目录

[Mysql day04 1](#_Toc500949226)

[1 函数 1](#_Toc500949227)

[2 字符串函数 2](#_Toc500949228)

[3 数字函数 4](#_Toc500949229)

[4 日期函数 5](#_Toc500949230)

[5 null值处理函数 8](#_Toc500949231)

[6 加密函数 8](#_Toc500949232)

[7 分支 9](#_Toc500949233)

[8 多行函数 10](#_Toc500949234)

[8.1 group子句,分组求多行函数 11](#_Toc500949235)

[8.2 having子句,对分组多行函数结果进行过滤 12](#_Toc500949236)

[9 mysql分页查询 13](#_Toc500949237)

[10 作业 17](#_Toc500949238)

Mysql day04

# 函数

* 字符串处理函数
* 数字处理函数
* 日期处理函数
* null值处理函数
* 加密函数
* 分支
* 多行函数

# 字符串函数

char\_length('a中') - 字符数

length('a中') - 字节数

concat('a','b','cde','fff') - 字符串连接，其他数据库可用 || 连接字符串，'abc' || 'def'

concat\_ws(';','abc','def','ggg') - 用分隔符连接字符串

instr('abcdefgdef','def') - 返回第一个子串的位置，从1开始，找不到返回0

locate('abc', '---abc---abc---abc-') - 返回第一个子串的位置，从1开始，找不到返回0

locate('abc', '---abc---abc---abc-',5) - 从指定位置向后找

insert('abcdefghijkl',2, 11, '---') - 用子串取代从2位置开始的11个字符

lower('AdFfLJf') - 变为小写

upper('AdFfLJf') - 变为大写

left('AdFfLJf',3) - 返回最左边的三个字符

right('AdFfLJf',3) - 返回最右边的三个字符

lpad('abc', 8, '\*') - 左侧填充，指定长度比源字符串少，相当于left

rpad('abc', 8, '\*') - 右侧填充，指定长度比源字符串少，相当于left

trim(' a bc ') - 去除两端空格

substring('abcdefghijklmn', 3) - 从3位置开始的所有字符

substring('abcdefghijklmn', 3, 6) - 从3位置开始的6个字符

repeat('abc', 3) - 重复三遍abc

REPLACE('Hello MySql','My','Your') - 子串替换

REVERSE('Hello') - 翻转字符串

SPACE(10) - 返回10个空格

字符串函数测试

set names gbk;

select length('a中'); -- 字节数

select char\_length('a中'); -- 字符数

select concat('abc','def','ghi'); -- 连接

select concat\_ws('/', 'abc','def','ghi'); -- 分隔符连接

select locate('bc', 'abc abc abc'); -- 找子串位置

select locate('bc', 'abc abc abc', 3); -- 从3向后找

select insert('abcdefghijk', 3, 5, '\*\*'); -- 替换从3开始的5个

select lower('AbCd'); -- 小写

select upper('AbCd'); -- 大写

select left('abcde', 2); -- 左侧2个

select right('abcde', 2); -- 右侧2个

select lpad('abc', 8, '\*'); -- 左侧填充

select rpad('abc', 8, '\*'); -- 右侧填充

select lpad('abc', 2, '\*'); -- 截断

select rpad('abc', 2, '\*');

select substring('abcde', 3); -- 截取3到末尾

select substring('abcde', 3,2); -- 截取3开始的2个

select replace('abc abc abc','bc','-'); -- bc替换成减号

字符串函数查询测试

use hr;

-- 1. first\_name和last\_name首字母相同

select employee\_id,first\_name,last\_name

from employees

where left(first\_name,1)=left(last\_name,1);

-- 2. first\_name和last\_name长度相同

select employee\_id,first\_name,last\_name

from employees

where

char\_length(first\_name)=char\_length(last\_name);

-- 3. email不是用first\_name首字母+last\_name

select employee\_id,first\_name,last\_name,email

from employees

where

concat(left(first\_name,1), replace(last\_name,' ', ''))<>email;

-- 4. 连接姓名,空格居中显示

select

concat(lpad(first\_name,20,' '),' ',last\_name) name

from

employees;

# 数字函数

ceil(3.14) - 天花板,向上取整

floor(3.94) - 地板,向下取整

format(391.536, 2) - 数字格式化为字符串，###,###.###，四舍五入，第二个参数为小数位数

round(673.4974) - 四舍五入

round(673.4974, 2) - 四舍五入到小数点后两位

round(673.4974, -2) - 四舍五入到百

TRUNCATE(234.31, 1) - 舍去至小数点后1位

数字函数测试

select ceil(3.14);

select ceil(-3.14);

select round(52342.6675, 2);

select round(52342.6675, -2);

select truncate(52342.6675, 2);

select format(52342.6675, 2);

用数字函数查询

-- 5. 涨工资10.41%,向上取整到十位

select employee\_id,first\_name,

ceil(salary\*1.1041/10)\*10 sal

from employees;

# 日期函数

* 日期类型数据与字符串类型可以自动相互转换
* 自动转换日期必须符合格式: %Y-%m-%d

NOW()   返回当前的日期和时间

        CURDATE()   返回当前的日期

        CURTIME()   返回当前的时间

        DATE(时间)   提取日期或日期/时间表达式的日期部分

        TIME(时间)    提取日期或日期/时间表达式的时间部分

        EXTRACT(字段 From 日期)   返回日期/时间按的单独部分

            字段的合法值：

                  MICROSECOND

                  SECOND

                  MINUTE

                  HOUR

                  DAY

                  WEEK

                  MONTH

                  QUARTER

                  YEAR

                  SECOND\_MICROSECOND

                  MINUTE\_MICROSECOND

                  MINUTE\_SECOND

                  HOUR\_MICROSECOND

                  HOUR\_SECOND

                  HOUR\_MINUTE

                  DAY\_MICROSECOND

                  DAY\_SECOND

                  DAY\_MINUTE

                  DAY\_HOUR

                  YEAR\_MONTH

        DATE\_ADD(日期, INTERVAL 数量 字段)   给日期添加指定的时间间隔

            字段的合法值同上

        DATE\_SUB(日期, INTERVAL 数量 字段)   从日期减去指定的时间间隔

        DATEDIFF(日期1, 日期2)   返回两个日期之间的天数

        DATE\_FORMAT(日期, 格式)   用不同的格式显示日期/时间

            格式字符：  %Y-%m-%d %H:%i:%s

                        %d/%m/%Y

                        %Y年%m月%d日

                    %a  缩写星期名

                    %b  缩写月名

                    %c  月，数值

                    %D  带有英文前缀的月中的天

                    %d  月的天，数值(00-31)

                    %e  月的天，数值(0-31)

                    %f  微秒

                    %H  小时 (00-23)

                    %h  小时 (01-12)

                    %I  小时 (01-12)

                    %i  分钟，数值(00-59)

                    %j  年的天 (001-366)

                    %k  小时 (0-23)

                    %l  小时 (1-12)

                    %M  月名

                    %m  月，数值(00-12)

                    %p  AM 或 PM

                    %r  时间，12-小时（hh:mm:ss AM 或 PM）

                    %S  秒(00-59)

                    %s  秒(00-59)

                    %T  时间, 24-小时 (hh:mm:ss)

                    %U  周 (00-53) 星期日是一周的第一天

                    %u  周 (00-53) 星期一是一周的第一天

                    %V  周 (01-53) 星期日是一周的第一天，与 %X 使用

                    %v  周 (01-53) 星期一是一周的第一天，与 %x 使用

                    %W  星期名

                    %w  周的天 （0=星期日, 6=星期六）

                    %X  年，其中的星期日是周的第一天，4 位，与 %V 使用

                    %x  年，其中的星期一是周的第一天，4 位，与 %v 使用

                    %Y  年，4 位

                    %y  年，2 位

        LAST\_DAY(日期) - 返回当月最后一天

日期函数测试

select now(); -- 当前年月日时分秒

select curdate(); -- 当前年月日

select curtime(); -- 当前时分秒

select date(now()); -- 抽取年月日

select time(now()); -- 抽取时分秒

select extract(year from now());-- 抽取年

select extract(day from now());-- 抽取日

select date\_add(now(), interval 10 year); -- 加10年

select date\_add(now(), interval -10 year); -- 减10年

select datediff(now(), '1993-5-12'); -- 相差天数

select datediff('1993-5-12', now());

select

date\_format(now(),' %Y-%m-%d %H:%i:%s'); -- 格式化

用日期函数查询

-- 6. 入职30年以上的员工

select employee\_id,first\_name,salary,

hire\_date

from employees

where

-- hire\_date<

-- date\_add(now(),interval -30 year);

datediff(now(), hire\_date)>30\*365+7;

-- 7. 1997年上半年入职的人

select employee\_id,first\_name,salary,

hire\_date

from employees

where

-- hire\_date between '1997-1-1' and '1997-6-30';

extract(year from hire\_date)=1997 and extract(month from hire\_date)<7;

-- 8. 不论哪一年，所有1月份入职的人

select employee\_id,first\_name,salary,

hire\_date

from employees

where

extract(month from hire\_date)=1;

# null值处理函数

        IFNULL(数据1,数据2) - 数据1是null返回数据2；不是null返回数据1

        coalesce(数据1,数据2,数据3,......) - 从左向右第一个不是null的数据

用null值函数查询

-- 9. 求年薪,乘提成比例

select employee\_id,first\_name,

salary\*12\*(1+ifnull(commission\_pct, 0)) sal

from employees;

-- 10. 没有主管显示'老板'

select employee\_id,first\_name,salary,

coalesce(manager\_id, '老板')

from employees;

# 加密函数

md5() md5加密,常用来对密码加密

加密结果是32位16进制数字字符

sha() 比md5更安全的加密方式

加密函数测试

select md5('123');

select sha('123');

use jtdb;

insert into tb\_user(username, password)

values('zhangsan', '123456');

select \* from tb\_user;

insert into tb\_user(username, password)

values('lisi', md5('123456'));

select \* from tb\_user;

# 分支

if(1, 2, 3) 判断条件1,成立执行2,否则执行3

case

when 条件 then ...

when 条件 then ...

else

end

case 数据

when 值1 then ...

when 值2 then ...

else

end

分支测试

select if(rand()<0.5, 1, 2);

select case floor(rand()\*4)

when 0 then '分支1'

when 1 then '分支2'

when 2 then '分支3'

else '分支4'

end c;

用分支查询

-- 工资>10000,有钱

-- 工资>5000,小康

-- 工资>3000,温饱

-- 否则,穷B

select employee\_id,first\_name,salary,

case

when salary>10000 then '有钱'

when salary>5000 then '小康'

when salary>3000 then '温饱'

else '穷B'

end c

from employees

order by salary desc;

# 多行函数

多行数据,产生一个计算结果

* count() 行数
* sum() 求和
* avg() 求平均
* max() 求最大值
* min() 求最小值
* 多行函数不能和其它字段一起查询  
  select avg(salary)~~, first\_name~~

from ...

上面查询mysql 可以执行,会把第一条数据的first\_name查询出来

其它数据库中执行,会报错

* 多个多行函数可以一起查询  
  select count(id),max(id),min(id),avg(id),sum(id)
* 多行函数会忽略null值
* count(\*) 求行数  
  当数据量非常大时(千万),count(\*)效率很低

## group子句,分组求多行函数

* 按指定的字段分组,每一组数据分别求多行函数
* 分组后,分组字段可以和多行函数一起查询
* 只有求多行函数时,才做分组查询

select 多行函数, 分组字段

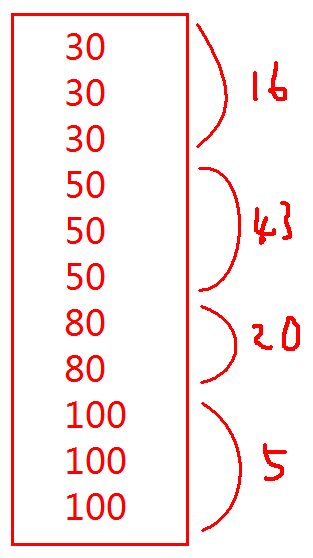
from ...

where ...

group by 分组字段

group by a,b

以字段的组合进行分组



## having子句,对分组多行函数结果进行过滤

求只有1个人的部门,先分组求人数,在按人数过滤

having子句

select ...count(\*)

where ...

group by a

having count(\*)=1

* having必须跟在group by之后
* 普通条件用 where 过滤,多行函数用having过滤
* having可以使用别名

# mysql分页查询

* 用count(\*)/每页数量,求出页数

select ... limit 10

前10条

select ... limit 0,10

第一条(0)开始的10条

select ... limit 10,10

第11行(10)开始的10条

select ... limit 20,10

第21行(20)开始的10条

第page页数据:

select ... limit (page-1)\*10, 10

多行函数测试

use hr;

select min(salary)~~, first\_name~~

from employees;

select

count(salary),

sum(salary),

avg(salary),

max(salary),

min(salary)

from employees;

多行函数查询

-- 11. 对提成计数,求平均

select count(commission\_pct),

avg(commission\_pct),

sum(commission\_pct)/ count(commission\_pct),

avg(ifnull(commission\_pct,0))

from employees;

-- 12. 有多少个部门,有多少个主管

select

count(distinct department\_id),

count(distinct manager\_id)

from

employees;

-- 13. 员工人数,表中的行数

select count(\*) from employees;

-- 14. 1997 年入职的人数

select count(\*) from employees

where

extract(year from hire\_date)=1997;

-- 15. 50 部门的人数

select count(\*) from employees

where department\_id=50;

-- 16. 最后进公司的人的入职时间

select max(hire\_date)~~, first\_name~~

from employees;

-- 17. 122号员工是个主管，他的手下有多少人

select count(\*) from employees

where manager\_id=122;

limit分页

select employee\_id,first\_name,salary

from employees

order by employee\_id

-- limit 10; -- 第一页

-- limit 10,10; -- 第二页

-- limit 20,10; -- 第三页

limit 100,10; -- 第11页

分组求多行函数

-- 18. 每个部门有多少人,并按人数排序

select count(\*) c,department\_id

from employees

where department\_id is not null

group by department\_id

order by c;

-- 19. 每种工作的平均工资 job\_id

select avg(salary),job\_id

from employees

group by job\_id;

-- 20. 每年的入职人数(提取年号进行分组)

select

count(\*),

extract(year from hire\_date) year

from employees

group by year;

-- 21. 每个部门中，每个主管的手下人数

-- 1 100 2

-- 1 120 5

-- 1 125 2

-- 2 100 5

-- 2 140 2

select

count(\*),

department\_id,manager\_id

from employees

where department\_id is not null and

manager\_id is not null

group by department\_id,manager\_id;

having过滤分组多行函数

-- 22. 只有一个人的部门,过滤null值

select

department\_id dept,count(\*) c

from employees

where department\_id is not null

group by dept

having c=1;

-- 23. 只有一个人入职的年份

select

extract(year from hire\_date) year,

count(\*) c

from employees

group by year

having c=1;

-- 24. 只有一个手下的主管id,过滤null值

select

manager\_id mgr, count(\*) c

from employees

where manager\_id is not null

group by mgr

having c=1;

-- 25. 平均工资大于10000的工作

select

job\_id job, avg(salary) sal

from employees

group by job

having sal>10000;

# 作业

订单表

create table tb\_order(

order\_id varchar(50) primary key,

user\_id bigint(20) not null,

payment Decimal(10,2),

payment\_type tinyint(1),

post\_fee Decimal(10,2),

status tinyint(2),

payment\_time datetime,

consign\_time datetime,

end\_time datetime,

close\_time datetime,

shipping\_name varchar(20),

shipping\_code varchar(20),

buyer\_message varchar(100),

buyer\_nick varchar(50),

buyer\_rate tinyint(1),

create\_time datetime,

update\_time datetime,

foreign key(user\_id) references tb\_user(id)

)engine=innodb charset=utf8;

订单详情

create table tb\_order\_item(

id varchar(20) primary key,

item\_id bigint(20) not null,

order\_id varchar(50) not null,

num int(10),

title varchar(200),

price Decimal(10,2),

total\_fee Decimal(10,2),

pic\_path varchar(200),

created datetime,

updated datetime,

foreign key(item\_id) references tb\_item(id),

foreign key(order\_id) references tb\_order(order\_id)

)engine=innodb charset=utf8;

1.       工作代码 job\_id 不是 IT\_PROG、SH\_CLERK、PU\_CLERK、ST\_CLERK、

2.       查询员工编号为 100,150,200,250 的员工

3.       查询 薪水小于3000的员工，和100部门所有的员工

4.       查询 薪水大于10000，且在100部门的 员工

5.       查询 员工 first\_name 是 Bruce 的员工

6.       薪水大于等于 10000  的员工，按姓名排序

4.2.4 商品规格参数表(tb\_item\_param)

笔记本规格模板

品牌: dell,lenovo,thinkpad...

内存: 2G,4G,8G,16G,24G,32G,4~8G

显卡: 独立显卡,集成显卡...

4.2.5 商品具体规格表 (tb\_item\_param\_item)

昭阳:

品牌: lenovo

内存: 4~8G

显卡: 独立显卡

4.2.7 内容分类表(tb\_content\_category)

4.2.6 内容表(tb\_content)