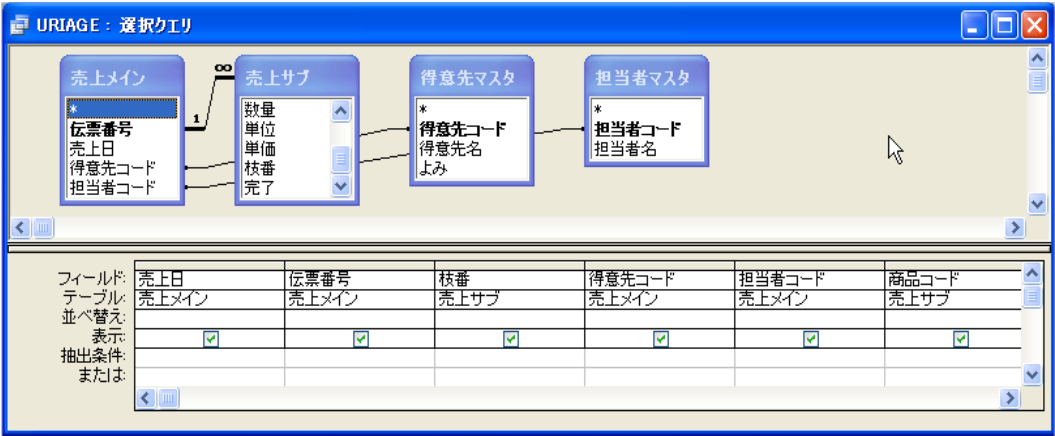


クエリの正体

平成 19 年 1 月 19 日
(有)ビット
浦 秀樹

今回はクエリについて考える。
クエリとはデータベースに対する問い合わせのこと。あるときはデータの中から目的のレコードを抽出し、あるときは目的のレコードを更新したり削除したりする。そういった作業を総称してクエリという。
クエリといって思い出されるのが次のような図だろう。



これは Access のクエリビルダであるが、Access 初心者はこのクエリビルダで作成されたものだけがクエリだと思い込んではいないだろうか。実は、クエリビルダで見えているものは見せ掛けであって、その実態は実は別のものであるということをご存知だろうか。

クエリの詳細とはどのようなものなのかを説明する前にクエリの種類について整理しておこう。
Access のクエリには次のような種類がある。 Access のヘルプより引用

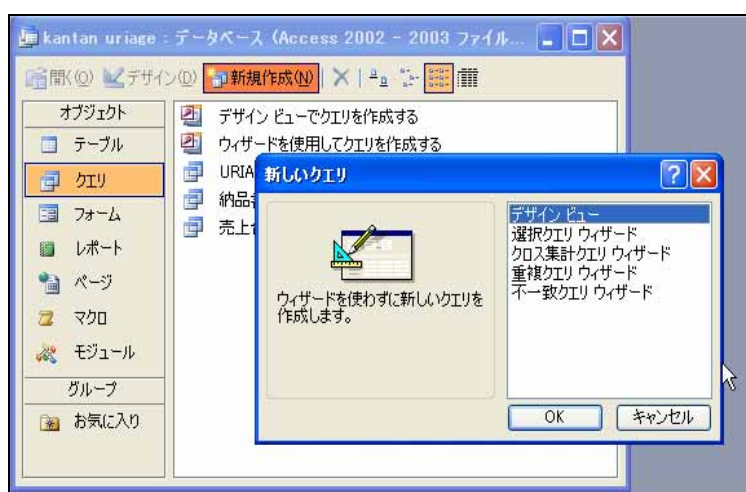
名称	説明
選択クエリ	通常で最も使用するクエリです。このクエリを使うと、1 つまたは複数のテーブルのデータを取得し、結果をデータシートに表示できます。データシートのレコードは、いくつかの制約がありますが、更新することができます。また、選択クエリを使ってレコードをグループ化し、データの合計、カウント、平均などを計算できます。
更新クエリ	1 つまたは複数のテーブルにある一連のレコードをまとめて変更します。たとえば、すべての乳製品の価格を 10% 引き上げたり、特定の職種の社員に対する給料を 5% 引き上げたりできます。更新クエリを使うと、既存のテーブルのデータを変更できます。
追加クエリ	1 つまたは複数のテーブルの一連のレコードを、1 つまたは複数のテーブルの最後に追加します。たとえば、新規の得意先を獲得し、それらの得意先の情報テーブルを含むデータベースを入手したとします。追加クエリを使用すると、入手した情報を [得意先] テーブルに追加できるので、その情報を自分のデータベースに入力する手間を省くことができます。
削除クエリ	1 つまたは複数のテーブルから一連のレコードを削除します。たとえば、削除クエリを使用すると、廃止になった商品や受注がないものを削除できます。削除クエリでは、レコード内の選択されたフィールドではなく、必ずレコード全体を削除します。
クロス集計クエリ	クロス集計クエリは、データをより分析しやすくするために、データの計算や構造変更を行うときに使用します。合計、平均、カウントなどの方法でデータを集計でき、集計値は、2 つの情報によって分類されます。分類に使用される情報の 1 つはデータシートの左

	端に表示され、もう 1 つはデータシートの上端に表示されます。
テーブル作成クエリ	1 つまたは複数のテーブルにあるデータのすべてまたは一部から新規テーブルを作成します。テーブル作成クエリは、ほかのデータベースにエクスポートするテーブル、または古いレコードを含むヒストリ テーブルを作成する場合に便利です。
SQL クエリ	SQL クエリは、SQL ステートメント (SQL 文字列/ステートメント : SELECT、UPDATE、または DELETE などの SQL コマンドを定義し、WHERE および ORDER BY などの句を含む式。通常、SQL 文字列/ステートメントはクエリおよび集計関数で使用されます。) を使って作成するクエリです。SQL (構造化照会言語) を使用すると、Access などのリレーショナル データベースの問い合わせ、更新、および管理を行うことができます。

ただし、クエリビルダではテーブル作成クエリは作れないこととなっている。

クエリ種類には、ほかにパススルークエリ、ユニオンクエリなどもある。

データベースウィンドウの[クエリ]を選択し、[新規作成]をクリックすると次のようなダイアログが表示される。



Access のクエリ作成ユーザーインターフェイスには多少の問題がある。というのは、上記画面をご覧いただくと分かるように、クエリの種類を選択するこの画面には追加クエリ、削除クエリ、更新クエリなどの選択肢がない。よって、初心者はまず最初にどうやってそれらのクエリを作ればよいか、その方法がなかなか分かりづらいのである。

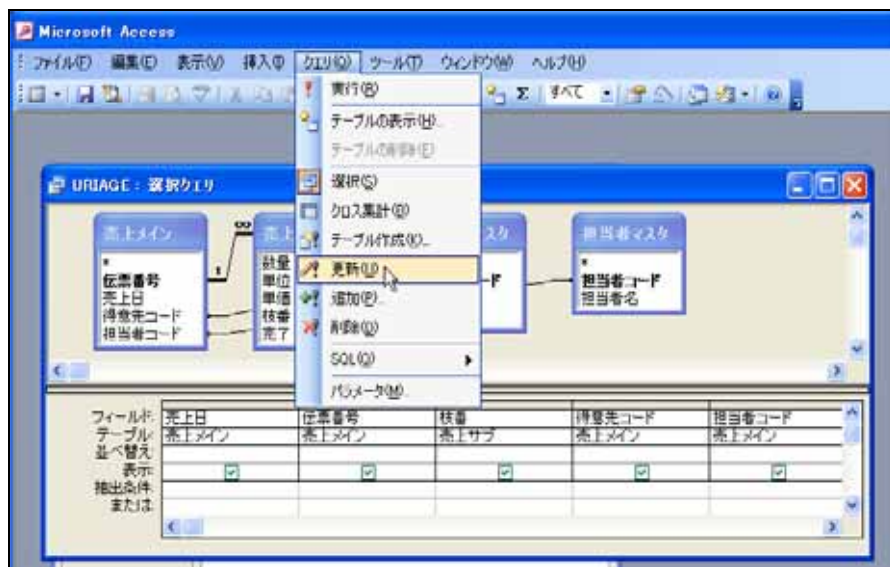
追加クエリ、削除クエリ、更新クエリは次のような 2 つの方法で作成される。

まず一つ目。

クエリビルダのデザイン画面を表示して、次にメニュー[クエリ]を選択して、[更新]、[追加]、[削除]などのクエリ種類を指定する方法。

二つ目は、最初に選択クエリを作成する要領でクエリビルダを操作し、その後でクエリの種類を変更するといった方法である。

次の図は選択クエリから更新クエリに移行しようとするところである。



選択クエリから更新クエリに変更した結果が次の図だ。



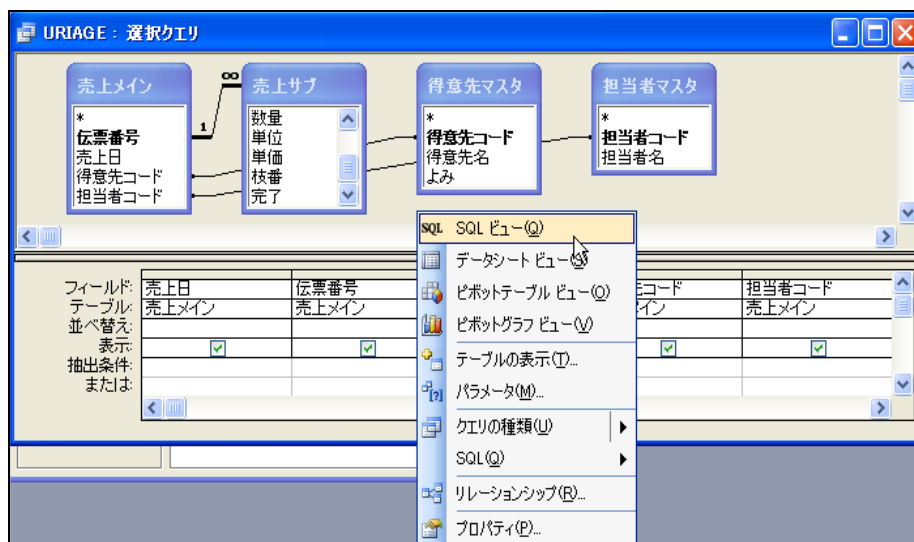
注意深く見なければその違いは分かりにくい。

よく見ると選択クエリ作成画面の下にあった[並べ替え]と[表示]が無くなり、そのかわりに[レコードの更新]という行が追加されていることが分かる。しかし、分かったとしてもそれがどのように作用するのかこのままでは見えてこない。初心者にはクエリを分かりにくくしているのは、おそらくこのあたりが原因しているのではと考えられる。要は、クエリによって何がどうなるのかが分かりづらいのである。

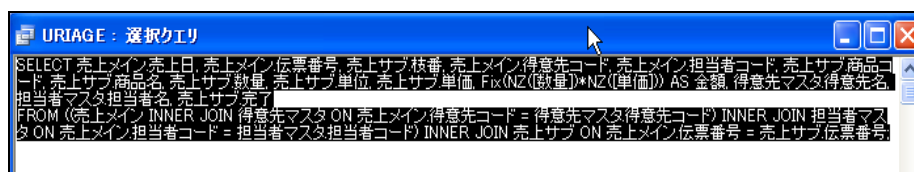
もしも、『クエリビルダは、SQL文を作るための単なるツール』ということを知っていたらどうだろうか。お恥ずかしい話だが、今から十数年ほど前、Accessを勉強し始めたころの私もしくはは知らなかった。だから、Access初心者がそれを知らなかったとしても私にはまったく不思議ではない。

SQLとはStructured Query Language(構造化問い合わせ言語)の略。簡単に言えばデータベースに問い合わせを行う言語だ。このSQLはAccess独自のものではなく、ある種の国際標準規格があって、Accessもそれに準じているのだ。ただし、多少の方言もあるが。

たとえば、選択クエリのクエリビルダでショートカットメニュー[SQLビュー]をクリックしてみよう。



すると、次のようなウィンドウが表示される。この文字列が SQL なのだ。



この SQL 文を良く見ると、先頭に SELECT という文字列があるが、Access のクエリビルダで選択クエリとして作成された SQL 文には必ずこの SELECT という文字が付いている。

同様に、更新クエリは UPDATE、削除クエリは DELETE、追加クエリには INSERT という文字が使われている。

すなわち、SQL の先頭文字列を見ればその SQL がこういった仕事をしようとしているかが分かるということなのである。

実に分かりやすいではないか。しかも SQL の構文はいくつかのパートに分かれてその役割が決められている。

たとえば、テーブル[住所録]の中からフィールド[氏名]に『鈴木』という文字列だけを探し出して表示したいとするクエリは次のようになる。

```
SELECT * FROM 住所録 WHERE 氏名="鈴木"
```

FROM の後ろにはテーブル名が、WHERE の後ろには抽出条件が指定される。
このクエリに並べ替えの指定を行うと次のようなものとなる。

```
SELECT * FROM 住所録 WHERE 氏名="鈴木" ORDER BY 氏名
```

ORDER BY の句が追加されたのが分かる。

また、同じく[住所録]テーブルの中の[氏名]が『浦』さんだけを抽出して、[分類コード]に 10 をセットする更新クエリは次のようになる。

```
UPDATE 住所録 SET 分類コード = 10 WHERE 氏名="浦"
```

この場合は、UPDATE の後ろがテーブル名となり、SET より後ろが代入式となる。
さらに、テーブル[売上メイン]と[売上サブ]の二つのテーブルをフィールド[伝票番号]が一致するもの同士で結合させ、一つのテーブルとしてすべてのフィールドを利用したいときは、次のようなSQL文となる。

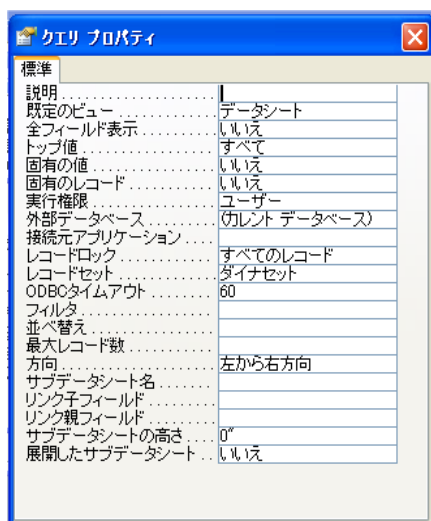
```
SELECT 売上メイン.*, 売上サブ.* FROM 売上メイン INNER JOIN 売上サブ ON 売上メイン.伝票番号 = 売上サブ.伝票番号
```

『INNER JOIN』は二つのテーブルを結合するということを意味し、その後に続く文字列は結びつける条件を指定している。

どうだろう、非常に分かりやすいと感じないだろうか。
その他の構文についてはSQLに関する Access のヘルプや参考書を参照していただくこととして、これらのことをあらかじめ知っておいてからあのクエリビルダを再度眺めてみると、これまでとは違った感想を抱くに違いない。

SQLを勉強する一つの方法として、まずクエリビルダでクエリを作成して、それからショートカットメニュー[SQL ビュー]を開いてSQL文を確認するといった方法も有効だろう。クエリビルダのどこを同変更すると、SQLの何がどうなるのかを確認すると分かりやすい。

また、Access のクエリビルダで作成されたクエリには Access 独自の追加機能も備えられている。ショートカットメニュー[プロパティ]を開いて各項目をチェックされたい。



『はじめにSQLありき』これが本音なのだが、Access 開発者は何とかしてもっと分かりやすい問い合わせ方法を提案したかった。キャラクターベースの小難しい問い合わせ方法を、グラフィカルなユーザーインターフェースで置き換えることで分かりやすさを前面に出そうとした。そしてその結果できあがったのがあのクエリビルダだったのだ。

以上