# Day08 MySQL语法

## 一.SQL

Struct query language,结构化查询语言.

## 二.数据库的SQL语句(‘增删改查’)

### 1.创建数据库

未指定编码,创建数据库:

create database [if not exists] db01;

指定编码,创建数据库:

create database if not exists db02 character set utf8

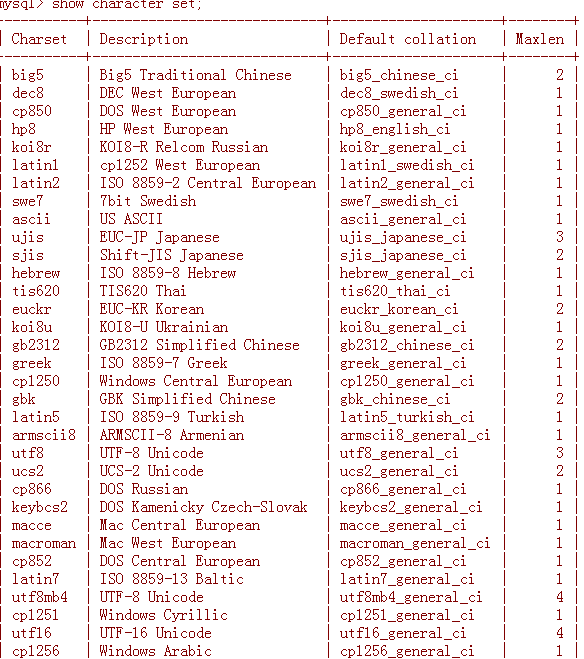
collate utf8\_general\_ci;

character set :指定编码,utf8,gbk

collate:校验规则

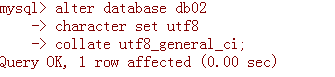
### 2.显示MySQL所支持的字符集

show character set;





3.修改数据库默认编码的命令



### 4.使用数据库

use 数据库名称;

例如:

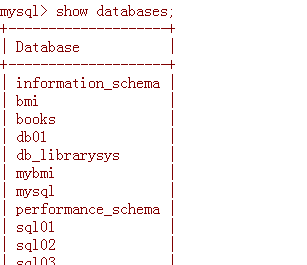
use db02;

### 5.删除数据库

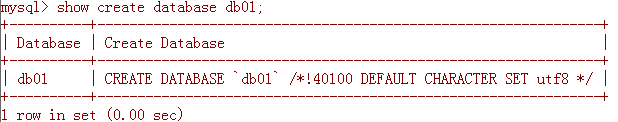
drop database db02;



### 6.显示有多少数据库



### 7.显示创建数据库的语句

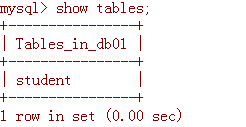


## 三.表的SQL语句(‘增删改查’)

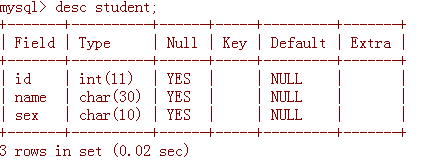
### 1.基本的建表语句



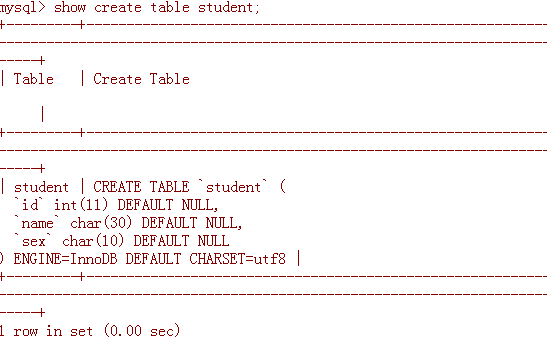
### 2.显示数据库中的所有表



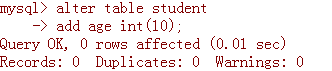
### 3.显示表的内部结构



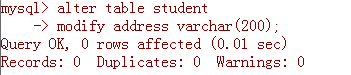
### 4.显示创建表的命令



### 5.修改表结构---添加新列(alter+ add)

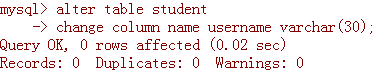


### 6.修改表结构---修改某一列的数据类型(alter+modify)

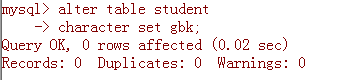


### 7.修改表结构---修改某一列的列名(alter+change column)

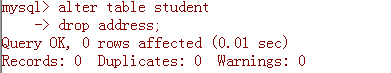
注意:change column 旧列名 新列名 新列的数据类型



### 8.修改表结构---修改表的默认编码(alter+character set)



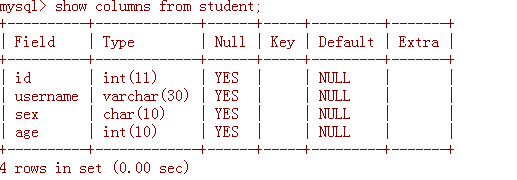
### 9.修改表结构---删除某一列(alter+drop)



### 10.修改表结构---修改表名(rename)



### 11.显示表中的列,效果等同于desc 表名



### 12.删除一个表



## 四.MySQL中字段的数据类型

二进制类型:bit

整型:intbigint,

文本类型:char,varchar, text(clob),blob(大二进制文本)

浮点型:float,double

日期时间类型:date,datetime,timestamp(时间戳)

布尔型:bool

## 五.SQL语句的类型

### 1.DDL(data define language),数据定义语言.

create/drop/truncate/alter....

规律:一般都是操作数据库或者表的结构的关键字,

基本上不涉及到具体的数据.

### 2.DML(data manipulation language),数据操作语言.

insert,delete,update,select ,特殊关键字:truncate(删除旧表,创建新表,但是表中的所有数据全都清空了).

### 3.DQL(data query language),数据查询语言.

select/show

4.DCL(data control language),数据控制语言

.(站在整个数据库服务器的层面来操作的).

grant(授权),revoke(撤销授权)

## 六. CRUD:增删改查

C:create

R:read(requery)

U:update(---🡪联想alter)

D:delete(drop)

## 七.DML之Insert

insert into 表名 (列名1,列名2....) values(值1,值2...);

### 给表中的全部列都添加值

写法一:



写法二:



### 2.给表的部分列插入值



### 3.列的数量与值的数量不一致!



### 4.列的类型与值的类型不匹配



### 5.数据太长,数据值的长度要与列类型规定的长度相符合.



### 6. 值的顺序要与列的顺序一致



### 7.字符和日期型数据要用’ ’单引号括起来;

8.null问题

如果没有插入值,或者insert into ...values(null),列中的值都是null.

## 八.DML之Update

update 表名 set 列名=新值 where 条件;

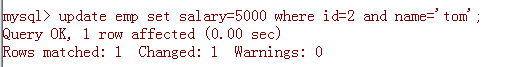
### 1.基本的修改



### 2.修改多个列(列之间用逗号隔开)



### 3.根据多个条件进行修改(and,or等符号)



### 4.修改语句中进行基本的运算(+,-,\*,/等)

将id=2的员工的薪资涨1000



## 九.DML之Delete

delete from 表名 where 条件;

### delete语句一般都要加where条件,否则会删除表中的全部数据.

### 2.delete是用来删除一行记录的,不能删除某一列的数据(如果要删除某一列的数据,用update);

### 3.delete与truncate的区别:

①.删除数据的方式不同:

delete是从上到下一行一行删除数据的;

truncate直接把整个表给删掉,然后再重现创建出来一个与之前的表完全一样的表(此时表中的数据已经没了);

②.delete删除掉的数是可以恢复的,truncate删除的数据无法恢复;

③.delete删除数据慢,truncate删除数据快.

案例:

1.按照条件来删除:



2.删除所有数据:



3.truncate删除表:



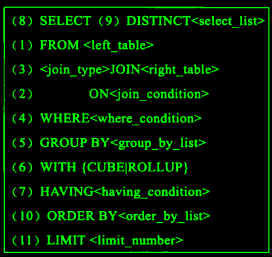
## 十.DQL之Select

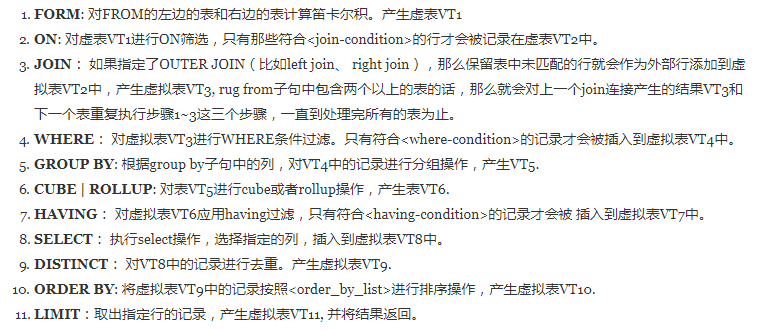
### 1.基本语法

select 列名 from 表名 where 条件 order by 列名 limit 0,5

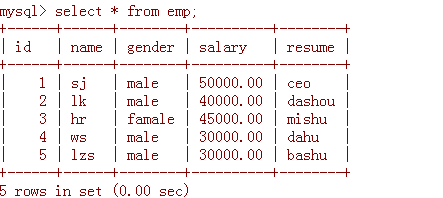
select \* from 表名;

### 2.SQL语句的执行顺序





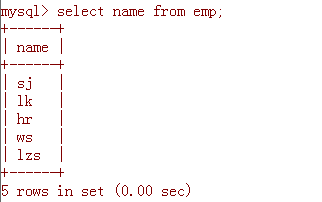
### 3.查询全部



写法二:

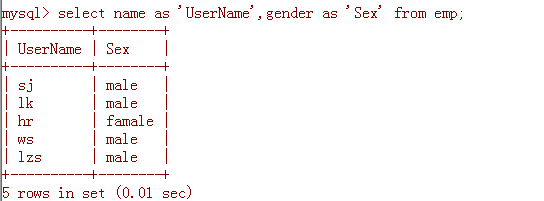
select id,name,gender,salary,resume from emp;

### 4.查询指定字段

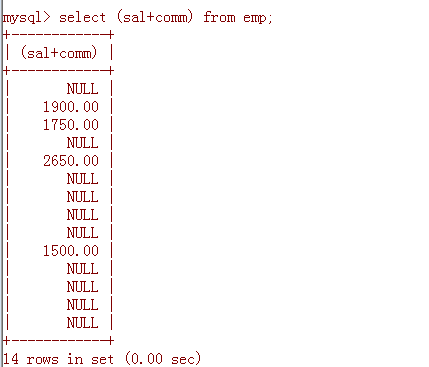


### 5.AS关键字指定列的别名

as可以给虚拟表中的列和表来指定别名!

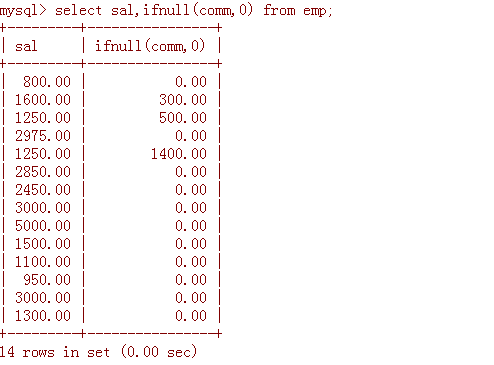


### 6.合并列

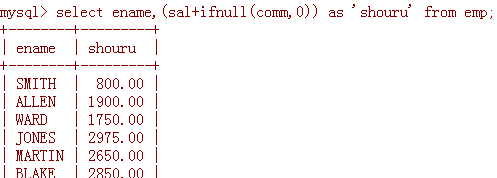


### 7.IFNULL(列名,0)函数

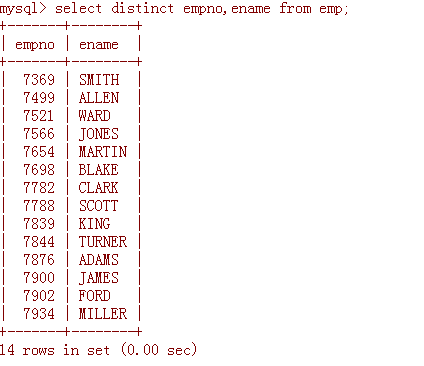
如果某一列为null,则用0来替换null值.



案例:查询公司中所有员工的收入.



### 8.消除重复数据,distinct



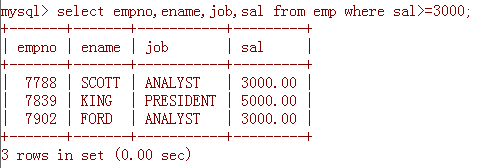
注意:distinct必须在列名的前面!

### 9.where条件查询

条件查询时的运算符:

#### 9.1 基本符号

>,>=,<,<=,=,!=(不等于),<>(不等于)



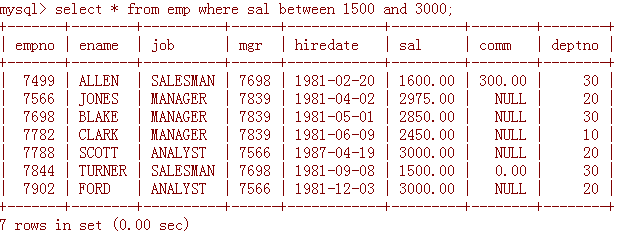
#### 9.2 between...and

案例:查询sal薪资范围是1500到3000之间的员工信息.

写法一:>=,<=

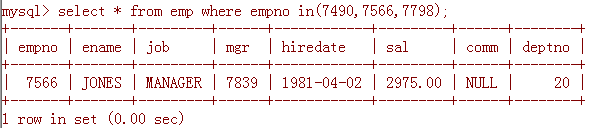


写法二:between...and...



#### 9.3 in(set)

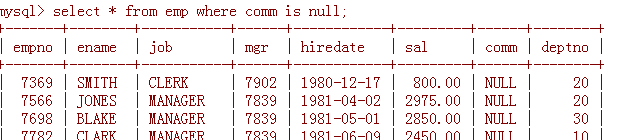
在一个集合范围内查询满足条件的结果.



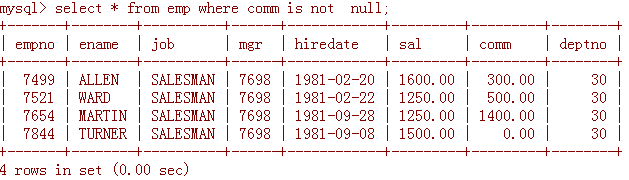
#### 9.4 is null与is not null

判断是否为空.

is null:

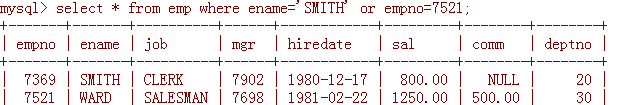


is not null:



#### 9.5 and or not,与或非

or:



and:



### 10.like关键字实现模糊查询

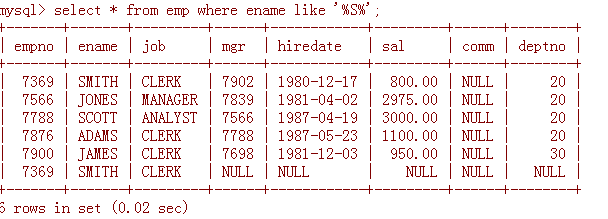
模糊查询的基本语法:

select 列 from 表 where 列 like ‘%....’;

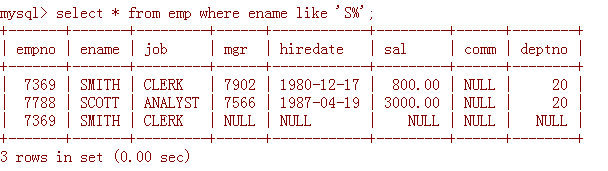
%:代表任意0个,1个或多个字符

\_(下划线):代表任意一个字符.

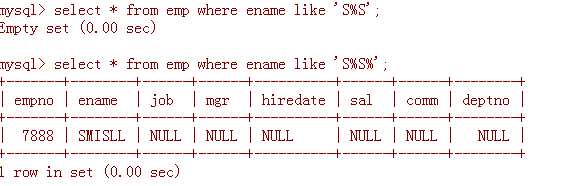
案例:查询姓名中包含S的人的信息.



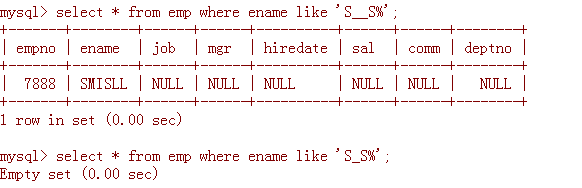
案例:查询姓名以S开头的人的信息.



案例:查询姓名中包含2个S的人的信息.



案例:查询姓名中以S开头,并且与姓名中的另一个S隔2个字符的人的信息.



### 11.Order by排序(在select执行顺序中排第10位)

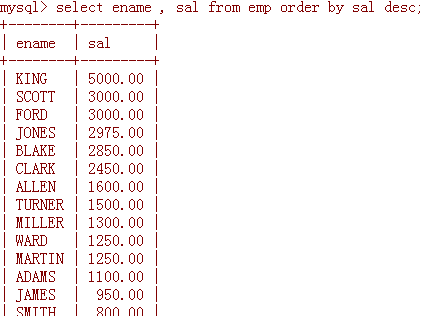
注意:order by是对查询结果的排序,所以如果没有查询出结果的时候,order by时不执行的.

order by 要排序的列名 asc|desc;

asc:升序,默认;

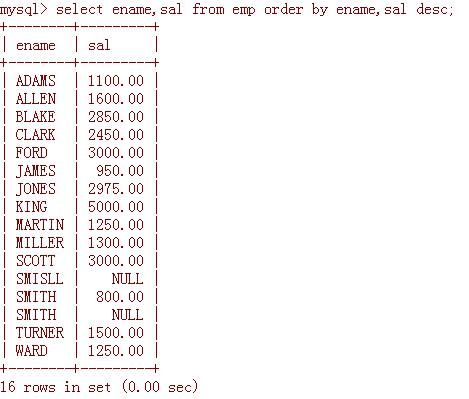
desc:降序.

案例:查询所有员工的薪资,并且降序排列.



注意:order by可以按照多列排序!

案例二:按照姓名和薪资排序.



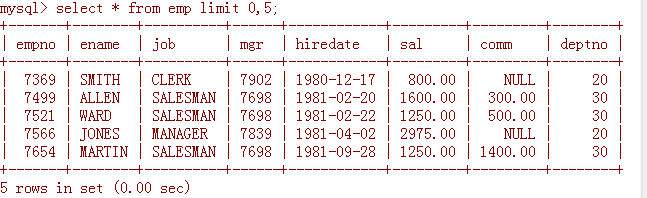
### 12.limit分页查询(在select中排第11位,最后才执行)

select 列 from 表 limit index,length;

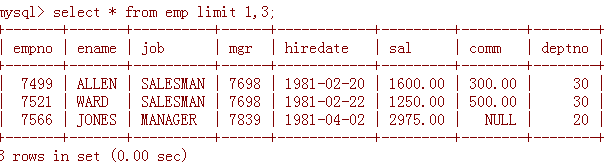
index:分页的起始索引值;

length:每次分页的记录的条数.

案例:查询emp表,每次输出5条记录

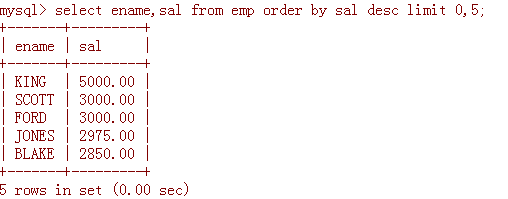


案例二:查询emp表,从第二条记录开始,输出3条记录.



案例三:

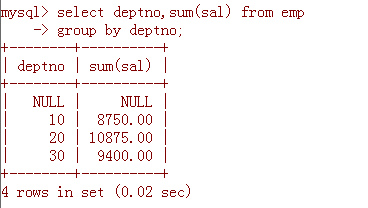
将order by 与limit结合使用.



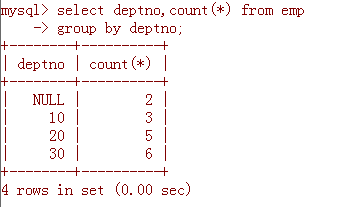
### 13.group by 分组(在select中排第5位)

select 列 from 表 group by 列名;

案例:统计emp表中每个部门的薪资总和.



案例:统计每个部门的员工数量.



案例三: 统计每个部门薪资高于1500的员工数量.

1. .统计员工数量:

select count(\*) from emp;

1. .查询有哪些部门:

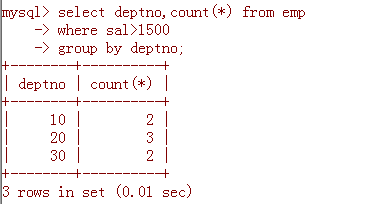
select deptno from emp;

1. .统计每个部门的员工数量:

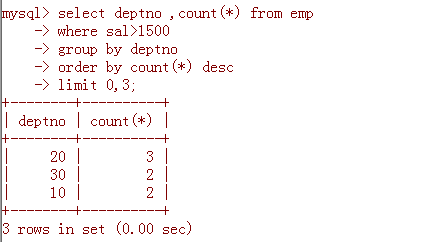
select deptno ,count(\*) from emp group by deptno;

1. .select deptno,count(\*) from emp where sal>1500

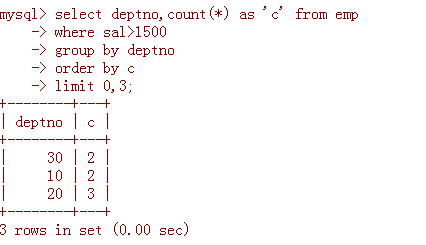
group by deptno;



案例四: 统计每个部门薪资高于1500的员工数量,并且按照数量降序排列,分页输出.



案例五:结合别名使用



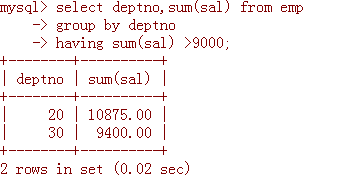
### 14.having短语(在select中排第7位).

select 列 from 表 group by 列 having 条件;

注意:where不可以和聚合函数一起使用;

having可以与聚合函数配合使用!

案例:统计每个部门的薪资总和大于8000的部门有哪些.



### 注意:

where与having的异同点.

相同点:都属于过滤条件.

不同点:

1. .书写位置和执行位置不同:where在group by执行书写和执行;having是在group by之后来书写和执行的.
2. where不能与聚合函数一起使用;having可以与聚合函数一起使用.

## 十一. 聚合函数

sum():求和;

count():计数;

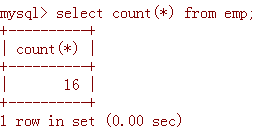
max():求最大值;

min():求最小值;

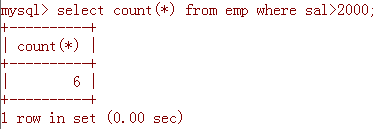
avg():求平均值.

### 1.count()函数

案例一:统计员工数量

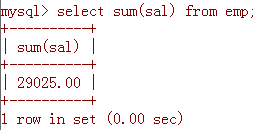


案例二:统计薪资高于2000的员工数量.

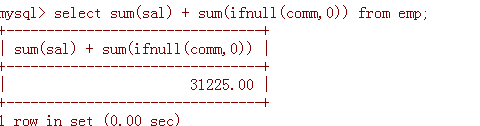


### sum()函数

案例:统计公司的底薪总和.

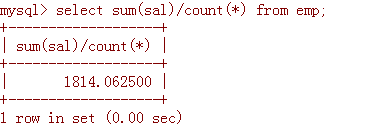


案例:统计公司的薪资(底薪+绩效)总和.

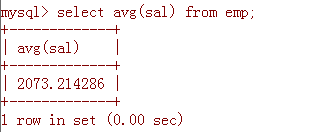


### avg()函数

案例:统计公司的平均收入

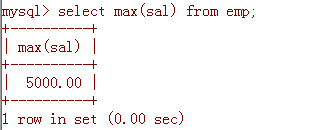


案例二:用avg()来实现.



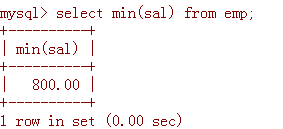
### max()函数

统计最高薪资.



### 5.min()函数

统计最低薪资



## 十二. MySQL常用的内置函数

### 日期时间类的函数

获取当前日前: current\_date()

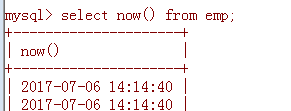
获取当前时间:current\_time()

获取时间戳:current\_timestamp();

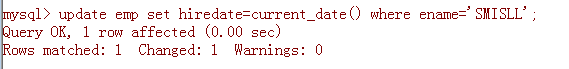
获取当前时间:now();

获取指定时间值的日期部分:date(时间值)

案例一:查询当前时间



案例二:将姓名为SMISLL的员工入职日期修改为当前日期.



### 字符串的函数

拼接字符串:concat(str1,str2,str3.....)

转为大写字母的方法:ucase(str);

转为小写字母的方法:lcase(str);

去掉左右两侧的空格:trim(str)

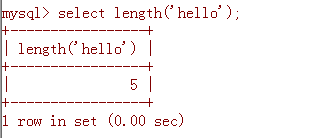
去掉左侧的空格:ltrim(str)

去掉右侧的空格:rtrim(str)

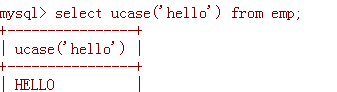
截取子串:substring(str,position,length)

获取字符串的长度:length(str)

案例:获取字符串长度



案例:小写转大写



### 数学类的函数

取绝对值:abs(num)

向下取整:floor(num)

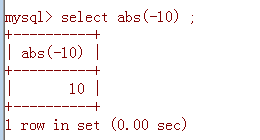
向上取整:ceiling(num)

随机数:rand(seed)

format(num,位数):格式化数字,比如将一个数字保留为几位小数点.

mod(num,operator):取余数

案例:对负数取绝对值



案例:取随机数

