Oracle Database 12c
The Most Valuable Under-promoted Features










Daniel A. Morgan
email: dmorgan@forsythe.com
mobile: +1 206-669-2949
skype: damorgan11g
twitter: @meta7solutions

Introduction

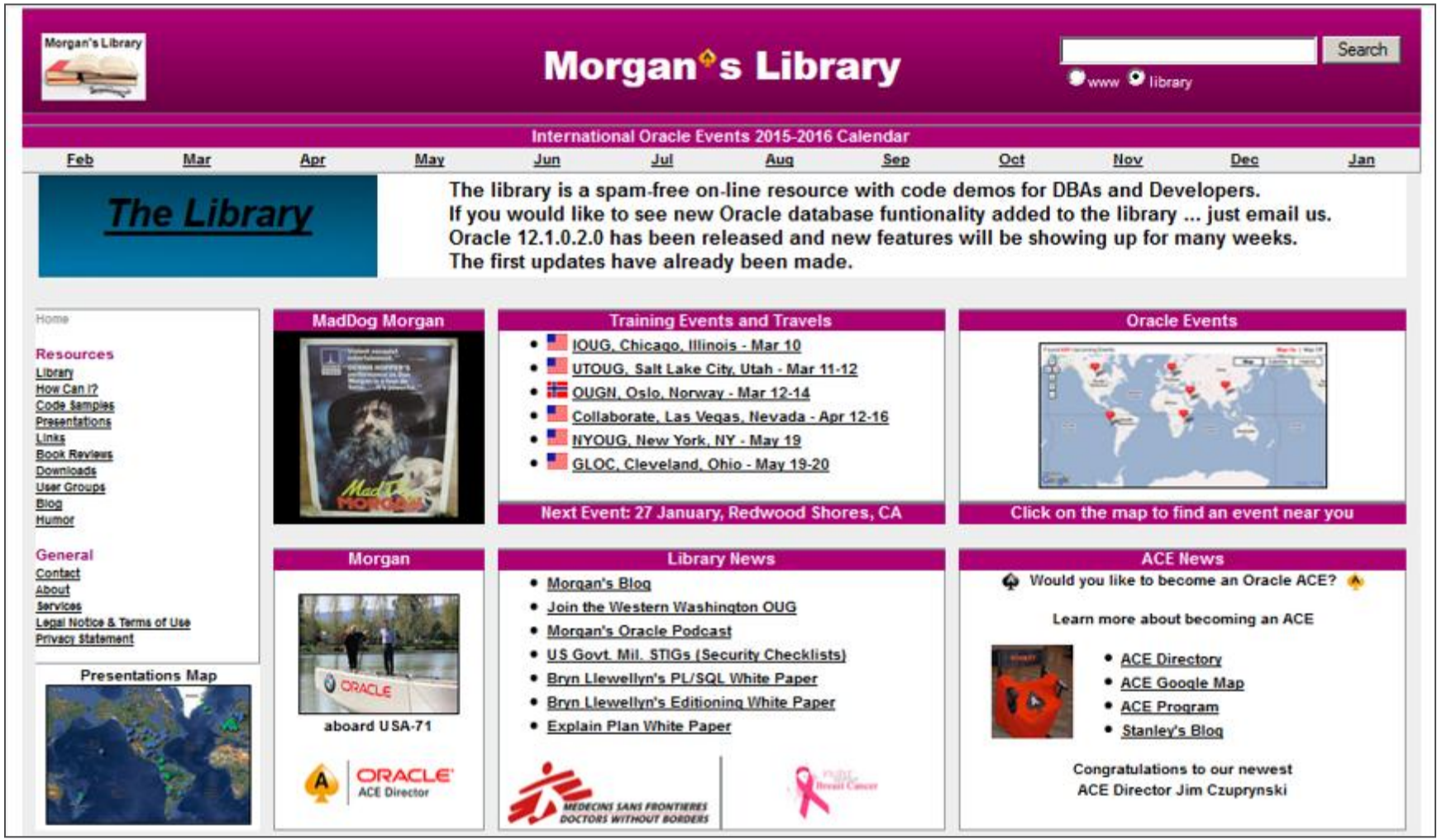
Daniel Morgan

- Oracle ACE Director
- 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, TimesTen, GoldenGate
  - The Morgan behind www.morganslibrary.org
  - Member Oracle Data Integration Solutions Partner Advisory Council
  - Co-Founder International GoldenGate Oracle Users Group
- Principal Adviser: Forsythe **Meta7**

System/370-145 system console

email: dmorgan@forsythe.com
Twitter: @damorgan12c

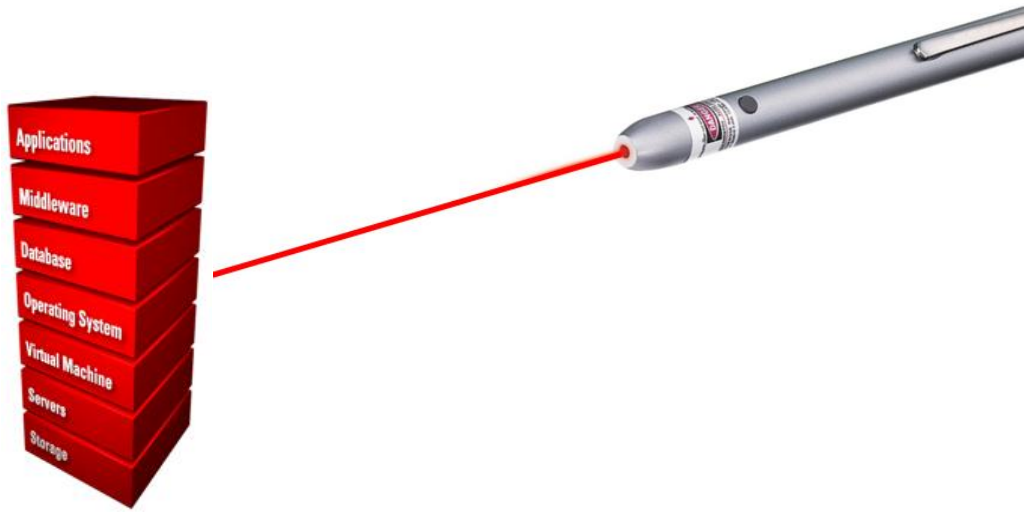
My Websites: Morgan's Library



www.morganslibrary.org

What Meta7 Brings To The Party

- The "Oracle Only" division of Forsythe focused on only the Red Stack
- A team of skilled professionals with
  - Extensive experience across multiple industries
  - Deep specialization in core Oracle technologies
    - Hardware
    - Licensing
    - Professional Services
  - 0% off-shoring: All work performed by US residents



- Reliable on-time and on-budget delivery
- Corporate headquarters in Chicago, Illinois
- New, State-of-the-Art Technology Evaluation Center
- Secure hosting and Managed Services in our own Tier 3 data center on the same power grid and fibre as O'Hare airport
- Flexible financial support

ORACLE Platinum Partner

What Meta7 Brings To The Party (2:2)

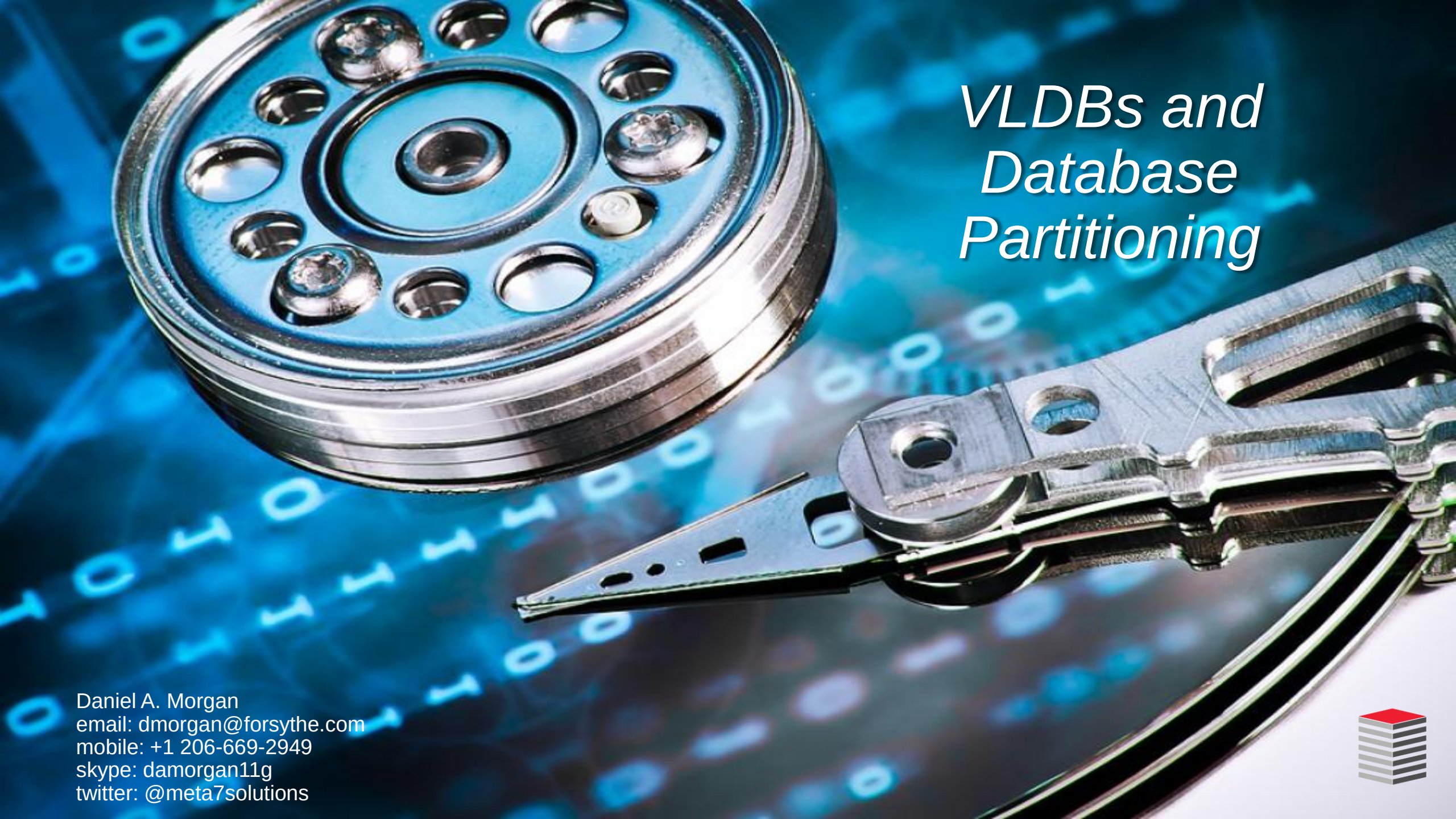
Product Area	Strategy
Cloud Solutions	Oracle Optimized Solution for Enterprise Cloud Infrastructure
Database	Oracle Database 11g
Database	Oracle Database 11g Data Warehousing
Database	Oracle Database 12c
Database	Oracle Enterprise Manager 12c
Database	Oracle Real Application Clusters 11g
Engineered Systems	Oracle Database Appliance Specialization
Engineered Systems	Oracle Exadata Database Machine
Industries	Professional Services
Middleware	GoldenGate 12.2
Servers and Storage Systems	Oracle Linux 5
Servers and Storage Systems	Oracle Solaris 10
Servers and Storage Systems	Oracle Solaris 11
Servers and Storage Systems	Oracle VM 3
Servers and Storage Systems	Oracle ZFS Storage
Servers and Storage Systems	SPARC Enterprise Entry-Midrange M-Series Servers
Servers and Storage Systems	SPARC T2 and T3-Based Servers
Servers and Storage Systems	SPARC T4-Based Servers
Servers and Storage Systems	SPARC T5-Based Servers
Servers and Storage Systems	Sun ZFS Storage Appliance

Zero Downtime Database Migrations with GoldenGate

Daniel A. Morgan
email: dmorgan@forsythe.com
mobile: +1 206-669-2949
skype: damorgan11g
twitter: @meta7solutions

Oracle Database Security

Daniel A. Morgan
email: dmorgan@forsythe.com
mobile: +1 206-669-2949
skype: damorgan11g
twitter: @meta7solutions



VLDBs and Database Partitioning

Daniel A. Morgan
email: dmorgan@forsythe.com
mobile: +1 206-669-2949
skype: damorgan11g
twitter: @meta7solutions

Database Performance

Daniel A. Morgan
email: dmorgan@forsythe.com
mobile: +1 206-669-2949
skype: damorgan11g
twitter: @meta7solutions

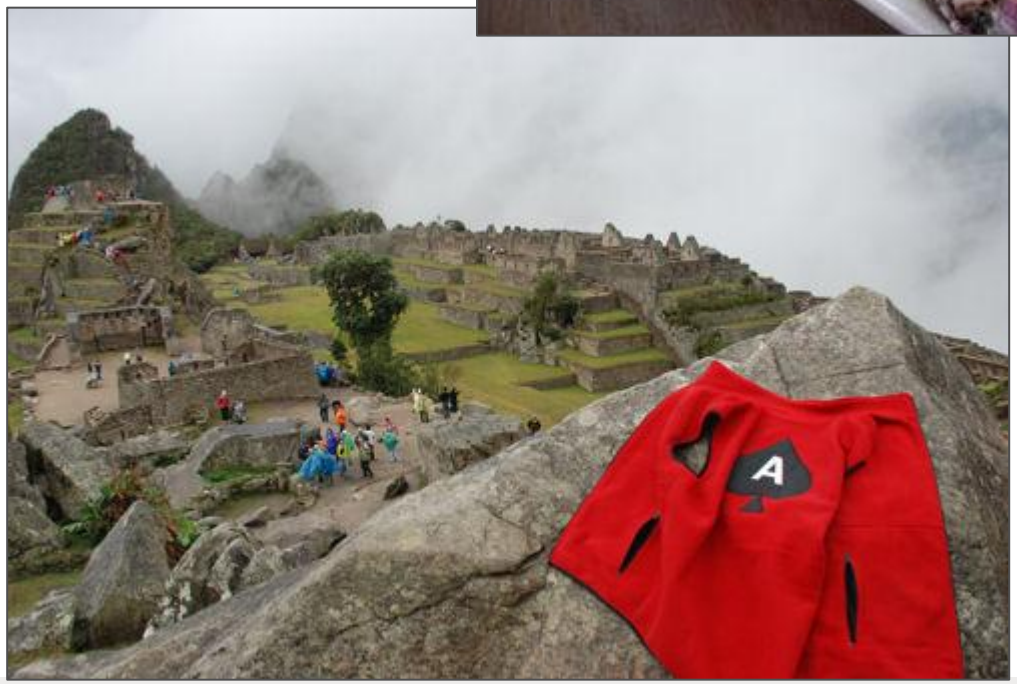
Oracle DBaaS Migration Road Map

Daniel A. Morgan
email: dmorgan@forsythe.com
mobile: +1 206-669-2949
skype: damorgan11g
twitter: @meta7solutions

IT Fire Fighting

Daniel A. Morgan
email: dmorgan@forsythe.com
mobile: +1 206-669-2949
skype: damorgan11g
twitter: @meta7solutions

Travel Log: 2010 - Lima Peru



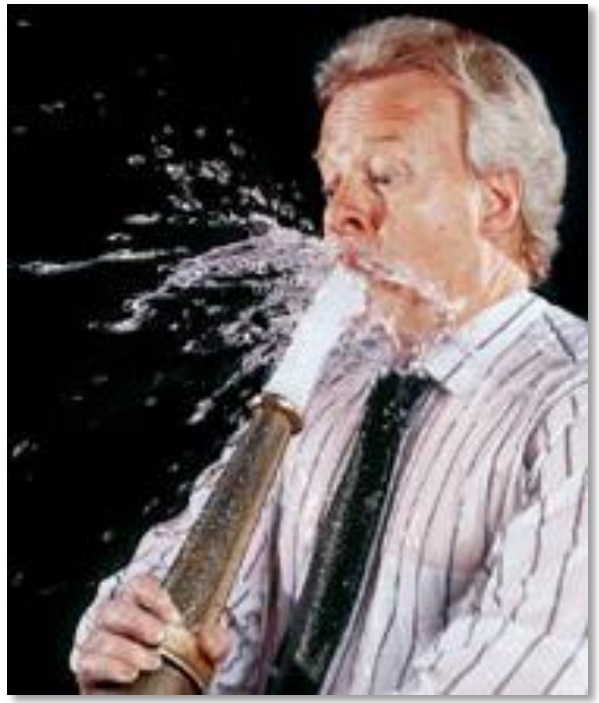
Travel Log: 2013 - Beijing China

Travel Log: 2014 - Galapagos Islands Ecuador

Travel Log: 2015 - Turkey

Travel Log: 2016 - California

Content Density Warning



Take Notes ... Ask Questions

Introduction

Non-Topics: Tom Kyte's 12 Features of Oracle Database 12c

- Can start SQL with PL/SQL
- Improved Defaults and Identity Columns
- Increased Size Limits
- Easy Top End Pagination Queries
- Row Pattern Matching
- Partitioning Improvements
- Adaptive Execution Plans
- Enhanced Statistics
- Temporary Undo
- Data Optimization (ILM)
- Application Continuity and Transaction Guard
- Pluggable Databases

Topics (1:2)

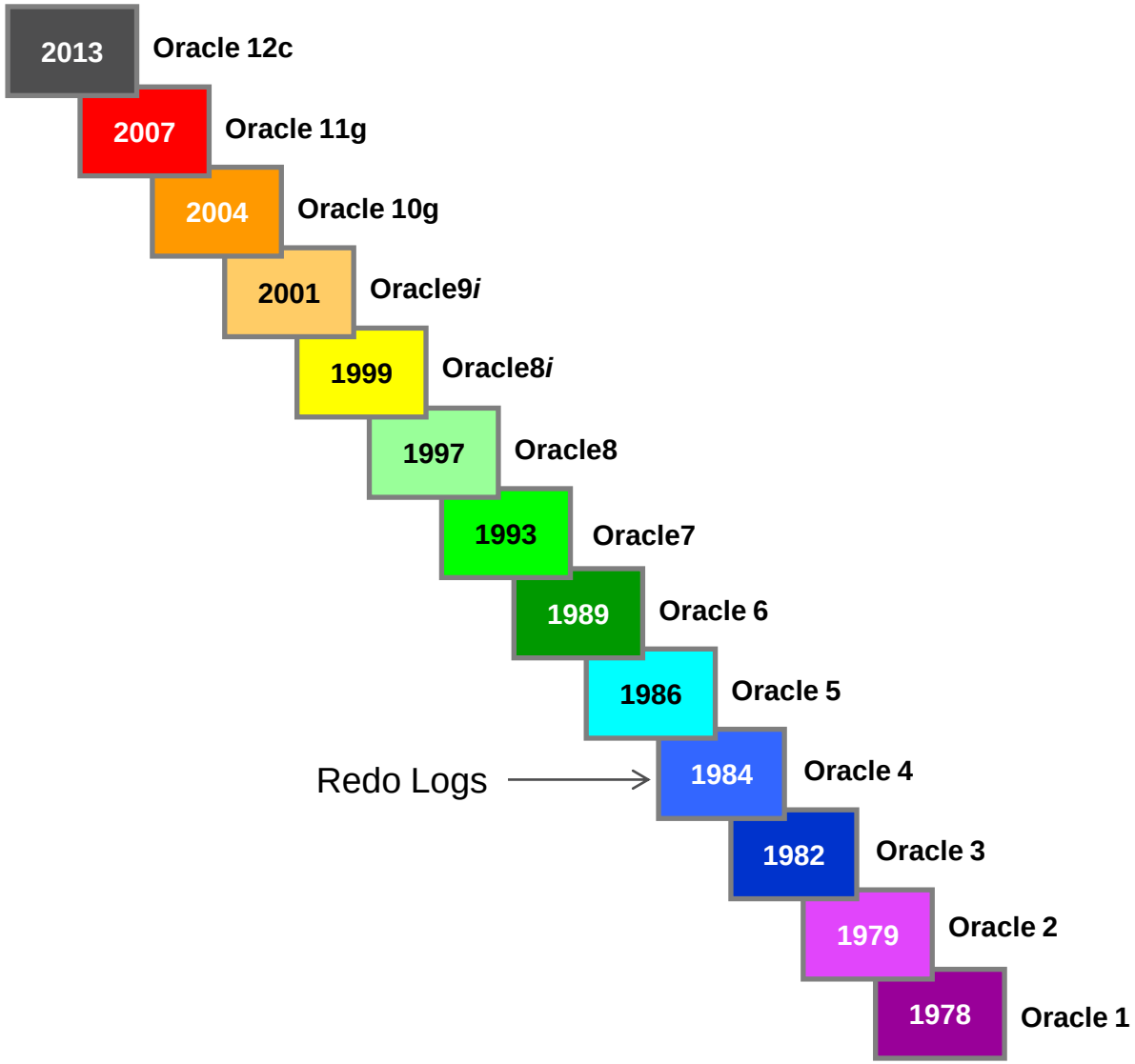
- Introduction
- New O/S Groups
- New Users with Escalated Privileges
- New Roles
- New System Privs
- New Feature Usage Reports
- Unified Audit Policies

Topics (2:2)

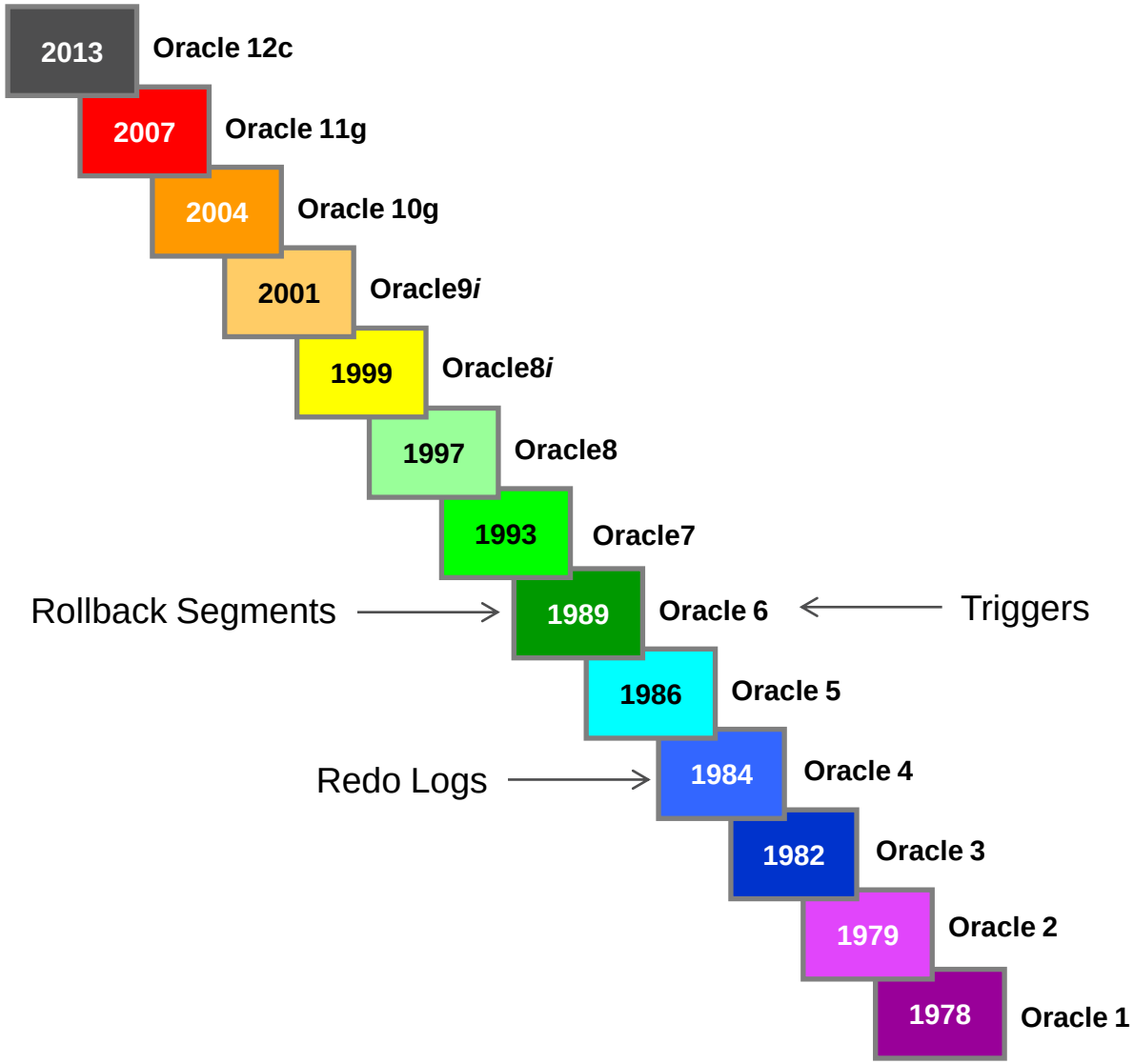
- SQL
  - Online Clause for DROP CONSTRAINT
  - Advanced Index Compression
  - Alter Index CLEANUP
  - Attribute Clustering
  - Table Invisible Columns
- PL/SQL
  - Predefined Inquiry Directives
  - ACCESSIBLE BY Clause
  - FETCH FIRST Clause
- Built-In Packages
  - SQL Translation Profiles

Almost everything you will see is included in your current license agreement ..... if you don't fully use what you purchased ..... you are just throwing money away

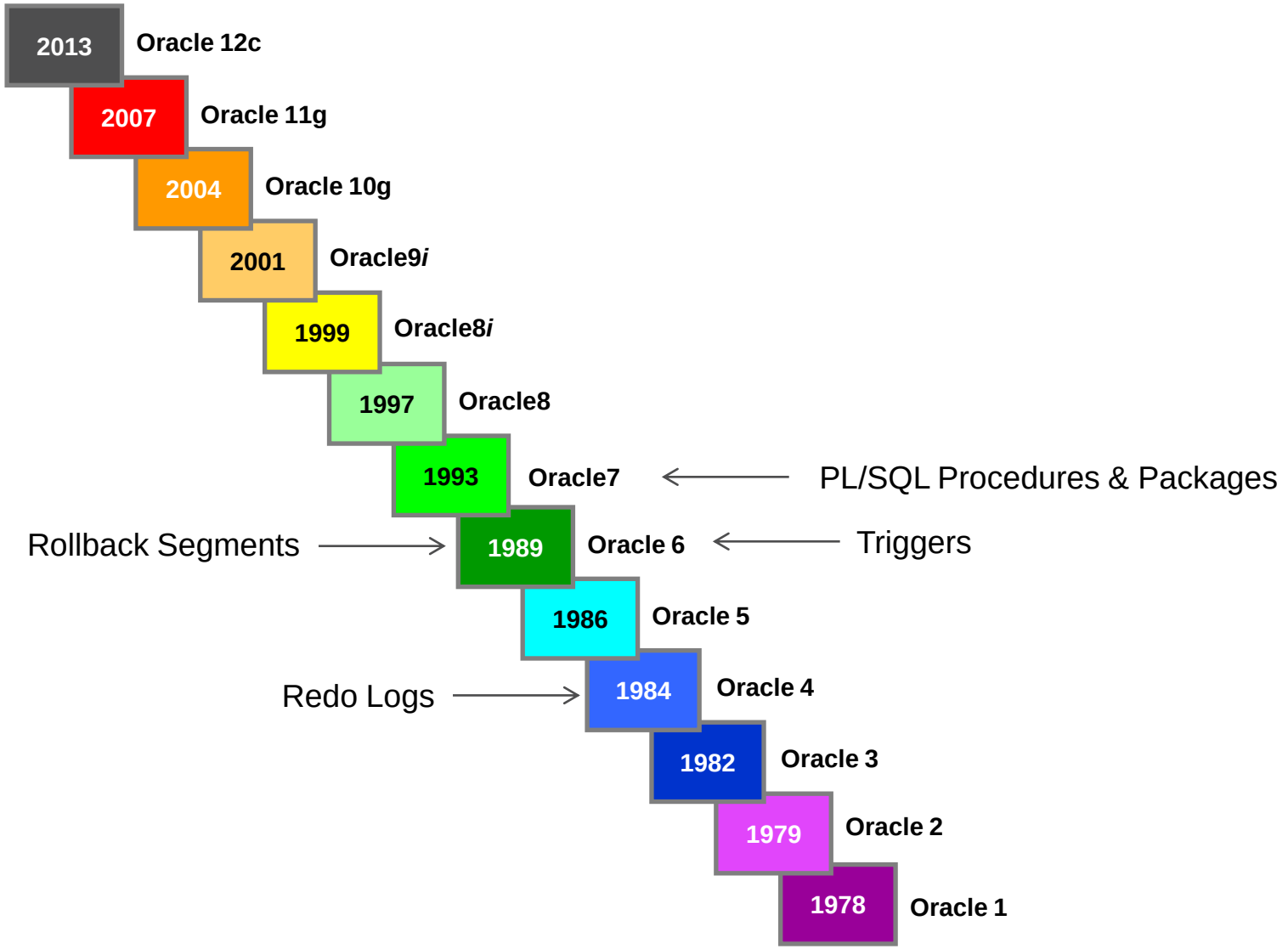
A Brief History of the Oracle Database



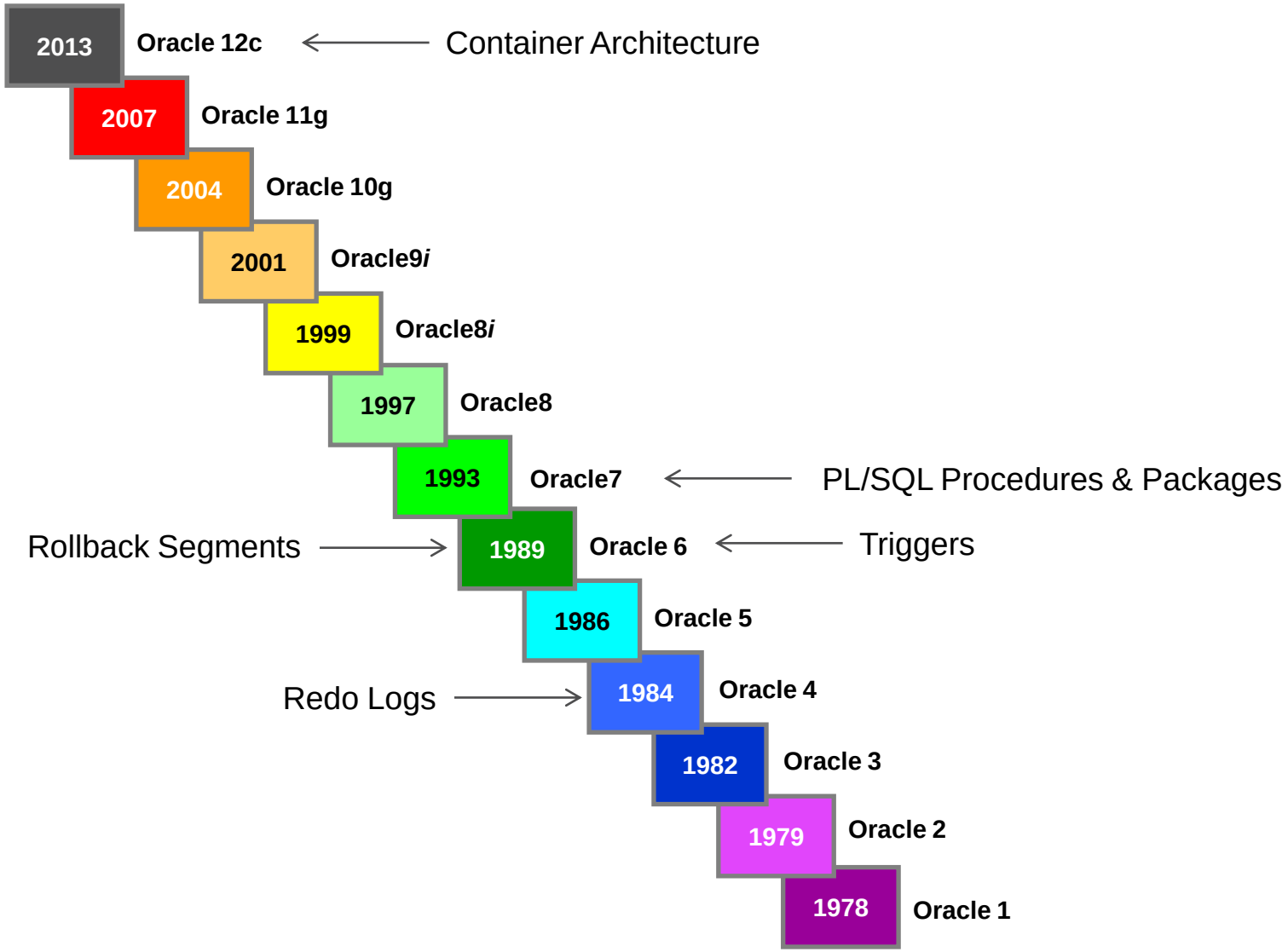
A Brief History of the Oracle Database



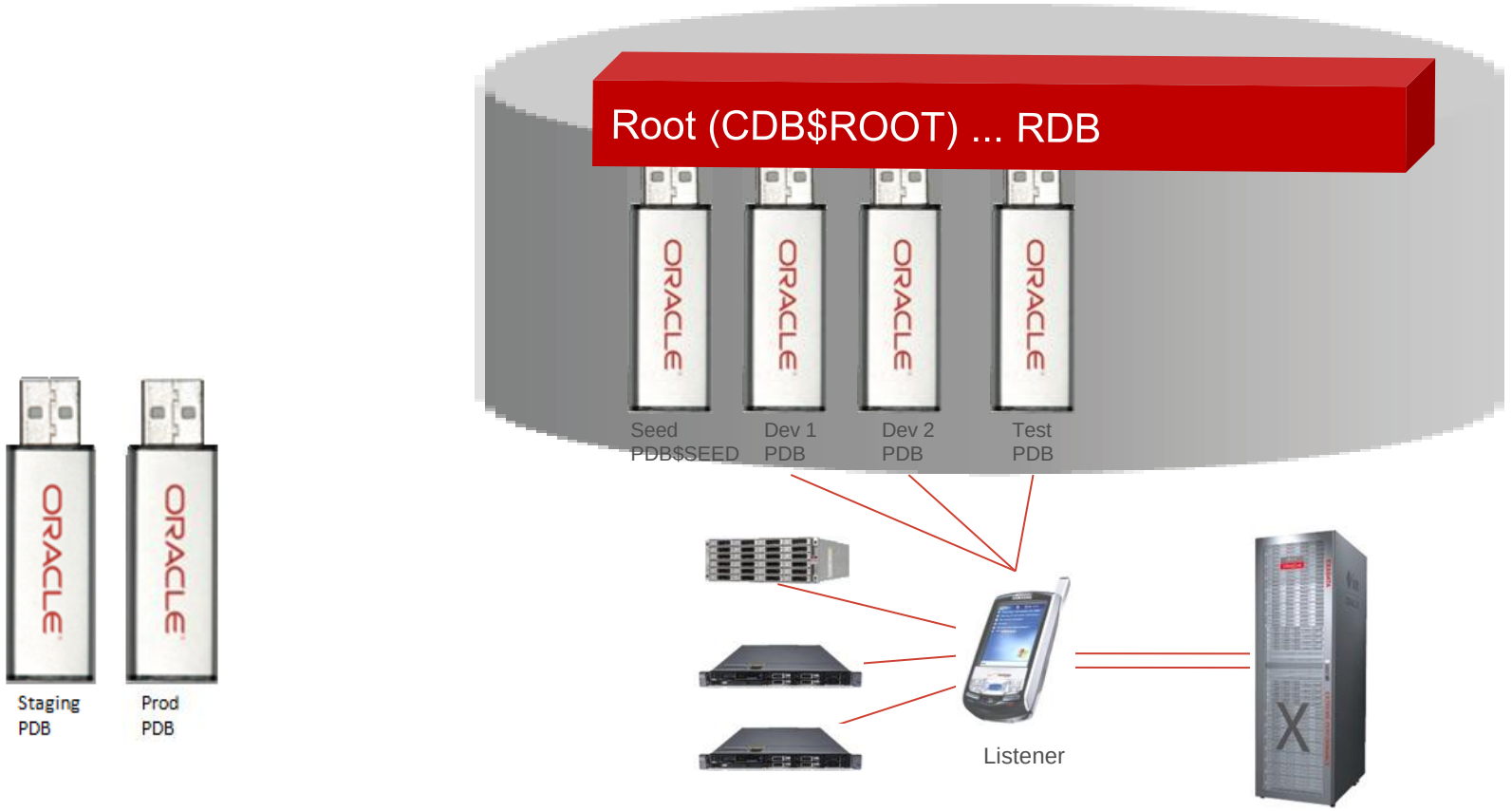
A Brief History of the Oracle Database



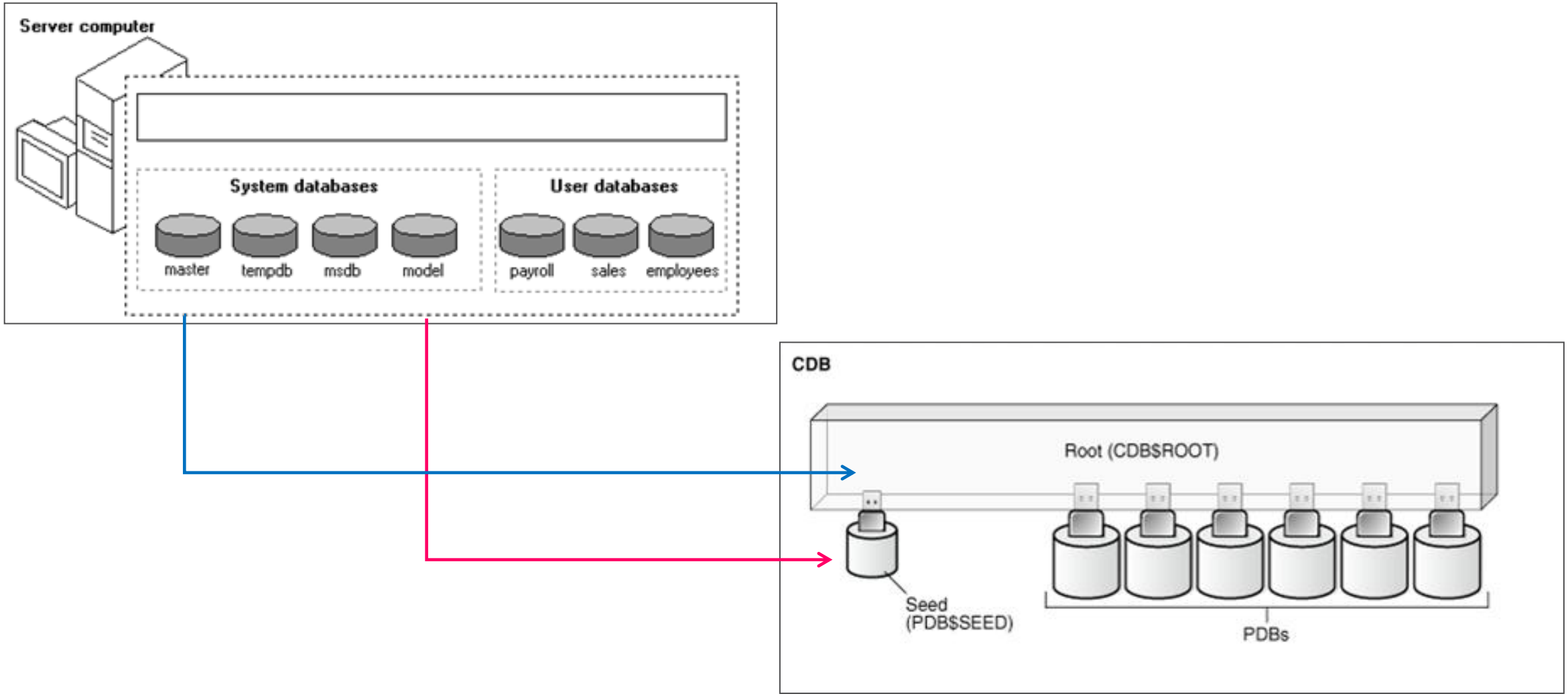
A Brief History of the Oracle Database



What Is Different: Container Architecture



New 12c Container Database Architecture



New Users with Escalated Privileges

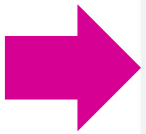
- Legacy
  - SYS
  - SYSTEM
- New
  - SYSBACKUP
  - SYSDG
  - SYSKM

New System Privileges

- AUDIT_ADMIN
- AUDIT_VIEWER
- CAPTURE_ADMIN
- CDB_DBA
- OPTIMIZER_PROCESSING_RATE
- PDB_DBA

```
SQL> desc dba_role_privs
 Name                  Null?    Type
 --------------------- -------- --------------
 GRANTEE                        VARCHAR2(128)
 GRANTED_ROLE                   VARCHAR2(128)
 ADMIN_OPTION                   VARCHAR2(3)
 DELEGATE_OPTION                VARCHAR2(3)
 DEFAULT_ROLE                   VARCHAR2(3)
 COMMON                         VARCHAR2(3)

SQL> SELECT *
  2  FROM dba_role_privs
  3* WHERE granted_role = 'OPTIMIZER_PROCESSING_RATE';

GRANTEE  GRANTED_ROLE                ADM DEL DEF COM
-------- --------------------------- --- --- --- ---
DBA      OPTIMIZER_PROCESSING_RATE   NO  NO  YES YES
SYS      OPTIMIZER_PROCESSING_RATE   YES NO  YES YES

SQL> SELECT privilege, table_name
  2  FROM role_tab_privs
  3  WHERE role = 'OPTIMIZER_PROCESSING_RATE';

PRIVILEGE       TABLE_NAME
--------------- ------------------------
UPDATE          OPT_CALIBRATION_STATS$
SELECT          OPT_CALIBRATION_STATS$
INSERT          OPT_CALIBRATION_STATS$
DELETE          OPT_CALIBRATION_STATS$
```

So far there is no information as to what this table will someday hold and how this new privilege might be used



New Roles

- 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

New Predefined Inquiry Directives

- Predefined Inquiry Directives act like built-in functions, but are not objects such as those in DBA_OBJECTS
- Predefined Inquiry Directives return values related to compiled PL/SQL objects and are of great value in creating robust debugging and error handling routines

```
SQL> CREATE OR REPLACE PROCEDURE pdid AUTHID DEFINER IS
  2  BEGIN
  3    dbms_output.put_line('I am ' || $$plsql_unit);
  4    dbms_output.put_line('I am owned by ' || $$plsql_unit_owner);
  5    dbms_output.put_line('I am a ' || $$plsql_unit_type);
  6    dbms_output.put_line('I am line ' || $$plsql_line || ' of the source code');
  7  END pdid;
  8  /

SQL> exec pdid
I am PDID
I am owned by SYS
I am a PROCEDURE
I am line 6 of the source code

PL/SQL procedure successfully completed.
```

Container Conversion Functions (1:2)

- Container Identifier Conversion Functions

```
SQL> SELECT con_id, dbid, con_uid, guid
   2 FROM v$pdbs;

    CON_ID       DBID    CON_UID GUID
---------- ---------- ---------- --------------------------------
2          4043696482 4043696482 EF72EF6B4DD0416E821AB0AE16B3A4E4
```

- CON_NAME_TO_ID
  - Returns the container ID based on the container's DBID

```
SQL> SELECT con_dbid_to_id(4043696482)
   2 FROM dual;

CON_DBID_TO_ID(4043696482)
--------------------------
                         2
```

- CON_GUID_TO_ID
  - In theory: Returns the container ID based on the container's GUID but so far in tests returns NULL

Container Conversion Functions (2:2)

- CON_NAME_TO_ID
  - Returns the container ID based on the container's name

```
SQL> SELECT con_name_to_id('PDB$SEED')
   2 FROM dual;

CON_NAME_TO_ID('PDB$SEED')
--------------------------
                         2
```

- CON_UID_TO_ID
  - Returns the container ID based on the container's UID

```
SQL> SELECT con_uid_to_id(4043696482)
   2 FROM dual;

CON_UID_TO_ID(4043696482)
-------------------------
                        2
```

ORA_INVOKING Functions

- ORA_INVOKING_USER
  - Returns the name of the database user who invoked the current statement or view

```
SQL> SELECT ora_invoking_user
  2  FROM dual;

ORA_INVOKING_USER
------------------------------
UWCLASS
```

- ORA_INVOKING_USERID
  - Returns the user id of the database user who invoked the current statement or view

```
SQL> SELECT ora_invoking_userid
  2  FROM dual;

ORA_INVOKING_USERID
------------------------------
110
```

SHOW Command (SQL*Plus)

- Show Container Identifier
- Show Container Name

```
C:\Users\oracle>sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Thu Sep 17 00:31:55 2015

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

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> sho user
USER is "SYS"
SQL> sho con_id

CON_ID
------------------------------
1
SQL> show con_name

CON_NAME
------------------------------
CDB$ROOT
SQL> ALTER SESSION SET CONTAINER=PDB$SEED;

Session altered.

SQL> sho con_id

CON_ID
------------------------------
2
SQL> show con_name

CON_NAME
------------------------------
PDB$SEED
```

STANDARD_HASH (1:2)

- Returns a hash value using one of several hash algorithms defined and standardized by the National Institute of Standards and Technology (NIST). This function is useful for performing authentication and maintaining data integrity in security applications such as digital signatures, checksums, and fingerprinting
- **`STANDARD_HASH(<value> ['<hashing_alogrithm>']) RETURN RAW;`**

STANDARD_HASH (2:2)

```
SQL> SELECT STANDARD_HASH('Morgan')
  2  FROM dual;

STANDARD_HASH('MORGAN')
----------------------------------------
8E4408B475D63385A73AED2FE911DD9818E82FB5

SQL> SELECT STANDARD_HASH('Morgan', 'SHA1')
  2  FROM dual;

STANDARD_HASH('MORGAN','SHA1')
----------------------------------------
8E4408B475D63385A73AED2FE911DD9818E82FB5

SQL> SELECT STANDARD_HASH('Morgan', 'SHA256')
  2  FROM dual;

STANDARD_HASH('MORGAN','SHA256')
----------------------------------------------------------------
02281B3B5DD57C4643681B8B113C9D56E9B8F1DC8C30A5BBA4C864BDD27D1ED7

SQL> SELECT STANDARD_HASH('Morgan', 'SHA384')
  2  FROM dual;

STANDARD_HASH('MORGAN','SHA384')
--------------------------------------------------------------------------------
D0739D820F3D82ED347EF68626FD6E08DC918CA98DEA41587C213ABEDACA7C25A46712D6E36D79857D775EC4A4CD9586

SQL> SELECT STANDARD_HASH('Morgan', 'SHA512')
  2  FROM dual;

STANDARD_HASH('MORGAN','SHA512')
--------------------------------------------------------------------------------
1E7C57248F1F665BCB46F6CB4FDF4765E1D6C533D4BAA360089FD30530CE82543ECCDB7A0526AEED0F637DBA147DC52DE41823179ECABCF5BBA8D0CE97EEB34F
```

SYS_CONTEXT (1:5)

- CDB_NAME ... container database name

```
SELECT sys_context('USERENV', 'CDB_NAME') FROM dual;

SYS_CONTEXT('USERENV','CDB_NAME')
----------------------------------------------------------------
pdbtest
```

- CLIENT_PROGRAM_NAME

```
SELECT sys_context('USERENV', 'CLIENT_PROGRAM_NAME') FROM dual;

SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME')
----------------------------------------------
sqlplus.exe
```

SYS_CONTEXT (2:5)

- CON_ID ... container identifier

```
SELECT sys_context('USERENV', 'CON_ID') FROM dual;

SYS_CONTEXT('USERENV','CON_ID')
------------------------------
1
```

- CON_NAME ... container name

```
SELECT sys_context('USERENV', 'CON_NAME') FROM dual;

SYS_CONTEXT('USERENV','CON_NAME')
--------------------------------
CDB$ROOT

-- connect to pluggable database
conn uwclass/uwclass@orabase

SYS_CONTEXT('USERENV','CON_NAME')
--------------------------------
ORABASE
```

SYS_CONTEXT (3:5)

- DB_SUPPLEMENTAL_LOG_LEVEL ... level of supplemental logging

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

SELECT sys_context('USERENV', 'DB_SUPPLEMENTAL_LOG_LEVEL')
FROM dual;

SYS_CONTEXT('USERENV','DB_SUPPLEMENTAL_LOG_LEVEL')
--------------------------------------------------
MINIMAL
```

DBLINK_INFO ... returns the source of a database link session

```
SELECT sys_context('USERENV', 'DBLINK_INFO') FROM dual;
```

- IS_APPLY_SERVER ... Returns TRUE if queried from within a SQL Apply server in a logical standby database

```
SELECT sys_context('USERENV', 'IS_APPLY_SERVER') FROM dual;

SYS_CONTEXT('USERENV','IS_APPLY_SERVER')
----------------------------------------
FALSE
```

SYS_CONTEXT (4:5)

- IS_DG_ROLLING_UPGRADE ... Returns TRUE if a rolling upgrade of the database software in a Data Guard configuration, initiated by way of the DBMS_ROLLING package, is active

```
SELECT sys_context('USERENV', 'IS_DG_ROLLING_UPGRADE') FROM dual;

SYS_CONTEXT('USERENV','IS_DG_ROLLING_UPGRADE')
----------------------------------------------
FALSE
```

- ORACLE_HOME ... returns the environment value of $ORACLE_HOME

```
SELECT sys_context('USERENV', 'ORACLE_HOME') FROM dual;

SYS_CONTEXT('USERENV','ORACLE_HOME')
------------------------------------
c:\app\oracle\product\12.1.0\dbhome_1
```

SYS_CONTEXT (5:5)

- PLATFORM_SLASH ... returns the forward or back-slash for the operating system environment

```
SELECT sys_context('USERENV', 'PLATFORM_SLASH') FROM dual;

SYS_CONTEXT('USERENV','PLATFORM_SLASH')
----------------------------------------
/
```

- SCHEDULER_JOB ... Returns Y if the current session belongs to a foreground job or background job

```
SELECT sys_context('USERENV', 'SCHEDULER_JOB') FROM dual;

SYS_CONTEXT('USERENV','SCHEDULER_JOB')
---------------------------------------
N
```

Write Messages To The Alert Log

- Undocumented but appears to be an attempt to move the four named PL/SQL objects out of DBMS_SYSTEM and into the new package DBMS_LOG
- Contains 4 objects
  - KSDDDT
    - Prints the date stamp to the target file (alert log and/or trace file)
  - KSDFLS
    - Flushes any pending output to the target alert log or trace file
  - KSDIND
    - Indents before the next write (ksdwrt) by printing that many colons (:) before the next write
  - KSDWRT
    - Prints a message to the target file (alert log and/or trace file)

```
exec dbms_system.ksdwrt(3, '-- Start Message --');
exec dbms_system.ksdwrt(3, 'Test Message');
exec dbms_system.ksdwrt(dbms_system.alert_file, '-- End Message --');
```

Credentials

- New 12cR1 built-in package: DBMS_CREDENTIAL
- Interface for authenticating and impersonating EXTPROC callout functions, as well as external jobs, remote jobs and file watchers from the SCHEDULER
- Contains 5 objects
  - CREATE_CREDENTIAL
  - DISABLE_CREDENTIAL
  - DROP_CREDENTIAL
  - ENABLE_CREDENTIAL
  - UPDATE_CREDENTIAL

```
DECLARE
 cname   user_credentials.credential_name%TYPE := 'UWCRED';
 uname   user_credentials.username%TYPE := 'UWCLASS';
 pwd     sys.scheduler$_credential.password%TYPE := 'ZzYzX6*';
 dbrole  VARCHAR2(30) := NULL;
 windom  sys.scheduler$_credential.domain%TYPE := NULL;
 comment user_credentials.comments%TYPE := 'Test Cred';
 enable  BOOLEAN := FALSE;
BEGIN
  dbms_credential.create_credential(cname, uname, pwd, dbrole, windom, comment, enable);
END;
/

SELECT *
 FROM scheduler$_credential;
```

Data Redaction (1:2)

- New 12cR1 built-in package: DBMS_REDACT
- Contains 8 objects
  - ADD_POLICY
    - Define a redaction policy on a table or view ALTER POLICY
  - ALTER_POLICY
    - Alter a data redaction policy
  - DISABLE_POLICY
    - Disable a data redaction policy
  - DROP_POLICY
    - Drop a data redaction policy
  - ENABLE_POLICY
    - Enable a disabled data redaction policy
  - FPM_MASK
    - Apply format-preserving Data Redaction to the input
  - FPM_UNMASK
  - Remove a format-preserving Data Redaction from the input

Data Redaction (2:2)

- UPDATE_FULL_REDACTION_VALUES
  - Update replacements for full redaction

```
-- syntax
dbms_redact.add_policy(
object_schema           IN VARCHAR2 := NULL,
object_name             IN VARCHAR2,
policy_name             IN VARCHAR2,
policy_description      IN VARCHAR2 := NULL,
column_name             IN VARCHAR2 := NULL,
column_description      IN VARCHAR2 := NULL,
function_type           IN BINARY_INTEGER := dbms_redact.full,
function_parameters     IN VARCHAR2 := NULL,
expression              IN VARCHAR2,
enable                  IN BOOLEAN := TRUE,
regexp_pattern          IN VARCHAR2 := NULL,
regexp_replace_string   IN VARCHAR2 := NULL,
regexp_position         IN BINARY_INTEGER := 1,
regexp_occurrence       IN BINARY_INTEGER := 0,
regexp_match_parameter IN VARCHAR2 := NULL);
```

Partitioning Support

- New 12cR1 built-in package: DBMS_PART
- Contains 3 objects
  - CLEANUP_GIDX
    - Gathers the list of global indexes where optimized asynchronous index maintenance has taken place to clean up entries pointing to data segments that no longer exist

```
-- syntax
dbms_part.cleanup_gidx(schema_name_in IN VARCHAR2 DEFAULT NULL,
                       table_name_in  IN VARCHAR2 DEFAULT NULL);

-- example
exec dbms_part.cleanup_gidx('SH');

exec dbms_part.cleanup_gidx(NULL, 'SALES');

exec dbms_part.cleanup_gidx('SH', 'SALES');
```

  - CLEANUP_ONLINE_OP
    - Clean up failed online move operations

```
dbms_part.cleanup_online_op(
schema_name     IN VARCHAR2 DEFAULT NULL,
table_name      IN VARCHAR2 DEFAULT NULL,
partition_name IN VARCHAR2 DEFAULT NULL);
```

Password Verify Function (1:3)

- $ORACLE_HOME/rdbms/admin/utlpwdmg.sql

CREATE OR REPLACE FUNCTION ora12c_verify_function

```
Rem
Rem $Header: rdbms/admin/utlpwdmg.sql /main/9 2013/11/07 08:58:18 jkati Exp $
Rem
Rem utlpwdmg.sql
Rem
Rem Copyright (c) 2006, 2013, Oracle and/or its affiliates.
Rem All rights reserved.
Rem
Rem    NAME
Rem      utlpwdmg.sql - script for Default Password Resource Limits
Rem
Rem    DESCRIPTION
Rem      This is a script for enabling the password management features
Rem      by setting the default password resource limits.
Rem
Rem    NOTES
Rem      This file contains a function for minimum checking of password
Rem      complexity. This is more of a sample function that the customer
Rem      can use to develop the function for actual complexity checks that the
Rem      customer wants to make on the new password.
Rem
Rem    MODIFIED    (MM/DD/YY)
Rem    jkati        10/16/13 - bug#17543726 : remove complexity_check,
Rem                            string_distance, ora12c_strong_verify_function
Rem                            since we now provide them by default with new db
Rem                            creation
```

Password Verify Function (2:3)

```
-- 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;
```

Password Verify Function (3:3)

- Note that the following part of the script is commented out ... not what I would want if I was responsible for database security

```
/**
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;
*/
```

Patching (1:2)

- New 12cR1 built-in package: DBMS_QOPATCH
- Contains 25 objects
- Examples
  - GET_OPATCH_FILES
    - Returns the list of files modified in the given patch number in XML format
  - GET_OPATCH_FILES
    - Returns a list of files modified in the given patch number in XML format
  - GET_OPATCH_INSTALL_INFO
    - Returns the XML element containing the ORACLE_HOME details such as patch and inventory location
  - GET_OPATCH_LIST
    - Returns a list of installed patches
  - GET_OPATCH_LSINVENTORY
    - Returns the complete opatch inventory
  - GET_OPATCH_PREREQUISITES
    - Returns prerequisite patches for a given patch as XML element

Patching (2:2)

```
SELECT dbms_qopatch.get_opatch_bugs
FROM dual;

<bugInfo>
  <bugs xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <bug id="17352756">
      <UId>FlexibleDataType-7b5f507b-b2cf-4321-bb00-e39aab61cea4</UId>
      <description>QPATCH DIRECTORIES - OPATCH_LOG_DIR &amp;amp; OPATCH_SCRIPT_DIR INCORRECTLY DEFINED.</description>
    </bug>
  </bugs>
</bugInfo>



SELECT dbms_qopatch.get_opatch_files('17352756')
FROM dual;

DBMS_QOPATCH.GET_OPATCH_FILES('17352756')
----------------------------------------------------------
<patchFiles>
   <patchID xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">17352756</patchID>
   <files xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <file>execqopi.sql</file>
     <file>17352756_apply.sql</file>
     <file>17352756_rollback.sql</file>
   </files>
</patchFiles>
```

Pluggable Database Support (1:2)

- New 12cR1 built-in package: DBMS_PDB
- Contains 14 objects
- Examples
  - CHECK_PLUG_COMPATIBILITY
    - Determines whether a pluggable database described by file pdb_descr_file is compatibile with the current CDB
  - DESCRIBE
    - Generates XML describing tablespaces and datafiles belonging to a pluggable database
  - SYNC_PDB
    - After plug, syncs the PDB with the CDB, so that it will be ready for use
  - UPDATE_CDBVW_STATS
    - Updates CDB View Stats
  - UPDATE_VERSION
    - Update PDB's VSN in container$ after upgrade

Pluggable Database Support (2:2)

```
conn / as sysdba

ALTER PLUGGABLE DATABASE pdborcl CLOSE;

ALTER PLUGGABLE DATABASE pdborcl OPEN READ ONLY;

exec dbms_pdb.describe('/home/oracle/pdborcl.xml', 'PDBORCL');

BEGIN
  IF dbms_pdb.check_plug_compatibility('/home/oracle/pdborcl.xml', 'PDBDEV') THEN
   dbms_output.put_line('TRUE');
  ELSE
    dbms_output.put_line('FALSE');
  END IF;
END;
/

SELECT *
FROM pdb_plug_in_violations;
```

Privilege Capture (1:2)

- New 12cR1 built-in package: DBMS_PRIVILEGE_CAPTURE
- Contains 5 objects
  - CREATE_CAPTURE
    - Creates a privilege analysis policy to show privilege use by database users
  - DISABLE_CAPTURE
    - Disables a defined capture so it no longer captures privileges
  - DROP_CAPTURE
    - Drops a defined capture
  - ENABLE_CAPTURE
    - Enables a defined capture to capture privileges
  - GENERATE_RESULT
    - Loads captured privilege information into the data dictionary

Privilege Capture (2:2)

```
conn sys@pdbdev as sysdba

BEGIN
  dbms_privilege_capture.create_capture('UWPrivCapt','Privilege Capture Demo', dbms_privilege_capture.g_database);

  dbms_privilege_capture.enable_capture('UWPrivCapt');

  dbms_privilege_capture.disable_capture('UWPrivCapt');

  dbms_privilege_capture.generate_result('UWPrivCapt');
END;
/

desc dba_priv_captures
desc dba_used_privs

col username format a15
col object_owner format a15
col object_name format a20
col obj_priv format a12

SELECT username, object_owner, object_name, obj_priv
FROM dba_used_objprivs
WHERE  username = 'OE';

exec dbms_privilege_capture.drop_capture('UWPrivCapt');
-- the rows in dba_used_objprivs are deleted when the Capture is dropped.
```

Rolling Upgrades (1:2)

- New 12cR1 built-in package: DBMS_ROLLING
- Contains 8 objects
  - BUILD_PLAN
    - Either builds a complete upgrade plan or modifies the remaining unprocessed portion of an existing plan
    - The build procedure interprets the configured rolling upgrade parameters to produce a customized upgrade plan
  - DESTROY_PLAN
    - Purges all rolling upgrade state from the database
    - Called on completion of a rolling upgrade
  - FINISH_PLAN
    - Executes the FINISH phase instructions in the upgrade plan
  - INIT_PLAN
    - The first procedure that must be called to prepare for a DBMS_ROLLING administered rolling upgrade
    - Communicates with the complete set of databases in the DG_CONFIG, and creates a default set of rolling upgrade parameters for building a rolling upgrade plans

Rolling Upgrades (2:2)

- ROLLBACK_PLAN
  - Rolls back the group of administered PDBs to their initial state
  - The package creates an initial set of guaranteed restore points for all participating PDBs
  - Flashbackes back all PDBs in the leading change group to their respective restore points if the switchover has not been performed
- SET_PARAMETER
  - 2 overloads
  - Called to set and unset rolling upgrade parameters
  - Changes to the plan parameters do not take effect until the user re-invokes the BUILD procedure to reconstruct the upgrade plan
- START_PLAN
  - Executes the START phase instructions in the upgrade plan
  - It is the first procedure that is called to initiate the rolling upgrade
  - Upon completion of this phase, the future primary will be ready to be upgraded
- SWITCHOVER
  - Executes the SWITCHOVER phase instructions in the upgrade plan
  - Called once the START procedure has completed execution of all START phase instructions

Row Level Security aka Virtual Private Database

- The DBMS_RLS package has been in the Oracle Database since version 8.1.5 but Oracle keeps adding to its capabilities
- New in 12cR1
  - ADD_GROUPED_POLICY
    - Add a row level security policy to a policy group for a table or view
  - ALTER_GROUPED_POLICY
    - Alter a row level security policy of a policy group
  - ALTER_POLICY
    - Alter a row level security policy

SQL Plan Management Directives (1:2)

- New 12cR1 built-in package: DBMS_SPD
- Contains 9 objects
  - ALTER_SQL_PLAN_DIRECTIVE
    - Change attributes of a SQL Plan Directive
  - CREATE_STG_TAB_DIRECTIVE
    - Creates a staging table to pack SQL Plan directives for export
  - DROP_SQL_PLAN_DIRECTIVE
    - Drop an existing SQL Plan Directive
  - FLUSH_SQL_PLAN_DIRECTIVE
    - Manually flushes a SQL Plan directives that has been automatically recorded in SGA memory while executing SQL statements
  - GET_PREFS
    - Returns the retention value preferences for SQL Plan Directives
  - PACK_STGTAB_DIRECTIVE
    - Exports SQL Plan Directives into a staging table

SQL Plan Management Directives (2:2)

- SET_PREFS
  - Setting different preferences for SQL Plan Directives
- UNPACK_STGTAB_DIRECTIVE
  - Unpacks (imports) SQL Plan Directives from a staging table

```
DECARE
  packing_list dbms_spd.objecttab := dbms_spd.ObjectTab();
  dir_cnt number;
 BEGIN
   packing_list.extend(1);
   packing_list(1).owner := 'SH'; -- schema name
   packing_list(1).object_name := null; -- all tables in SH
   packing_list(1).object_type := 'TABLE'; -- type of object

   dir_cnt := dbms_spd.unpack_stgtab_directive('mydirtab', obj_list => packing_list);
   dbms_output.put_line('Unpacked ' || TO_CHAR(dir_no) || ' directives');
 END;
 /
```

SQL Plan Management (1:3)

- New 12cR1 built-in package: DBMS_SPM
- Contains 23 new 12cR1 objects
  - ACCEPT_SQL_PLAN_BASELINE
    - Accept a plan based on the recommendation of an evolve task
  - CANCEL_EVOLVE_TASK
    - Cancels a currently executing evolve task
  - CREATE_EVOLVE_TASK (2 overloads)
    - Creates an advisor task and sets its parameters
  - DROP_EVOLVE_TASK
    - Drops an evolved task
  - DROP_SQL_PLAN_BASELINE
    - Drops a single plan, or all plans associated with a SQL statement
  - EXECUTE_EVOLVE_TASK
    - Executes a previously created evolve task
  - IMPLEMENT_EVOLVE_TASK
    - Implements a plan based on the recommendation of an evolve task

SQL Plan Management (2:3)

- INTERRUPT_EVOLVE_TASK
  - Interrupts a currently executing evolve task
- REPORT_AUTO_EVOLVE_TASK
  - Displays the results of an execution of an automatic evolve task
- REPORT_EVOLVE_TASK
  - Displays the results of an evolved task
- RESET_EVOLVE_TASK
  - Restarts an evolve task
- RESUME_EVOLVE_TASK
  - Resume an evolve task
- SET_EVOLVE_TASK_PARAMETER (2 overloads)
  - Sets a parameter of an evolve task

SQL Translation Profiles

- New 12cR1 built-in package: DBMS_SQL_TRANSLATOR
- Contains 18 objects
- Example objects
  - REGISTER_ERROR_TRANSLATION
    - Registers a custom translation of an Oracle error code and SQLSTATE in a SQL translation profile
  - REGISTER_SQL_TRANSLATION
    - Registers a custom translation of a SQL statement in a SQL translation profile

```
BEGIN
  dbms_sql_translator.register_error_translation(profile_name => 'UW_SQLTRANS', error_code => 1, translated_code => 2601);
END;
/

BEGIN
  dbms_sql_translator.register_sql_translation(
     profile_name => 'UW_SQLTRANS',
     sql_text => 'select top 5 * from emp',
     translated_text => 'select * from emp where rownum <= 5');
END;
/
```

Transparent Sensitive Data Protection (1:3)

- New 12cR1 built-in package: DBMS_TSDP_MANAGE
- Provides an interface to import and manage sensitive columns and sensitive column types in the database, and is used in conjunction with the DBMS_TSDP_PROTECT package with regard to transparent sensitive data protection (TSDP) policies
- Contains 9 objects
  - ADD_SENSITIVE_COLUMN
  - ADD_SENSITIVE_TYPE
  - ALTER_SENSITIVE_COLUMN
  - DROP_SENSITIVE_COLUMN
  - DROP_SENSITIVE_TYPE
  - DROP_SENSITIVE_TYPE_SOURCE
  - IMPORT_DISCOVERY_RESULT
  - IMPORT_SENSITIVE_TYPES
  - REMOVE_DISCOVERY_RESULT

Transparent Sensitive Data Protection (2:3)

- New 12cR1 built-in package: DBMS_TSDP_PROTECT
- Provides an interface to configure transparent sensitive data protection (TSDP) policies in conjunction with the DBMS_TSDP_MANAGE package
- Contains 10 objects
  - ADD_POLICY
  - ALTER_POLICY
  - ASSOCIATE_POLICY
  - DISABLE_PROTECTION_COLUMN
  - DISABLE_PROTECTION_SOURCE
  - DISABLE_PROTECTION_TYPE
  - DROP_POLICY
  - ENABLE_PROTECTION_COLUMN
  - ENABLE_PROTECTION_SOURCE
  - ENABLE_PROTECTION_TYPE

Transparent Sensitive Data Protection (3:3)

```
DECLARE
  redact_feature_opts dbms_tsdp_protect.feature_options;
  pol_conditions      dbms_tsdp_protect.policy_conditions;
 BEGIN
   redact_feature_opts('expression') := 'SYS_CONTEXT(''USERENV'', ''SESSION_USER'') = ''SYS''';
   redact_feature_opts ('function_type') := 'DBMS_REDACT.PARTIAL';
   redact_feature_opts ('function_parameters') := 'STR, VVVVVVVVV,VVVVVVVVV, *, 1, 6';
   pol_conditions (DBMS_TSDP_PROTECT.DATATYPE) := 'VARCHAR2';

   dbms_tsdp_protect.alter_policy('PARTIAL_MASK', redact_feature_opts, pol_conditions);
 END;
 /
```

Unified Audit Policies

- Like traditional auditing but Audit Policies are new to Database 12c and make possible substantial improvements in the way auditing is defined which is of great value when deploying a container database
- DDL Variants
  - CREATE
  - ALTER
  - DROP

```
CREATE AUDIT POLICY uw_priv_clause PRIVILEGES ALTER ANY TABLE;

CREATE AUDIT POLICY uw_actions_clause ACTIONS LOGOFF, ALL ON sys.user$;

CREATE AUDIT POLICY uw_actions_component ACTIONS COMPONENT = datapump EXPORT;

CREATE AUDIT POLICY uw_role_clause ROLES DBA;

CREATE AUDIT POLICY uw_multi_clause PRIVILEGES ALTER ANY TABLE
ACTIONS LOGOFF ROLES DBA;

CREATE AUDIT POLICY uw_full_clause PRIVILEGES ALTER ANY TABLE
ACTIONS LOGOFF ROLES DBA
WHEN 'SYS_CONTEXT(''USERENV'', ''ISDBA'') = ''TRUE'''
EVALUATE PER STATEMENT
CONTAINER = ALL;
```

DBMS_METADATA

- In the Oracle Database since version 9.0.1 but Oracle keeps adding to its capabilities
- New in 12cR1
  - GET_INDPART_TS
    - Returns the tablespace number for the tablespace of a partitioned index component from KU$_INDEX_VIEW

```
SQL> SELECT object_id
    2 FROM dba_objects
    3 WHERE object_name = (
    4 SELECT index_name
    5 FROM dba_ind_partitions
    6 WHERE rownum = 1);

OBJECT_ID
----------
      8610
     18124
     93318
     93632
     93631
     93456

SQL> SELECT dbms_metadata.get_indpart_ts(8610)
   2 FROM dual;

DBMS_METADATA.GET_INDPART_TS(8610)
----------------------------------
                                 1
```

DBMS_REDEFINITION

- DBMS_REDEFINITION has been in the Oracle Database since version 9.0.1 but Oracle keeps adding to its capabilities
- New in 12cR1
  - FINISH_REDEF_TABLE
    - Has new CONTINUE_AFTER_ERRORS and DML_LOCK_TIMEOUT parameters
  - REDEF_TABLE
    - Provides a single push-button interface that integrates several redefinition steps
  - START_REDEF_TABLE
    - Has new COPY_VPD_OPT and CONTINUE_AFTER_ERRORS parameters
  - SYNC_INTERIM_TABLE
    - Has new CONTINUE_AFTER_ERRORS parameter

ORIGINAL_SQL_TXT

- This is undocumented ... but what great functionality for auditing

```
SQL> conn sys@pdbdev as sysdba

SQL> CREATE OR REPLACE TRIGGER test
  2  AFTER GRANT
  3  ON DATABASE
  4  DECLARE
  5   stmnt_list dbms_standard.ora_name_list_t;
  6   n          PLS_INTEGER;
  7  BEGIN
  8    IF (ora_sysevent = 'GRANT') THEN
  9      n := dbms_standard.original_sql_txt(stmnt_list);
 10      dbms_output.put_line(TO_CHAR(n));
 11      dbms_output.put_line(stmnt_list(n));
 12    END IF;
 13  END test;
 14  /

SQL> GRANT all ON tab$ TO scott;
 1
GRANT all ON tab$ TO scott

Grant succeeded.
```

DBMS_STATS (1:2)

- The DBMS_STATS package has been in the Oracle Database since version 8.1.5 but Oracle keeps adding to its capabilities
- New in 12cR1
  - Some functions replaced by Constants
  - CLOB_TO_VARRAY
    - Converts a CLOB to multiple elements of a VARRAY
  - DELETE_DATABASE_STATS
  - DELETE_PENDING_SYSTEM_STATS
  - DELETE_PROCESSING_RATE
  - EXPORT_PENDING_SYSTEM_STATS
  - EXPORT_STATS_FOR_DP
  - EXPORT_SYSTEM_STATS
  - GATHER_PROCESSING_RATE
  - GET_STAT_TAB_VERSION
  - IMPORT_STATS_FOR_DP

DBMS_STATS (2:2)

- PUBLISH_PENDING_SYSTEM_STATS
- REMAP_STAT_TABLE
- REPORT_GATHER_AUTO_STATS
- REPORT_GATHER_DATABASE_STATS
- REPORT_GATHER_DICTIONARY_STATS
- plus 6 more ...

```
SQL> DECLARE
  2    c CLOB;
  3    v ds_varray_4_clob;
  4  BEGIN
  5    c := 'A' || RPAD('A', 3999) || RPAD('B', 4000) || RPAD('C', 2000);
  6    dbms_output.put_line(TO_CHAR(LENGTH(c)));
  7    v := dbms_stats.clob_to_varray(c);
  8    dbms_output.put_line(v.COUNT);
  9* END;
 10  /
10000
3

PL/SQL procedure successfully completed.
```

Conclusion

- If the only thing you are going to get out of an Oracle Database is columns and rows you are wasting your money
- There are, of course, a lot of valuable capabilities in the Oracle Database you can leverage for an extra licensing fee ... and I like a lot of them ...
  - Active Data Guard
  - Advanced Compression
  - Advanced Security
  - Partitioning
- But there is a staggering amount of value in the many database extensions that are included in the license agreement you already purchased

- I encourage you to learn how they work ... and to make them work for you

```
*
ERROR at line 1:
ORA-00028: your session has been killed
```

Thank You

Daniel A. Morgan
email: dmorgan@forsythe.com
mobile: +1 206-669-2949
skype: damorgan11g
twitter: @meta7solutions