

SQL参照制約のデータ操作言語変換方式の提案

3 G - 1

河村信男*, 土田正士*, 山平耕作**

*日立製作所システム開発研究所, **日立製作所ソフトウェア開発本部

1. はじめに

参照制約は、UAPで行っていたデータ間の整合性チェックの処理をDBMSが行うので、システム開発・保守の生産性が高まり、システムの性能が向上する機能である。

参照制約の実装に際しては、いかにDBMSが効率よく実行するかが課題となる。一般的に、UAPで記述されたSQLは、DBMSが内部処理手続きとして生成したオブジェクトを利用することによって、実行時のSQLのコンパイル処理を削減している。このことに着目し、参照制約の処理手続きも実質的に同じ処理手続きとして表現すれば、最適なアクセス手順の選択や処理手続きの管理の簡素化を図ることができる。

本稿では、DBMSが参照制約を実行する場合に、参照制約によって行うべき処理をSQLのデータ操作言語に変換して実行する方式を提案する。

2. 参照制約用データ操作言語変換原理

2.1 副問い合わせ変換方式

参照制約の定義時に参照動作を指定しない場合、参照制約によって行うチェック処理を次のように副問い合わせとして表現することができる。

■主キー値の更新、削除

図1に示すように、NOT EXISTS述語を使用した副問い合わせに変換することができる。変換した副問い合わせは、更新要求のSQLの探索条件にANDで連結する。

```
NOT EXISTS(SELECT * FROM 参照表
WHERE 外部キー構成列1=主キー更新対象値1
AND 外部キー構成列2=主キー更新対象値2
.
.
AND 外部キー構成列n=主キー更新対象値n)
```

図1 主キー値更新時の整合性チェック用データ操作言語

■外部キー値の更新、挿入

図2に示すように、EXISTS述語を使用した副問い合わせに変換することができる。変換した副問い合わせは、更新要求のSQLの探索条件にANDで連結する。ただし、外部キー更新値がNULLの場合はチェック不要であるので、OR述語により、外部キー更新値があらかじめNULLであるか否かをチェックする必要がある。

```
(外部キー更新対象値1 IS NULL OR
外部キー更新対象値2 IS NULL OR . . .)
OR EXISTS(SELECT * FROM 被参照表
WHERE 主キー構成列1=外部キー更新対象値1
AND 主キー構成列2=外部キー更新対象値2
.
.
AND 主キー構成列n=外部キー更新対象値n)
```

図2 外部キー値更新時の整合性チェック用データ操作言語

2.2 データ操作言語変換方式

参照制約では、ある表の主キーを参照する外部キーの定義時に、参照動作を指定することが可能であり、参照動作が指定される場合、主キー値の更新に伴って参照する外部キー値の値を指定された規則に従って更新する必要がある。ただし、RESTRICTが指定される場合は、2.1節と同等である。参照動作が指定される場合、参照動作によって行う処理を意味的に同等なデータ操作言語として表現することができる。

(1)更新規則

更新規則が指定される場合、指定されるアクション(CASCADE, SET NULL, SET DEFAULT)によって主キー値と同じ値をもつ外部キー値を更新する。

図3に示すように、UPDATE文を使用した参照動作の波及を行うデータ操作言語に変換できる。

```
UPDATE 参照表
SET 外部キー構成列1=更新値1
,外部キー構成列2=更新値2
.
.
,外部キー構成列n=更新値n
WHERE 外部キー構成列1=更新前値1
AND 外部キー構成列2=更新前値2
.
.
AND 外部キー構成列n=更新前値n
```

図3 主キー値更新時の更新動作波及用データ操作言語

(2)削除規則

削除規則が指定される場合、指定されるアクションがCASCADEの場合のみ、削除対象となった行の主キー値と同じ値をもつ外部キー値を持つ行を削除する。SET NULL, SET DEFAULTの場合は、更新規則と同じでよい。

図4に示すように、DELETE文を使用した参照動作の波及を行うデータ操作言語に変換することができる。

```
DELETE FROM 参照表
WHERE 外部キー構成列1=削除対象値1
AND 外部キー構成列2=削除対象値2
.
.
AND 外部キー構成列n=削除対象値n
```

図4 主キー値更新時の更新動作波及用データ操作言語

3. 参照制約実現方式

3.1 副問い合わせ変換方式

副問い合わせ変換方式の場合、2.1節で示したように要求された更新SQLに対して、SQLの解析時に参照制約用副問い合わせをANDで連結する。

図5に、本方式の実現方式を概略する。

【特徴】

- ・参照制約の処理を副問い合わせで表現できるので、最適化処理において、アクセス手順の最適化が図れる。
- ➡あらかじめ更新対象となる値が判明している場合、参照制約の処理を行う副問い合わせを先に評価するようなアクセス手順を選択できる。
- ・参照制約の処理を先に行うことができる場合、更新を行う前に制約違反を検出できるので制約違反が発生しても、ロールバック処理に要する時間を削減できる。

3.2 データ操作言語変換方式

データ操作言語変換方式の場合、2.2節で示したように要求されたSQLの種別に応じて対応する規則に準じたデータ操作言語を呼び出して実行する。この場合、参照制約の定義時に各規則に対応したデータ操作言語を生成し、ディクショナリ等に格納しておく。

図6に、参照制約用データ操作言語の生成方式について概観する。

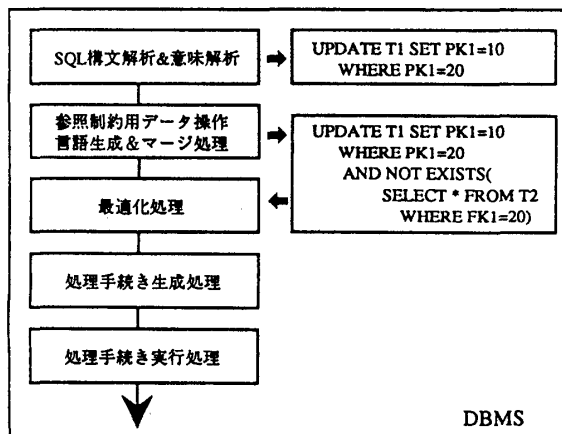


図5 副問い合わせ変換方式の場合の実現方式の概略

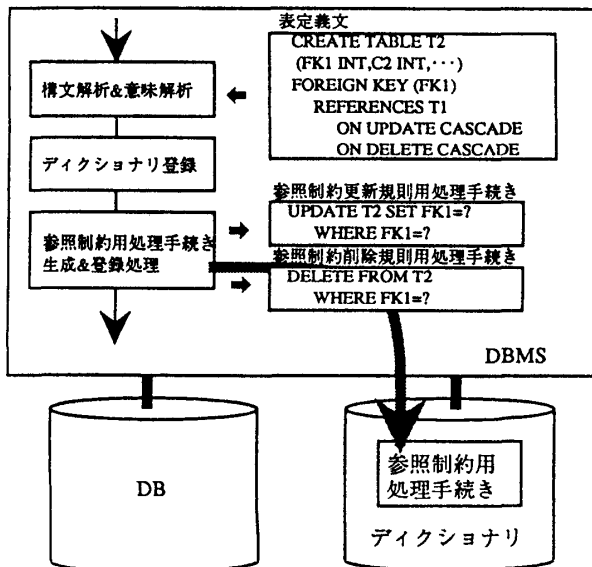


図6 参照制約用データ操作言語の生成方式

図7に、本方式による参照制約の実現方式を概略する。

- ①主キーを更新するSQLの処理手続きの実行
- ②主キー値の更新操作
- ③主キーを参照する表T2の外部キーを更新するための処理手続きをディクショナリより読み出し、実行

【特徴】

- ・参照制約用に生成したデータ操作言語は、複数のユーザによって共用することができる。
- ・参照制約の定義変更に伴うアプリケーションプログラムの再コンパイルの必要性がない。

4. おわりに

参照制約を実現するために、次の2つの方式を提案した。

- (1)参照動作が指定されない場合、制約用のデータ操作言語を副問い合わせとして表現する方式。
- (2)参照動作が指定される場合、制約用のデータ操作言語をあらかじめ生成しておき、SQLの実行時に制約用のデータ操作言語を呼び出し、実行する方式。

これにより、参照制約の効率的な実現が可能となった。

【参考文献】

- 1.JIS:X3005-1990 データベース言語SQL
- 2.「リレーショナルを越えて」,日経コンピュータ,1991.6.3

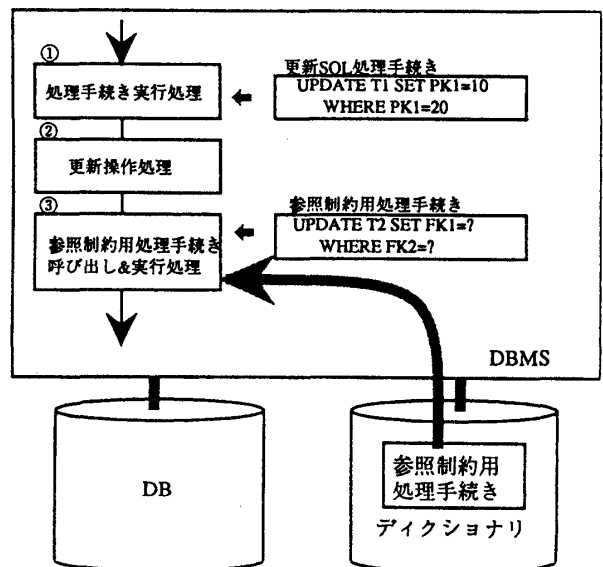


図7 データ操作言語変換方式の場合の実現方式の概略