07丨什么是SQL函数？为什么使用SQL函数可能会带来问题？

2019-06-26 陈旸

函数在计算机语言的使用中贯穿始终，在SQL中我们也可以使用函数对检索出来的数据进行函数操作，比如求某列数据的平均值，或者求字符串的长度等。从函数定义的角度出发，我们可以将函数分成内置函数和自定义函数。在SQL语言中，同样也包括了内置函数和自定义函数。内置函数是系统内置的通用函数，而自定义函数是我们根据自己的需要编写的，下面讲解的是SQL的内置函数。

你需要从以下几个方面掌握SQL函数：

1. 什么是SQL函数？
2. 内置的SQL函数都包括哪些？
3. 如何使用SQL函数对一个数据表进行操作，比如针对一个王者荣耀的英雄数据库，我们可以使用这些函数完成哪些操作？
4. 什么情况下使用SQL函数？为什么使用SQL函数有时候会带来问题？

什么是SQL函数

当我们学习编程语言的时候，也会遇到函数。函数的作用是什么呢？它可以把我们经常使用的代码封装起来，需要的时候直接调用即可。这样既提高了代码效率，又提高了可维护性。

SQL中的函数一般是在数据上执行的，可以很方便地转换和处理数据。一般来说，当我们从数据表中检索出数据之后，就可以进一步对这些数据进行操作，得到更有意义的结果，比如返回指定

条件的函数，或者求某个字段的平均值等。

常用的SQL函数有哪些

SQL提供了一些常用的内置函数，当然你也可以自己定义SQL函数。SQL的内置函数对于不同的数据库软件来说具有一定的通用性，我们可以把内置函数分成四类：

1. 算术函数
2. 字符串函数
3. 日期函数
4. 转换函数

这4类函数分别代表了算术处理、字符串处理、日期处理、数据类型转换，它们是SQL函数常用的划分形式，你可以思考下，为什么是这4个维度？

函数是对提取出来的数据进行操作，那么数据表中字段类型的定义有哪几种呢？

我们经常会保存一些数值，不论是整数类型，还是浮点类型，实际上对应的就是数值类型。同样我们也会保存一些文本内容，可能是人名，也可能是某个说明，对应的就是字符串类型。此外我们还需要保存时间，也就是日期类型。那么针对数值、字符串和日期类型的数据，我们可以对它们分别进行算术函数、字符串函数以及日期函数的操作。如果想要完成不同类型数据之间的转换，就可以使用转换函数。

算术函数

算术函数，顾名思义就是对数值类型的字段进行算术运算。常用的算术函数及含义如下表所示：

函数名	定义
ABS()	取绝对值
MOD()	取余
ROUND()	四舍五入为指定的小数位数，需要有两个参数，分别为字段名称、小数位数

这里我举一些简单的例子，你来体会下：

SELECT ABS(-2)，运行结果为2。

SELECT MOD(101,3)，运行结果2。

SELECT ROUND(37.25,1)，运行结果37.3。

字符串函数

常用的字符串函数操作包括了字符串拼接，大小写转换，求长度以及字符串替换和截取等。具体的函数名称及含义如下表所示：

函数名	定义
CONCAT()	将多个字符串拼接起来
LENGTH()	计算字段的长度，一个汉字算三个字符，一个数字或字母算一个字符
CHAR_LENGTH()	计算字段的长度，汉字、数字、字母都算一个字符
LOWER()	将字符串中的字符转化为小写
UPPER()	将字符串中的字符转化为大写
REPLACE()	替换函数，有3个参数，分别为：要替换的表达式或字段名、想要查找的被替换字符串、替换成哪个字符串
SUBSTRING()	截取字符串，有3个参数，分别为：待截取的表达式或字段名、开始截取的位置、想要截取的字符串长度

这里同样有一些简单的例子，你可以自己运行下：

SELECT CONCAT('abc', 123)，运行结果为abc123。

SELECT LENGTH('你好')，运行结果为6。

SELECT CHAR_LENGTH('你好')，运行结果为2。

SELECT LOWER('ABC')，运行结果为abc。

SELECT UPPER('abc')，运行结果ABC。

SELECT REPLACE('fabcd', 'abc', 123)，运行结果为f123d。

SELECT SUBSTRING('fabcd', 1,3)，运行结果为fab。

日期函数

日期函数是对数据表中的日期进行处理，常用的函数包括：

函数名	定义
CURRENT_DATE()	系统当前日期
CURRENT_TIME()	系统当前时间，没有具体的日期
CURRENT_TIMESTAMP()	系统当前时间，包括具体的日期+时间
EXTRACT()	抽取具体的年、月、日
DATE()	返回时间的日期部分
YEAR()	返回时间的年份部分
MONTH()	返回时间的月份部分
DAY()	返回时间的天数部分
HOUR()	返回时间的小时部分
MINUTE()	返回时间的分钟部分
SECOND()	返回时间的秒部分

下面是一些简单的例子，你可自己运行下：

SELECT CURRENT_DATE()，运行结果为2019-04-03。

SELECT CURRENT_TIME()，运行结果为21:26:34。

SELECT CURRENT_TIMESTAMP()，运行结果为2019-04-03 21:26:34。

SELECT EXTRACT(YEAR FROM '2019-04-03')，运行结果为2019。

SELECT DATE('2019-04-01 12:00:05')，运行结果为2019-04-01。

这里需要注意的是，DATE日期格式必须是yyyy-mm-dd的形式。如果要进行日期比较，就要使用DATE函数，不要直接使用日期与字符串进行比较，我会在后面的例子中讲具体的原因。

转换函数

转换函数可以转换数据之间的类型，常用的函数如下表所示：

函数名	定义
CAST()	数据类型转换，参数是一个表达式，表达式通过AS关键词分割了2个参数，分别是原始数据和目标数据类型
COALESCE()	返回第一个非空数值

这两个函数不像其他函数，看一眼函数名就知道代表什么、如何使用。下面举了这两个函数的例子，你需要自己运行下：

SELECT CAST(123.123 AS INT)，运行结果会报错。

SELECT CAST(123.123 AS DECIMAL(8,2))，运行结果为123.12。

SELECT COALESCE(null,1,2)，运行结果为1。

CAST函数在转换数据类型的时候，不会四舍五入，如果原数值有小数，那么转换为整数类型的时候就会报错。不过你可以指定转化的小数类型，在MySQL和SQL Server中，你可以用DECIMAL(a,b)来指定，其中a代表整数部分和小数部分加起来最大的位数，b代表小数位数，比如DECIMAL(8,2)代表的是精度为8位（整数加小数位数最多为8位），小数位数为2位的数据类型。所以SELECT CAST(123.123 AS DECIMAL(8,2))的转换结果为123.12。

用SQL函数对王者荣耀英雄数据做处理

我创建了一个王者荣耀英雄数据库，一共有69个英雄，23个属性值。SQL文件见Github地址：https://github.com/cystanford/sql_heros_data。

id	name	hp_max	hp_growth	hp_start	.....	role_assist	birthdate
10000	夏侯惇	7350	288.8	3307		战士	2016-07-19
10001	钟无艳	7000	275	3150		坦克	
10002	张飞	8341	329	3450		辅助	
.....	.....	.....	.....	.....	.....	.....	.....
10068	百里守约	5611	185	3019		刺客	2017-08-08

我们现在把这个文件导入到MySQL中，你可以使用Navicat可视化数据库管理工具将.sql文件导入到数据库中。数据表为heros，然后使用今天学习的SQL函数，对这个英雄数据表进行处理。

首先显示英雄以及他的物攻成长，对应字段为attack_growth。我们让这个字段精确到小数点后一位，需要使用的是算术函数里的ROUND函数。

```
SQL：SELECT name, ROUND(attack_growth,1) FROM heros
```

代码中，ROUND(attack_growth,1)中的attack_growth代表想要处理的数据，“1”代表四舍五入的位数，也就是我们这里需要精确到的位数。

运行结果为：

name	ROUND(attack_growth, 1)
夏侯惇	12.0
钟无艳	11.0
张飞	11.0
……	……
百里守约	16.0

假设我们想显示英雄最大生命值的最大值，就需要用到MAX函数。在数据中，“最大生命值”对应的列数为hp_max，在代码中的格式为MAX(hp_max)。

```
SQL：SELECT MAX(hp_max) FROM heros
```

运行结果为9328。

假如我们想要知道最大生命值最大的是哪个英雄，以及对应的数值，就需要分成两个步骤来处理：首先找到英雄的最大生命值的最大值，即SELECT MAX(hp_max) FROM heros，然后再筛选最大生命值等于这个最大值的英雄，如下所示。

```
SQL：SELECT name, hp_max FROM heros WHERE hp_max = (SELECT MAX(hp_max) FROM heros)
```

运行结果：

name	hp_max
廉颇	9328

假如我们想显示英雄的名字，以及他们的名字字数，需要用到CHAR_LENGTH函数。

```
SQL：SELECT CHAR_LENGTH(name), name FROM heros
```

运行结果为：

CHAR_LENGTH(name)	name
3	夏侯惇
3	钟无艳
2	张飞
.....	.....
4	百里守约

假如想要提取英雄上线日期（对应字段birthdate）的年份，只显示有上线日期的英雄即可（有些英雄没有上线日期的数据，不需要显示），这里我们需要使用EXTRACT函数，提取某一个时间元素。所以我们需要筛选上线日期不为空的英雄，即WHERE birthdate is not null，然后再显示他们的名字和上线日期的年份，即：

```
SQL： SELECT name, EXTRACT(YEAR FROM birthdate) AS birthdate FROM heros WHERE birthdate is NOT NU
```

或者使用如下形式：

```
SQL: SELECT name, YEAR(birthdate) AS birthdate FROM heros WHERE birthdate is NOT NULL
```

运行结果为：

name	birthdate
夏侯惇	2016
牛魔	2015
吕布	2015
.....	.....
百里守约	2017

假设我们需要找出在2016年10月1日之后上线的所有英雄。这里我们可以采用DATE函数来判断birthdate的日期是否大于2016-10-01，即WHERE DATE(birthdate)>'2016-10-01'，然后再显示符合要求的全部字段信息，即：

```
SQL： SELECT * FROM heros WHERE DATE(birthdate)>'2016-10-01'
```

需要注意的是下面这种写法是不安全的：

```
SELECT * FROM heros WHERE birthdate>'2016-10-01'
```

因为很多时候你无法确认birthdate的数据类型是字符串，还是datetime类型，如果你想对日期部分进行比较，那么使用DATE(birthdate)来进行比较是更安全的。

运行结果为：

id	name	hp_max	hp_growth	hp_start	.....	role_assist	birthdate
10009	东皇太一	7669	319	3201			2017-03-30
10011	太乙真人	6835	242	3443		坦克	2016-11-24
10033	干将莫邪	5583	171	3189			2017-05-22
.....	.....	.....	.....	.....	.....	.....	.....
10068	百里守约	5611	185	3019		刺客	2017-08-08

假设我们需要知道在2016年10月1日之后上线英雄的平均最大生命值、平均最大法力和最高物攻

最大值。同样我们需要先筛选日期条件，即WHERE DATE(birthdate)>'2016-10-01'，然后再选择AVG(hp_max), AVG(mp_max), MAX(attack_max)字段进行显示。

```
SQL： SELECT AVG(hp_max), AVG(mp_max), MAX(attack_max) FROM heros WHERE DATE(birthdate)>'2016-1
```

运行结果为：

AVG(hp_max)	AVG(mp_max)	MAX(attack_max)
6611.5000	1821.5000	410

为什么使用SQL函数会带来问题

尽管SQL函数使用起来会很方便，但我们使用的时候还是要谨慎，因为你使用的函数很可能在运行环境中无法工作，这是为什么呢？

如果你学习过编程语言，就会知道语言是有不同版本的，比如Python会有2.7版本和3.x版本，不过它们之间的函数差异不大，也就在10%左右。但我们在使用SQL语言的时候，不是直接和这门语言打交道，而是通过它使用不同的数据库软件，即DBMS。DBMS之间的差异性很大，远大于同一个语言不同版本之间的差异。实际上，只有很少的函数是被DBMS同时支持的。比如，大多数DBMS使用（||）或者（+）来做拼接符，而在MySQL中的字符串拼接函数为Concat()。大部分DBMS会有自己特定的函数，这就意味着采用SQL函数的代码可移植性是很差的，因此在使用函数的时候需要特别注意。

关于大小写的规范

细心的人可能会发现，我在写SELECT语句的时候用的是大写，而你在网上很多地方，包括你自己写的时候可能用的是小写。实际上在SQL中，关键字和函数名是不用区分字母大小写的，比如SELECT、WHERE、ORDER、GROUP BY等关键字，以及ABS、MOD、ROUND、MAX等函数名。

不过在SQL中，你还是要确定大小写的规范，因为在Linux和Windows环境下，你可能会遇到不同的大小写问题。

比如MySQL在Linux的环境下，数据库名、表名、变量名是严格区分大小写的，而字段名是忽略大小写的。

而MySQL在Windows的环境下全部不区分大小写。

这就意味着如果你的变量名命名规范没有统一，就可能产生错误。这里有一个有关命名规范的建议：

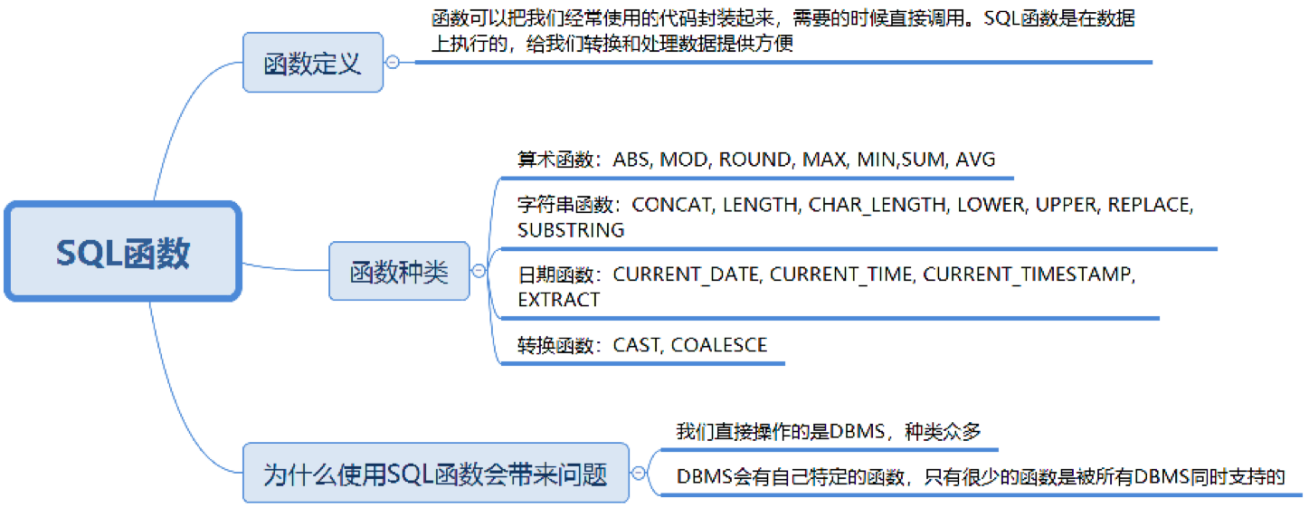
1. 关键字和函数名称全部大写；
2. 数据库名、表名、字段名称全部小写；
3. SQL语句必须以分号结尾。

虽然关键字和函数名称在SQL中不区分大小写，也就是如果小写的话同样可以执行，但是数据库名、表名和字段名在Linux MySQL环境下是区分大小写的，因此建议你统一这些字段的命名规则，比如全部采用小写的方式。同时将关键词和函数名称全部大写，以便于区分数据库名、表名、字段名。

总结

函数对于一门语言的重要性毋庸置疑，我们在写Python代码的时候，会自己编写函数，也会使用Python内置的函数。在SQL中，使用函数的时候需要格外留意。不过如果工程量不大，使用的是同一个DBMS的话，还是可以使用函数简化操作的，这样也能提高代码效率。只是在系统集成，或者在多个DBMS同时存在的情况下，使用函数的时候就需要慎重一些。

比如CONCAT()是字符串拼接函数，在MySQL和Oracle中都有这个函数，但是在这两个DBMS中作用却不一样，CONCAT函数在MySQL中可以连接多个字符串，而在Oracle中CONCAT函数只能连接两个字符串，如果要连接多个字符串就需要用（||）连字符来解决。



讲完了SQL函数的使用，我们来做一道练习题。还是根据王者荣耀英雄数据表，请你使用SQL函数作如下的练习：计算英雄的最大生命平均值；显示出所有在2017年之前上线的英雄，如果英雄没有统计上线日期则不显示。

欢迎你在评论区与我分享你的答案，也欢迎点击”请朋友读“，把这篇文章分享给你的朋友或者同事。

精选留言

时间是最真的答案

41

基础篇可以快点更新，很多同学都有一些基础，看懂不难；进阶篇和高级篇可以正常更新，大家可以花多点时间在难点上，可能会更大提高大家的学习效率

2019-06-26

flow

8

由于sql中存在数据类型隐式转换的问题，我查了些资料，有一些不明白的地方。
首先，对于 select * from heros where date(birthdate) > '2016-10-01'，老师你说不使用date函数不安全，但是创建表的时候，birthday声明的就是date类型，那它就不可能是其他类型了啊；其次，我给 birthday 加上索引以后，使用explain查看执行计划，不使用 date 函数直接比较的效果要优于使用 date 函数 (参考的是 explain 结果的 type 属性，使用函数为ALL，不使用函数为range); 还有就是既然birthday 是 date 类型，那 被比较对象 '2017-01-01' 是字符串，这就不同同一种数据类型了，我把它转换成 date 类型后进行比较，explain的结果却是一样的，那这里是否存在隐式的类型转换？

隐式类型转换的资料主要参考这篇文章：https://blog.csdn.net/u010695794/article/details/78509191

2019-06-26

菜菜

2

学得我想打王者荣耀了

2019-06-28

作者回复

很好 我已经很久没打了，昨天整理稿子的时候也有冲突想开一局，不过还是忍住了，一方面手机存储空间满了，另一方面还是多学习和工作，会让状态更好

2019-06-28

盛 2

关于SQL函数问题：老师应当漏了一个关键点吧；它会导致查询不走索引，直接全表遍历，导致慢查询-这才是最重要的问题。现实工作中一般是禁止where条件出现函数：看到了就让开发代码重写。

2019-06-26

Fred 2

SELECT AVG(mp_max) as avg_max FROM hero;
SELECT name WHERE YEAR（birthdate）<2017 AND birthdate IS NOT NULL

2019-06-26

flow 2

```
select avg(hp_max) as avg_hp
from heros;

select name, birthdate
from heros
where birthdate < date('2017-01-01');
```

2019-06-26

supermouse 1

计算英雄的最大生命平均值：
SELECT AVG(hp_max) FROM heros;

显示所有在2017年之前上线的英雄：
SELECT name FROM heros WHERE birthdate IS NOT NULL AND YEAR(birthdate) < 2017;

2019-06-27

作者回复

正确

2019-06-28

Andre 1

答案：SELECT avg(hp_max) as avg_hp
FROM heros;

```
SELECT `name`
FROM heros
```

WHERE birthdate is NOT NULL AND DATE(birthdate)<'2017-01-01';
另外赞同时间是最真的答案的说法，应该来讲基础篇大家都还是学起来不费力的，希望基础篇能够快点更新，然后尽快的进入进阶篇

2019-06-27

作者回复

谢谢 基础篇让大家都有个共识的基础，掌握的人也可以快速复习一遍，进阶篇会有更多关于性能调优，高可用架构等

2019-06-28

mickey 1

#计算英雄的最大生命平均值；
#MySQL实现：
SELECT avg(hp_max) as '英雄的最大生命平均值' FROM heros;

#查询结果：
英雄的最大生命平均值

6580.478260869565

#显示出所有在 2017 年之前上线的英雄，如果英雄没有统计上线日期则不显示。
#MySQL实现：
SELECT * FROM heros WHERE birthdate is NOT NULL AND EXTRACT(YEAR FROM birthdate) < '2017';

#查询结果：
id name hp_max ... birthdate

10000 夏侯惇 7350 ... 2016-07-19
10003 牛魔 8476 ... 2015-11-24
10004 吕布 7344 ... 2015-12-22
10006 芈月 6164 ... 2015-12-08
.....

2019-06-26

圆子蛋 1

1. SELECT AVG(max_hp) FROM heros;
2. SELECT name,YEAR(birthdate) AS birthdate FROM heros WHERE birthdate is NOT NULL AND YEAR(birthdate)<2017

2019-06-26

啦啦啦 1

看了今天的课知道了日期比较不用DATE函数不安全，以前都没用过

2019-06-26

Golden Lee 0

最后两道思考题解答：

SELECT AVG(hp_max) FROM heros;

SELECT name FROM heros WHERE DATE(birthdate) < '2017-01-01' AND birthdate IS NOT NULL;

2019-07-01

华夏 0

SELECT AVG(hp_max) FROM heros;

```
+------------------+
| AVG(hp_max) |
+------------------+
| 6580.478260869565 |
+------------------+
1 row in set (0.02 sec)
```

SELECT name, birthdate FROM heros WHERE birthdate IS NOT NULL AND YEAR(birthdate) < 2017;

```
+--------------+------------+
| name | birthdate |
+--------------+------------+
| 夏侯惇 | 2016-07-19 |
| 牛魔 | 2015-11-24 |
| 吕布 | 2015-12-22 |
| 芈月 | 2015-12-08 |
| 太乙真人 | 2016-11-24 |
| 刘邦 | 2016-04-26 |
| 关羽 | 2016-06-28 |
| 马可波罗 | 2016-08-23 |
| 李元芳 | 2016-04-12 |
| 虞姬 | 2016-05-24 |
| 成吉思汗 | 2016-09-27 |
| 不知火舞 | 2016-05-12 |
| 貂蝉 | 2015-12-15 |
| 周瑜 | 2015-11-10 |
| 张良 | 2015-10-26 |
| 钟馗 | 2016-03-24 |
| 蔡文姬 | 2016-07-08 |
| 花木兰 | 2016-01-01 |
| 李白 | 2016-03-01 |
```

杨戬	2016-10-11
刘备	2016-02-02
宫本武藏	2015-10-30
娜可露露	2016-02-22
+--------------+------------+
23 rows in set (0.01 sec)

2019-06-30

莫莫

0

-- 计算英雄的最大生命平均值
SELECT `name`,MAX(hp_max) FROM heros

-- 显示出所有在 2017 年之前上线的英雄，如果英雄没有统计上线时间则不显示
SELECT `name`,birthdate FROM heros
WHERE DATE(birthdate)< '2017-01-01' AND birthdate IS NOT NULL

2019-06-30

Neo

0

平均生命值：SELECT AVG(hp_max) from heros;
2017年之前上线的英雄： SELECT name, birthdate FROM heros WHERE birthdate IS NOT NULL AND DATE(birthdate)<'2017';

2019-06-30

motorlei

0

练习题1计算英雄的最大生命平均值
SELECT ROUND(AVG(hp_max), 2) AS avg_hp_max #平均值四舍五入保留两位小数
FROM heros;
练习题2显示出所有在 2017 年之前上线的英雄
SELECT `name`, birthdate
FROM heros
WHERE birthdate > DATE('2017-01-01');

2019-06-28

微凉

0

SELECT AVG(hp_max) FROM heros；
SELECT name FROM heros WHERE birthdate IS NOT NULL AND YEAR(birthdate) < '2017';
;

2019-06-28

Amanda

0

> SELECT ROUND(AVG(hp_max), 2) FROM heros;
+----------------------+
| ROUND(AVG(hp_max), 2) |

```
+----------------------+
| 6580.48 |
+----------------------+
```

```
> SELECT name FROM heros WHERE birthdate IS NOT NULL AND DATE(birthdate)<'2017-01-01';
```

2019-06-27

作者回复

正确，你也可以使用YEAR(birthdate)<2017

2019-06-28

Vic

0

老师，mysq drop database后磁盘空间未释放，怎么操作可以让磁盘空间释放呀？

2019-06-27

业余草

0

用了函数，并不是说全部都不走索引了。有些函数走索引，有些函数换种用法也会走索引！

2019-06-27