

文章

[姚鑫](#) · 四月 3 阅读大约需分钟

第101章 SQL函数 NVL

第101章 SQL函数 NVL

测试 NULL 并返回适当表达式的函数。

大纲

`NVL(check-expression, replace-expression)`

参数

- check-expression - 要计算的表达式。
- replace-expression - 如果 check-expression 为 NULL, 则返回的表达式。

NVL 返回与检查表达式相同的数据类型。

描述

NVL 计算检查表达式并返回两个值之一：

- 如果 check-expression 为 NULL, 则返回 replace-expression。
- 如果 check-expression 不为 NULL, 则返回 check-expression。

参数 check-expression 和 replace-expression 可以具有任何数据类型。如果它们的数据类型不同, SQL 会将 replace-expression 转换为 check-expression 的数据类型, 然后再进行比较。返回值的数据类型始终与 check-expression 的数据类型相同, 除非 check-expression 为字符数据, 此时返回值的数据类型为 VARCHAR2。

请注意, NVL 支持 Oracle 兼容性并且与 ISNULL 函数相同。

日期和时间显示转换

某些检查表达式数据类型要从逻辑模式转换为 ODBC 模式或显示模式。例如 DATE 和 TIME 数据类型。如果 replace-expression 的值不是同一个数据类型, 这个值不能在 ODBC 模式或 Display 模式转换, 会产生 SQLCODE 错误: -146 for DATE data type; -147 用于 TIME 数据类型。例如, ISNULL(DOB, 'nodate') 不能在 ODBC 模式或 Display 模式执行; 它发出 %msg Error: 'nodate' is an invalid ODBC/JDBC Date value or Error: 'nodate' is an invalid DISPLAY Date value.。要在 ODBC 模式或显示模式执行此语句, 必须将值转换为适当数据类型 ISNULL(DOB, CAST('nodate' as DATE))。这导致日期 0, 显示为 1840-12-31。

NULL 处理函数比较

表显示了种 SQL 比较函数。如果逻辑比较测试为 True(A 与 B 相同), 则每个函数返回一个值, 如果逻辑比较测试为 False(A 与 B 不同), 则返回另一个值。这些函数允许执行 NULL 逻辑比较。不能在相等或不相等条件比较中指定 NULL。

SQL Function	Comparison	Test Return Value
IFNULL(ex1,ex2) [two-argument form]	ex1 = NULL	True returns ex2 False returns NULL
IFNULL(ex1,ex2,ex3) [three-argument form]	ex1 = NULL	True returns ex2 False returns ex3
{fn IFNULL(ex1,ex2)}	ex1 = NULL	True returns ex2 False returns ex1
ISNULL(ex1,ex2)	ex1 = NULL	True returns ex2 False returns ex1
NVL(ex1,ex2)	ex1 = NULL	True returns ex2 False returns ex1
NULLIF(ex1,ex2)	ex1 = ex2	True returns NULL False returns ex1
COALESCE(ex1,ex2,...)	ex = NULL for each argument	True tests next ex argument. If all ex arguments are True (NULL), returns NULL. False returns ex

示例

以示例返回换表达式 (99), 因为检查表达式为 NULL:

```
SELECT NVL(NULL,99) AS NullTest
```

```
99
```

以示例返回检查表达式 (33), 因为检查表达式不为 NULL:

```
SELECT NVL(33,99) AS NullTest
```

```
33
```

如果 FavoriteColors 为 NULL, 则以动态 SQL 示例返回字符串 "No Preference"; 否则, 它返回 FavoriteColors 的值:

```
/// d ##class(PHA.TEST.SQLFunction).Nvl()
ClassMethod Nvl()
{
    s myquery = 3
    s myquery(1) = "SELECT Name,"
    s myquery(2) = "NVL(FavoriteColors,'No Preference') AS ColorChoice "
    s myquery(3) = "FROM Sample.Person"
    s tStatement = ##class(%SQL.Statement).%New()
    s qStatus = tStatement.%Prepare(.myquery)
    s rset = tStatement.%Execute()
    d rset.%Display()
    w !,"End of data"
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).Nvl()
Name      ColorChoice
yaoxin    $lb("Red","Orange","Yellow")
xiaoli    No Preference
??        No Preference
```

```
?? No Preference
?? No Preference
?? $lb("Red", "Orange", "Yellow", "Green")
?? $lb("Red", "Orange", "Yellow", "Green", "Green")
```

[#SQL](#) [#Caché](#)

源 URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC101%E7%AB%A0-sql%E5%87%BD%E6%95%B0-nvl>