Oracle Multitenant and Database 12c

Spring 2017

Daniel A. Morgan
email: vp@tcoug.org
mobile: +1 206-669-2949

Friday: April 21, 2017

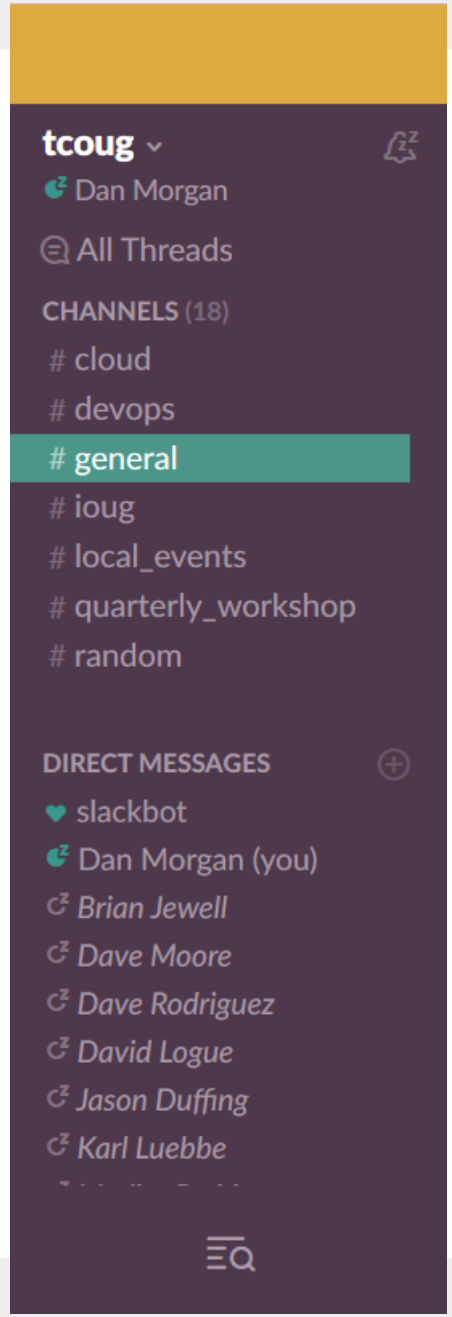
Welcome

- Dan Morgan
  - Vice President TCOUG
  -  Oracle ACE Director Alumni
- Our TCOUG Slack Community
- Local Events
- ODTUG
- OpenWorld 2017

Welcome

- Dan Morgan
  - Vice President TCOUG
  - Oracle ACE Director Alumni
- Our TCOUG Slack Community
- Local Events
- ODTUG
- OpenWorld 2017

`https://tcoug.slack.com/`



Welcome

- Dan Morgan
  - Vice President TCOUG
  - Oracle ACE Director Alumni
- Our TCOUG Slack Community
- Local Events
- ODTUG
- OpenWorld 2017

`https://www.oracle.com/search/events ... Search: "Minneapolis"`

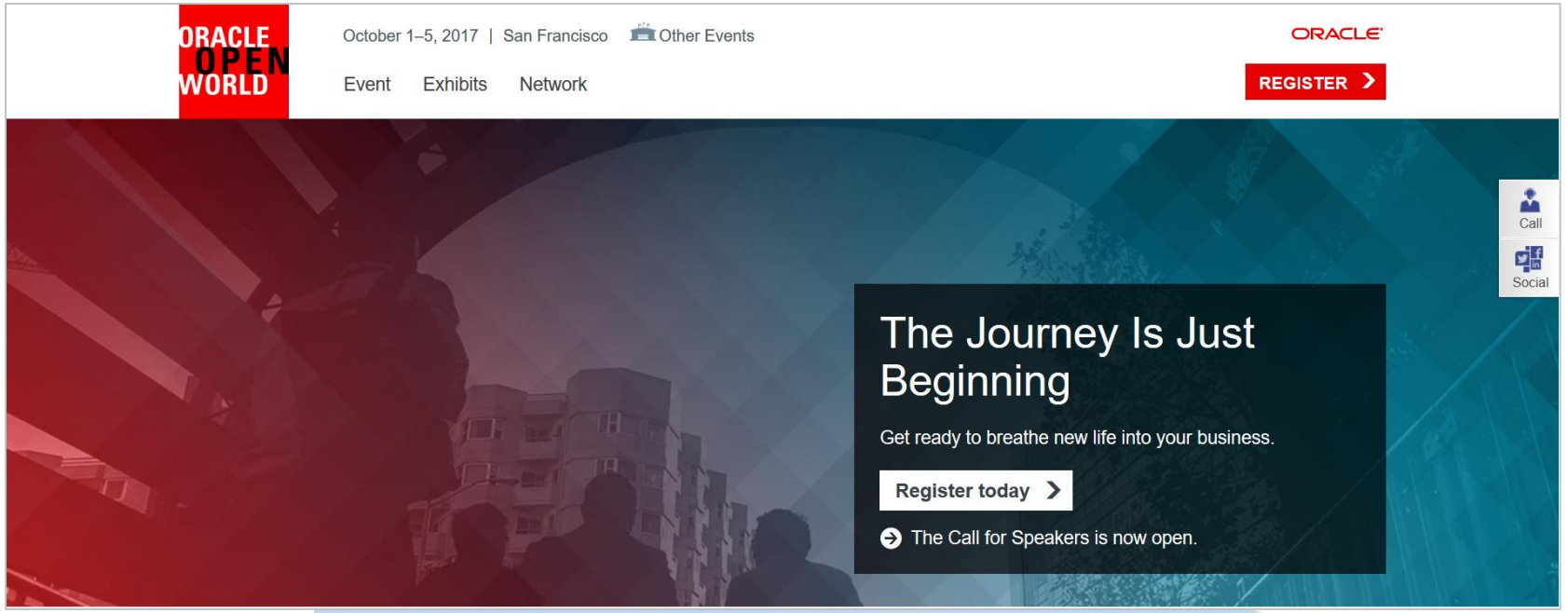
Welcome

- Dan Morgan
  - Vice President TCOUG
  - Oracle ACE Director Alumni
- Our TCOUG Slack Community
- Local Events
- ODTUG
- OpenWor

http://kscope17.com/

Welcome

- Dan Morgan
  - Vice President TCOUG
  - Oracle ACE Director Alumni
- Our TCOUG Slack Community
- Local Events
- ODTUG
- OpenWorld 2017



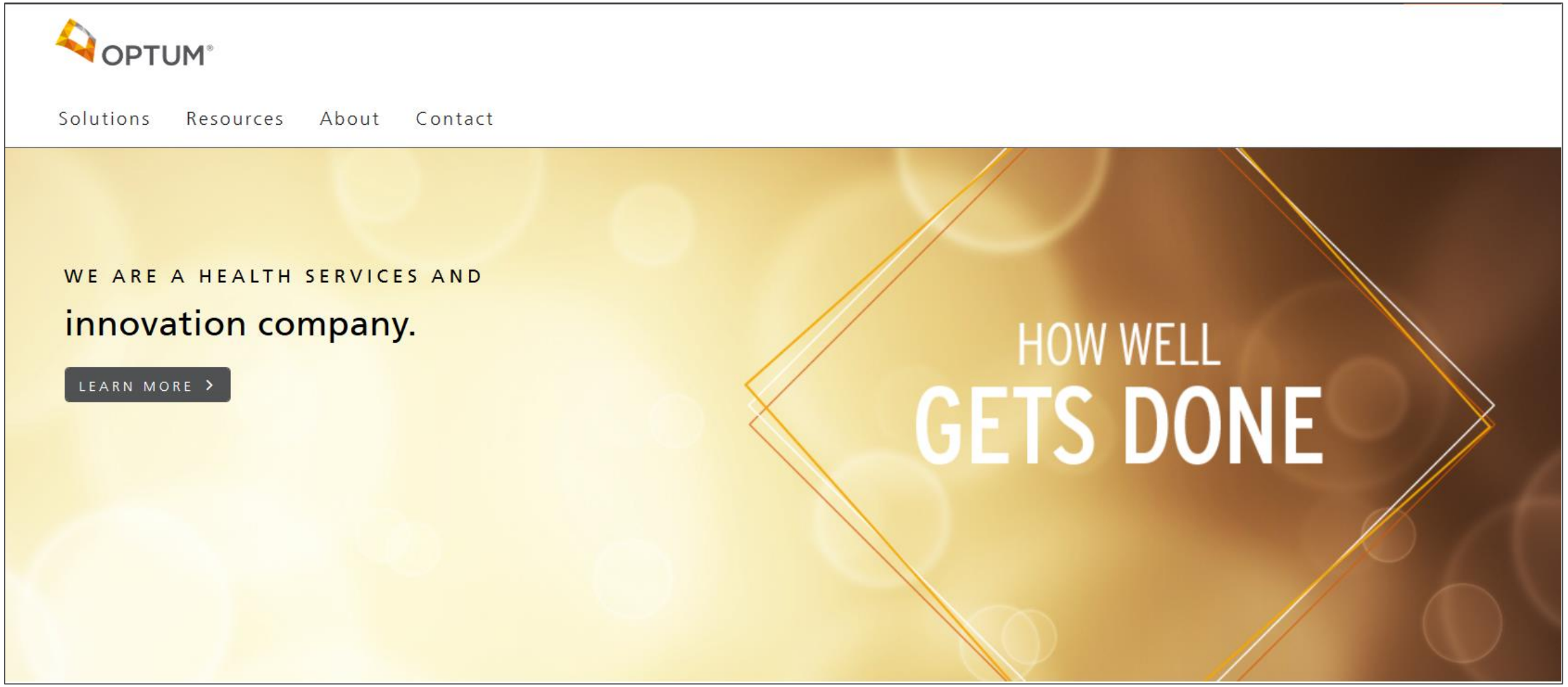
`https://www.oracle.com/openworld/index.html`

Welcome

Time	Description
08:30 - 09:00	Registration and Networking
09:00 - 09:30	Welcoming Remarks, Sponsor and Speaker Introductions
09:30 - 10:30	What's new in 12.2: Dan Morgan
10:30 - 10:45	Break
10:45 - 11:45	Oracle Multitenant: The Road Ahead: Patrick Wheeler, Oracle
11:45 - 12:00	Lightning Session: Native Compilation in PL/SQL: Dan Morgan
12:00 - 01:00	Lunch
01:00 - 01:15	Lightning Session: Platinum Gateway Security Concerns, Brian Bream, Collier IT
01:15 - 02:00	Q & A
02:00 - 03:00	Oracle Multitenant: Ideal Architecture for World-Class SaaS: Patrick Wheeler, Oracle
03:00 - 03:15	Break
03:15 - 03:45	Oracle Multitenant: Live in SQL*Plus: Dan Morgan
03:45 - 04:00	Giveaways, Prizes, and Closing Remarks

Sponsor Introductions

Sponsors


OPTUM
Solutions
Resources
About
Contact
WE ARE A HEALTH SERVICES AND
innovation company.
LEARN MORE
HOW WELL
GETS DONE

Sponsors

Sponsors

Sponsors

Speaker Introductions

Patrick Wheeler

- Senior Director, Product Management, Oracle Database
- Oracle
  - Joined Oracle in 1986
  - Consultant and Consulting Practice Manager for several large financial institutions in London and San Francisco
  - Closely involved with the development of Oracle CASE and was responsible for the worldwide launch of CASE and establishment of custom consulting practices in the early nineties
- Siebel
  - Joined Siebel Systems in 1995 and was a member of the Founder's Circle
  - As Data Architect and Director of Data Modeling he oversaw the development of Siebel's Data Model from Sales Force Automation startup to CRM juggernaut
  - Establish the Reliability Engineering practice guiding the product from pre-release instability to production release solidity
  - Awarded 3 patents as Inventor
- Now back at Oracle
  - Patrick's Focus is on Oracle Database 12c and the Multitenant feature
  - Follow Patrick on Twitter @OraclePDB

Brian Bream

- Brian is the Chief Technology Officer at Collier IT
- Oracle ACE in Systems Technologies
- He has over 36 years in the Information Technology field starting in the US Navy in 1981
- Although proficient in many technologies his passion has been in the Operating Environment, Engineered Systems, Data Center grade servers, Big Data/Hadoop, and storage
- He has been teaching for Sun Microsystems / Oracle since 2000
- Brian actively writes a blog and provides formal presentations on various Oracle and related technology topics
- You can follow Brian on Twitter @Snatchbrain

Dan Morgan

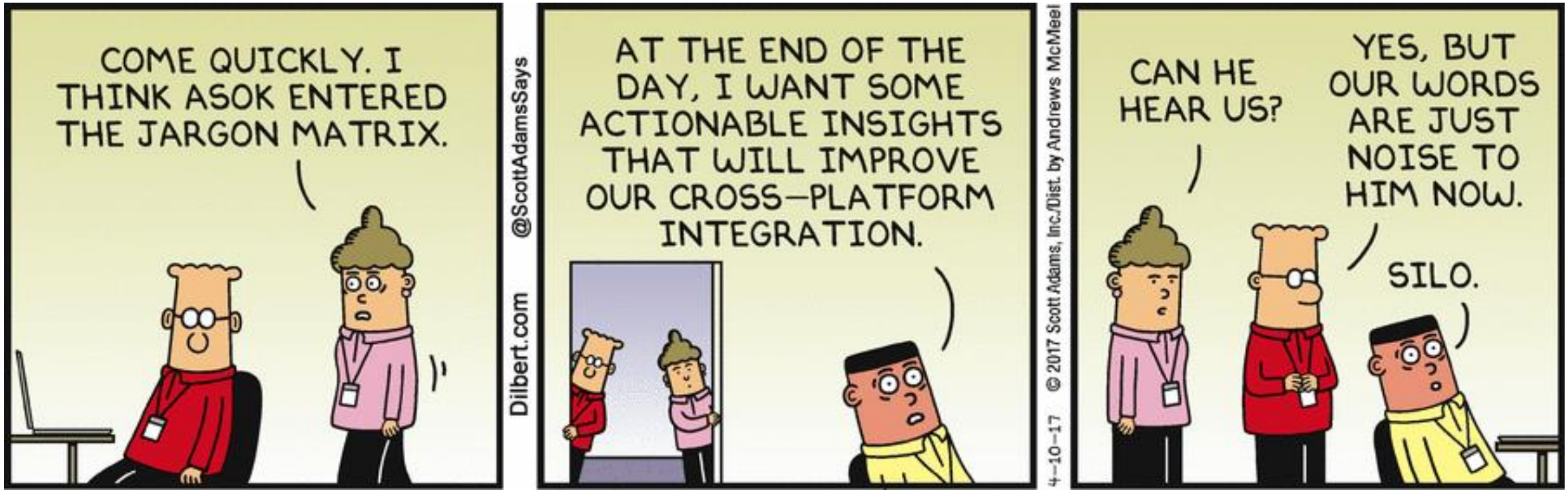
- Oracle ACE Director Alumni
- Oracle Educator
  - Curriculum author and primary program instructor at University of Washington
  - Consultant: Harvard University
  - University Guest Lecturers
    - APAC: University of Canterbury (NZ)
    - EMEA: University of Oslo (Norway)
    - Latin America: Universidad Latina de Panama and Technologico de Costa Rica
- IT Professional
  - First computer: IBM 360/40 in 1969: Fortran IV
  - Oracle Database since 1988-9
  - Beta Tester 10g, 11g, 12c, GoldenGate, TimesTen
  - The Morgan behind www.morganslibrary.org
  - Member Oracle Data Integration Solutions Partner Advisory Council
  - Co-Founder International GoldenGate Oracle Users Group
  - Vice President Twin Cities Oracle Users Group
- Principal Adviser: Forsythe **Meta7**

System/370-145 system console

Learning Experience Alert

Content Density Warning

Take Notes

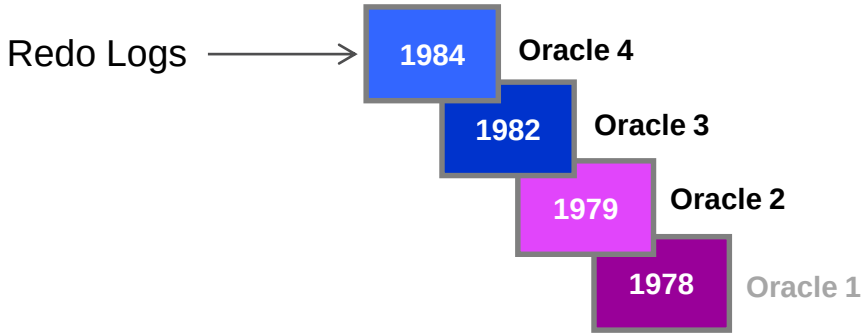

COME QUICKLY. I THINK ASOK ENTERED THE JARGON MATRIX.
@ScottAdamsSays
Dilbert.com
AT THE END OF THE DAY, I WANT SOME ACTIONABLE INSIGHTS THAT WILL IMPROVE OUR CROSS-PLATFORM INTEGRATION.
4-10-17
© 2017 Scott Adams, Inc./Dist. by Andrews McMeel
CAN HE HEAR US?
YES, BUT OUR WORDS ARE JUST NOISE TO HIM NOW.
SILO.

What's New in 12.2

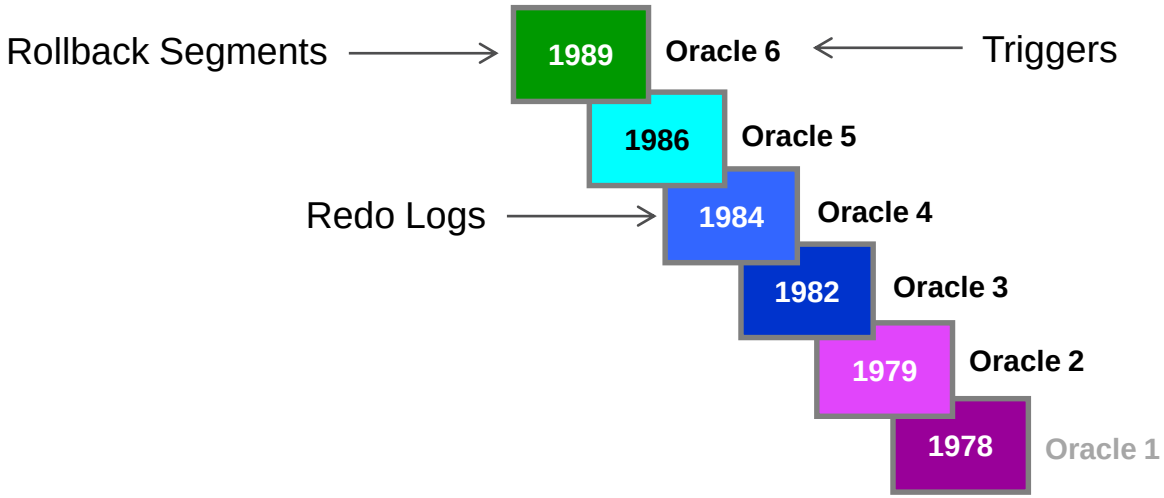
Introduction to 12cR2

TWIN CITIES ORACLE USERS GROUP

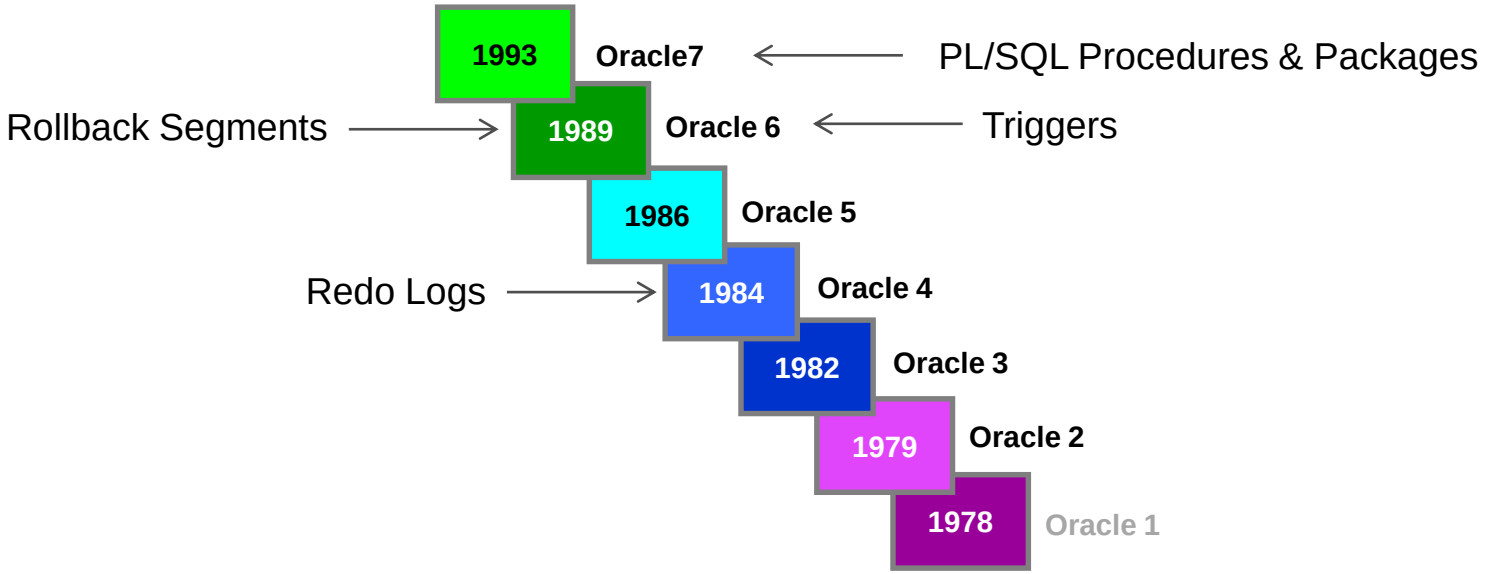
A Brief History of the Oracle Database



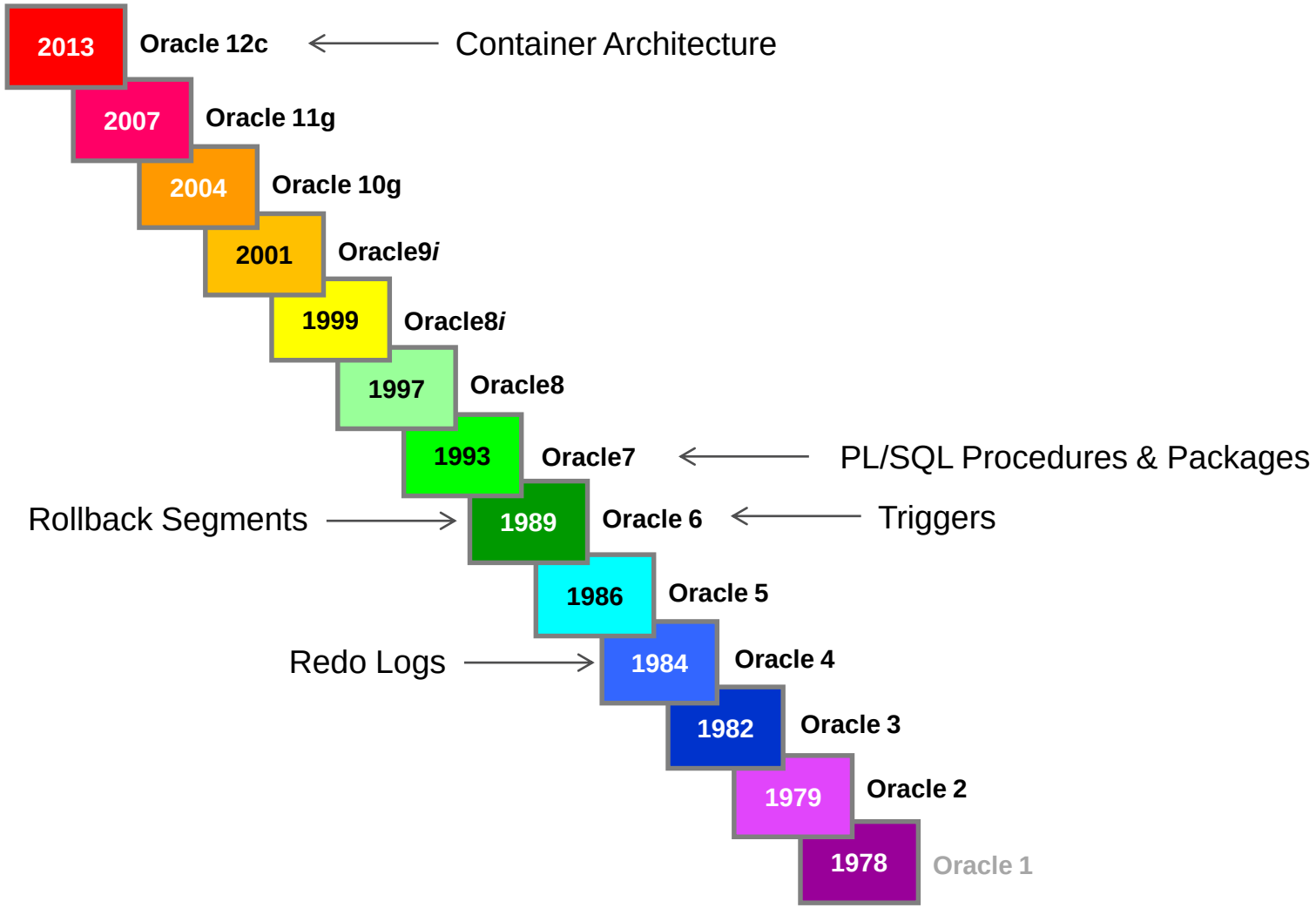
A Brief History of the Oracle Database



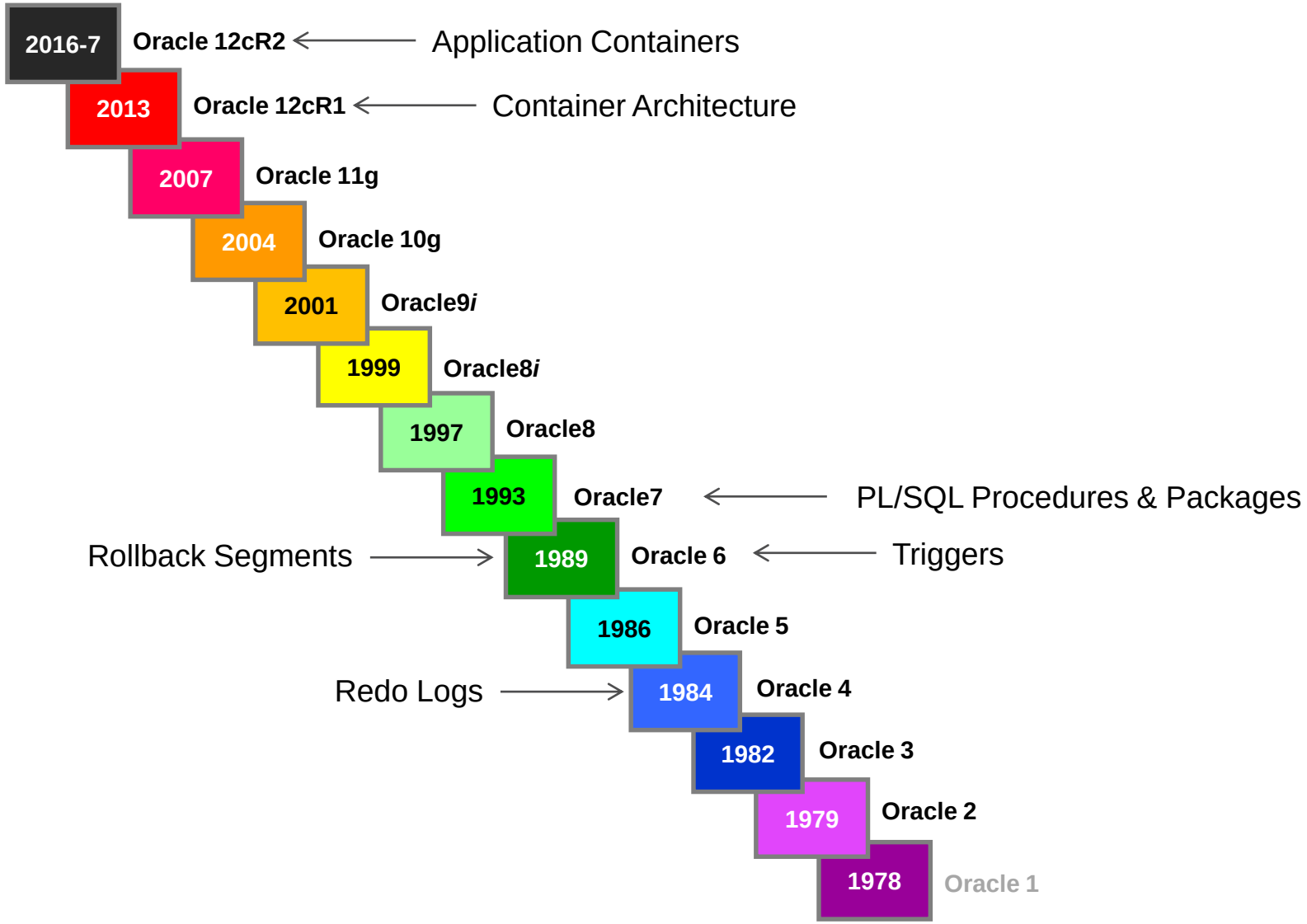
A Brief History of the Oracle Database



A Brief History of the Oracle Database

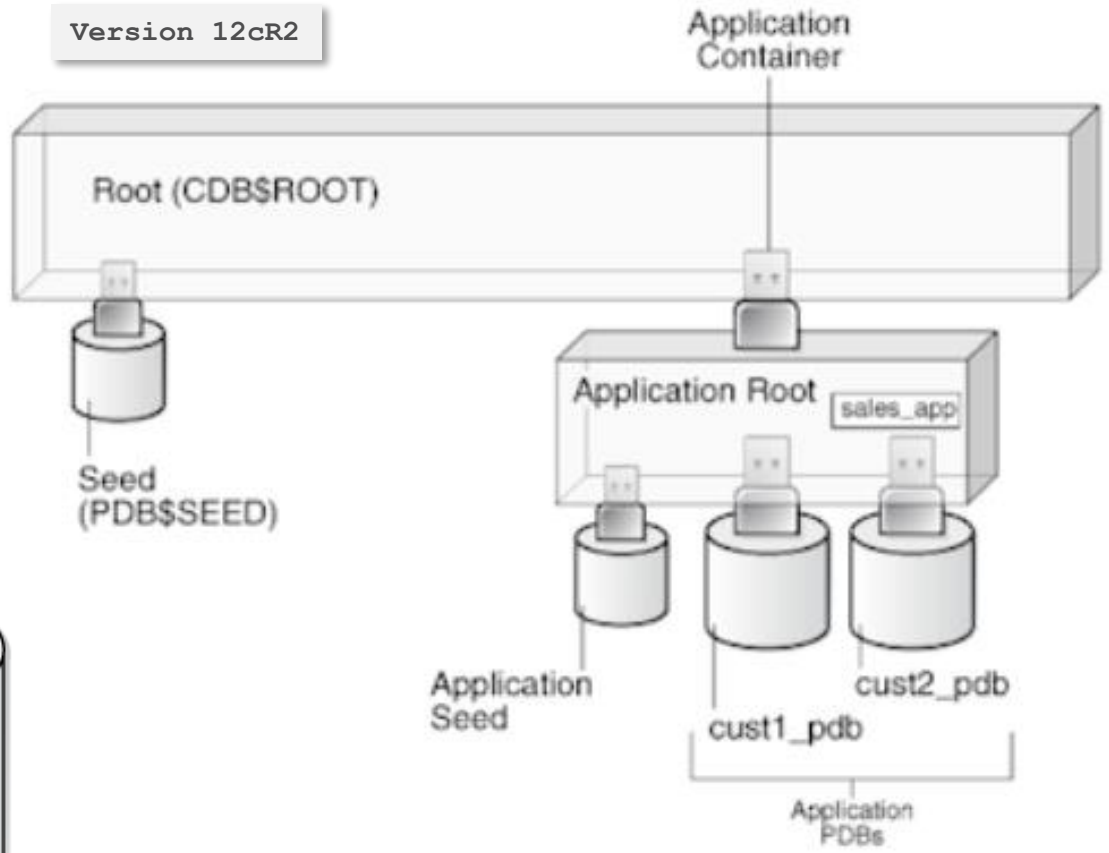


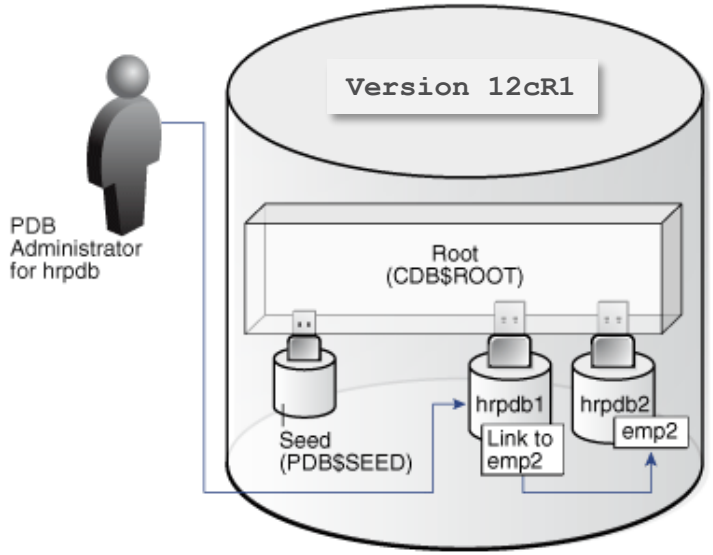
A Brief History of the Oracle Database



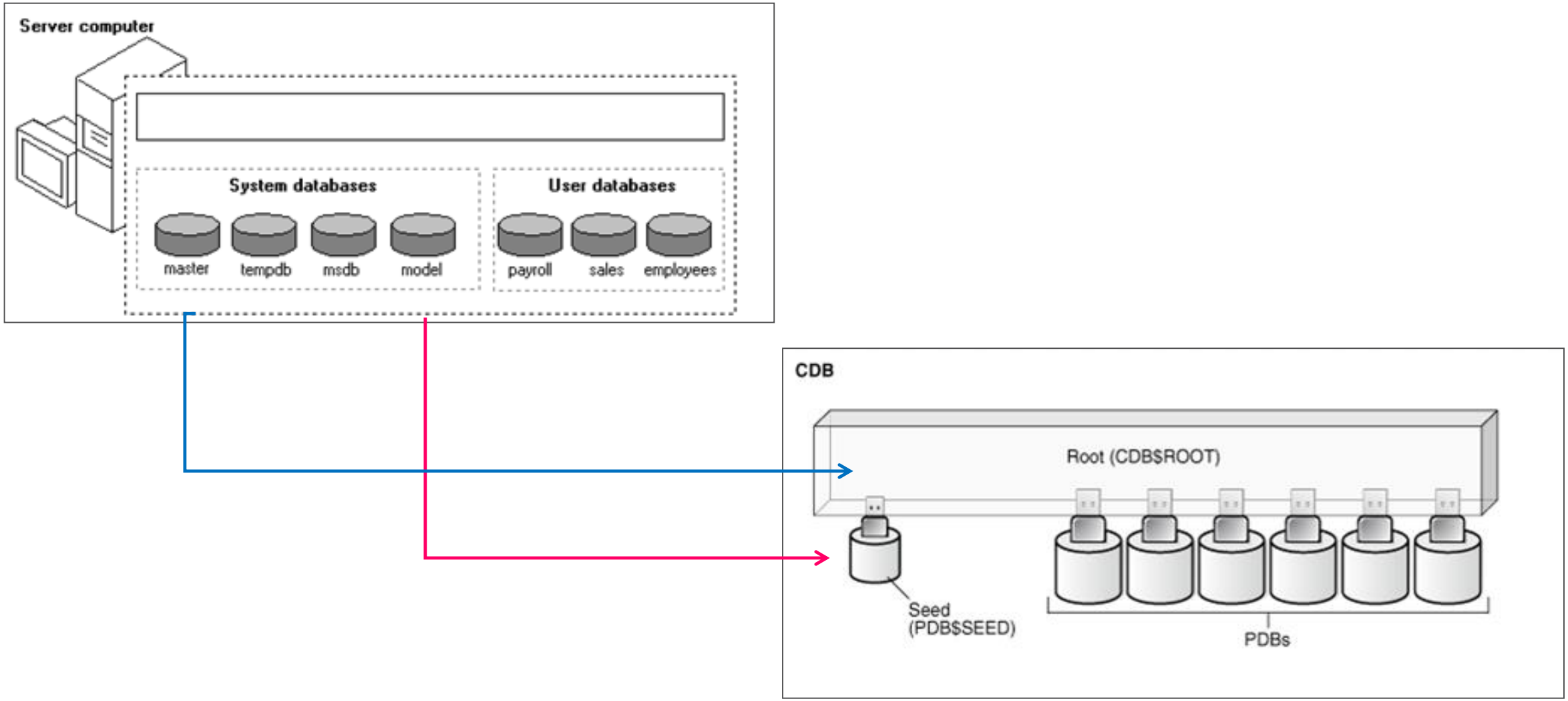
New Physical Architecture

New 12cR2 Container Database Architecture

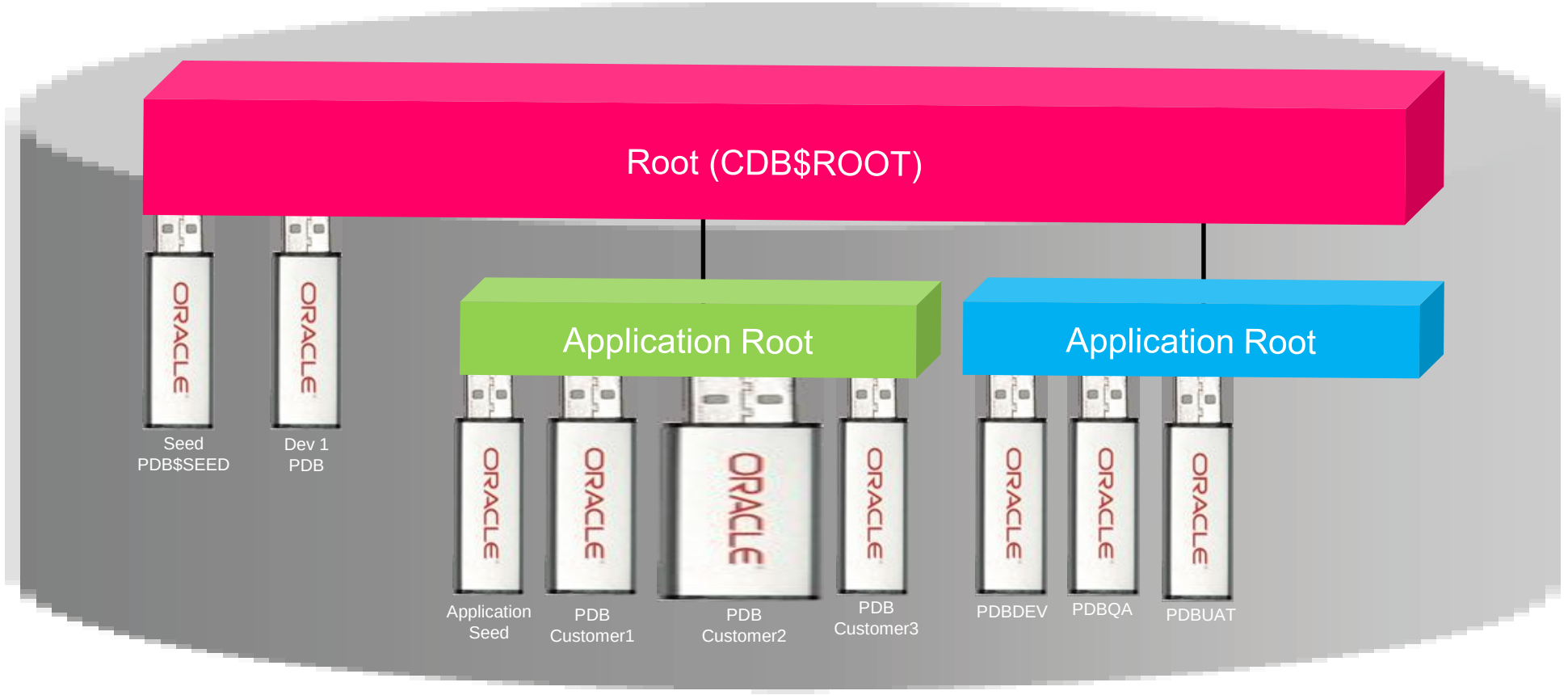




New 12cR1 Container Database Architecture



New 12cR2 Container Database Architecture



New Features

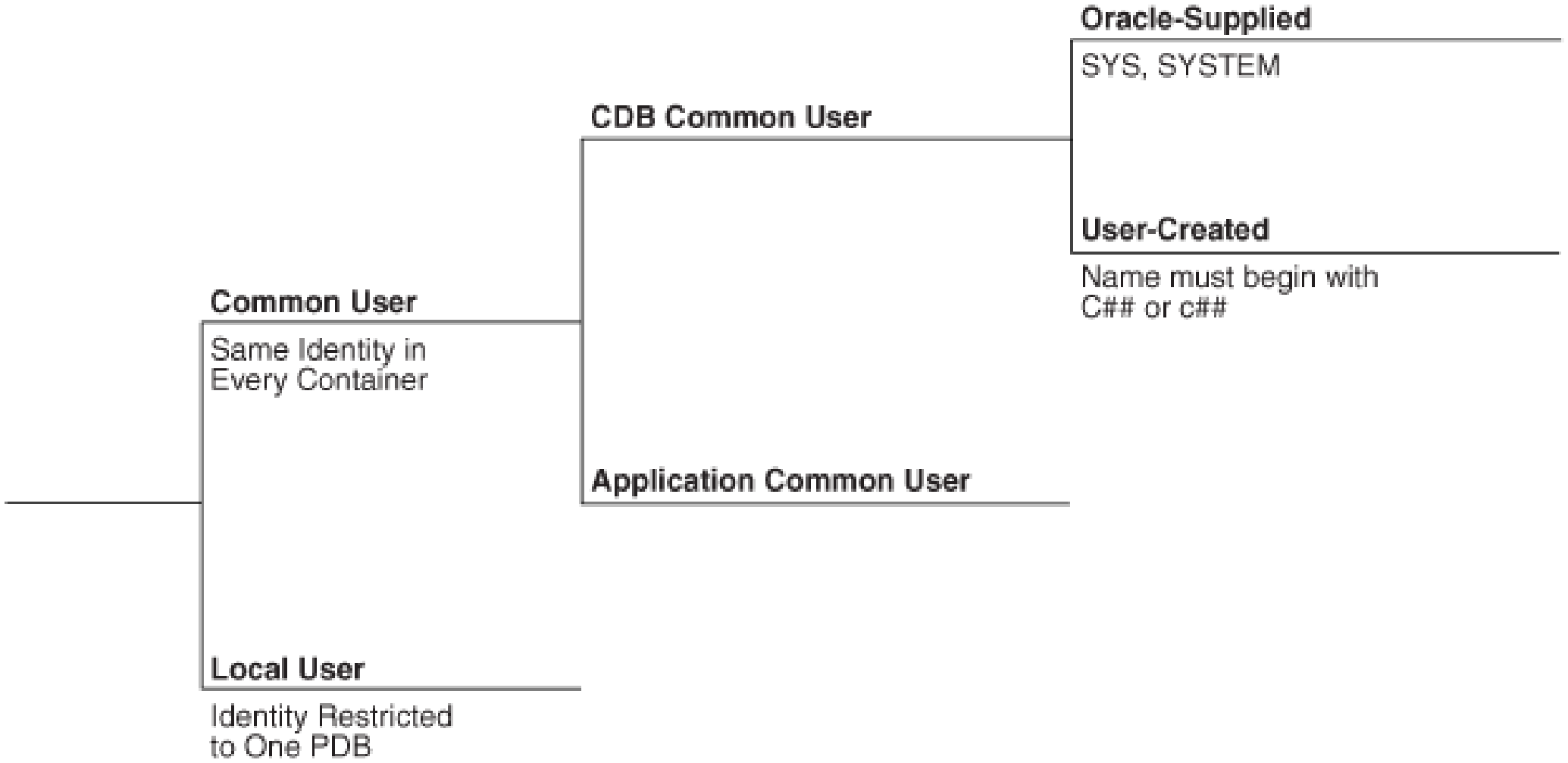
What Is The Definition Of "New Feature"?

- As with everything in Oracle ... it depends.
- There are the new features as defined by newly licensed options
- There are the new features as defined by Oracle's monitoring
- There are the new features as defined by the "New Features Guide"
- And then there is everything that is truly new because it has not existed in previous incarnations of our favourite database

Feature Usage Report Additions

Feature Proc	Description
DBMS_FEATURE_ACFS	Determines whether ACFS Drivers are loaded
DBMS_FEATURE_ACFS_ENCR	Detects usage of ACFS Encryption
DBMS_FEATURE_ACFS_SNAPSHOT	Detects usage of ACFS Snapshots
DBMS_FEATURE_AFD	Detects usage of ASM Filter Driver
DBMS_FEATURE_CLOUD_EHCC	Detects usage of Cloud Exadata Hybrid Columnar Compression
DBMS_FEATURE_FLEX_ASM	Detects usage of Flex ASM
DBMS_FEATURE_HCCCONV	Detects usage of Hybrid Columnar Compression Conventional Load
DBMS_FEATURE_IMFS	Detects usage of In-Memory FastStart
DBMS_FEATURE_IM_ADO	Detects usage of In-Memory ADO Policies
DBMS_FEATURE_IM_EXPRESSIONS	Detects usage of In-Memory Expressions
DBMS_FEATURE_IM_FORSERVICE	Detects usage of In-Memory For Service usage
DBMS_FEATURE_IM_JOINGROUPS	Detects usage of In-Memory Join Groups
DBMS_FEATURE_SHARD	Detects whether the current container is a database shard
DBMS_FEATURE_THP	Detects the usage of ASM Thin Provisioning
DBMS_FEATURE_UTILITIES5	Detects the usage of database utilities for external tables {ORACLE_LOADER)
DBMS_FEATURE_UTILITIES6	Detects the usage of database utilities for external tables (ORACLE_BIGSQL)
DBMS_LOGMNR_FFVTOLOGMNRT	Not Documented: Code Wrapped
DBMS_PDB_NUM	Count of PDBs to determine whether Multitenant is in use

New 12cR2 Container Database Architecture



12.2: History

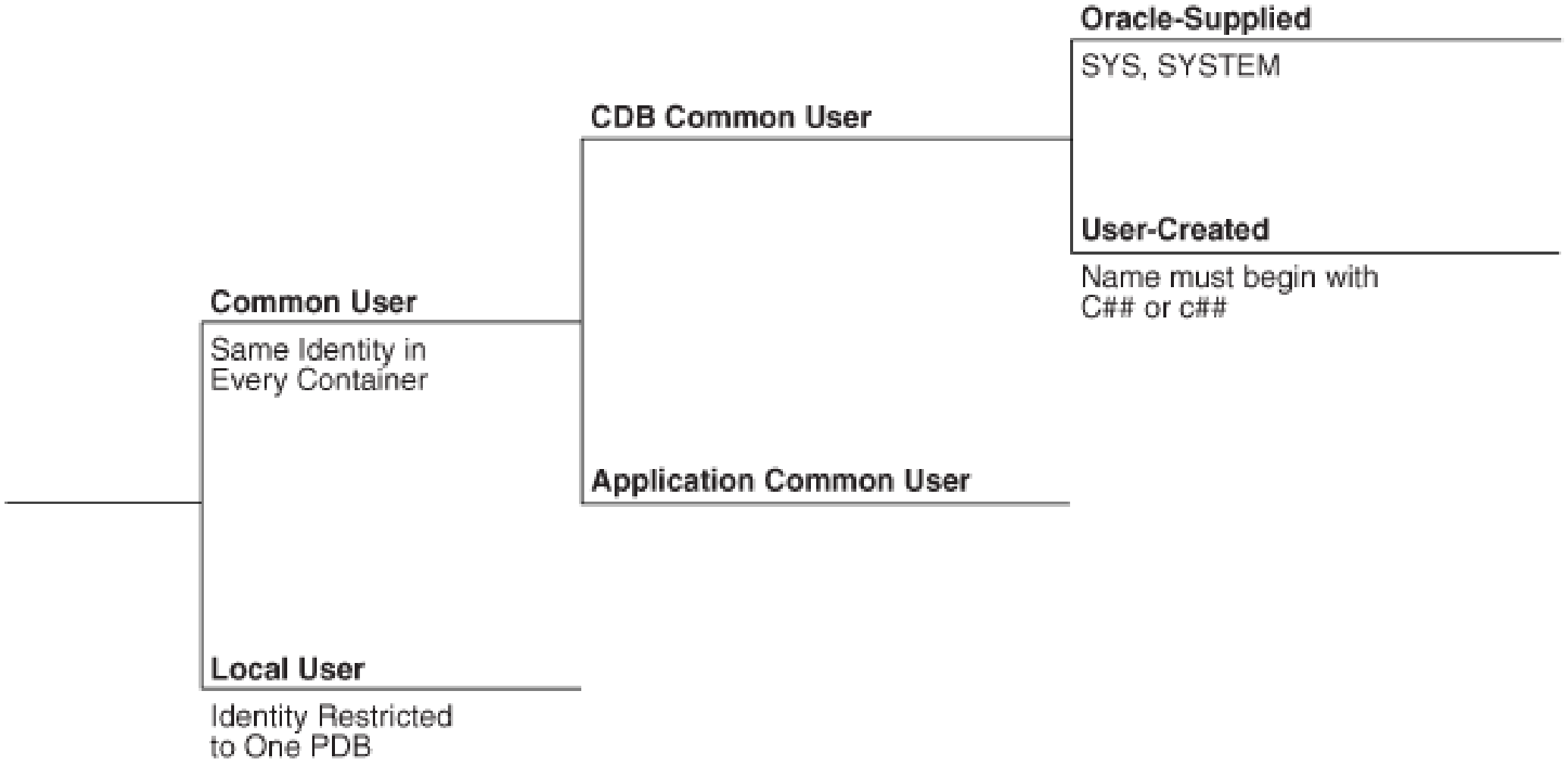
```
SQL> show history
history is OFF

SQL> set history on

SQL> SELECT COUNT(*) FROM tab$;

  COUNT(*)
----------
      2150

SQL> SELECT COUNT(*) FROM obj$;

  COUNT(*)
----------
     72629

SQL> SELECT COUNT(*) FROM source$;

  COUNT(*)
----------
     12992

SQL> hist
  1  SELECT COUNT(*) FROM tab$;
  2  SELECT COUNT(*) FROM obj$;
  3  SELECT COUNT(*) FROM source$;
```

```
SQL> run 2
  1* SELECT COUNT(*) FROM source$

  COUNT(*)
----------
     12992
```

```
HIST[ORY] [n RUN | EDIT | DEL[ETE]] | [CLEAR | LIST]

SQL> hist list
  1  SELECT COUNT(*) FROM tab$;
  2  SELECT COUNT(*) FROM obj$;
  3  SELECT COUNT(*) FROM source$;
  4  cl scr
  5  run 3
  6  run 2
  7  edit 3
  8  run 2
  9  edit 2
```

New Users

New 12cR2 Container Database Architecture



Users (1:2)

New: 12cR1

AUDSYS
GSMADMIN_INTERNAL
GSMCATUSER
GSMUSER
PDBADMIN
SYSBACKUP
SYSDG
SYSKM

New: 12cR2

APEX_050100
APEX_INSTANCE_ADMIN_USER
APEX_LISTENER
APEX_REST_PUBLIC_USER
DBJSON
DBSFWUSER
GGSYS
HRREST
OBE
ORDS_METADATA
ORDS_PUBLIC_USER
PDBADMIN
REMOTE_SCHEDULER_AGENT
RESTFUL
SYS$UMF
SYSRAC
XDBEXT
XDBPM
XFILES

Dropped
BI, OE, PM, SH, and SPATIAL_WFS_USR

New Users With Escalated Privs

USERNAME	Usage
GGSYS	The internal account used by Oracle GoldenGate. It should not be unlocked or used for a database login.
SYSBACKUP	This privilege allows a user to perform backup and recovery operations either from Oracle Recovery Manager (RMAN) or SQL*Plus.
SYSDG	This privilege allows a user to perform Data Guard operations can use this privilege with either Data Guard Broker or the DGMGRL command-line interface.
SYSKM	This privilege allows a user to perform Transparent Data Encryption keystore operations.
SYSRAC	This privilege allows the Oracle agent of Oracle Clusterware to perform Oracle Real Application Clusters (Oracle RAC) operations. SYSRAC facilitates Oracle Real Application Clusters (Oracle RAC) operations by connecting to the database by the Clusterware agent on behalf of Oracle RAC utilities such as SRVCTL.

New Profiles, Roles & System Privs

TWIN CITIES ORACLE USERS GROUP

New Security Features

- 12cR1
  - New password validation functions
  - New Profiles
  - New Roles
  - New System Privileges
  - New secconf.sql file
  - Unified key management interface for Transparent Data Encryption with ADMINISTER KEY MANAGEMENT statement
- 12cR2
  - New Roles
  - New System Privileges

Profiles

12cR1 Default

```
COMPOSITE_LIMIT           UNLIMITED
CONNECT_TIME              UNLIMITED
CPU_PER_CALL              UNLIMITED
CPU_PER_SESSION           UNLIMITED
FAILED_LOGIN_ATTEMPTS      10
IDLE_TIME                 UNLIMITED

LOGICAL_READS_PER_CALL    UNLIMITED
LOGICAL_READS_PER_SESSION UNLIMITED
PASSWORD_GRACE_TIME          7
PASSWORD_LIFE_TIME        180
PASSWORD_LOCK_TIME           1
PASSWORD_REUSE_MAX        UNLIMITED
PASSWORD_REUSE_TIME       UNLIMITED
PASSWORD_VERIFY_FUNCTION  NULL
PRIVATE_SGA               UNLIMITED
SESSIONS_PER_USER         UNLIMITED
```

12cR2 ORA_STIG_PROFILE

```
COMPOSITE_LIMIT           UNLIMITED
CONNECT_TIME              UNLIMITED
CPU_PER_CALL              UNLIMITED
CPU_PER_SESSION           UNLIMITED
FAILED_LOGIN_ATTEMPTS       3
IDLE_TIME                 15
INACTIVE_ACCOUNT_TIME     35
LOGICAL_READS_PER_CALL    UNLIMITED
LOGICAL_READS_PER_SESSION UNLIMITED
PASSWORD_GRACE_TIME          5
PASSWORD_LIFE_TIME         60
PASSWORD_LOCK_TIME        UNLIMITED
PASSWORD_REUSE_MAX        10
PASSWORD_REUSE_TIME       265
PASSWORD_VERIFY_FUNCTION  ORA12C_STIG_VERIFY_FUNCTION
PRIVATE_SGA               UNLIMITED
SESSIONS_PER_USER         UNLIMITED
```

Starting with this release, you can use the INACTIVE_ACCOUNT_TIME parameter to automatically lock the account of a database user who has not logged in to the database instance in a specified number of days.

Roles

12cR1 New
ADM_PARALLEL_EXECUTE_TASK
APEX_GRANTS_FOR_NEW_USERS_ROLE
AUDIT_ADMIN
AUDIT_VIEWER
CAPTURE_ADMIN
CDB_DBA
DBHADOOP
DV_AUDIT_CLEANUP
DV_GOLDENGATE_ADMIN
DV_GOLDENGATE_REDO_ACCESS
DV_MONITOR
DV_PATCH_ADMIN
DV_STREAMS_ADMIN
DV_XSTREAM_ADMIN
EM_EXPRESS_ALL
EM_EXPRESS_BASIC
GSMADMIN_ROLE
GSMUSER_ROLE
GSM_POOLADMIN_ROLE
HS_ADMIN_SELECT_ROLE
LBAC_DBA
OPTIMIZER_PROCESSING_RATE
PDB_DBA
PROVISIONER
XS_CACHE_ADMIN
XS_NAMESPACE_ADMIN
XS_RESOURCE
XS_SESSION_ADMIN

12cR1 Dropped
DELETE_CATALOG_ROLE

12cR2 New
APEX_ADMINISTRATOR_READ_ROLE
APPLICATION_TRACE_VIEWER
DATAPATCH_ROLE
DBJAVASCRIPT
DBMS_MDX_INTERNAL
DV_POLICY_OWNER
GGSYS_ROLE
RDFCTX_ADMIN
RECOVERY_CATALOG_OWNER_VPD
SODA_APP
SYSUMF_ROLE
XFILES_ADMINISTRATOR
XFILES_USER
XS_CONNECT

12cR2 Dropped
DBAHADOOP
SPATIAL_WFS_ADMIN
WFS_USR_ROLE
XS_RESOURCE

12cR1 Profile Script (1:2)

- $ORACLE_HOME/rdbms/admin/utlpwdmg.sql
- Contains the following password verification functions
  - CREATE OR REPLACE FUNCTION ora12c_verify_function
  - CREATE OR REPLACE FUNCTION verify_function_11G
  - CREATE OR REP

```
-- This script alters the default parameters for Password Management
-- This means that all the users on the system have Password Management
-- enabled and set to the following values unless another profile is
-- created with parameter values set to different value or UNLIMITED
-- is created and assigned to the user.

ALTER PROFILE DEFAULT LIMIT
PASSWORD_LIFE_TIME 180
PASSWORD_GRACE_TIME 7
PASSWORD_REUSE_TIME UNLIMITED
PASSWORD_REUSE_MAX  UNLIMITED
FAILED_LOGIN_ATTEMPTS 10
PASSWORD_LOCK_TIME 1
PASSWORD_VERIFY_FUNCTION ora12c_verify_function;

/**
The below set of password profile parameters would take into consideration
recommendations from Center for Internet Security[CIS Oracle 11g].

ALTER PROFILE DEFAULT LIMIT
PASSWORD_LIFE_TIME 90
PASSWORD_GRACE_TIME 3
PASSWORD_REUSE_TIME 365
PASSWORD_REUSE_MAX  20
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_LOCK_TIME 1
PASSWORD_VERIFY_FUNCTION ora12c_verify_function;
*/

/**
The below set of password profile parameters would take into
consideration recommendations from Department of Defense Database
Security Technical Implementation Guide[STIG v8R1].

ALTER PROFILE DEFAULT LIMIT
PASSWORD_LIFE_TIME 60
PASSWORD_REUSE_TIME 365
PASSWORD_REUSE_MAX  5
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_VERIFY_FUNCTION ora12c_strong_verify_function;
*/
```

12cR1 Profile Script (2:2)

- $ORACLE_HOME/rdbms/admin/utlpwdmg.sql
- Contains the following password profile alterations

```
-- This script alters the default parameters for Password Management. This means that all the users on the
-- system have Password Management enabled and set to the following values unless another profile is  created
-- with parameter values set to different value or UNLIMITED is created and assigned to the user.

ALTER PROFILE DEFAULT LIMIT
PASSWORD_LIFE_TIME 180
PASSWORD_GRACE_TIME 7
PASSWORD_REUSE_TIME UNLIMITED
PASSWORD_REUSE_MAX  UNLIMITED
FAILED_LOGIN_ATTEMPTS 10
PASSWORD_LOCK_TIME 1
PASSWORD_VERIFY_FUNCTION ora12c_verify_function;

/**
The below set of password profile parameters would take into consideration
recommendations from Center for Internet Security[CIS Oracle 11g].

ALTER PROFILE DEFAULT LIMIT
PASSWORD_LIFE_TIME 90
PASSWORD_GRACE_TIME 3
PASSWORD_REUSE_TIME 365
PASSWORD_REUSE_MAX  20
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_LOCK_TIME 1
PASSWORD_VERIFY_FUNCTION ora12c_verify_function;
*/

/**
The below set of password profile parameters would take into consideration recommendations from Department of
Defense Database Security Technical Implementation Guide[STIG v8R1].

ALTER PROFILE DEFAULT LIMIT
PASSWORD_LIFE_TIME 60
PASSWORD_REUSE_TIME 365
PASSWORD_REUSE_MAX  5
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_VERIFY_FUNCTION ora12c_strong_verify_function;
*/
```

System Privileges

12cR1 New
ADMINISTER KEY MANAGEMENT
ALTER ANY CUBE BUILD PROCESS
ALTER ANY MEASURE FOLDER
ALTER ANY SQL TRANSLATION PROFILE
CREATE ANY CREDENTIAL
CREATE ANY SQL TRANSLATION PROFILE
CREATE CREDENTIAL
CREATE PLUGGABLE DATABASE
CREATE SQL TRANSLATION PROFILE
DROP ANY SQL TRANSLATION PROFILE
EM EXPRESS CONNECT
EXEMPT ACCESS POLICY
EXEMPT DDL REDACTION POLICY
EXEMPT DML REDACTION POLICY
EXEMPT IDENTITY POLICY
EXEMPT REDACTION POLICY
INHERIT ANY PRIVILEGES
KEEP_DATE TIME
KEEP_SYSGUID
LOGMINING
PURGE DBA_RECYCLEBIN
REDEFINE ANY TABLE
SELECT ANY CUBE BUILD PROCESS
SELECT ANY MEASURE FOLDER
SET CONTAINER
SYSBACKUP
SYSDG
SYSKM
TRANSLATE ANY SQL
USE ANY SQL TRANSLATION PROFILE

12cR2 New
ALTER ANY ANALYTIC VIEW
CREATE ANALYTIC VIEW
CREATE ANY ANALYTIC VIEW
DROP ANY ANALYTIC VIEW

ALTER ANY ATTRIBUTE DIMENSION
CREATE ANY ATTRIBUTE DIMENSION
CREATE ATTRIBUTE DIMENSION
DROP ANY ATTRIBUTE DIMENSION

ALTER ANY HIERARCHY
CREATE ANY HIERARCHY
CREATE HIERARCHY
DROP ANY HIERARCHY

ALTER LOCKDOWN PROFILE
CREATE LOCKDOWN PROFILE
DROP LOCKDOWN PROFILE

DEBUG CONNECT ANY

INHERIT ANY REMOTE PRIVILEGES

SYSRAC

USE ANY JOB RESOURCE

12cR2 Modified
SELECT ANY DICTIONARY (altered in 12.1.0.2 to exclude some objects)

12cR1 Security Configuration Script (1:3)

- $ORACLE_HOME/rdbms/admin/secconf.sql

```
Rem $Header: rdbms/admin/secconf.sql /main/11 2014/02/26 04:16:54 risgupta Exp $
Rem
Rem secconf.sql
Rem
Rem Copyright (c) 2006, 2014, Oracle and/or its affiliates.
Rem All rights reserved.
Rem
Rem    NAME
Rem      secconf.sql - SECure CONFiguration script
Rem
Rem    DESCRIPTION
Rem      Secure configuration settings for the database include a reasonable
Rem      default password profile, password complexity checks, audit settings
Rem      (enabled, with admin actions audited), and as many revokes from PUBLIC
Rem      as possible. In the first phase, only the default password profile is
Rem      included.
Rem
Rem
Rem    NOTES
Rem      Only invoked for newly created databases, not for upgraded databases
Rem
Rem BEGIN SQL_FILE_METADATA
Rem SQL_SOURCE_FILE: rdbms/admin/secconf.sql
Rem SQL_SHIPPED_FILE: rdbms/admin/secconf.sql
Rem SQL_PHASE: SECCONF
Rem SQL_STARTUP_MODE: NORMAL
Rem SQL_IGNORABLE_ERRORS: NONE
Rem SQL_CALLING_FILE: rdbms/admin/execsec.sql
Rem END SQL_FILE_METADATA
Rem
```

12cR1 Security Configuration Script (2:3)

- $ORACLE_HOME/rdbms/admin/secconf.sql

```
Rem    MODIFIED    (MM/DD/YY)
Rem    risgupta     02/17/14 - Bug 18174384: Remove Logon/Logoff actions from
Rem                            ORA_SECURECONFIG audit policy
Rem    surman       01/22/14 - 13922626: Update SQL metadata
Rem    vpriyans     09/21/13 - Bug 17299076: Added ORA_CIS_RECOMMENDATIONS audit
Rem                            policy
Rem    jkati        02/04/13 - bug#16080525: Enable audit on DBMS_RLS by default
Rem    amunnoli     02/18/13 - Bug #16310544: add CREATE/DROP/ALTER PLUGGABLE
Rem                            DB actions to default audit configuration
Rem    vpriyans     06/05/12 - Bug 12904308: Audit CREATE DIRECTORY by default
Rem    vpriyans     03/22/12 - Bug 13413683: Rename predefined audit policies
Rem                            and add few more actions and privileges
Rem    nkgopal      09/08/11 - Bug 12794116: Configure Audit based on input
Rem                            argument
Rem    apsrivas     09/30/08 - bug 7428539: Add missing audit settings
Rem    asurpur      06/16/06 - audit changes for sec config
Rem    rburns       06/12/06 - secure configuration script
Rem    rburns       06/12/06 - Created
Rem

@@?/rdbms/admin/sqlsessstart.sql

Rem Secure configuration settings. Currently, only the default password
Rem profile is included, without the password complexity check and has
Rem the recommended audit settings. We will add the revokes from PUBLIC, and
Rem the password complexity checks.

-- Create password profile without a password complexity routine, for backward
-- compatibility. Add the routine if possible without breaking tests
```

12cR1 Security Configuration Script (3:3)

- $ORACLE_HOME/rdbms/admin/secconf.sql
- Creates the following audit policies
  - ORA_ACCOUNT_MGMT
  - ORA_DATABASE_PARAMETER
  - ORA_LOGON_FAILURES ACTIONS LOGON
  - ORA_SECURECONFIG
  - ORA_CIS_RECOMMENDATIONS

New SQL

SQL Object Changes (1:3)

- Identifiers can now be 128 bytes

```
SQL>
SQL> create table thisisatotallyobnoxiouslynamedtable (
  2  col1 DATE);

Table created.

SQL> desc thisisatotallyobnoxiouslynamedtable
 Name                                                      Null?    Type
 ----------------------------------------------------------------- --------
 COL1                                                               DATE
```

  - The Java Virtual Machine supports them too
  - Writing development guidelines immediately!
- Materialized Views
  - Real-Time Materialized Views
    Materialized views can be used for query rewrite even if they are not fully synchronized with the base tables and are considered stale
  - Statement-Level Refresh
    In addition to ON COMMIT and ON DEMAND refresh, the materialized join views can be refreshed when a DML operation takes place, without the need to commit such a transaction. This is predominantly relevant for star schema deployments

New SQL Functions (1:4)

- 12.1
  - APPROX_COUNT_DISTINCT
- 12.2
  - Enhanced
    - ListAgg can now handle string overflow conditions
  - New
    - APPROX_COUNT_DISTINCT_AGG
    - APPROX_COUNT_DISTINCT_DETAIL
    - APPROX_MEDIAN
    - APPROX_PERCENTILE
    - APPROX_PERCENTILE_AGG
    - APPROX_PERCENTILE_DETAIL
    - TO_APPROX_COUNT_DISTINCT
    - TO_APPROX_PERCENTILE

```
SQL> SELECT COUNT(DISTINCT(object_name)) FROM dba_objects;

COUNT(DISTINCT(OBJECT_NAME))
----------------------------
                       60311

SQL> SELECT APPROX_COUNT_DISTINCT(object_name) FROM dba_objects;

APPROX_COUNT_DISTINCT(OBJECT_NAME)
----------------------------------
                             60954
```

New SQL Functions (2:4)

- COLLATION (sorting)
- FEATURE_COMPARE (document comparison)
- JSON_DATAGUIDE (json)
- NLS_COLLATION_ID
- NLS_COLLATION_NAME
- ORA_DM_PARTITION_NAME (data mining)

```
SELECT 1-feature_compare(esa_wiki_mod
                         USING 'There are several PGA tour golfers from South Africa' text
                       AND USING 'Nick Price won the 2002 Mastercard Colonial Open' text)
                          AS SIMILARITY
FROM DUAL;


SIMILARITY
----------
      .258
```

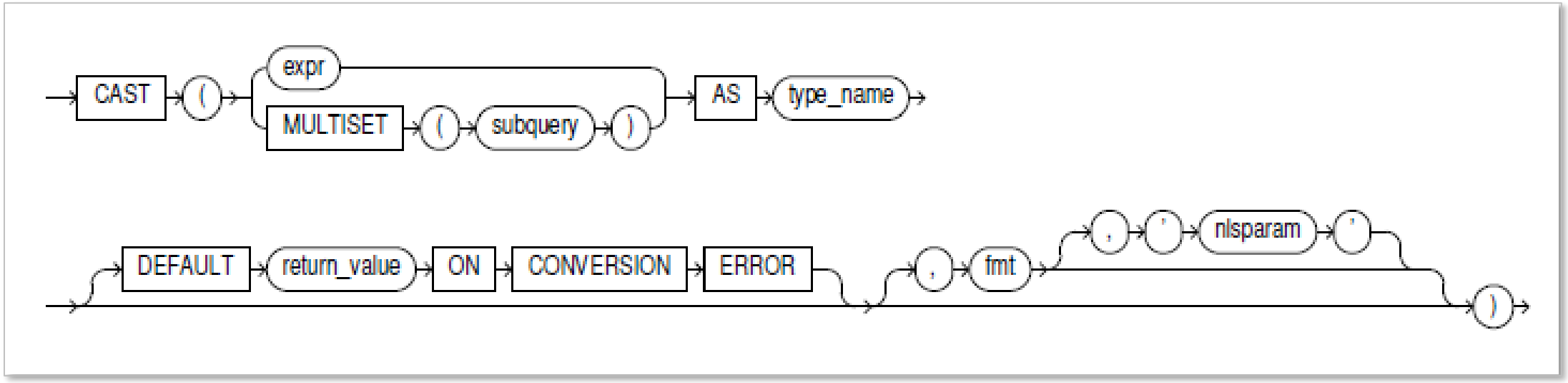
New SQL Functions (3:4)

- VALIDATE_CONVERSION

```
VALIDATE_CONVERSION(<expression> AS <expression_type> [format [, <'nls_parameter>']]);

SQL> SELECT VALIDATE_CONVERSION('$42.95' AS BINARY_FLOAT)
FROM DUAL;

VALIDATE_CONVERSION('$42.95'ASBINARY_FLOAT)
-------------------------------------------
                                          0

SQL> SELECT VALIDATE_CONVERSION('$42.95' AS BINARY_FLOAT, '$99D99')
FROM DUAL;

VALIDATE_CONVERSION('$42.95'ASBINARY_FLOAT,'$99D99')
----------------------------------------------------
                                                   1
```

New SQL Functions (4:4)

- Enhancing CAST Function With Error Handling



Enhanced JSON Support

- DataGuides for identifying JSON document structures
- Improved search indexes
- JSON docs can be created and altered directly from SQL and PL/SQL
- Path expressions have simplified syntax
- PL/SQL support for JSON operators
- Sharding Support

12.2: JSON DataGuides

- A data guide is a summary of the structural and type information contained in a set of JSON documents that record metadata about the fields used in those documents
- There are two formats for a data guide: flat and hierarchical available to SQL and PL/SQL as CLOB data
- JSON data-guide information can be saved persistently as part of the JSON search index infrastructure, and this information is updated automatically as new JSON content is added
- When you create a JSON search index
  - data-guide information is part of the index infrastructure
  - You can use a data guide:
    - As a basis for developing applications that involve data mining, business intelligence, or other analysis of JSON documents
    - As a basis for providing user assistance about requested JSON information, including search
    - To check or manipulate new JSON documents before adding them to a document
    - set (for example: validate, type-check, or exclude certain fields)

New PL/SQL

New PL/SQL (1:3)

- White Lists (ACCESSIBLE BY) Enhancements
  White lists can be defined for individual subprograms in a package. The ACCESSIBLE BY clause specifies a list of PL/SQL units that are considered safe to invoke the subprogram, and blocks all others.
- Binding PL/SQL-Only Data Types to SQL Statements Using DBMS_SQL
  - Release 12.1 introduced the ability to bind values of PL/SQL only data types, most notably PLS_INTEGER tables of records to SQL statements
  - There were some restrictions which are lifted in this release. The PL/SQL only data types can now be bound using the DBMS_SQL API and by invoking a C external procedure
  - This improvement brings the DBMS_SQL API in parity with the native dynamic SQL
- Improving the PL/SQL Debugger
- PL/Scope Reports on Static SQL Statements and Call Sites for Dynamic SQL
- New PL/SQL Pragma to Mark an Item as Deprecated
- New Pragmas
  - Coverage
  - Deprecation

New PL/SQL (2:3)

- Static PL/SQL Expressions Now Allowed Where Previously Literals Were Required
- Some examples of places where, in earlier releases, a literal was required are:
  - The length of the constraint in a VARCHAR2 declaration
  - The precision and scale in a NUMBER declaration
- Now you can use expressions, but the values must allow computation at compile time
- You can now write PL/SQL programs so that the intention is self-evident without comments
- You can also change PL/SQL programs to reflect changed requirements by making changes at far fewer sites
- The canonical example is the VARCHAR2 that holds the text of a simple SQL identifier
- This needs to be 128 bytes plus 2 additional bytes; 128 bytes for the name, and 2 bytes to enable double quoting.

New PL/SQL (3:3)

```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> DECLARE
  2    y VARCHAR2(20/1);
  3  BEGIN
  4    NULL;
  5  END;
  6  /
 y VARCHAR2(20/1);
           *
ERROR at line 2:
ORA-06550: line 2, column 13:
PLS-00491: numeric literal required
```

```
Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
With the Parationing, OLAP, Advanced Analytics and Real Application testing Options

SQL> DECLARE
  2    y VARCHAR2(20/5);
  3  BEGIN
  4    NULL;
  5  END;
  6  /

PL/SQL procedure successfully completed.
```

12.2: JavaScript Stored Procedures (1:2)

- 12.2
  - A new built-in package DBMS_JAVASCRIPT to allow what is referred to as "Java Stored Procedures"

```
-- create file and store at $ORACLE_HOME/rdbms/jlib
var Driver = Packages.oracle.jdbc.OracleDriver;
var oracleDriver = new Driver();
var url = "jdbc:default:connection:";   // server-side JDBC driver
var query ="SELECT first_name, last_name from employees";
// Establish a JDBC connection
var connection = oracleDriver.defaultConnection();
// Prepare statement
var preparedStatement = connection.prepareStatement(query);
// execute Query
var resultSet = preparedStatement.executeQuery();
// display results
     while(resultSet.next()) {
     print(resultSet.getString(1) + "== " + resultSet.getString(2) + " " );
     }
// cleanup
resultSet.close();
preparedStatement.close();
connection.close();
```

12.2: JavaScript Stored Procedures (2:2)

```
SQL> CREATE ROLE c##nashorn;

Role created.

SQL> exec dbms_java.grant_permission('C##NASHORN', 'SYS:java.lang.RuntimePermission', 'createClassLoader', '' );

Call completed.

SQL> exec dbms_java.grant_permission('C##NASHORN', 'SYS:java.lang.RuntimePermission', 'getClassLoader', '' );

PL/SQL procedure successfully completed.

SQL> exec dbms_java.grant_permission('C##NASHORN', 'SYS:java.util.logging.LoggingPermission', 'control', '' );

PL/SQL procedure successfully completed.

GRANT c##nashorn TO hr;

Grant succeeded.

SQL> exec dbms_java.loadjava('-v -r rdbms/jlib/database.js');

PL/SQL procedure successfully completed.

SQL> hr/hr@orcl
SQL> serveroutput on
SQL> exec dbms_java.set_output(80000);

PL/SQL procedure successfully completed.

SQL> exec dbms_javascript.run('rdbms/jlib/database.js');
Dan== Morgan

PL/SQL procedure successfully completed.

SQL> exec dbms_java.dropjava('-s rdbms/jlib/database.js');

PL/SQL procedure successfully completed.
```

New Built-in Packages

TWIN CITIES ORACLE USERS GROUP

Deprecated/Dropped Packages

- **12cR1**
  - SYS.BLAST_CUR
  - DVSYS.CODE$_PRIV
  - DVSYS.COMMAND_RULE$_PRIV
  - SYS.CWM2_OLAP_INSTALLER
  - SYS.DBMSNCDB
  - SYS.DBMS_AMD
  - SYS.DBMS_APPCTX
  - SYS.DBMS_DBLINK
  - SYS.DBMS_JDM_INTERNAL
  - SYS.DBMS_OWB
  - SYS.DBMS_RULE_COMPATIBLE_90
  - SYS.DBMS_SQL2
  - SYS.DBMS_XDB
  - SYS.DBMS_XDSUTL
  - SYS.DBMS_XS_MTCACHE_FFI
  - SYS.DBMS_XS_PRINCIPALS_INT
  - SYS.DBMS_XS_PRINCIPAL_EVENTS_INT
  - SYS.HTMLDB_SYSTEM
  - SYS.MGMT_BSLN
  - SYS.MGMT_BSLN_INTERNAL
  - SYS. MGMT_DM
  - SYS. MGMT_PREFERENCES
  - SYS. MGMT_TARGET
  - SYS.MGMT_USER
  - ODM_ABN_MODEL
  - OWA_DEBUG
  - OWA_DEBUG_DEMO
  - OWA_DEBUG_JDWP
  - OWA_DEBUG_PROFILER
  - OWA_DEBUG_TRACE
  - XS$CATVIEW_UTIL

- **12cR2**
  - DBMSOBJG
  - DBMSOBJG2
  - DBMSOBJGWRAPPER
  - DBMSOBJG_DP
  - DBMS_ASYNCRPC_PUSH
  - DBMS_CLR
  - DBMS_PREUP
  - Everything in Advanced Replication

DBMS_OFFLINE_SNAPSHOT
DBMS_OFFLINE_UTL
DBMS_RECTIFIER_DIF
DBMS_RECTIFIER_DIF_INTERNAL
DBMS_RECTIFIER_FRIENDS
DBMS_REPCAT_ADD_MASTER
DBMS_REPCAT_ADMIN
DBMS_REPCAT_AUTH
DBMS_REPCAT_CACHE
DBMS_REPCAT_COMMON_UTL
DBMS_REPCAT_CONF
DBMS_REPCAT_DECL
DBMS_REPCAT_DEFINER
DBMS_REPCAT_EXP
DBMS_REPCAT_FLA
DBMS_REPCAT_FLA_MAS
DBMS_REPCAT_FLA_UTL
DBMS_REPCAT_INSTANTIATE
DBMS_REPCAT_INTERNAL
DBMS_REPCAT_INTERNAL_PACKAGE
DBMS_REPCAT_MAS
DBMS_REPCAT_MIG
DBMS_REPCAT_MIGRATION
DBMS_REPCAT_MIG_INTERNAL
DBMS_REPCAT_OBJ_UTL
DBMS_REPCAT_OUTPUT
DBMS_REPCAT_RGT
DBMS_REPCAT_RGT_ALT
DBMS_REPCAT_RGT_CHK
DBMS_REPCAT_RGT_CUST
DBMS_REPCAT_RGT_CUST2
DBMS_REPCAT_RGT_EXP
DBMS_REPCAT_RGT_UTL
DBMS_REPCAT_RPC
DBMS_REPCAT_RPC_UTL
DBMS_REPCAT_RQ
DBMS_REPCAT_SNA
DBMS_REPCAT_SNA_INTERNAL
DBMS_REPCAT_SNA_UTL
DBMS_REPCAT_SQL_UTL
DBMS_REPCAT_UNTRUSTED
DBMS_REPCAT_UTL
DBMS_REPCAT_UTL2
DBMS_REPCAT_UTL3
DBMS_REPCAT_UTL4
DBMS_REPCAT_VALIDATE
DBMS_REPUTIL2

12.1 New Documented Packages (1:2)

- DBMS_APP_CONT package with 1 object
- DBMS_AUTO_REPORT package with 7 objects
- DBMS_HEAT_MAP package with 6 objects
- DBMS_ILM package with 10 objects
- DBMS_ILM_ADMIN package with 3 objects
- DBMS_INMEMORY package with 11 objects
- DBMS_PART package with 2 objects
- DBMS_PDB package with 14 objects
- DBMS_PERF package with 7 objects
- DBMS_PRIVILEGE_CAPTURE package with 5 objects
- DBMS_REDACT package with 8 objects
- DBMS_ROLLING package with 8 objects
- DBMS_SPD package with 9 objects
- DBMS_SQL_MONITOR package with 6 objects

12.1 New Documented Packages (2:2)

- DBMS_SQL_TRANSLATOR package with 18 objects
- DBMS_SYNC_REFRESH with 14 objects
- DBMS_TSDP_MANAGE with 9 documented objects
- DBMS_TSDP_PROTECT with 10 objects
- UTL_CALL_STACK package with 13 objects

12.2 New Documented Packages

- SYS.DBMS_AUDIT_UTIL with 4 functions
- SYS.DBMS_DBCOMP with 1 procedure
- SYS.DBMS_HADOOP with 2 procedures
- SYS.DBMS_HIERARCHY with 2 procs and 2 functions
- SYS.DBMS_INMEMORY_ADMIN with 9 objects
- SYS.DBMS_JAVASCRIPT with 1 proc
- SYS.DBMS_JSON with 14 objects
- SYS.DBMS_MVIEW_STATS with 3 objects: 1 overloaded
- SYS.DBMS_PDB_ALTER_SHARING with 7 objects
- SYS.DBMS_PLSQL_CODE_COVERAGE with 3 objects
- SYS.DBMS_PROCESS with 3 objects
- SYS.DBMS_UMF with 16 objects

12.2 DBMS_AUDIT_UTIL

- A valuable enhancement to auditing database activities supporting governance and security
- Enables formatting the output of queries to the DBA_FGA_AUDIT_TRAIL, DBA_AUDIT_TRAIL, UNIFIED_AUDIT_TRAIL, and V$XML_AUDIT_TRAIL views so that the output appears in separate rows

```
SELECT db_user, object_name, sql_text, rls_predicate, rls_policy_type, rls_policy_owner, rls_policy_name
FROM TABLE(
  dbms_audit_util.decode_rls_info_atrail_fga(
  CURSOR(SELECT * FROM dba_fga_audit_trail)));


SELECT object_name, sql_text, rls_predicate, rls_policy_type, rls_policy_owner, rls_policy_name
FROM TABLE(
  dbms_audit_util.decode_rls_info_atrail_xml(
  CURSOR (SELECT * FROM v$xml_audit_trail)));
```

12.2 DBMS_DBCOMP

- Assumes that a primary database and one or more Data Guard physical standby databases are deployed. The databases should be at least mounted or open before block comparison is run
- Logical standby databases, Far Sync instances, and cascaded standbys cannot be the target database

```
exec dbms_dbcomp.dbcomp('ALL', '/home/oracle/lost_write_check.txt', TRUE);

-- in a separate SQL*Plus session
SELECT target_desc, sofar, totalwork
FROM v$session_longops
WHERE opname = 'BlockCompare';

TARGET_DESC                    SO FAR  TOTALWORK
------------------------------ --------- ----------
Compared Blocks                367104     403142
Lost Writes                         0          0
```

12.2 DBMS_GOLDENGATE_ADM (1:2)

- Interfaces to configure automatic conflict detection and resolution in an Oracle GoldenGate configuration that replicates tables between Oracle databases
- When more than one replica of a table allows changes to the table, a conflict can occur when a change is made to the same row in two different databases at nearly the same time
- GoldenGate replicates changes using row logical change records (LCRs)
- It detects a conflict by comparing the old values in the row LCR with the current values of the corresponding table row identified by the key columns
- If any column value does not match, then there is a conflict
- After a conflict is detected, GoldenGate can resolve the conflict by overwriting values in the row with some values from the row LCR, ignoring the values in the row LCR, or computing a delta to update the row values

12.2 DBMS_GOLDENGATE_ADM (2:2)

- ADD_AUTO_CDR_COLUMN_GROUP
  - Adds a column group and configures GoldenGate automatic conflict detection and resolution for the column group

```
dbms_goldengate_adm.add_auto_cdr_column_group(
schema_name              IN VARCHAR2,
table_name               IN VARCHAR2,
column_list              IN VARCHAR2,
column_group_name        IN VARCHAR2                  DEFAULT NULL,
existing_data_timestamp IN TIMESTAMP WITH TIME ZONE DEFAULT NULL);
```

```
exec dbms_goldengate_adm.alter_auto_cdr_column_group('UWCLASS', 'SERVERS', 'LATITUDE,LONGITUDE', 'POSITION_COLGRP');
```

12.2 DBMS_INMEMORY_ADMIN

- Provides interfaces for managing In-Memory Expressions (IM expressions) and the In-Memory FastStart (IM FastStart) area
- Analytic queries often contain complex expressions or calculations that can consume significant CPU and memory during execution
- Use DBMS_INMEMORY_ADMIN procedure to identify these frequently used (“hot”) expressions and populate them in the IM column store
- Avoids repeated computations and improves performance

```
dbms_inmemory_admin.faststart_enable(
tbs_name  IN VARCHAR2,
nologging IN BOOLEAN DEFAULT TRUE);
```

```
SQL> exec dbms_inmemory_admin.faststart_enable('USERS', FALSE);

PL/SQL procedure successfully completed.
```

12.2 DBMS_JSON

- Provides an interface for DataGuide operations for those working with Java Script Object Notation inside the Oracle database

Object	Description
ADD_VIRTUAL_COLUMNS	Add virtual columns based on data-guide information. This has no effect when running on the shard catalog server - no virtual column is added.
CREATE_VIEW	Create a view with relational columns and scalar JSON fields as specified in a data guide.
CREATE_VIEW_ON_PATH	Create a view based on data-guide information, with relational columns, top-level scalar types, and fully expanded subtree under a given path. When running on the shard catalog server this raises an error stating that the data guide is empty.
DROP_VIRTUAL_COLUMNS	Drop virtual columns created by procedure ADD_VIRTUAL_COLUMNS. This has no effect when running on the shard catalog server.
GET_INDEX_DATAGUIDE	Get JSON data guide from a data guide enabled JSON search index. When running on the shard catalog server this returns a single empty row as result.
RENAME_COLUMN	Set the preferred name for a view column or a virtual column creating using a data guide. This has no effect when running on the shard catalog server.

```
???
```

12.2 DBMS_PDB_ALTER_SHARING

- This package can set a database object to one of the following types of common objects in a PDB: data-linked object, extended data-linked object, or metadata-linked object
- An application can be migrated to CDB$ROOT or to an application PDB. For example, an application can be migrated from an application installed in a PDB plugged into a 12.1 CDB to a PDB in a 12.2 CDB
- Subprograms
  - REMOVE_LINK
  - SET_DATA_LINKED
  - SET_EXT_DATA_LINKED
  - SET_METADATA_LINKED
  - SET_PROFILE_EXPLICIT
  - SET_ROLE_EXPLICIT
  - SET_USER_EXPLICIT

```
exec dbms_pdb_alter_sharing.set_metadata_linked('C##UWCLASS','ACCOUNTS', 1);
```

12.2 DBMS_PROCESS

- Provides an interface to manage the pre-spawned servers
- By default, Oracle Database can pre-spawn foreground processes to improve the performance of client connections
- A pre-spawned process refers to a process that has been spawned but does not have a session yet
- When a user connects to the database or a service process is needed, the process performs further initialization as needed
- To manage foreground processes, use the DBMS_PROCESS package
- The procedures in this package configure the number of foreground processes for a connection pool, start a connection pool, and stop a connection pool

```
dbms_process.configure_pool(
pool_name   IN VARCHAR2       DEFAULT SYS_DEFAULT_FOREGROUND_POOL',
min_count   IN BINARY_INTEGER DEFAULT 10,
batch_count IN BINARY_INTEGER DEFAULT 20,
init_count  IN BINARY_INTEGER DEFAULT 0);
```

```
exec dbms_process.configure_pool('UW_FRGRND_POOL', 50, 100, 0);
exec dbms_process.start_pool('UW_FRGRND_POOL');
```

Performance Tuning

TWIN CITIES ORACLE USERS GROUP

Adaptive Execution Plans (1:4)

- A set of capabilities that enables the adaptive optimizer to make run-time adjustments to execution plans and discover additional information that can lead to better optimizer statistics
- Helpful when existing statistics are not sufficient to generate an optimal plan
- An adaptive plan is one that learns about the data as it is executing

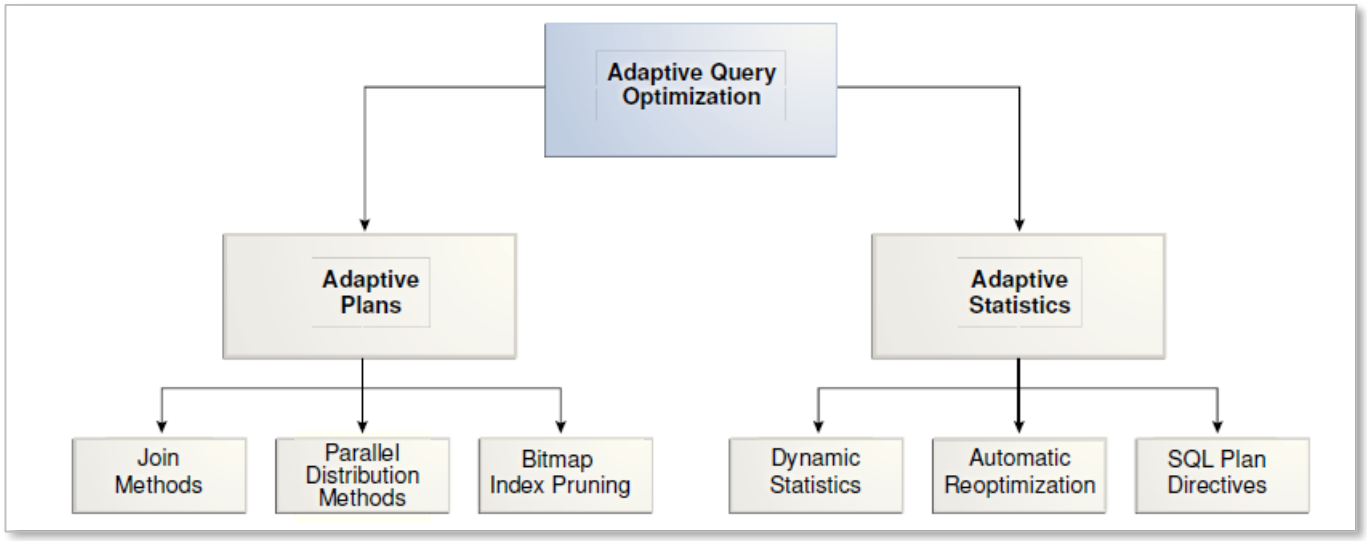
```
NAME                                 TYPE        VALUE
------------------------------------ ----------- -----------
optimizer_adaptive_features          boolean     TRUE
optimizer_adaptive_reporting_only    boolean     FALSE
optimizer_capture_sql_plan_baselines boolean     FALSE
optimizer_dynamic_sampling           integer     2
optimizer_features_enable            string      12.1.0.2
optimizer_index_caching              integer     0
optimizer_index_cost_adj             integer     100
optimizer_mode                       string      ALL_ROWS
optimizer_secure_view_merging        boolean     TRUE
optimizer_use_invisible_indexes      boolean     FALSE
optimizer_use_pending_statistics     boolean     FALSE
optimizer_use_sql_plan_baselines     boolean     TRUE
```

```
DECLARE
 i NUMBER;
 j NUMBER;
 k CLOB;
BEGIN
  dbms_feature_adaptive_plans(i, j, k);
  dbms_output.put_line('1: ' || i);
  dbms_output.put_line('2: ' || j);
  dbms_output.put_line('3: ' || k);
END;
/
1: 1
2:
3: Total number of queries: 501
Number of queries with an adaptive plan: 35
Percentage of queries with an adaptive plan:
6.98602794411177644710578842315369261477
Are the queries running in reporting mode ? : No
```

Adaptive Execution Plans (2:4)

- An option available to the DBMS_XPLAN built-in package is through the use of the format constant ADAPTIVE that
  - Displays the final plan, or the current plan if the execution has not completed
    - This section includes notes about runtime optimizations that affect the plan, such as switching from a Nested Loops join to a Hash join
  - Plan lineage
    - This section shows the plans that were run previously due to automatic reoptimization
    - It also shows the default plan, if the plan changed due to dynamic plans
  - Recommended plan
    - In reporting mode, the plan is chosen based on execution statistics displayed
    - Note that displaying the recommended plan for automatic reoptimization requires re-compiling the query with the optimizer adjustments collected in the child cursor
    - Displaying the recommended plan for a dynamic plan does not require this
  - Dynamic plans
    - This summarizes the portions of the plan that differ from the default plan chosen by the optimizer

Adaptive Execution Plans (3:4)



- The 12.1 parameter **OPTIMIZER_ADAPTIVE_FEATURES** has been made OBSOLETE (i.e. must be removed from the SPFILE when upgrading) in Oracle Database 12.2

```
-- 12.1.0.2
SQL> show parameter adaptive

NAME                                 TYPE     VALUE
------------------------------------ -------- ------
optimizer_adaptive_features          boolean  TRUE
optimizer_adaptive_reporting_only    boolean  FALSE
parallel_adaptive_multi_user         boolean  TRUE
```

```
-- 12.2.0.1
SQL> show parameter adaptive

NAME                                 TYPE     VALUE
------------------------------------ -------- ------
optimizer_adaptive_plans             Boolean  TRUE
optimizer_adaptive_reporting_only    Boolean  FALSE
optimizer_adaptive_statistics        Boolean  FALSE
parallel_adaptive_multi_user         Boolean  FALSE
```

Adaptive Execution Plans (4:4)

« OOW 2016 Update | Main | New Oracle Optimizer... »

Optimizer Adaptive Features in Oracle Database 12c Release 2

By nbayliss-Oracle on Oct 12, 2016

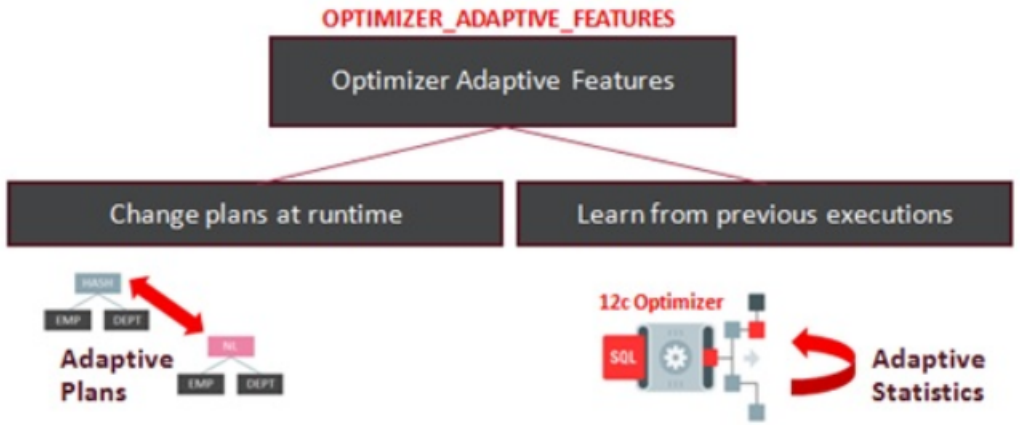
Introduction

In Oracle Database 12c Release 2 we have changed the way optimizer adaptive features can be controlled. In this post, I'll present what has changed and give you guidance on how to you can choose what settings to use.

These changes are also relevant for Oracle Database 12c Release 1. If you want to know more about that, there's information at end of this post.

What's Changed

In Oracle Database 12c Release 1, the database parameter *optimizer_adaptive_features* controls all of the adaptive features like this:



About

The Oracle Optimizer blog is written by members of the Optimizer development team. The goal of this blog is to provide an insight into the workings of the Optimizer and the statistics it relies on. The views expressed on this blog are our own and do not necessarily reflect the views of Oracle and its affiliates. The views and opinions expressed by visitors on this blog are theirs solely and may not reflect ours.

Search

Enter search term:

Recent Posts

Efficient Statistics Maintenance for Partitioned Tables Using Incremental Statistics – Part 3

Optimizer Adaptive Features and Upgrading to Oracle Database 12c Release 2

New Optimizer Statistics White Paper

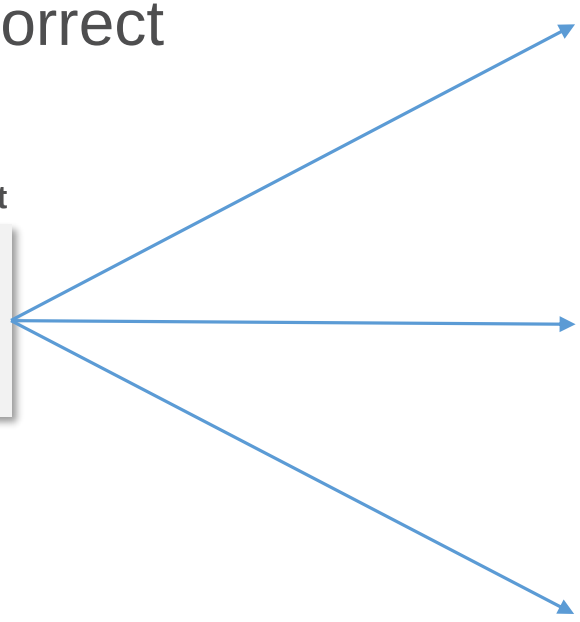
https://blogs.oracle.com/optimizer/entry/optimizer_adaptive_features_in_the
http://docs.oracle.com/database/121/TGSQL/tgsql_optcncpt.htm#TGSQL192

Explain Plan

- With every version the optimizer makes different decision even when the query is based upon the same data with the same organization
- It has taken years ... but finally Oracle has the math correct

The SQL Statement

```
SELECT srvr_id
FROM servers
INTERSECT
SELECT srvr_id
FROM serv_inst;
```




```
-------------------------------------------------------------------------------
| Id | Operation              | Name         | Rows | Bytes | Cost (%CPU)|
-------------------------------------------------------------------------------
|  0 | SELECT STATEMENT       |              |  141 | 4560  |     6  (84)|
|  1 |  INTERSECTION          |              |      |       |            |
|  2 |   SORT UNIQUE NOSORT   |              |  141 |  564  |     2  (50)|
|  3 |    INDEX FULL SCAN     | PK_SERVERS   |  141 |  564  |     1   (0)|
|  4 |   SORT UNIQUE          |              |  999 | 3996  |     4  (25)|
|  5 |    INDEX FAST FULL SCAN| IX_SERV_INST |  999 | 3996  |     3   (0)|
-------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------
| Id  | Operation              | Name         | Rows | Bytes | Cost (%CPU)|
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |              |  141 |  4560 |    20  (10)|
|   1 |  INTERSECTION          |              |      |       |            |
|   2 |   SORT UNIQUE          |              |  141 |   564 |    10  (10)|
|   3 |    TABLE ACCESS FULL   | SERVERS      |  141 |   564 |     9   (0)|
|   4 |   SORT UNIQUE          |              |  999 |  3996 |    10  (10)|
|   5 |    TABLE ACCESS FULL   | SERV_INST    |  999 |  3996 |     9   (0)|
-------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------
| Id  | Operation               | Name         | Rows | Bytes | Cost (%CPU)|
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT        |              |  141 |  4560 |     6  (34)|
|   1 |  INTERSECTION           |              |      |       |            |
|   2 |   SORT UNIQUE NOSORT    |              |  141 |   564 |     2  (50)|
|   3 |    INDEX FULL SCAN      | PK_SERVERS   |  141 |   564 |     1   (0)|
|   4 |   SORT UNIQUE           |              |  999 |  3996 |     4  (25)|
|   5 |    INDEX FAST FULL SCAN | PK_SERV_INST |  999 |  3996 |     3   (0)|
-------------------------------------------------------------------------------
```

more examples: www.morganslibrary.org/reference/explain_plan.html

High Availability

12cR2: New HA Features

- Reasoned "What If" Commands
  - Provided a set of evaluation commands and APIs to determine the impact of a certain operation before executing the operation
  - The reasoned What-If command evaluation feature provides the rationale behind the policy decisions and explains the entities involved, their attributes, and the criteria used to arrive at each of the potential actions
  - Why-If command evaluations help applications, cluster, and system administrators involved in capacity planning and configuration management to set up and test resource management policies
- Server Weight-Based Node Eviction
  - Acts as a tie-breaker mechanism in situations where Oracle Clusterware needs to evict a particular node or a group of nodes from a cluster
  - The server weight-based node eviction mechanism helps to identify the node or the group of nodes to be evicted based on additional information about the load on those servers
- Load-Aware Resource Placement
  - Prevents overloading a server with more applications than the server is capable of running

New Miscellaneous

Miscellaneous New Features (1:2)

- APEX 5.0
- Extensibility
  - DBAs no longer need to manually manage composite domain indexes
  - Enhanced extensible indexing framework
- Globalization
  - Case-Insensitive Database
  - Column-Level Collation (sorting ordering)
  - Default Character Set = AL32UTF8
  - Unicode 7.0 supported
- In-Memory
  - Performance enhancements for JSON

Miscellaneous New Features (2:2)

- SQL*Plus
  - New ability to reissue commands similar to command history in the UNIX shell
  - New commands
    - SET PREFETCH
    - SET LOBPREFETCH
    - SET STATEMENTCACHE

```
SET STATEMENTCACHE <0 | 32767>

SET STATEMENTCACHE 50
```

- Workspace Manager
  - Data changes enhanced so control change propagation
  - Enhanced ability to defer deletions

Break

Oracle Multitenant: The Road Ahead

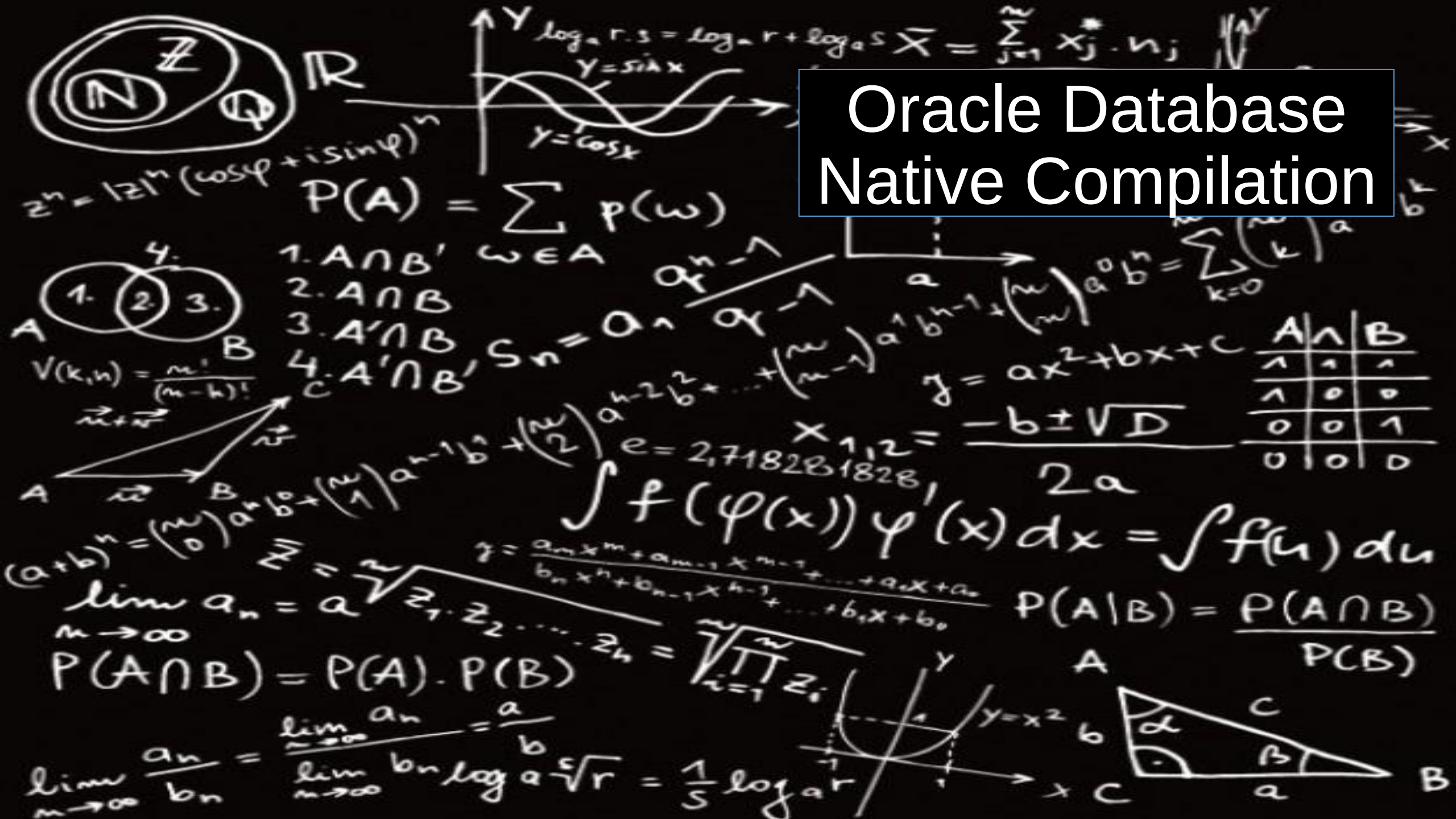
Patrick Wheeler, Oracle

Lightning Session: Native Compilation in PL/SQL

Dan Morgan, Meta7

Lightning Sessions: Something New

- A short presentation 5-15 minutes
- A phenomenal way to share your knowledge and experience with our group and our community
- A great way to develop your presentation skills
- I can lead workshops to help our members develop their presentations and presentation skills


Oracle Database
Native Compilation

PL/SQL Compilation

- By default the Oracle database installs with an initialization parameter that instructs it to compile PL/SQL into **p**ortable machine **code** (pcode)
- Native compilation instructs the compiler to compile PL/SQL as C

```
SQL> conn sys@pdbdev as sysdba
Connected.

SQL> show parameter plsql

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------
plsql_ccflags                        string
plsql_code_type                      string      INTERPRETED
plsql_debug                          boolean     FALSE
plsql_optimize_level                 integer     2
plsql_v2_compatibility               boolean     FALSE
plsql_warnings                       string      ENABLE:ALL

SQL> ALTER SYSTEM SET plsql_code_type = 'NATIVE' SCOPE=BOTH;

System altered.

SQL> show parameter plsql_code

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------
plsql_code_type                      string      NATIVE
```

Native Compilation: Why and When

- What you are going to see has been in Oracle since at least version 10.1
- You can write in-database executable C without having to include stdio
- Code execution speed has no impact on I/O
- The speed of code execution has a substantial effect on math operations
- Utilize native compilation when a performance improvement can be achieved by improving calculation speed

- PCODE can be exported and imported by DataPump without recompilation
  PCODE can be backed up and restored by RMAN without recompilation
- NATIVE code may require a recompilation after import or recovery
  - You must check
  - You may need a NATIVE recompilation of invalid objects

Proof of Concept (1:5)

- Create two identical functions

```
SQL> conn uwclass/uwclass@pdbdev
Connected.

SQL> CREATE OR REPLACE FUNCTION factorial_interpreted(p_n NUMBER) RETURN NUMBER AUTHID DEFINER IS
  2  BEGIN
  3    IF (p_n = 1) THEN
  4      RETURN 1;
  5    ELSE
  6      RETURN factorial_interpreted(p_n-1) * p_n;
  7    END IF;
  8  END factorial_interpreted;
  9  /

Function created.

SQL>
SQL> CREATE OR REPLACE FUNCTION factorial_native(p_n NUMBER) RETURN NUMBER AUTHID DEFINER IS
  2  BEGIN
  3    IF (p_n = 1) THEN
  4      RETURN 1;
  5    ELSE
  6      RETURN factorial_native(p_n-1) * p_n;
  7    END IF;
  8  END factorial_native;
  9  /

Function created.
```

Proof of Concept (2:5)

- Recompile the factorial_native function into C

```
SQL> col comp_mode format a25

SQL> SELECT o.object_name, o.object_type, s.param_value comp_mode
  2  FROM user_stored_settings s, user_objects o
  3  WHERE o.object_id = s.object_id
  4  AND param_name = 'plsql_compiler_flags'
  5  AND o.object_name LIKE 'FACTOR%';

OBJECT_NAME                    OBJECT_TYPE                COMP_MODE
------------------------------ -------------------------- -------------------------
FACTORIAL_NATIVE               FUNCTION                   INTERPRETED,NON_DEBUG
FACTORIAL_INTERPRETED          FUNCTION                   INTERPRETED,NON_DEBUG

SQL> ALTER FUNCTION factorial_native
  2  COMPILE PLSQL_CODE_TYPE=NATIVE REUSE SETTINGS;

Function altered.

SQL> SELECT o.object_name, o.object_type, s.param_value comp_mode
  2  FROM user_stored_settings s, user_objects o
  3  WHERE o.object_id = s.object_id
  4  AND param_name = 'plsql_compiler_flags'
  5  AND o.object_name LIKE 'FACTOR%';

OBJECT_NAME                    OBJECT_TYPE                COMP_MODE
------------------------------ -------------------------- -------------------------
FACTORIAL_INTERPRETED          FUNCTION                   INTERPRETED,NON_DEBUG
FACTORIAL_NATIVE               FUNCTION                   NATIVE,NON_DEBUG
```

Proof of Concept (3:5)

- Compare the performance looping 10,000 times

```
SQL> set serveroutput on

SQL> DECLARE
  2   l_start NUMBER;
  3   l_n     NUMBER;
  4  BEGIN
  5    l_start := dbms_utility.get_time;
  6    FOR i IN 1 .. 10000 LOOP
  7      l_n := factorial_interpreted(50);
  8    END LOOP;
  9    dbms_output.put_line('Interpreted: ' || (dbms_utility.get_time-l_start) || ' hsecs...' || l_n);
 10    l_start := dbms_utility.get_time;
 11    FOR i IN 1 .. 10000 LOOP
 12      l_n := factorial_native(50);
 13    END LOOP;
 14    dbms_output.put_line('Native:      ' || (dbms_utility.get_time-l_start) || ' hsecs...' || l_n);
 15* END;
SQL> /
Interpreted: 11 hsecs...30414093201713378043612608166064768844300000000000000000000000000
Native:       9 hsecs...30414093201713378043612608166064768844300000000000000000000000000

PL/SQL procedure successfully completed.
```

Proof of Concept (4:5)

- Rerun the test 5 more times

```
SQL> /
Interpreted: 12 hsecs...3041409320171337804361260816606476884430000000000000000000000000
Native:       6 hsecs...3041409320171337804361260816606476884430000000000000000000000000

PL/SQL procedure successfully completed.

SQL> /
Interpreted: 11 hsecs...3041409320171337804361260816606476884430000000000000000000000000
Native:       6 hsecs...3041409320171337804361260816606476884430000000000000000000000000

SQL> /
Interpreted: 11 hsecs...3041409320171337804361260816606476884430000000000000000000000000
Native:       6 hsecs...3041409320171337804361260816606476884430000000000000000000000000

PL/SQL procedure successfully completed.

SQL> /
Interpreted: 12 hsecs...3041409320171337804361260816606476884430000000000000000000000000
Native:       6 hsecs...3041409320171337804361260816606476884430000000000000000000000000

PL/SQL procedure successfully completed.
```

- The results show a consistent pattern ... Interpreted compilation takes more time to perform the same work as Native compilation

SIMPLE_INTEGER (1:5)

- In the previous two functions the IN parameter was defined as type NUMBER
- The NUMBER data type is optimized for SQL and PL/SQL not for C
- Let's try this again optimized by creating a third function with the SIMPLE_INTEGER data type

```
SQL> CREATE OR REPLACE FUNCTION factorial_simple(p_n SIMPLE_INTEGER) AUTHID DEFINER RETURN NUMBER IS
  2  BEGIN
  3    IF (p_n = 1) THEN
  4      RETURN 1;
  5    ELSE
  6      RETURN factorial_simple(p_n-1) * p_n;
  7    END IF;
  8  END factorial_simple;
  9  /

Function created.
```

SIMPLE_INTEGER (2:5)

- A SIMPLE_INTEGER variable cannot be NULL

```
SQL> DECLARE
  2   x SIMPLE_INTEGER;
  3  BEGIN
  4    NULL;
  5  END;
  6  /
x SIMPLE_INTEGER;
   *
ERROR at line 2:
ORA-06550: line 2, column 4:
PLS-00218: a variable declared NOT NULL must have an initialization assignment
```

SIMPLE_INTEGER (3:5)

- Create the function and recompile as NATIVE

```
SQL> CREATE OR REPLACE FUNCTION factorial_simple(p_n SIMPLE_INTEGER) AUTHID DEFINER RETURN NUMBER IS
  2  BEGIN
  3    IF (p_n = 1) THEN
  4      RETURN 1;
  5    ELSE
  6      RETURN factorial_simple(p_n-1) * p_n;
  7    END IF;
  8  END factorial_simple;
  9  /

Function created.

SQL> ALTER FUNCTION factorial_simple COMPILE PLSQL_CODE_TYPE=NATIVE REUSE SETTINGS;

Function altered.

SELECT o.object_name, o.object_type, s.param_value comp_mode
FROM user_stored_settings s, user_objects o
WHERE o.object_id = s.object_id
AND param_name = 'plsql_compiler_flags'
AND o.object_name LIKE 'FACTOR%';

OBJECT_NAME                    OBJECT_TYPE                 COMP_MODE
------------------------------ --------------------------- -------------------------
FACTORIAL_INTERPRETED          FUNCTION                    INTERPRETED,NON_DEBUG
FACTORIAL_NATIVE               FUNCTION                    NATIVE,NON_DEBUG
FACTORIAL_SIMPLE               FUNCTION                    NATIVE,NON_DEBUG
```

SIMPLE_INTEGER (4:5)

- Performance the POC test with 10,000 loops

```
SQL> DECLARE
  2   l_start NUMBER;
  3   l_n     NUMBER;
  4  BEGIN
  5    l_start := dbms_utility.get_time;
  6    FOR i IN 1 .. 10000 LOOP
  7      l_n := factorial_simple(50);
  8    END LOOP;
  9    dbms_output.put_line('Simple:      ' || (dbms_utility.get_time-l_start) || ' hsecs...' || l_n);
 10* END;
 11  /
Simple:      6 hsecs...30414093201713378043612608166064768844300000000000000000000000000

PL/SQL procedure successfully completed.

SQL> /
Simple:      6 hsecs...30414093201713378043612608166064768844300000000000000000000000000

PL/SQL procedure successfully completed.

SQL> /
Simple:      4 hsecs...30414093201713378043612608166064768844300000000000000000000000000

PL/SQL procedure successfully completed.
```

SIMPLE_INTEGER (5:5)

- Increase the number of loops to 100,000

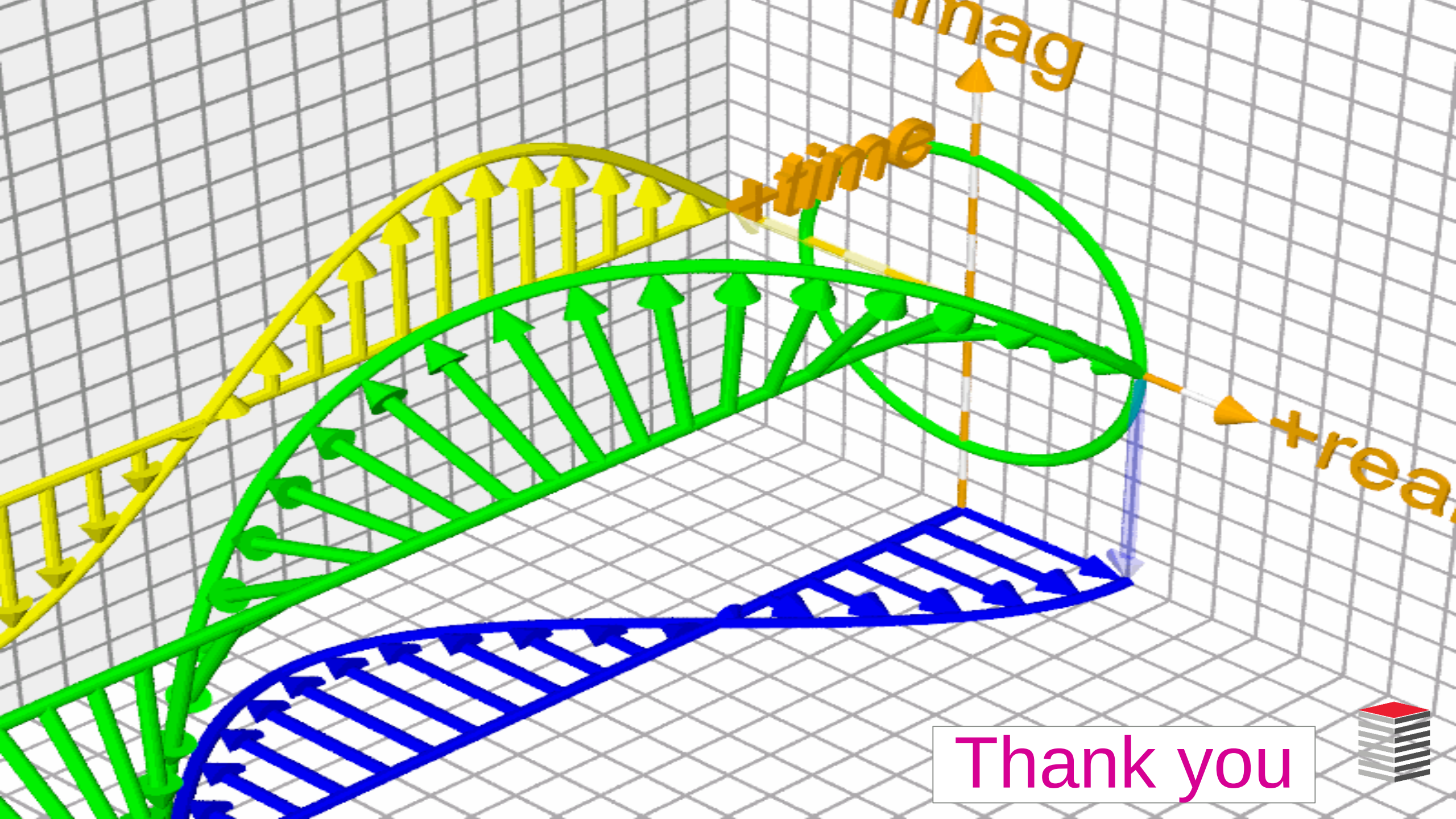
```
SQL> /
Interpreted: 111 hsecs...3041409320171337804361260816606476884430000000000000000000000000
Native:        8 hsecs...3041409320171337804361260816606476884430000000000000000000000000
Simple:        7 hsecs...3041409320171337804361260816606476884430000000000000000000000000
```

- Did you see what happened? Look at the 10,000 loop results again

```
SQL> /
Interpreted: 11 hsecs...3041409320171337804361260816606476884430000000000000000000000000
Native:       6 hsecs...3041409320171337804361260816606476884430000000000000000000000000
```

- Increasing the number of loops by an order of magnitude increased the run time of the Interpreted pcode by an order of magnitude (10X) while only increasing the time of the natively compiled code by 33% ... a staggering performance improvement

- **Note:** I find that in 12.1.0.2 SIMPLE_INTEGER does not produce the consist improvement seen in 11.2.0.3 and I am looking forward to seeing how SIMPLE_INTEGER behaves in the March GA release of 12.2.0.1



Thank you

Time	Description
08:30 - 09:00	Registration and Networking
09:00 - 09:30	Welcoming Remarks, Sponsor and Speaker Introductions
09:30 - 10:30	What's new in 12.2: Dan Morgan
10:30 - 10:45	Break
10:45 - 11:45	Oracle Multitenant: The Road Ahead: Patrick Wheeler, Oracle
11:45 - 12:00	Lightning Session: Native Compilation in PL/SQL: Dan Morgan
12:00 - 01:00	**Lunch**
01:00 - 01:15	**Lightning Session: Platinum Gateway Security Concerns, Brian Bream, Collier IT**
01:15 - 02:00	**Q & A**
02:00 - 03:00	**Oracle Multitenant: Ideal Architecture for World-Class SaaS: Patrick Wheeler, Oracle**
03:00 - 03:15	**Break**
03:15 - 03:45	**Oracle Multitenant: Live in SQL*Plus: Dan Morgan**
03:45 - 04:00	**Giveaways, Prizes, and Closing Remarks**

Lunch

Lightning Session: Platinum Gateway Security Concerns

Brian Bream, Collier IT

Q&A

Patrick Wheeler, Brian Bream, Dan Morgan

Oracle Multitenant: Ideal Architecture for World-Class SaaS

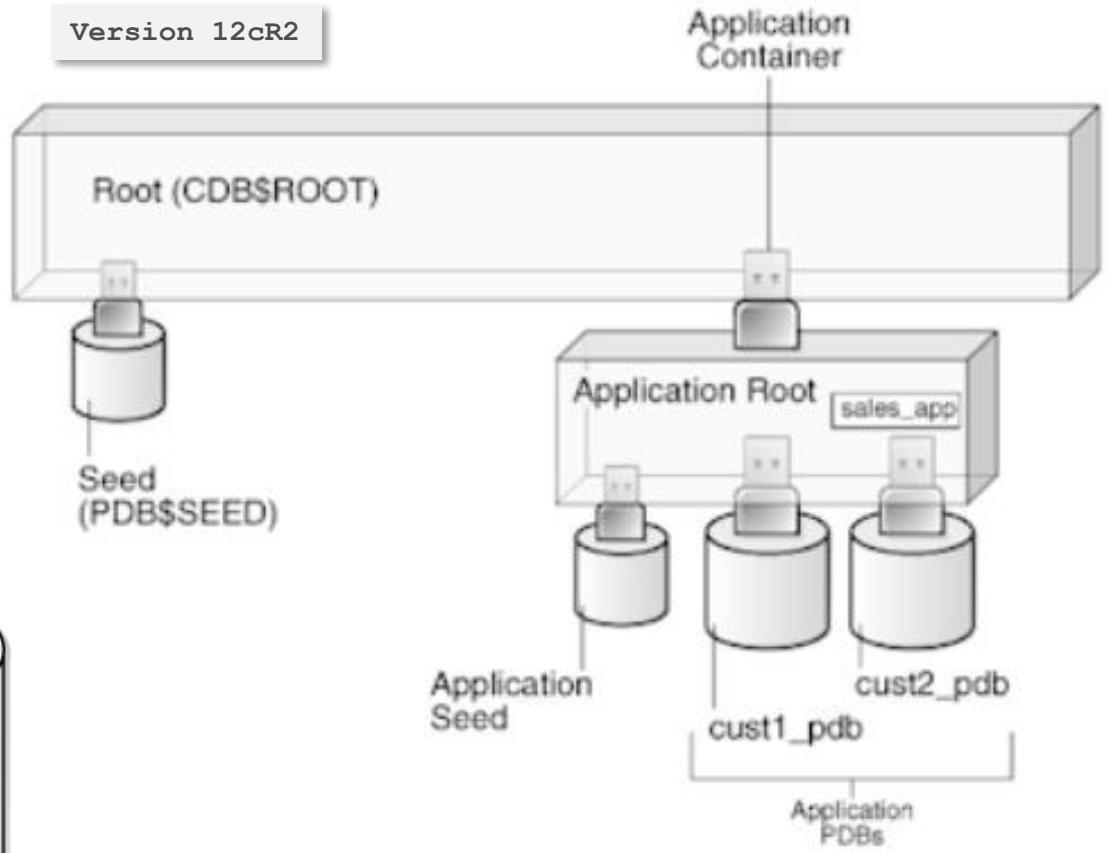
Patrick Wheeler, Oracle

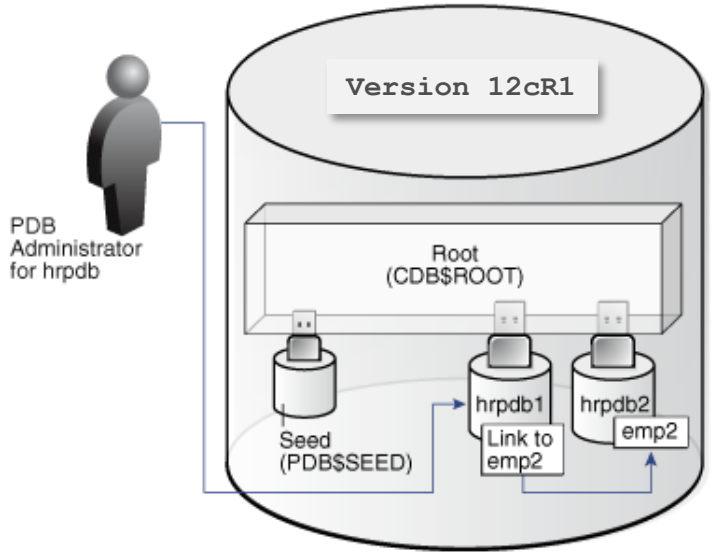
Break

Oracle Multitenant: Live in SQL*Plus

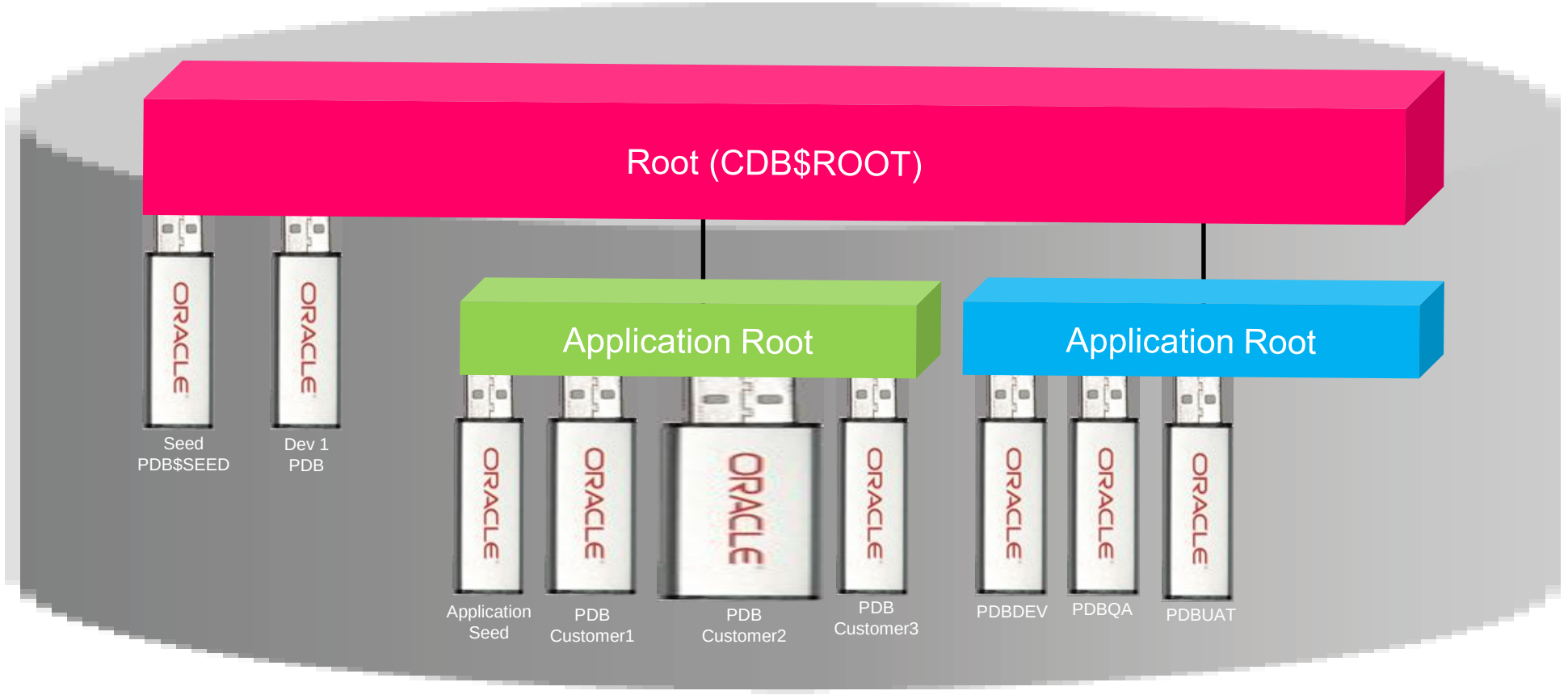
Dan Morgan

New 12cR2 Container Database Architecture





New 12cR2 Container Database Architecture



Create Application Root Container

Application Containers Demo 1: Create Application Root (1:4)

```
CREATE PLUGGABLE DATABASE <pdb_name> AS <APPLICATION CONTAINER | SEED>
ADMIN USER <admin_user_name> IDENTIFIED BY <password>
[ROLES = (<comma_delimited_list_of_roles>)]
[PARALLEL <integer>]
[DEFAULT TABLESPACE <tablespace_name>]
[<pdb_storage_clause>]
[<file_name_convert_clause>]
[<service_name_convert_clause>]
[<path_prefix_clause>]
[TEMPFILE REUSE]
[<user_tablespace_clause>]
[<standby_database_clause>]
[<LOGGING | NOLOGGING | FILESYSTEM_LIKE_LOGGING>]
[<create_file_dest_clause>]
[HOST = '<host_name>']
[PORT = <port_number>];
```

Application Containers Demo 1: Create Application Root (2:4)

```
conn sys@orabase as sysdba
Enter password:
Connected.

sho con_id

CON_ID
------------------------------
1

show con_name

CON_NAME
------------------------------
CDB$ROOT

SELECT name, open_mode, application_root,
application_pdb, application_seed, pdb_count
FROM v$containers
ORDER BY con_id;

NAME       OPEN_MODE  APP APP APP PDB_COUNT
---------- ---------- --- --- --- ----------
CDB$ROOT   READ WRITE NO  NO  NO           2
PDB$SEED   READ ONLY  NO  NO  NO           0
PDBDEV     READ WRITE NO  NO  NO           0
```

```
-- as desirable as it would be to do so you cannot use
special characters in a PDB name
CREATE PLUGGABLE DATABASE uwapp_root
AS APPLICATION CONTAINER
ADMIN USER uwAdmin IDENTIFIED BY uwAdmin
ROLES = (CDB_DBA)
DEFAULT TABLESPACE uwapp_tbs
FILE_NAME_CONVERT = ('/pdbseed/', '/uwapp/')
USER_TABLESPACES = NONE
LOGGING;

Pluggable database created.

SELECT name, open_mode, application_root,
application_pdb, application_seed, pdb_count
FROM v$containers
ORDER BY con_id;

NAME       OPEN_MODE  APP APP APP PDB_COUNT
---------- ---------- --- --- --- ----------
CDB$ROOT   READ WRITE NO  NO  NO           3
PDB$SEED   READ ONLY  NO  NO  NO           0
PDBDEV     READ WRITE NO  NO  NO           0
UWAPP_ROOT MOUNTED    YES NO  NO           0

ALTER PLUGGABLE DATABASE uwapp_root OPEN;

Pluggable database altered.
```

Application Containers Demo 1: Create Application Root (3:4)

```
SELECT name, creation_date, clb_goal, pdb
FROM v$services
ORDER BY 1;

NAME             CREATION_DATE        CLB_G PDB
---------------- -------------------- ----- ------------
SYS$BACKGROUND   26-JAN-2017 13:54:44 SHORT CDB$ROOT
SYS$USERS        26-JAN-2017 13:54:44 SHORT CDB$ROOT
uwapp_root       26-MAR-2017 17:09:28 LONG  UWAPP$ROOT
pdbdev           02-MAR-2017 07:57:37 LONG  PDBDEV
orabase          02-MAR-2017 07:52:46 LONG  CDB$ROOT
orabaseXDB       02-MAR-2017 07:52:46 LONG  CDB$ROOT

ALTER SESSION SET CONTAINER=uwapp_root;

Session altered.

sho con_id

CON_ID
-------
     4
show con_name

CON_NAME
------------------------------
UWAPP$ROOT
```

Application Containers Demo 1: Create Application Root (4:4)

```
SELECT tablespace_name TBS_NAME, file_name
FROM dba_data_files
UNION
SELECT tablespace_name, file_name
FROM dba_temp_files
ORDER BY 1;

TBS_NAME FILE_NAME
-------- ---------------------- ------------------------------------------------------
SYSAUX    /u01/app/oracle/oradata/orabase/uwapp/sysaux01.dbf
SYSTEM    /u01/app/oracle/oradata/orabase/uwapp/system01.dbf
TEMP      /u01/app/oracle/oradata/orabase/uwapp/temp012017-03-02_07-53-20-031-AM.dbf
UNDOTBS1  /u01/app/oracle/oradata/orabase/uwapp/undotbs01.dbf
```

Application Installation

Application Containers Demo 2: Application Installation (1:5)

```
ALTER PLUGGABLE DATABASE APPLICATION
{ { app_name
{ BEGIN INSTALL 'app_version' [ COMMENT 'comment' ]
| END INSTALL [ 'app_version' ]
| BEGIN PATCH number [ MINIMUM VERSION 'app_version' ] [ COMMENT
'comment' ]
| END PATCH [ number ]
| BEGIN UPGRADE 'start_app_version' TO 'end_app_version' [ COMMENT
'comment' ]
| END UPGRADE [ TO 'end_app_version' ]
| BEGIN UNINSTALL
| END UNINSTALL
| SET PATCH number
| SET VERSION 'app_version'
| SET COMPATIBILITY VERSION { 'app_version' | CURRENT }
| SYNC }
|
{ ALL SYNC }
}
```

Application Containers Demo 2: Application Installation (2:5)

```
ALTER PLUGGABLE DATABASE APPLICATION uw_app BEGIN INSTALL '1.0';

Pluggable database altered.

CREATE TABLESPACE uwapp_tbs
DATAFILE '/u01/app/oracle/oradata/orcl12c/uwapp/uwapp_tbs'
SIZE 25M AUTOEXTEND ON NEXT 25M;

Tablespace created.

-- create user
CREATE USER uwapp_user IDENTIFIED BY uwapp_user
DEFAULT TABLESPACE uwapp_tbs
TEMPORARY TABLESPACE temp
QUOTA UNLIMITED ON uwapp_tbs;

User created.

GRANT create session TO uwapp_owner;
GRANT create procedure TO uwapp_owner
GRANT create table TO uwapp_owner;
GRANT create view TO uwapp_owner;

Grant succeeded.
```

Application Containers Demo 2: Application Installation (3:5)

```
CREATE OR REPLACE PROCEDURE uwapp_owner.who_am_i AUTHID DEFINER
IS
BEGIN
  dbms_output.put_line('I do not know');
END who_am_i;
/

CREATE TABLE uwapp_owner.t1 (
tid       NUMBER(10),
last_name VARCHAR2(20));

ALTER TABLE uwapp_owner.t1
ADD PRIMARY KEY (tid);

CREATE TABLE uwapp_owner.t2(
tid NUMBER(10),
last_name VARCHAR2(20));

ALTER TABLE uwapp_owner.t2
ADD PRIMARY KEY (tid);

CREATE VIEW uwapp_owner.t1t2_view AS
SELECT t1.tid, t2.last_name
FROM uwapp_user.t1, uwapp_user.t2
WHERE t1.tid = t2.tid;
```

Application Containers Demo 2: Application Installation (4:5)

```
INSERT INTO uwapp_owner.t1 VALUES (1, 'MORGAN');
INSERT INTO uwapp_owner.t1 VALUES (2, 'KYTE');
INSERT INTO uwapp_owner.t1 VALUES (3, 'LEWIS');
INSERT INTO uwapp_owner.t2 VALUES (1, 'TOWNSEND');
INSERT INTO uwapp_owner.t2 VALUES (2, 'KURIAN');
COMMIT;

SELECT * FROM uwapp_user.t1t2_view;

SQL> SELECT * FROM uwapp_user.t1t2_view;

TID         LAST_NAME
---------- --------------------
         1 TOWNSEND
         2 KURIAN

ALTER PLUGGABLE DATABASE APPLICATION uw_app END INSTALL;

Pluggable database altered.
```

Application Containers Demo 2: Application Installation (5:5)

```
col app_name format a10
col app_version format a12

SELECT app_name, app_version, app_status, app_implicit
FROM dba_applications
WHERE app_name = 'UW_APP';

APP_NAME   APP_VERSION  APP_STATUS   A
---------- ------------ ------------ -
UW_APP     1.0          NORMAL       N

-- if there are application PDBs, under the application root
-- container, exit synchronize them with their root the
-- instructions for doing so can be found below under ALTER
-- APPLICATION PDB
```

Application Containers Demo 3: Sharable Objects (1:3)

```
SQL> CREATE TABLE servers (
  2  srvr_id NUMBER(10),
  3  network_id NUMBER(10),
  4  status VARCHAR2(1),
  5  latitude FLOAT(20),
  6  longitude FLOAT(20),
  7  netaddress VARCHAR2(15));

Table created.

SQL> CREATE TABLE serv_inst
  2  SHARING=METADATA (
  3  siid NUMBER(10),
  4  si_status VARCHAR2(15),
  5  type VARCHAR2(5),
  6  installstatus VARCHAR2(1),
  7  location_code NUMBER(10),
  8  custacct_id VARCHAR2(10),
  9  srvr_id NUMBER(10),
 10* ws_id NUMBER(10));
SHARING=METADATA
*
ERROR at line 9:
ORA-00922: missing or invalid option

SQL> show parameter default_sharing

NAME                 TYPE        VALUE
-------------------- ----------- ---------
default_sharing      string      METADATA
```

Application Containers Demo 3: Sharable Objects (2:3)

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION uw_app BEGIN INSTALL '1.0';
ALTER PLUGGABLE DATABASE APPLICATION uw_app BEGIN INSTALL '1.0'
*
ERROR at line 1:
ORA-65221: application UW_APP exists already

SQL> ALTER PLUGGABLE DATABASE APPLICATION uw_app
  2  BEGIN UPGRADE '1.0' TO '2.0'
  3  COMMENT 'Adding New Table With Sharing';

SQL> CREATE TABLE serv_inst
  2  SHARING=METADATA (
  3  siid NUMBER(10),
  4  si_status VARCHAR2(15),
  5  type VARCHAR2(5),
  6  installstatus VARCHAR2(1),
  7  location_code NUMBER(10),
  8  custacct_id VARCHAR2(10),
  9  srvr_id NUMBER(10),
 10* ws_id NUMBER(10));

Table created.
```

Application Containers Demo 3: Sharable Objects (3:3)

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION uw_app END UPGRADE;

SQL desc dba_applications

SQL> desc dba_applications
Name                      Null? Type
------------------------- -------- --------------
APP_NAME                           VARCHAR2(128)
APP_ID                             NUMBER
APP_VERSION                        VARCHAR2(30)
APP_STATUS                         VARCHAR2(12)
APP_IMPLICIT                       VARCHAR2(1)
APP_CAPTURE_SERVICE                VARCHAR2(64)
APP_CAPTURE_MODULE                 VARCHAR2(64)

col app_name format a37

SQL> SELECT app_name, app_id, app_version, app_status, app_implicit
  2  FROM dba_applications;

APP_NAME                                APP_ID  APP_VERSION  APP_STATUS   A
------------------------------------- ------- ------------ ------------ -
APP$4BAF1A01C5964D55E0530100007F821B         2 1.0          NORMAL       Y
UW_APP                                      21 2.0          NORMAL       N
```

Application Seed Creation

Application Containers Demo 4: Create Seed (1:5)

```
CREATE PLUGGABLE DATABASE AS SEED FROM uwapp_root
ADMIN USER uwappAdmin IDENTIFIED BY uwappAdmin
FILE_NAME_CONVERT = ('/uwapp/', '/uwappseed/')
LOGGING;

SQL> SELECT con_id, name, open_mode, application_root, application_pdb,
application_seed
  2  FROM v$containers
  3  ORDER BY con_id;

CON_ID  NAME             OPEN_MODE   APP APP APP
------- ---------------- ----------- --- --- ---
     4  UWAPP_ROOT       READ WRITE  YES NO  NO
     6  UWAPP_ROOT$SEED  MOUNTED     NO  YES YES

SQL> ALTER PLUGGABLE DATABASE uwapp_root$seed OPEN;

Warning: PDB altered with errors.

SQL> ALTER PLUGGABLE DATABASE uwapp_root$seed OPEN READ ONLY;

Warning: PDB altered with errors.
```

Application Containers Demo 4: Create Seed (2:5)

```
SQL> host
[oracle@vbgeneric ~]$ cd $ORACLE_BASE/diag/orabase/orabase/trace
[oracle@vbgeneric trace]$ tail alert_orabase.log
UWAPP_ROOT$SEED(6):Opatch validation is skipped for PDB UWAPP_ROOT$SEED (con_id=0)
UWAPP_ROOT$SEED(6):************************************************************
UWAPP_ROOT$SEED(6):WARNING: Pluggable Database UWAPP_ROOT$SEED with pdb id - 6 is
UWAPP_ROOT$SEED(6): altered with errors or warnings. Please look into
UWAPP_ROOT$SEED(6): PDB_PLUG_IN_VIOLATIONS view for more details.
UWAPP_ROOT$SEED(6):************************************************************
2017-04-19T18:45:58.662039-04:00
UWAPP_ROOT$SEED(6):Opening pdb with no Resource Manager plan active
Pluggable database UWAPP_ROOT$SEED opened read only
UWAPP_ROOT(4):Completed: ALTER PLUGGABLE DATABASE uwapp_root$seed OPEN READ ONLY
```

```
[oracle@vbgeneric ~]$exit
SQL> col time format a29
SQL> col name format a16
SQL> col type format a6
SQL> col cause format a30
SQL> col action format a22

SQL> SELECT time, name, cause, status, action
  2* FROM pdb_plug_in_violations;

TIME                          NAME             CAUSE                                    STATUS   ACTION
----------------------------- ---------------- ---------------------------------------- -------- ----------------------
19-APR-17 06.45.57.958082 PM  UWAPP_ROOT$SEED  Non-Application PDB to Application PDB  PENDING  Run pdb_to_apppdb.sql.
```

Application Containers Demo 4: Create Seed (3:5)

```
SQL> @?/rdbms/admin/pdb_to_apppdb.sql
SQL> SET FEEDBACK 1
SQL> SET NUMWIDTH 10
SQL> SET LINESIZE 80
SQL> SET TRIMSPOOL ON
SQL> SET TAB OFF
SQL> SET PAGESIZE 100
SQL>
SQL> WHENEVER SQLERROR EXIT;
SQL>
SQL> VARIABLE cdbname VARCHAR2(128)
SQL> VARIABLE pdbname VARCHAR2(128)
SQL> VARIABLE appname VARCHAR2(128)
SQL> BEGIN
2 -- Disallow script in non-CDB
3 SELECT sys_context('USERENV', 'CDB_NAME')
4 INTO :cdbname
5 FROM dual
6 WHERE sys_context('USERENV', 'CDB_NAME') is not null;
7 -- Disallow script in CDB Root
8 -- Disallow script in PDB$SEED (Bug 22550952)
9 SELECT sys_context('USERENV', 'CON_NAME')
10 INTO :pdbname
11 FROM dual
12 WHERE sys_context('USERENV', 'CON_NAME') <> 'CDB$ROOT'
13 AND sys_context('USERENV', 'CON_NAME') <> 'PDB$SEED';
14 -- Disallow script outside of Application Container
15 SELECT sys_context('USERENV', 'APPLICATION_NAME')
16 INTO :appname
17 FROM dual
18 WHERE sys_context('USERENV', 'APPLICATION_NAME') is not null;
19 -- Disallow script in Proxy PDB (Bug 22550952). This query works
20 -- because remote mapping in Proxy PDB has been disabled using
21 -- the underscore parameter.
22 SELECT /*+ OPT_PARAM('_ENABLE_VIEW_PDB', 'FALSE') */ name
23 INTO :pdbname
24 FROM v$pdbs
25 WHERE proxy_pdb='NO';
26 END;
27 /
BEGIN
*
ERROR at line 1:
ORA-01422: exact fetch returns more than requested number of rows
ORA-06512: at line 22

Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
[oracle@vbgeneric ~]$
```

Application Containers Demo 4: Create Seed (4:5)

- On dissecting the pdb_to_apppdb.sql script the following was identified as the root cause of the error

```
SQL> SELECT /*+ OPT_PARAM('_ENABLE_VIEW_PDB', 'FALSE') */ name
  2  FROM v$pdbs
  3  WHERE proxy_pdb='NO';

NAME
------------------------------
UWAPP_ROOT
UWAPP_ROOT$SEED
```

- Further examination of the script(s) found numerous examples of this

```
select TO_NUMBER('NOT_IN_APPLICATION_PDB') from v$pdbs
where con_id=sys_context('USERENV', 'CON_ID') and application_pdb<>'YES';
```

- Clearly the script has no choice but to fail

- More news on this in our Slack group as it becomes available

TNSNAMES.ORA

TNSNAMES Configuration

- Every time you add a new PDB ... you must also make a manual entry to TNSNAMES.ORA

```
# tnsnames.ora Network Configuration File:
C:\app\oracle\product\12.1.0\dbhome_1\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

PDBDEV =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 127.0.0.1)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = pdbdev)
    )
  )

PDBTEST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 127.0.0.1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdbtest)
    )
  )

ORACLR_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
    (CONNECT_DATA =
      (SID = CLRExtProc)
      (PRESENTATION = RO)
    )
  )
```

```
PDBPROD =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 127.0.0.1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdbprod)
    )
  )

ORABASE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 127.0.0.1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orabase)
    )
  )
```

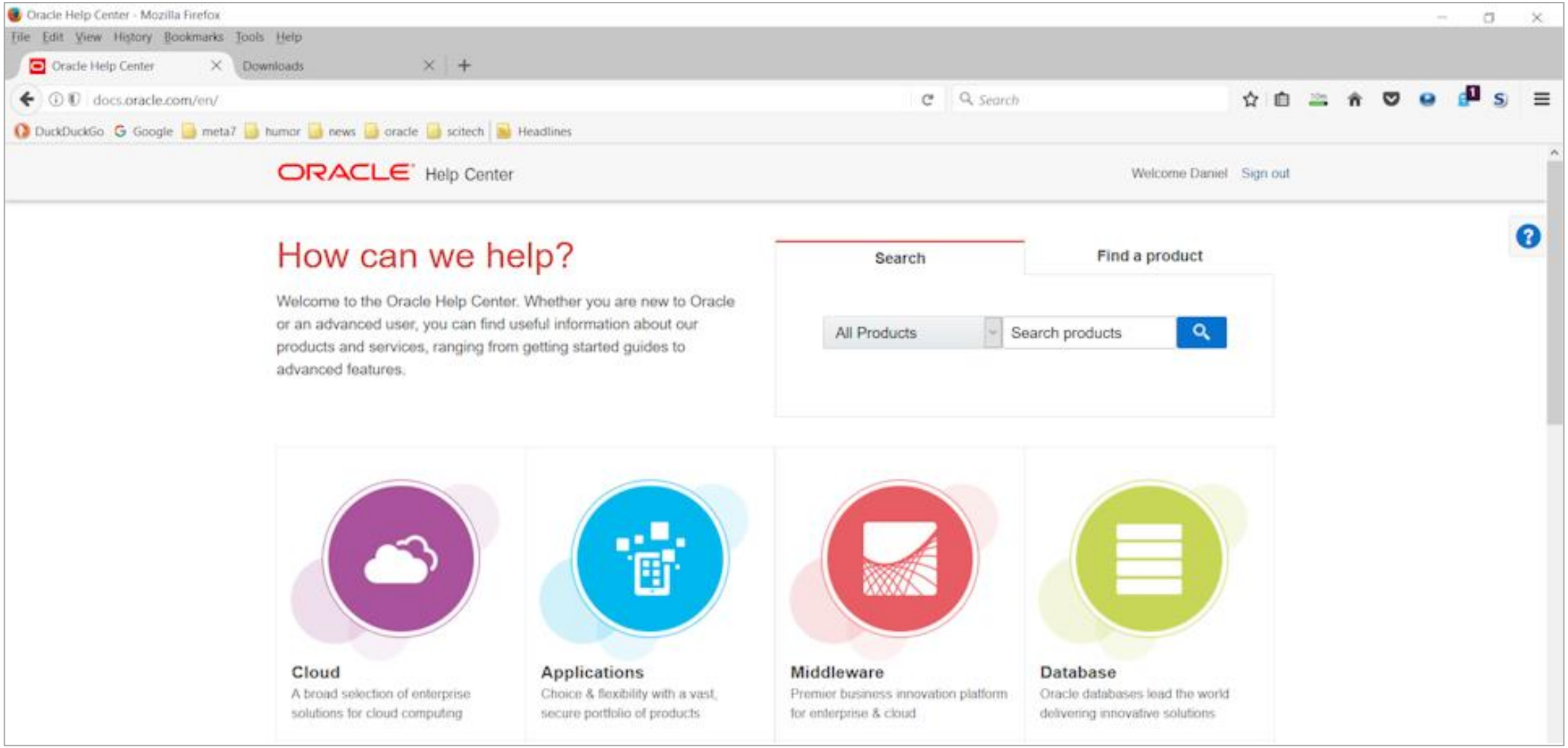
Giveaways, Prizes & Closing Remarks

Conclusion

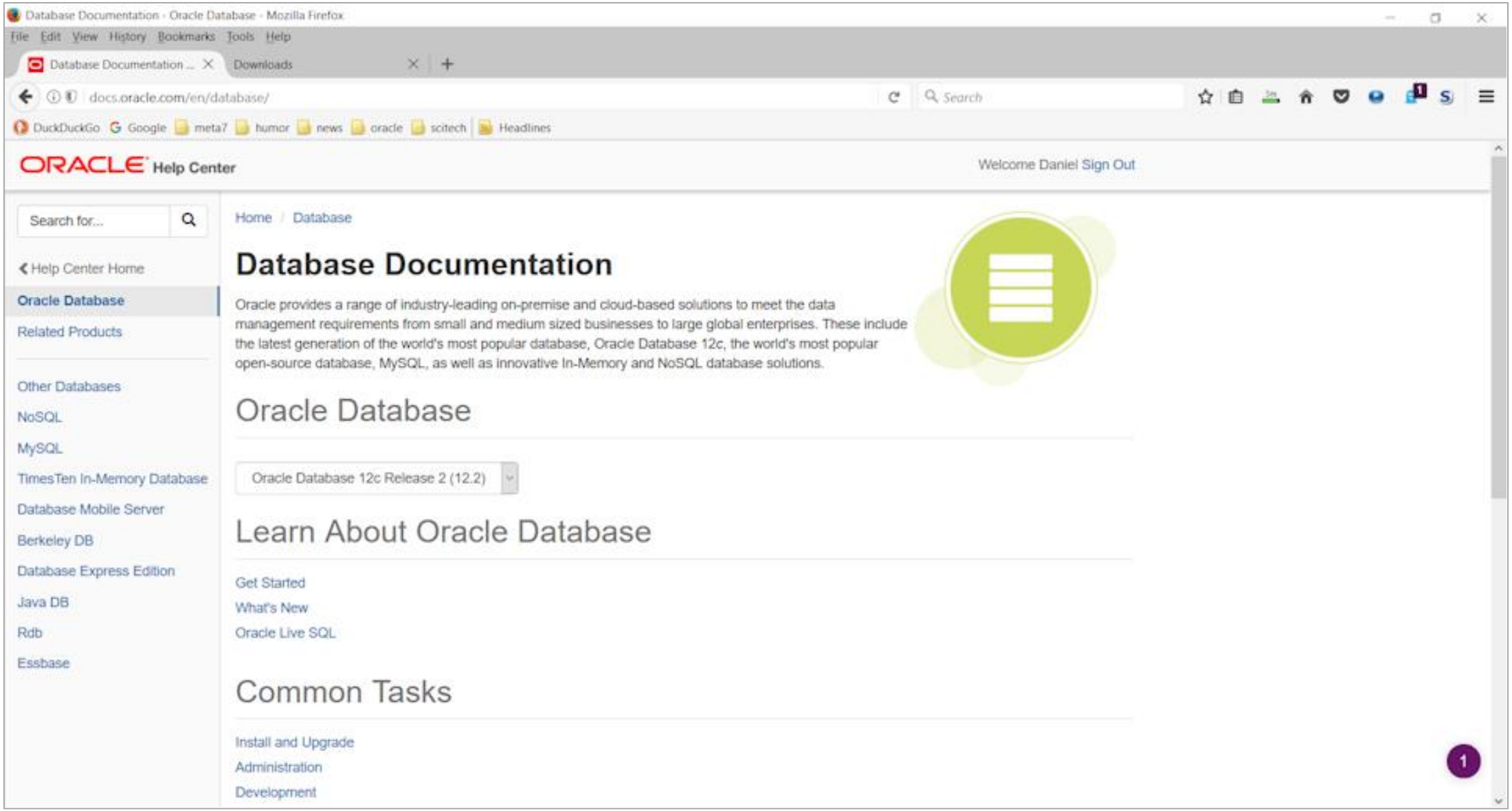
- Read the docs
- Download it
- Install it
- Learn it

- For demos that work in SQL*Plus
  www.morganslibrary.org/library.html

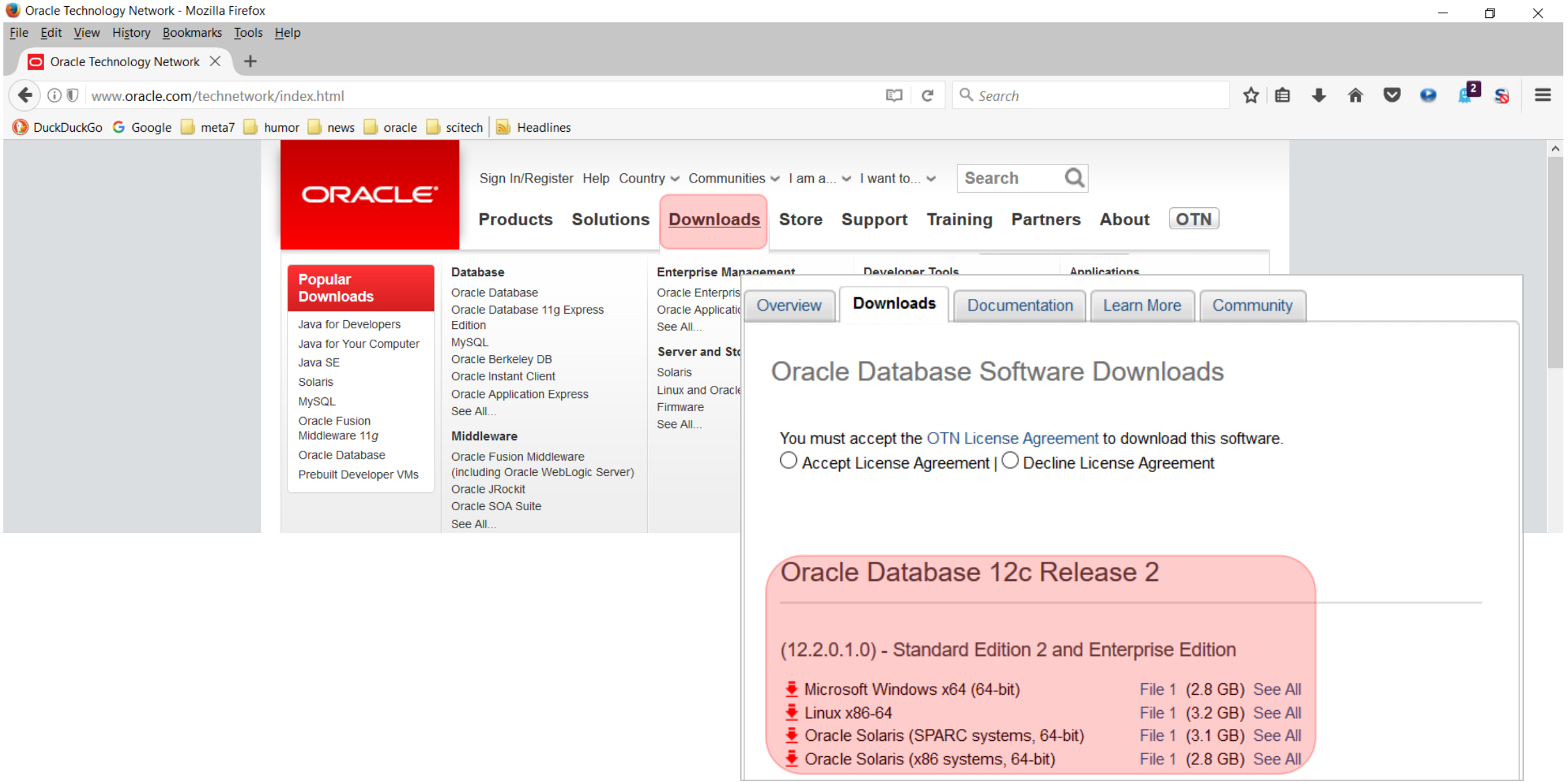
Docs: https://docs.oracle.com (1:2)



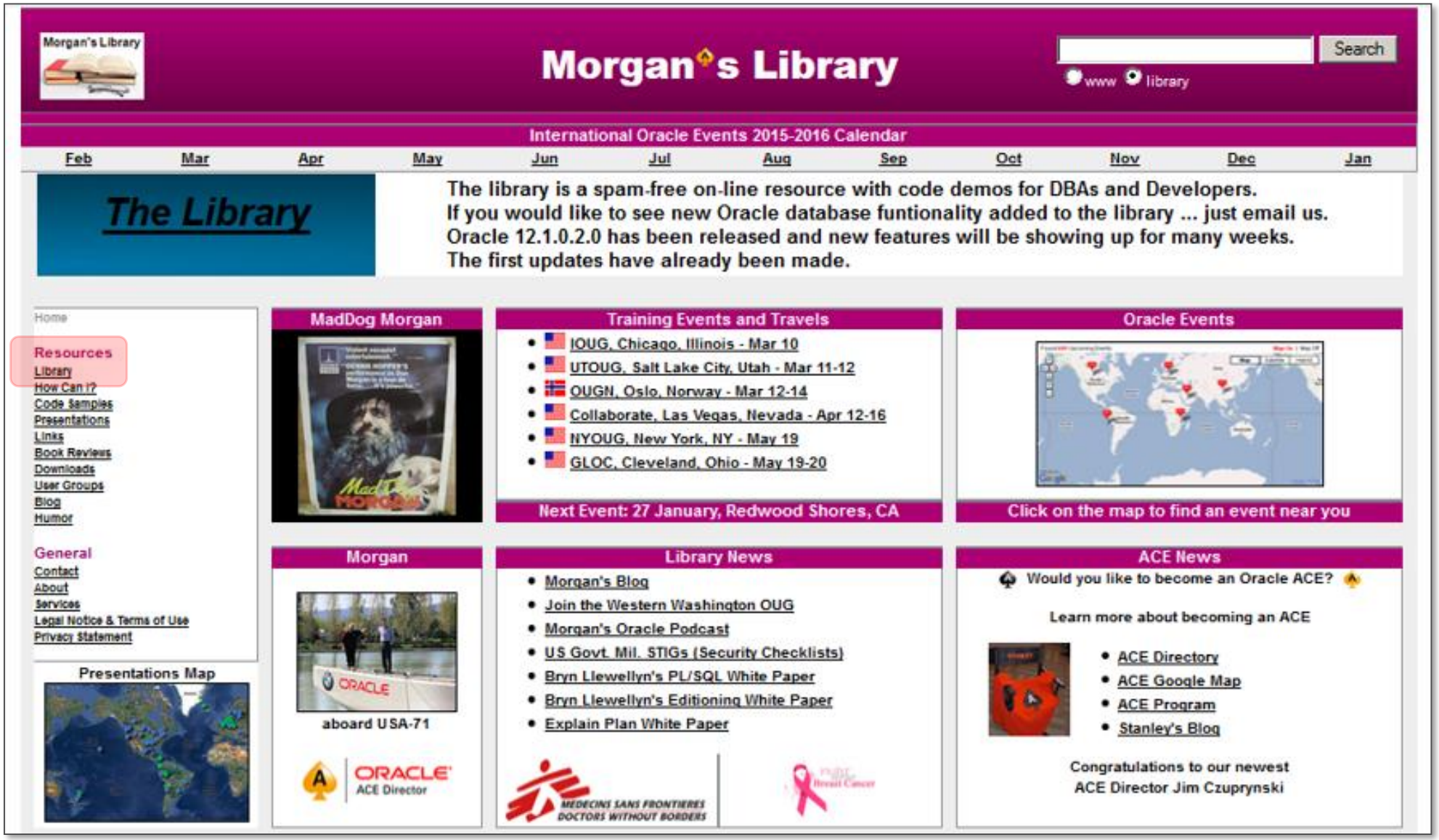
Docs: https://docs.oracle.com (2:2)



Downloads: https://otn.oracle.com



My Websites: Morgan's Library



www.morganslibrary.org

TWIN CITIES ORACLE USERS GROUP

I hope today has been valuable for you and I encourage you to contact me ... vp@toug.org ... with suggestions and to volunteer to present at our July meeting

Daniel A. Morgan
mobile: +1 206-669-2949