

スタンドアロン型からクライアントサーバー型へのデータベース移行

キーワード：Access，SQL Server，システム，データベース，アップサイズ

概要

業務効率化のために、大企業だけでなく中小企業においても業務管理システムを導入するケースが増えてきました。大企業においては Oracle などの本格的なクライアントサーバー型データベースシステムの導入が一般的ですが、中小企業においては費用面の制約からスタンドアロン型の簡易データベースソフトである Microsoft Access（以下 Access）を用いて自前のシステムを構築していることが多いのが現状です。

Access は Oracle などのデータベースソフトに比べ比較的扱いやすく、パソコンをある程度使いこなせる人であればデータベースの構築も容易にできるというメリットがあります。しかし、Access は簡易データベースであることから、複数のユーザーの同時アクセスに不向きで、また扱えるデータベースの最大サイズが 2 GB であるなどの制約があります。

このことから、企業の規模や事業が拡大し扱うデータ量が増えてきた場合には本格的なデータベースに移行する必要があります。データベースの移行にはデータを格納しているテーブルのほかに、Access で作成したクエリを移行する必要があります。しかし、この移行を行うには SQL とよばれるデータベース言語に関する知識が必要で、Access のみが見えるという方には難しい作業になります。

このような問題を解決するために Access にはアップサイジングウィザードという機能があります。この機能を用いることで Access から Microsoft 社のデータベースソフト SQL Server へデータベースの移行を自動的に行うことができます。

課題と対策

アップサイジングウィザードはこのように

大変便利な機能ですが、対策を行わずに Access 単体での開発を進めた場合、アップサイジングウィザードの実行段階でエラーが頻発します。エラーの内容は使用文字の制約からクエリ構造まで複数多岐にわたります。私たちはこれら一つ一つの課題を検証し対策方法を導出しました。Access 単体での開発を行う際に以下のエラー対策を実施した上でアップサイジングウィザードを実行することでエラーの発生を回避することができます。

アップデートファイルのインストール
Access2000 をアップサイジングウィザードに対応させるため、「Office Service Release 1」と「SQL Server2000 対応アップデート」をインストールします。それぞれ、Microsoft 社のホームページから入手できます。

クエリの制約

以下のクエリはアップサイズできないため、VBA (Visual Basic for Application) で置き換えます。

- ・入れ子構造のアクションクエリ
- ・パラメータを指定したアクションクエリ
- ・クロス集計クエリ
- ・SQL パススルークエリ
- ・フォーム上の値を指定したクエリ ()
- ・ユニオンクエリ
- ・SQL データ定義言語 (DDL) クエリ

フォーム上の値の指定は、[フォーム上のオブジェクト名]をクエリの抽出条件に記述することで例外的に可能になります。

使用文字の制約

特殊文字 (/ , など) や 1 文字だけの名称をクエリなどの名前に用いないようにします。

VBA 制約

VBA の記述方式には DAO と ADO がありますが、アップサイジングウィザードは DAO に対応していないため ADO で記述するよう

にします。現行の Access を DAO で記述している場合は ADO への変更に多くの時間が必要となります。これから VBA を記述する場合は、必ず ADO で行うようにして下さい。

SQLServer2000 の認証制約

SQLServer2000 の認証モード設定は仕様制約から「混合モード」を選択して下さい。

実施例

前述のエラー対策を施した「中小企業向け在庫移動管理システム」を Access 単体で開発し、アップサイジングウィザードによるクライアントサーバー型システムへの移行を行いました。

移行前の Access (図 1) はデータベース部とインターフェース部(フォーム、レポート)が同じファイルで管理されておりスタンドアロン型のシステムになっています。この Access の拡張子は「mdb」です。

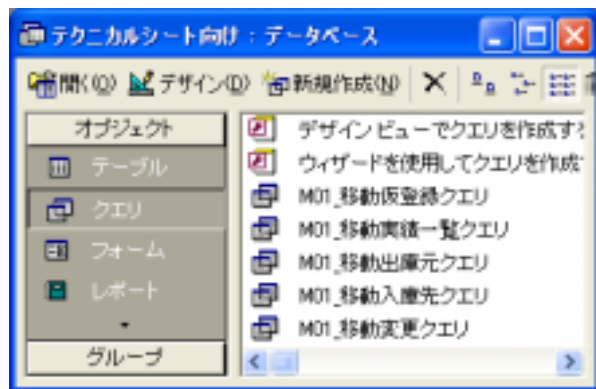


図 1 移行前 Access 管理画面

アップサイジングウィザードを実行することで、クライアント側にクライアントサーバー型の Access (図 2) が自動生成されます。この Access の拡張子は「adp」で、インターフェース部を管理し、データベース部の管理は SQLServer (図 3) に移管しています。SQLServer では Access のクエリにあたる機能をビューとストアドプロシージャで担うため、図 1 のクエリが図 2 でビューとストアドプロシージャに自動変換されていることが確認できます。

アップサイジングウィザードの完了には数分の時間を要しましたが、SQLServer に手作業でテーブル、ビュー、ストアドプロシージャを作成するのに比べて作業時間を大幅に削減することができました。移行後、自動生成された Access ファイルを利用したい端末にコピー & ペーストすることで複数の端末からサーバーの在庫移動管理システムを利用できるようになりました。

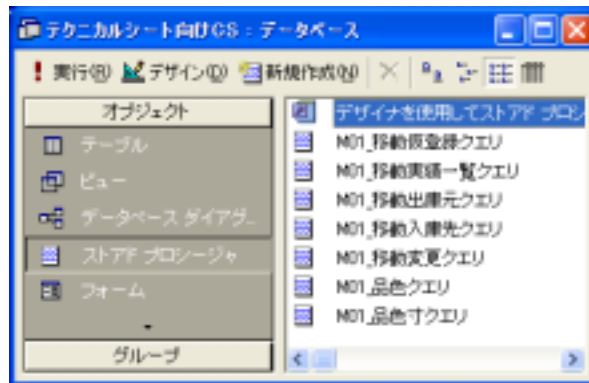


図 2 移行後 Access 管理画面

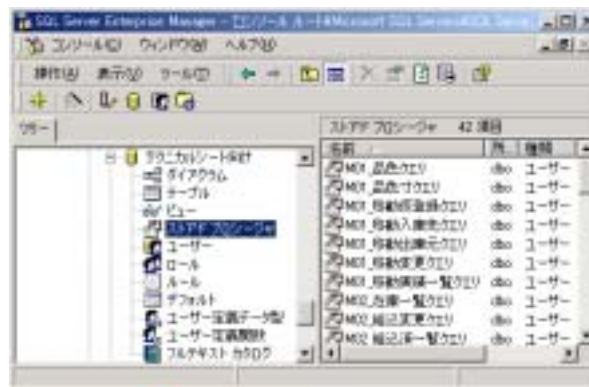


図 3 移行後 SQLServer 管理画面

まとめ

アップサイジングウィザードは、実行のための制約も多くありますが、手軽にクライアントサーバー型のシステムを実現できるメリットがあります。Access 単体から Access + SQLServer への移行が求められているシステムはもちろん、これから Access 単体でのシステム開発を進める場合でもアップサイジングウィザードの制約をあらかじめ考慮した開発を行うことで将来のデータベースの移行が簡素化できます。システム開発の際に検討してみたいかがでしょうか。