



Fundamentals of Query Optimization

From basics of Indexing to fast query results

Welcome

Thursday, January 15, 2009

Allen Berezovsky
aberezovsky@harborobjects.com

January 15th



SQL Geek Talk

- ◆ create table CarMakes(
 id int identity
 ,Make varchar(32))
- ◆ insert into CarMakes (Make)
values
('Accura'),('Buick'),('Cadilac'),('Honda')
- ◆ How do you print results as follows:
Accura|Buick|Cadilac|Honda ?



Intended Audience

- ◆ Database Developers
- ◆ Application/Web Developers
 - ▶ Why does my grid takes so long to populate?
- ◆ Database Administrators
 - ▶ How do I make sure that I provide optimal performance for developers
 - ▶ How do I maintain that performance over time



Topics Covered

- ◆ What are “Indexes” and why we use them
- ◆ Fill Factor
- ◆ Optimizer basics
- ◆ What are “Statistics” and how do they relate to Indexing
- ◆ Real Live examples
- ◆ Developing a good maintenance plan



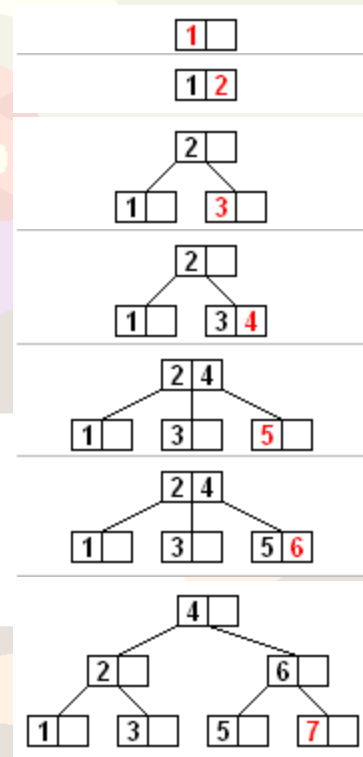
What is an Index?

- ◆ An Index is a COPY OF PART OF A TABLE used to improve the speed of data retrieval from that table
- ◆ Indexes can be created using one or more columns of a table
- ◆ Disk space required to store the index is typically less than required by the table
- ◆ An Index can be either Clustered or Non-Clustered
- ◆ SQL Server implements indexes using a B-Tree structure



What is a B-Tree?

- ◆ A tree data structure that keeps data sorted and allows searches, insertions, and deletions in logarithmic amortized time
- ◆ Binary Tree is searched starting at the root, and is traversed top to bottom.
- ◆ If the leaf node contains fewer than the maximum legal number of elements, there is room for one more. Insert the new element in the node, keeping the node's elements ordered.
- ◆ Otherwise the leaf node is split into two nodes.





Non-Clustered Indexes

- ◆ Think of an index in a textbook
- ◆ The data is stored in one place, the index in another, with pointers to the storage location of the data.
- ◆ The items in the index are stored in the order of the index key values, but the information in the table is stored in a different order
- ◆ One Table can have up to 256 non-clustered indexes



Non-Clustered Indexes

- ◆ Optimal choice for exact match queries
- ◆ Columns that contain a large number of distinct values, such as a combination of last name and first name
- ◆ Queries that do not return large result sets.
- ◆ Decision-support-system applications for which joins and grouping are frequently required. Create multiple nonclustered indexes on columns involved in join and grouping operations.



Clustered Indexes

- ◆ Determines the physical order of data in a table
- ◆ Think of a phone book
- ◆ Because the clustered index dictates the physical storage order of the data in the table, a table can contain only one clustered index
- ◆ PRIMARY KEY constraints create clustered indexes automatically if no clustered index already exists on the table and a non-clustered index is not specified when you create the PRIMARY KEY constraint



Clustered Indexes

- ◆ Use for queries that return a range of values using operators such as BETWEEN, >, >=, <, and <=
- ◆ Columns that are accessed sequentially
- ◆ Queries that return large result sets
- ◆ Columns that are frequently accessed by queries involving join or GROUP BY and ORDER BY clauses
- ◆ Define the clustered index key with as few columns as possible



Fill Factor

- ◆ Data is stored in pages
- ◆ When a new row is added to a full index page, SQL Server moves approximately half the rows to a new page to make room for the new row
- ◆ Page splitting can impair performance and fragment the storage of the data in a table
- ◆ When creating an index, you can specify a fill factor to leave extra gaps and reserve a percentage of free space on each leaf level page of the index to accommodate future expansion in the storage of the table's data and reduce the potential for page splits



Fill Factor

- ◆ 100% means use up the entire page. (Use only on read-only pages)
- ◆ Database reads typically outnumber database writes by a factor of 5 to 10.
- ◆ Therefore, specifying a fill factor other than the default (80%) can degrade database read performance by an amount inversely proportional to the fill factor setting
- ◆ It is useful to set the fill factor option to another value only when a new index is created on a table with existing data, and then only when future changes in that data can be accurately predicted



Query Execution

- ◆ SQL Server processor consists of two components
 - ▶ Query Optimizer
 - ▶ Query Execution Engine
- ◆ SQL Server breaks queries down into fundamental building blocks that we call operators or iterators
- ◆ Iterators can be Nonblocking and Blocking



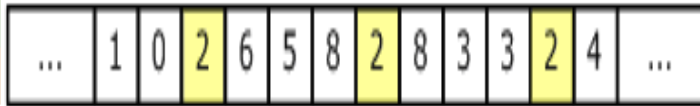
Iterators

- ◆ Iterators that consume input rows and produce output rows at the same time are “nonblocking”
- ◆ Iterators that consume all input rows before producing any output rows are called “blocking” iterators
- ◆ Table or index scan, Index seek, Bookmark Lookup, Data filter, Data Sort, Data Join, etc. Are examples of iterators

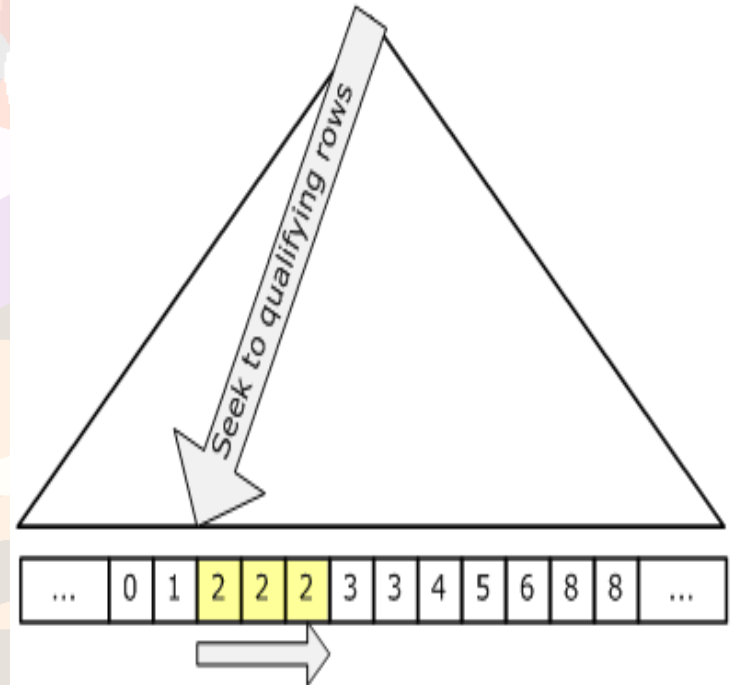


Scan VS Seek

Scan



Seek





Statistics

- ◆ Statistics is information that the server collects about the distribution of data in columns and indexes, and are used by the query optimizer to determine the best execution plan
 - ▶ The number of rows and pages occupied by a table's data
 - ▶ The time that statistics were last updated
 - ▶ The average length of keys in a column
 - ▶ Histograms showing the distribution of data in a column
 - ▶ String summaries that are used when performing LIKE queries on character data



Updating Statistics

- ◆ SQL updates statistics as soon as the need is discovered
- ◆ If the rowcount in the table has gone from zero to any nonzero value
- ◆ If the number of rows in the table was under 500, and at least 500 rows have been modified
- ◆ If the number of rows in the table was over 500, and at least 500 plus 20% of the rowcount rows have been modified
- ◆ `AUTO_UPDATE_STATISTICS_ASYNC`
- ◆ `UPDATE STATISTICS` table
- ◆ `Sp_UpdateStats`



Key Resources

- ◆ http://www.sql-server-performance.com/articles/per/index_data_structures_p1.aspx
- ◆ <http://en.wikipedia.org/wiki/B-tree>
- ◆ <http://www.developer.com/db/article.php/3622881>
- ◆ <http://blogs.msdn.com/sqlqueryprocessing/>



Contact Information

- ◆ EMAIL

- ▶ Aberzovsky@HarborObjects.com

- ◆ FACEBOOK

- ◆ LINKEDIN djaldogz@aol.com