Design a DB2 for z/OS table to hold just one row

Sreeharsha Naik (harsha_pace@yahoo.com)
DB2 application DBA
Mphasis

DB2® tables are meant to hold enormous amounts of data for application testing, development, and production purposes. But at times, the need arises to design the table to hold just one row at a time dynamically. This tutorial introduces the basic concepts of having a DB2 for z/OS® table hold only one row.

Why do we need a table with just one row?

We may have an application flow that may comprise a control table (input table) and the job/process flow (in the form of Job Control Language (JCL)) accesses the table for input from the concerned user (application developer). We might need a mechanism to lock the table out if the user inserts a row into the control table. No other user should be able to insert a row into the table until the process flow (job) is released. The job upon successful completion clears the entry in the control table.

How to achieve this concept

• **Step 1**: Assuming a simple CONTROL table presence in your subsystem.
• **Step 2**: Alter the table and add a column with the name LOCK, and add a check constraint with the name FLAG using the below syntax:
  
  ```sql
  ALTER TABLE CONTROL_TABLE
  ADD COLUMN LOCK CHAR(3) NOT NULL;
  ALTER TABLE CONTROL_TABLE
  ADD CONSTRAINT FLAG CHECK (LOCK IN ('YES'));
  ```

• **Step 3**: Create a Primary index on the column LOCK.

Designing the control table

**Step 1**: Let's assume we have a table where users input data using SQL processor using file input (SPUFI) or any other tool. We now need to restrict other users from inserting a DB2 row if there is a row present in the table. Below, we see a table with columns PROGRAM, CREATOR, ACCESS, and EMAIL. The column names can vary based on your control table requirement.
Step 2: By introducing another column named LOCK with data type CHARACTER, length 3, and NOT NULL by establishing a CHECK constraint on the LOCK column, we ensure that the data being passed to this column is YES. The ALTER commands ensure that no other value will be allowed to be inserted into the LOCK column. Notice how the value of YES can be repeated. Any other forced insert with a value other than YES will result in SQL error code -545.

Step 3: Creating a primary index on the LOCK column ensures that only one value can be retained in the table. That is the value YES, as indicated below.
Advantages

- Eradicates the need of complex logic(s) for such scenarios
- Easily modifiable as per requirements and needs of the scenario
- Low usage of system resources since the table is designed to hold just one row
- Harnesses the simple and efficient features of DB2 concepts

Conclusion

DB2 for z/OS tables are known to hold enormous data. Using simple concepts and restrictions on the tables, they can be made to hold just one row at a given point in time. Introducing a new control column will not hamper the application or process flow design logic. This clearly shows and demonstrates the flexibility, availability, and scalability of the product with such controlled inserts.
Resources

- The Information Management area on developerWorks provides resources for architects, developers, and engineers.
- Stay current with developer technical events and webcasts focused on a variety of IBM products and IT industry topics.
- Follow developerWorks on Twitter.
- Watch developerWorks demos ranging from product installation and setup demos for beginners, to advanced functionality for experienced developers.
- Get involved in the developerWorks Community. Connect with other developerWorks users while you explore developer-driven blogs, forums, groups, and wikis.
About the author

Sreeharsha Naik

Sreeharsha Naik started his DB2 for z/OS career at MphasiS, an HP company. He is currently working as a DB2 application DBA for a manufacturing and telecommunications firm. He has extensive knowledge and experience in DB2 application design, development, and performance tuning. He constantly seeks new challenges and opportunities on DB2 for System z and is very keen on sharing his DB2 knowledge with the technical community. He is an IBM certified database associate (DB2 9 Fundamentals) and the POC for the Chennai DB2 Regional Users Group of IDUG. You can check out his blog at www.db2champ.com.

© Copyright IBM Corporation 2015
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)