DB2 V4 In-Line Views

By Craig S. Mullins

A new feature of DB2 Version 4, in-line views, may greatly impact the number of actual views that your shop may need. In-line views, sometimes referred to as nested tables, enable a programmer to specify a SELECT statement in the FROM clause of another SELECT statement. Any table expression can be written in the FROM clause, an area previously reserved for table, view, synonym, and alias names.

Why would anyone want to use an in-line view, instead of simply creating a view prior to issuing the SELECT? Well, the first potential benefit is that an in-line view expression can be easier to understand. Instead of attempting to query the DB2 Catalog to extract the SQL definition of a view, the SQL is clearly displayed in the SELECT statement. Secondly, in-line views provide direct SQL support for certain complex queries which required a view prior to DB2 V4. For example,

consider the typical situation where detail and aggregated information from a single table must be returned from a single query. A prime example is reporting on column length information from the DB2 Catalog. The request is, for each table provide all column details, and on each row, also report the maximum, minimum, and average column lengths for that table.

The pre DB2 V4 solution is to create a view. Consider the COL_LENGTH view based on SYSIBM.SYSCOLUMNS shown below:

```
CREATE VIEW  COL_LENGTH
     (TABLE_NAME, MAX_LENGTH,
     MIN_LENGTH, AVG_LENGTH)
AS   SELECT TBNAME, MAX(LENGTH),
            MIN(LENGTH), AVG(LENGTH)
     FROM   SYSIBM.SYSCOLUMNS
GROUP BY TBNAME;
```

After the view is created, the following SELECT statement can be issued joining the view to the base table, thereby providing both detail and aggregate information on each report row:

```
SELECT TBNAME, NAME, COLNO, LENGTH,
       MAX_LENGTH, MIN_LENGTH, AVG_LENGTH
FROM   SYSIBM.SYSCOLUMNS C,
       authid.COL_LENGTH V
WHERE  C.TBNAME = V.TABLE_NAME
ORDER BY 1, 3;
```

The solution using in-line views in DB2 V4 is to skip the view creation step and simply execute the following SQL statement:

```
SELECT TBNAME, NAME, COLNO, LENGTH,
       MAX_LENGTH, MIN_LENGTH, AVG_LENGTH
FROM   SYSIBM.SYSCOLUMNS C,
       (SELECT     TBNAME AS TABLE_NAME,
                   MAX(LENGTH) AS MAX_LENGTH,
                   MIN(LENGTH) AS MIN_LENGTH,
                   AVG(LENGTH) AS AVG_LENGTH
        FROM       SYSIBM.SYSCOLUMNS
        GROUP BY   TABLE_NAME) AS V
WHERE  C.TBNAME = V.TABLE_NAME
ORDER BY 1, 3;
```

The same result is returned in a single SQL statement, but without using a view.

In-line view expressions must be enclosed in parentheses and must contain a correlation name. The correlation name for the in-line view expression can not be referred to elsewhere in the same FROM clause. It can be used outside the FROM clause, just like any other table or view name, as the qualifier of a column name.

Synopsis

In-line views can be used by shops running DB2 Version 4 to reduce the number of views that need to be created and maintained. Of course, the new SQL syntax may take some time to get used to, and appears to be more complex than the old syntax, but these (perceived) drawbacks should outweigh the potential cost of object maintenance required of views.

```
//STEP1    EXEC DSNUPROC,UID='DBAPCSM.CPY2CPYT',
//         UTPROC='',
//         SYSTEM='V71A',DB2LEV=DB2A
//SYSIN    DD *
//COPY2    DD DSN=COPY002F.IFDY01,UNIT=SYSDA,VOL=SER=CPY02I,
//         SPACE=(CYL,(15,1)),DISP=(NEW,CATLG,CATLG)
//SYSIN    DD *

   COPYTOCOPY TABLESPACE DSN8D71A.DSN8S71E COPYDDN(,COPY2)
```

```
/*
```

From DB2 Update (Xephon) November 2001.

Home.