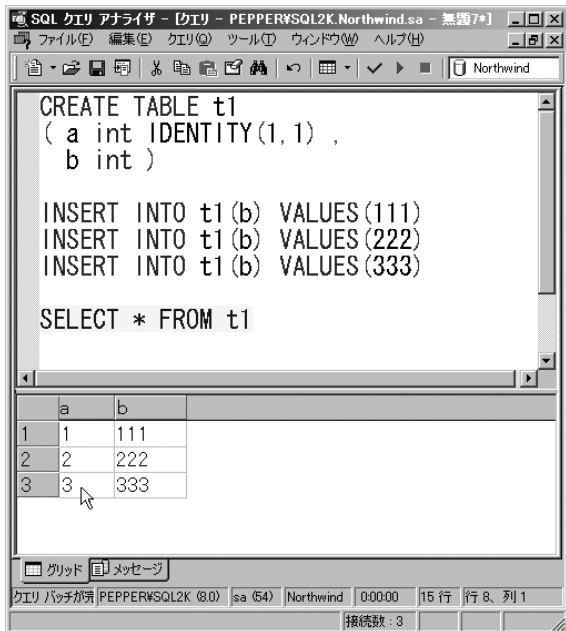




図1：IDENTITYプロパティで自動追加された値の確認。3件データを追加しているので、「1、2、3」と連番が振られている



「SCOPE\_IDENTITY」または「@@IDENTITY」というシステム関数で取得できる (図2)。SCOPE\_IDENTITYは、SQL Server 2000からの新機能で、@@IDENTITYの欠点を補うために追加された。この欠点は、@@IDENTITYでは、IDENTITYプロパティを設定したテーブルにトリガがある場合に、トリガ内で生成されたIDENTITY値を取得してしまうというものである。たとえば、t1テーブルに次のトリガが作成されていたとする。

```

CREATE TRIGGER tri1
ON t1 FOR INSERT
AS
INSERT INTO t2(y) VALUES(777)
    
```

このトリガは、t1テーブルのINSERT時に起動し、「t2」という別のテーブルにデータを挿入している。

また、t2テーブルにはIDENTITYプロパティが設定された列があるとする。このとき、t1テーブルにデータをINSERTすると、異なる結果になる (図3)。

```

INSERT INTO t1(b) VALUES(555)
SELECT @@IDENTITY, SCOPE_IDENTITY()
    
```

図2：自動追加されたIDENTITY値はSCOPE\_IDENTITY関数で取得できる

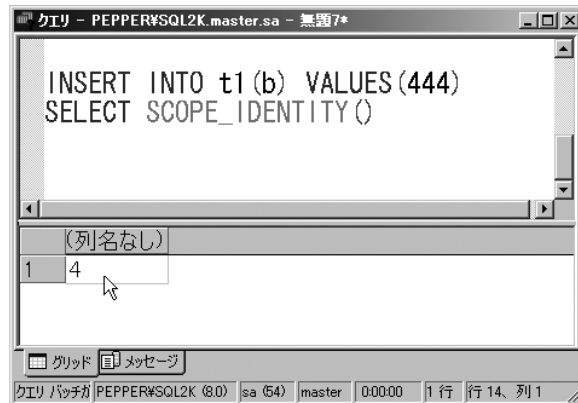


図3：@@IDENTITYはトリガ内で生成されたIDENTITY値を取得してしまう



@@IDENTITYは、直前のステートメントが生成したIDENTITY値ではなく、トリガ内で生成されたIDENTITY値 (t2テーブルに挿入された値) を取得してしまうのである。したがって、トリガ内でIDENTITY値を挿入するようなテーブルの場合は、@@IDENTITYを使わないようにし、SCOPE\_IDENTITYを利用しなければならない。



## ADOからIDENTITY値の取得

ADOからIDENTITY値を取得するには、リスト1のように記述する。リスト1では、SCOPE\_IDENTITYを利用しているが、ひとつのCommandTextに複数のステートメントを入れているのがポイントである。