

■ フィールドの型

数値を扱う上で最も気をつけたいのはフィールドの型をどれにするかだ。
たとえば、次のようなフィールドがあるとしよう。

フィールド名	型	値
数値 1	倍精度浮動小数点型	2.3
数値 2	通貨型	2.3

次の計算を実行してみると結果はどうなるか。

[数値 1]*100 = ?

[数値 2]*100 = ?

答えは同じではない。

倍精度の型を用いた場合は 229 となり、通貨型を用いた場合は 230 となる。

これはコンピュータの丸め誤差からくるものだが、このことから販売管理システムなどの数量や単価、税率、値引率などのお金に関する項目には全て通貨型を利用することを推奨する。

ただし、気をつけておきたいのは、

通貨型では小数点以下第 4 位までしか保証されない

ということ。

仮にそれよりも小さい数値が発生しうる場合は倍精度などの型を用い、計算式に次のような方法を用いる。

例：Ccur([数値 1])*100

Ccur 関数は値を通貨型に変換する関数である。このように、あらかじめ通貨型に変換すればその結果は期待したものとなる。

オートナンバー型フィールドは整数として扱われる。よって、このオートナンバー型のフィールドをキーとして関連を持たせる外部テーブルのフィールド型は整数（一般には長整数）でなければならない。

注意

入力用トランザクションテーブルにこのオートナンバーを用いる場合は注意が必要だ。たとえば、オートナンバーを有するテーブルのデータをすべて削除し、その後にデータベースに最適化を実行すると、そのテーブルのオートナンバーフィールドの初期値は元に戻ってしまうからだ。

数値フィールドで小数点以下が発生しないことがあらかじめ分かっている場合は、整数型または長整数型を用いる。

日付や時刻を用いる場合は素直に日付/時刻型を用いる。

商品コードや品名を用いるフィールドはテキスト型でよい。

Access mdb のデータ型またはフィールド型についてのおさらい

テキスト型

最大で 255 文字まで格納できる。"FieldSize/フィールドサイズ" プロパティで、入力できる最大文字数を設定する。

メモ型

最大で 65,536 文字まで格納できる。

数値型

バイト型、整数型、長整数型、倍精度浮動小数点型などが格納される。数値型の種類は、"FieldSize/フィールドサイズ" プロパティで定義する。

日付/時刻型

日付と時刻に使用する。

通貨型

通貨の値および数値データが格納される。計算時に丸めによる誤差が生じないように配慮されている。

オートナンバー型

レコードが追加されると、連続番号か乱数が自動的に挿入される。連続番号は、1 から付けられる。

Yes/No 型

Yes/No、True/False、On/Off など、二者択一のデータに使用する。Null 値を格納することはできない。

OLE オブジェクト型

OLE (OLE : アプリケーション間で情報を共有するために使用するプログラム統合技術) プロトコルを使用して他のプログラムで作成された、Microsoft Word 文書、Microsoft Excel ワークシート、画像、音声、その他のバイナリ データなどの OLE オブジェクトに使用する。最大で 1GB まで格納できる。

ハイパーリンク型

ハイパーリンクに使用する。ハイパーリンクには UNC または URL を指定し、インターネットやイントラネット上のオブジェクト、ドキュメント、Web ページ、その他のリンク先の場所を指定するアドレス。たとえば、<http://www.microsoft.com/> など) を利用する。最大で 64,000 文字まで格納できる。

ルックアップ ウィザード

コンボ ボックスを使って別のテーブルまたは値リストの値を選択する形式のフィールドを作成する (Access 固有のフィールド)。データ型リストでこのオプションを選択すると、この定義を行うウィザードが起動される。ルックアップ フィールド に対応する主キーの記憶領域と同じサイズ (通常は 4 バイト) を必要とする。

■ 主キーについて

Access のテーブルは主キーが存在しなくても許可される。しかし、Access のテーブルを SQL Server などに移行しようとするならば注意が必要だ。

あらかじめ移行がスケジュールにあるのなら Access のテーブル設計時にも主キーを置くということを推奨する。

■ Null と長さ 0 の文字列

Null とは、値が入力されていないフィールドの値のこと。

ならば、文字列型フィールドの長さ 0 の文字列 とは？

長さ 0 の文字列 とは、
文字を 1 つも含まない文字列。長さ 0 の文字列を使うと、意図的にフィールドに値がないようにしていることを示すことができる。
長さ 0 の文字列を入力するには、スペースを含めずに二重引用符を二度続けて (" ") 入力する。

実験してみると分かるが、長さ 0 の文字列を入力したフィールドと、もともとそこに一度も入力が行われていないもの (Null 状態) とでは見た目は同じである。見た目は同じだが実際の値は同じではないということになる。

しかし、販売管理システムなどでこの両者の使い分けに気を配ることはほとんどない。問題となるのはむしろ Null の扱いだ。

一般に、販売管理システムなどでの数量フィールドでは値を意図的に Null としたいことも多いが、Null を許可した場合はその後の計算式に配慮が必要となる。

たとえば、次の計算式を見て欲しい。

例：[数量]*[単価]

上記の計算で数量または単価のいずれかに Null が存在すると計算式は成り立たなくなる。クエリーなどでこういった計算式を記述して実行されると、Null が見つかったレコードでエラーが返される。

そこで、

例：Nz([数量],0)*Nz([単価],0)

などと記述する必要があるわけだ。

Nz 関数は、値が Null 値である場合、0、長さ 0 の文字列 ("")、または別の特定の値を返す。たとえば、この関数を使用して、Null 値を別の値に変換し、式が Null 値に評価されないようにすることができる。

例：Nz([数値],0)

← 値が Null なら 0 を返す

例：Nz([文字列],"値がありません") ← 値が Null なら "値がありません" という文字列を返す

数値フィールドで、Null を許可せずに 0 または何がしかの数値を要求したいのであれば、既定値を 0 とし、値要求を”はい”とする。こういった Null に対するわずらわしい配慮を避けるために、このような設定とすることはよくある。

以上