

## Sybase ASE 15 – Semantic Partitions to the Rescue

---

BY PETER DOBLER, DOBLER CONSULTING

Data explosion and the always present request for faster data retrieval led to the introduction of more granular data distribution on physical disks. Even with faster disks, the increase in parallel processing created more contention on a single table.

The solution to these database challenges was to split a single table into multiple partitions that could be accessed independently and still maintain the data integrity of a single table. A second solution was the introduction of row level locking to drastically reduce data insert contention.

Still, more data distribution challenges call for other solutions. Partitions were slowly introduced with Sybase ASE 11 to reduce contention and support parallel processing. However, it was not until ASE 15 that partitions jumped back into the spotlight.

### **BENEFITS USING SEMANTIC PARTITIONS**

There are several advantages to using semantic partitions. For example, you'll have reduced maintenance time and more predictable maintenance tasks that will reduce "just in case" maintenance, because you can perform maintenance tasks on just one or more partitions, instead of the whole table (which many DBAs practice).

A second benefit is for applications that are dealing with date and time based data, like sales data, telecommunication information, bank transactions or patient insurance information. They will all experience a dramatic performance gain with range partitioning.

Reduced maintenance windows and improved query performance with range partitioning also boosts productivity on more than one level.

Two of the semantic partition types offer specific gains.

Hash partitioning doesn't provide the flexibility of range partitioning. Yet, its strength is in spreading the data over as many partitions as possible to provide a performance boost to non-time based queries.

Rolling range partitions benefit from archive data placements onto cheaper hardware. This means that data can "age", and older static data rows will reside on partitions that can be moved from "tier 1" storage to "tier 3" storage. Cost savings between "tier 1" and "tier 3" storage is significant. The more data, you can move to "tier 3" storage, the bigger the savings. Rolling range partitions are not a function of Sybase's semantic partitions, but a technique to effectively manage date driven data. The key is to create date range partitions without a MAX limit. This will manage the data growth and future data distribution into the range partitions.

### **REDUCE MAINTENANCE TIME USING SEMANTIC PARTITIONS**

By using the new semantic partitions in Sybase ASE 15, DBAs are learning that having increased operational scalability is a genuine reality, as a result of reduced maintenance time.

Recent internal benchmarking at a client installation tracked time reductions for update statistics tasks by a factor of 25 – 30. The 25 – 30 factor is absolutely correct . . . reductions for update statistic tasks by double digit factors, not percentages! These reductions are significant, and allow DBAs to reclaim their maintenance window. In other words, by using semantic partitions, a 3 hour update statistics job completes in 6 minutes.

## IMPROVE AVAILABILITY AND MAINTENANCE USING SEMANTIC PARTITIONS

By reducing the maintenance windows, application availability automatically increases. As mentioned earlier, reducing maintenance jobs from 3 hours down to 6 minutes will repurpose at least 2 hours for application use, which translates in 2 hours of added productivity for applications and users.

Semantic partitions also allow DBAs to be more selective and pro-active in the choice of maintenance tasks. In large systems, it is not uncommon for maintenance to be spread over every day of the week to guarantee a complete weekly cycle. The time window allowed for maintenance is too short to complete the task in one step. Having the ability to complete this maintenance in the short time frame is a huge advantage. By using the semantic partitions, previously occupied system resources quickly become available again to applications.

## WHAT TYPE OF PARTITIONS DOES SYBASE USE?

There are different types of partitions that can be used to take control over the data distribution of a single table. Each table can maintain its own partition type.

### Round-Robin Partitioning

This was the first partition type introduced in Sybase ASE 11.0.3. The data is evenly distributed amongst the total number of partitions available. There are drawbacks with this partition type. The most severe is that you cannot dynamically add a new partition to an already partitioned table. The only way to add a new partition is to unpartition the original table, and then repartition the table.

Another drawback to Round-Robin partitioning is that partitions need to be rebalanced manually and often. Each time you rebalance, you have to drop and recreate the clustered index for this table. Most installations used a very low number of partitions, no more than 10, because the rebalancing was labor-intensive, complex, and time-consuming.

This is the only partition type available prior ASE 15.

### Semantic Partitioning Options

Sybase introduced semantic partitions in its ASE 15 version. Semantic partitioning uses the data values to decide which partition the data resides in vs. the session id.

Sybase has 3 types of semantic partitioning: range, hash and list.

### Range Partitioning

The Range partitioning is the most flexible and adaptable partition type in ASE 15. Some of the advantages to using range partitioning are:

- Reduces maintenance time by eliminating the need to run maintenance tasks on older static data rows.
- Implements rolling, alternating or migrating partitioning schemes to constantly archive older data rows.

Range partitioning offers, by far, the most features and flexibility to manage time-period aware data. In today's data explosion, the bulk of the data is growing in this area. An excellent use for range partitioning is financial transaction information, with time-period aware data.

### Hash Partitioning

Like the Round-Robin partitioning, hash partitioning is distributed evenly over the available partitions. The advantage to hash partitioning is that no rebalancing is necessary. The Hash partition type is most effective with data that does not have a life cycle, like sales records, or data that does not require any form of range queries. This partition type is most likely used on primary keys.

The drawback is you have the same flexibility limitations for hash partitioning you had with Round-Robin partitioning.

### List Partitioning

This partition type allows you to dynamically add new partitions without unpartitioning and repartitioning the table. Like hash partitioning, the list partitioning offers the best performance on queries with exact matches. This means "column = constant" works best with this partition type.

A natural fit for list partitioning data is data organized by geographic region or separating branch data by store location. The key to a successful list partitioning implementation is low cardinality. Based on performance benchmarks cardinality of less than 10 is most effective with list partitioning.

## NEW INDEX TYPES

With the introduction of semantic partitions, Sybase also introduced a new type of indexes. Global and local indexes are being used in tandem to effectively manage the data on partitions. Local indexes are used to create an index on a single partition. Local indexes have no knowledge about data outside the partition they are managing. Thus, creating a unique key on a local index is not possible. The advantage to using local indexes is that the index tree has few levels, which reduces the amount of I/O. For example, if you can reduce local index levels from 6 or 7 to 3 or 4, your I/O reduction will be significant. You will have less contention on the index root page also.

Global indexes spawn all partitions for the entire table, and index all rows in the entire table. To enforce primary key uniqueness, a global index is necessary. As a rule of thumb, use global indexes only if absolutely necessary. A performance benchmark study compared a drop partition task performed on a partitioned table with a global index vs. local indexes. The tasks completed in seconds when using local indexes, but ran for a couple of hours to keep the global index in sync.

## SEMANTIC PARTITIONS OFFER FULL CONTROL OVER DATA PLACEMENT

When partitions were first introduced, the primary goal was to reduce contention during insert statements by creating several last data pages. Up to this point, clustered indexes were the only method to distribute inserts.

Prior to the semantic partitions in Sybase ASE 15, only Round Robin partitioning was available as a partition type. Round Robin partitioning solved the insert contention problem, and improved parallel SQL execution, but left another serious problem. Round robin partitioning gave DBAs no control over the data placement within the partitions.

With the introduction of semantic partitions, users now have full control over data placement within individual partitions of a table. This opens the door to very finely tuned maintenance tasks that slash the time required to complete these activities. Prior to semantic partitions, all maintenance tasks were at the table level. At that time, maintenance on all partitions of a table needed to be completed to finish the maintenance. Partitions now can be used to spawn several database processes in parallel, speeding up execution significantly.

## WHERE SEMANTIC PARTITIONS ARE NOT A GOOD FIT

Semantic partitions can improve almost any data distribution situation. In many cases, there are performance improvements with queries as well. It is fair to say that adding semantic partitions offers improvement opportunities with almost no side effects.

However, applying semantic partitions to all tables is not a free ride for DBAs. Administrative overhead still exists. DBAs need to keep an eye on partitions to ensure optimal performance to the applications.

Internal analysis showed that partitions have no effect on contention elimination when used on data row locking scheme tables. This means that row level locking provides enough separation to deal with table contention.

Before using range or hash partitioning, DBAs need to consider:

- Although the range partitioning is the most powerful semantic partition type, studies show there were no performance improvements past the date range search arguments on time period based data.
- Applying hash partitioning to time based data is not a good idea. It can have a negative impact on both the query performance and the maintenance time allocated to maintain this constellation.

Sybase provides the tools and metrics to determine the best use of semantic partitions. The two considerations mentioned above will help DBAs make the best decisions for their situation. DBAs can use Sybase's expertise, along with deciding what their own best performance needs are.

## WHAT'S NEXT

Sybase is planning to introduce several new features on semantic partitions in the near future. All of them will further enhance the performance and the maintainability of partitioned data.

The most important future enhancements are:

- Unique keys on local indexes
- Merge partitions
- Split partitions

Sybase is committed to provide the best tools supporting DBAs to tame the data volume explosions and providing the highest level of database uptime to end-users and applications.

## CONCLUSION

At first glance, semantic partitions may only seem to be valuable during maintenance tasks, but a closer look under the covers reveals that there are more benefits than meets the eye.

Hash partitioning and even Round-Robin partitioning over many partitions can dramatically improve performance over primary key data access. Hash partitioning thrives with equality search arguments in queries. List partitioning over low cardinality data, like geographic location distribution, can open new maintenance opportunities for DBAs maintaining a global data and user community. Performing data maintenance on partitions for geographic regions that are separated by global time zones allows DBAs to maintain optimal performance on databases that do not allow for downtime.

Rolling range partitions provide the capability to automatically “roll off” archive data, based on date ranges, onto cheaper disk hardware. This will save money and does not require additional maintenance. With range partitioning, maintenance tasks can be applied to individual partitions without the need to perform maintenance on all partitions of a table. With the `datachange()` function, DBAs can determine which partitions need maintenance. This allows for surgical precision in maintenance efforts and dramatically reduces the time frame to execute these tasks.

---

### About the Author

*Peter Dobler is an accomplished IT database professional who “makes a difference” by improving efficiencies and reducing costs for small and medium-sized businesses. He founded Dobler Consulting, ([www.doblerconsulting.com](http://www.doblerconsulting.com)) a Tampa, Florida consulting firm that delivers implementation expertise for Oracle, Sybase, and MS SQL Server in 2000. His 25 plus years in technology started in Basle, Switzerland in 1985. Peter uses his extensive experience to hone his talent as a proven resource for producing streamlined IT solutions. He currently engages in strategic alliances and special projects with Sybase, the enterprise software and services company. He can be reached at: [pdobler@doblerconsulting.com](mailto:pdobler@doblerconsulting.com) or 813-322-3240.*

*Read his Database Trends Blog (Behind the Scenes of Database Evolution) at: <http://www.peterdobler.com>, and his Technology Tips Blog (Step-by- step Instructions on Today's Challenging Technology) at <http://www.techtipsntrick.com>.*

### Other Articles of Interest:

*“Sybase ASE 15.5 —The Need for Speed”, Database Journal, [www.databasejournal.com](http://www.databasejournal.com), May 2010*

### Reviews on Peter Dobler's articles

*“IMDB said to be the New Frontier of Database Architecture,” <http://www.dbajobsandcareers.com>, Database Blog, Review, June 1, 2010*

*“Performance Enhancement and the In-Memory DBMS Opportunity”, International Sybase User's Group, <http://www.isug.com/common/Index.html>, March, 2010*