Edition Based Redefinition

Zero Downtime Application Upgrades

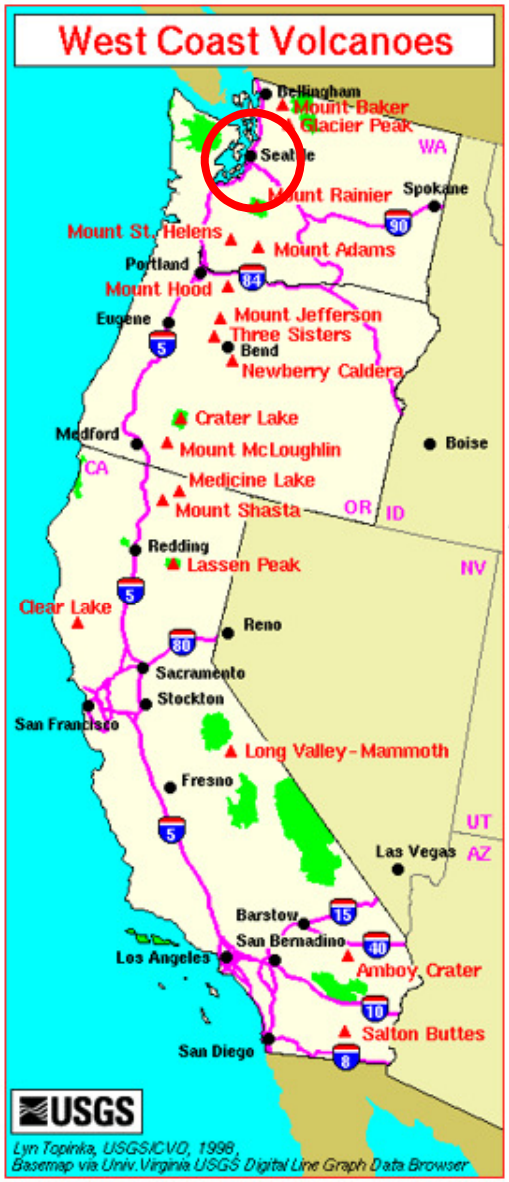
NZOUG Webinar
July 15, 2010

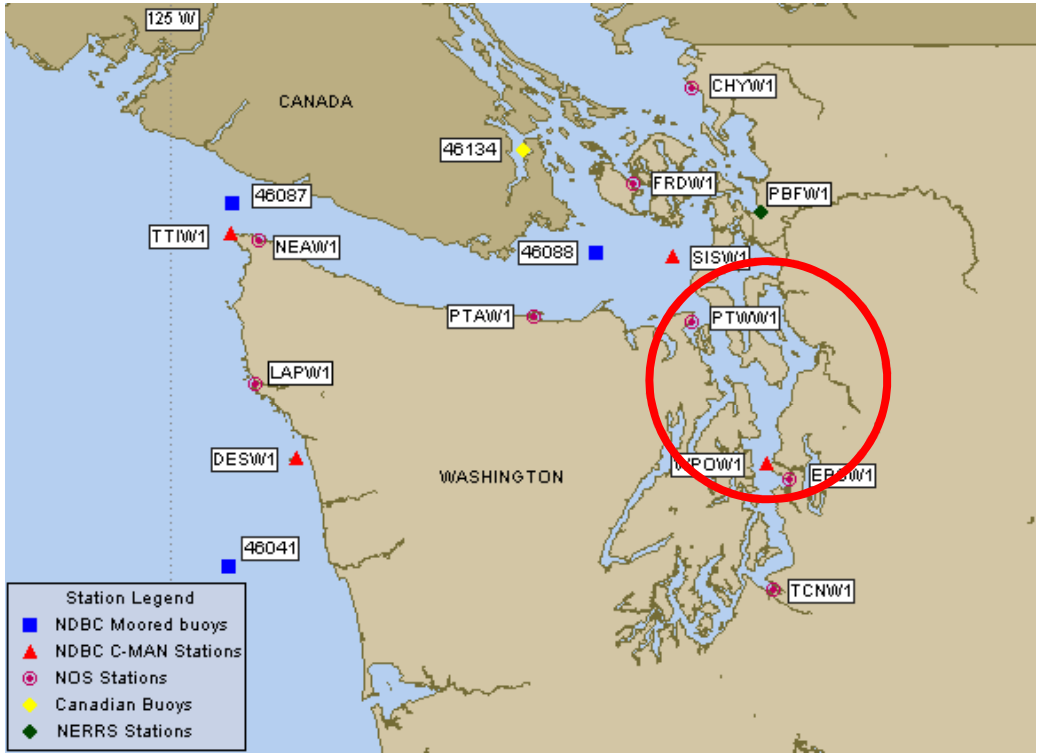
Daniel A. Morgan

- Oracle ACE Director
- University of Washington Oracle Instructor for 10 years
- Morgan of Morgan's Library on the web
  - www.morganslibrary.org
- Member UKOUG and Oracle Applications User Group
- Conference Speaker
  - OpenWorld, Collaborate, Kaleidoscope, Canada, Chile, Denmark, Estonia, Finland, Germany, Japan, New Zealand, Norway, Sweden, UK & US
- Beta tester for Oracle and TimesTen Databases

cd $MORGAN_HOME

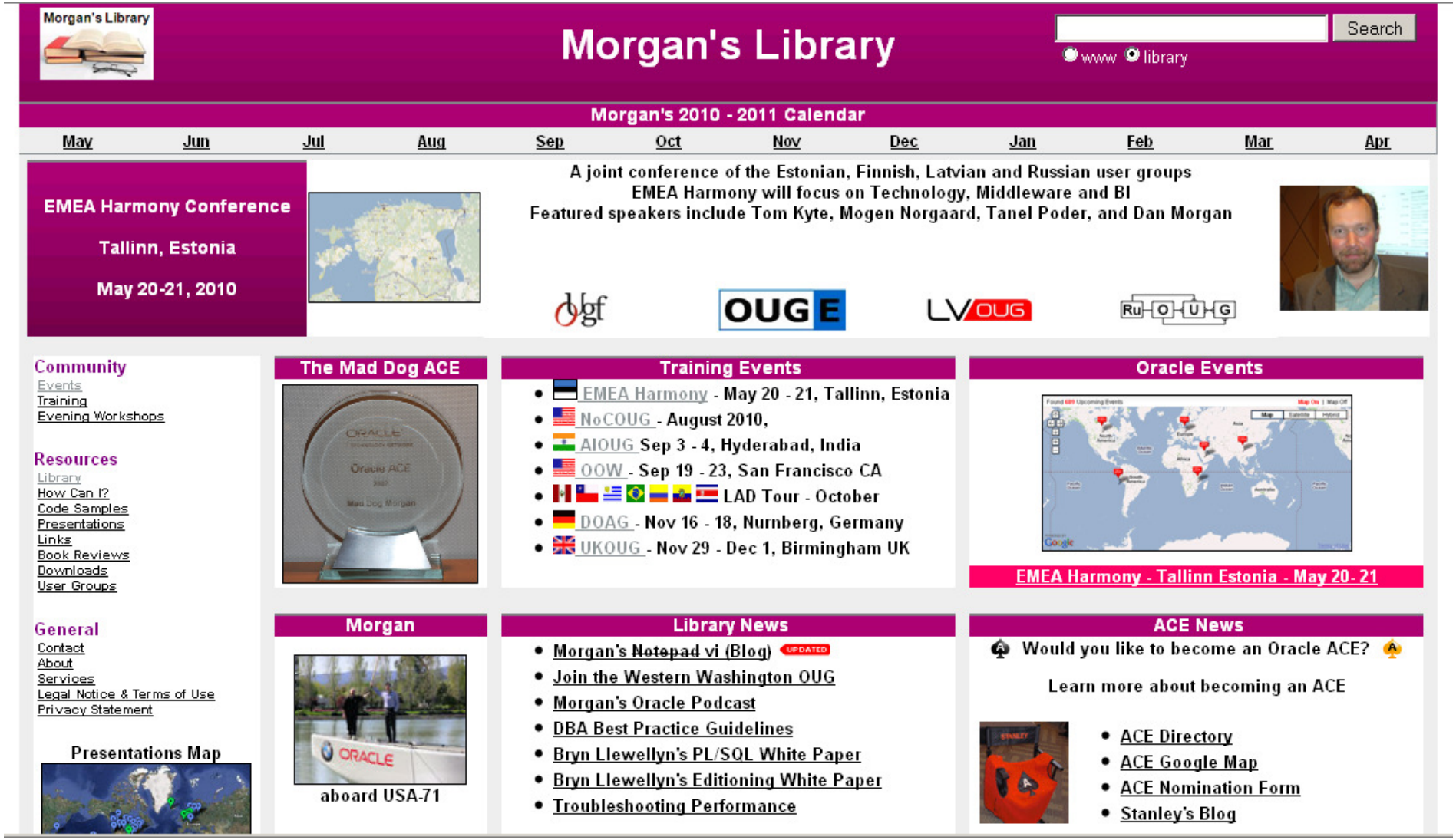
cd $MORGAN_HOME





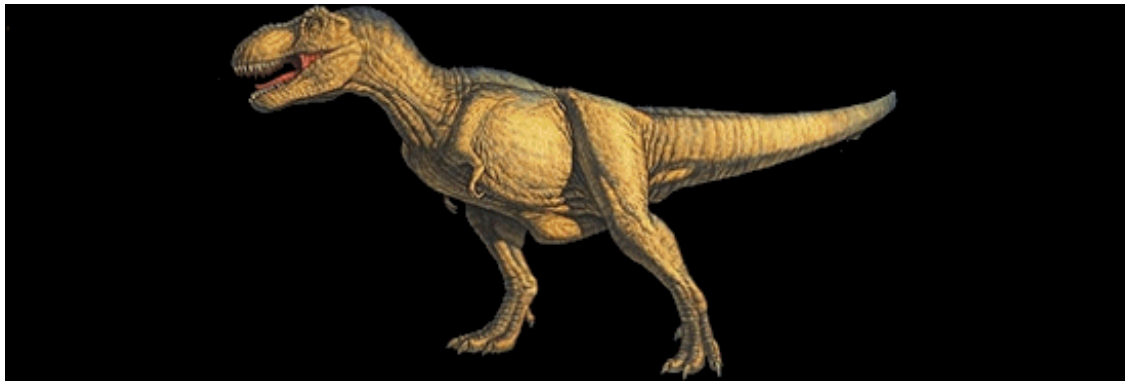
cd $MORGAN_HOME

Morgan's Library: www.morganslibrary.org



Mythology & Dinosaurs

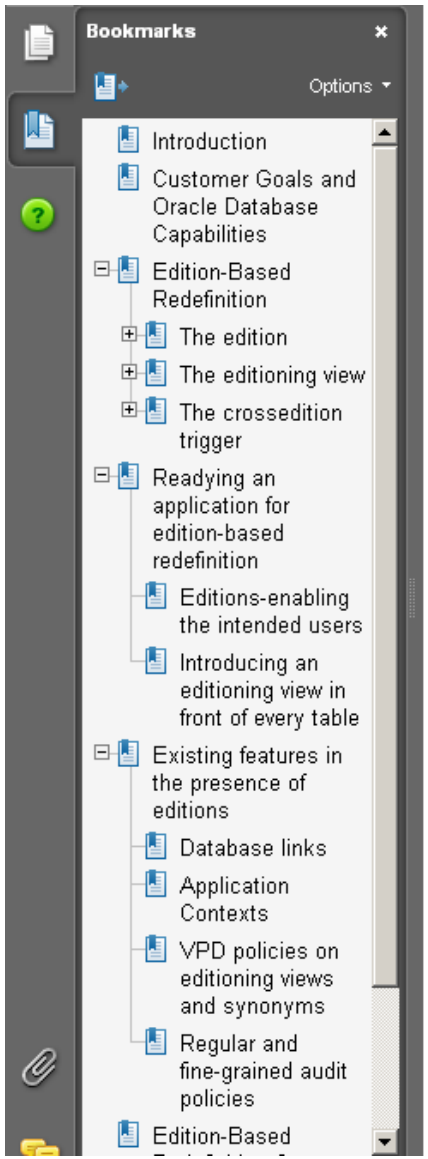
- Most Oracle DBAs and Developers are using Oracle 10g as though it was 7.3.4



Self-Image

Reality

Bryn Llewellyn's White Paper

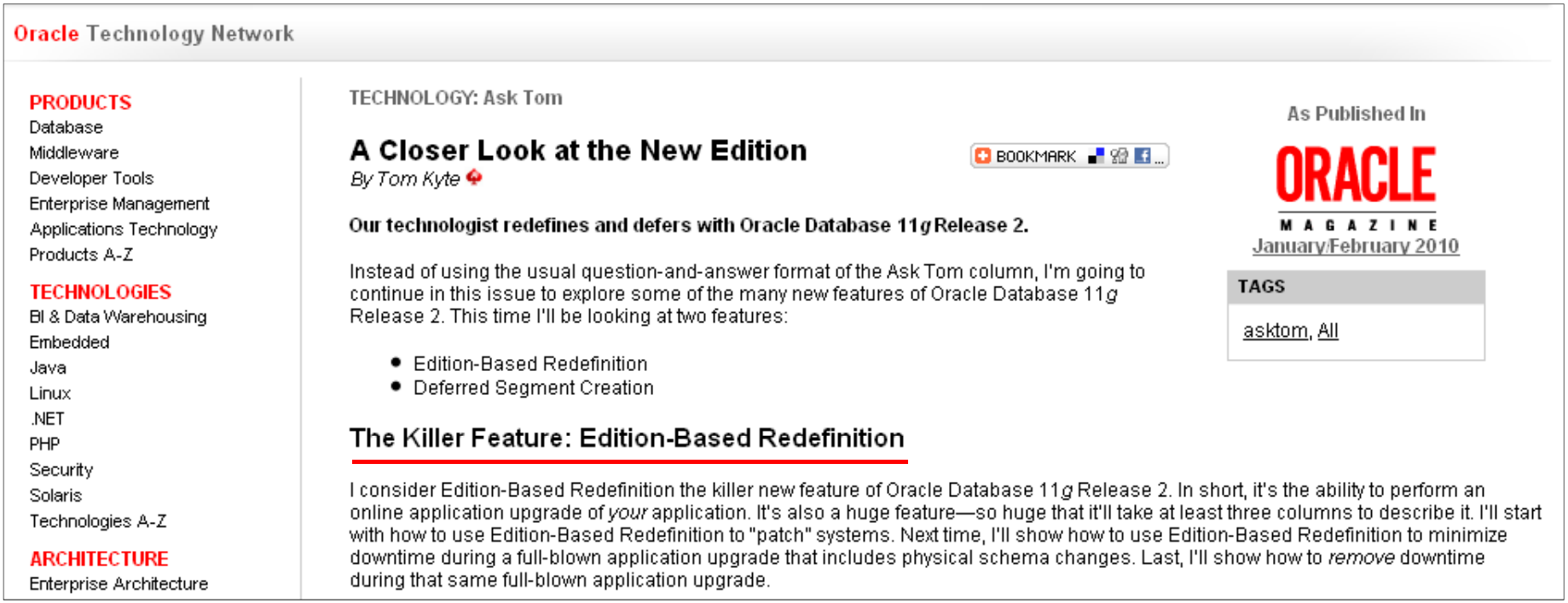


An Oracle White Paper
July 2009

Edition-Based Redefinition

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

Tom Kyte on Edition Based Redefinition



Oracle Technology Network

PRODUCTS
Database
Middleware
Developer Tools
Enterprise Management
Applications Technology
Products A-Z

TECHNOLOGIES
BI & Data Warehousing
Embedded
Java
Linux
.NET
PHP
Security
Solaris
Technologies A-Z

ARCHITECTURE
Enterprise Architecture

TECHNOLOGY: Ask Tom

A Closer Look at the New Edition

BOOKMARK

By Tom Kyte

Our technologist redefines and defers with Oracle Database 11*g* Release 2.

Instead of using the usual question-and-answer format of the Ask Tom column, I'm going to continue in this issue to explore some of the many new features of Oracle Database 11*g* Release 2. This time I'll be looking at two features:

- Edition-Based Redefinition
- Deferred Segment Creation

The Killer Feature: Edition-Based Redefinition

I consider Edition-Based Redefinition the killer new feature of Oracle Database 11*g* Release 2. In short, it's the ability to perform an online application upgrade of *your* application. It's also a huge feature—so huge that it'll take at least three columns to describe it. I'll start with how to use Edition-Based Redefinition to "patch" systems. Next time, I'll show how to use Edition-Based Redefinition to minimize downtime during a full-blown application upgrade that includes physical schema changes. Last, I'll show how to *remove* downtime during that same full-blown application upgrade.

As Published In
ORACLE MAGAZINE
January/February 2010

TAGS
asktom, All

OpenWorld 2010

RE: Oracle ACE Director speaking slots at OpenWorld Inbox X

Victoria Lira to me show details May 6 (3 days ago) Reply

Hi Dan-
I wanted to let you know that your submitted paper:
S313426 Edition-Based Redefinition: Live in SQL*Plus

Has been accepted as an ACE paper. You will receive an official OpenWorld paper acceptance email with detailed instructions – to be sent out beginning May 18. Just thought you might like to know a little early.

Congratulations!
-Vikki

From: Daniel Morgan [mailto:damorgan11g@gmail.com]
Sent: Thursday, March 18, 2010 1:18 AM
To: Victoria Lira
Subject: Re: Oracle ACE Director speaking slots at OpenWorld

Session ID: S313426
Title: Edition-Based Redefinition: Live in SQL*Plus

Abstract:
Edition Based Redefinition is a new and critical component adding abilities that greatly enhance high availability in Oracle Database 11g Release 2. This revolutionary new capabilities that allow online application upgrade with uninterrupted availability using three new database objects: The Edition, the Editioning View, and Crossedition Triggers.

This session, with live a demonstration live in SQL*Plus, will show how each of the key components of editioning work and explain how they can be leveraged with single-instance stand-alone databases and with Real Application Clusters.

New editioning enhancements in version 11.2.0.2 will also be shown.

EBR Basics

Delusions of Competence Quiz

- Can you create a before insert table trigger on a view?
- Can two different objects exist in the same schema with the same name (other than package spec and body)?
- Can you real-time replace a PL/SQL object without down-time while it is being used?
- Can all views be created with a WHERE clause?
- If you have two triggers on the same object can you force one to always fire before, or after, the other?
- Can your database have an object without an owner?
- What is visible in DBA_OBJECTS_AE?
- Do you know how to actualize a stored procedure?
- What does it mean to grant USE to a schema?

Editioning to English Dictionary

- Actualize
  - An inherited object compiled or created in the child edition when the inheritance link is broken. This "bug" will be fixed in 12gR1.
- 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
  - The child edition after it becomes the default edition
- Parent Edition
  - The edition from which a child has been, in essence, cloned. Changes to the parent "should" not roll forward into the child.

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
  - Your customers expect the same from you
- Data Center Failure
  - Data Guard
- Server Failure
  - Real Application Clusters
- Storage Failure
  - ASM
  - RAID
  - Resumable Transactions
  - RMAN (recover from backups or standby database)
- Network Failure
  - VLANs, Multiplexing and Bonding

Why Should We Care?

- Human Failure
  - Flashback Database
  - Flashback Drop
  - Flashback Table
  - Flashback Transaction
  - Log Miner
  - RMAN
  - Transaction Backout
- Oracle Upgrade
  - Rolling Patches
- Application Upgrade and Maintenance
  - Tables: DBMS_REDEFINITION
    - but always tied to some code somewhere
  - 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 after upgrade
  - Seamlessly roll changes forward and backward

  - Be safe
  - Be secure
  - Be fully supported by Oracle
  - Be free (no extra licensing cost)

What is EBR?

- A revolutionary new capability
  - Code changes are installed in the privacy of an edition
- Editionable object types
  - PL/SQL objects of all kinds
  - Synonyms
  - Views
- Requires new kinds of object
  - Edition
    - dba_editions, dba_edition_comments
  - Editioning View
    - dba_editioning_views, dba_editioning_views_ae, dba_editioning_view_cols
  - Crossedition Trigger
    - dba_triggers

Three New Object Types

- Edition (only replacing PL/SQL, synonyms, and views)
  - 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 the child edition
- Editioning View (changing tables)
  - 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 (migrate data forward and backward)
  - 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 (not granted by default)
- Roles
  - All three system privileges are granted to the DBA role (only)
  - USE is not, by default, granted to any user or role
- Enable Editioning
  - **`ALTER USER <user_name> ENABLE EDITIONS;`**
  - **`ALTER SESSION SET EDITION = <edition_name>;`**

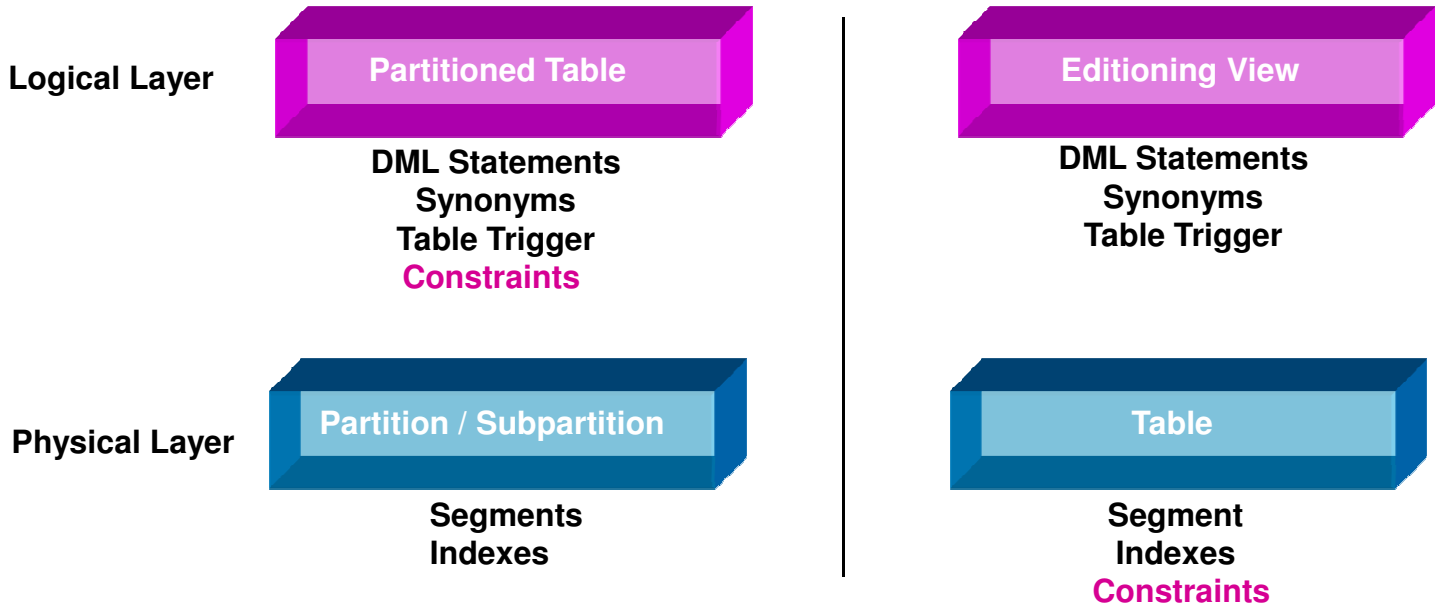
Edition Enabled Data Dictionary Views

- 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 no order by no distinct no concatenations: just no ... no ... no and no
  - Your only choice is which columns to select (project)



If you can not do it when 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

Edition View DDL

```
CREATE OR REPLACE EDITIONING VIEW person AS
SELECT *
FROM person_tab;
```

An editioning view can NEVER be more complex than this

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) post-upgrade transactions into the pre-upgrade edition
- Two types
  - FORWARD
  - REVERSE
- Control trigger firing order control with [FOLLOWING and PRECEDING] keywords
- Nothing we do should affect the current application so crossedition triggers are always created in the child

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
  - Only fired by code running in the parent edition
  - Transforms from the old representation to the new
- Reverse Crossedition Triggers
  - Only fired by code running in the child edition
  - Transforms from the new representation to the old

Crosseditioning Trigger DDL

```
CREATE OR REPLACE TRIGGER Contacts_Fwd_Xed
BEFORE INSERT OR UPDATE ON Contacts_Table
FOR EACH ROW
FORWARD CROSSEDITION
DISABLE
BEGIN
  Set_First_And_Last_Name(:NEW.Name_1, :NEW.First_Name_2, :NEW.Last_Name_2);
  Set_Country_Code_And_Phone_No(:NEW.Phone_Number_1, :NEW.Country_Code_2, :NEW.Phone_Number_2);
END Contacts_Fwd_Xed;
/
```

```
CREATE OR REPLACE TRIGGER Contacts_Rvrs_Xed
BEFORE INSERT OR UPDATE ON Contacts_Table
FOR EACH ROW
REVERSE CROSSEDITION
DISABLE
BEGIN
  :NEW.Name_1 := :NEW.Last_Name_2||', '||:NEW.First_Name_2;
  :NEW.Phone_Number_1 :=
     CASE :New.Country_Code_2 WHEN '+1' THEN
       REPLACE(:NEW.Phone_Number_2, '-', '.')
     ELSE
       '011.'||LTRIM(:NEW.Country_Code_2, '+')||'.'|| REPLACE(:NEW.Phone_Number_2, '-', '.')
     END;
END Contacts_Rvrs_Xed;
/
```

Crossedition Trigger Related Data Dictionary Views

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

AE = All Editions

Tracing Crossedition Triggers Footnote

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
- Invisible Indexes
- SYS_CONTEXT Function

Invisible Indexes

- A real index, invisible to the cost-base optimizer, for a default session

```
CREATE INDEX ix_mobile_net_lat
ON mobile_net_tab(latitude)
INVISIBLE;


CREATE OR REPLACE TRIGGER enable_invisible_indexes
AFTER LOGON ON SCHEMA
DECLARE
 parent_edition all_editions.edition_name%TYPE;
BEGIN
  SELECT parent_edition_name
  INTO parent_edition
  FROM all_editions
  WHERE edition_name = (
  SELECT sys_context('USERENV', 'CURRENT_EDITION_NAME') FROM dual);

  IF parent_edition IS NOT NULL THEN
    execute immediate 'ALTER SESSION SET "optimizer_use_invisible_indexes" = TRUE';
    dbms_output.put_line('enabled');
  END IF;
END enable_invisible_indexes;
/
```

EBR for DBAs

Editioning for DBAs: Special Considerations

- Seamless integration with Real Application Clusters
- Seamless integration with Physical Data Guard
- Seamless integration with TAF (transparent application failover)
- Seamless integration with FCF (fast connection failover)

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 an 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

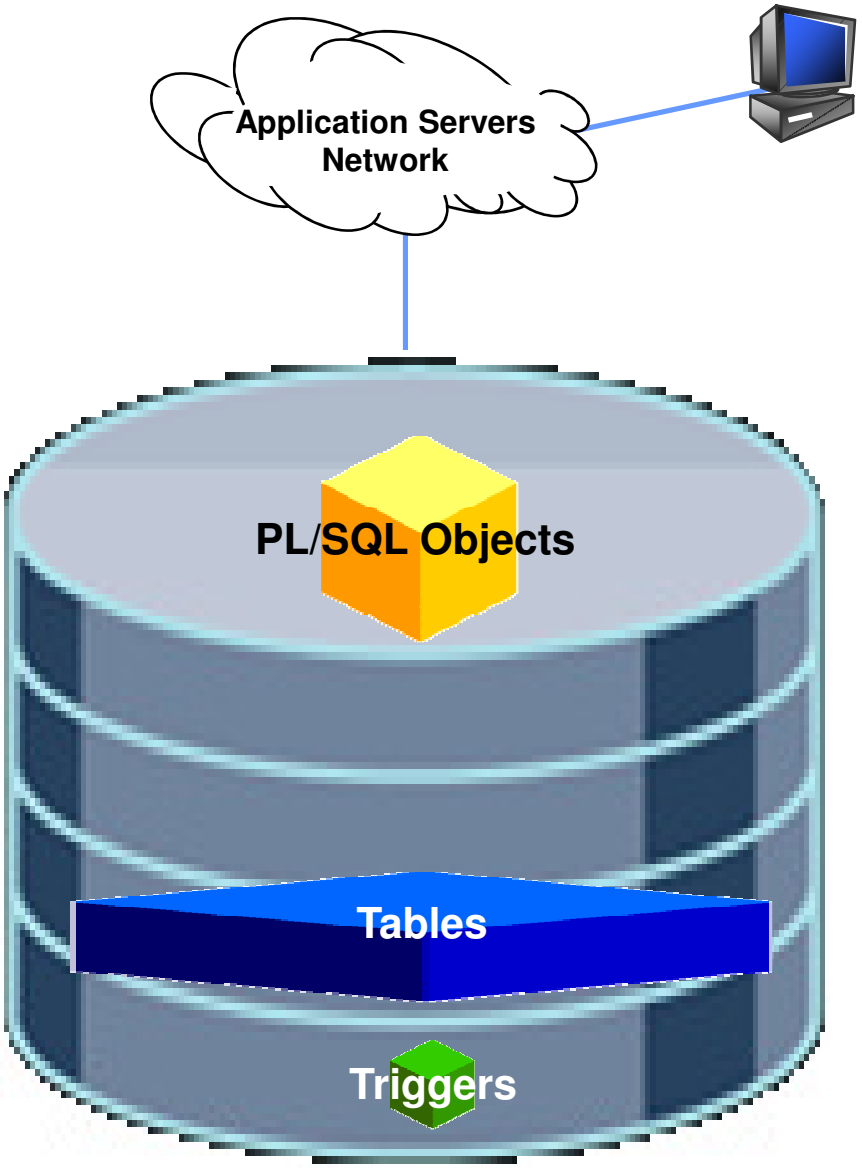
EBR Bugs

Only one worth mentioning

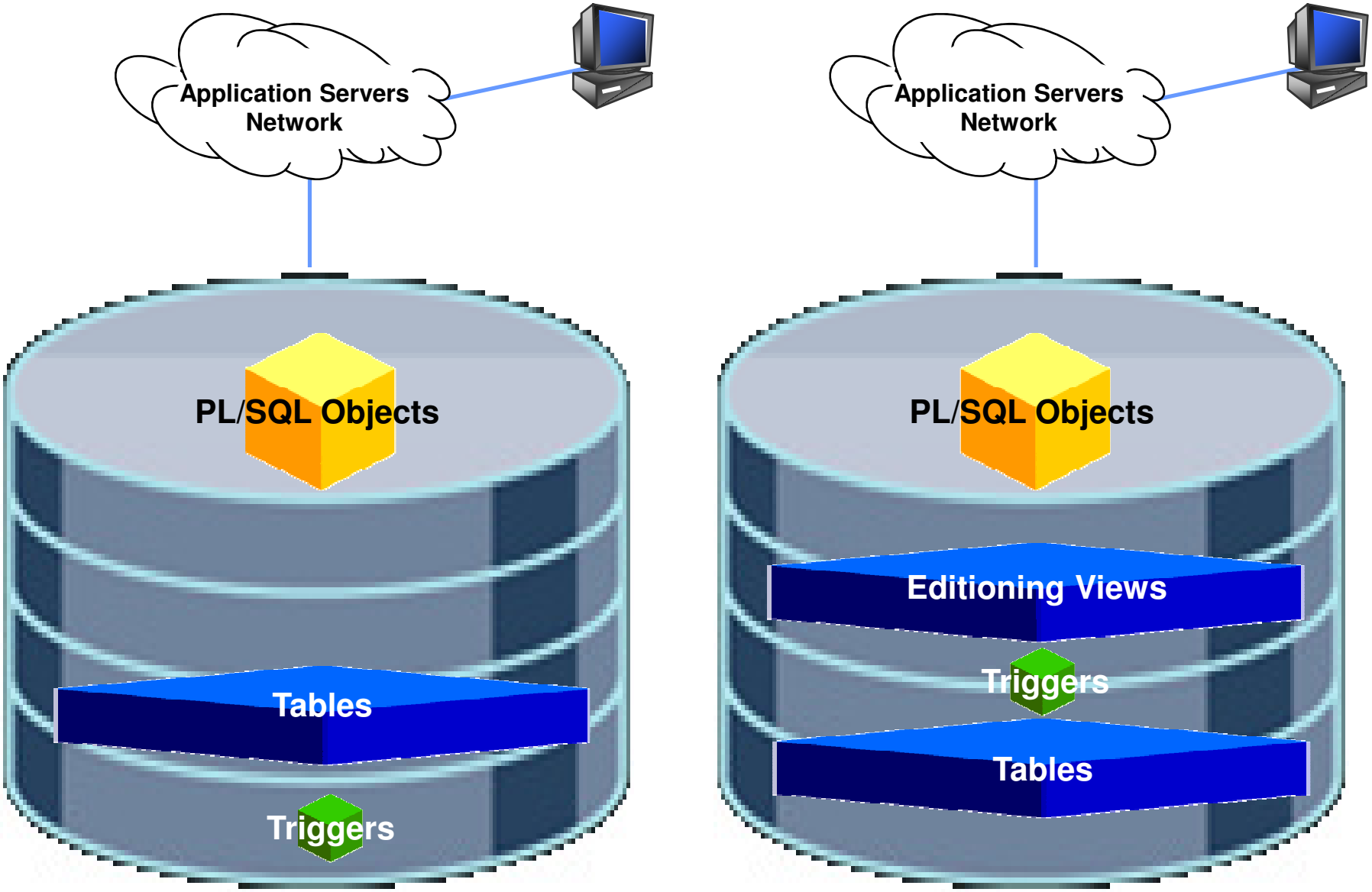
- Mental model ... objects are copied
- Physical model ... the appearance of polymorphism and inheritance
- The mental model will become the physical model so do not rely on this "feature"
- This will be addressed in 12gR1

EBR Implementation

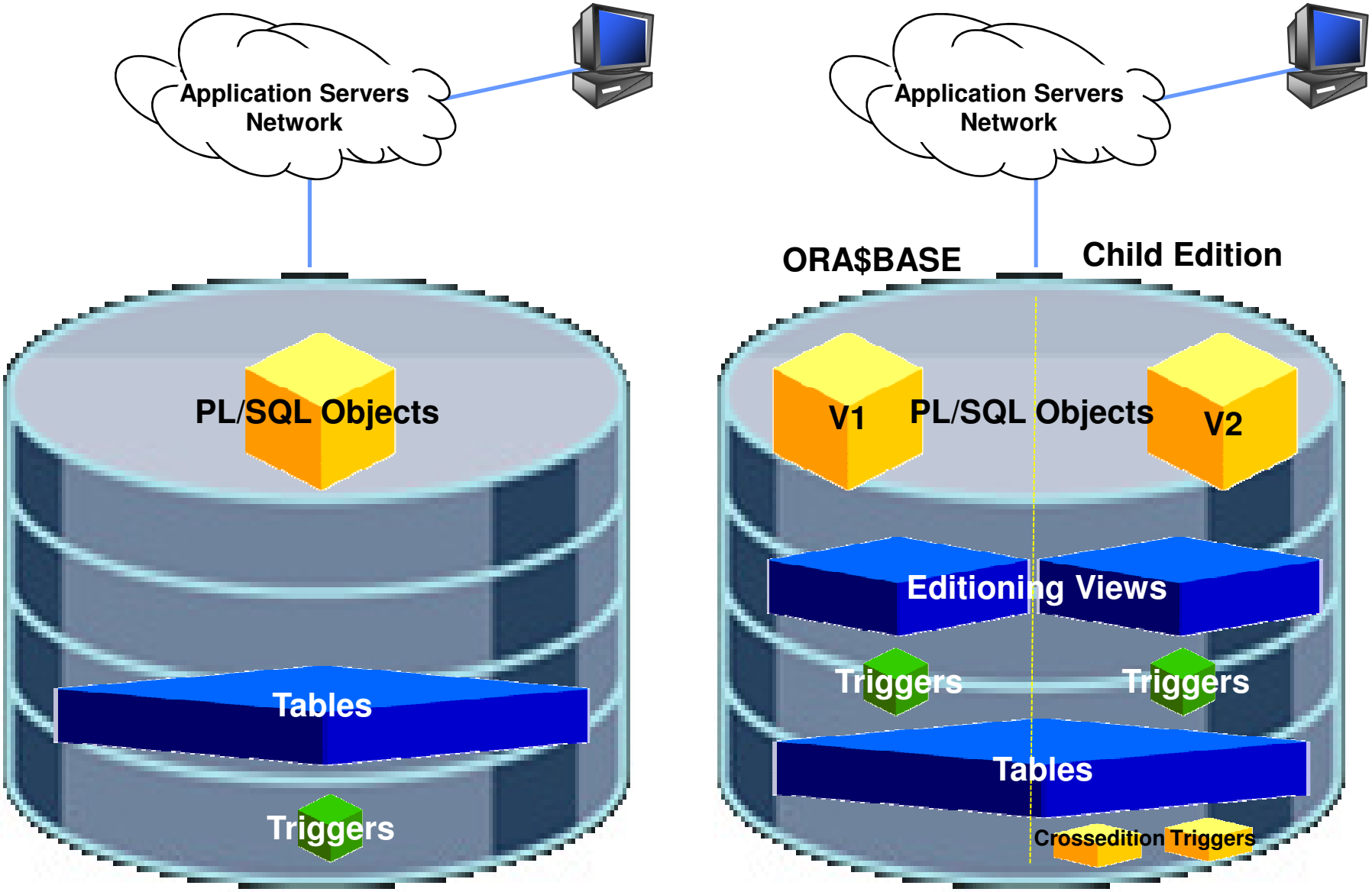
Application Version 1



Preparing For EBR



Create Second Edition



Health Warning

Due to complaints made to the European Union's Directorate General Health and Consumer Protection ...

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