

11null値(22)(23)

null値(22)(23)

真偽のいずれかの値を返す条件判断や、それらの間の論理演算は、多くのプログラム 言語においてもなじみのものであるが、データベースでは、論理演算と関連して特殊な 値が定義されているのが大きな特徴である。それがnull値である。null値は、テーブル 中にカラムの値として現れることができる。値がnullだということは、その値がいまだ 我々に知られていないこと（未知）、あるいは、その値が未だに定まっていないこと（未定）、あるいは、どうなっているのか分からない（不明）ことを表していると解釈 される。

今、テーブル「仕入」中のある部品の単価がnullであったとしよう。すなわち、ある部品の単価がその時点で不明だったとしよう。それでは、この部品は次の二種類の検索の どちらで検索されるであろうか？

```
select 部門番号 , 部品番号, 業者番号, 数量
from 仕入
where 単価 < 100
```

```
select 部門番号 , 部品番号, 業者番号, 数量
from 仕入
where 単価 >= 100
```

一見するとすべての部品は、どちらかの検索で見つかるように思えるかも知れない。しかし、定価のカラムがnull値をとっているということは、単価が分からないということであり、当然、「この部品の単価は百円未満である」という命題と「この部品の単価は百円以上である」という命題のどちらが正しいか分からないということを意味している。だから、単価がnull値の部品は、上の二つの検索のいずれでも見つけだすことは出来ないことになる。こうした解釈は、null値というのが、単にカラムの値であるだけでなく、真あるいは偽とは区別された、「不明・未定・未知」といった状態を表す第三の真理値として も、解釈されるべきことを示唆している。

null値の導入は、データベースの実際の運用上では非常に大きなメリットを持っている が、なかなか面倒な問題も引き起こす。次のような検索を考えてみよう。

```
select 部門番号 , 部品番号, 業者番号, 数量
from 仕入
where 単価 = null
```

気持ちとしては、この検索は、単価がnull値をとっている部品を探しているのであるが、残念ながらこの検索は成功しない。確かに、null = null であるならば、この検索は 成功するのだが、ここで比較されているのは単価の値である。未知な単価と未知な単価 とは、一般には等しくはない。そうした意味では、カラムの値としてのnull値は、自分 自身とさえ一致しないのである。

それでは、null値をとる項目をどうしたら選び出すことが出来るのであろうか？ SQLでは、その為に、次のような構文が特別に用意されているのである。

例 22: 単価が不明の本を検索

```
select 部門番号, 部品番号, 業者番号, 数量
from 仕入
where 単価 is null
```

部門番号	部品番号	業者番号	数量
10	20	1	500

次の例は、いうまでもなく、例22の逆、すなわち、null値をとらない項目の検索である。ここでも、'is not null' を、'!= null' で置き換えることは出来ないことに 注意しよう。

```
select 部門番号, 部品番号, 業者番号, 数量
from 仕入
where 単価 != null (この書式はSQLアナライザ2000ではサポートしない)
```

例 23: 定価が分かっている部品を検索

```
select 部門番号, 部品番号, 業者番号, 数量
from 仕入
where 単価 is not null
```

この検索結果で158件ヒットする。