



Data Management: Table Partitioning and Multi-Tenancy

Libor Laubacher

DBA Consultant

October 2018



Sample Agenda

- What this session is about
- What is Table Partitioning?
- What is Multi-Tenancy?



What this session is about

About

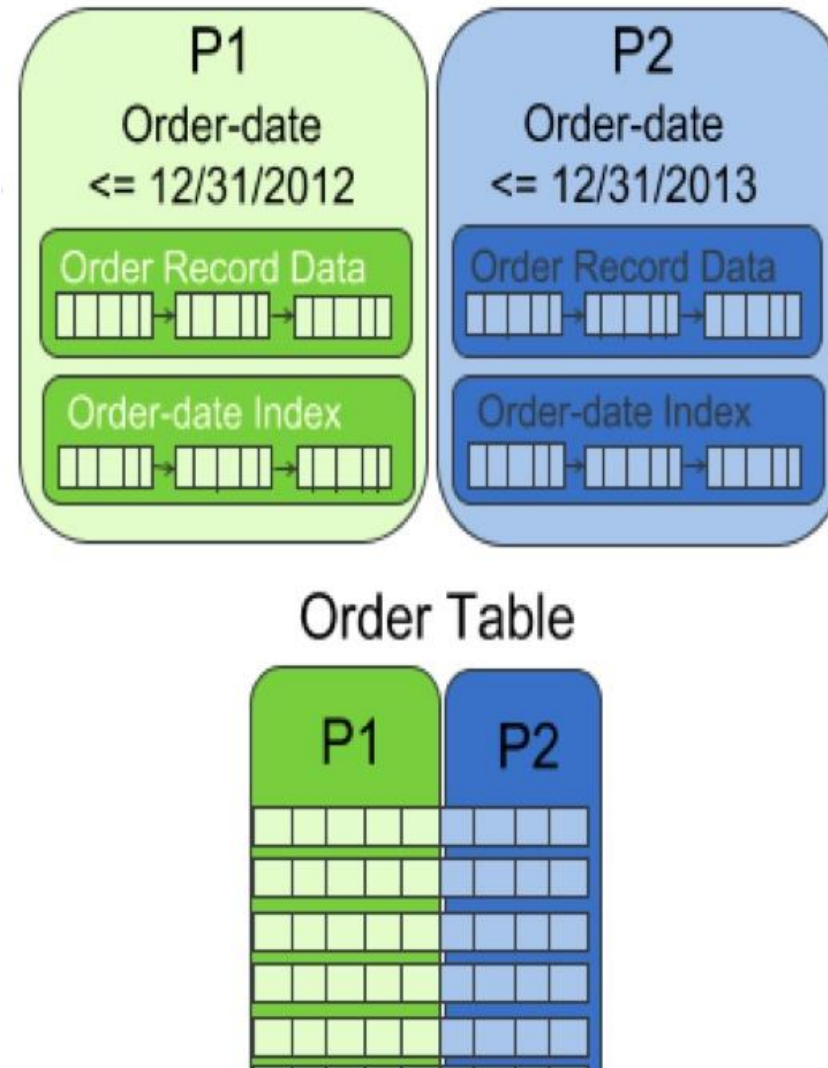
- Table Partitioning and Multi-tenancy are two methods of securely dividing an OpenEdge database. One increases performance and scale, especially for analytics needs, while the other can drive down costs and increase margins in cloud environments. Learn how to apply these technologies for a competitive advantage.



What is Table Partitioning?

What is Horizontal Table Partitioning?

- Table instance stored in multiple self-contained locations
 - Typically used in OLTP deployments
 - Based on user specified column value and storage area
 - Different from partitioning by user identity (multi-tenancy)
 - Row does not span partitions
 - Referred to as just **Table Partitioning**



Why Partition?

Maintenance

- Data re-org
- Data rebuild
- Data purging
- Data archival

Availability

- Data repair
- Data isolation
- Historic data access

Performance

- “Hot” table
- “Hot” index access

What is it good for?

Advantages

- Performance Impact
 - Partition elimination for queries (“pruning”)
 - Improved concurrency
 - For random activity
- Availability
- Maintenance advantages
 - Purge, Archive
 - Repair, rebuild
 - Partition level tuning

Disadvantages

- Partition alignment and lookup (insert/delete)
 - Update of partition aligned key values
 - Missing aligned columns in where clause
- DBA getting it right
 - Knowledge of application table definition and physical layout
 - Repartitioning costs
- More complex deployment

Features

- Create “named” data partitioned storage based on column value
 - Based on single value = List Partitioning
 - Based on value range = Range Partitioning
 - Sub-partitioning partitioning
 - Combination of multiple list and/or range partitions on same table
 - Composite partitioning
 - Multiple partition definitions with storage in the same physical location
 - Table partitioning based on multiple values in single column
 - Initially supported for migration only
 - Partition level pruning (partition elimination)
 - Partition level locking (administrative operations)

Key Benefits

- More focused maintenance
- Considerable flexibility in partition management
- Greater query efficiency
- Transparency to the application, with minimal changes required

Types of Partitions

List Partitioning

OR

Range Partitioning

OR

Sub-partitioning
(up to 15 levels!)

Order Table
Western Region
Northern Region
Southern Region

Order Table
12/31/2011
12/31/2013
12/31/2015

Order Table		
Western Region	Western Region	Western Region
12/31/2011	12/31/2013	12/31/2015
Northern Region	Northern Region	Northern Region
12/31/2011	12/31/2013	12/31/2015
Southern Region	Southern Region	Southern Region
12/31/2011	12/31/2013	12/31/2015

List Partitioning

- Look for a “well known” grouping by “static data value”
 - Known at creation time, changes infrequently
- Data organized geographically or grouped by specific entities
 - Exact match
 - Country, Region, Company, Division
 - Why or why not Sales Rep?
- Consider number of unique data values
 - 32,765 max defined partitions per table
- For best performance: Spread the data out

List Partitioning



Range Partitioning

- Data organized by ranges of values
 - Range rather than single value to identify groups of data
 - Date (by year is most typical)
 - Usage: Calendar year, fiscal year, quarter
 - Order Date vs Ship Date
 - Consider affect on index choice
 - Alphabetic or Numeric range
 - Product Code
 - Usage vs Balance: Group related products, balance A-Z spread
- For best performance: Spread the data out

Range Partitioning



Sub-Partitioning

- Slice and dice data further
 - By Region By Order Date
- For best performance
 - Sub-partition AND spread the data out

Sub-partitioning

Order Table		
Western Region 12/31/2011	Western Region 12/31/2013	Western Region 12/31/2015
Northern Region 12/31/2011	Northern Region 12/31/2013	Northern Region 12/31/2015
Southern Region 12/31/2011	Southern Region 12/31/2013	Southern Region 12/31/2015

Indexing

- Local index support
 - One index b-tree per partition, index MUST be partition aligned
- Global index support
 - One index b-tree containing all the table's data
 - Ability to index across partitions
 - Typically for non-partitioned aligned sort order
 - Provides uniqueness support for non-partitioned columns
 - Unique cust-num or sort by name for example
- Composite index support
 - One index b-tree containing entries for multiple partition definitions
 - Table data for composite partition stored in same partition

Restrictions

- Must be Type II storage area
- Cannot span storage areas
- Re-partitioning only supported via dump and load
- No overlapping range values or gaps for range partitioning
- Only 1 range component per partitioned table
 - Must be the last component
- No partitioning of multi-tenant tables
- Indexing restrictions
 - Local indexes ALWAYS follow table partition definition, partition aligned
 - Word indexes cannot be local indexes, not partition aligned
 - Local index partition id same as matching table partition id



What is Multi-Tenancy?

Multi-Tenancy

- Provides database support to a number of separate and distinct groups of users for effective deployment and management of cloud applications.

What is it?

- Partitioning by a user identity

Terminology

- Tenant
 - Named group of users
 - Share same application AND data
- Multi-tenancy
 - Deployment supporting multiple tenants
- Default Tenant
 - Users who do not assert tenant identity
- Super Tenant
 - Tenant with ability to access/manage data of any tenant

Tenant Types

- A tenant is a separate organizational entity within a multi-tenant database with....
 - It's own private data segment for each multi-tenant table
 - Except for groups and Super-tenants
 - One or more ABL security domains
 - Its own users
- Each multi-tenant database user belongs to some domain and by default some type of tenant
 - Default tenant
 - Regular tenant
 - Super tenant

Definitions

■ Table Instance

- Defined by a single table definition (schema)
- Each table instance contains
 - A data segment for the record data, a separate data segment for each index/lob of the table
- Each tenant/group has a different table instance
- A tenant need not instantiate every multi-tenant table

■ Data segment

- Tenant/group specific chain of data for each table/index/lob
- Stored/encapsulated in its own physical data partition

■ Physical data partition

- Managed storage for each object of a multi-tenant table instance
- Does not span storage areas

Users, Domains and Tenants

- User logging in with no domain association
 - Belongs to the blank domain, normally has access to default tenant
- User logging in as member of a domain, non blank, non super tenant
 - Has access as a regular type of tenant
- User logging in as a member of a domain associated with super tenant
 - Not a normal tenant user because they have no data segments of their own, but can get temporary access to regular tenant data

Table Types

- Non multi-tenant tables (shared)
 - Are tables in a non-multi tenant database, or tables in a multi-tenant database that are not multi-tenant, such as:
 - In the Sports DB, State Table (AK, AZ, etc)
 - Temp-tables
 - Schema tables
 - Can be accessed by users of any type of tenant subject to normal access privileges
- Multi-tenant tables
 - Have been made multi-tenant in a multi-tenant database
 - Are in a single private data segment for each regular tenant
 - Have a default data segment for the default tenant

User Access to Tenants

- Default tenant users
 - Cannot access regular tenant data
- Regular tenant users
 - Can access the private data segments of multi-tenant tables owned by that tenant
 - Access is subject to the user's normal access rights
 - Cannot access the private segments of any other regular tenants
- Super tenant users
 - Cannot access regular tenant data unless the super tenant user uses ABL language elements
 - SET-EFFECTIVE-TENANT and TENANT-WHERE allow access to regular tenant data
 - Access is still subject to the super tenant users normal access rights

Restrictions

- Must be Type II storage area
- No table partitioning of multi-tenant tables
- The domain is needed for tenants in in order to provide multi-domain security with Client-Principal support
 - The tenant feature security is based on and extends ABL version 10 security support

Thank You!

Libor Laubacher

 *llaubach@progress.com*

Open to Your Feedback. And, on the **Edge** of Our Seats.

Join the Progress Customer Validation Program today!

- **Roadmap Surveys**—Short surveys on product direction and enhancement prioritizes
- **Virtual Open Houses**—Live, interactive webinars sharing new development and architectural runways
- **Usability Surveys**—Show us how you actually use the product
- **Sprint Reviews**—Ongoing showcases that highlight development progress
- **Pre-release software** —Preview, test and share your feedback

Visit community.progress.com and search:
OpenEdge CVP



EMEAPUG[★] CHALLENGE

Enter our **raffle** to win
Apple AirPods



<https://prgress.co/PUGChallenge>



