Edition Based Redefinition

Zero Downtime Application Upgrades

presentation for:

Vancouver Oracle Users Group
March 2010

Daniel A. Morgan | damorgan11g@gmail.com | www.morganslibrary.org

Introduction

- Daniel Morgan – damorgan11g@gmail.com
- Oracle Ace Director
- University of Washington, retired
- The Morgan of Morgan's Library on the web
  - www.morganslibrary.org/library.html
- Board of Directors: Western Washington Oracle Users Group
- Member: UK Oracle Users Group
- Former Member: Oracle Applications Users Group
- Frequent speaker . . . . . .

- Oracle since version 6
- 11g beta test site

Morgan's Library

UNIVERSITY OF WASHINGTON

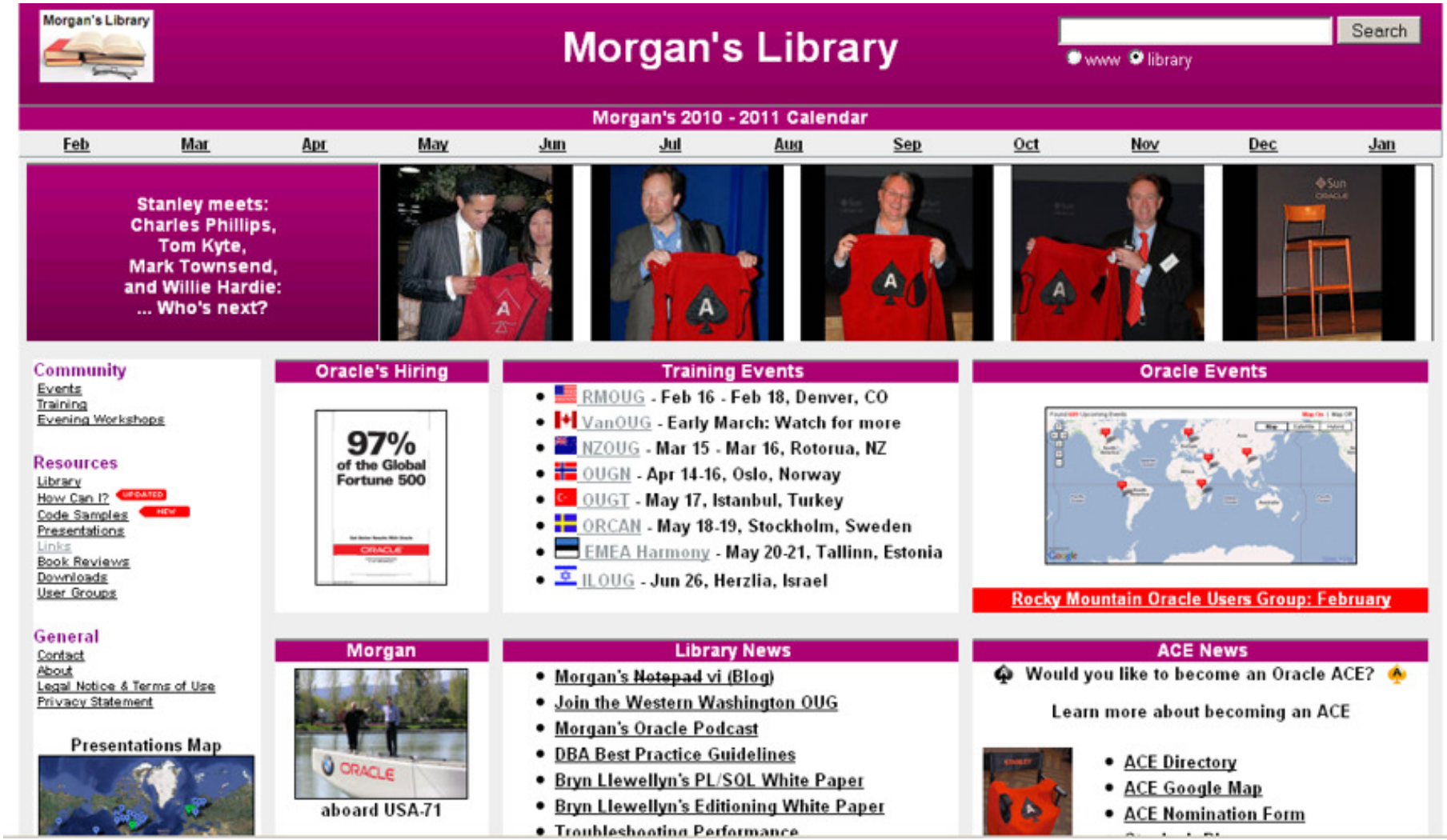
WWOUG

UKOUG

affiliate of OAUG oracle applications users group

Official Beta Site ORACLE DATABASE 11g

Morgan's Library: www.morganslibrary.org



America's Cup Boat USA-71

EBR Basics

We all have our favorite customers: This is mine . . . on a good day

Store More Data

Maintain Performance

Honor the same Service Level Agreement

What's the big deal?

Why Should We Care?

- High availability
  - Amazon and Google are up 7 x 24 x 365
  - Our customers expect the same from us
- Data Center Failure
  - Data Guard
- Server Failure
  - Real Application Clusters
- Storage Failure
  - ASM
  - RAID
  - Resumable Transactions
  - RMAN
- Network Failure
  - Multiplexing and Bonding

Why Should We Care?

- Human Failure
  - Flashback Database
  - Flashback Drop
  - Flashback Transaction
  - Log Miner
  - RMAN
  - Streams
  - Transaction Backout
- Oracle Upgrade
  - Rolling Patches
- Application Upgrade and Maintenance
  - Tables: DBMS_REDEFINITION
  - PL/SQL Objects: Without Editioning ... downtime is unavoidable

Why Do We Need EBR?

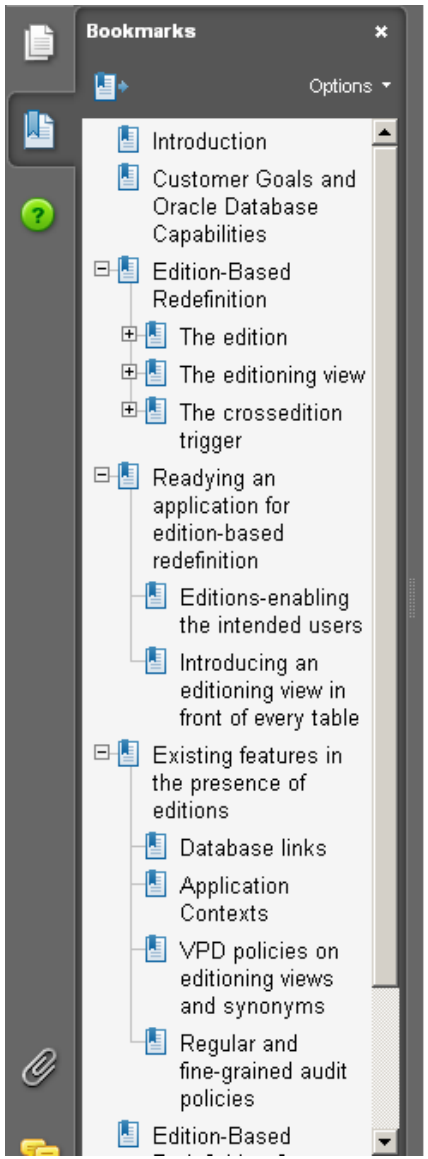
- Application upgrades need to:
  - Not perturb users
  - Not corrupt data
  - Reflect all pre-upgrade transactions
  - Seamlessly roll changes forward or backward

  - Be safe
  - Be secure
  - Be fully supported by Oracle
  - Be free
- Because you need to keep your brain agile
- You are starting to have delusions of competence

Delusions of Competence Quiz

- Can you attach a table trigger to a view?
- Can two different objects exist in the same schema with the same name?
- Can you real-time replace a PL/SQL object that is in-use without downtime?
- Can views be created with a WHERE clause?
- If you have two triggers on the same object can you force one to fire before the other?
- Can your database have an object without an owner?
- Do you know what is stored in DBA_OBJECTS_AE?
- Do you know how to actualize a stored procedure?
- What does it mean to grant USE to a schema?

Bryn Llewellyn's White Paper



An Oracle White Paper
July 2009

Edition-Based Redefinition

a new capability in Oracle Database 11g Release 2
to support online application upgrade

What is EBR?

- A revolutionary new capability
  - Code changes are installed in the privacy of an edition
- Editionable object types
  - Functions (including Pipelined)
  - Libraries
  - Packages
  - Stored Procedures
  - Synonyms
  - Triggers
  - Views
- New object types
  - Edition
  - Editioning View
  - Crossedition Trigger

Three New Object Types

- Edition
  - All pre-upgrade editionable objects are part of a parent edition
  - New editions inherit (by pointer) editionable objects from the parent edition
  - All post-edition editionable objects are part of a child edition
- Editioning View
  - Exposes a different projection of a table into each edition
  - Allows each edition to see only its own columns
  - Data changes are made safely by writing only to new columns or new tables not seen by the old edition
  - Allows different "table" triggers to fire in each edition
- Crossedition Trigger
  - Propagates data changes made by the parent edition into the child edition's columns, or (in hot-rollover) *vice-versa*.

What is an Edition?

- A nonschema object, uniquely, identified by only its name
- Like another non-schema object, the directory, is listed in DBA_OBJECTS as owned by SYS but has no owner
- Every database from 11.2 onwards, whether brand new or the result of an upgrade from an earlier version, non-negotiably, has at least one edition
- The default edition name is ORA$BASE
- Every foreground database session, at every moment throughout its lifetime, non-negotiably, uses a single edition
- A new edition must be the child of an existing edition
- A child edition is all that is required if an upgrade involves only synonyms, views, and PL/SQL objects

Edition Privileges

- System Privileges
  - CREATE ANY EDITION
  - ALTER ANY EDITION
  - DROP ANY EDITION
- Object Privileges
  - USE
- Roles
  - All three system privileges are granted to the DBA role (only)
  - USE is not, by default, granted to user or role
- Enable Editioning
  - **`ALTER USER <user_name> ENABLE EDITIONS;`**
  - **`ALTER SESSION SET EDITION = <edition_name>;`**

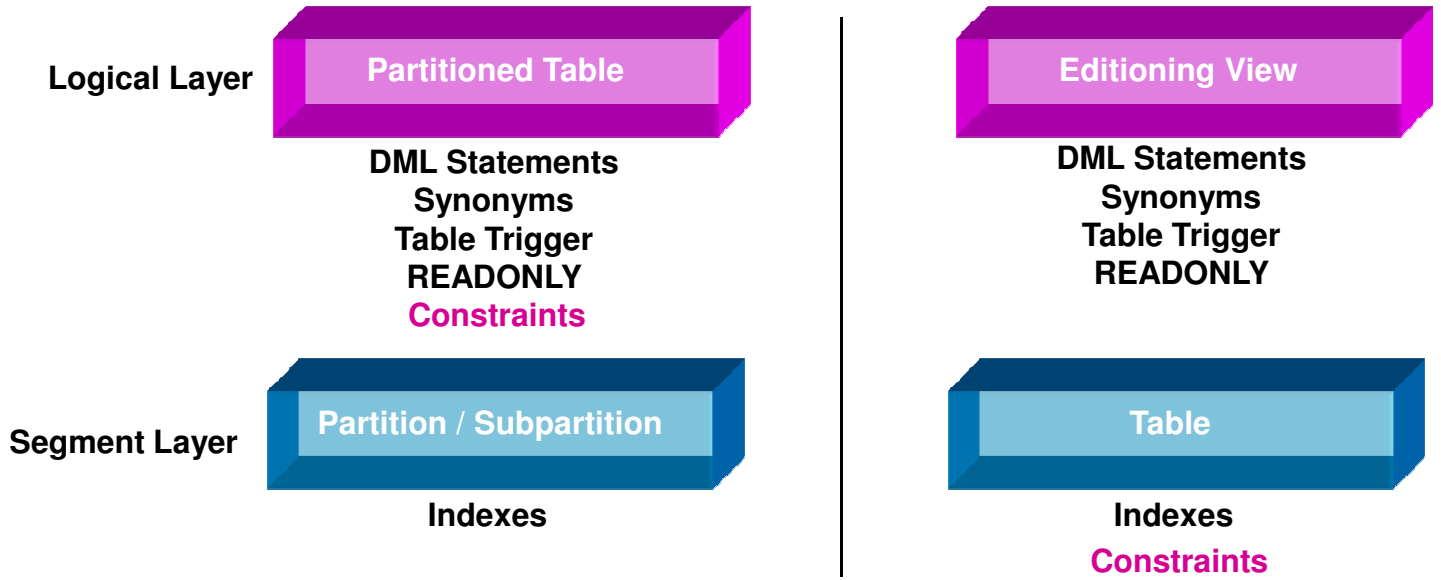
Edition Related Data Dictionary Views

- AUD$ (obj$edition)
- DBA_EDITIONS (edition_name, parent_edition_name)
- DBA_OBJECTS (edition_name)
- DBA_OBJECTS_AE (edition_name)
- DBA_SOURCE_AE (edition_name)
- DBA_USERS (editions enabled)
- FGA_LOG$ (obj$edition)
- UTL_RECOMP_ALL_OBJECTS (edition_name)
- V$LOGMNR_CONTENTS (edition_name)
- V$SESSION (session_edition_id)

AE = All Editions

What is an Editioning View?

- A view that you may think of a partitioned table that can only have a single partition
  - Both must present all data "as is" ... no filters, no joins, no functions, no operators, no group by no having clause no order by no distinct no concatenation no ... no ... no ... no ....
  - Your only choice is which columns to select



If you can not do it in partitioning a table you can not do it in an editioning view

Editioning View Related Data Dictionary Views

- DBA_EDITIONING_VIEW_COLS
- DBA_EDITIONING_VIEW_COLS_AE
- DBA_EDITIONING_VIEWS
- DBA_EDITIONING_VIEWS_AE
- DBA_ERRORS_AE (editioning_name)
- DBA_OBJECTS_AE (editioning_name)
- DBA_VIEWS (editioning_view)

AE = All Editions

What is a Crossedition Trigger?

- A new, and special type of trigger specific to editioning
- Distinct from application code
- Can only be created on a table (not on an editioning view)
- Populates pre-upgrade transactions into the post-upgrade edition (or)
- Populates post-upgrade transactions into the pre-upgrade edition
- Two types
  - FORWARD
  - REVERSE
- Leverage trigger firing order control with [FOLLOWING and PRECEDING

Crossedition Trigger Firing Rules

- Assumptions
  - All DDL is performed in the child edition so as not to disturb the working production application
  - All DDL to editioned objects is done in the post-upgrade edition
  - Pre-upgrade column changes are only changed in the parent
  - Post-upgrade columns are only changed in the child
- Forward Crossedition Triggers
  - Should only be fired by code running in the parent edition
  - Transforms from the old representation to the new
- Reverse Crossedition Triggers
  - Should only be fired by code running in the child edition
  - Transforms from the new representation to the old

Crossedition Trigger Related Data Dictionary Views

- DBA_TRIGGERS
- DBA_TRIGGER_ORDERING
- DBA_ERRORS_AE (editioning_name)
- DBA_OBJECTS_AE (editioning_name)

AE = All Editions

DBMS_OUTPUT and Crossedition Triggers

It is typically not possible to trace the behavior of a crossedition trigger using DBMS_OUTPUT.PUT_LINE. This is because the procedure accumulates the lines in a DBMS_OUTPUT package global collection so that, when the server call terminates, SQL*Plus can traverse the collection to print out the lines. However, as has been explained (see "Package state when the same package is instantiated in more than one edition" on page 18), when a session uses different editions during its lifetime, then a particular package is separately instantiated in each edition from which a reference to the package is made. It is for this reason that the more cumbersome approach, using UTL_FILE, is used. This method of tracing, using UTL_FILE to open the trace file in append mode, write one line, and then to close the file is very inefficient. However, in a test such as this, the inefficiency is undetectable.

Page 27: #57

Other Editioning Related PL/SQL Objects

- DBMS_EDITIONS_UTILITIES
  - SET_EDITIONING_VIEWS_READ_ONLY
- DBMS_METADATA_UTIL.GET_EDITIONID
- DBMS_PARALLEL_EXECUTE.RESUME_TASK
- DBMS_PARALLEL_EXECUTE.RUN_TASK
- DBMS_SESSION.SET_EDITION_DEFERRED
- DBMS_SQL.PARSE
- DBMS_UTILITY.VALIDATE
- SYS_CONTEXT Function

Other Editioning Related Database Capabilities

- Invisible Indexes
- Fine Grained Dependency Tracking

EBR for DBAs

Editioning for DBAs: Special Considerations

- All foreground processes use an edition
- Background processes that issue SQL statements, such as MMON, are tied to an edition
- Thus when dropping and edition make sure that it is not the default edition for sufficient time for MMON and other SQL issuing process to change to the new default (else you will generate an ORA-38805: edition is in use)
- Once a schema is edition enabled there is no disable
- When retiring the pre-upgrade edition revoke USE
- Dropping parent editions can be done for elegance but is not required

Health Warning

Due to complaints made to Minister of Health Leona Aglukkaq at Health Canada ...

You are now entering ...

... a ...

Questions

ERROR at line 1:

ORA-00028: your session has been killed

All demos at morganslibrary.org

- **Library**
- **How Can I?**

damorgan11g@gmail.com

Thank you

Editioning to English Dictionary

- Actualize
  - An inherited object compiled or created in the child edition. The inheritance link is broken
- Child Edition
  - A new edition that inherits the editionable objects from the previously existing "parent" edition
- Crossedition Trigger
  - A trigger that propagates transactions between editions
- Edition
  - A non-schema logical object
- Editionable Object
  - An object that is editionable in the current database version
- Editioning View
  - A new kind of view that acts much like a partitioned table

Editioning to English Dictionary

- Leaf Edition
  - Same as child edition
- Parent Edition
  - A trigger that propagates transactions between editions