

【テーブルとビューを極める（その2）】

ビューの定義と利用

下図（ビューの利用）にビューの定義から更新可能なビューまでの流れを簡単に説明します。

ビューの定義

SQL文のDDLの1つである「**CREATE VIEW ビュー名 AS SELECT**・・・」により、AS以下のSELECT文がビュー名で保存される。つまり「**CREATE TABLE**」によるテーブル（実表）は、実表自体が記憶装置に保存されるが、「**CREATE VIEW**」による**ビュー（仮想表又は導出表）は、SELECT文による定義だけが保存される。**

ただし、頻繁に**ビュー**にアクセスする場合には、**体現ビュー**として、実表のように記憶装置に保存することもできる。



ビューの参照

SQL文のDMLである「**SELECT * FROM ビュー名**」を実行するたびに、最新の実表が反映された仮想表が作成される。



ビューを介しての実表の更新等

SQL文のDMLである**INSERT文**、**UPDATE文**及び**DELETE文**にテーブル名ではなくビュー名を指定することにより、ビューが呼び出されて、ビューにデータが追加、更新及び削除（更新等）することができる。しかし、ビューに更新等を行っても、仮想表自体は保存されないため、ビューの元となるテーブルが更新等されることで、次回にビューを参照した時に更新等が反映されることになる。

そう考えると、ビューを介して、元のテーブルのデータを改ざんできることになるが、その点については、ビューの更新等の権限で規制するしかない。詳細は別の機会で説明する。

また、ビューの更新等は、元のテーブルの更新等ができることが条件のため、ビューが「**GROUP BY句**」とその仲間たちである「**HAVING句**」、「**DISTINCT句**」、**計算例及び集約関数等を使用**している場合は、ビューはグループ化されたビューになってしまうため、もはやテーブルの行を特定することはできなくなり、テーブルの更新等はできなくなる。

上記の様な条件をクリアすることで、更新等が実行できるビューを『**更新可能なビュー**』と呼ぶ。

ストアドプロシージャ

ビュー名でビューを呼び出すことを知った時、ストアドプロシージャ（ストアド）が思い浮かびました。実際には異なるものですが、ついでにストアドについてもまとめておきます。

- ①SQLを用いた一連の処理をDBで行うプログラムである。
- ②SQL文のDDLの1つである「**CREATE PROCEDURE プロシージャ名**…」でストアドを保存（定義）して「**CALL プロシージャ名**…」で呼び出します。この辺りが、VIEWと似ていると感じた部分です。
- ③クライアントはプロシージャ名だけを送信すれば、DBがストアドを実行してくれるため、通信量の削減にもなります。

本資料は正確性に欠く部分及び誤字脱字等も多いと思います。そのため、本資料に起因した損害等については、管理人として責任を負いかねますので御了承ください。

ビューを生成する権限

ビューを生成する場合、SQLのDDLの1つである「CREATE VIEW ビュー名 AS SELECT～」を使用しますが、AS以下のSELECT文を実行するためには、FROM句で指定するビューの元となる全てのテーブルに対するSELECT権限が必要となります。そのため、テーブルの所有者等は、ビューの作成者に対して、以下のGRANT文で権限を付与します。

```
GRANT SELECT ON ビューの元となる全てのテーブル TO ビューの作成者
```

ビューを使用する権限

上記のGRANT文により、ビューの作成を許可された者は、ビューを作成し、「SELECT * FROM ビュー名」でビューを見ることが出来ます。しかし、ビューの更新等（INSERT文、UPDATE文、DELETE文の使用）はできません。なぜなら、先の投稿で説明した通り、ビューの更新等により元のテーブルも更新等が実行されてしまうため、ビューの作成者は、元の表に対するSELECT権限だけでなく、更新等（INSERT、UPDATE、DELETE）の権限も必要だからです。

そこで、ビューの作成に加えて、更新等又は全ての行為（ALL PRIVILEGES）を行えるように、テーブルの所有者等は、ビューの作成者に対して、以下のGRANT文で権限を付与します。

```
GRANT SELECT、INSERT、UPDATE、DELETE ON ビューの元となる全てのテーブル TO ビューの作成者  
又は  
GRANT ALL PRIVILEGES ON ビューの元となる全てのテーブル TO ビューの作成者
```

ビューの所有者以外の者がビューを使用する権限

ビューの所有者がビューの所有者以外の者（第3者）に自由に権限を与えることができると、元のテーブルの所有者が知らないまま、元のテーブルの更新等が行われる可能性があります。

そこで、元のテーブルの所有者は、先ほどの2つのGRANT文に対して、ひと手間加えることで、ビューの所有者のふるまいを制限することができます。

①ビューの所有者が、第3者に対して、ビューを見る権限だけを与える場合のGRANT文

```
GRANT SELECT ON ビューの元となる全てのテーブル TO ビューの作成者 WITH GRANT OPTION
```

②ビューの所有者が、第3者に対して、ビューの更新等又は全ての権限を与える場合のGRANT文

```
GRANT SELECT、INSERT、UPDATE、DELETE ON ビューの元となる全てのテーブル TO ビューの作成者  
WITH GRANT OPTION  
又は  
GRANT ALL PRIVILEGES ON ビューの元となる全てのテーブル TO ビューの作成者 WITH GRANT OPTION
```