

## Sequences, Unique Indexes and Uniqueness

During the recent meeting of the Finnish DB2 Users Group on Friday 23<sup>rd</sup> September I presented a session that concentrated on Data Partitioning Secondary Indexes (DPSI's). This prompted some discussion around the areas of uniqueness, sequences and indexes defined as unique. The following brief document will hopefully clear up any confusion, should the following raise any further questions or you wish to discuss this further please feel free to contact me at [Andrew\\_ward@bmc.com](mailto:Andrew_ward@bmc.com).

I made the statement during my presentation that a sequence could 'guarantee uniqueness'. For the most part this is true, but the word 'guarantee' is probably a little strong. The only current way to guarantee that a value is 100% unique is to define a unique index on that column or group of columns, if this is a secondary index a partitioned index only strategy is not directly possible as a DPSI cannot be defined with the UNIQUE attribute. But, as I stressed during the presentation, moving to a strategy of using only partitioned indexes does have its benefits. I must make it clear again that this is not something that should be undertaken lightly, especially with existing applications, as performance may well suffer. But, if the correct planning has been put in place and an application that only makes use of partitioned indexes has been proved to be a good performer, then you are potentially left with a problem of ensuring certain values in certain columns remain unique. This is where sequences could come to your rescue.

Let's take a look at an example. Imagine you had an existing application which had an NPSI defined with the unique attribute on column account\_number which resided in a partitioned table. The sole purpose of this index is to guarantee the uniqueness of the values in that column, if you can get unique values from another source this index is not required for queries. For the sake of this example other columns make up the partitioning key. If you wished to move towards a partitioned index only strategy this index would cause you some problems. It couldn't just be changed into a DPSI as these cannot be defined as UNIQUE and the NPSI can't remain if your goal is to achieve full partition independence. This is where a SEQUENCE could be used to generate a unique value.

However, a SEQUENCE is not guaranteed to be unique as it can be altered (via a standard ALTER statement) to be restarted at a number that has already been allocated. They can also be defined to run in cycles, these can also cause already used values to be re-used if the sequence is defined to do so. There are other situations in which a duplicate sequence value can be generated, for a full list see page 336 of the Redbook 'DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know, ... and More' (SG24-6079-00).

If the proper procedures are put in place it would be possible to use a SEQUENCE to resolve the issue in the example above. Proper planning would be required and recovery instructions would need to be reviewed to ensure that the uniqueness of the SEQUENCE was never jeopardised. ALTERing of production SEQUENCES would need to be tightly controlled. Moving to a partitioned only index structure requires a change in thinking, likely changes to existing SQL to provide good performance and rigorous testing to name a few. SEQUENCES could play a part in this if there is a requirement and the correct procedures are in place. At the end of the day, if uniqueness is absolutely critical to the application the only way to guarantee it 100% is to define an index with the UNIQUE attribute. This will of course ruin any dreams of complete partition independence for that application, if this scenario ever occurs at your site SEQUENCES should certainly be a consideration. I hope this clears up any confusion.