

教育用データベース構築エディタの設計

5K-6

新田雅道

尚美学園短期大学

1. はじめに

ビジネス分野ではワープロ、表計算、データベースが三種の神器といわれ、今日のパソコン環境ではこれらは日常的に活用されている。しかし、システムとしてデータベースを構築するような立場にあるエンジニアにとっては、どうしても標準的なSQLを理解する必要がある。

本エディタはGUIを用いて視覚的にデータベースを構築しながらSQLの構文が学習できるようにしたもので、主に専門学校などでデータベースエンジニアを育成する場合を想定した。

2. エディタ設計のコンセプト

市販のデータベースアプリケーションには、様々なものがある。たとえば、ヴァル研究所のファラオやMicrosoft社のAccess、Lotus社のアプローチなどである。いずれもパソコン用のデータベースアプリケーションとして優れた特徴があり使いやすい。しかし、オフコンや汎用機などで一般的に用いられているデータベースはSQL (Structured Query Language) を使用しており、データベースを操作するということはこのSQLをマスターするということになる。

ユーザがSQLでデータベースを操作するには2通りの方法がある。ひとつは、直接データベース環境に入ってダイレクトコマンドでインタラクティブに操作する方法。もうひとつは、アプリケーションにデータベースコマンドが埋

め込まれていて、定型業務として操作する方法である。後者の場合、SQLを理解しているばかりでなく、親言語も十分に扱えなければならない。

親言語にはCOBOLやPL/I、あるいはC言語が一般的であるが、いきなりSQLを組み込んだデータベースアプリケーションの開発手法の教育には飛躍しえない。教育手順としては、SQLと親言語のそれぞれの教育をしてから、より高度な使い方としてSQLの組み込みを教えるというのが順当なところだろう。

さて、この教育用データベース構築エディタだが、エディタといってもSQLの文法を学習するのがメインの教育用システムである。いわばCAI (Computer Aided Instruction) に近い位置づけとして開発した。市販のSQLエディタには、すべてコマンドを入力するものや、OS/2に組み込まれているSQLエディタのようにGUIを使ったユーザフレンドリなものがあるが、いずれも教育という観点からするとメリットデメリットがある。機能が豊富なため操作が複雑だったり、ユーザフレンドリな反面、複雑な集合演算が指定できないというようなことがあるからである。

これらのことを検討し、このエディタには次のようなコンセプトを持たせた。

- ①. 操作はなるべくシンプルにする。
- ②. コマンドの種類は当面必要なものだけとする。
- ③. GUIとウィンドウをうまく使い分ける。
- ④. 動作がシミュレーションできる。

つまり、コマンド名のスペルミスや表名、項目名の不一致といった「間違わなくてもいい間違い」は発生しないようにし、視覚的に動作状況が確認できるようなシミュレーション機能を持たせることによって、より早くSQLが理解できるように工夫したのである。

3. エディタの開発と機能

前述のようにGUIを活用するため、開発環境はWindowsとし、開発言語はVisual BASICを用いることにした。また、関係データベースの表をウィンドウ単位で表現しようというアイデアもあったので、Windows環境は必須であった。

SQLコマンドは当面必要なものだけが使えるようにした。表1にコマンド一覧を示す。

表1 利用できるコマンド

D D L	CREATE TABLE
	DROP TABLE
	GRANT
D M L	INSERT
	DELETE
	UPDATE
	SELECT

このエディタはスタンドアロン用なので、権限に関するコマンド（GRANT）については、文法チェックのみとした。

図1は本エディタの一画面である。基本はコマンド選択ウィンドウで、ここでコマンドを指定すると、コマンド作成ウィンドウが開く。コマンドを作成する上で必要となる表はウィンド

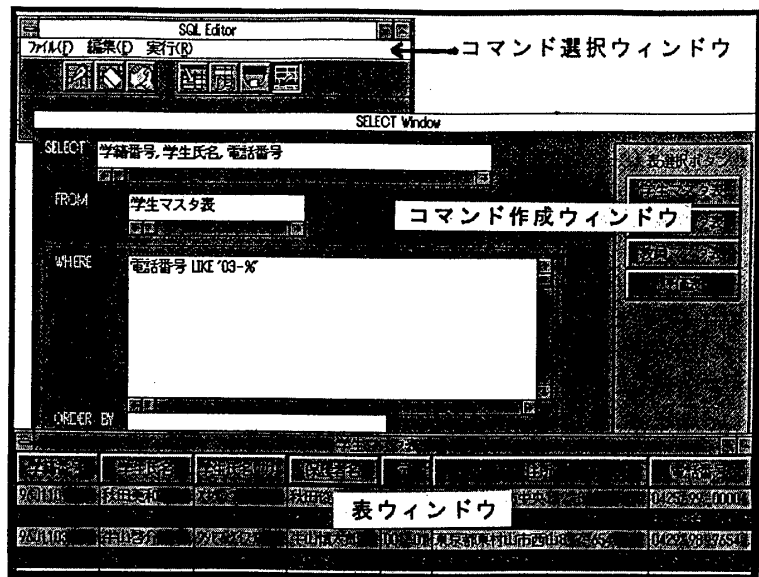


図1 エディタの画面例

ウとして開かれる。この場合ウィンドウタイトルが表名になっており、ウィンドウの中は項目名（列名）と組（タプル）のマトリックスになっている。

図1はSELECTコマンドを作成しているところである。項目名を指定するときは、表ウィンドウの項目名を直接クリックする。こうすることにより、項目名の入力ミスが無くなる。また、選択すると表名の指定欄には自動的に表名が設定されるようになっている。

4. おわりに

本エディタはそれぞれのウィンドウに機能を持たせたという点に意味があると思うが、反面取り扱う表の数が増えると、その分ウィンドウも増えてしまうというデメリットが出てくる。これを少なくする工夫として、コマンド選択ウィンドウとコマンド作成ウィンドウを一つにしたり、表となるウィンドウのフォントを落として小さめの表にするなどがあると思う。

また、集合演算の機能が弱いため、あまり複雑な条件設定はできない。この点についてもバージョンを重ねながら機能アップを図っていきたい。