SQL Server null知多少?

null是什么?

不知道。我是说,他的意思就是不知道 (unknown)。

它和true、false组成谓词的三个逻辑值,代表"未知"。与true和false相比,null最难以令人捉摸,因为它没有明确的值,在不同的场景下,它能代表不同的含义。下文以例子的方式给大家分享下null使用的典型场景及对应的用法。

1.check约束与null

之前在SQL ServerCentral.com上看到一个关于check约束的null问题,

作者创建了一个表,在字段orderstatus上设置了check约束,只能插入指定的value的行,现在插入几行数据,其中有一行的value为null,最后问最终有几行可以插入成功。

原文如下:

I want to ensure that the status column for my Orders table only contains specific values. I decide to use this code:

```
create table Orders
( OrderID int primary key
, OrderTotal MONEY
, OrderStatus VARCHAR(20)
constraint Orders_Status_Code check( OrderStatus in ('ACTIVE', 'INACTIVE', 'TBD'))
);
go
```

Now I want to insert data into the table. I run this batch.

```
insert Orders select 1, 435.43, 'Active'
insert Orders select 2, 554.66, 'InActive'
insert Orders select 3, 129.12, 'Not Active'
insert Orders select 4, 1228.00, NULL
```

How many rows are in the table? I am running on a default, SQL Server 2014 instance with US English defaults.

(大家先想想答案,如果没有把握就找个测试环境试一试)

《T-SQL基础教程》中关于check约束与null的描述,著者用了一句言简意赅的口诀"check约束拒绝false,接受true和null"。

在上面的例子中,当orderstatus为'Avative'和'InActive'时,check约束判断的结果是true,所以会插入成功,当为'Not Active'判断的结果为false,插入不成功,最后当为'Null'时,判断的结果是null,插入成功。

所以,正确答案是3。

2.比较运算与null

null一个特殊性在于它无法比较(和计算)。null与任何值的任何比较(和计算)都等于null。(unique约束除外,在unique约束中,null是相等的,同一个字段不允许出现两次null)

比如判断null=null的结果是null,判断null<>null的结果也还是null。下面我以不等于(<>)为例,演示比较运算对null的判断。

我先创建一个表,然后插入多行数据,其中有一行orderstatus的值为null,

```
if object_id(N'Orders') is not null drop table orders

create table Orders
( OrderID int primary key
, OrderTotal MONEY
, OrderStatus VARCHAR(20)
);
go

insert Orders select 1, 435.43, 'Active'
insert Orders select 2, 554.66, 'InActive'
insert Orders select 3, 129.12, 'Not Active'
insert Orders select 4, 1228.00, NULL
```

现在我执行了一个where orderstatus<>'Active' 的查询,

```
select * from orders where OrderStatus<>'Active'
```

大家想想null所在的行会不会在查询结果里面。

在上面的例子中,当orderstatus为'InActive' 和'Not Active' 时,where条件判断的结果为true,但当orderstatus为'null' 时,where OrderStatus<>'Active'等价于where null <>'Active',而null与任何一个值的比较结果还是null,所以where条件判断的结果为null。

在SQL Server中,where筛选的原则是"接受true,拒绝false和null"(《T-SQL基础教程》)。所以orderstatus为'InActive' 和'Not Active'的行显示在结果集总,而orderstatus为null的行不会出现在结果集中。

最终, 正确答案是: 只会返回两行

	₩0rderID	OrderTotal	OrderStatus
1	2	554.66	InActive
	3	129. 12	Not Active

3.Not in与null和Not exists与null

not in和not exists都可以用来判断某个对象的存在与否,在大多数场景下两者可以相互替换,但在遇到null时,因为前者是三值逻辑(true|false|unknow)判断而后者只会返回true或false,因此处理的结果会有很大不同。

为了演示两者的区别,我们还是沿用上文的表,分别使用not in和not exists执行一个查询,找出OrderStatus 不为'Active'和'InActive'的行。

```
if object_id(N'Orders') is not null drop table orders

create table Orders
( OrderID int primary key
, OrderTotal MONEY
, OrderStatus VARCHAR(20)
);
go

insert Orders select 1, 435.43, 'Active'
insert Orders select 2, 554.66, 'Inactive'
insert Orders select 3, 129.12, 'Not Active'
insert Orders select 4, 1228.00, NULL
```

3.1Not In与null

在下面这个查询中,where子句中使用not in来过滤数据,where子句的条件是OrderStatus not in ('Active', 'InActive'),我们期望结果集中包含orderstatus为'Not Active'、'NULL'这两行的数据。

```
select * from orders where OrderStatus not in ('Active','InActive')
```

这个查询中,当OrderStatus为null时, 原where子句等价于where null <>'Active' AND null<>'InActive', 这就变成了上文中介绍的比较运算与null的问题。where的判断结果还是null,所以该行不会出现在结果集中。而当OrderStatus为'Not Active'时,满足where筛选的为true的条件,会显示在结果集中。

最终,正确答案是:只有一行。

	OrderID	OrderTotal	OrderStatus
1	3	129.12	Not Active

说明: in与null的关系与此同理。

3.2Not exists与null

现在我们还是期望结果集中包含orderstatus为'Not Active'、'NULL'这两行的数据,这次用Not exists。

在这个查询中,子查询先求出OrderStatus='Active' or OrderStatus='InActive的行,然后外部查询用not exists过滤子查询的结果,将剩下的行显示在最终结果集中。

```
SELECT *

FROM orders AS o1

WHERE NOT EXISTS(

SELECT *

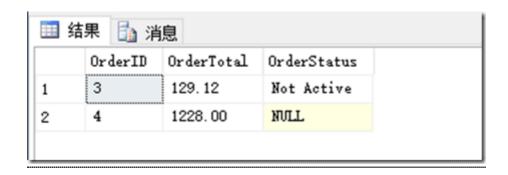
FROM orders AS o2

WHERE o1.OrderStatus = o2.OrderStatus

AND ( o2.OrderStatus = 'Active'

OR o2.OrderStatus = 'InActive'

));
```



为了方便理解,我们将子查询改写成自表连接的方式,

```
select * from orders as o2 where o1.OrderStatus=o2.OrderStatus and (o2.OrderStatus='Active' or
```

改写成:

```
SELECT *
FROM orders AS o2
```

```
INNER JOIN orders of ON of OrderStatus = 02.OrderStatus

AND ( o2.OrderStatus = 'Active'

OR o2.OrderStatus = 'InActive'

);
```

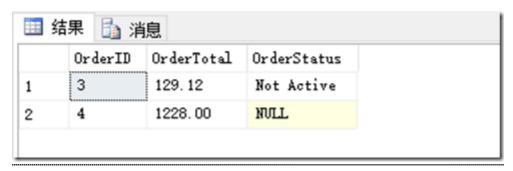
返回的结果集为:



然后我们再看外层查询,

外部查询期望使用not exists返回orders表中不包含子查询结果集的行,也就是说,只要orders表没有子查询结果 集中的行就返回true,否则返回false(只有存在和不存在,没有unknown的说法)。

按照这个逻辑, orderID为3和4的行不在子查询的结果集中, 因此not exists判断为true, 而orderID为1和2的行已包含在子查询的结果集中, 所以not exists判断为false。最后根据where筛选"接受true, 拒绝false和null"的原则, 最终只有orderID为3和4的行显示在结果集中。



说明: exists与null的关系与此同理。

3.3Not in和Not exists的区别

not in实际上是对一个对象的比较运算,而比较存在true|false|unknow三种逻辑值。

not exsits判断某个对象存在或者不存在,它只有这两种状态,没有unknown的说法。因此相比not in而言,not exists只会有true和false这两种逻辑值。

总结:

上文介绍了null在不同场景中的含义,考虑到SQL不同的语言元素对null的不同处理方式,平常我们在写SQL语句的时候应该清晰思考自己编写的每个查询对null或三值逻辑的处理,避免出现逻辑错误。