

SQL Server と InfoPath, Access による 教務システム OBA 開発の概要

兵庫県立西宮香風高等学校教諭 総務部システム管理課長
松本 吉生

1. OBA 開発とは何か

OBA は Office Business Applications の頭文字をつないだ言葉で、一般のデスクトップユーザーが使い慣れたいわゆる Office アプリケーションを利用し、基幹業務システムなどと連携して業務を行うことをいう。私は本校でデータベースに Microsoft SQL Server を使い、これに対してクライアントコンピュータから InfoPath と Access によってデータにアクセスする形で、勤務校である兵庫県立西宮香風高等学校の教務システムを開発、運用している。このシステムが扱う範囲は、生徒の基本情報、保健情報、講座編成、受講、教科書、出欠、考查点、評価、などで、教務的な処理を中心とする生徒情報の一元管理を実現している。

OBA 開発のメリットは、C # や Visual Basic などのプログラミング言語を使うことなくデータベースに対するクライアントアプリケーションを作ることができることである。ここで「作る」という言葉を使ったが、実際は InfoPath や Access がもともと持っているデータベースに接続する機能を利用するので、「作る」というより「構成する」という言葉が適切かもしれない。OBA 開発で InfoPath や Access に対して必要なことは、通常の機能を使ってフォームやレポートを構成することだけである。

2. 教務システムを自主開発するメリット

本校のような多部制単位制高校はもちろん、多くの単位制高校や総合学科、全日制普通高校でもコンピュータ処理による教務システムを運用している学校は多いだろう。兵庫県では県立須磨東高等学校教諭の仲正博氏による「IK_system」が全日制普通高校の多くで利用されている。この「IK_system」は仲正博氏によるボランティアベースで開発されているが、氏の長年の教職経験の積み重ねの上に、利用者のニーズを不断に反映させる開発スタイルによって

完成度の高いものになっている。「IK_system」を使いこなせば、通常の全日制普通高校における教務処理はうまくいくだろう。

しかし総合学科や単位制高校、本校のような多部制単位制高校では、どうしても定型では収まらない学校独自の処理が必要となる。これらの学校ではシステム開発業者に発注した教務システムを運用している学校も多いと思われるが、自主開発をすることで、学校が求める処理にあったシステムを作ることができる。また教務内規の変更や例外処理などに対して柔軟に対応することができる。本校は自主開発によるシステムを運用して今年度で4年目を迎えるが、新入生の受け入れから卒業まで、教務的に必要な処理を網羅し、帳票や入力フォームの変更、データ構造の変更などに柔軟に対応し、進化し続けるシステムとなっている。また新教育課程の切り替えにも対応できる準備ができています。

3. SQL Server データベースの開発

システムの根幹はデータベースである。Microsoft SQL Server は企業内データベースや Web データベースとしても利用される信頼性の高い業務用データベースだが、開発は非常に簡単である。また開発手法はインターネットや書籍から多くの情報を利用することができる。

SQL Server はデータベースサーバー、いわゆる DBMS であり、Windows Server の上で稼働するサービスである。したがって校内で Windows Server を動かし、そこにインストールして稼働させる。情報は SQL Server の「テーブル」に格納する。必要な情報を正しく管理し、効率よく利用できるようにするためには、テーブル構造に熟考を要する。SQL Server の開発で最も中心になるのは「テーブル」の作成である。

SQL Server はサーバーにインストールするサー

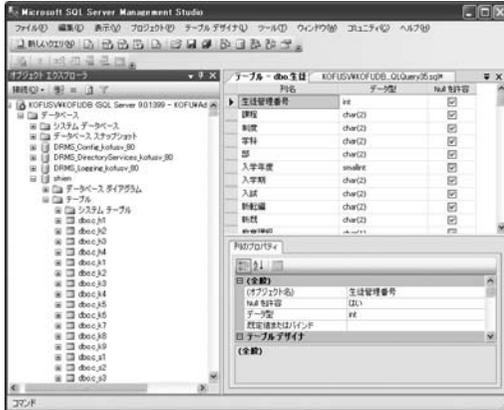


Fig. 1 SQL Server Management Studioによるテーブル作成画面

ビスだが、テーブル作成などの開発はクライアントコンピュータから行うことができる。

SQL Serverの開発は、SQL Server Management Studioによって行う。Fig. 1はManagement Studioによってテーブル作成を行っているところである。Management Studioはクライアントコンピュータにインストールして使え、サーバーの前に行かなくても開発をすることができる。

Management Studioの画面の左側では、SQL Serverに作成したテーブルやビューなどのオブジェクトが一覧できる。右上の部分でテーブル構造を決定し、列名、データ型、Nullの許容などを対話的に編集できる。右下では列のプロパティを詳細に編集できる。テーブルの作成はAccessと似ており、Accessを使ったことがあるなら直感的に理解できるはずだ。

SQL Serverの開発でしなければならないことはテーブル作成以外にもいくつかあるが、まずはテーブル作成と次に述べるActive Directoryとの連携によるアクセス権の設定だけでシステムの基本形ができる。

4. Active Directoryとの連携

テーブルのデータには、データを読み出すことができるSELECT権、新しいデータを挿入することができるINSERT権、既存のデータを書き換えるUPDATE権、データを削除するDELETE権などの権限があり、データの性質に応じて必要なユーザーに必要な権限を割り当てる必要がある。このとき



Fig. 2 Active Directory ユーザーとコンピュータの設定



Fig. 3 受講テーブルの権限設定

Active Directoryのユーザー管理が利用できる。

Fig. 2はActive Directoryのセキュリティグループで、「教務部長」や「単位管理者」など教務処理に応じた権限グループが作成してある。SQL Server側でこれらのセキュリティグループを使い、各テーブルに対する権限を設定することができる。

Fig. 3は生徒の受講情報を保存する「受講」テーブルに設定された権限の一覧であり、この「ユーザーまたはロール」はActive Directoryの権限に基づくものである。このようにActive DirectoryとSQL Serverの権限を連動させることで、ユーザーはコンピュータにログオンした時点で権限情報を取得し、データベースに接続するたびにIDやパスワードを求められることなく、適切な権限によるアクセスが可能となるシングルサインオンが実現できる。

5. InfoPath フォームの実際

InfoPath フォームはSQL Serverに対して、クラ



Fig. 4 InfoPathによるフォームデザイン

クライアントコンピュータからデータの入出力をするために利用する。InfoPathのフォーム開発はInfoPathを起動し「フォームのデザイン」で行う。

Fig. 4はInfoPathによるフォームデザインの画面である。右のペインに「コントロールの挿入」メニューがあり「基本コントロール」として「テキストボックス」や「リストボックス」などが用意されている。入力したいデータの性質に応じて、最もふさわしい入力コントロール部品を選んで左側のビューにドラッグし、入出力のレイアウトを作っていく。

「テキストボックス」は標準の入力コントロールである。フィールド内に自由記述で文字や数字を入力することができる。「ドロップダウンリストボックス」はクリックすることで設定されたリストの項目から選択して入力するもので、テキストボックスに次いで利用頻度の高いコントロールである。リストの項目はドロップダウンリストボックスに列挙するか、データベースの特定のテーブルに接続して利用することもできる。システム全体の整合性を考えると、データベースからリスト項目を選択することが良いだろう。「チェックボックス」や「オプションボタン」を使えばデータ入力が簡単になり、間違いも少なくなる。

Fig. 5は出欠入力用のInfoPathフォームである。出欠入力は各授業の講座ごとに毎時間記録をつけるので、最も良く使われるフォームである。このフォームには、入力ミスを防ぎ効率よく出欠入力を行い、授業担当者と担任が情報を共有するための仕掛けがしてある。

このフォームは授業時間ごとに出席生徒の一覧を

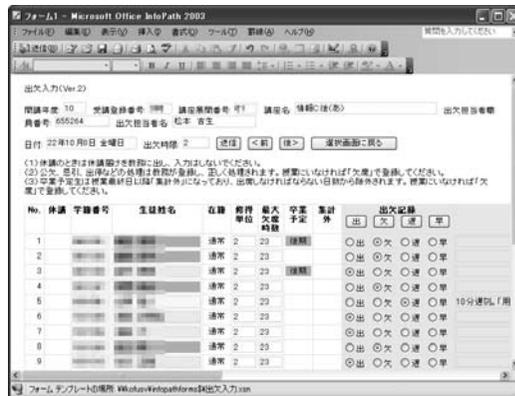


Fig. 5 出欠入力用のInfoPath フォーム

表示し、出欠を記録する形になっている。出欠記録は生徒ごとに出席、欠席、遅刻、早退のうち一つを「オプションボタン」で選択する。「ドロップダウンリストボックス」を使えばリストの表示と選択の2クリックが必要だが、「オプションボタン」なら1クリックで記録できる。生徒の出欠をつけると、氏名が青や赤、黄などの色に塗られるように表示される。これにより行の見間違いなどのミスを減らすことができる。出欠記録には簡単なコメントを入力できる欄がある。ここに生徒の授業中の学習活動を入力すると、担任が読めるようになっている。本校のような完全な単位制高校では、講座がクラスごとではないので学級担任と授業担当者の連携が疎になりがちであり、こうした仕掛けが有効である。

また生徒の在籍状態の表示、修得単位数、最大欠席時間数、卒業予定情報などの表示によって、授業担当者は生徒の状態を確認しながら出欠入力をしていく。本校は前期後期制をとっており、半期講座は週4時間の授業で2単位を認定するが、生徒によっては前期後期と連続して4単位の履修をしている場合がある。このときは修得単位数や履修認定に必要な欠席時間の最大数が異なるので気を付けなければならない。また卒業予定生徒についても留意しなければならない。

これら出欠入力フォームの工夫や必要な生徒情報の表示項目については、授業担当者や学級担任、教務部の出欠担当者などの意見を汲み上げ、少しずつフォームを改訂しながら現在の形になった。運用しながら開発をすすめるOBA開発の利点がここにある。

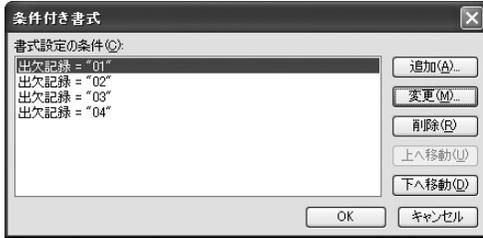


Fig. 6 条件付き書式の設定



Fig. 7 条件付き書式の詳細設定

出欠入力フォームで出欠記録によって生徒の氏名欄の色が変わる設定は「条件付き書式」で設定する。これはテキストボックスのプロパティにあり、Fig. 6のように条件を設定する。

Fig. 7は条件付き書式の詳細設定で、画面上で項目を選択しながら書式設定を行う。「出欠記録」が「01」と入力されたとき、これは「出席」を表すコードだが、ここでは氏名欄のテキストボックスに薄い青の網掛けをする設定を行っている。

6. Access プロジェクトの利用

InfoPath フォームによってデータの入力を効率よく行うことができ、また確認のためのデータ表示も行うことができる。しかし InfoPath フォームに苦手なことは、レイアウトの細かな調整と、複数ページにわたる帳票の印刷である。そこで通知表などの帳票印刷は、Access プロジェクトを使い、レポートとして作成する。

Fig. 8は通知表を印刷する Access プロジェクトである。通知表に必要なデータ項目は、出欠集計や評価などのテーブルから集約し、データベース側にビューを作っておく。Access をプロジェクトとして既存のデータベースに接続し、レポートによって通知表を作成する。

7. まとめ

このように SQL Server と InfoPath、Access を利



Fig. 8 通知表の Access プロジェクト



Fig. 9 Access プロジェクトによる通知表の表示

用した OBA 開発は簡単であり、運用と一体の開発が可能となる。開発に必要なコストやエネルギーは少なく済み、開発者はデータ処理の本質に専念できる。必要なことは、どのようなデータを用意し、どんな処理が必要であり、どういうアウトプットが欲しいのか、という全体像を考えることである。このことは、まさに、教務処理の全体像を考えることに等しい。単位制高校や総合学科などで教務処理に苦労しているなら、システムの独自開発をすべきである。独自開発をすることによって何をどう処理しなければならないかが再認識され、学校の持つノウハウを組み込むことができ、運用とともに使いやすいシステムに成長するのである。

参考 URL

- IK_system Web Site
<http://iksystem.dyndns.org/index.html>
 県立須磨東高等学校 教諭 仲正博氏による