

# Table Partitioning

So you want to get Horizontal?

Brian Bowman  
OpenEdge Product Management



# Agenda

---

- Setting the Groundwork
- Talking about “the truth”
- How can you help yourself?
- How can we help you?

# What problem are we trying to solve?

---

We all need to get more out of our database and application.

The problem is we are not given much to improve with.

So we continue to struggle to move forward and find tools to help...

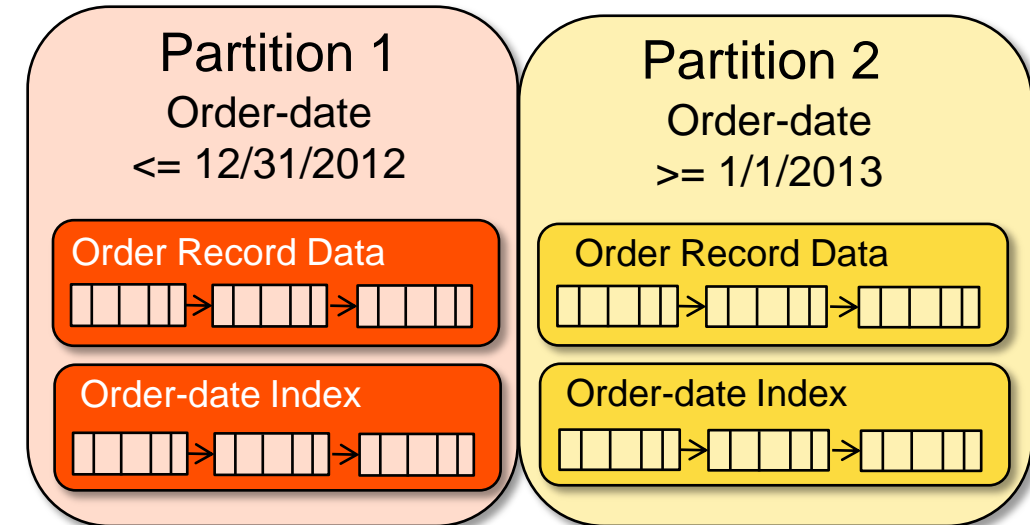
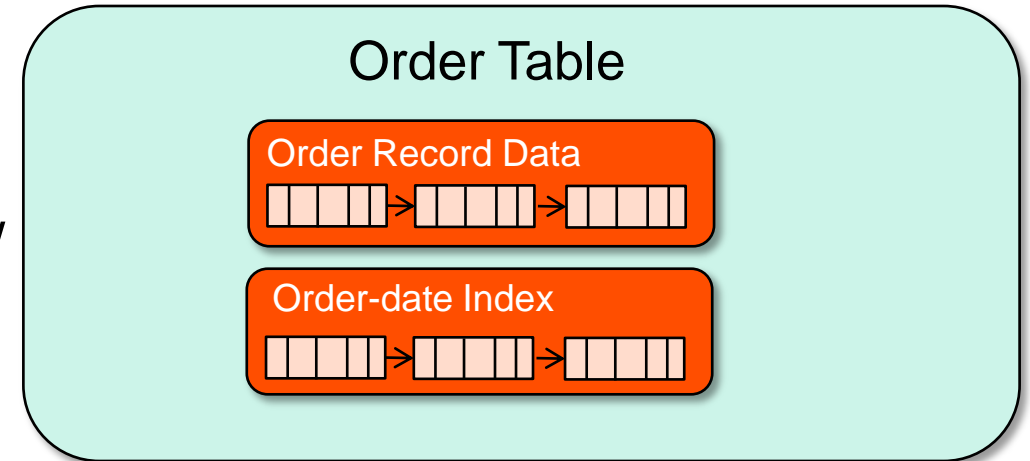
“Treading water is the same thing as  
drowning for people like you and me.”

*Frank Underwood (House of Cards)*



# What is Table Partitioning?

- Traditional Table Storage
  - Table stored in a single place
  - Queries can read the whole table which can slow down performance
  - Maintenance hard to perform
- Partitioned Table Storage
  - Tables split into multiple, self-contained locations
  - Queries only access what they need
  - Maintenance can be on a partition basis



# Table Partitioning: Why do I need it?

---

- Improve Performance

- Multiple Caveats Apply: YMMV, YMMSV, Read my lips..., The check's in the mail, How long is that piece of string?, It all boils down to size (of your database). It all depends., I ran out of gas., I had a flat tire., I didn't have enough money for cab fare., My tux didn't come back from the cleaners., An old friend came in from out of town., Someone stole my car., There was an earthquake., A terrible flood., Locusts!, All references to anyone or anything live or dead are purely coincidental., caveat, caveat actor, cave carum, bonitas non est pessimis esse meliorem
- Limiting data access for efficiency
- Improving data access performance
- Utilizes pruning for Queries
- Example: Running a report that is for last year's orders



*Important Note: You will see more opportunity for better performance with A) larger databases and B) more users!*

# Table Partitioning: Why do I need it (2)?

---

- Maintenance Advantages
  - Working only with what is important
  - Database utilities can run faster
  - Scanning, Tuning
  - Example: Archiving Solution
  - For example: Only backing up current data



*Important Note: Seeing Maintenance Advantages happens with most database sizes. It all depends on what you are doing now and what you want to be better.*

# Table Partitioning: Why do I need it (2A)?

---

- So – what can you do with a partitioned table?
  - Add, Split, Delete, Merge, Truncate, Table Move, Index Move, Index Check, Index Rebuild, Index Fix, Index Compact, dbtool, Binary Dump & Load, SQL Dump & Load\*
  - In-Place Migration
  - Automatic Partition Pruning
  - Partition level configuration – Create & Toss Limits, Alternate Buffer Pool
  - \*analysis Tools – Partition aware

*Are these all online?*

# Table Partitioning: Why do I need it (3)?

---

- Availability
  - Database available during partition processing
    - List processes that can be taken advantage of
  - Improve processing
    - Partition isolation allows for “different” things to happen at the same time.
  - Rebuild Indexes at the partition level
    - Partition is “taken offline” for things like Index Rebuild



*This is all about concurrency of operations and processing.*





“It's supposed to be hard. If it wasn't hard, everyone would do it. The "hard" is what makes it great.”

*Tom Hanks – A League of their Own*

# Table Partitioning: Three Types

---

## *List*

List Partition by  
Region

*Order Table*

*Western  
Region*

*Northern  
Region*

*Southern  
Region*

# Table Partitioning: Three Types

---

*List, Range*

List Partition by  
Region

Range Partition  
by Order Date

*Order Table*

*Western  
Region*

*Northern  
Region*

*Southern  
Region*

*Order Table*

*2010*

*2011*

*2012*

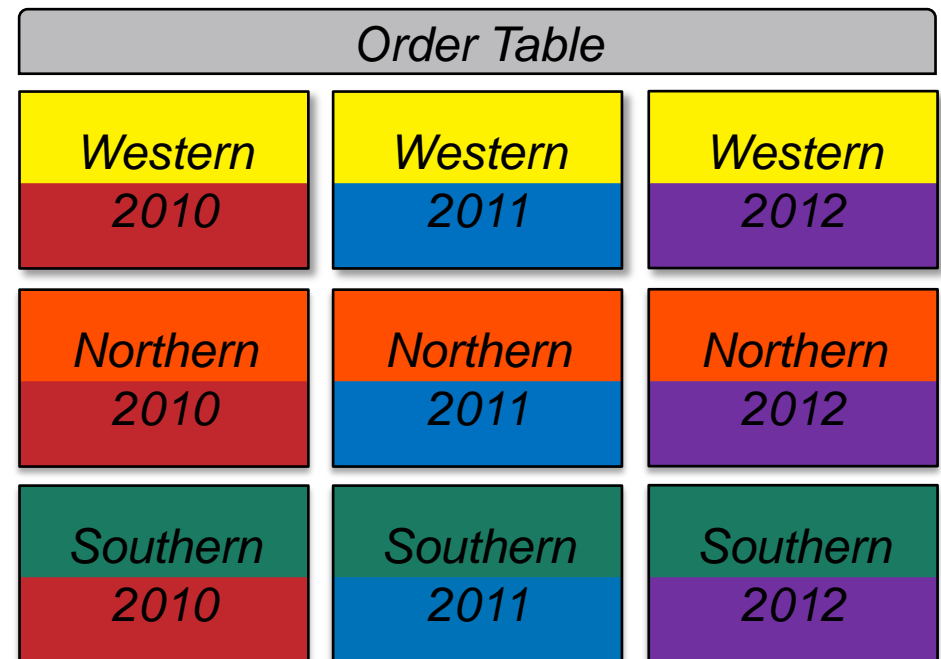
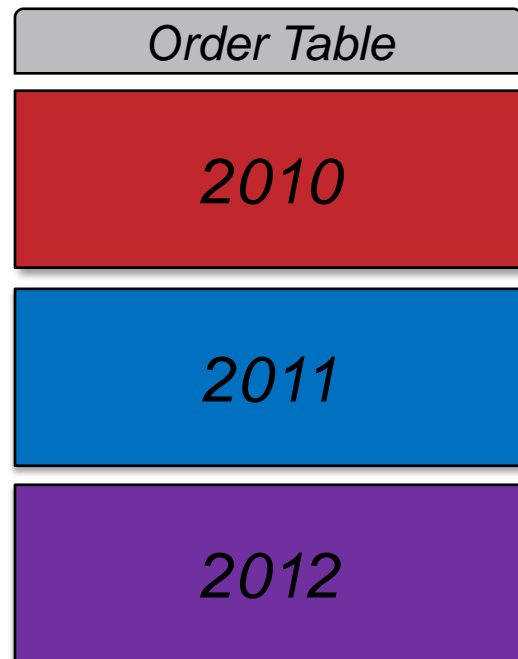
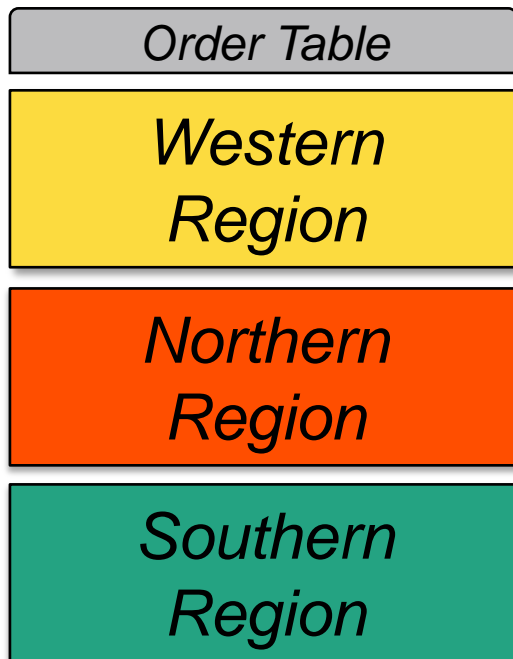
# Table Partitioning: Three Types

*List, Range and Sub-Partition*

List Partition by  
Region

Range Partition  
by Order Date

Sub-partition by  
Region and Order Date



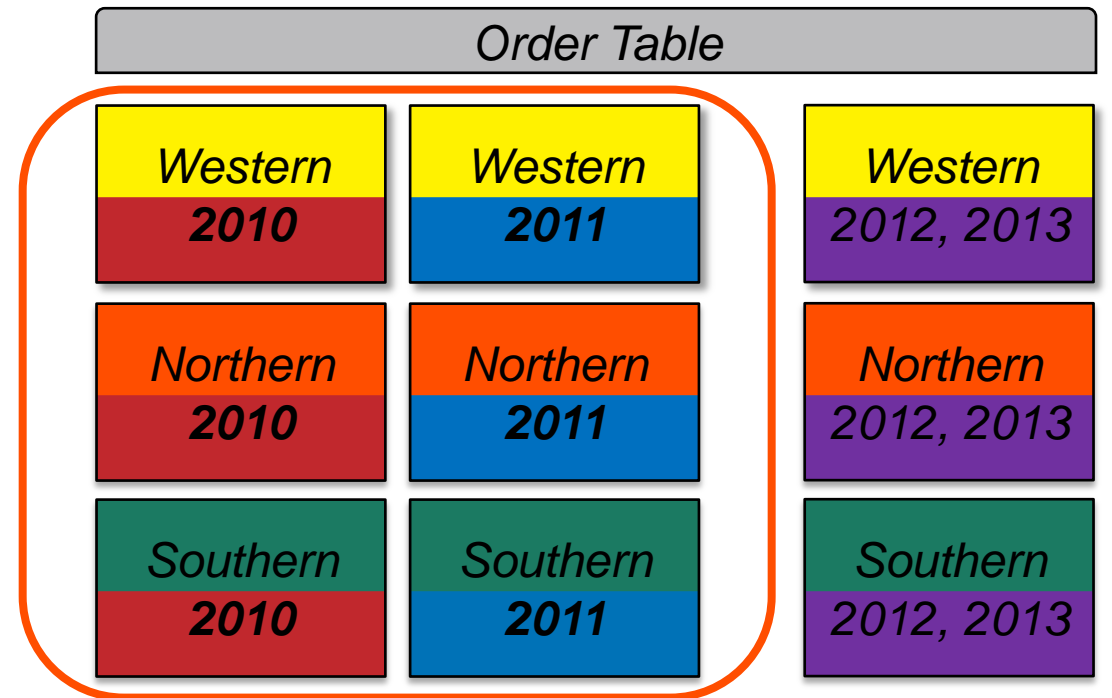
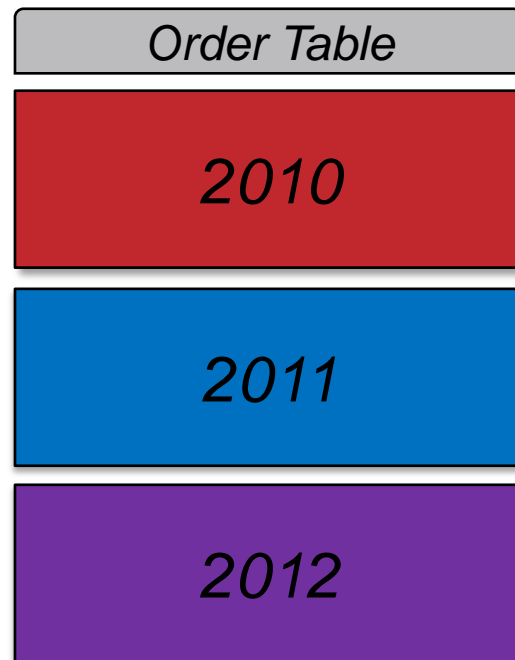
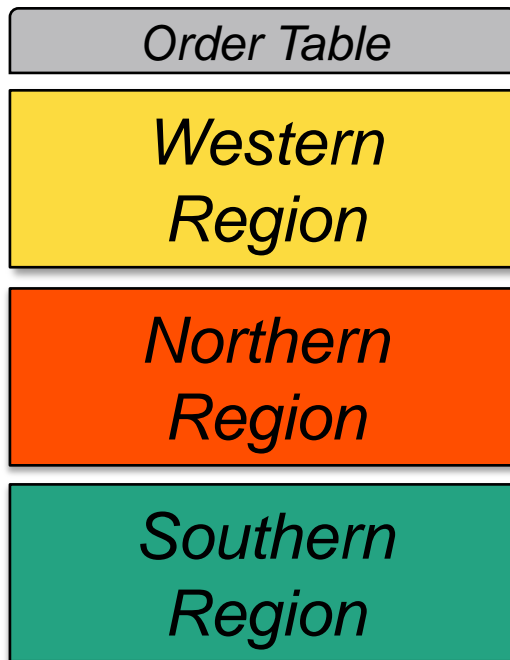
# Table Partitioning: Three Types (mostly)

*List, Range and Sub-Partition, Composite*

List Partition by  
Region

Range Partition  
by Order Date

Sub-partition by  
~~Region and Order Date~~  
List Range



# Different Partition Types - Details

---

- List Partition
  - Grouped based on a field or group of fields
  - Business can segment and report by region for efficiency
  - Example: Reporting for a specific country only
- Range Partition (by Date)
  - Grouped based on a range of data
  - Business can archive and make historical data read-only
  - Example: Archiving every year but the current year
- Sub-partitioning (By Region by Date)
  - Group based on a combination of the above choices
  - Application can further isolate specific data access
  - Example: Running a report for orders for Canada for FY 2012
- Composite – for migration only



# Data and Index Differences...

---

- Data

- Stored in separate areas based on partition definitions (usually but not “required”)

- Indexes

- Local index support
  - One index b-tree per partition
- Global index support
  - One index b-tree containing all the table’s data
  - Ability to index across partitions
  - Provides uniqueness support for non-partitioned columns
- Composite index support
  - One index b-tree containing entries for multiple partition definitions
  - Table data for composite partition stored in same partition

# Table Partitioning – Why it might go wrong?

---

- Partition for the wrong reason
- Partitioning on the wrong column (ie. Update Date, ROWID)
- Incorrect Indexing – Leading components
- Corruption
- Need to Re-partition?



# Is it Right for You?

---

- Do you have a “large” or bigger database?
- Do you want to get better performance from some larger, specific queries?
- Do you have Type II Storage Areas?

# Table Partitioning Summary

---

## *Partition Options*

- Types
  - List Partitioning
  - Range Partitioning
  - Sub-partitioning
  - Composite Partitioning

## *Ease of Implementation*

- Migration
  - In place migration
  - No application changes
  - List and range migration

## *Indexing Support*

- Global Index
- Local Index (aligned)
- Composite
- Automatic pruning

## *New Maintenance*

- Truncate / Delete
- View / Rename
- Split / Merge

# The Real Summary

---

- What problem are you trying to solve?

We all need to get more out of our database and application.

The problem is we are not given much to improve with.

So we continue to struggle to move forward and find tools to help...

- Can Table Partitioning help solve the problem?

Increase Maintainability

Increase Availability

Increase Performance

- Action Items (for you):

1. Go to Rich Banville's talk on Table Partitioning – This room - 3:30 PM

2. Join the ESAP program and work with Table Partitioning – cost is \$0

3. Read the whitepaper...

# Table Partitioning - Early Software Access Program

---

Objective: Demonstrate and test table partition functionality

- Capabilities\*

- Partition creation & maintenance
- Migration examples

*\*Subject to feature availability at the time*

- White papers and examples
- Special presentations and webinars
- Looking for customers to potentially
  - Benchmark HTP
  - Speak at Exchange
  - Be a reference

- Participation

- Access by enrollment only
- For enrollment & more information
  - Contact: Brian Bowman  
[bowman@progress.com](mailto:bowman@progress.com)
  - Send email to ESAP enrollment  
[early-software-admin@progress.com](mailto:early-software-admin@progress.com)

# Additional Resources

---

- Case Study # 1 – MIP Holding
  - <http://www.progress.com/customers/mip-holdings>
- Case Study # 2 – Qvantel
  - <http://www.progress.com/customers/qvantel>
- Whitepaper on Choosing the Right Table for Table Partitioning
  - <https://community.progress.com/technicalusers/m/technicalusersgallery/169.aspx>
- ESAP Forum available on community for customers who join the ESAP program



**PROGRESS**