

BUSINESS INTELLIGENCE

with Jackie

Indexing with DB2 UDB for iSeries

It is said that the database is the heart of a data warehouse. It is also said that DB2 UDB for iSeries (DB2 UDB) is the heart of the iSeries (AS/400).



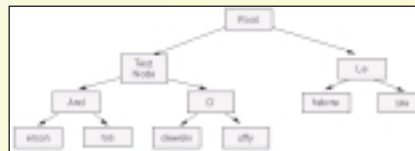
Jackie Jansen

This month I want to cover, at a high level, the indexing strategy for DB2 UDB. Next month my plan is to cover the tools available to analyze your current use of indexes.

Lets start with the basics. A database index is very similar to an index in a book. It points you to the pages of the database table that you want to locate. Without an index in a book you have to look through the entire book to find what you want. The same is true with tables on the iSeries. Without an index you would have to scan a complete table to find the records that meet your query criteria. Index entries are usually much shorter than the table rows themselves. This allows for many more indexes on a page and dramatically reduces the I/O required to service your query. With a relational database manager the query optimizer is used to determine how to retrieve the data. Does it use a full table scan, or does it uses indexes? Which indexes in what combination are used? The answers to these questions are based on statistics such as the number of records you expect to retrieve and the number of times a key field is found in a table. A unique feature of the iSeries is that these statistics are constantly maintained by the database manager. On other systems you need a database administrator to run programs to update the database statistics.

The basic relational database index structure is a binary radix tree which allows a large number of key values to be stored efficiently while minimizing access times. A single key can be located

very quickly with a small number of tests. This index structure is very good for finding a small number of rows. This is the structure most often used in an OLTP environment where you may want to find all outstanding orders for a specific customer. Here is an example based on our famous TUG directors:



The binary radix tree is less efficient for ad hoc queries where you don't know what fields the user will be querying and when you are retrieving large number of records.

Bitmap indexes were designed to solve some of these query inefficiencies. Generally in a bitmap index each unique key value has associated with it an array of bits equal to the number of records in the file. Each bit in the array is a 1 or a 0 depending on whether or not its corresponding record contains the key value. For example, if your file has 5 records in it and records 2 and 3 have a key value of Ontario the array associated with Ontario would look like "01100". Maintenance of bitmap indexes becomes a problem with very large database files.

SYMBOL TABLE:					VECTOR:	
Key Value	Code	First Row	Last Row	Count	Code 1	Code 17
Arizona	1	1	80805	5000	Code 18	Code 8
Arkansas	2	5	89790	7300	Code 2	Code 7
...	Code 38	Code 38
Virginia	37	1222	30111	340	Code 1	...
Wyoming	38	7	83800	2760

With V4R3 IBM introduced a great new patented technology, currently only available on the iSeries called Encoded Vector Indexes (EVIs). EVIs were created primarily to support business intelligence and ad-hoc query environments.

An EVI consists of 2 basic components, a symbol table and a vector. The symbol table contains a distinct key list, key statistics and a unique code. The vector contains a byte code value for each row in the table. The byte code represents the key for that row.

The iSeries optimizer will choose the best access method available for your query.

For more information on indexing see "www-1.ibm.com/servers/eserver/series/db2" and look at "Articles and Whitepapers". Here you will find Kent Milligan's article "DB2 UDB for AS/400 Query Processing with Encoded Vector Indexes".

For those of you even more adventurous review the white paper "Indexing Strategies for DB2 UDB for iSeries" by Amy Anderson and Michael Cain. [TUG](http://www.ibm.com)

Jackie Jansen is a Certified Consulting IT Specialist. She currently works in the IBM Americas Business Intelligence Solutions Centre. Jackie is a frequent speaker at AS/400 Technical Conferences and User Group meetings. Contact her at jjansen@ca.ibm.com.