

【NULLを更に詳しく】

前回の投稿でNULLの性質を説明しましたが、更に詳しく考察していきます。

NULLは値ではないため、他の値と比較はできない

NULLと他の値を比較した場合、その結果は必ず**unknown**（不定）となる。
例えば、 $(5 > \text{NULL})$ or $(4 > 5)$ or $(\text{NULL} = 5)$ は、**unknown** or false or **unknown**である。
ここで、**unknown**とfalseの論理和が出てきたため、true（真）とfalse（偽）の2値論理では解決できない。
そこで登場するのが、2値論理に**unknown**を加えた3値論理である。
3値論理における基本は以下のとおりである。

- ①強い順に並べると、論理和（OR）は、true, unknown, falseである。
- ②強い順に並べると、論理積（AND）は、false, unknown, trueである。
- ③論理和及び論理積におけるunknown同士の結果はunknownであり、unknownの否定もunknownである。

以上のことをまとめたものが下表（論理和、論理積、否定）である。

表：論理和

X	Y	結果
true	true	true
true	false	true
true	unknown	true
false	false	false
false	unknown	unknown
unknown	unknown	unknown

表：論理積

X	Y	結果
true	true	true
true	false	false
true	unknown	unknown
false	false	false
false	unknown	false
unknown	unknown	unknown

表：否定

X	結果
true	false
false	true
unknown	unknown

よって、**unknown** or false or **unknown** は上表に当てはめると**unknown**となる。

属性のデータ数をカウントする時、NULLの行はカウントしない

- ①前提として、属性Aのデータ型は**数値**である。データ型は**文字列**、**数値**、**日付**の3つに大別できるが、集約関数の**SUM**や**AVG**は**数値**だけ可、**MAX**や**MIN**は**数値**に加え**日付**も可
- ②よって、**SUM** (A) は属性Aの合計。ただし、**SUM** (CASE WHEN 条件 THEN **A** ELSE **0** END) は、条件を満たすAの合計。それ以外は0が加算されるので、合計に影響なし。
- ③また、**COUNT** (CASE WHEN 条件 THEN **1** ELSE **0** END) は、条件を満たす行数をカウントすると解釈されがちであるが、1や0の数字は関係なく、数字を与えらると、1行としてカウントされる。
- ④よって、NULLの行はカウントしない性質を利用して、**COUNT** (CASE WHEN 条件 THEN **1** ELSE **NULL** END) とすることで、条件を満たさない場合はNULLなのでカウントされなくなる。

注) 上記①でデータ型は3つに大別できると説明したが、その他に画像、動画及び音声等の様々なバイナリーデータを扱うデータ型であるBLOB (**B**inary **L**arge **O**bject) がある。

UNIQUE制約のNULLは可

主キー制約とUNIQUE制約は両者とも一意性制約が含まれますが、下表（比較表）のとおり、相違点は多くあります。

表：比較表

制約名	一意性制約	設定数	非NULL属性	特記事項
主キー制約	有	列又は列の組は 1つのみ	有	主キーの真部分集合も NULLは不可
UNIQUE制約	有	列又は列の組を 複数指定可能	無	<ul style="list-style-type: none">● NULLの重複が可の場合もある。● 逆にNULL禁止は以下のとおり CREATE TABLE 商品 (..... 携帯番号 CHAR(15) UNIQUE NOT NULL ,)

NULLを回避する方法

COALESCE(コウアレス)関数は、NULL(欠損値)をデフォルト値に置き換える関数であり、例えば「SELECT ID, SUM(A), COALESCE(MIN(B), 0)…」とすることで、MIN(B)がNULLになる場合には、NULLではなく、ゼロ表示されます。

また、応用として「COALESCE(C, D, 0)」とすると、以下の順番で処理されます。

- ① 属性CがNOT NULLならば**Cを表示**。CがNULLならば、②へ
- ② 属性DがNOT NULLならば**Dを表示**。DがNULLならば、③へ
- ③ **ゼロを表示**

ANDとOR

先ほどの3値理論の際に、ふとANDとORはどちらが強いのかという疑問が出てきたので調べてみると、結果としては、優先順位が高い順に、**NOT、AND、OR**となります。なので、優先順位を変える時は括弧等を使用するしかありません。