



下载APP



10 | 如何进行数学计算、字符串处理和条件判断?

2021-03-30 朱晓峰

MySQL 必知必会

[进入课程 >](#)



讲述：朱晓峰

时长 12:30 大小 11.46M



你好，我是朱晓峰。

MySQL 提供了很多功能强大，而且使用起来非常方便的函数，包括数学函数、字符串处理函数和条件判断函数等。

在很多场景中，我们都会用到这些函数，比如说，在超市项目的实际开发过程中，会有这样的需求：

会员积分的规则是一元积一分，不满一元不积分，这就要用到向下取整的数学函数 `FLOOR()`；



在打印小票的时候，收银纸的宽度是固定的，怎么才能让打印的结果清晰而整齐呢？这个时候，就要用到 `CONCAT()` 等字符串处理函数；

不同数据的处理方式不同，怎么选择正确的处理方式呢？这就会用到 `IF(表达式, V1, V2)` 这样的条件判断函数；

.....

这些函数对我们管理数据库、提高数据处理的效率有很大的帮助。接下来，我就带你在这解决实际问题的过程中，帮你掌握使用这些函数的方法。

数学函数

我们先来学习下数学函数，它主要用来处理数值数据，常用的主要有 3 类，分别是取整函数 `ROUND()`、`CEIL()`、`FLOOR()`，绝对值函数 `ABS()` 和求余函数 `MOD()`。

知道了这些函数，我们来看看超市经营者的具体需求。他们提出，为了提升销量，要进行会员营销，主要是给会员积分，并以积分数量为基础，给会员一定的优惠。

积分的规则也很简单，就是消费一元积一分，不满一元不积分，那我们就需要对销售金额的数值进行取整。

这里主要用到四个表，分别是销售单明细表、销售单头表、商品信息表和会员信息表。为了方便你理解，我对表结构和数据进行了简化。

销售单明细表：

transactionid (流水编号, 联合主键)	itemnumber (商品编号, 联合主键)	quantity (销售数量)	price (价格)	discount (折扣率)	salesvalue (销售金额)
1	1	2	89	0.99	176.22
1	2	5	5	0.99	24.75
2	1	3	89	0.88	234.96
2	2	6	5	0.88	26.40
3	1	1	89	0.95	84.55
3	2	10	5	0.95	47.50

销售单头表:

transactionid (流水编号, 主键)	transactionno (流水单号)	cashierid (收款机编号)	memberid (会员编号)	operatorid (操作员编号)	transdate (交易时间)
1	0120201201000001	1	1	1	2020-12-01 14:25:56
2	0120201202000001	1	2	2	2020-12-02 10:50:50
3	0120201202000002	1	NULL	1	2020-12-02 12:10:05

商品信息表:

itemnumber (商品编号)	barcode (条码)	goodsname (商品名称)	specification (规格)	unit (单位)	salesprice (售价)
1	1	书	16开	本	89
2	2	笔	NULL	包	5

会员信息表：

memberid (会员编号)	branchid (门店编号)	cardno (会员卡号)	membername (会员名称)	address (地址)	phone (电话)	pid (身份证)	memberpoints (积分值)	memberdeposit (储值金额)
1	1	10000001	张三	北京	13812345678	110198197001018697	0	0
2	1	10000002	李四	天津	18512345678	210123198001019283	0	0

这个场景下，可以用到 MySQL 数学函数中的取整函数，主要有 3 种。

向上取整 CEIL(X) 和 CEILING(X)：返回大于等于 X 的最小 INT 型整数。

向下取整 FLOOR(X)：返回小于等于 X 的最大 INT 型整数。

舍入函数 ROUND(X,D)：X 表示要处理的数，D 表示保留的小数位数，处理的方式是四舍五入。ROUND(X) 表示保留 0 位小数。

现在积分的规则是一元积一分，不满一元不积分，显然是向下取整，那就可以用 FLOOR () 函数。

首先，我们要通过关联查询，获得会员消费的相关信息：

复制代码

```
1 mysql> SELECT
2     -> c.membername AS '会员',    -- 从会员表获取会员名称
3     -> b.transactionno AS '单号',-- 从销售单头表获取单号
4     -> b.transdate AS '交易时间', -- 从销售单头表获取交易时间
```

```

5      -> d.goodsname AS '商品名称', -- 从商品信息表获取商品名称
6      -> a.salesvalue AS '交易金额'
7      -> FROM
8      -> demo.transactiondetails a
9      -> JOIN
10     -> demo.transactionhead b ON (a.transactionid = b.transactionid)
11     -> JOIN
12     -> demo.membermaster c ON (b.memberid = c.memberid)
13     -> JOIN
14     -> demo.goodsmaster d ON (a.itemnumber = d.itemnumber);
15 +-----+-----+-----+-----+-----+
16 | 会员 | 单号 | 交易时间 | 商品名称 | 交易金额 |
17 +-----+-----+-----+-----+-----+
18 | 张三 | 0120201201000001 | 2020-12-01 14:25:56 | 书 | 176.22 |
19 | 张三 | 0120201201000001 | 2020-12-01 14:25:56 | 笔 | 24.75 |
20 | 李四 | 0120201202000001 | 2020-12-02 10:50:50 | 书 | 234.96 |
21 | 李四 | 0120201202000001 | 2020-12-02 10:50:50 | 笔 | 26.40 |
22 +-----+-----+-----+-----+-----+
23

```

接着，我们用 `FLOOR (a.salesvalue)`，对销售金额向下取整，获取会员积分值，代码如下：

复制代码

```

1  mysql> SELECT
2      -> c.membername AS '会员',
3      -> b.transactionno AS '单号',
4      -> b.transdate AS '交易时间',
5      -> d.goodsname AS '商品名称',
6      -> a.salesvalue AS '交易金额',
7      -> FLOOR(a.salesvalue) AS '积分' -- 使用FLOOR函数向下取整
8      -> FROM
9      -> demo.transactiondetails a
10     -> JOIN
11     -> demo.transactionhead b ON (a.transactionid = b.transactionid)
12     -> JOIN
13     -> demo.membermaster c ON (b.memberid = c.memberid)
14     -> JOIN
15     -> demo.goodsmaster d ON (a.itemnumber = d.itemnumber);
16 +-----+-----+-----+-----+-----+
17 | 会员 | 单号 | 交易时间 | 商品名称 | 交易金额 | 积分 |
18 +-----+-----+-----+-----+-----+
19 | 张三 | 0120201201000001 | 2020-12-01 14:25:56 | 书 | 176.22 | 176 |
20 | 张三 | 0120201201000001 | 2020-12-01 14:25:56 | 笔 | 24.75 | 24 |
21 | 李四 | 0120201202000001 | 2020-12-02 10:50:50 | 书 | 234.96 | 234 |
22 | 李四 | 0120201202000001 | 2020-12-02 10:50:50 | 笔 | 26.40 | 26 |
23 +-----+-----+-----+-----+-----+
24 4 rows in set (0.01 sec)

```

你看，通过 FLOOR()，我们轻松地获得了超市经营者需要的积分数据。

类似的，如果用户的积分规则改为“不满一元积一分”，其实就是对金额数值向上取整，这个时候，我们就可以用 CEIL() 函数。操作方法和前面是一样的，我就不具体解释了。

最后，我再来讲一讲舍入函数 ROUND () 的使用方法。

超市经营者提出，收银的时候，应收金额可以被设定四舍五入到哪一位。比如，可以设定四舍五入到元、到角，或者到分。

按照指定的位数，对小数进行四舍五入计算，这样的场景就要用到 ROUND (X,D) 了。它的作用是通过四舍五入，对数值 X 保留 D 位小数。

根据超市经营者的要求，我们把函数 ROUND(X,D) 中的保留小数的位数 D 设置成 0、1 和 2。

如果要精确到分，我们可以设置保留 2 位小数：

 复制代码

```
1 mysql> SELECT ROUND(salesvalue,2) -- D设置成2，表示保留2位小数，也就是精确到分
2 -> FROM demo.transactiondetails
3 -> WHERE transactionid=1 AND itemnumber=1;
4 +-----+
5 | ROUND(salesvalue,2) |
6 +-----+
7 | 176.22 |
8 +-----+
9 1rows in set (0.00 sec)
```

如果要精确到角，可以设置保留 1 位小数：

 复制代码

```
1 mysql> SELECT ROUND(salesvalue,1) -- D设置成1，表示保留1位小数，也就是精确到角
2 -> FROM demo.transactiondetails
3 -> WHERE transactionid=1 AND itemnumber=1;
4 +-----+
5 | ROUND(salesvalue,1) |
6 +-----+
7 | 176.2 |
```

```

8 +-----+
9 1 rows in set (0.00 sec)

```

如果要精确到元，可以设置保留 0 位小数：

 复制代码

```

1 mysql> SELECT ROUND(salesvalue,0)-- D设置成0，表示保留0位小数，也就是精确到元
2 -> FROM demo.transactiondetails
3 -> WHERE transactionid=1 AND itemnumber=1;
4 +-----+
5 | ROUND(salesvalue,0) |
6 +-----+
7 | 176 |
8 +-----+
9 1 rows in set (0.00 se

```

除了刚刚我们所学习的函数，MySQL 还支持绝对值函数 ABS () 和求余函数 MOD ()，ABS (X) 表示获取 X 的绝对值；MOD (X, Y) 表示获取 X 被 Y 除后的余数。

这些函数使用起来都比较简单，你重点掌握它们的含义就可以了，下面我再带你学习下字符串函数。

字符串函数

除了数学计算，我们还经常会遇到需要对字符串进行处理的场景，比如我们想要在金额前面加一个 “¥” 的符号，就会用到字符串拼接函数；再比如，我们需要把一组数字以字符串的形式在网上传输，就要用到类型转换函数。

常用的字符串函数有 4 个。

CONCAT (s1,s2,...)：表示把字符串 s1、s2.....拼接起来，组成一个字符串。

CAST (表达式 AS CHAR)：表示将表达式的值转换成字符串。

CHAR_LENGTH (字符串)：表示获取字符串的长度。

SPACE (n)：表示获取一个由 n 个空格组成的字符串。

接下来我还是借助超市项目中的实际应用场景，来说明一下怎么使用这些字符串函数。

顾客交了钱，完成交易之后，系统必须要打出一张小票。打印小票时，对格式有很多要求。比如说，一张小票纸，57 毫米宽，大概可以打 32 个字符，也就是 16 个汉字。用户要求一条流水打 2 行，第一行是商品信息，第二行要包括数量、价格、折扣和金额 4 种信息。那么，怎么才能清晰地在小票上打印出这些信息，并且打印得整齐漂亮呢？这就涉及对字符串的处理了。

首先，我们来看一下如何打印第一行的商品信息。商品信息包括：商品名称和商品规格，而且商品规格要包含在括号里面。这样就必须把商品名称和商品规格拼接起来，变成一个字符串。

这时，我们就可以用合并字符串函数 `CONCAT ()`，如下所示：

[复制代码](#)

```
1 mysql> SELECT
2 -> CONCAT(goodsname, '(', specification, ')') AS 商品信息 -- 这里把商品名称、括号和
3 -> FROM
4 -> demo.goodsmaster
5 -> WHERE itemnumber = 1;
6 +-----+
7 | 商品信息 |
8 +-----+
9 | 书(16开) |
10 +-----+
11 1 row in set (0.00 sec)
```

这样，我们就得到了商品编号是 1 的商品，它的商品信息是：“书（16 开）”。

第二步，我们来看一下如何打印第二行。第二行包括数量、价格、折扣和金额，一共是 4 种信息。

因为一行最多是 32 个字符，我们给数量分配 7 个字符，价格分配 7 个字符，折扣分配 6 个字符，金额分配 9 个字符，加上中间 3 个空格，正好是 32 个字符。

为啥这么分配呢？我简单解释下。

数量 7 个字符，就是小数点前面给 3 位，小数点后面给 3 位，外加小数点 1 位，最大 999.999，基本满足零售的需求了。

同样道理，价格给 7 位，意思是小数点前面 4 位，小数点后面 2 位，外加小数点，这样最大可以表示 9999.99。

折扣 6 位，小数点后面 2 位，小数点前面 2 位，加上小数点和 “%”，这样是够用的。

金额 9 位，最大可以显示到 999999.99，也够用了。

分配好了各部分信息的字符串大小，我再讲一下格式处理，因为数据的取值每次都会不同，如果直接打印，会参差不齐。这里我以数量为例，来具体说明一下。因为数量比较有代表性，而且比较简单，不像金额或者折扣率那样，有时还要根据用户的需求，加上 “¥” 或者 “%”。

第一步，把数量转换成字符串。这里我们需要用到把数值转换成字符串的 CAST () 函数，如下所示：

[复制代码](#)

```
1 mysql> SELECT
2 -> CAST(quantity AS CHAR) -- 把decimal类型转换成字符串
3 -> FROM
4 -> demo.transactiondetails
5 -> WHERE
6 -> transactionid = 1 AND itemnumber =1;
7 +-----+
8 | CAST(price AS CHAR) |
9 +-----+
10 | 2.000 |
11 +-----+
12 1 rows in set (0.00 sec)
```

第二步，计算字符串的长度，这里我们要用到 CHAR_LENGTH () 函数。

需要注意的是，虽然每个汉字打印的时候占 2 个字符长度，但是这个函数获取的是汉字的个数。因此，如果字符串中有汉字，函数获取的字符串长度跟实际打印的长度是不一样的，需要用空格来补齐。

我们可以通过下面的查询，获取数量字段转换成字符串后的字符串长度：

[复制代码](#)

```
1 mysql> SELECT
```

```

2 -> CHAR_LENGTH(CAST(quantity AS CHAR)) AS 长度
3 -> FROM
4 -> demo.transactiondetails
5 -> WHERE
6 -> transactionid = 1 AND itemnumber =1;
7 +-----+
8 | 长度 |
9 +-----+
10 | 5 |
11 +-----+
12

```

第三步，用空格补齐 7 位长度。这时，我们要用到 SPACE () 函数。

因为我们采用左对齐的方式打印（左对齐表示字符串从左边开始，右边空余的位置用空格补齐），所以就需要先拼接字符串，再在字符串的后面补齐空格：

 复制代码

```

1 mysql> SELECT
2 -> CONCAT(CAST(quantity AS CHAR),
3 -> SPACE(7 - CHAR_LENGTH(CAST(quantity AS CHAR)))) AS 数量
4 -> FROM
5 -> demo.transactiondetails
6 -> WHERE
7 -> transactionid = 1 AND itemnumber = 1;
8 +-----+
9 | 数量 |
10 +-----+
11 | 2.000 |
12 +-----+
13 1 row in set (0.00 sec)

```

除此以外，MySQL 还支持 SUBSTR ()、MID ()、TRIM ()、LTRIM ()、RTRIM ()。我画了一张图来展示它们的含义，你可以了解一下。

函数	功能
SUBSTR (s,n) SUBSTR (s FROM n)	获取从字符串“s”的第“n”个位置开始，到“s”结尾的子字符串
MID (s,n,len)	从字符串“s”的第“n”个位置，获取长度为“len”的子字符串
TRIM (s1 FROM s)	除去字符串“s”的两端所有的字符串“s1”
LTRIM (s)	除去字符串“s”左边的所有空格
RTRIM (s)	除去字符串“s”右边的所有空格

一般来说，关于字符串函数，你掌握这些就足够了。不过，MySQL 支持的字符串函数还有很多，如果你在实际工作中，遇到了更复杂的情况，可以参考 MySQL 官方的 [文档](#)。

条件判断函数

我们刚才在对商品信息字符串进行拼接的时候，会有一种例外的情况，那就是当规格为空的时候，商品信息会变成“NULL”。这个结果显然不是我们想要的，因为名称变成 NULL，顾客会觉得奇怪，也不知道买了什么商品。我们希望，如果规格是空值，就不用加规格了。怎么实现呢？这就要用到条件判断函数了。

条件判断函数的主要作用，就是根据特定的条件返回不同的值，常用的有两种。

IFNULL (V1, V2)：表示如果 V1 的值不为空值，则返回 V1，否则返回 V2。

IF (表达式, V1, V2)：如果表达式为真 (TRUE)，则返回 V1，否则返回 V2。

我们希望规格是空的商品，拼接商品信息字符串的时候，规格不要是空。这个问题，可以通过 IFNULL(specification, '') 函数来解决。具体点说就是，对字段“specification”是否为空进行判断，如果为空，就返回空字符串，否则就返回商品规格 specification 的值。代码如下所示：

 复制代码

```
1 mysql> SELECT
2     -> goodsname,
3     -> specification,
4     -> CONCAT(goodsname, '(', IFNULL(specification, ''), ')') AS 拼接 -- 用条件判断
5     -> FROM
```

```

6      -> demo.goodsmaster;
7  +-----+-----+-----+
8  | goodsname | specification | 拼接 |
9  +-----+-----+-----+
10 | 书        | 16开          | 书(16开) |
11 | 笔        | NULL           | 笔()     |
12 +-----+-----+-----+
13 ~~~~~ (0.00 sec)

```

结果是，如果规格为空，商品信息就变成了“商品信息 ()”，好像还不错。但是也存在一点问题：商品名称后面的那个空括号“()”会让客人觉得奇怪，能不能去掉呢？

如果用 IFNULL (V1, V2) 函数，就不容易做到，但是没关系，我们可以尝试用另一个条件判断函数 IF (表达式, V1, V2) 来解决。这里表达式是 ISNULL(specification)，这个函数用来判断字段"specificaion"是否为空，V1 是返回商品名称，V2 是返回商品名称拼接规格。代码如下所示：

[复制代码](#)

```

1  mysql> SELECT
2      -> goodsname,
3      -> specification,
4      -> -- 这里做判断，如果是空值，返回商品名称，否则就拼接规格
5      -> IF(ISNULL(specification),
6      -> goodsname,
7      -> CONCAT(goodsname, '(', specification, ')')) AS 拼接
8      -> FROM
9      -> demo.goodsmaster;
10 +-----+-----+-----+
11 | goodsname | specification | 拼接 |
12 +-----+-----+-----+
13 | 书        | 16开          | 书(16开) |
14 | 笔        | NULL           | 笔      |
15 +-----+-----+-----+
16 2 rows in set (0.02 sec)

```

这个结果就是，如果规格为空，商品信息就是商品名称；如果规格不为空，商品信息是商品名称拼接商品规格，这就达到了我们的目的。

总结

今天，我们学习了用于提升数据处理效率的数学函数、字符串函数和条件判断函数。

函数	功能
数学计算函数	FLOOR(X)：获得不大于X的最大整数
	CEIL(X)：获取不小于X的最小整数
	ROUND(X,D)：通过四舍五入，获得最接近X的，保留D位小数的数值
	ABS()：获取X的绝对值
	MOD()：获取X被Y除后的余数
	CONCAT (s1,s2,...)：把字符串s1,s2,...拼接起来
	CHAR_LENGTH (s)：获取字符串“s”中字符的个数，一个汉字算一个字符

这些函数看起来很容易掌握，但是有很多坑。比如说，ROUND (X) 是对 X 小数部分四舍五入，那么在“五入”的时候，返回的值是不是一定比 X 大呢？其实不一定，因为当 X 为负数时，五入的值会更小。你可以看看下面的代码：

复制代码

```
1 mysql> SELECT ROUND(-1.5);
2 +-----+
3 | ROUND(-1.5) |
4 +-----+
5 |          -2 |
6 +-----+
7 1 row in set (0.00 sec)
```

所以，我建议你在学习的时候，**多考虑边界条件的场景，实际测试一下**。就像这个问题，对于 ROUND(X,0)，并没有指定 X 是正数，那如果是负数，会怎样呢？你去测试一下，就明白了。

思考题

这节课，我介绍了如何用 FLOOR () 函数来计算会员积分，那么，如果不用 FLOOR () ， 有没有其他办法来实现会员积分的计算呢？

欢迎在留言区写下你的思考和答案，我们一起交流讨论。如果你觉得今天的内容对你有所帮助，也欢迎你分享给你的朋友或同事，我们下节课见。

提建议

12.12 大促

每日一课 VIP 年卡

10分钟，解决你的技术难题

¥159/年 ¥365/年

每日一课
VIP 年卡

仅3天，【点击】图片，立即抢购 >>>

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 09 | 时间函数：时间类数据，MySQL是怎么处理的？

下一篇 11 | 索引：如何提高查询的速度？

精选留言 (3)

写留言



Devo

2021-04-01

如果不用 FLOOR () , 那么可以使用TRUNCATE(X,0)达到相同效果



lesserror

2021-03-30

这个ROUND(X), X为负数的返回结果确实很正数的结果不一样, 值得注意ROUND数值的取值范围。

课后练习题答案是使用: ROUND(X) 表示保留 0 位小数, 只取整数部分的结果, 忽略小数部分。整数部分就是会员积分值。

展开 ∨



右耳朵猫咪

2021-03-30

round函数和convert函数有什么区别呢?

展开 ∨

