Mercury
Consulting Ltd.

Bahnhofstrasse 7, FL-9494 Schaan, Tel. +423 236 44 06 Fax +423 236 44 07
e-mail: office@Mercury-Consulting-Ltd.com
http://www.Mercury-Consulting-Ltd.com

Daily, Weekly or Monthly Partitions ?

A discussion of several factors for this important decision

iT Checklists
www.IT-Checklists.com

Published in July 2006

Conventions

The following typographical conventions are used in this book:

Constant width

Used for source code

Constant width bold

user input

[ ] optional elements

{ } mandatory choice

| separates choices

Table of Contents

Summary

This document evaluates the different factors impacting the decision if daily, weekly or monthly partitions should be used. Surprisingly even the legal department can impact that decision, and the other areas as performance and manageability become just secondary decision factors.

As the optimal decision depends only on implementation (customer) specific circumstances there exists no general answer to this question. But the criteria listed and the impact evaluated in this document will enable you to find the best decision for your situation.

Audience

- Operations Manager
- Application Support Staff
- System-DBA, Application-DBA, Development-DBA
- Legal Department / Compliance Officer

Tabular Overview

	Low number of partitions ***(Monthly, Quarterly, Yearly)***	***High number of partitions*** ***(e.g. daily partitions)***
Legal / Compliance		
[L.1] Meeting legal requirement not to keep certain data more than <xx> days and business requirement to keep as long as possible (allowed).	(-)	(+)
Performance		
[P.1] Cache Hit Rate on Indexes during insert *(assumption: nearly all inserts from current day)*	(-) lower	(+) higher – daily index-partitions are small, and only those of current day are accessed during inserts.
[P.2] Index-access using indexes not starting with partition-range column	(+) fast, only low numbers of indexes to access	(-) slower, high number of indexes to access. *Might be mitigated by application logic, see details.*
[P.3] Performance on Full-Table-Scans		Depend on the usual conditions
Maintenance		
[M.1] Size of partition (segment size)	bigger	smaller
[M.2] Daily backup volume (and time to restore) if only current partition is in "read/write" tablespace, all other in "read-only" tablespaces	bigger	smaller
[M.3] Partition maintenance (adding, removing)		When fully automated, then constant, short daily job which can be easily integrated into monitoring systems.
[M.5] Partition Maintenance (statistics)		
[M.5] Partition maintenance (sorting)	Long duration (+ big archivelog volume) For long time unsorted records	Short duration only a few days unsorted records
M.6] Partition Maintenance – Index-Rebuild		

Table 1: Overview - Factors to be considered

Evaluation

The indicators in square brackets refer to the related cell in the tabular overview, e.g. [A-L] refers to column “Low number of partitions” of row A.

Legal / Compliance

In many cases this regulations or laws might not be applicable to your type of data, then just skip that.

In those cases where laws or regulations require that records must be deleted after a certain time, then it might not be easy to meet this legal requirement when using monthly partitions.

Example: Data Privacy Protection Laws in some countries require that the details of telephone calls must be removed <xx> days after the customer received the invoice and did not dispute the invoice.

[P] Performance

[P.1] Cache-Hit Ratio on Inserts

Example: 100 rows of 120 bytes average row length are inserted, database block size = 16 KB.

Segment	***Blocks to be accessed***
Database table	1 database table block
Index on Sequence-Number (artificial Primary Key)	1 index leaf block
Index on customer_id	100 index leaf blocks
Index on last_name, (|| first_name)	100 index leaf blocks

Table 2: Number of index- and table blocks to be accessed when inserting 100 records

Although only one empty database block needs to be read from the disk for inserting 100 records, 201 blocks might be read from disk (if not in cache) for updating the index! This example clearly demonstrates why indexes do slow down inserts!

The critical factor of INSERT-performance
is the cache hit ratio on the indexes!

Partitions....	***Segment size of one index partition***	***Cache Hit rate if INSERT's only from same day***
Daily partitions	small	good
Weekly partitions	7x bigger	
Monthly partitions	30x bigger	Low (except there is such much block buffer cache to cache even the monthly partitions.

Table 3: Impact of partition range on cache hit ratio during inserts

Comment: If you load the data just once a week, but the data are ordered by date (or each single file contains only data from one day), the there is still an advantage of daily partitions, as at the start of the loading job the partitions from first day will be in cache, and afterwards the index-partitions from 2nd day will be in cache, aging out those from first day.

If your index-partitions are not fully cached, then you will have disk-reads on the index-segments *(but none on the database-table),* and your INSERT will wait on those disk reads!

[P.2] Index-Access

As global indexes cause significant performance degradation when dropping old partitions (and when exchanging partitions) usually only local indexes are used.

In our example the table is range-partitioned by date. Therefore index access on columns as customer_id, product_id, .. require the access of all local index-partitions. The next table shows the performance impact:

Example: 9 month of order-history online

Partition-unit ➡ ***WHERE-Clause*** ⬇	***Daily*** ***(275 partitions)***	***Weekly*** ***(40 partitions)***	***Monthly*** ***(9-10 partitions)***
Select ... from <table> where customer_id = :v_cust_id	275	40	9 (10)
Select ... from <table> where customer_id = :v_cust_id and transaction_date > trunc(sysdate)-31;	32	5	2
Select ... from <table> where customer_id = :v_cust_id and transaction_date > trunc(sysdate)-7;	8	2	1 (sometimes 2)

Table 4: Number of index-partitions to be accessed

In case that for one customer_id there are several records per day, the overhead caused by the higher number of index-partitions is very low at a first glance. But there is a second factor to be considered: How many records of the same customer_id are in the same data block?

Assumption: 120 bytes/record, 16 KB db block size, 10% pct_free ➡120 rows/block

Partition-unit ➡ ***WHERE-Clause ⬇***	***Daily*** ***(275 partitions)***	***Weekly*** ***(40 partitions)***	***Monthly*** ***(9 partitions)***
Select ... from <table> where customer_id = :v_cust_id	275 indexes +	40 indexes +	9 indexes +
1 row per months unsorted / sorted	9 / 9 db-blocks	9 / 9 db-blocks	9 / 9 db-blocks
1 row per week unsorted / sorted	40 / 40 db-blocks	40 / 40 db-blocks	40 / 9 db-blocks
1 row per day unsorted / sorted	275 / 275 db-blocks	275 / 40 db-blocks	275 / 9 db-blocks
10 rows per day unsorted / sorted	2,750 / 275 db-blocks	2,750 / 40 db-blocks	2,750 / 27 db-blocks
.....10 rows per day, only last month: unsorted / sorted	310 / 31 db-blocks	310 / 5 db-blocks	310 / 3 db-blocks
.....10 rows per day, only last week: unsorted / sorted	70 / 7 db-blocks	70 / .5 db-blocks	70 /..1 db-block

Table 5: Number of data blocks to be accessed when selecting by e.g. customer_id

Note: when your data are sorted by customer_id when inserting, you do not need to sort the partition. As history of usage is usually not cached, you can expect a physical read for each data-block. At 6 – 12 milliseconds per physical read 275 physical reads will result in 1.6 to 3.3 seconds

In case of high average number of order-details, phone-calls or other details per customer and day the default-limit of showing only first week or first months should be really discussed when specifying the requirements.

[P.3] Performance on Full-Table-Scans

If daily or weekly partitions are an advantage depends solely on the typical date-ranges of reports.

[M] Maintenance

[M.1] Size of partition (segment size)

A important, but simple factor. And definitely not the only one.

[M.2] Daily backup volume (and time to restore) if only current partition is in “read/write” tablespace, all other in “read-only” tablespaces

Reducing Time-to-Recover

50% to 90% of OLTP-data are historic usage data. The smallest recoverable unit is a tablespace. Placing partitions in different tablespaces allows you in case of restore to restore and recover your database with the most current partitions quite fast. Restoring the other 50%-90% can be postponed until the OLTP-backlog is processed.

Reducing Backup-Volume

Placing each partition (or e.g. 7 daily partitions) into a separate tablespace will allow you to set after a certain time period the tablespace into ‟read-only” mode. If you adjust your backup concept to fully utilize this great database feature then you can significantly reduce your daily backup time and backup volume. You do need to understand and test this concept very well, and you do need the willingness of your backup team and disaster recovery manager to agree on this more sophisticated and complex backup concept. And again, your new backup concept MUST automatically deal with the scenario that a tablespace which was since long time in „read only‟ mode, was set back „to read-write‟ and again to „read-only‟, perhaps just for a few seconds or minutes to process one old record.

[M.3] Partition maintenance (adding, removing)

Manual creation of statistics for one year ahead has the risk that after one year nobody thinks about that and you run out of partitions – you definitely need to monitore the number of available partitions!

A fully scripted daily partition maintenance (adding, removing,...) including all error-handling is a pretty complex challange.

[M.4] Partition Maintenance – Statistics

It makes definitely no sense to analyze the empty partition.

Analyzing after 1 hour and 1 day of usage might be a simple idea.

The most effective approach requires some little scripting:

- Analyze one representative parttion and their indexes.
- Export those statistics once.
- Import those statistics into each new created partition.

[M.5] Partition maintenance - Sorting

If sorting the partitions will provide benefit at all needs to be determined, see [P.2].

If your application will benefit from “clustered” data, then you need to decide if an index- or hash-clustered table should be used. Alternatively you should consider “sorting”:

When to sort ?

- Are there “delayed” inserts, that means inserts are a few days after the value of the date-column used for partitioning ?

If yes, then you should sort when nearly all delayed inserts were processed.

When using **monthly partitions** you can only sort after end of month, that means you will work between 1 and 31 days (in average 15 days) on an unsorted partition, and access to the last month might be the most frequent. **Daily partitions** could be sorted next day or after a few days (dependent on the average delay).

Duration of sort / maintenance window

Sorting of single partitions is usually achieved by exchanging that partition to a table, sorting this table and exchanging the sorted table into a partition.

Sorting a monthly partition will take much more time than sorting a daily partition.

If you don't use “nologging” option (e.g. because you use a physical standby database, or if you in general don't use “nologging” on your critical OLTP, then sorting monthly usage (and index-rebuild) will create huge amount of archive log. - Make sure that your archivelog destination is big enough or your backup-to-tape is fast (and reliable) enough!

Dependent on how you execute this sort this table might not be accessible for all or certain functions of your application.

Monthly partitions: long duration, huge archivelog amount

Daily partitions: faster sorting, less archivelog-amount, but only feasible when fully automated.

[M.6] Partition Maintenance – Index-Rebuild

The main question is:
Is index-rebuild necessary at all ? Are there massive deletes and updates at all ?
As removing old data by dropping the partition (DDL-operation) instead of using the DELETE-command (causing huge server load and archivelog / transaction log volume) major DELETE's are not expected.

How Mercury Consulting Ltd. can help you

Mercury Consulting Limited (MCL) is a professional consultancy providing experience, support and training in IS/IT operations for companies during high-time-pressure startup phase and following consolidation phase, especially in the Telecom-market.

Our Products

you can purchase online and immediately download at our eBook-Shop

iT Checklists
www.IT-Checklists.com

Our Checklists and Templates will help you to ensure that good or best practice is not only known but consistently applied!

Database Independent Products

Product	***Benefit***
150 Non functional Requirements.	Requirements Template with 150 non functional requirements for selecting or developing robust, reliable, stable and manageable applications to meet the Service Level Agreements (SLA's). For external RFP's (Request for Proposal) and for internal development.
Checklist for Data Migration	65 important questions to identify and address or exclude typical migration pitfalls in an early phase of the project, thus ensuring the confidence for keeping the time plan.
Template: Systems- and Operations Handbook	Template to establish that documentation auditors like to see!
Interface Checklist	Those questions which you need to ask before starting the development! Requirements, Checklist and Template for Planning, Defining and Documenting Application Interfaces.
Checklist for Production Release and System Handover	Checklist for small and medium projects focusing on non-functional aspects for operations team. Simple but effective!
Business Requirements for Archiving and Purging	Template with Business Requirements for Archiving & Restore & Purging. Not removing old customer data can cause conflicts with privacy laws. ***Business must act*** and clearly specify what to purge and what to archive!
Application Health Check: Stability Assessment	Using this template to check your systems - and DOCUMENT the findings might show you even more potential issues beyond invalid objects.
Technology Selection for Disaster Recovery (DR)	Standby database using transfer of transaction log / archive logs or replicating the database files ? Host-based, Storage based or Switch/Fabric based replication ? This document evaluates the different techologies and their advantages and constraints from an operational point of view.
Backup SLA / OLA: Operations Level Agreement with the Backup-Team	A template which supports the creation of an operational level agreement (OLA) between application support, database administrators and backup team.

Database Specific Products

Product	***Benefit***
Template for Database Operations Level Agreement (OLA) / SLA	If a Service Level Manager needs to offer to Business a Service Level Agreement (SLA) for an End-to-End IS/IT-Service, he sign this SLA only after he arranged within IS/IT for each system or component used to provide this service an **Operations Level Agreement (OLA)** with the providing team or department. This document provides a template for such an Operations Level Agreement (OLA) for Databases containing the agreed values and for QA-purpose also the measures implemented to reduce the likelihood of violations of those agreed values. It does not only deal with availability, but contains also comprehensive service catalogue of advanced DBA services and a template for the Service Level Reporting (SLR).
Database Health Check - Part 1: Stability Assessment	Stability Assessment of your Database. Most Application- and Database-Crashes can be avoided when detecting early indicators and reacting to them. ***Be Proactive - Check Now!***
DBA and Application Support: Job Description and Self Assessment	Checklist to ensure that all 60 DBA-duties are assigned: System DBA, Development DBA and Application DBA versus Application Support If you have just a job role "DBA", but not a dedicated job role "Application DBA" those 19 duties must be explicitly assigned to either "Application Support" or to the "DBA" - otherwise they might not be executed! ***Detect unassigned tasks before an auditor reports them !*** This product addresses disputes between System DBA and Application Support (or, if existing, dedicated Application DBA) regarding the responsibility for the application's database objects.

Our free Whitepapers

http://www.mercury-consulting-ltd.com/wp/whitepapers.html

Whitepaper	***Benefit and Description***
The Importance of Application Level Monitoring.	Keeping your applications free from invalid objects is an important task, but does not guarantee error free operations. This free whitepaper explains the difference between "Application Level Monitoring" to "Database Monitoring" and "System/Server/OS"-Monitoring.
Important Facts about Redolog- and Archivelog / Transaction Logs in Databases for Change Managers and Application Support Staff	Understanding the topic “archivelog volume” can avoid unexpected troubles when applying changes or conducting upgrades. This awareness paper explains why Change Managers must ask questions about the archivelog volume created during changes and upgrades and why application support staff – and not only the DBA - must understand this topic.
The Danger of Invalid Database Objects	An awareness paper for Operations Managers and Application Support describing the problems and potential risks caused by invalid objects in Databases.
Private or Public Synonyms – or no Synonyms at all ?	A decision support paper visualizing the pro's and contras on a single page in tabular form and evaluating the arguments.
Daily, weekly or monthly partitions?	This document evaluates the different factors impacting the decision if daily, weekly or monthly partitions should be used. Legal- , performance- and maintenance aspects are evaluated.

Our Services

http://www.Mercury-Consulting-Ltd.com/services.htm

About the Author

The Author of this white paper is an

ORACLE Certified Professional

ORACLE 10*g* Certified Professional

With 12 years experience as DBA and 6 years experience in Telecommunications Companies.