`goods\_sn` varchar(60) NOT NULL DEFAULT '',

`goods\_name` varchar(120) NOT NULL DEFAULT '',

`click\_count` int(10) unsigned NOT NULL DEFAULT '0',

`goods\_number` smallint(5) unsigned NOT NULL DEFAULT '0',

`market\_price` decimal(10,2) unsigned NOT NULL DEFAULT '0.00',

`shop\_price` decimal(10,2) unsigned NOT NULL DEFAULT '0.00',

`add\_time` int(10) unsigned NOT NULL DEFAULT '0',

`is\_best` tinyint(1) unsigned NOT NULL DEFAULT '0',

`is\_new` tinyint(1) unsigned NOT NULL DEFAULT '0',

`is\_hot` tinyint(1) unsigned NOT NULL DEFAULT '0',

PRIMARY KEY (`goods\_id`)

)ENGINE=MyISAM AUTO\_INCREMENT=33 DEFAULT CHARSET=utf8;

导入数据

insert into goods select goods\_id,cat\_id,goods\_sn,goods\_name,click\_count,goods\_number,market\_price,shop\_price,add\_time,is\_best,is\_new,is\_hot from ecshop.ecs\_goods;

查询详细练习

1.where条件 in

select goods\_id,goods\_name,shop\_price from goods where cat\_id in (4,11);

2.between xxx and xxx 是包括边界值的

select goods\_id,cat\_id,goods\_name,shop\_price from goods where shop\_price

between 100 and 500;

3.优先级 not and or

1.9:取出第3个栏目下面价格<1000或>3000,并且点击量>5的系列商品

select goods\_id,cat\_id,goods\_name,shop\_price,click\_count from ecs\_goods where

cat\_id=3 and (shop\_price <1000 or shop\_price>3000) and click\_count>5;

这里必须要用括号

1.12:取出名字为"诺基亚Nxx"的手机

select goods\_id,cat\_id,goods\_name,shop\_price from ecs\_goods

where goods\_name like '诺基亚N\_\_';

用\_ 可以通配任意单个字符

where是对磁盘中的表发挥作用，查询出结果放在结果表中 所以 where中不能有结果表中的列，此时可用having对结果表进行筛选

having的使用：

select goods\_id,goods\_name,(market\_price-shop\_price) as discount from goods having discount>300;

不使用having的用法：

select goods\_id,goods\_name,(market\_price-shop\_price) as discount from goods where market\_price-shop\_price>300;where 后面只能使用磁盘表中已有的字段，不能使用内存中结果表的字段

列就是变量,在每一行上,列的值都在变化.

Where条件是表达式,在哪一行上表达式为真,

哪一行就取出来

select count(\*)就是数绝对行数，如果count(id)如果id为null则不计算在内！！！所以要数记录数，最好用count(\*)

在myisam表中count(1)和count(\*)是一样的

count(列名)查询到的是该列不为null的所有行的行数

某一列的最大值：max() 最小值min()

总和 sum() 平均值 avg()

查询模型：where磁盘表得到行数不一样的表 -> group by -> 计算得出结果表 -> having 筛选结果表 -> 输出结果 就是取出结果表中的部分结果

查询总货款

select sum(goods\_number\*shop\_price) from goods;

查询每种商品的总货款

select cat\_id,sum(goods\_number\*shop\_price) from goods group by cat\_id;

查询积压货款超过20000的

select cat\_id,sum(goods\_number\*shop\_price) as hk from goods group by cat\_id having hk>20000;

insert into result values

('张三','数学',90),

('张三','语文',50),

('张三','地理',40),

('李四','语文',55),

('李四','政治',45),

('王五','政治',30);

查出两门或两门以上不及格同学的平均分

insert into result values

('赵六','数学',90),

('赵六','语文',100),

('赵六','地理',70);

思路：首先要得到一个平均分和挂科数的结果表，然后再在这个结果表中筛选

注意：字段即变量，也可以进行逻辑和关系运算，关系运算结果为1 0

select name,avg(score),sum(score<60) as gks from result group by name having gks>=2;

排序是放在having之后的，也就是取出结果集最后再进行排序

通过where子查询查出每个栏目id最大的商品的详细信息

select goods\_name,goods\_number from goods where goods\_id in (select max(g

oods\_id) from goods group by cat\_id);

查出各栏目中最贵商品的名称，价格

select cat\_id,goods\_name,shop\_price from goods where shop\_price in (select max(shop\_price) from goods group by cat\_id);

查询各栏目最新上架商品信息，也就是商品的add\_time最大，或者goods\_id最大的

select cat\_id,goods\_name,shop\_price,from\_unixtime(add\_time) '上架时间' from goods where add\_time in (select max(add\_time) from goods group by cat\_id);

子查询的关键是设法查出记录的一个具体字段，然后通过外查询结合这个字段查出整条记录的详细信息

标量关系运算符>< = 单值

in 集合

from 表

from型子查询

select goods\_id,cat\_id,goods\_name from goods order by cat\_id,goods\_id desc;

select \* from (select goods\_id,cat\_id,goods\_name from goods order by cat\_id,goods\_id desc) as temp group by cat\_id;

注意:通过子查询出来的表必须要设置别名

对于group by 列名 需要注意的是：前面的select 部分必须是 列名 max()min()avg()sum(),因为分组就是把多条具有相同特性的记录压缩成一组，一组就只能有一个作为代表的字段值 所以只有用于分组的列名和分组统计结构可以作为此字段值

如果指定给select的是一个非法字段值，那么会取出该字段自然排序第一个值，然而这并不符合sql标准

深入理解exists子查询（要用到两张表）

把外层sql的结果拿到内层sql逐一匹配，符合条件，则把此外层结果去取出

建议张栏目表

create table category(

cat\_id int AUTO\_INCREMENT primary key,

cat\_name varchar(20) not null default ''

)engine myisam charset utf8;

insert into category select cat\_id,cat\_name from ecshop.ecs\_category;

select cat\_id,count(\*) from goods group by cat\_id;

以上有14个栏目，先要求只取出有商品的栏目，因为有的栏目用来作为父类，用于包括子类了

select \* from category where exists (select \* from goods where category.cat\_id=cat\_id);

注意:exists后面的子查询如果取出数据，返回真，取出外查询相应记录

所以 select \* 或者任意字段都是可以的 关键是能取出数据 where部分顺序随意

select \* from 大表 where exists (select \* from 小表);

要从大表中取出小表中有的所有记录

奇怪的null

为什么不希望记录某字段的值为null呢？

答：null是空，什么都没有 比较时很麻烦，

null的比较用 is null 和 is not null

碰到运算符一律返回null，效率不高，影响提高索引效果

所以建表时 not null default ''/0

新手1+N模式

第一次查询：查出一个数组

N次查询：在循环数组中，在查另外一张表，取出追加到 $row['cat\_name']=$cat['cat\_name']

header('content-type:text/html;charset=utf-8');

笛卡尔积：集合的元素，做两两组合组成一个新的大的集合

表与集合的关系：

一张表就是一个集合

每一行就是集合的一个元素

mysql内部每一行是不可能重复的，有个rowid

create table test2(

id int AUTO\_INCREMENT primary key,

sname varchar(20)

)engine myisam charset utf8;

drop table if exists test3;

create table test3(

id int auto\_increment primary key,

cname varchar(20),

cat\_id int

)engine myisam charset utf8;

insert into test2 values(1,'云彩'),(2,'月亮'),(3,'星星');

insert into test3 values(null,'猴子',95),(null,'老虎',96),(null,'大象',97);

每条记录就是集合的一个元素，记录之间的组合，就是元素之间的组合

内连接 通过两个表的id连接，匹配上才连接上，连接上后在取出记录

select \* from test2 inner join test3 on test2.id=test3.id;

左连接

select \* from test2 left join test3 on test2.id=test3.id;

右连接

select \* from test2 right join test3 on test2.id=test3.id;

用连接来查询商品表和栏目表中 cat\_id 和 cat\_name之间的对应关系

select g.cat\_id,cat\_name,goods\_name,shop\_price from goods as g left join category as cat on g.cat\_id=cat.cat\_id;

select \* from test2,test3;

以上 两表全相乘 两表中的记录两两组合，结果集字段为两表字段的并集

也可以取出两表相乘的部分字段

select test2.sname,test3.cname from test2,test3;

带条件的多表查询，相当于内连接

select \* from test2,test3 where test2.id=test3.id;

由于test2和test3有相同的字段，因此可以使用自然连接，他会自动将相同的字段作为连接条件

select \* from test2 natural join test3;

select \* from test2 natural join test3 having id<4 order by id desc ;

ambiguous意思：模糊不清的，字段模糊，此时应指明 表名

对于连接中的条件用 on 而不用 where

因为on是针对连接的，符合条件才进行连接，而where则是先连接成一个笛卡尔积再根据where后的条件进行过滤，内存中放一个笛卡尔积大表是极其浪费资源的

这样交叉连接得出的也是笛卡尔积

select \* from test2 cross join test3;

select \* from test2 join test3;

select \* from test2,test3;

能用左连接尽量用左连接，左连接可以用到索引，速度更快；多表查询并不好，索引用不到，速度慢

取出第四栏目下的商品，以及商品栏目名

select goods\_name,goods\_id,goods.cat\_id,cat.cat\_name from goods left join category as cat on goods.cat\_id = cat.cat\_id where goods.cat\_id=4;

除了可以通过where来限制，还可以直接用having对前面的结果集进行筛选，筛选后可以排序

select goods\_name,goods\_id,goods.cat\_id,cat.cat\_name from goods left join category as cat on goods.cat\_id = cat.cat\_id having cat\_id=4;

select goods\_name,goods\_id,goods.cat\_id,cat.cat\_name from goods left join category as cat on goods.cat\_id = cat.cat\_id having cat\_id=4 order by goods\_id desc limit 2;

取出表的全部字段，using默认去重

select test2.\*,test3.\* from test2 left join test3 using(id);

使用using和on做条件时，同样连接显示结果不一样，对我们取数据没影响

select \* from test2 right join test3 using(id);

select \* from test2 right join test3 on test3.id=test2.id;

注意：左右连接 并不是单纯的表示保留哪一边，而是以哪一边的表作为基准保留

左右连接可以互相转换，可以改变两个表的位置或者使用 left / right 关键字 ， 为方便移植，尽量使用左连接

内连接是左右连接的交集

使用union实现全连接

(select test2.sname '表二姓名',test3.cname from test2 left join test3 using(id))

union

(select test2.id ,test3.cname from test2 right join test3 on test3.id=test2.id);

字段数不一致产生错误 因为using会压缩相同字段 id

The used SELECT statements have a different number of columns

使用union是一定要确保字段数和字段类型上下一致 不一致的话可能出错或者类型转换

(select \* from test2 left join test3 on test3.id=test2.id)

union

(select \* from test2 right join test3 on test3.id=test2.id);

union产生的结果集的字段以 第一个select子句的字段为准

经典连接面试题讲解

建表

create table m(

mid int ,

hid int,

gid int,

mres varchar(10),

matime date

)engine myisam charset utf8;

create table t (

tid int,

tname varchar(20)

)engine myisam charset utf8;

insert into m

values

(1,1,2,'2:0','2006-05-21'),

(2,2,3,'1:2','2006-06-21'),

(3,3,1,'2:5','2006-06-25'),

(4,2,1,'3:2','2006-07-21');

insert into t

values

(1,'国安'),

(2,'申花'),

(3,'公益联队');

如何查出对阵，结果，时间

对同一张表连接两次，每次取出相同字段

select t1.tname hometeam,t2.tname guestteam,mres,matime from m letf join t t1 on hid=t1.tid left join t t2 on gid=t2.tid;

对于不能区分的字段，一定要用表名 如果设同一张表，则要设置别名 as 可以省略

select t1.tname hometeam,mres,t2.tname guestteam,matime from m letf join t t1 on hid=t1.tid left join t t2 on gid=t2.tid having month(matime)=6;#取出六月份比赛结果

select t1.tname hometeam,t2.tname guestteam,mres,matime from m letf join t t1 on hid=t1.tid left join t t2 on gid=t2.tid ;

合并数组

array array\_merge ( array $array1 [, array $... ] )

array\_merge() 将一个或多个数组的单元合并起来，一个数组中的值附加在前一个数组的后面。返回作为结果的数组。

大战乱码专题

ascii 0-127表示，因为键盘上所有字符加起来总数为127

一个字节八位，可以有256种组合，所以只需01111111 127中组合，一个字节就够存放所有的ascii字符了

但是到了其他国家，比如中国，有很多汉字，

于是用两个字节 2^16=65536 255^2=65536

有六万多种组合，够了

像这样，用两个字节表示的一个中文汉字，就是GB2312字符集

比如：中[202,197]

但是

中[202,197] [69 197]这里的69是整个理解，还是理解为ascii对应的字符呢？单字节<127的值，就有歧义，如果严格两两绑定理解为一个汉字。那么gb2312就不能显示英文了。

如何实现兼容ascii和双字节表示中文呢？

ascii 0-127 0xxx xxxx

干脆gb2312 完全从129-255之间开始

130 134 65 95 142 152

但是中文组合有少了，只有一万+

实际上gb2312只收录了6763个

后来出现了gbk，完全兼容gb2312

gbk也是两个字节表示一个汉字，其特点是，它的第二字节，地位，不在局限于129-255，<127也能用 这样就能表示更多字符

gbk中收录了两万+汉字

所以gbk在碰到>128的，就再往后找一个字节，两字节理解成一个汉字

但但是，到了日本 就成了 jis

如果要显示世界各国的字符呢？

于是就有了 Unicode 万国码表

Unicode用四个字节，来编号 40多亿，天文数字，足够用了

但我们常用的集中在前65535个标号里，因此，两字节就够了

但是Unicode只负责编号，而且4个字节都用来分配编号

把高位浪费的0值，用一定的规则舍弃，形成的编码方式 ， 比较出名的是utf-8

Unicode transform format utf

unicode 与 utf-8 的关系就是源文件和压缩文件之间的关系，utf-8 用于网络上传输

这两者是编码和解码的关系，因此可以相互转换

utf-8占几个字节呢?

答：utf-8 占1-6个字节 是变长的

Unicode中最高位有n个1就截 取n个字节

如何截取多国语言无乱码？

答：从头开始，取一个字节，通过位运算，计算连续的1的个数

gbk是怎样转成utf-8的？

答：gbk->unicode ->utf-8

anscii表示本地字符集 中国 gbk

日本 jis

乱码是怎样形成的？

答：1.解码时与实际编码不一致

2.传输中，编码不一致导致字节丢失

比如：utf-8 转为 gbk 大转小 必然导致字节丢失

utf-8的bom问题

bom也就是文件开头的记录编码的信息，注意，在cookie和session输出前不能有任何输出，包括bom头，

utf-8+表示带有bom头的utf-8

面试题：中文截取无乱码