

0乘任何数等于0,null加任何数等于null

我们知道 0乘以任何数都等于0， SqlServer 中 `null` 加任何数都等于 `null` 。

故事还得从 一个存储过程说起。插入题外话，文章末尾会有 `0乘以任意数都等于0的数学论证`

目录

- 故事的开头
- 缘起，定位到问题
- 结论
- 0乘以任意数都等于0的数学论证

故事的开头

故事是这样的，LES系统要开发一个空器具上线功能，将获取源库位的逻辑封装成一个 `proc LES.PROC_EMPTY_BOX_ONLINE_REQUIRE_GET_SOURCELOCATION` 。

既然代码写完了，当然步入开心的测试环节了，说罢立马开干，写起测试代码，如下：

```
declare @IsOK_Rst INT;
declare @Rmsg_Rst nvarchar(255);
DECLARE @source_location nvarchar(50);


-- 调存储过程，获取源库位
EXEC LES.PROC_EMPTY_BOX_ONLINE_REQUIRE_GET_SOURCELOCATION
    '5205',--@PLANT
    6,    --@PALLET_TYPE
    '1',--@TASK_TYPE
    @IsOK_Rst OUTPUT ,
    @Rmsg_Rst OUTPUT ,
    @source_location OUTPUT ;

select @IsOK_Rst as IsOK_Rst,@Rmsg_Rst as Rmsg_Rst,@source_location as
source_location
```

我们罗列测试结果的预期表格

执行结果	出参	预期结果
成功	IsOK_Rst	1
	Rmsg_Rst	获取源库位成功！
失败	IsOK_Rst	0
	Rmsg_Rst	具体的错误信息

截图看下 实际的执行结果



```
SQLQuery7.sql - ...ESDB_DL_ZH (120))  SQLQuery6.sql - ...ESDB_DL_ZH (201))* X
```

```
declare @IsOK_Rst INT;
declare @Rmsg_Rst nvarchar(255);
DECLARE @source_location nvarchar(50);

EXEC LES.PROC_EMPTY_BOX_ONLINE_REQUIRE_GET_SOURCELOCATION
'5205', --@PLANT
6, --@PALLET_TYPE
'1', --@TASK_TYPE
@IsOK_Rst OUTPUT,
@Rmsg_Rst OUTPUT,
@source_location OUTPUT;

select @IsOK_Rst as IsOK_Rst, @Rmsg_Rst as Rmsg_Rst, @source_location as source_location
```

99 %

结果 消息

	IsOK_Rst	Rmsg_Rst	source_location
1	0	NULL	NULL

返回NULL 不符合预期

IsOK_Rst 返回的是0 说明执行失败了，那么Rmsg_Rst预期结果应该是 具体的错误信息而不是null，看起来代码有bug了。哪里有bug呢？

缘起，定位到问题

如何定位问题呢？先打开存储过程看看具体实现把。下列代码做了些 删除处理，聚焦主干。

```
BEGIN TRY
```

```
    BEGIN
```

```
        SET @Rmsg='获取源库位成功！';
```

```
        SET @IsOK=1;
```

```
        --判断是否有策略
```

```
        IF NOT EXISTS ( xxxx )
```

```
            BEGIN
```

```
                SET @Rmsg='器具：'+@PALLET_VALUE+'，任务类型：'+@TASK_VALUE+'不存在
器具存储区搜索策略';
```

```
                SET @IsOK=0;  -- A site
```

```
                RETURN;
```

```
            END
```

```
        IF NOT EXISTS( xxxxxx )
```

```
            BEGIN
```

```
                SET @Rmsg='器具：'+@PALLET_VALUE+'，任务类型：'+@TASK_VALUE+'，存储
区不存在状态为【占用】的库位';
```

```
                SET @IsOK=0;  -- B site
```

```
                RETURN;
```

```
            END
```

```

-- 省略了 很多代码

IF(@LOC_SEARCH_STYLE=1)--按序号升序
    BEGIN
        SELECT TOP 1 @SOURCE_LOCATION=DLOC FROM
        LES.TM_BAS_WAREHOUSE_LOCATION WHERE PLANT=@PLANT AND ZONE_NO=@ZONE_NO ORDER BY
        SEQUENCE_NO ASC
    END
    ELSE IF(@LOC_SEARCH_STYLE=2)--按序号降序
    BEGIN
        SELECT TOP 1 @SOURCE_LOCATION=DLOC FROM
        LES.TM_BAS_WAREHOUSE_LOCATION WHERE PLANT=@PLANT AND ZONE_NO=@ZONE_NO ORDER BY
        SEQUENCE_NO DESC
    END
    IF(ISNULL(@SOURCE_LOCATION,'')='')
    BEGIN
        SET @Rmsg='存储区：'+@ZONE_NO+'，不存在库位';
        SET @IsOK=0; -- C site
        RETURN;
    END
END
END TRY
BEGIN CATCH
    -- 记录异常信息
END CATCH;

```

我们带着第一个疑问来阅读代码。

疑问1 代码里 哪个地方 @IsOK 返回的是0，而@Rmsg 设置为null 了？

阅读代码发现 @IsOK 设置为0的地方有三处，分别给出上注释标识，

-- A site

-- B site

-- C site

发现并没有 同时设置 @IsOK=0, @Rmsg=null 的地方。

疑问2 神奇了，是哪里蹦出的@Rmsg=null？

我们做假设, 假设以上三处 A site、B site、C site 都有可能导致 @Rmsg，那么如何验证我们假设呢？

答案很简单，将 @IsOK 设置为特殊的编码 8，至于为何是8 而不是7啊或者其他数，我的回答是 anyway，代码中 对@IsOK的赋值只有 1和0，只要不是1和0 即可。

说罢，立马修改代码，先验证是不是A Site 导致的 @Rmsg 为null 。如下：

```

1  .
2  BEGIN TRY
3  .....
4  BEGIN
5  SET @Rmsg='获取源库位成功!';
6  SET @IsOK=1;
7
8  --判断是否有策略
9  IF NOT EXISTS (.....)
10 BEGIN
11 SET @Rmsg='器具: '+@PALLET_VALUE+', 任务类型: '+@TASK_VALUE+'不存在器具存储区搜索策略';
12 SET @IsOK=0;.....A.site
13 RETURN;
14 END
15
16 IF NOT EXISTS (.....)
17 BEGIN
18 SET @Rmsg='器具: '+@PALLET_VALUE+', 任务类型: '+@TASK_VALUE+', 存储区不存在状态为【占用】的库位';
19 SET @IsOK=0;.....B.site
20 RETURN;
21 END
22 --省略了很多代码
23
24 IF (@LOC_SEARCH_STYLE=1) --按序号升序
25 BEGIN
26 SELECT TOP 1 @SOURCE_LOCATION=DLOC FROM LES.TM_BAS_WAREHOUSE_LOCATION WHERE PLANT=@PLANT AND ZONE_NO=@ZONE_NO
27 END
28 ELSE IF (@LOC_SEARCH_STYLE=2) --按序号降序
29 BEGIN
30 SELECT TOP 1 @SOURCE_LOCATION=DLOC FROM LES.TM_BAS_WAREHOUSE_LOCATION WHERE PLANT=@PLANT AND ZONE_NO=@ZONE_NO
31 END

```

执行结果：

SQLQuery7.sql -...ESDB_DL_ZH (120))*

SQLQuery6.sql -...ESDB_DL_ZH (201))*

```
declare @IsOK_Rst INT;
declare @Rmsg_Rst nvarchar(255);
DECLARE @source_location nvarchar(50);

EXEC LES.PROC_EMPTY_BOX_ONLINE_REQUIRE_GET_SOURCELOCATION
    '5205', --@PLANT
    6,      --@PALLET_TYPE
    '1',    --@TASK_TYPE
    @IsOK_Rst OUTPUT ,
    @Rmsg_Rst OUTPUT ,
    @source_location OUTPUT ;

select @IsOK_Rst as IsOK_Rst, @Rmsg_Rst as Rmsg_Rst, @source_location as source_location
```

99 %

结果 消息

	IsOK_Rst	Rmsg_Rst	source_location
1	8	NULL	NULL

结果喜人呐，在A site 便试验出结果了，就是这个地方导致的 @Rmsg 为null。

单独拎出来这行代码

```
SET @Rmsg='器具：'+@PALLET_VALUE+'，任务类型：'+@TASK_VALUE+'不存在器具存储区搜索策略';
```

这无论如何也不会导致 @Rmsg 被赋值为 null 啊，已经有字符串存在了，哪怕变量 @PALLET_VALUE 为null也不可能.....，

慢着 是不是就是 @PALLET_VALUE 或 @TASK_VALUE 为null 导致的字符串拼接结果就是 为null ?

直接上代码，验证猜想

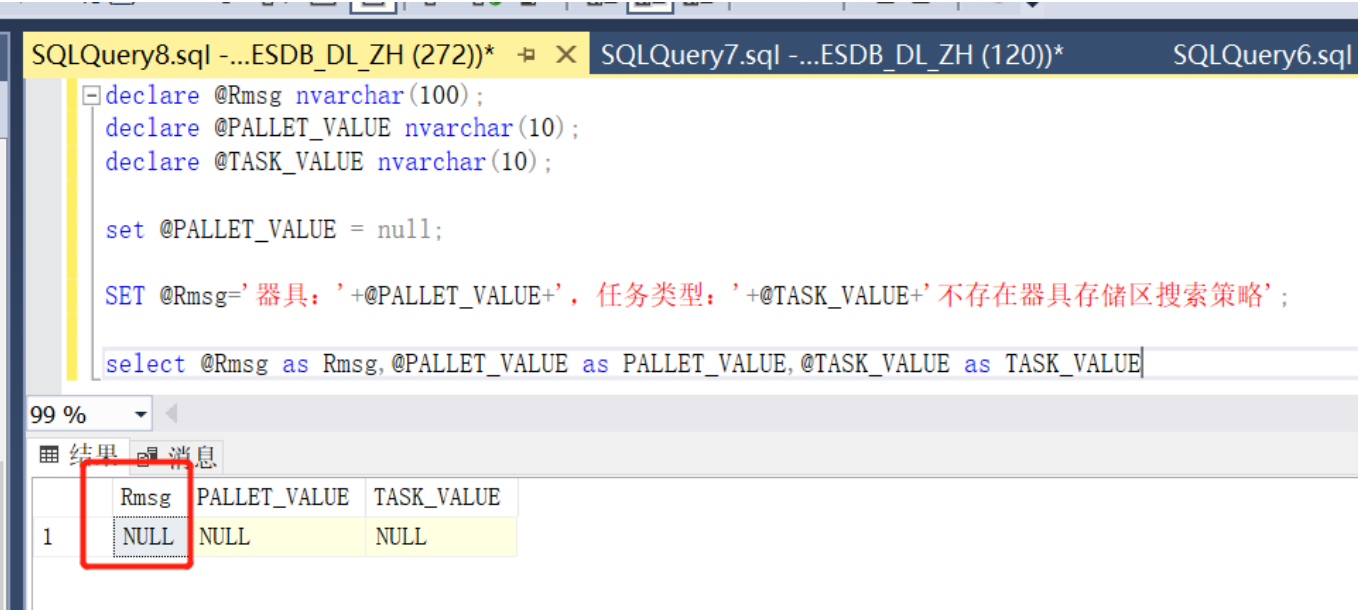
```
declare @Rmsg nvarchar(100);
declare @PALLET_VALUE nvarchar(10);
declare @TASK_VALUE nvarchar(10);

set @PALLET_VALUE = null;

SET @Rmsg='器具：'+@PALLET_VALUE+'，任务类型：'+@TASK_VALUE+'不存在器具存储区搜索策略';

select @Rmsg as Rmsg,@PALLET_VALUE as PALLET_VALUE,@TASK_VALUE as TASK_VALUE
```

看执行结果：



看结果，果然是因为null的缘故，导致字符串拼接的 整个结果是 null。

找到问题了很开心，那么多想一步，是否 null 的 任何运算结果都是null 呢？

计算机中存在三种运算: 算术运算、逻辑运算、位运算，这些发现留待您来验证结果把。哈哈

结论

我们通过提出假设，验证假设，这么一步步的探究出问题的 真相：SqlServer 中 null 加任何数都等于 null 我想 这种做事的方法，是大家可以借鉴的，工程领域就是这样试错得到 答案。

0乘以任意数都等于0的数学论证

写在后面，这里是一些数学上的 趣味阅读了，真佩服您，竟然还真的看这里的 推导，哈哈。

把代数里的公理列一列哈，我们都是讲道理的人,做事都会有理有据,不可能骗人的

- A1 加法交换律：对任意两个数a和b，都有 $a+b=b+a$
- A2 加法结合律：对任意三个数 a、b和c,有 $a+(b+c)=(a+b)+c$
- A3 0是加法单位元：对任意数a,有 $0+a=a$

M1 乘法交换律 : 对任意两个数 a 和 b ,有 $ab=ba$

M2 乘法结合律 : 对任何三个数 a 、 b 和 c ,有 $a (bc) = (ab) c$

M3 1 是乘法单位元 : 对任意数 a ,有 $1a=a$

D 分配律 : 对任意三个数 a 、 b 和 c ,有 $(a+b)c = ac+bc$

$0 = 1 \times 0$	M3
$= (0 + 1) \times 0$	A3
$= 0 \times 0 + 1 \times 0$	D
$= 0 \times 0 + 0$	M3
$= 0 + 0 \times 0$	A1
$= 0 \times 0$	A3