Neuedu

东软睿道教育产品解决方案-云计算与大数据

MySQL数据库开发技术

----- 常用函数

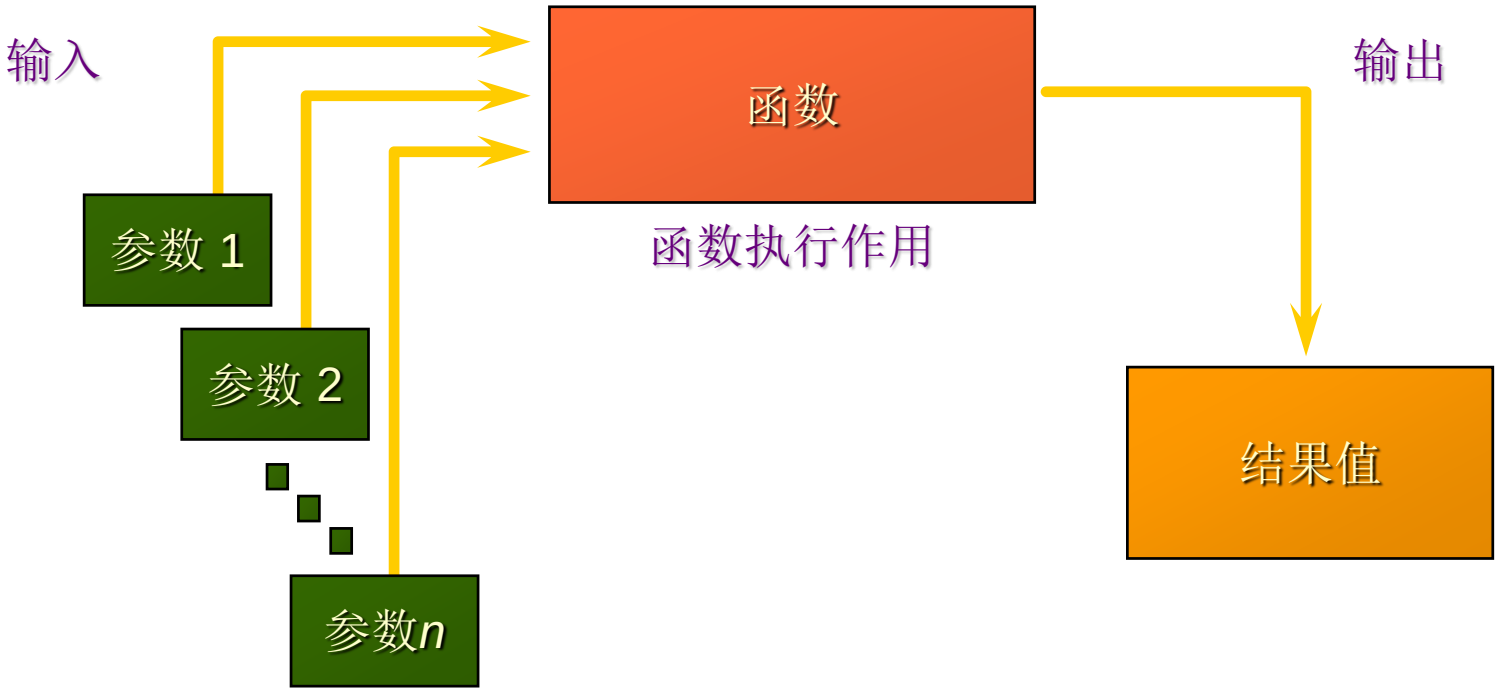
V1.0

本章内容

节	知识点	掌握程度	难易程度
MySQL函数概述	MySQL函数概述	理解	
	单行函数语法和特征	理解	
	函数分类	了解	
数学函数	绝对值、平方、取余函数	了解	
	截断、四舍五入函数	掌握	
	随机数函数	掌握	
	其他算术运算函数	了解	
字符串函数	大小写转换函数	掌握	
	字符串操作函数	掌握	
日期和时间函数	获取日期和时间函数	掌握	
	日期和时间运算函数	掌握	
	日期和时间格式化函数	掌握	
流程控制函数	CASE语句	掌握	
	IF语句	掌握	
	IFNULL、NULLIF函数	掌握	
其他函数	取得数据库、版本、用户信息函数	了解	
	取得网址及编码函数	了解	

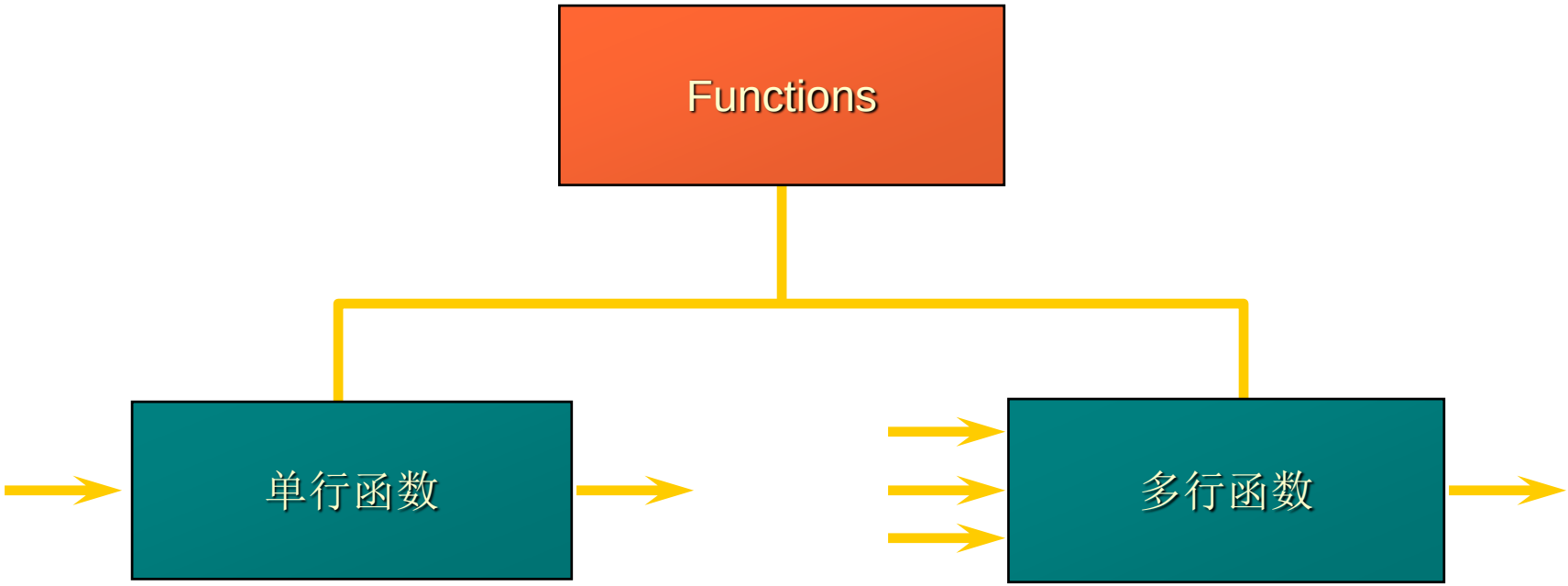
MySQL函数概述

- MySQL提供了很多功能强大、方便易用的函数，在进行数据库管理以及数据的查询和操作时，帮助我们提高对数据库的管理效率



MySQL函数概述

- 函数分类



MySQL函数概述

- 单行函数语法
  - 语法:

    函数名[(参数1，参数2,…)]
  - 其中的参数可以是以下之一:
    - 变量
    - 列名
    - 表达式

MySQL函数概述

- 单行函数特征
  - 单行函数对单行操作
  - 每行返回一个结果
  - 有可能返回值与原参数数据类型不一致
  - 单行函数可以写在SELECT、WHERE、ORDER BY子句中
  - 有些函数没有参数，有些函数包括一个或多个参数
  - 函数可以嵌套

MySQL函数概述

Neuedu

- 常用函数分类
  - 数学函数
  - 字符串函数
  - 日期和时间函数
  - 流程控制函数
  - 其他函数

数学函数

Neuedu

- ABS(x):返回x的绝对值;
- SQRT(x)： 返回非负数x的平方根;
- PI():返回圆周率;
- MOD(x,y)或%:返回x被y除的余数;
- CEIL(x)、CEILING(x)： 返回大于或者等于x的最小整数值;
- FLOOR(x)： 返回小于或者等于x的最大整数值;
- ROUND(x,y):返回保留小数点后面y位，四舍五入的整数;
- TRUNCATE(x,y):返回被舍弃的小数点后y位的数字x;
- RAND()： 每次产生不同的随机数;
- SIGN(x)： 返回参数的符号;

数学函数

Neuedu

- POW(x,y)和POWER(x,y)： 返回x的y次乘方的结果值；
- EXP(x)：返回以e为底的x乘方后的值；
- LOG(x)：返回x的自然对数，x相对于基数e的对数；
- LOG10(x)：返回x的基数为10的对数；
- RADIANS(x)：将参数x由角度转化为弧度；
- DEGREES(x)：将参数x由弧度转化为度。
- SIN(x)：返回x正弦，其中x为弧度值；
- ASIN(x)返回x的反正弦，即正弦为x的值；
- COS(x)：返回x的余弦；
- ACOS(x)：返回x反余弦
- TAN(x)：返回x的正切；
- ATAN(x)返回x的反正切；

数学函数

Neuedu

- 绝对值函数：ABS(x)；返回圆周率的函数：PI()

```
mysql> SELECT ABS(2),ABS(-3.3),ABS(3.3),PI();
+---------+-----------+----------+----------+
| ABS(2)  | ABS(-3.3) | ABS(3.3) | PI()     |
+---------+-----------+----------+----------+
|       2 |       3.3 |      3.3 | 3.141593 |
+---------+-----------+----------+----------+
1 row in set (0.00 sec)
```

- 非负数的平方根函数：SQRT(x)

```
mysql> SELECT SQRT(9),SQRT(-9),SQRT(12);
+----------+-----------+--------------------+
| SQRT(9)  | SQRT(-9)  | SQRT(12)           |
+----------+-----------+--------------------+
|        3 |      NULL | 3.4641016151377544 |
+----------+-----------+--------------------+
1 row in set (0.02 sec)
```

- 求余函数：MOD(x,y)，x被y除后的余数

```
mysql> SELECT MOD(30,8),MOD(30.5,8),MOD(30.8,8);
+-----------+-------------+-------------+
| MOD(30,8) | MOD(30.5,8) | MOD(30.8,8) |
+-----------+-------------+-------------+
|         6 |         6.5 |         6.8 |
+-----------+-------------+-------------+
1 row in set (0.00 sec)
```

数学函数

Neuedu

- 获取整数的函数CEIL(x)、CEILING(x)：返回大于x的最小整数值；
- FLOOR(x)：返回小于x的最大整数值。

```
mysql> SELECT CEIL(-3.5),CEIL(3.5),CEILING(-3.5),CEILING(3.5),FLOOR(-3.5);
+------------+-----------+--------------+-------------+-------------+
| CEIL(-3.5) | CEIL(3.5) | CEILING(-3.5) | CEILING(3.5) | FLOOR(-3.5) |
+------------+-----------+--------------+-------------+-------------+
|         -3 |         4 |            -3 |           4 |          -4 |
+------------+-----------+--------------+-------------+-------------+
1 row in set (0.00 sec)
```

- 四舍五入函数ROUND(x)：只保留整数部分；

```
mysql> SELECT ROUND(-1.23),ROUND(1.23),ROUND(-1.56),ROUND(1.56);
+--------------+-------------+--------------+-------------+
| ROUND(-1.23) | ROUND(1.23) | ROUND(-1.56) | ROUND(1.56) |
+--------------+-------------+--------------+-------------+
|           -1 |           1 |           -2 |           2 |
+--------------+-------------+--------------+-------------+
1 row in set (0.00 sec)
```

练习1

- 1. 写一个查询，分别计算100. 456 四舍五入到小数点后第2位，第1位，整数位的值。
- 2. 写一个查询，分别计算100. 456 从小数点后第2位，第1位，整数位截断的值。

数学函数

Neuedu

- 四舍五入ROUND(x,y)：返回接近于x的值，保留小数点后面y位，若y为负数，就保留到小数点左边y位；

```
mysql> SELECT ROUND(-12.35,1),ROUND(12.34,0),ROUND(1.23,-1),ROUND(123.45,-2);
+-----------------+----------------+----------------+------------------+
| ROUND(-12.35,1) | ROUND(12.34,0) | ROUND(1.23,-1) | ROUND(123.45,-2) |
+-----------------+----------------+----------------+------------------+
|           -12.4 |             12 |              0 |              100 |
+-----------------+----------------+----------------+------------------+
1 row in set (0.00 sec)
```

- 截取函数TRUNCATE(x,y)：返回被舍弃的小数点后y位的数字x。

```
mysql> SELECT TRUNCATE(1.23,1),TRUNCATE(1.88,1),TRUNCATE(1.88,0),TRUNCATE(1.88,-
1);
+-----------------+-----------------+-----------------+------------------+
| TRUNCATE(1.23,1) | TRUNCATE(1.88,1) | TRUNCATE(1.88,0) | TRUNCATE(1.88,-1) |
+-----------------+-----------------+-----------------+------------------+
|             1.2 |             1.8 |               1 |                0 |
+-----------------+-----------------+-----------------+------------------+
1 row in set (0.00 sec)
```

数学函数

Neuedu

- 获取随机数的函数RAND()：每次产生不同的随机数；
- RAND(x)：返回一个范围在0~1之间的随机浮点数，x为随机种子

```
mysql> SELECT RAND(),RAND(),RAND(10),RAND(10)
    -> ;
+--------------------+--------------------+--------------------+--------------------+
| RAND()             | RAND()             | RAND(10)           | RAND(10)           |
+--------------------+--------------------+--------------------+--------------------+
| 0.43555609829309966 | 0.6776701423131583 | 0.6570515219653505 | 0.6570515219653505 |
+--------------------+--------------------+--------------------+--------------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT RAND(curtime()),RAND(CURRENT_TIME());
+-------------------------+-------------------------+
| RAND(curtime())         | RAND(CURRENT_TIME())    |
+-------------------------+-------------------------+
| 0.4913224386901804      | 0.4913224386901804      |
+-------------------------+-------------------------+
1 row in set (0.00 sec)
```

数学函数

Neuedu

- SIGN(x)返回参数的符号，x的值为负、零或正时返回结果依次为-1、0或1。

```
mysql> SELECT SIGN(-10),SIGN(10),SIGN(0);
+--------------+-------------+-----------+
| SIGN(-10)    | SIGN(10)    | SIGN(0)   |
+--------------+-------------+-----------+
|           -1 |           1 |         0 |
+--------------+-------------+-----------+
1 row in set (0.00 sec)
```

- 幂函数POW(x,y)和POWER(x,y)： 返回x的y次乘方的结果值；EXP(x)返回以e为底的x乘方后的值。

```
mysql> SELECT POW(2,2),POW(2,-2),POW(-2,3),POWER(2,2),POWER(2,-2),POWER(-2,3);
+----------+-----------+-----------+------------+-------------+-------------+
| POW(2,2) | POW(2,-2) | POW(-2,3) | POWER(2,2) | POWER(2,-2) | POWER(-2,3) |
+----------+-----------+-----------+------------+-------------+-------------+
|        4 |      0.25 |        -8 |          4 |        0.25 |          -8 |
+----------+-----------+-----------+------------+-------------+-------------+
1 row in set (0.01 sec)
```

数学函数

Neuedu

- LOG(x)返回x的自然对数，x相对于基数e的对数，对数定义域不能为负数；LOG10(x)返回x的基数为10的对数。

```
mysql> SELECT LOG(5),LOG(-5),LOG10(5),LOG10(-5);
+------------------+---------+--------------------+-----------+
| LOG(5)           | LOG(-5) | LOG10(5)           | LOG10(-5) |
+------------------+---------+--------------------+-----------+
| 1.6094379124341003 |    NULL | 0.69897000433601 89 |      NULL |
+------------------+---------+--------------------+-----------+
1 row in set (0.00 sec)
```

- RADIANS(x)将参数x由角度转化为弧度；DEGREES(x)将参数x由弧度转化为度。

```
mysql> SELECT RADIANS(180),DEGREES(PI());
+--------------------+------------------+
| RADIANS(180)       | DEGREES(PI())    |
+--------------------+------------------+
| 3.141592653589793  |              180 |
+--------------------+------------------+
1 row in set (0.00 sec)
```

数学函数

Neuedu

- SIN(x)返回x正弦，其中x为弧度值；ASIN(x)返回x的反正弦，即正弦为x的值；
- COS(x)返回x的余弦；ACOS(x)返回x反余弦,即余弦是x的值。若x不在-1到1的范围之内，则返回NULL；
- TAN(x)返回x的正切；ATAN(x)返回x的反正切，即正切为x的值。

```
mysql> SELECT SIN(PI()/2),ASIN(1),COS(PI()),ACOS(1);
+--------------+----------------------+-----------+----------+
| SIN(PI()/2)  | ASIN(1)              | COS(PI()) | ACOS(1)  |
+--------------+----------------------+-----------+----------+
|            1 | 1.5707963267948966 |        -1 |        0 |
+--------------+----------------------+-----------+----------+
1 row in set (0.00 sec)
```

字符串函数

- CHAR_LENGTH(str)：返回字符串str的所包含字符个数;
- LENGTH(str):返回字符串str的长度;
- CONCAT(s1,s2,...)： 字符串连接;
- CONCAT_WS(x,s1,s2,…)：字符串连接，x是其它参数的分隔符;
- INSERT(s1,x,len,s2) ：返回字符串s1，s1中插入字符串s2;
- LOWER (str)|LCASE (str)：将字符串全部转换成小写字母;
- UPPER(str)|UCASE(str)：将字符串全部转换成大写字母;
- LEFT(s,n)：返回最左边指定长度的字符;
- RIGHT(s,n)：返回最右边指定长度的字符;
- LPAD(s1,len,s2)| RPAD(s1,len,s2) ：填充字符串函数;
- TRIM(s1 FROM s)|LTRIM(s) |RTRIM(s):删除空格函数;

字符串函数

- REPEAT(s,n)：重复生成字符串函数；
- SPACE(n)：返回一个由n个空格组成的字符串；
- REPLACE（s，s1，s2）：字符串替换函数；
- STRCMP(s1,s2)：比较字符串大小函数；
- SUBSTRING(s,n,len)：获取子串函数；
- LOCATE(str1,str)|POSITION(str1 IN str)|INSTR(str, str1)：匹配子串开始位置函数；
- REVERSE(s)：将字符串s反转；
- ELT(N,字符串1,字符串2,字符串3,…)：返回指定位置函数；

字符串函数

- CHAR_LENGTH(str)：返回字符串str的所包含字符个数。一个多字节字符算作一个单字符。使用utf8编码时，1个汉字是2个字节，一个数字或字母算一个字节 。
- LENGTH（str）返回字符串的字节长度。

```
mysql> SELECT CHAR_LENGTH("MySQL"),CHAR_LENGTH("数据库"),LENGTH("数据库");
+-----------------------+-------------------------+--------------------+
| CHAR_LENGTH("MySQL") | CHAR_LENGTH("数据库") | LENGTH("数据库") |
+-----------------------+-------------------------+--------------------+
|                     5 |                       3 |                  6 |
+-----------------------+-------------------------+--------------------+
1 row in set (0.00 sec)
```

字符串函数

- 字符串连接CONCAT(s1,s2,…)函数：如果有参数为null，则返回null。
- CONCAT_WS(x,s1,s2,…)：第一个参数x是其它参数的分隔符，分隔符的位置放在要连接的两个字符串之间，忽略null。

```
mysql> SELECT CONCAT('MySQL','入门'),CONCAT('MySQL',null,'入门'),
    -> CONCAT_WS('*','MySQL','5.5','入门');
+------------------------+-----------------------------+----------------------
-------------+
| CONCAT('MySQL','入门') | CONCAT('MySQL',null,'入门') | CONCAT_WS('*','MySQL','
5.5','入门') |
+------------------------+-----------------------------+----------------------
-------------+
| MySQL入门              | NULL                        | MySQL*5.5*入门
             |
+------------------------+-----------------------------+----------------------
-------------+
1 row in set (0.00 sec)
```

字符串函数

- 插入字符串INSERT(s1,x,len,s2)函数：返回字符串s1，返回字符串s1，在位置x起始的子串且len个字符长的子串由字符串s2代替。 ，如果子串超过s1字符串长度，返回原串；若有参数为null，则返回null。

```
mysql> SELECT INSERT('MySQL',3,1,'DATA'),INSERT('MySQL',-3,4,'DATA');
+----------------------------+-----------------------------+
| INSERT('MySQL',3,1,'DATA') | INSERT('MySQL',-3,4,'DATA') |
+----------------------------+-----------------------------+
| MyDATAQL                   | MySQL                       |
+----------------------------+-----------------------------+
1 row in set (0.02 sec)
```

字符串函数

- LOWER（str)或者LCASE（str)将字符串str中的字母字符全部转换成小写字母。
- UPPER(str)或者UCASE(str)将字符串str中的字母字符全部转换成大写字母。

```
mysql> SELECT LOWER('MySQL'),LCASE('DATA'),UPPER('mysql'),UCASE('Data');
+---------------+---------------+---------------+---------------+
| LOWER('MySQL') | LCASE('DATA') | UPPER('mysql') | UCASE('Data') |
+---------------+---------------+---------------+---------------+
| mysql          | data          | MYSQL          | DATA          |
+---------------+---------------+---------------+---------------+
1 row in set (0.01 sec)
```

字符串函数

- 获取指定长度的字符串的函数：LEFT(s,n)返回字符串s开始的最左边n个字符；RIGHT(s,n)返回字符串s最右边n字符。

```
mysql> SELECT LEFT('I LIKEMY SQL',5),RIGHT('I LIKE MYSQL',5);
+-----------------------+-------------------------+
| LEFT('I LIKEMY SQL',5) | RIGHT('I LIKE MYSQL',5) |
+-----------------------+-------------------------+
| I LIK                 | MYSQL                   |
+-----------------------+-------------------------+
1 row in set (0.00 sec)
```

练习2

- 1. 显示所有员工姓名的前三个字符
- 2. 显示正好为5个字符的员工的姓名，工资，部门号

字符串函数

- 填充字符串函数：LPAD(s1,len,s2)返回字符串s1,其左边由字符串s2填补到len字符长度；RPAD(s1,len,s2)返回字符串s1，其右边被字符串s2填补至len字符长度。

```
mysql> SELECT LPAD('MYSQL',10,'*'),RPAD('MYSQL',10,'*');
+-----------------------+-----------------------+
| LPAD('MYSQL',10,'*') | RPAD('MYSQL',10,'*') |
+-----------------------+-----------------------+
| *****MYSQL            | MYSQL*****            |
+-----------------------+-----------------------+
1 row in set (0.00 sec)
```

字符串函数

Neuedu

- 删除字符串空格函数：LTRIM(s)返回字符串s，字符串左侧空格字符被删除。RTRIM(s)返回字符串s，字符串右侧空格字符被删除。
- 删除指定字符串函数：TRIM(s1 FROM s)删除字符串s中两端所有的子字符串s1。

```
mysql> SELECT LTRIM('  MYSQL'),RTRIM('MYSQL '),TRIM('I' FROM 'ILIKEMYSQLI'),TRIM('  MYSQL  ');
+----------------+----------------+-----------------------------+------------------+
| LTRIM('  MYSQL') | RTRIM('MYSQL ') | TRIM('I' FROM 'ILIKEMYSQLI') | TRIM('  MYSQL  ') |
+----------------+----------------+-----------------------------+------------------+
| MYSQL          | MYSQL          | LIKEMYSQL                   | MYSQL            |
+----------------+----------------+-----------------------------+------------------+
1 row in set (0.00 sec)
```

- 重复生成字符串函数：REPEAT(s,n)返回一个由重复的字符串s组成的字符串，字符串s的数目等于n。如果n <= 0，返回一个空字符串。如果s或n是NULL，返回NULL。

字符串函数

- 空格函数SPACE(n)：返回一个由n个空格组成的字符串。
- 字符串替换函数REPLACE（s，s1，s2）：将s字符串中的s1字符替换成s2。
- 比较字符串大小函数STRCMP(s1,s2)：若所有的字符串均相同，则返回0，若根据当前分类次序，第一个参数小于第二个，则返回-1，其它情况返回1。

```
mysql> SELECT LENGTH(SPACE(3)),REPLACE('I LIKE MYSQL','I','i'),STRCMP('MYSQLI','MYSQL');
+------------------+-----------------------------------+--------------------------+
| LENGTH(SPACE(3)) | REPLACE('I LIKE MYSQL','I','i') | STRCMP('MYSQLI','MYSQL') |
+------------------+-----------------------------------+--------------------------+
|                3 | i LiKE MYSQL                      |                        1 |
+------------------+-----------------------------------+--------------------------+
1 row in set (0.00 sec)
```

字符串函数

- 获取子串函数：SUBSTRING(s,n,len)带有len参数的格式从字符串s返回一个长度同len字符相同的子字符串，起始于位置n。MID(s,n,len)其作用相同。

```
mysql> SELECT SUBSTRING('I LIKE MYSQL',3,4),SUBSTRING('I LIKE MYSQL',3);
+-------------------------------+-------------------------------+
| SUBSTRING('I LIKE MYSQL',3,4) | SUBSTRING('I LIKE MYSQL',3)   |
+-------------------------------+-------------------------------+
| LIKE                          | LIKE MYSQL                    |
+-------------------------------+-------------------------------+
1 row in set (0.01 sec)
```

- 匹配子串开始位置函数：LOCATE(str1,str)、POSITION(str1 IN str)和INSTR(str, str1)3个函数作用相同，返回子字符串str1在字符串str中的开始位置。

```
mysql> SELECT LOCATE('LIKE','I LIKE MYSQL'),POSITION('LIKE' IN 'I LIKE MYSQL'),INSTR('I LIKE MYSQL','LIKE');
+-----------------------------+-----------------------------------+-----------------------------+
| LOCATE('LIKE','I LIKE MYSQL') | POSITION('LIKE' IN 'I LIKE MYSQL') | INSTR('I LIKE MYSQL','LIKE') |
+-----------------------------+-----------------------------------+-----------------------------+
|                           3 |                                 3 |                           3 |
+-----------------------------+-----------------------------------+-----------------------------+
1 row in set (0.00 sec)
```

字符串函数

- 字符串逆序函数REVERSE(s)：将字符串s反转，返回的字符串的顺序和s字符顺序相反。

```
mysql> SELECT REVERSE('I LIKE MYSQL');
+-------------------------+
| REVERSE('I LIKE MYSQL') |
+-------------------------+
| LQSYM EKIL I            |
+-------------------------+
1 row in set (0.00 sec)
```

- 返回指定位置ELT(N,字符串1,字符串2,字符串3,...，字符串N)函数：若N = 1，则返回值为字符串1，若N=2，则返回值为字符串2，以此类推。若N小于1或大于参数的数目，则返回值为NULL。

```
mysql> SELECT ELT(2,'I','LIKE','MYSQL');
+-----------------------------+
| ELT(2,'I','LIKE','MYSQL') |
+-----------------------------+
| LIKE                        |
+-----------------------------+
1 row in set (0.00 sec)
```

练习3

- 1. 写一个查询, 用首字母大写，其它字母小写显示雇员的 ename，显示名字的长度，并给每列一个适当的标签，条件是满足所有雇员名字的开始字母是J、A 或 M 的雇员，并对查询结果按雇员的ename升序排序。（提示：使用length、substr）
- 2. 查询员工姓名中中包含大写或小写字母A的员工姓名。
- 3. 显示所有员工的姓名, 用a替换所有"A"

- 4. 查询部门编号为10或20，入职日期在81年5月1日之后，并且姓名中包含大写字母A的员工姓名，员工姓名长度

- 5. 查询每个职工的编号, 姓名，工资
  - 要求将查询到的数据按照一定的格式合并成一个字符串.
  - 前10位：编号, 不足部分用*填充, 左对齐
  - 中间10位：姓名，不足部分用*填充, 左对齐
  - 后10位：工资，不足部分用*填充, 右对齐

日期和时间函数

- CURDATE()和CURRENT_DATE() ： 获取当前日期函数;
- NOW()： 返回服务器的当前日期和时间;
- CURTIME() ： 返回当前时间，只包含时分秒;
- UTC_DATE() ： 返回世界标准时间日期函数;
- UTC_TIME()： 返回世界标准时间函数;
- TIMEDIFF(expr1, expr2)： 返回两个日期相减相差的时间数;
- DATEDIFF(expr1, expr2)： 返回两个日期相减相差的天数;
- DATE_ADD(date,INTERVAL expr type):日期加上一个时间间隔值;
- DATE_SUB(date,INTERVAL expr type):日期减去一个时间间隔值;
- DATE(date)、TIME(date)、YEAR（date）：选取日期时间的各个部分:
- EXTRACT(unit FROM date)：从日期中抽取出某个单独的部分或组合;
- DAYOFWEEK(date) 、DAYOFMONTH(date) 、DAYOFYEAR(date):返回日期在一周、一月、一年中是第几天
- DAYNAME、MONTHNAME：返回日期的星期和月份名称;
- DATE_FORMAT(date,format)： 格式化日期;
- TIME_FORMATE(time,formate) ： 格式化时间;

日期和时间函数

- 获取当前日期的函数：CURDATE()和CURRENT_DATE()、CURRENT_DATE函数作用相同，将当前日期按照‘YYYY-MM-DD’或YYYYMMDD格式的值返回，具体格式根据函数用在字符串或是数字语境中而定。
- 可通过CURDATE()+0，将当前日期转换成数值。

```
mysql> SELECT CURDATE(),CURRENT_DATE(),CURDATE()+0;
+------------+----------------+-------------+
| CURDATE()  | CURRENT_DATE() | CURDATE()+0 |
+------------+----------------+-------------+
| 2017-09-18 | 2017-09-18     |    20170918 |
+------------+----------------+-------------+
1 row in set (0.00 sec)
```

日期和时间函数

- 获取当前时间函数NOW()，返回当前日期和时间值，格式为‘YYYY-MM-DD HH:MM:SS’或YYYYMMDDHHMMSS，具体格式根据函数是否用在字符串或数字语境而定。
- 可通过NOW()+0，将当前时间转换成数值。
- 和NOW()函数的作用相同的还有：CURRENT_TIMESTAMP 、CURRENT_TIMESTAMP()、LOCALTIMESTAMP 、 LOCALTIMESTAMP()、LOCALTIME 、 LOCALTIME()

```
mysql> SELECT CURRENT_TIMESTAMP(),LOCALTIME(),NOW(),SYSDATE(),NOW()+0;
+---------------------+---------------------+---------------------+---------------------+----------------------+
| CURRENT_TIMESTAMP() | LOCALTIME()         | NOW()               | SYSDATE()           | NOW()+0              |
+---------------------+---------------------+---------------------+---------------------+----------------------+
| 2017-09-18 15:00:38 | 2017-09-18 15:00:38 | 2017-09-18 15:00:38 | 2017-09-18 15:00:38 | 20170918150038.000000 |
+---------------------+---------------------+---------------------+---------------------+----------------------+
1 row in set (0.00 sec)
```

日期和时间函数

- 获取当前时间的函数：CURTIME()和CURRENT_TIME()、CURRENT_TIME函数作用相同，将当前时间按照‘HH:MM:SS’或者‘HHMMSS’格式的值返回，具体格式根据函数用在字符串或是数字语境中而定。
- 可通过CURDTIME()+0，将当前时间转换成数值。

```
mysql> SELECT CURTIME(),CURRENT_TIME, CURRENT_TIME(),CURTIME()+0;
+-----------+--------------+----------------+----------------+
| CURTIME() | CURRENT_TIME | CURRENT_TIME() | CURTIME()+0    |
+-----------+--------------+----------------+----------------+
| 15:22:43  | 15:22:43     | 15:22:43       | 152243.000000 |
+-----------+--------------+----------------+----------------+
1 row in set (0.00 sec)
```

日期和时间函数

- 返回世界标准时间日期函数：UTC_DATE()，其格式为‘YYYY-MM-DD’或YYYYMMDD，具体格式取决于函数是否用在字符串或数字语境中。
- 返回世界标准时间函数：UTC_TIME()，其格式为‘HH：MM：SS’或‘HHMMSS’

```
mysql> SELECT UTC_DATE(),UTC_TIME();
+------------+------------+
| UTC_DATE() | UTC_TIME() |
+------------+------------+
| 2017-09-18 | 07:03:46   |
+------------+------------+
1 row in set (0.00 sec)
```

日期和时间函数

- TIMEDIFF(expr1, expr2)：返回两个日期相减（expr1 – expr2 ）相差的时间数（两个参数类型必须相同）

```
mysql> select timediff('18:32:59','60000');
+-----------------------------+
| timediff('18:32:59','60000') |
+-----------------------------+
| 12:32:59                    |
+-----------------------------+
1 row in set (0.00 sec)
```

- DATEDIFF(expr1, expr2)：返回两个日期相减（expr1 – expr2 ）相差的天数

```
mysql> select datediff('2017-9-24 18:32:59','2016-9-1');
+-----------------------------------------+
| datediff('2017-9-24 18:32:59','2016-9-1') |
+-----------------------------------------+
|                                     388 |
+-----------------------------------------+
1 row in set (0.00 sec)
```

日期和时间函数

- 计算日期和时间的函数，分别为给定的日期date加上(add)或减去(sub)一个时间间隔值expr
  - DATE_ADD(date, INTERVAL expr unit)
  - DATE_SUB(date, INTERVAL expr unit)
    - interval是间隔类型关键字
    - expr是一个表达式，对应后面的类型
    - unit是时间间隔的单位(间隔类型)，如下：

HOUR	小时
MINUTE	分
SECOND	秒
MICROSECOND	毫秒
YEAR	年

MONTH	月
DAY	日
WEEK	周
QUARTER	季
YEAR_MONTH	年和月

DAY_HOUR	日和小时
DAY_MINUTE	日和分钟
DAY_ SECOND	日和秒
HOUR_MINUTE	小时和分
HOUR_SECOND	小时和秒
MINUTE_SECOND	分钟和秒

日期和时间函数

Neuedu

```
mysql> SELECT NOW(),DATE_ADD(NOW(),INTERVAL 1 DAY);
+---------------------+-----------------------------------+
| NOW()               | DATE_ADD(NOW(),INTERVAL 1 DAY)    |
+---------------------+-----------------------------------+
| 2017-09-19 14:11:51 | 2017-09-20 14:11:51               |
+---------------------+-----------------------------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT DATE_SUB('2005-01-01 00:00:00',INTERVAL '1 1:1:1' DAY_SECOND);
+-------------------------------------------------------------+
| DATE_SUB('2005-01-01 00:00:00',INTERVAL '1 1:1:1' DAY_SECOND) |
+-------------------------------------------------------------+
| 2004-12-30 22:58:59                                         |
+-------------------------------------------------------------+
1 row in set (0.00 sec)
```

- 不使用函数，也可以写表达式进行日期的加减：
  - date + INTERVAL expr unit
  - date - INTERVAL expr unit

```
mysql> SELECT '2008-12-31 23:59:59' + INTERVAL 1 SECOND,'2005-01-01' - INTERVAL 1 SECOND;
+-----------------------------------------+-----------------------------------+
| '2008-12-31 23:59:59' + INTERVAL 1 SECOND | '2005-01-01' - INTERVAL 1 SECOND |
+-----------------------------------------+-----------------------------------+
| 2009-01-01 00:00:00                     | 2004-12-31 23:59:59               |
+-----------------------------------------+-----------------------------------+
1 row in set (0.00 sec)
```

日期和时间函数

- 选取日期时间的各个部分：日期、时间、年、季度、月、日、小时、分钟、秒、微秒（常用）
  - SELECT now(),date(now()); -- 日期
  - SELECT now(),time(now()); -- 时间
  - SELECT now(),year(now()); -- 年
  - SELECT now(),quarter(now()); -- 季度
  - SELECT now(),month(now()); -- 月
  - SELECT now(),week(now()); -- 周
  - SELECT now(),day(now()); -- 日
  - SELECT now(),hour(now()); -- 小时
  - SELECT now(),minute(now()); -- 分钟
  - SELECT now(),second(now()); -- 秒
  - SELECT now(),microsecond(now()); -- 微秒

日期和时间函数

- EXTRACT(unit FROM date)：从日期中抽取出某个单独的部分或组合
  - SELECT now(),extract(YEAR FROM now()); -- 年
  - SELECT now(),extract(QUARTER FROM now()); -- 季度
  - SELECT now(),extract(MONTH FROM now()); -- 月
  - SELECT now(),extract(WEEK FROM now()); -- 周
  - SELECT now(),extract(DAY FROM now()); -- 日
  - SELECT now(),extract(HOUR FROM now()); -- 小时
  - SELECT now(),extract(MINUTE FROM now()); -- 分钟
  - SELECT now(),extract(SECOND FROM now()); -- 秒
  - SELECT now(),extract(YEAR_MONTH FROM now()); -- 年月
  - SELECT now(),extract(HOUR_MINUTE FROM now()); -- 时分

日期和时间函数

- 个性化显示时间日期
  - dayofweek(date)
  - dayofmonth(date)
  - dayofyear(date)

```
mysql> SELECT now(),dayofweek(now()),dayofmonth(now()),dayofyear(now());
+---------------------+------------------+-------------------+-----------------+
| now()               | dayofweek(now()) | dayofmonth(now()) | dayofyear(now()) |
+---------------------+------------------+-------------------+-----------------+
| 2017-09-19 14:50:44 |                3 |                19 |             262 |
+---------------------+------------------+-------------------+-----------------+
1 row in set (0.00 sec)
```

日期和时间函数

- 分别返回日期的星期和月份名称，名称是中文or英文的由系统变量lc_time_names控制(默认值是'en_US')
  - dayname()
  - monthname()

```
mysql> show variables like 'lc_time_names';
+----------------+-------+
| Variable_name  | Value |
+----------------+-------+
| lc_time_names  | en_US |
+----------------+-------+
1 row in set (0.00 sec)
```

```
mysql> select dayname(now()),monthname(now());
+----------------+------------------+
| dayname(now()) | monthname(now()) |
+----------------+------------------+
| Tuesday        | September        |
+----------------+------------------+
1 row in set (0.00 sec)
```

日期和时间函数

- 更改系统变量，如果出现乱码，再设置编码为gbk

```
mysql> set lc_time_names='zh_CN';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> set names gbk;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select dayname(now()),monthname(now());
+----------------+------------------+
| dayname(now()) | monthname(now()) |
+----------------+------------------+
| 星期二         | 九月             |
+----------------+------------------+
1 row in set (0.01 sec)
```

日期和时间函数

- 将日期和时间格式化的函数： DATE_FORMAT(date,format)、TIME_FORMAT(time,format)
  - 根据format 指定的格式显示日期或者时间值。
  - date 参数是合法的日期
  - format 规定日期/时间的输出格式

```
mysql> SELECT date_format(NOW(),'%Y-%m-%d %H:%i:%s'),time_format(curtime(),'%H:%i:%s');
+---------------------------------------+-----------------------------------+
| date_format(NOW(),'%Y-%m-%d %H:%i:%s') | time_format(curtime(),'%H:%i:%s') |
+---------------------------------------+-----------------------------------+
| 2017-09-19 15:32:21                   | 15:32:21                          |
+---------------------------------------+-----------------------------------+
1 row in set (0.00 sec)
```

日期和时间函数

- 可以使用的格式有：

格式	描述
%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

练习4

- 1. 查询服务器当前时间
- 2. 查询部门10, 20的员工截止到2000年1月1日，工作了多少个月，入职的月份。
- 3. 如果员工试用期6个月，查询职位不是MANAGER的员工姓名，入职日期，转正日期，入职日期是第多少月，第多少周

流程控制函数

- 常见的控制流程函数如下：
  - CASE
  - IF
  - IFNULL
  - NULLIF

流程控制函数

- CASE value WHEN [compare-value] THEN result [WHEN [compare-value] THEN result ...] [ELSE result] END
- CASE WHEN [condition] THEN result [WHEN [condition] THEN result ...] [ELSE result] END
- 在第一个方案的返回结果中， value=compare-value。
- 而第二个方案的返回结果是第一种情况的真实结果。如果没有匹配的结果值，则返回结果为ELSE后的结果，如果没有ELSE 部分，则返回值为 NULL。

- SELECT CASE 11 WHEN 1 THEN 'one'
  WHEN 2 THEN 'two' ELSE 'more' END;
- SELECT CASE WHEN 1>0 THEN 'true' ELSE 'false' END;

流程控制函数

Neuedu

```
mysql> select empno,sal, case when sal<2000 then 'low' else 'high' end from emp;
+-------+---------+-----------------------------------------+
| empno | sal     | case when sal<2000 then 'low' else 'high' end |
+-------+---------+-----------------------------------------+
|  7369 |  800.00 | low                                     |
|  7499 | 1600.00 | low                                     |
|  7521 | 1250.00 | low                                     |
|  7566 | 2975.00 | high                                    |
|  7654 | 1250.00 | low                                     |
|  7698 | 2850.00 | high                                    |
|  7782 | 2450.00 | high                                    |
|  7788 | 3000.00 | high                                    |
|  7839 | 5000.00 | high                                    |
|  7844 | 1500.00 | low                                     |
|  7876 | 1100.00 | low                                     |
|  7900 |  950.00 | low                                     |
|  7902 | 3000.00 | high                                    |
|  7934 | 1300.00 | low                                     |
+-------+---------+-----------------------------------------+
14 rows in set (0.00 sec)
```

流程控制函数

- IF(expr1,expr2,expr3)
  - 如果 expr1 是TRUE (expr1 <> 0 and expr1 <> NULL)，则 IF()的返回值为expr2; 否则返回值则为 expr3。
  - IF() 的返回值为数字值或字符串值，具体情况视其所在语境而定。

- SELECT IF(1>2,2,3);
- SELECT IF(1<2,'yes ','no');

流程控制函数

- IFNULL(expr1,expr2)|NULLIF(expr1,expr2)
  - 假如expr1 不为NULL ，则IFNULL() 的返回值为expr1 ； 否则其返回值为expr2 。
  - IFNULL() 的返回值是数字或是字符串，具体情况取决于其所使用的语境

```
mysql> select empno,comm, ifnull(comm,0) from emp;
+-------+---------+----------------+
| empno | comm    | ifnull(comm,0) |
+-------+---------+----------------+
|  7369 |    NULL |           0.00 |
|  7499 |  300.00 |         300.00 |
|  7521 |  500.00 |         500.00 |
|  7566 |    NULL |           0.00 |
|  7654 | 1400.00 |        1400.00 |
|  7698 |    NULL |           0.00 |
|  7782 |    NULL |           0.00 |
|  7788 |    NULL |           0.00 |
|  7839 |    NULL |           0.00 |
|  7844 |    0.00 |           0.00 |
|  7876 |    NULL |           0.00 |
|  7900 |    NULL |           0.00 |
|  7902 |    NULL |           0.00 |
|  7934 |    NULL |           0.00 |
+-------+---------+----------------+
14 rows in set (0.00 sec)
```

其他函数

- Database()
- Version()
- User()
- Inet_aton(ip)
- Inet_ntoa(num)
- Password(str)
- Md5()

其他函数

- Database():返回使用utf8 字符集的默认(当前) 数据库名
- Version():返回指示MySQL 服务器版本的字符串。
- User():返回当前MySQL 用户名和机主名

```
mysql> select database(),version(),user();
+------------+----------------------+----------------+
| database() | version()            | user()         |
+------------+----------------------+----------------+
| mysqldb    | 5.0.67-community-nt | root@localhost |
+------------+----------------------+----------------+
1 row in set (0.00 sec)
```

其他函数

- Inet_aton():给出一个作为字符串的网络地址的点地址表示，返回一个代表该地址数值的整数。
- Inet_ntoa():给定一个数字网络地址，返回作为字符串的该地址的点地址表示。

```
mysql> select inet_aton('192.168.1.1'),inet_ntoa('3232235777');
+--------------------------+--------------------------+
| inet_aton('192.168.1.1') | inet_ntoa('3232235777') |
+--------------------------+--------------------------+
|               3232235777 | 192.168.1.1              |
+--------------------------+--------------------------+
1 row in set (0.01 sec)
```

其他函数

- Password(str):从原文密码str 计算并返回密码字符串，当参数为NULL 时返回NULL。
- Md5(str):为字符串算出一个MD5 128 比特检查和。

```
mysql> select password('123456'),md5('123456');
+-------------------------------------------+----------------------------------+
| password('123456')                        | md5('123456')                    |
+-------------------------------------------+----------------------------------+
| *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 | e10adc3949ba59abbe56e057f20f883e |
+-------------------------------------------+----------------------------------+
1 row in set (0.00 sec)
```

小结

- 了解函数的概念，掌握常用的函数
  - 数学函数
  - 字符串函数
  - 日期和时间函数
  - 流程控制函数
  - 其他函数

课后作业

Neuedu

1. 计算2000年1月1日到现在有多少月，多少周（四舍五入）。
2. 查询员工ENAME的第三个字母是A的员工的信息(使用2个函数)。
3. 使用trim函数将字符串‘hello’、‘ Hello ’、‘bllb’、‘ hello ’分别处理得到下列字符串ello、Hello、ll、hello。
4. 将员工工资按如下格式显示：123,234.00 RMB 。

课后作业

5. 查询员工的姓名及其经理编号，要求对于没有经理的显示“No Manager”字符串。
6. 将员工的参加工作日期按如下格式显示：月份/年份。
7. 在员工表中查询出员工的工资，并计算应交税款：如果工资小于1000,税率为0，如果工资大于等于1000并小于2000，税率为10%，如果工资大于等于2000并小于3000，税率为15%，如果工资大于等于3000，税率为20%。
8. 创建一个查询显示所有雇员的 ename和 sal。格式化sal为 15 个字符长度，用 $ 左填充，列标签 SALARY。

Neuedu