



Craig S. Mullins

[Return to Home Page](#)

October/November 2003



zData Perspectives

by Craig S. Mullins

DB2 V8: Online Schema Evolution

Making changes to database structures can be one of the biggest causes of database downtime. This is so because many of the most common changes require DBAs to create convoluted scripts that capture DDL from the system catalog, unload data, DROP objects, and then re-CREATE the objects with the desired changes. Such changes can be tedious and error-prone unless you are using a software tool to manage the process.

Of course, not all database changes require such draconian measures. Some changes can be made using a simple ALTER to modify specific characteristics of a DB2 object. And database changes are more flexible with DB2 V8 due to online schema evolution. Online schema evolution begins the process of making more types of database change possible without requiring objects to be dropped and re-created. IBM also refers to this feature as simple schema evolution, online schema changes, or just schema changes.

Online schema evolution, over time, will allow us to alter DB2 database structures with minimal or no outages. Of course, this is the long-term goal. We are in the first phases of schema evolution with DB2 V8 - and remember, "evolution" is a very lengthy process.

Of course, DB2 has offered some degree of online change management even prior to V8. For example, adding a column to the end of a table or renaming a table are both simple to accomplish. But online schema evolution introduces new capabilities for managing database changes. What exactly can be changed with DB2 Version 8?

- You can extend the length of a column to a greater size (but not to a smaller length).
- You can switch the data type of a column within character data types (CHAR, VARCHAR); within numeric data types (SMALLINT, INTEGER, FLOAT, REAL, FLOAT, DOUBLE, DECIMAL); and within graphic data types (GRAPHIC, VARGRAPHIC). But you cannot change character to numeric or graphic; numeric to character or graphic; or graphic to numeric or character.
- The previous data type changes are permitted even for columns that are part of an index or referenced within a view.
- You can alter identity column characteristics.
- You can add a column to an index.
- You can change the clustering index for a table.
- You can make many changes to partitioned table spaces and indexes that were previously not allowed. For example, you can drop the partitioning index, create a table without a partitioning index, add a partition to the end of a

table to extend the limit key value, rotate partitions, and even automatically re-balance partitions.

- You can better support indexes built on VARCHAR columns. Prior to V8 all indexes on variable columns were padded to their maximum size in the index. Now you can CREATE or ALTER an index to specify non-padded variable keys.
- You can better support utility processing for database objects in utility-pending states (REORG pending, RECOVER pending, REBUILD pending).

But there is no such thing as a free lunch. When making online schema changes the modification is made to the system catalog details right away, but not to the underlying data. Every time the data is accessed DB2 will have to transform it from its old type and length to its new type and length. This adds overhead to queries and will cause performance to degrade.

Let's consider a simple example. You can use the SET DATATYPE clause of the ALTER TABLE statement to

change the data type and length of a column. For example

```
ALTER TABLE DSN8810.EMP  
  ALTER COLUMN EDLEVEL  
  SET DATATYPE DECIMAL(7,0);
```

After the ALTER runs successfully, DB2 creates a new “version” of the table space. The definition of the data type is stored in the DB2 Catalog and immediately applies to the data. (Up to 256 concurrent versions of a table space and up to 16 concurrent versions of an index can be maintained by DB2.) But remember, the existing data is not changed or reformatted on disk. Instead, when data is accessed, the changed column(s) will be materialized in the new format.

Updating or inserting data will cause the row to be saved using the format of the new data type. When the object is reorganized or rebuilt, the data will be converted to the format of the latest version specified in the DB2 Catalog. This technique allows DB2 to offer the greatest availability to users, albeit with some performance degradation.

Keep in mind that we just covered online schema change at a high-level. Space limits prohibit a complete discussion of the other types of changes supported by DB2 V8. Suffice it to say, it becomes easier to make changes to database structures under DB2 Version 8 – but this ease of use does not come free. You will need to manage versions and your queries will pay the cost to transform data until it is modified, reorganized or rebuilt.

From [zJournal](#), October/November 2003.

© 2003 Craig S. Mullins, All rights reserved.

[Home](#).