ＳＱＬ／ＧＲＥＯ：可搬性を考慮したデータ定義機能の構築

5G－3

山岸義徳　喜田智子　安藤隆朗　岩崎浩文
三菱電機（株）　情報システム研究所

１．はじめに

ＳＱＬ／ＧＲＥＯ[1]は、データベースプロセッサＧＲＥＯ[2]を利用することを特長とするリレーショナルデータベース管理システム（RDBMS）である。我々は、ＳＱＬ／ＧＲＥＯの開発において、ＳＱＬ[3]の持つ自己記述性に着目することで、将来のプラットフォームの変更等に対する可搬性を考慮したデータ定義機能の構築を行った。以下では、この設計アプローチ、実現方式について報告する。

２．設計アプローチ

ＳＱＬにおけるデータ定義処理は、スキーマ、表、権限等の定義情報をカタログに登録することに等しい。ＳＱＬではこれら定義情報をカタログによって一括管理する。カタログは、図-１に示すように複数の表から構成され、相互に関係しあっている。これらカタログ構成表に対しては、一般的な表と同様、ＳＱＬ文によるデータ操作（SELECT、INSERT、UPDATE、DELETE）を行うことができる。このことは、データ定義処理がＳＱＬ文によるデータ操作によって記述できることを示している。例えば、実表生成（CREATE TABLE文）は、表名や表タイプ等の表情報、列名や列位置等の列情報、権限種類や権限受領者等の権限情報を対応したカタログ構成表であるTABLES、COLUMNS、TABLE_PRIVILEGESにINSERT文で追加する操作、SCHEMATAに対してスキーマ情報をUPDATE文で更新する操作等を行うことにより実現できる。逆に、実表削除（DROP TABLE 文）は、実表生成で各カタログ構成表に対して追加した情報をDELETE文で削除する操作、SCHEMATAに対してUPDATE文により更新する操作等を行うことにより実現できる。

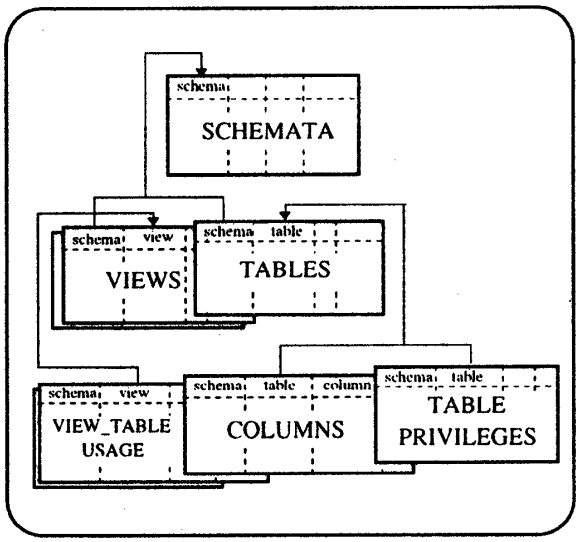

図-１　カタログ構成表

ＳＱＬの自己記述性を利用し、ＳＱＬ文でデータ定義処理を記述することにより、以下の利点が得られる。

（１）可搬性：ＳＱＬはシステムに依存しないデータベース言語であるため、プラットフォームの変更等に対して可搬性が高い。

（２）保守性：データ定義機能の理解や機能追加、修正が容易である。

（３）生産性：データ定義機能のステップ数を少なくできる。また、ＳＱＬコンパイラ[4]がデータ操作を行うＳＱＬ文の最適化を行うので、データ定義機能製作者の負荷を減少させることができる。

以上の点から、データ定義機能の構築においては、ＳＱＬ文を使ってカタログへのデータ操作を行う設計アプローチを採ることとした。

A Highly Portable Implementation of SQL Data Definition Function
Yoshinori YAMAGISHI, Tomoko KITA,
Takaaki ANDO, Hirofumi IWASAKI
Mitsubishi Electric Co.
5-1-1 Ofuna,Kamakura,Kanagawa 247, Japan

3．実現方式

本設計アプローチに従い、ＳＱＬにおけるデータ定義機能を実現する手順を図-２に示す。

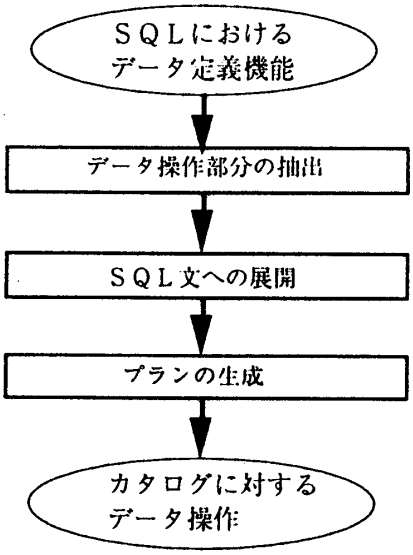

図-２　実現手順

（１）　データ操作部分の抽出

ＳＱＬの各データ定義文に対して、その処理内容を分析し、カタログに対するデータ操作部分を調査する。すなわち、どのカタログ構成表に対して、どのようなデータ操作が必要かを調べる。

（２）　ＳＱＬ文への展開

抽出したデータ操作部分をＳＱＬで記述する。実表生成（CREATE TABLE文）におけるカタログへのデータ操作部分をＳＱＬで記述した例を図-３に示す。

（３）　プランの生成

ＳＱＬ文をＳＱＬコンパイラを用いて翻訳し、実行に最適なプラン[4]に変換する。プランは、ＳＱＬ文の実行オブジェクトであり、ＳＱＬ中核部によって解釈され実行される。

4．まとめ

我々はデータ定義機能の構築に際して、ＳＱＬの自己記述性に着目し、カタログに対する操作をＳＱＬ自身で記述することで、可搬性の高いデータ定義機能を実現した。

ＳＱＬでの記述はプログラムの理解がしやすく、

・表情報の追加
INSERT INTO TABLES（スキーマ名，表名，
表タイプ，列数，・・・）

・列情報の追加
INSERT INTO COLUMNS（列名，位置，長さ，
データタイプ，・・・）

・権限情報の追加
INSERT INTO TABLE_PRIVILEGES（権限付与者，
権限受領者，スキーマ名，表名，権限，・・・）

・スキーマ所有の表数情報の更新
UPDATE SCHEMATA SET NUM_OF_TABLE=
NUM_OF_TABLE+1　WHERE
SCHEMA_NAME=スキーマ名

⋮

図-３　表生成におけるＳＱＬ文への展開

開発量の面においては、ＳＱＬを用いない、低レベルなインタフェースで記述したデータ操作による方法と比較して、プログラムのステップ数で約３０％の削減が得られ、その有効性を確認した。また、プログラムの機能追加、修正が容易で保守性が高いことも実際にデータ定義機能を構築する過程において確認することができた。

参考文献

[1]佐藤重雄、道下　学、岩崎浩文、安藤隆朗：SQL/GREO:データベースプロセッサＧＲＥＯを用いた並列処理方式、情報処理学会　第５０回全国大会講演論文集（１９９５）

[2]伏見信也、武田保孝、岩崎浩文、小宮富士夫、中込　宏：データベースプロセッサＧＲＥＯ、情報処理、Vol.33、No. 12、pp. 1416- 1423（１９９２）

[3]ISO/ANSI,「Database Language SQL2 and SQL3」X3H2-90-001（１９９０）

[4]伏見信也、岩崎浩文、安藤隆朗、佐藤重雄、小宮富士夫、樋口雅宏：高速ハードウェアソータを用いたＳＱＬシステムの実現、信学技報、Vol.91、No.259、ＤＥ 91-38、pp.1-8（１９９１）