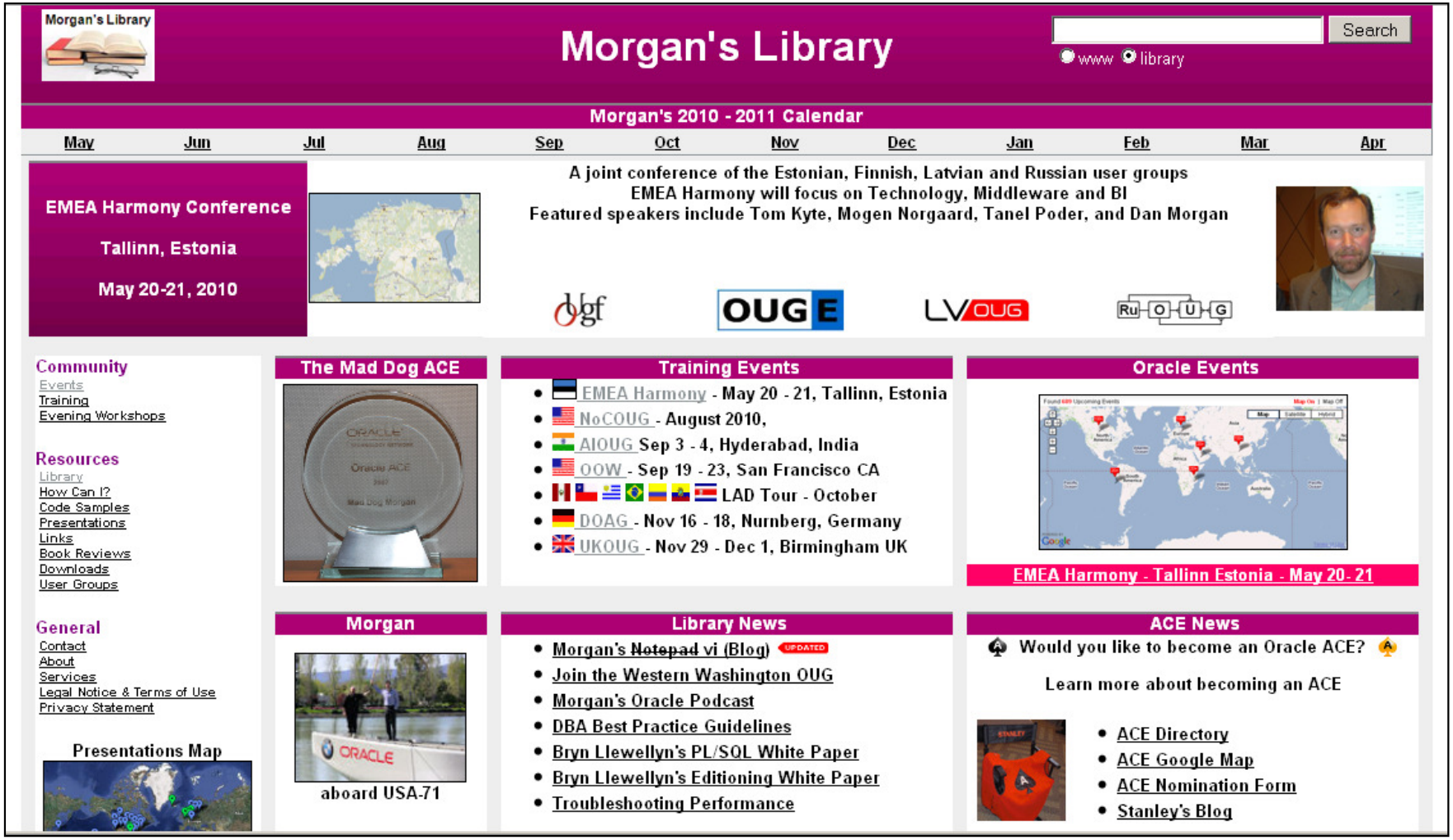
Oracle Database 12c
One Year of Hands On Later

Daniel A. Morgan: Oracle ACE Director

Daniel Morgan

- More than 45 years technology experience
  - First computer was an IBM 360/40 mainframe in 1970
  - Fortran IV and Punch Cards
- Oracle ACE Director
- Curriculum author and primary Oracle instructor at University of Washington
- Guest lecturer on Oracle at Harvard University
- Decades of hands-on SQL, PL/SQL, and DBA experience
- The "Morgan" behind Morgan's Library on the web
  www.morganslibrary.org
- 10g, 11g, and 12c Beta tester
- Co-founder International GoldenGate Oracle Users Group

The Morgan's Library Web Site

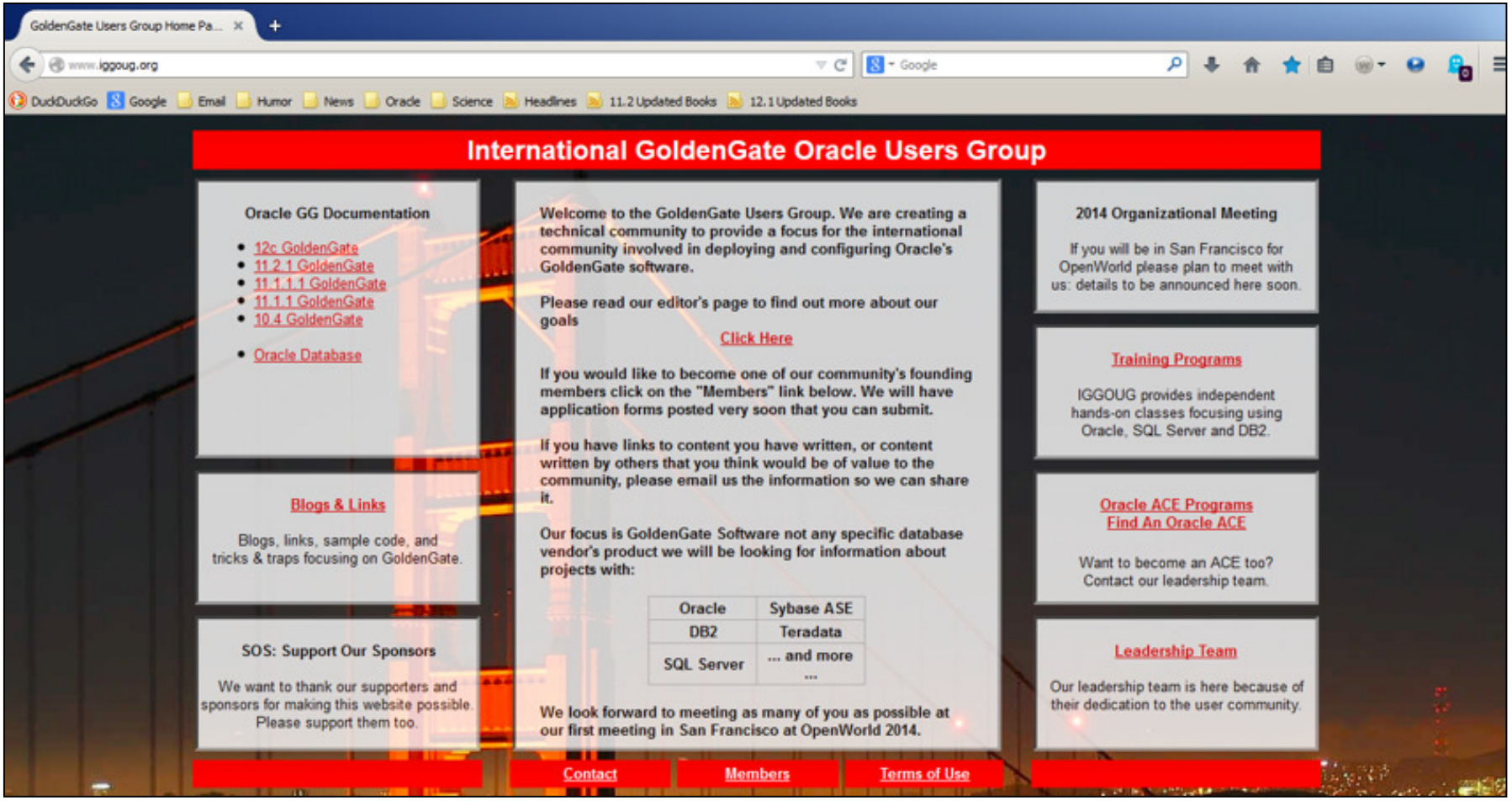


cd $MORGAN_BASE/Seattle

cd $MORGAN_BASE/San_Francisco

Under the GoldenGate Bridge

IGGOUG: The International GoldenGate OUG



www.iggoug.org

San Francisco

To Watch Larry's AC72

Objectives

- Learn the logical and physical changes in Database 12c and how they impact upgrade procedures and processes
- Learn about changes in how the 12c database must be managed
- Learn about changes to the 12c data dictionary
- Learn about resource management between PDBs
- If time permits ... explore some additional value in Database 12c
- Conclusions

Please Participation

Take Notes: Ask Questions

In the beginning

- With Chris Date at Oracle User Group Finland 2011

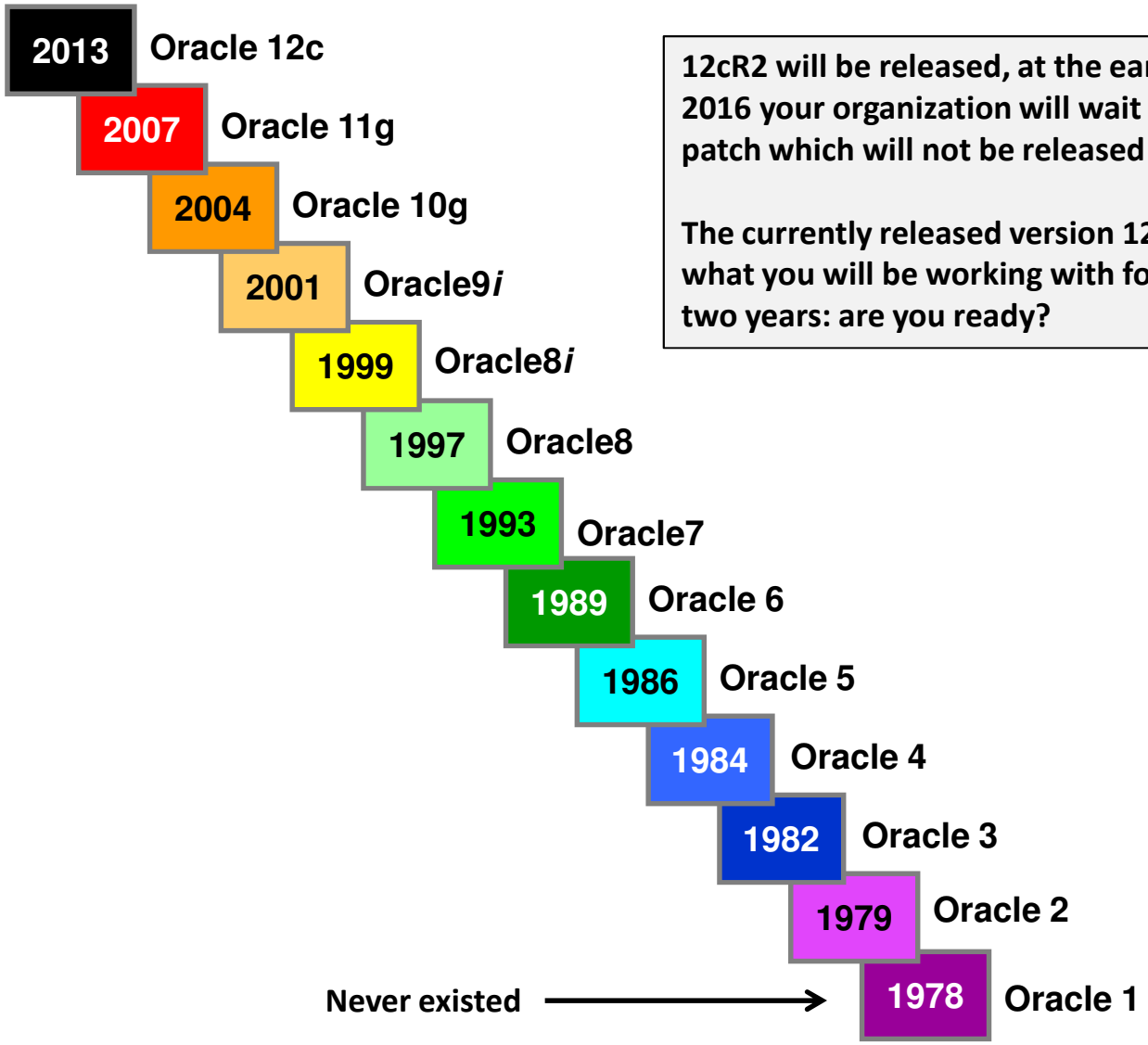
A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network

Oracle Server History

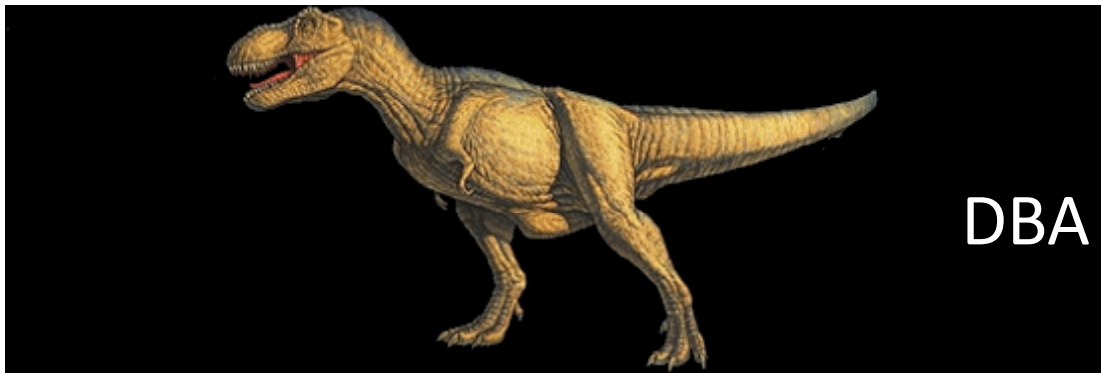


12cR2 will be released, at the earliest, Fall 2016 your organization will wait for the first patch which will not be released before 2017

The currently released version 12.1.0.2 is what you will be working with for at least two years: are you ready?

Mythology and Dinosaurs

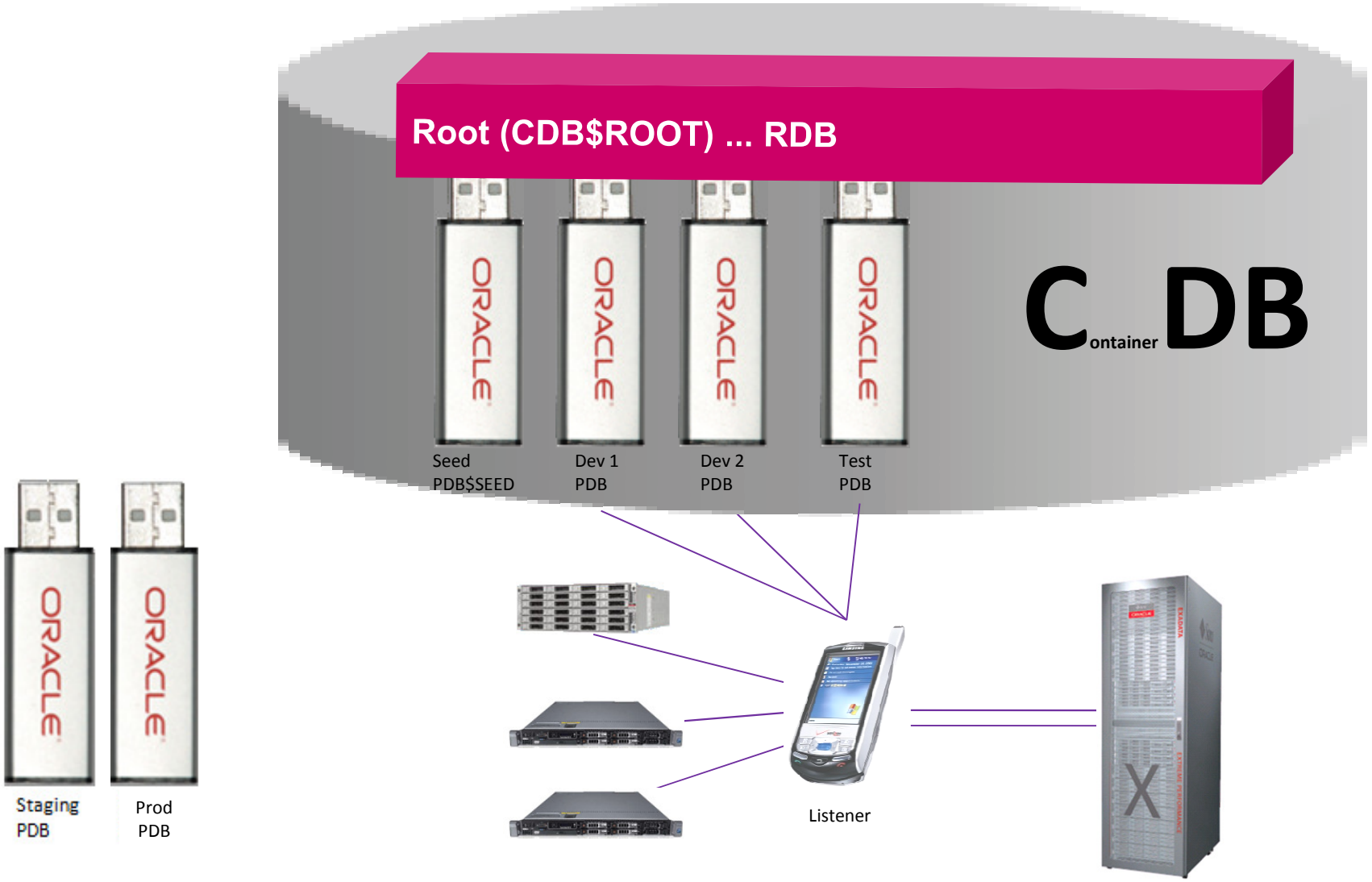
- Most 11gR2 was managed as though it was 7.3.4
- You can't do that with 12cR1 and beyond and be successful



DBA Self-Image

DBA Reality

Oracle Marketing View of the 12c Database



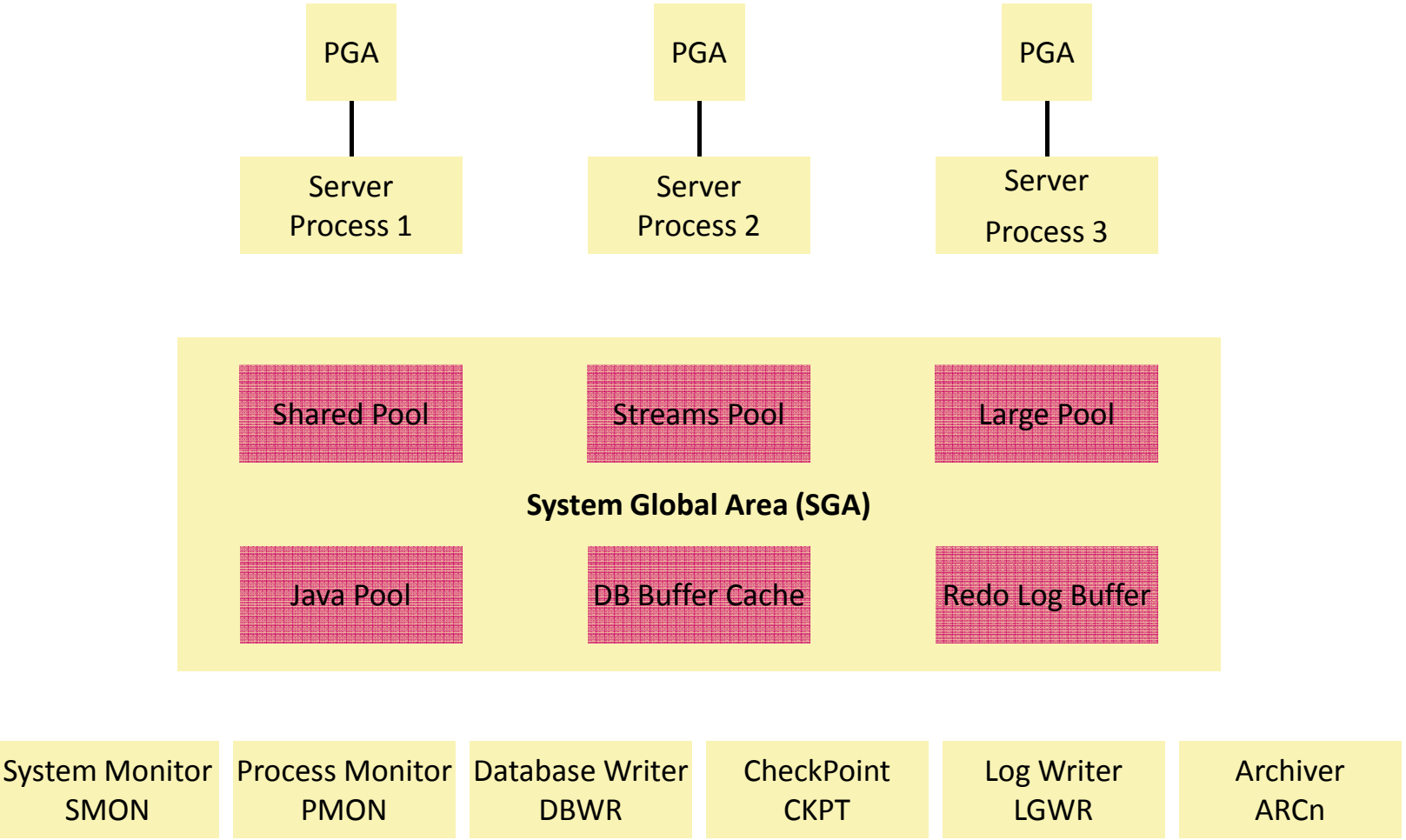
Everything Oracle is a Map: Think Latitude and Longitude

- What is physical
  - spfile
  - control files
  - redo logs
  - data files
  - temp files
- What is logical
  - Everything else
    - Tablespaces are maps to the location of physical data files
    - Tables are maps to where in a datafile a specific row is stored
    - Indexes are maps providing a shorter search path to where a row is stored
    - Columns are maps to where in a row a specific value is stored

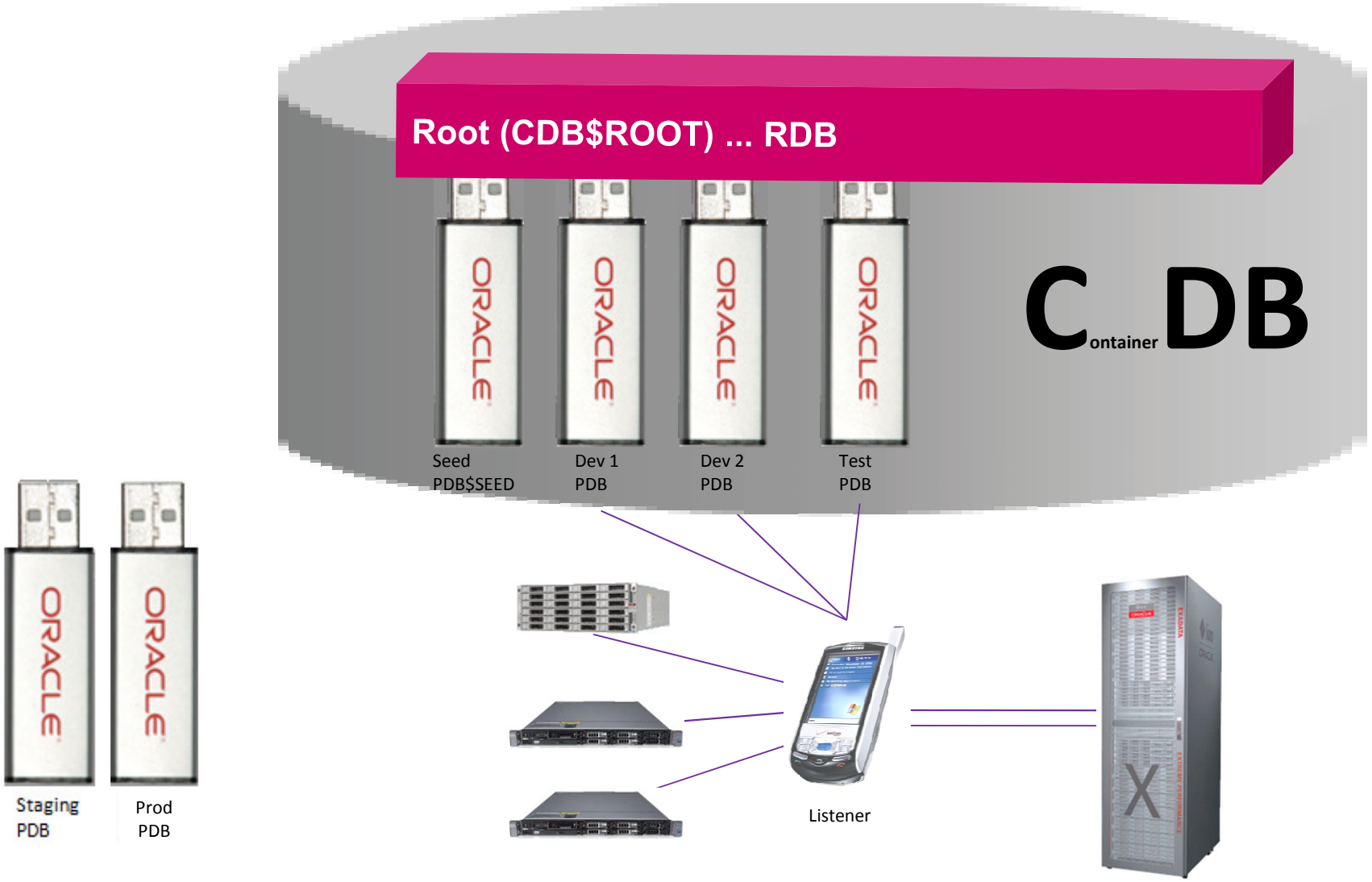
Instance vs Database

- Instance
  - Processes (pmon, smon, lgwr, dbwr, arch)
  - Memory Structures (SGA, PGA, UGA, Java Pool, Buffer Cache)
- Database
  - SPFILE
  - Control File
  - Data Files
  - Undo Files
  - Temp Files
  - Redo Logs

Instance Components



Oracle Marketing View of the 12c Database



What Is Mandatory in 12c? (1:2)

- Container 0
  - Background processes
- Container 1: CDB$ROOT
  - Holds the base data dictionary objects
  - Holds the base AWR objects
- Container 2: PDB$SEED
  - A read-only clone-ready pluggable database
  - Inherits data dictionary and AWR objects by pointer
  - Rough functional equivalent to model database in SQL Server
- Containers 3 ... 254
  - Customer created pluggable databases
  - Inherits CDB data dictionary, AWR, and code objects by pointer
  - Makes its metadata available to CDB$ROOT by pointer

What Is Mandatory in 12c? (2:2)

- Container 0
  - One set of redo logs for all containers
- Container 1: CDB$ROOT
  - Control Files
  - SYSTEM, SYSAUX, TEMP tablespaces
  - One UNDO tablespace for all containers
- Container 2: PDB$SEED
  - SYSTEM, SYSAUX, TEMP tablespaces
- Containers 3 ... 254
  - SYSTEM, SYSAUX, TEMP tablespaces
  - Customer application data tablespaces

Data Dictionary Changes (1:3)

- X$: In-Memory Linked Lists
- GV$: Global (multi-instance) views based on X$ linked lists
- V$: Single instance views based on X$ linked lists
- CDB_
  - Everything in every container plugged into and including CDB$ROOT
- DBA_
  - Everything in all schemas in a single container
- ALL_
  - Everything created and owned or with permissions granted in a single container
- USER_
  - Everything created and owned by a single schema in a single container

Data Dictionary Changes (2:3)

- The following dynamic performance views behave differently from all others
  - [g]v$sysstat
  - [g]v$sys_time_model
  - [g]v$system_event
  - [g]v$system_wait_class
- When queried from the root, these views return instance-wide data, with 0 in the CON_ID column for each row returned. However, you can query equivalent views that behave the same as other container data objects. The following views can return specific data for each container in a CDB
  - [g]v$con_sysstat
  - [g]v$con_sys_time_model
  - [g]v$con_system_event
  - [g]v$con_system_wait_class

Data Dictionary Changes (3:3)

- Many data dictionary column sizes are larger

```
SQL> desc dba_objects;
 Name                                      Null?    Type
 ----------------------------------------- -------- -------------
 OWNER                                               VARCHAR2(128)
 OBJECT_NAME                                         VARCHAR2(128)
 SUBOBJECT_NAME                                      VARCHAR2(128)
 OBJECT_ID                                           NUMBER
 DATA_OBJECT_ID                                      NUMBER
 OBJECT_TYPE                                         VARCHAR2(23)
 CREATED                                             DATE
 LAST_DDL_TIME                                       DATE
 TIMESTAMP                                           VARCHAR2(19)
 STATUS                                              VARCHAR2(7)
 TEMPORARY                                           VARCHAR2(1)
 GENERATED                                           VARCHAR2(1)
 SECONDARY                                           VARCHAR2(1)
 NAMESPACE                                           NUMBER
 EDITION_NAME                                        VARCHAR2(128)
 SHARING                                             VARCHAR2(13)
 EDITIONABLE                                         VARCHAR2(1)
 ORACLE_MAINTAINED                                   VARCHAR2(1)
```

You can make them display properly by formatting in glogin.sql

Only One Set of Control Files and Redo Logs

```
SQL> SELECT value
  2  FROM v$parameter
  3  WHERE name LIKE 'control_files%';

VALUE
--------------------------------------------------------------------------------
/app/oracle/oradata/orabase/control01.ctl,/app/oracle/fast_recovery_area/orabase/control02.ctl




SQL> SELECT group#, member, con_id
  2  FROM v$logfile
  3* ORDER BY 1,2;

    GROUP# MEMBER                                             CON_ID
---------- -------------------------------------------------- ----------
         1 /app/oracle/fast_recovery_area/orabase/redo01B.log          0
         1 /app/oracle/oradata/orabase/redo01a.log                     0
         2 /app/oracle/fast_recovery_area/redo02B.log                  0
         2 /app/oracle/oradata/orabase/redo02a.log                     0
         3 /app/oracle/fast_recovery_area/orabase/redo03b.log          0
         3 /app/oracle/oradata/orabase/redo03a.log                     0
```

One Undo Tablespace for All Containers

```
SQL> SELECT tablespace_name, con_id
  2  FROM cdb_tablespaces
  3  WHERE contents LIKE '%UNDO%';

TABLESPACE_NAME                     CON_ID
------------------------------ ----------
UNDOTBS1                                1
```

Many Data and Temp Files Each Container Specific

```
SQL> SELECT tablespace_name, COUNT(*)
  2  FROM cdb_tablespaces
  3  GROUP BY tablespace_name
  4  ORDER BY 1;

TABLESPACE_NAME                 COUNT(*)
------------------------------ ----------
CATTBSP                                 1
CATTEMP                                 1
EXAMPLE                                 1
ORADATA                                 1
SYSAUX                                  4
SYSTEM                                  4
TEMP                                    4
UNDOTBS1                                1
USERS                                   2
UWDATA                                  1

10 rows selected.
```

Should You Deploy a Container DB?

- To create a container DB: Yes or Yes?
  - Improved separation of duties
  - Faster and more flexible deployment
  - You don't have a choice
- To multi-tenant consolidate: Yes or No?
  - Patching requirements
  - Upgrade requirements
  - Backup and restore requirements
  - Security requirements
  - Infrastructure
  - DR Replication
- To enable in-memory database: Yes or No?

Consolidation Considerations (1:2)

- What are the implications of consolidation?
  - Processes
  - Transactions
  - Connections
  - Memory
  - CPU
  - I/O
    - *Do you really want to use only one TEMP tablespace?*
    - *How do you calculate space allocation for UNDO?*
    - *What is size and distribution of the AWR data in the SYSAUX tablespace?*
  - Networks Latency and Bandwidth
  - Backups and Restores
  - DataGuard Replication

Consolidation Considerations (2:2)

- Can we use our current servers for consolidation ...
  or more importantly "should" we use our current servers
  - Pizza Boxes (1U, 2U, 3U)
  - Blades (H/P and Cisco UCS)
  - M10 (Fujitsu)
  - P Series (IBM)
  - Z Series (IBM)
  - T Series (Sun-Oracle)
  - M Series (Sun-Oracle)
  - ODA
  - Sparc SuperCluster (T4-4, T5-8, M6-32)
  - Exadata

Resource Management (1:4)

- Three different resources must be considered
  - cpu
  - i/o
  - memory
- cpu and i/o can be controlled between PDBs within a single container database with DBMS_RESOURCE_MANAGER
  - DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN
  - DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE
- There is no way to control memory allocation between PDBs

Resource Management (2:4)

- Plans can be created by PDB

```
dbms_resource_manager.create_cdb_plan(
plan    IN VARCHAR2,
comment IN VARCHAR2 DEFAULT '');



dbms_resource_manager.create_cdb_plan_directive(
plan                  IN VARCHAR2,
pluggable_database    IN VARCHAR2,
comment               IN VARCHAR2 DEFAULT '',
shares                IN NUMBER   DEFAULT NULL,
utilization_limit     IN NUMBER   DEFAULT NULL,
parallel_server_limit IN NUMBER   DEFAULT NULL);
```

Resource Management (3:4)

- Initialization parameters can be modified at many levels

```
SQL> desc v$parameter
 Name                          Null?    Type
 ----------------------------- -------- ----------------
 NUM                                    NUMBER
 NAME                                   VARCHAR2(80)
 TYPE                                   NUMBER
 VALUE                                  VARCHAR2(4000)
 DISPLAY_VALUE                          VARCHAR2(4000)
 ISDEFAULT                              VARCHAR2(9)
 ISSES_MODIFIABLE                       VARCHAR2(5)
 ISSYS_MODIFIABLE                       VARCHAR2(9)
 ISPDB_MODIFIABLE                       VARCHAR2(5)
 ISINSTANCE_MODIFIABLE                  VARCHAR2(5)
 ISMODIFIED                             VARCHAR2(10)
 ISADJUSTED                             VARCHAR2(5)
 ISDEPRECATED                           VARCHAR2(5)
 ISBASIC                                VARCHAR2(5)
 DESCRIPTION                            VARCHAR2(255)
 UPDATE_COMMENT                         VARCHAR2(255)
 HASH                                   NUMBER
 CON_ID                                 NUMBER
```

Resource Management (4:4)

- 47% of initialization parameters can be modified for the PDB
- None of those you can modify relate to memory management

```
SQL> SELECT ispdb_modifiable, COUNT(*)
  2  FROM v$parameter
  3  GROUP BY ispdb_modifiable;

ISPDB   COUNT(*)
----- ----------
TRUE         171
FALSE        196
```

```
ALTER SYSTEM SET sec_max_failed_login_attempts = 3 CONTAINER = ALL SCOPE = SPFILE;
```

In-Memory Database

- Looks very good but we do not yet have sufficient experience to know when and where to deploy it in production
  - Set up a POC and test with your application to find out if it is right for your applications and usage: The value can be dramatic

```
SQL> ALTER SYSTEM SET inmemory_size = 500M SCOPE = spfile;

 SQL> shutdown immediate;
 Database closed.
 Database dismounted.
 ORACLE instance shut down.
 SQL> startup;
 ORACLE instance started.

 Total System Global Area 2097152000 bytes
 Fixed Size 3047568 bytes
 Variable Size 1140854640 bytes
 Database Buffers 402653184 bytes
 Redo Buffers 13725696 bytes
 In-Memory Area 536870912 bytes
 Database mounted.
 Database opened.
```

12c Walk-Around

Focus Areas

- Disk layout
- ORACLE_BASE unchanged
- ORACLE_HOME unchanged
- One or more ORACLE_SIDs

12c Walk-Around: What Is Open? (1:3)

```
SQL> SELECT * FROM v$version;

BANNER                                                                          CON_ID
------------------------------------------------------------------------------ ----------
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production             0
PL/SQL Release 12.1.0.2.0 - Production                                          0
CORE    12.1.0.2.0      Production                                              0
TNS for 64-bit Windows: Version 12.1.0.2.0 - Production                         0
NLSRTL Version 12.1.0.2.0 - Production                                          0

SQL> sho user con_name con_id
USER is "SYS"

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

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

SQL> SELECT con_id, name, open_mode, total_size
  2  FROM v$pdbs;

    CON_ID NAME                           OPEN_MODE  TOTAL_SIZE
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY   283115520
         3 ORCL                           READ WRITE  393216000
         4 PDBDEV                         READ WRITE  429916160
```

12c Walk-Around: What Is Open? (2:3)

```
SQL> SELECT con_id, COUNT(*)
  2  FROM v$session
  3  GROUP BY con_id
  4  ORDER BY 1;

    CON_ID   COUNT(*)
---------- ----------
         0         30
         1          1
         3          1
         4          1
```

12c Walk-Around: What Is Open? (3:3)

- Your view depends on
  - Where you are
  - What PDBs are open

```
SQL> sho user
USER is "SYS"
SQL> SELECT COUNT(*) FROM user_objects;

  COUNT(*)
----------
     41911

SQL> SELECT COUNT(*) FROM dba_objects;

  COUNT(*)
----------
     90955

SQL> SELECT COUNT(*) FROM cdb_objects;

  COUNT(*)
----------
    181691

SQL> alter pluggable database all open;

Pluggable database altered.

SQL> SELECT COUNT(*) FROM cdb_objects;

  COUNT(*)
----------
    365993
```

12c Walk-Around: New Privileged Users

```
SQL> SELECT username
  2  FROM dba_users
  3  ORDER BY 1;

USERNAME
------------------------------
ANONYMOUS
APEX_040200
APEX_PUBLIC_USER
APPQOSSYS
AUDSYS
CTXSYS
DBSNMP
DIP
DVF
DVSYS
FLOWS_FILES
GSMADMIN_INTERNAL
GSMCATUSER
GSMUSER
LBACSYS
MDDATA
MDSYS
OJVMSYS
OLAPSYS
ORACLE_OCM
ORDDATA
ORDPLUGINS
ORDSYS
OUTLN
```

```
USERNAME
------------------------------
SI_INFORMTN_SCHEMA
SPATIAL_CSW_ADMIN_USR
SPATIAL_WFS_ADMIN_USR
SYS
SYSBACKUP
SYSDG
SYSKM
SYSTEM
WMSYS
XDB
XS$NULL

35 rows selected.
```

12c Walk-Around: Location Is Critically Important

```
SQL> SELECT username, COUNT(*)
  2  FROM cdb_users
  3  GROUP BY username
  4  ORDER BY 1;

USERNAME                       COUNT(*)
------------------------------ ----------
ANONYMOUS                               4
APEX_040200                             4
APEX_PUBLIC_USER                        4
APPQOSSYS                               4
AUDSYS                                  4
BI                                      1
CTXSYS                                  4
DBSNMP                                  4
DIP                                     4
DVF                                     4
DVSYS                                   4
EXFSYS                                  1
FINAPP                                  1
FLOWS_FILES                             4
GSMADMIN_INTERNAL                       4
GSMCATUSER                              4
GSMUSER                                 4
HR                                      1
IX                                      1
LBACSYS                                 4
MDDATA                                  4
MDSYS                                   4
MDSYS                                   4
```

```
MGDSYS                                  1
MLIB                                    1
NOAA                                    1
OE                                      1
OJVMSYS                                 4
OLAPSYS                                 4
ORACLE_OCM                              4
ORDDATA                                 4
ORDPLUGINS                              4
ORDSYS                                  4
OUTLN                                   4
PDBADMIN                                1
PM                                      1
RMANADMIN                               1
SCOTT                                   1
SH                                      1
SI_INFORMTN_SCHEMA                      4
SPATIAL_CSW_ADMIN_USR                   4
SPATIAL_WFS_ADMIN_USR                   4
SQLTXADMIN                              1
SQLTXPLAIN                              1
SYS                                     4
SYSBACKUP                               4
SYSDG                                   4
SYSKM                                   4
SYSTEM                                  4
UWCLASS                                 2
WMSYS                                   4
XDB                                     4
XS$NULL                                 4
```

Location Location Location

Location Matters

- You've been a DBA for years so answer these basic questions
  - You logon as SYS ... what database did you connect to?
  - You started the database ... what did you start?
  - You did a shutdown abort ... why can most users still connect and work?
  - The database just crashed ... what crashed?
  - You moved system01.dbf ... what did you move?
  - You created a tablespace but no one can use it ... why?
  - You created a new user ... what type and where?
  - You created a new role ... what type and where?
  - You patched the database ... what did you patch?
  - You performed an RMAN backup ... what did you back up?
  - You assigned privileges to a role and they do not work ... why?
  - Every cron job that worked in 11gR2 no longer works ... why?

STARTUP (1:3)

- What does STARTUP do?
  The answer depends on where you are
  - In CDB$ROOT starts the instance
    - The CDB$ROOT container is opened read/write
    - The PDB$SEED container is opened read only
    - **All PDBs are MOUNTed but not opened**
  - In any PDB
    - Changes the current PDB state from MOUNT to OPEN

STARTUP (2:3)

```
SQL> startup
ORACLE instance started.

Total System Global Area 2505338880 bytes
Fixed Size                  2405760 bytes
Variable Size             671091328 bytes
Database Buffers         1811939328 bytes
Redo Buffers               19902464 bytes
Database mounted.
Database opened.

SQL> sho user
USER is "SYS"

SQL> conn sys@pdbdev as sysdba
Enter password:
Connected.

SQL> sho user
USER is "SYS"

SQL> SELECT COUNT(*) FROM tab$;
SELECT COUNT(*) FROM tab$
                     *
ERROR at line 1:
ORA-01219: database or pluggable database not open: queries allowed on fixed tables or
views only
```

STARTUP (2:3)

```
SQL> conn / as sysdba
Connected.

SQL> SELECT con_id, name, open_mode
  2  FROM v$pdbs;

    CON_ID NAME                           OPEN_MODE
---------- ------------------------------ ----------
         2 PDB$SEED                       READ ONLY
         3 PDBORCL                        MOUNTED
         4 PDBDEV                         MOUNTED
```

STARTUP (3:3)

```
CREATE TABLE startup_pdbs(
pdb_name   VARCHAR2(30),
start_flag VARCHAR2(1))
PCTFREE 0;


INSERT INTO startup_pdbs (pdb_name, start_flag) VALUES ('PDBDEV', 'Y');
INSERT INTO startup_pdbs (pdb_name, start_flag) VALUES ('PDBORCL', 'N');


CREATE OR REPLACE NONEDITIONABLE TRIGGER sys.pdb_start_trig
AFTER STARTUP
ON DATABASE
BEGIN
  FOR r IN (SELECT pdb_name FROM startup_pdbs WHERE start_flag = 'Y') LOOP
    EXECUTE IMMEDIATE 'ALTER PLUGGABLE DATABASE ' || r.pdb_name || ' OPEN';
  END LOOP;
EXCEPTION
  WHEN OTHERS THEN
    NULL;
END pdb_start_trig;
/


ALTER TRIGGER sys.pdb_start_trig DISABLE;
ALTER TRIGGER sys.pdb_start_trig ENABLE;
```

SHUTDOWN

- What does SHUTDOWN IMMEDIATE do?
  The answer depends on where you are
  - In CDB$ROOT closes all PDBs
  - In any PDB
    - Changes the current state from OPEN to MOUNT
- Issue SHUTDOWN ABORT in one place and you can bring down everything or just close one PDB

SHUTDOWN ABORT in a PDB (1:2)

```
SQL> conn sys@pdbdev as sysdba
Enter password:
Connected.

SQL> SELECT open_mode
  2  FROM v$pdbs;

OPEN_MODE
----------
READ WRITE

SQL> shutdown abort;
ORA-00603: ORACLE server session terminated by fatal error
ORA-00372: file 5 cannot be modified at this time
ORA-01110: data file 5: 'C:\APP\ORACLE\ORADATA\ORABASE2\UNDOTBS01.DBF'
ORA-00372: file 5 cannot be modified at this time
ORA-01110: data file 5: 'C:\APP\ORACLE\ORADATA\ORABASE2\UNDOTBS01.DBF'
Process ID: 7236
Session ID: 355 Serial number: 11
```

SHUTDOWN ABORT in a PDB (2:2)

```
SQL> conn sys@pdbdev as sysdba
Enter password:
Connected.

SQL> SELECT open_mode
  2  FROM v$pdbs;
ERROR:
ORA-03114: not connected to ORACLE

SQL> conn / as sysdba
Connected.

SQL> SELECT name, open_mode
  2  FROM v$pdbs;

NAME                           OPEN_MODE
------------------------------ ----------
PDB$SEED                       READ ONLY
PDBORCL                        READ WRITE
PDBDEV                         MOUNTED
```

Data and Metadata Visibility

- In CDB$ROOT you can see PDB metadata ... but not data
- In a PDB you can see only what is in that PDB
- ASH and AWR
  - AWR Report run in CDB$ROOT sees all
  - AWR Report run in a PDB sees only that one PDB

Important Changes For DBAs

Common Users and Roles

- Common users and roles can only be created in CDB$ROOT

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

SQL> CREATE USER oracle1
  2  IDENTIFIED BY oracle1;
CREATE USER oracle1
            *
ERROR at line 1:
ORA-65096: invalid common user or role name

SQL> CREATE USER c##oracle1
  2  IDENTIFIED BY oracle1;

User created.
```

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

SQL> CREATE USER c##oracle1
  2  IDENTIFIED BY oracle1;
CREATE USER c##oracle1
            *
ERROR at line 1:
ORA-65094: invalid local user or role name

SQL> CREATE USER oracle1
  2  IDENTIFIED BY oracle1;

User created.
```

cron Jobs and Shell Scripts

- 12c breaks almost every shell script ever written
- 12c makes cron jobs a Paleolithic artifact
- It is time to put your scheduled jobs where they belong: Into DBMS_SCHEDULER

```
BEGIN
   dbms_scheduler.create_job(
   job_name   => 'EXT_LOAD',
   job_type   => 'EXECUTABLE',
   job_action => 'c:\app\oracle\product\12.1.0\db_1\bin\sqlldr.exe',
   number_of_arguments => 1,
   start_date => SYSTIMESTAMP,
   enabled    => FALSE,
   comments   => 'SQL*Loader Job Demo');

   dbms_scheduler.set_job_argument_value('EXT_LOAD', 1,
     argument_value => 'userid=uwclass/uwclass
     control=c:\temp\sqlldr02.ctl log=c:\temp\sqlldr02.log');

   dbms_scheduler.enable('EXT_LOAD');
 END;
 /
```

Memory Management

- Database 10g introduced ASMM
- Database 11g introduced AMM
- Database 12c acknowledges that AMM was a failure and returns to ASMM
  - As does 11.2.0.4
  - Highly recommend all 11g databases be moved to ASMM to reduce resizing and locking issues

Important Changes For Developers

Developer Changes

- From an application code standpoint ... none
- From the standpoint of logging and exception handling
  - Do you know which instance (RAC)
  - Do you know which host (DataGuard Standby)
  - Do you know which edition (Edition Based Redefinition)
  - Do you know which PDB (Container DB)

```
SELECT sys_context('USERENV', 'SERVICE_NAME') FROM dual;
SELECT sys_context('USERENV', 'INSTANCE_NAME') FROM dual;
SELECT sys_context('USERENV', 'HOST') FROM dual;

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

SELECT sys_context('USERENV', 'CURRENT_EDITION_NAME') FROM dual;
 SELECT sys_context('USERENV', 'SESSION_EDITION_NAME') FROM dual;

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

  - Logging handling needs to record "where" not just "what"

Operating System Groups & Enhanced Priv. Users

- Legacy Operating System
  - users
    - *root*
    - *oracle*
  - groups
    - *oinstall*
    - *dba*
    - *oper*
- Legacy Database
  - users
    - *sys*
    - *system*
  - roles
    - *dba*

Users with Escalated Privileges

```
SQL> SELECT username FROM dba_users ORDER BY 1;

USERNAME
------------------------------
ANONYMOUS
APEX_040200
APEX_PUBLIC_USER
APPQOSSYS
AUDSYS
CTXSYS
DBSNMP
DIP
DVF
DVSYS
FLOWS_FILES
GSMADMIN_INTERNAL
GSMCATUSER
GSMUSER
LBACSYS
MDDATA
MDSYS
OJVMSYS
OLAPSYS
ORACLE_OCM
ORDDATA
ORDPLUGINS
ORDSYS
OUTLN
SI_INFORMTN_SCHEMA
SPATIAL_CSW_ADMIN_USR
SPATIAL_WFS_ADMIN_USR
SYS
SYSBACKUP
SYSDG
SYSKM
SYSTEM
WMSYS
XDB
XS$NULL
```

New Roles Enhancing Separation of Duties

```
SQL> SELECT * FROM dba_roles ORDER BY 1;

ROLE                           PASSWORD AUTHENTICAT COM O
------------------------------ -------- ---------- --- -
ADM_PARALLEL_EXECUTE_TASK      NO       NONE       YES Y
APEX_ADMINISTRATOR_ROLE        NO       NONE       YES Y
APEX_GRANTS_FOR_NEW_USERS_ROLE NO       NONE       YES Y
AQ_ADMINISTRATOR_ROLE          NO       NONE       YES Y
AUDIT_ADMIN                    NO       NONE       YES Y
CDB_DBA                        NO       NONE       YES Y
DBA                            NO       NONE       YES Y
DBFS_ROLE                      NO       NONE       YES Y
GATHER_SYSTEM_STATISTICS       NO       NONE       YES Y
GSMADMIN_ROLE                  NO       NONE       YES Y
GSMUSER_ROLE                   NO       NONE       YES Y
GSM_POOLADMIN_ROLE             NO       NONE       YES Y
HS_ADMIN_EXECUTE_ROLE          NO       NONE       YES Y
HS_ADMIN_ROLE                  NO       NONE       YES Y
HS_ADMIN_SELECT_ROLE           NO       NONE       YES Y
LBAC_DBA                       NO       NONE       YES Y
LOGSTDBY_ADMINISTRATOR         NO       NONE       YES Y
OLAP_DBA                       NO       NONE       YES Y
OLAP_XS_ADMIN                  NO       NONE       YES Y
OPTIMIZER_PROCESSING_RATE      NO       NONE       YES Y
ORDADMIN                       NO       NONE       YES Y
PDB_DBA                        NO       NONE       YES Y
PROVISIONER                    NO       NONE       YES Y
RECOVERY_CATALOG_OWNER         NO       NONE       YES Y
SCHEDULER_ADMIN                NO       NONE       YES Y
SPATIAL_CSW_ADMIN              NO       NONE       YES Y
SPATIAL_WFS_ADMIN              NO       NONE       YES Y
WM_ADMIN_ROLE                  NO       NONE       YES Y
XDBADMIN                       NO       NONE       YES Y
XS_CACHE_ADMIN                 NO       NONE       YES Y
XS_NAMESPACE_ADMIN             NO       NONE       YES Y
XS_SESSION_ADMIN               NO       NONE       YES Y
```

System Privileges

- New system privileges related to new capabilities and the new container architecture

```
SQL> SELECT /* edited listing */ UNIQUE privilege
     FROM dba_sys_privs;

PRIVILEGE
----------------------------------
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
EXEMPT DDL REDACTION POLICY
EXEMPT DML REDACTION POLICY
REDEFINE ANY TABLE
SET CONTAINER
USE ANY SQL TRANSLATION PROFILE
```

Credentials

- DBMS_CREDENTIAL
  Interface for authenticating and impersonating EXTPROC callout functions, as well as external jobs, remote jobs and file watchers from the SCHEDULER

```
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;
/
```

Identity Columns (1:2)

```
SQL> CREATE TABLE t1 (
  2  rid NUMBER GENERATED ALWAYS AS IDENTITY,
  3  col VARCHAR2(3));

Table created.

SQL> desc t1
 Name                          Null?    Type
 ----------------------------- -------- ------------
 RID                           NOT NULL NUMBER
 COL                                    VARCHAR2(3)

SQL> INSERT INTO t1 (col) VALUES ('A');

1 row created.

SQL> INSERT INTO t1 (col) VALUES ('B');

1 row created.

SQL> INSERT INTO t1 (col) VALUES ('C');

1 row created.

SQL> SELECT * FROM t1;

       RID COL
---------- ---
         1 A
         2 B
         3 C
```

Identity Columns (2:2)

```
desc dba_tab_cols
desc dba_tab_columns

SQL> SELECT *
  2  FROM sys.idnseq$;

      OBJ#    INTCOL#    SEQOBJ#  STARTWITH
---------- ---------- ---------- ----------
     96482          1      96483          1

CREATE TABLE t2 (
rid NUMBER GENERATED ALWAYS AS IDENTITY INCREMENT BY 2,
col VARCHAR2(3));

 CREATE TABLE t3 (
 rid NUMBER GENERATED ALWAYS AS IDENTITY
  INCREMENT BY 2
  START WITH 100
  MAXVALUE 110
  MINVALUE 100
  CYCLE
  CACHE 5
  NOORDER,
 col VARCHAR2(3));
```

Advanced Index Compression

- Another good reason to get the Advanced Compression license
  - Save disk space which saves money
  - Improved performance

```
CREATE INDEX ix_airplanes_custid_deldate
ON uwclass.airplanes(customer_id, delivered_date)
COMPRESS ADVANCED LOW;
```

Other Changes Important for DBAs

- 32K VARCHAR2s in tables and views
- SecureFiles are now the default storage for LOBs
  - DataPump can now import LOB columns as SecureFiles

New Features

Automatic Data Optimization (ADO)

"To implement an ILM strategy, use database Heat Maps to track data access and modification. Automatic Data Optimization (ADO) can also automate the compression and movement of data between storage tiers. The DBMS_ILM package supports immediate evaluation or execution of Automatic Data Optimization (ADO) related tasks. The package supports the following two ways for scheduling ADO actions.

- A database user schedules immediate ADO policy execution on a set of objects
- A database user views the results of evaluation of ADO policies on a set of objects

The user then adds or deletes objects from the set and reviews the results of ADO policy evaluation again. Repeat these steps to determine the set of objects for ADO execution. You can then perform immediate scheduling of ADO actions on this set of objects.

The following procedures support the two usage modes. Before describing the procedures, we introduce the notion of an ADO task as an entity that helps to track a particular evaluation or (an evaluation and execution) of ADO policies. A particular ADO task could be in one of the following states: active, completed, and inactive.

Integrated Lifecycle Management (ILM)

- Essentially in-database automated data tiering
- Three main packages
  - DBMS_ILM
    - *Provides an interface for implementing Information Lifecycle Management (ILM) strategies using Automatic Data Optimization (ADO) policies.*
  - DBMS_ILM_ADMIN
    - *Provides an interface to customize Automatic Data Optimization (ADO) policy execution. In combination with partitioning and compression, ADO policies can be used to help implement an Information Lifecycle Management (ILM) strategy.*
  - DBMS_HEAT_MAP
    - *Provides heatmap information at block/extent/segment object and tablespace levels.*
    - *Contains the definitions for processing heatmaps for top N objects and tablespaces.*
    - *The execution privilege is granted to PUBLIC. Procedures in this package run under the caller security. The user must have ANALYZE privilege on the object.*

ADO Code Examples

```
-- Tablespace Level Policy
ALTER TABLE uwclass.servers
ILM ADD POLICY TIER TO ilmtbs;

-- Compress segment (table/partition) after there have been no modifications for one week
ALTER TABLE uwclass.servers
ILM ADD POLICY ROW STORE
COMPRESS ADVANCED segment AFTER 7 DAYS OF NO MODIFICATIONS;

 -- Compress  rows after no changes for 2 weeks
ALTER TABLE uwclass.servers
ILM ADD POLICY ROW STORE
COMPRESS ADVANCED row AFTER 14 DAYS OF NO MODIFICATIONS;
```

Heat Map Code Examples

```
-- Return the block level ILM statistics for a segment
SELECT *
FROM TABLE(dbms_heat_map.block_heat_map('SYS', 'OBJ$'));

-- Run the advisor job to analyze heat maps
exec dbms_heat_map.auto_advisor_heatmap_job;

SELECT * FROM wri$_heatmap_top_tablespaces;

SELECT * FROM wri$_heatmap_top_objects;

 -- Return the minimum, maximum and average access times for all the segments
-- belonging to an object.
SELECT tablespace_name, segment_name, segment_type, segment_size,
avg_readtime, max_lookuptime
FROM TABLE(dbms_heat_map.object_heat_map('SYS', 'TAB$'));
```

View ILM Parameters

```
-- Examine the current ILM execution environment

SQL> SELECT *
  2  FROM ilm_param$;

    PARAM# PARAM_NAME                 PARAM_VALUE
---------- ------------------------- -----------
         7 ENABLED                             1
         5 JOB LIMIT                          10
         4 EXECUTION MODE                      3
         1 EXECUTION INTERVAL                 15
         8 TBS PERCENT USED                   85
         9 TBS PERCENT FREE                   25
```

Modify ILM Parameters

```
-- Customize the ILM execution environment to take effect with the next job run

DECLARE
 param NUMBER := dbms_ilm_admin.joblimit;
 pval  NUMBER := 9;
BEGIN
  dbms_ilm_admin.customize_ilm(param, pval);
  dbms_ilm_admin.customize_ilm(dbms_ilm_admin.tbs_percent_used, 90);
  dbms_ilm_admin.customize_ilm(dbms_ilm_admin.tbs_percent_free, 20);
  dbms_ilm_admin.customize_ilm(dbms_ilm_admin.enabled, 15);
END;
/
```

Store Java Script Object Notation (JSON) Documents

```
-- Create a table holding JSON documents

SQL> CREATE TABLE j_purchase_order (
     doc_id       RAW (16) NOT NULL,
     date_loaded TIMESTAMP(6) WITH TIME ZONE,
     po_document CLOB);

SQL> desc j_purchase_order
Name                          Null?    Type
----------------------------- -------- ----------------------------
DOC_ID                        NOT NULL RAW(16)
DATE_LOADED                            TIMESTAMP(6) WITH TIME ZONE
PO_DOCUMENT                            CLOB

ALTER TABLE j_purchase_order
ADD CONSTRAINT ensure_json
CHECK (po_document IS JSON);
```

Insert a JSON Document

```
INSERT INTO j_purchase_order VALUES (SYS_GUID(), SYSTIMESTAMP,
'{"PONumber" : 1600,
"Reference" : "SCATZ-20140421",
"Requestor" : "Safra Catz",
"User" : "ABULL",
"CostCenter" : "A50",
"ShippingInstructions" : {"name" : "Lawrence Ellison",
                          "address": {"street" : "500 Marine Drive",
                          "city" : "Redwood Shores",
                          "state" : "CA",
                          "zipCode" : 94000,
                          "country" : "United States of America"},
                          "phone" : [{"type" : "Office", "number" : "650-506-7000"},
                                     {"type" : "Mobile", "number" : "415-555-1234"}]},
"Special Instructions" : null,
"AllowPartialShipment" : true,
"LineItems" : [{"ItemNumber" : 1,
                "Part" : {"Description" : "Hawaiian Island: Lanai",
                          "UnitPrice" : 99.95,
                          "UPCCode" : 13131092899},
                          "Quantity" : 1.0},
               {"ItemNumber" : 2,
                "Part" : {"Description" : "Hotels on Boardwalk",
                          "UnitPrice" : 19.95,
                          "UPCCode" : 85391628927},
                          "Quantity" : 4.0}]}');
```

SELECTing a JSON Document

```
SELECT jt.phones
FROM j_purchase_order,
JSON_TABLE(po_document, '$.ShippingInstructions'
COLUMNS (phones VARCHAR2(100) FORMAT JSON PATH '$.Phone')) AS jt;


SELECT jt.*
FROM j_purchase_order,
JSON_TABLE(po_document, '$.ShippingInstructions.Phone[*]'
COLUMNS (row_number FOR ORDINALITY, phone_type VARCHAR2(10)
PATH '$.type', phone_num VARCHAR2(20) PATH '$.number')) AS jt;
```

New RMAN Recovery Capabilities

- SYSBACKUP Privilege
- Active Database Deduplication
- Use the NOOPEN command to specify a new database not be opened using RESETLOGS
- PDB-Level Recovery
- Single Table Recovery
- Single Partition Recovery
- Improved support for third-party snapshot technologies
- Most SQL commands, for example DESC, be issued in RMAN

```
RECOVER TABLE SCOTT.EMP UNTIL SEQUENCE 5466
AUXILARY DESTINATION '/tmp/recover'
REMAP TABLE 'SCOTT'.'EMP':'MY_EMP';
```

Full Database Caching

- In-Memory Column Store
  - `/* INMEMORY([@queryblock] <tablespace>) */`
  - `/* INMEMORY_PRUNING ([@queryblock] <tablespace>) */`
- Full Database Caching

```
ALTER DATABASE FORCE FULL DATABASE CACHING;

SELECT force_full_database_caching
FROM v$database;
```

Conclusions (1:2)

- Many of these new features require substantial effort to master and will be heavily promoted by Oracle sales teams
- You are not going to be competent or conversant in 12c if you are not investing time in reading the docs and writing demos
- Your first experience with these features should be in a sandbox environment not on your employer's production databases
- You MUST download and read the docs or you will fail
- You MUST download and install a copy of 12cR1 and start making mistakes in your own computing environment so you don't make them at work
- You must rewrite diagnostic scripts to account for the differences in architecture

Conclusions (2:2)

- You must rewrite cron jobs to account for the differences in architecture
- Exception handling should be enhanced to provide persistent storage of
  - server host in case if RAC is deployed
  - primary vs. standby if DataGuard is deployed
  - edition name if Edition Based Redefinition (EBR) is deployed
  - container_id and/or name for 12c multi-tenant deployments
- Customers wanting to move from 10g or 11g to 12c will require extra time and extensive planning to correct legacy issues before the move

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

Thank you