CDB那些事儿

公益讲座11：00准时开始，请大家先浏览云技术微信公众号技术文章。资料会在各群同步发布，已入群客户请勿重复入群！

20-21

数据库和云讲座群

甲骨文云技术公众号

B站专家系列课程

CDB 那些事儿

甲骨文技术公益课 - 数据库专场

2023 年 7 月 28 日 11:00

线上直播

郭俊龙

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

CDB那些事儿

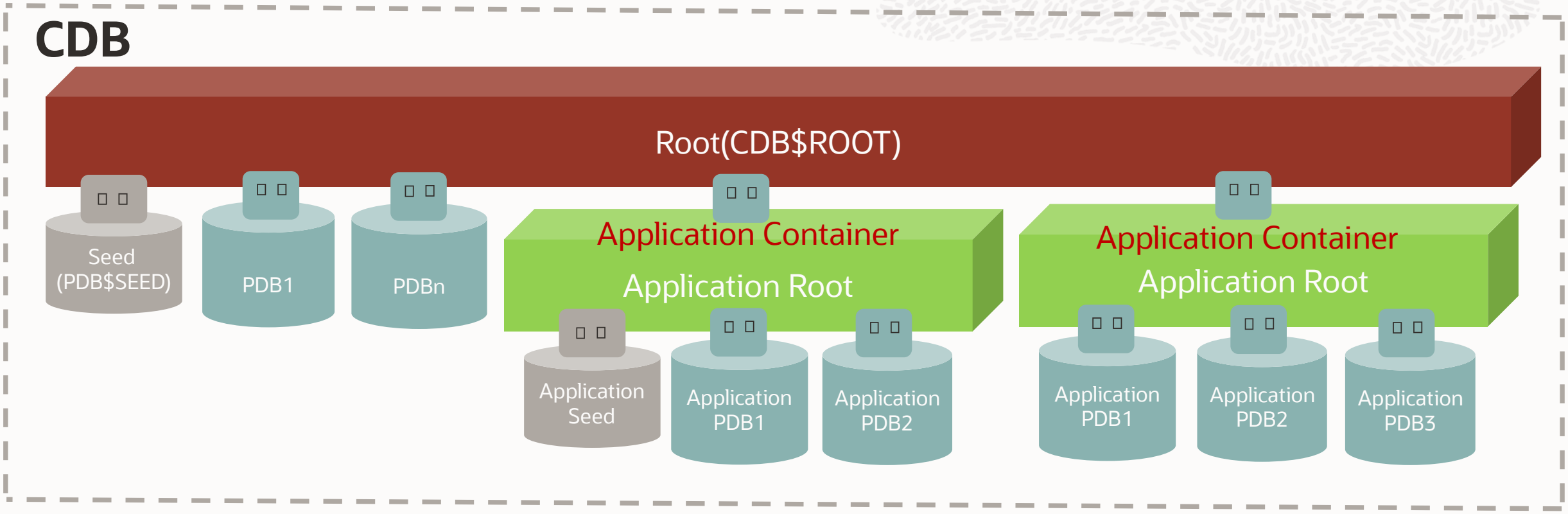
目录

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

1. CDB架构

1.1 CDB逻辑架构：



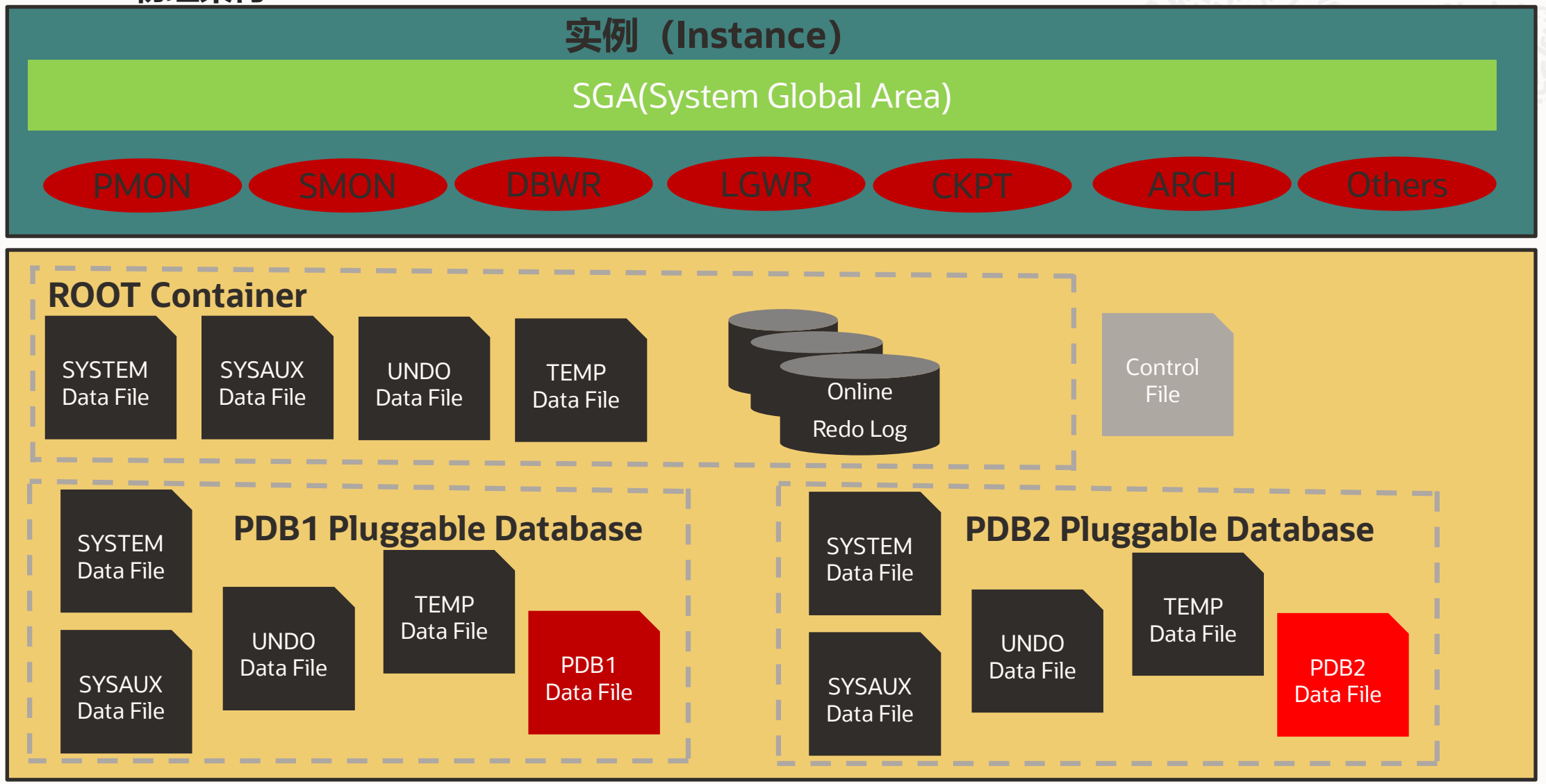
- ✓ CDB$ROOT: 有且只有一个CDB$ROOT。
- ✓ PDB$SEED: 有且只有一个PDB$SEED。
- ✓ PDB: 有零个或多个用户创建的PDB。
- ✓ Application Root: 有零个或多个Application Root。
- ✓ Application Seed: 有零个或多个Application Seed。
- ✓ Application PDB: 有零个或多个Application PDB。

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

1. CDB架构

1.2 CDB物理架构：



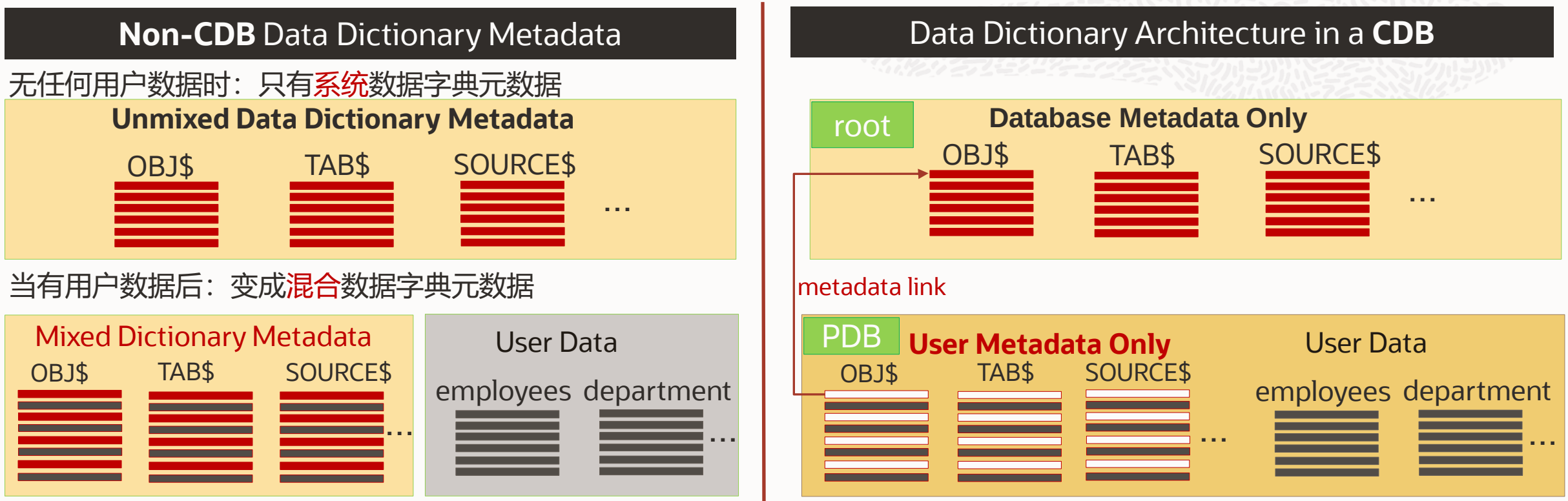
说明：共享实例资源、控制文件、REDO等，每个PDB拥有自己的数据文件: SYSTEM、SYSAUX、UNDO、TEMP等。

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

1. CDB架构

1.3 多租户数据字典架构：



CDB的数据字典架构具有如下两个好处：

1）减少重复

例如，CDB不需要将DBMS_ADVISOR PL/SQL包的源代码存储在每个PDB中，而是仅将其存储在CDB$ROOT中，这样可以节省磁盘空间。

2）易于数据库升级

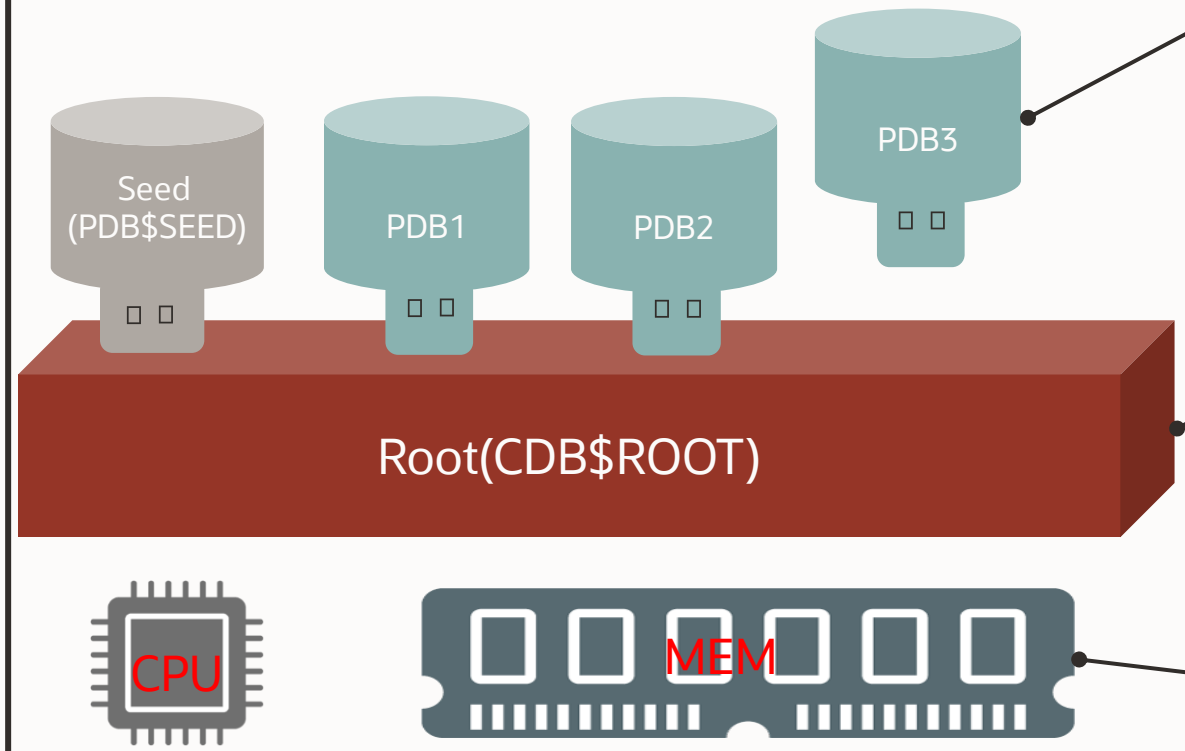
如果数据字典表的定义存在于每个PDB 中，并且定义在新版本中发生更改，则每个PDB都需要单独升级以应用这些更改。而在Root中只存储一次表定义，因此可消除此问题。

Copyright © 2023, Oracle and/or its affiliates
免费咨询热线：400-699-8888

1. CDB架构

1.4 多租户架构优势：

➢ 降低数据库整体成本支出，提高灵活性，易于采用和使用。



每个应用程序独立的PDB

- ✓ 应用程序无需修改
- ✓ 快速供给（通过克隆）
- ✓ 可移植性（通过可插拔性）

公共操作在CDB级执行

- ✓ CDB级别整体管理
  （如升级、打补丁、HA、备份）
- ✓ 通过参数，资源管理器等进行PDB级别管理
- ✓ 结合OEM进行统一管理

实例级别共享CPU、内存和后台进程

- ✓ 每台服务器运行更多的应用程序
- ✓ 对比VM模式能最大化利用系统资源

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

2. 资源管理

2.1 资源管理级别：

在CDB中，多个PDB内的工作负载可能会争夺系统和CDB资源，资源管理计划能解决了这个问题。
在多租户环境中，资源管理器（Resource Manager）运行在两个级别上：

CDB级别：

资源管理器（Resource Manager）可以管理争用系统和CDB资源的多个PDB的工作负载。
您可以指定如何将资源分配给PDB，还可以限制特定PDB的资源利用率。
主要工具是：CDB resource plan。

PDB级别：

资源管理器（Resource Manager）可以管理每个PDB中的工作负载。
主要工具是：PDB resource plan 。

资源管理器通过两步来分配资源：

第一步：将系统的一部分资源分配给每个PDB。
第二步：在特定的PDB中，将第一步中获得的系统资源的一部分分配给连接到该PDB的每个会话。

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

2. 资源管理

2.2 PDB资源管理：

PDB: CPU管理

通过CDB资源管理器（Resource Manager）控制：Shares、utilization_limit、parallel_server_limit
PDB实例囚笼（Instance caging）控制：通过初始化参数CPU_COUNT和RESOURCE_MANAGER_PLAN

PDB: 内存管理

通过初始化参数控制：
SGA: SGA_MIN_SIZE、SGA_TARGET
PGA: PGA_AGGREGATE_LIMIT、PGA_AGGREGATE_TARGET

PDB: 会话管理

通过初始化参数控制：SESSIONS、MAX_IDLE_TIME、MAX_IDLE_BLOCKER_TIME

PDB: I/O管理

通过初始化参数控制：MAX_IOPS、MAX_MBPS （注意：Exadata采用IORM来管理）

PDB: 存储管理

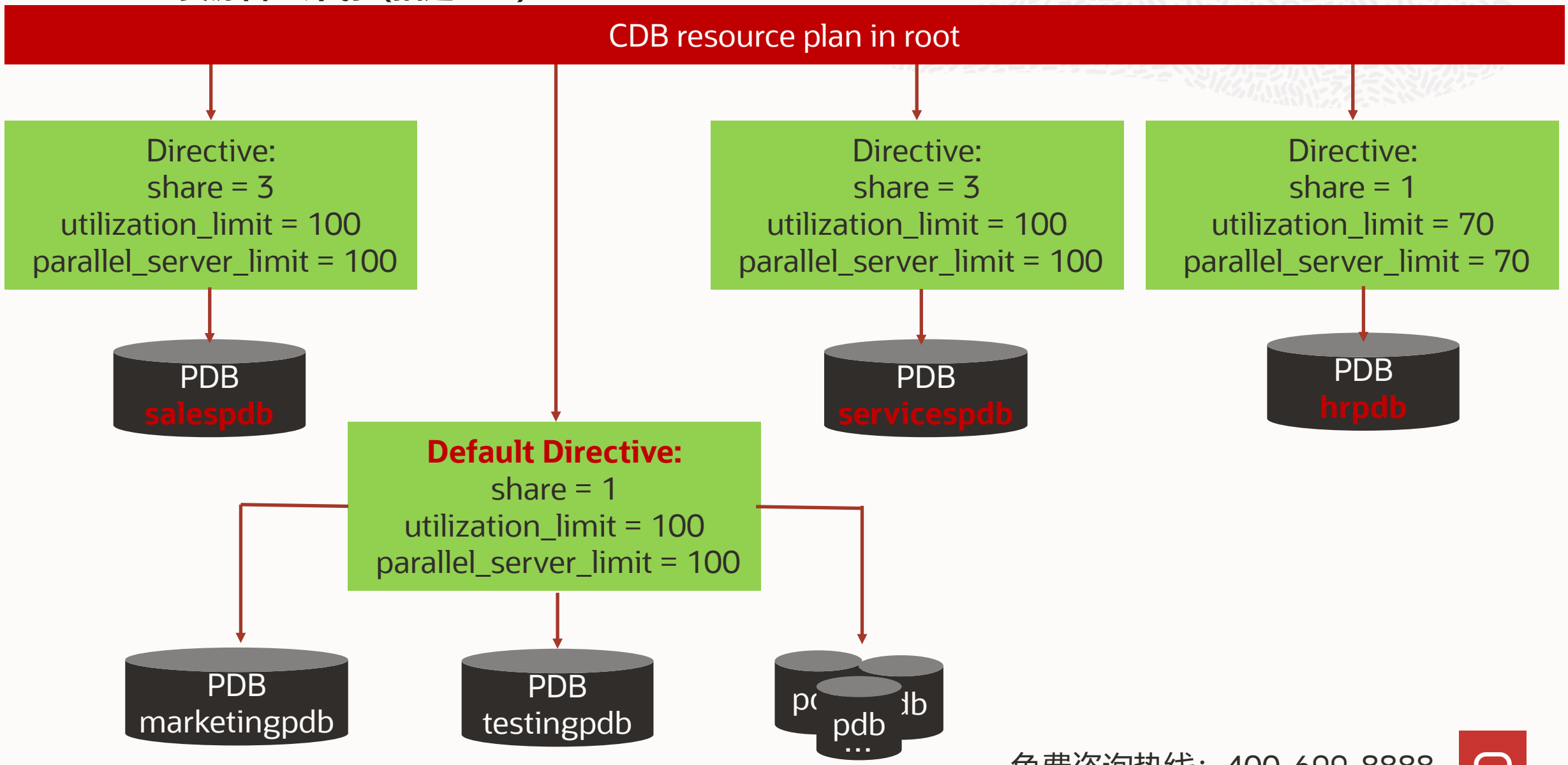
创建PDB|修改PDB属性：STORAGE(MAXSIZE *x*G)

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

2. 资源管理

2.3 CDB资源管理计划（指定PDB）：



免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

2. 资源管理

2.4 创建CDB资源管理计划: （给每个pdb指定分配的资源）

1）创建PENDING AREA：

```
SQL>exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

2）创建资源管理计划：

```
SQL>BEGIN DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN(
        plan => 'newcdb_plan',comment => 'CDB resource plan for newcdb');
        END;
       /
```

3）创建资源管理计划指令：

```
SQL> BEGIN DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
        plan => 'newcdb_plan',pluggable_database => 'salespdb',shares => 3,utilization_limit => 100,parallel_server_limit => 100);
        END;
         /
SQL> BEGIN DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
         plan => 'newcdb_plan',pluggable_database => 'servicespdb',shares => 3,utilization_limit => 100,parallel_server_limit => 100);
         END;
         /
SQL> BEGIN DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
         plan => 'newcdb_plan',pluggable_database => 'hrpdb',shares => 1,utilization_limit => 70,parallel_server_limit => 70);
         END;
         /
```

4）验证PENDING AREA：

```
SQL> exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5）提交PENDING AREA：

```
SQL>  exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

6）启动资源计划：

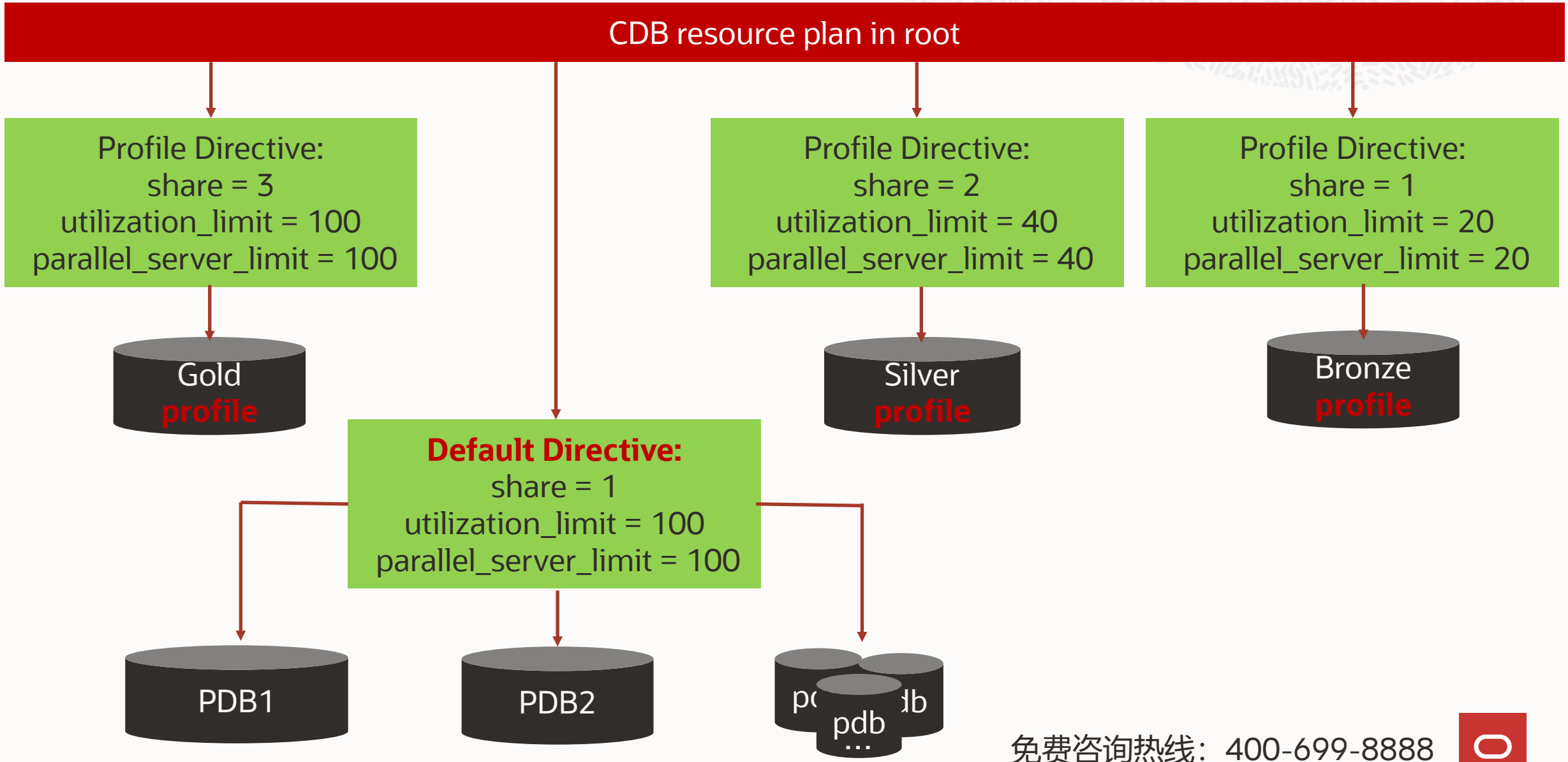
```
SQL> ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'newcdb_plan';
```

 Copyright © 2023, Oracle and/or its affiliates

免费咨询热线：400-699-8888

2. 资源管理

2.5 CDB资源管理计划（指定Performance Profiles ）：



 Copyright © 2023, Oracle and/or its affiliates

免费咨询热线：400-699-8888

2. CDB资源管理

2.6 创建CDB资源管理计划: （设置不同的Performance Profiles)

1）创建PENDING AREA：

```
SQL>exec DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
```

2）创建资源管理计划：

```
SQL>BEGIN DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN(
        plan => 'newcdb_plan',comment => 'CDB resource plan for newcdb');
        END;
        /
```

3）创建资源管理计划指令：

```
SQL> BEGIN DBMS_RESOURCE_MANAGER.CREATE_CDB_PROFILE_DIRECTIVE(
        plan => 'newcdb_plan',profile => 'gold',shares => 3,utilization_limit => 100,parallel_server_limit => 100);
        END;
         /
SQL> BEGIN DBMS_RESOURCE_MANAGER.CREATE_CDB_PROFILE_DIRECTIVE(
         plan => 'newcdb_plan',profile => 'silver',shares => 2,utilization_limit => 40,parallel_server_limit => 40);
         END;
        /
SQL> BEGIN DBMS_RESOURCE_MANAGER.CREATE_CDB_PROFILE_DIRECTIVE(
          plan => 'newcdb_plan',profile => 'bronze',shares => 1,utilization_limit => 20,parallel_server_limit => 20);
          END;
          /
```

4）验证PENDING AREA：

```
SQL> exec DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
```

5）提交PENDING AREA：

```
SQL>  exec DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
```

6）给pdb指定DB_PERFORMANCE_PROFILE参数：

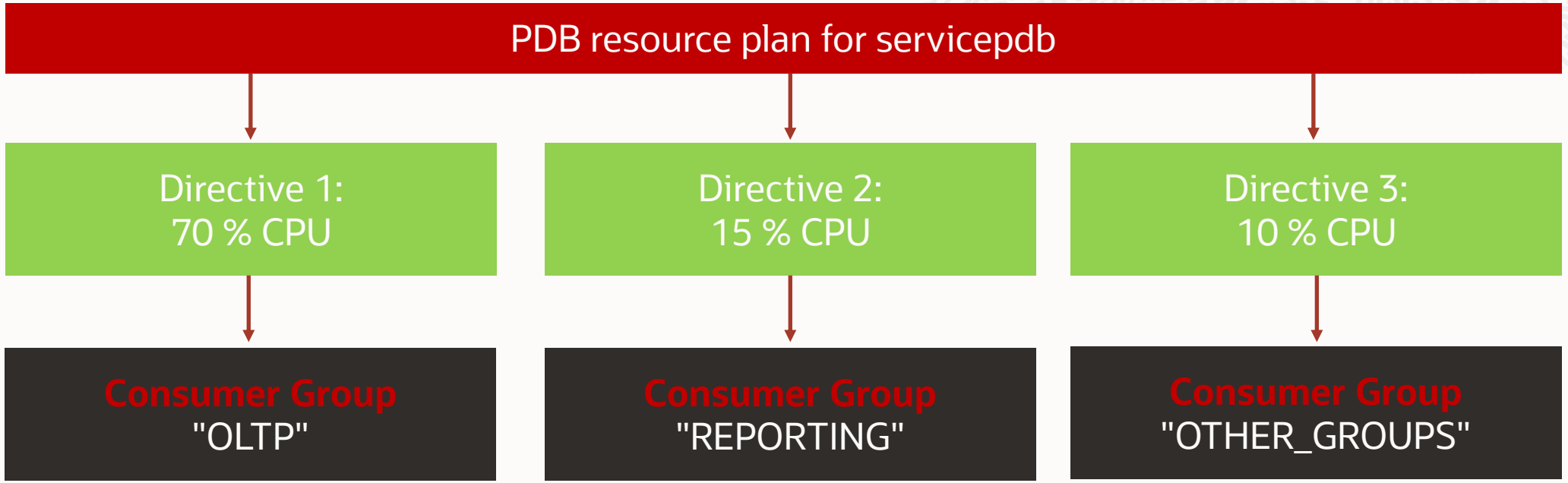
```
SQL>ALTER SYSTEM SET DB_PERFORMANCE_PROFILE=gold SCOPE=spfile;
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

2. 资源管理

2.7 PDB资源管理计划: （设置不同的Consumer Group ）



- ✓ PDB资源管理计划将特定PDB的资源分配给该PDB内的各个消费者组 （Consumer group ）。
- ✓ PDB资源管理计划类似于non-CDB的资源管理计划。
- ✓ PDB资源管理计划不同于CDB资源管理计划， CDB资源管理计划给每个PDB的分配资源。

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

2. CDB资源管理

2.7 PDB资源管理计划: （示例）

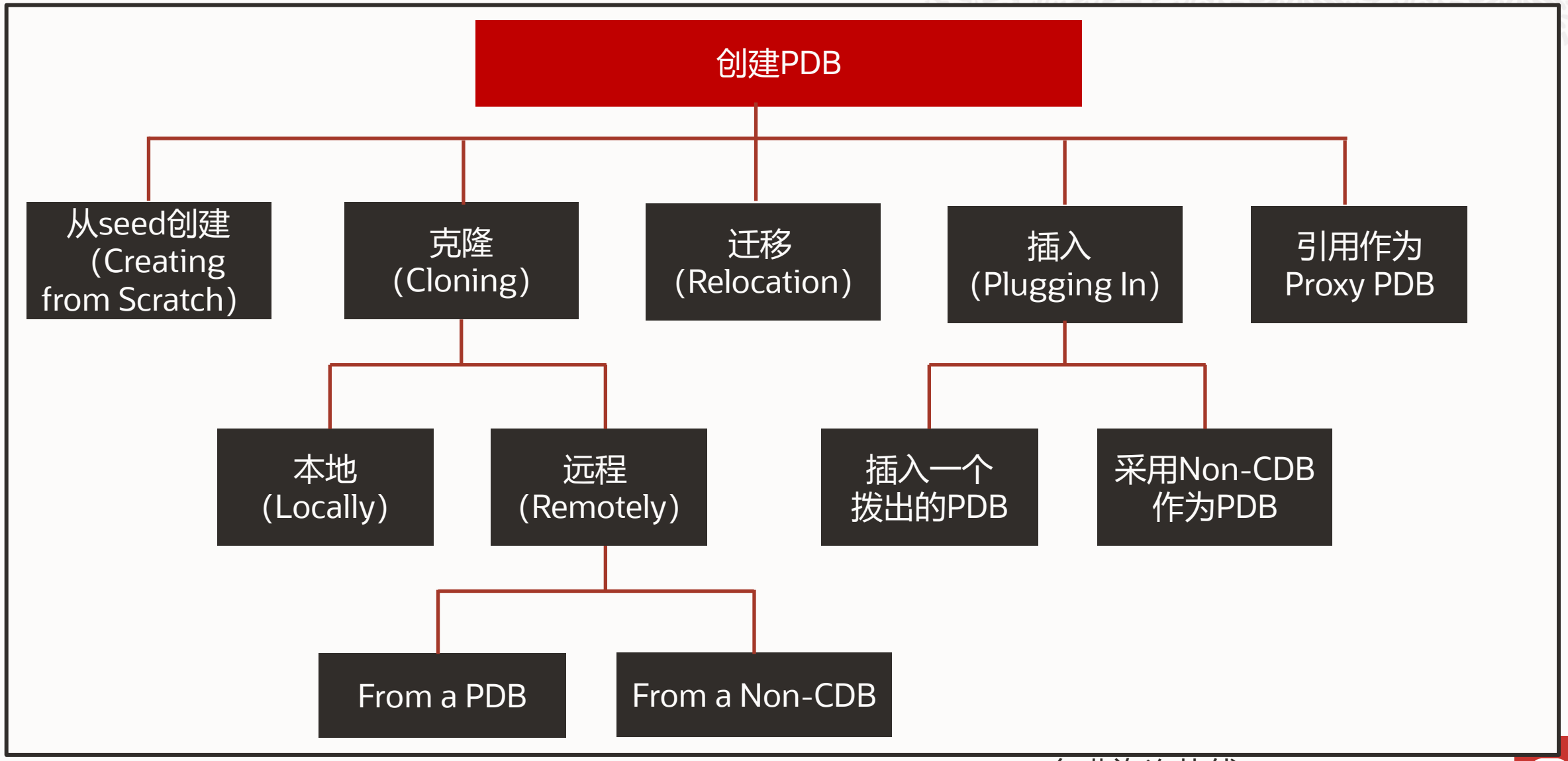
```
SQL> ALTER SESSION SET CONTAINER=servicepdb;

SQL> BEGIN
        DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
        DBMS_RESOURCE_MANAGER.CREATE_PLAN(PLAN => 'cgplan',COMMENT => 'cgplan');
        DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP(CONSUMER_GROUP =>
        'oltp', COMMENT => 'RMCG for OLTP jobs');
        DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP(CONSUMER_GROUP =>
        'reporting', COMMENT => 'RMCG for BATCH jobs');
        DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN => 'cgplan',
        GROUP_OR_SUBPLAN => 'oltp', COMMENT => 'OLTP sessions', MGMT_P1 => 70);
        DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN => 'cgplan',
        GROUP_OR_SUBPLAN => 'reporting', COMMENT => 'BATCH sessions', MGMT_P1 => 15);
        DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(PLAN => 'cgplan',
         GROUP_OR_SUBPLAN => 'OTHER_GROUPS', COMMENT => 'mandatory', MGMT_P1 => 10);
        DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
        DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
        END;
        /
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

2. 资源管理

2.8 PDB创建方式:



免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

3. PDB快照

3.1快照（Snapshot）：

PDB快照是一个PDB指定时间点的副本。在创建快照时，源PDB可以是只读的或读/写的。
可以创建手动或自动的快照，手动快照使用CREATE PLUGGABLE DATABASE|ALTER PLUGGABLE DATABASE加SNAPSHOT子句，自动快照使用EVERY interval子句，如果存储系统支持sparse clones，则会创建一个sparse copy，否则会创建一个full copy。

创建手动快照：

```
SQL> alter session set container=<pdb_name>;
SQL> ALTER PLUGGABLE DATABASE SNAPSHOT [snapshot_name];
```

注意：如果不指定snapshot_name，则由数据库自动分配以SNAP_开头的快照名称。

查看快照信息：

```
SQL> SELECT CON_ID AS id, CON_NAME, SNAPSHOT_NAME,
            SNAPSHOT_SCN AS snap_scn, FULL_SNAPSHOT_PATH
     FROM DBA_PDB_SNAPSHOTS
     ORDER BY SNAP_SCN;
```

删除快照：

```
SQL> ALTER PLUGGABLE DATABASE DROP SNAPSHOT [snapshot_name];
或： SQL> ALTER PLUGGABLE DATABASE SET MAX_PDB_SNAPSHOTS=0;
```

设置为0会快速删除所有快照

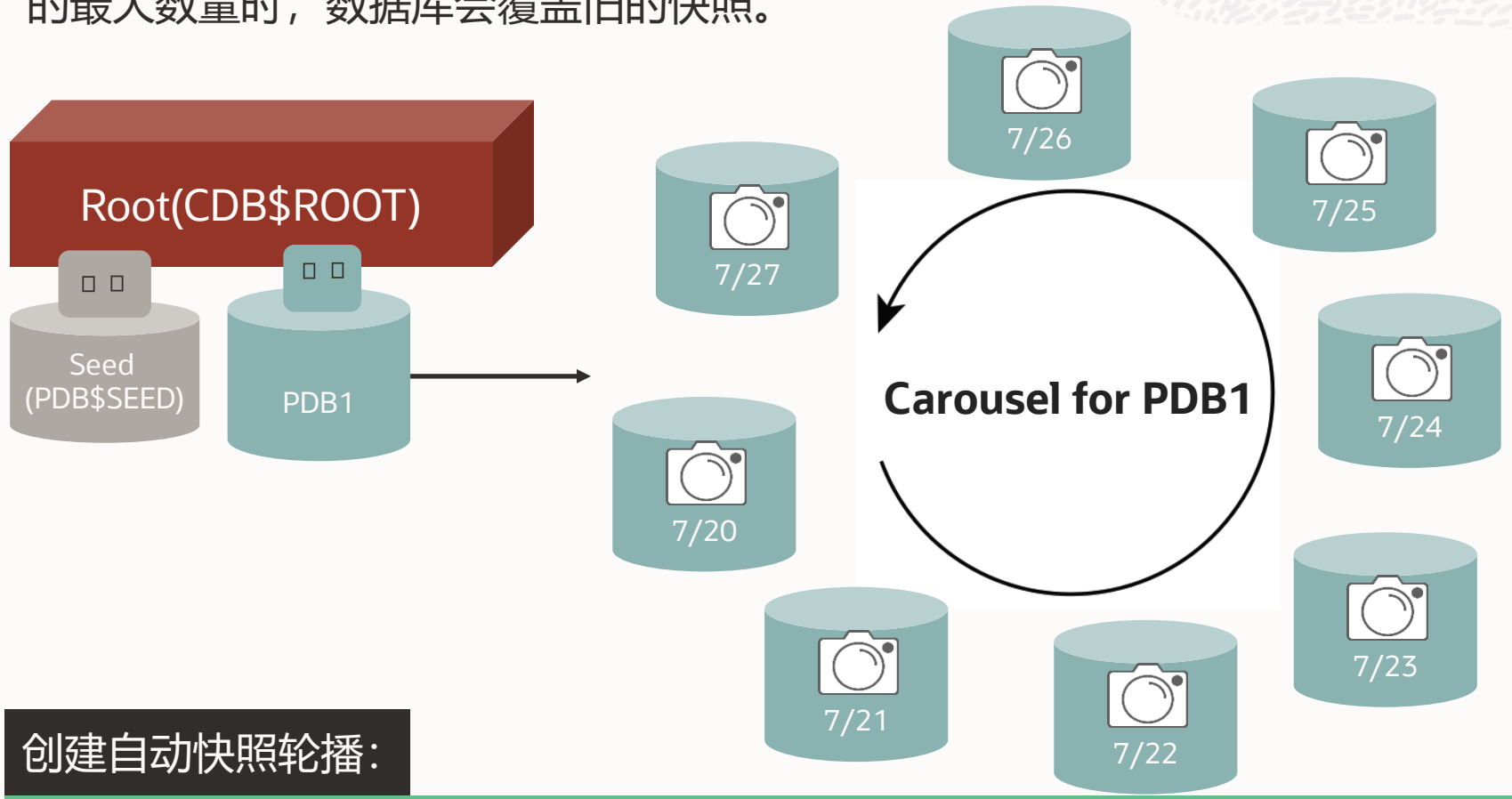
免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

3. PDB快照

3.2 快照轮播（Snapshot Carousel）：

PDB快照轮播是该PDB副本的循环库，数据库根据需要或自动在轮播中创建连续副本，当达到快照限制的最大数量时，数据库会覆盖旧的快照。



- ✓ PDB快照轮播是PDB所有当前快照的集合。
- ✓ CDB_PROPERTIES视图中MAX_PDB_SNAPSHOTS属性指定快照的最大数量，默认为8，可以修改。
- ✓ 左图中pdb1每天自动获取一个PDB快照，最多保持8个。创建前8个快照后，每个新创建快照都会覆盖最旧的快照，如7月28日快照会覆盖7月20日快照，7月29日快照会覆盖7月21日快照。

创建自动快照轮播：

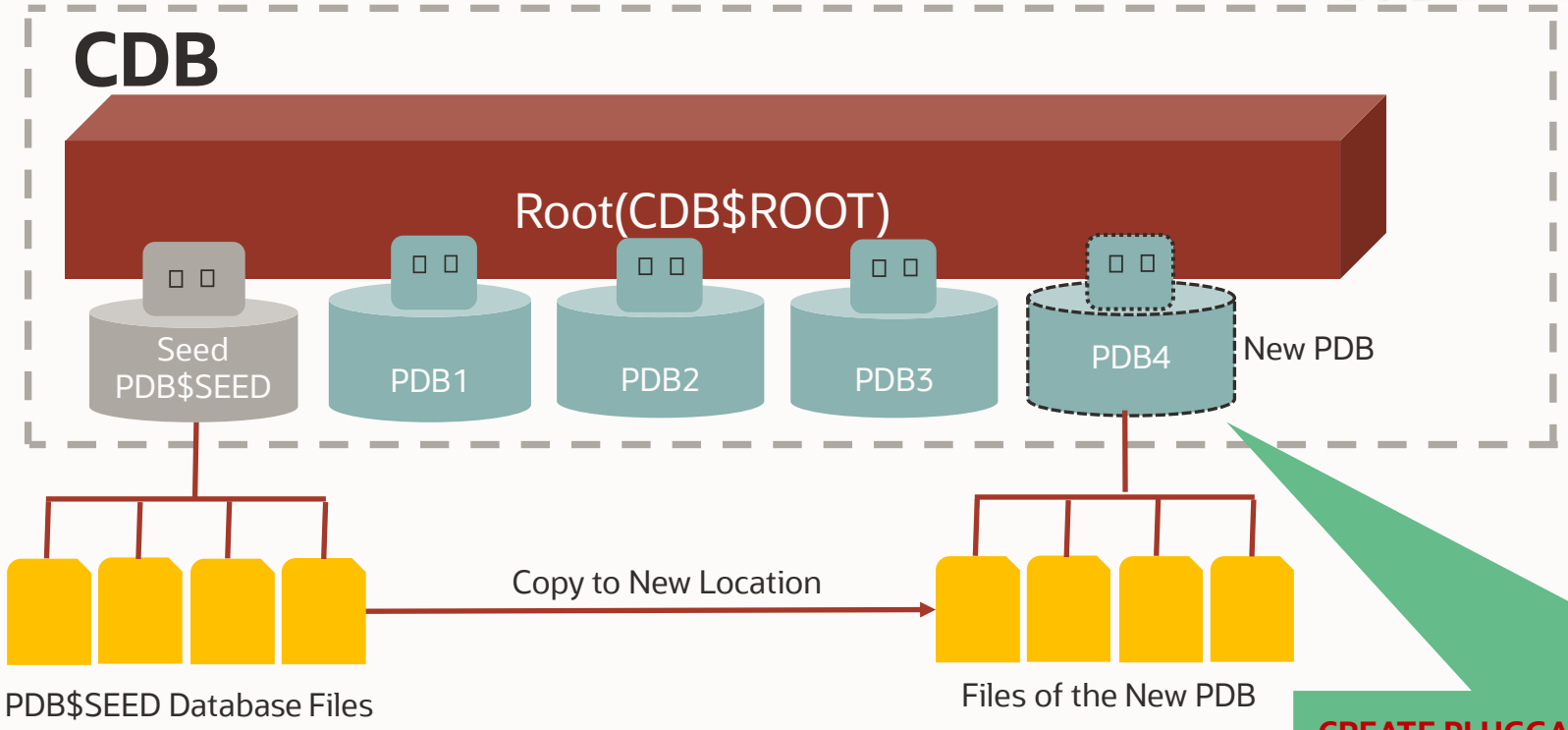
```
SQL> ALTER SESSION SET CONTAINER=pdb1;
SQL> ALTER PLUGGABLE DATABASE pdb1 SNAPSHOT MODE EVERY 24 HOURS;
SQL> SELECT pdb_name,snapshot_mode,snapshot_interval FROM cdb_pdbs where pdb_name='PDB1';
```

 Copyright © 2023, Oracle and/or its affiliates

免费咨询热线：400-699-8888

4. PDB克隆:

4.1 本地克隆（使用seed）：



- ✓ Seed是用来快速创建另一个PDB的模板。
- ✓ 从Seed创建PDB复制Seed PDB的部分或全部内容，然后分配新的唯一标识符。
- ✓ Seed分类：
  - PDB seed (PDB$SEED):
    由系统提供，不能删除和修改；
  - Application seed
    用户为Application root创建的；

CREATE PLUGGABLE DATABASE ... ADMIN USER ... IDENTIFIED BY...

使用seed创建PDB:

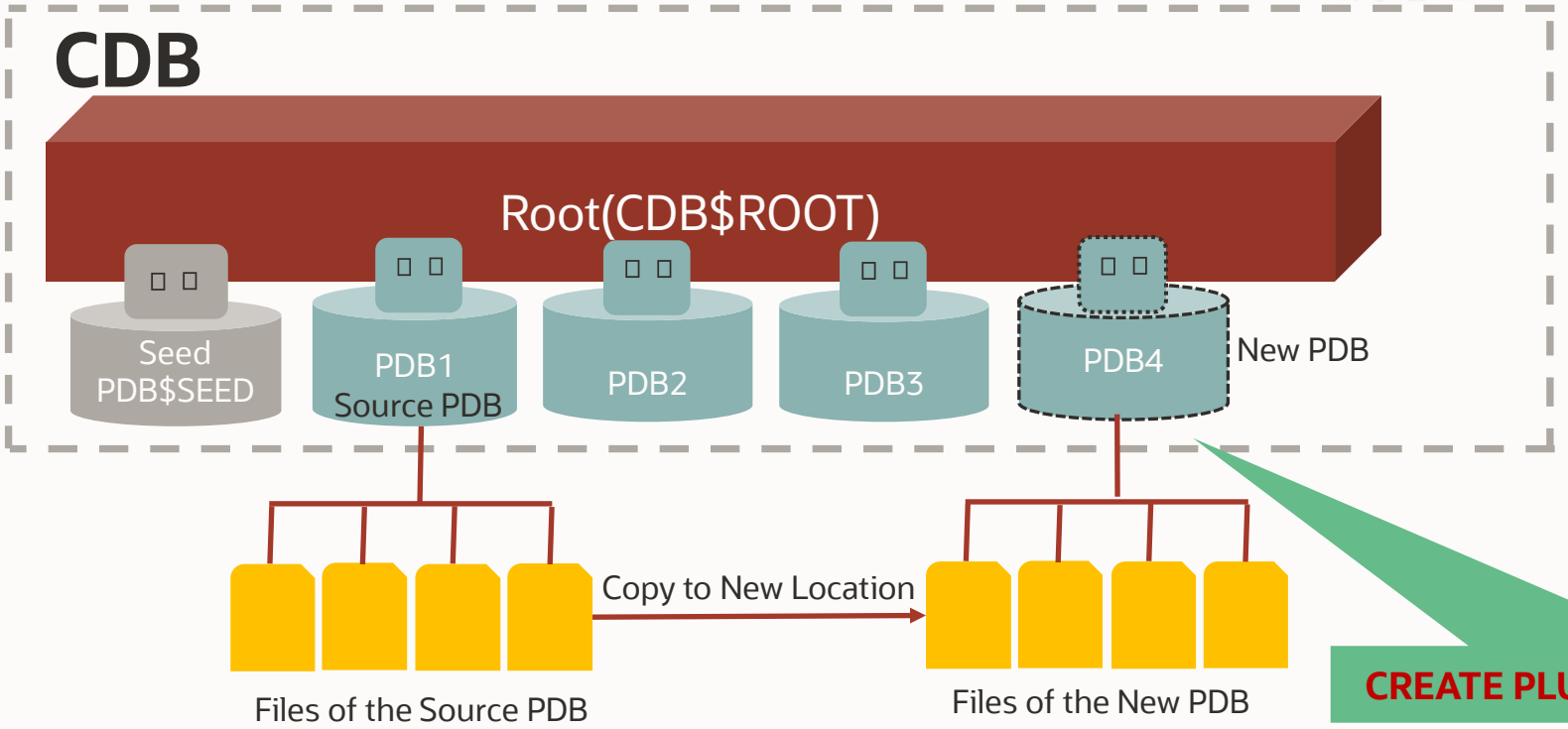
```
SQL> CREATE PLUGGABLE DATABASE pdb4 ADMIN USER pdbadm1 IDENTIFIED BY welcome1;
SQL> show pdbdetails
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

4. PDB克隆:

4.2 本地克隆（使用现有PDB）：



- ✓ 当前用户必须具有CREATE PLUGGABLE DATABASE权限。
- ✓ 源PDB不能关闭。
- ✓ 如果CDB没有采用本地UNDO模式，则源PDB必须使用只读方式打开。
- ✓ 如果CDB没有开启归档模式，则源PDB必须使用只读方式打开。
- ✓ 如果要创建应用PDB，则应用PDB必须与应用容器使用相同的数据库字符集和国家字符集。

CREATE PLUGGABLE DATABASE ... From ...

v$session_longops监控进度
v$px_session监控并行进程

使用源库创建PDB:

```
SQL> CREATE PLUGGABLE DATABASE pdb4 FROM pdb1 [parallel 8];
SQL> show pdbdetails
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

4. PDB克隆:

4.3 本地克隆（使用快照）：

第一步，创建PDB快照:

```
SQL> alter session set container=PDB1;
SQL> alter pluggable database snapshot pdb_snap1;
```

第二步，使用指定快照创建PDB:

```
SQL> CREATE PLUGGABLE DATABASE pdb1_copy FROM PDB1 USING SNAPSHOT pdb_snap1;
```

Snapshot copy PDB是一个基于底层存储系统快照的PDB。
Snapshot copy PDB减少了测试所需的存储量，并显著减少了创建时间（只复制数据文件header）。

创建一个Snapshot copy PDB :

```
SQL> CREATE PLUGGABLE DATABASE pdb1_snap_copy FROM PDB1 SNAPSHOT COPY;
```

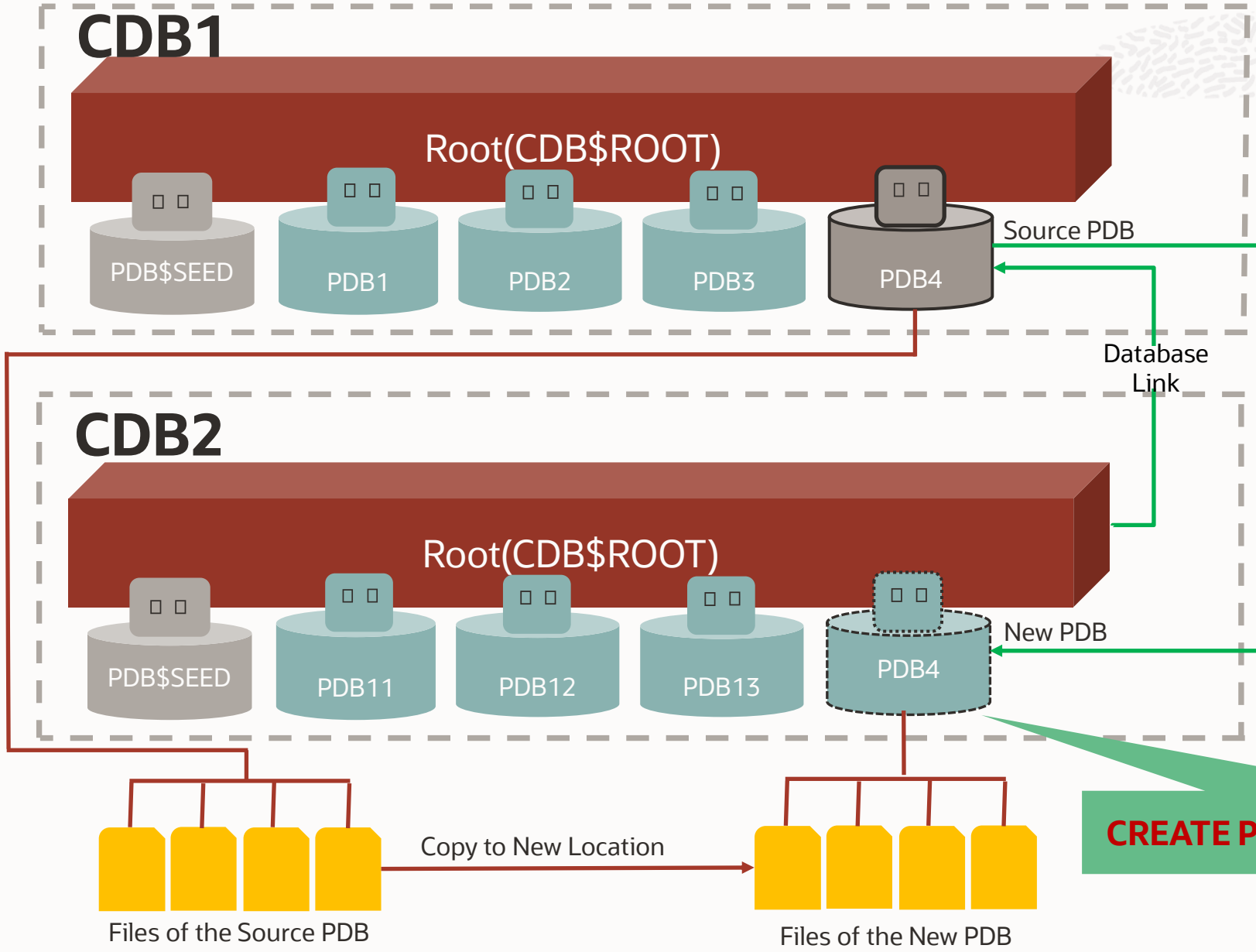
注意： USING SNAPSHOT与SNAPSHOT COPY子句创建的PDB具有不同的属性，
不能在单个CREATE PLUGABLE DATABASE命令中同时指定这两个子句。
USING SNAPSHOT子句创建一个完整的、独立的、不需要物化（materialized）的PDB。
SNAPSHOT COPY子句创建一个稀疏PDB，如果要删除它所基于的存储级快照，则必须进行物化。
对Snapshot copy PDB进行物化操作：SQL> ALTER PLUGGABLE DATABASE MATERIALIZE;

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

4. PDB克隆

4.4.1 远程克隆（源库是PDB）：



- ✓ 当前用户在目标CDB root中必须具有CREATE PLUGGABLE DATABASE权限。
- ✓ 源平台和目标平台必须满足以下要求：
  - 它们必须具有相同的字节序。
  - 源平台上安装的数据库选件必须与目标平台上安装的数据库选件相同或其子集。
- ✓ 如果要创建应用程序PDB，则源PDB的应用程序名称和版本必须与目标应用容器中应用程序的名称和版本相匹配。

CREATE PLUGGABLE DATABASE ... FROM ...

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

4. PDB克隆

4.4.1 远程克隆（源库是PDB）：

远程克隆PDB使用SQL语句（克隆CDB1中的pdb4到CDB2）：

第一步，CDB1中创建用于dblink的公共用户：

```
SQL>GRANT CREATE SESSION,CREATE PLUGGABLE DATABASE,SYSOPER TO c##clopdb IDENTIFIED BY
welcome1 CONTAINER=ALL;
```

第二步，登录CDB2 root或application root：

```
# su – oracle
$ export ORACLE_SID=CDB2
$ sqlplus / as sysdba
```

第三步，CDB2中创建到远程CDB1的database link：

```
SQL> CREATE DATABASE LINK pdb4_link CONNECT TO c##clopdb IDENTIFIED BY welcome1 USING
'HOST:PORT/pdb4';
```

第四步，CDB2中执行克隆操作：

克隆新库处于mounted状态

```
SQL> CREATE PLUGGABLE DATABASE pdb4 FROM pdb4@pdb4_link [parallel 4];
```

第五步，CDB2中以读写方式打开迁移PDB:

```
SQL> ALTER PLUGGABLE DATABASE pdb4 open [read write];
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

4. PDB克隆

4.4.1 远程克隆（源库是PDB）：

远程克隆PDB使用DBCA命令行工具:

第一步，CDB1中创建用于dblink的公共用户：

```
SQL>GRANT CREATE SESSION,CREATE PLUGGABLE DATABASE,SYSOPER TO c##clopdb IDENTIFIED BY
welcome1 CONTAINER=ALL;
```

第二步，在CDB2主机上执行，克隆CDB1中pdb4到CDB2：

```
$ dbca –silent \
-createPluggableDatabase \
-createFromRemotePDB \
-sourceDB cdb2 \
-remotePDBName pdb4 \
-remoteDBConnString cdb1_host:1521/cdb1_inst \
-remoteDBSYSDBAUserName cdb1_SYS \
        -remoteDBSYSDBAUserPassword cdb1_syspwd \
-dbLinkUsername c##clopdb \
       -dbLinkUserPassword welcome1 \
-sysDBAUserName cdb2_SYS \
       -sysDBAPassword cdb2_syspwd \
-pdbName clone_pdb4
```

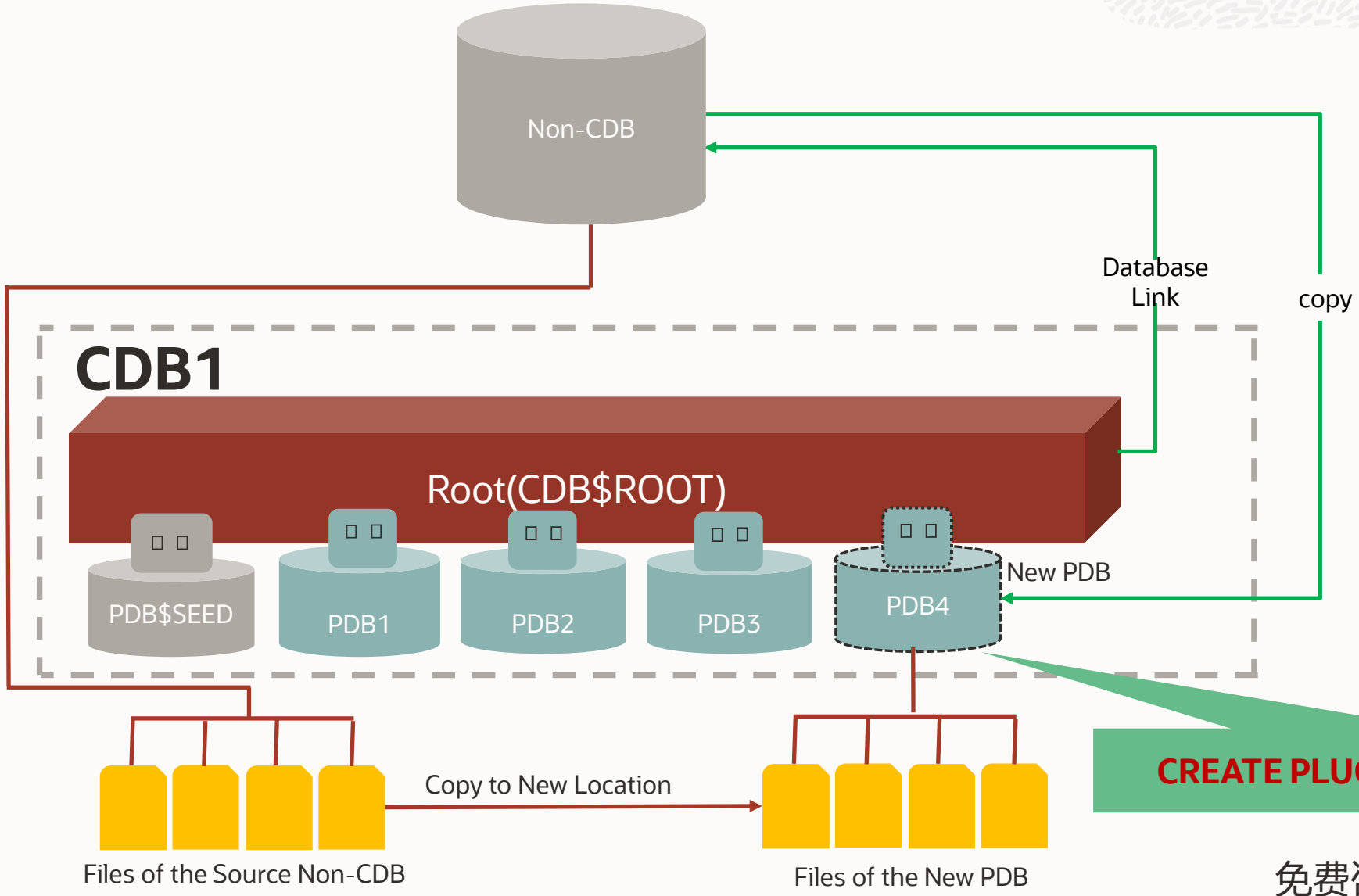
克隆新库处于READ WRITE状态

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

4. PDB克隆

4.4.2 远程克隆（源库non-CDB）：



- ✓ CDB与non-CDB必须运行Oracle Database12.1.0.2或更高版本。
- ✓ CDB与non-CDB必须运行相同的Oracle数据库版本。
- ✓ 新创建PDB的数据块大小必须与CDB匹配。
- ✓ 如果non-CDB处于非归档模式，则必须以只读方式打开。
- ✓ 如果non-CDB处于归档模式，则可以以只读或读/写方式打开。

CREATE PLUGGABLE DATABASE ... FROM ...

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

4. PDB克隆

4.4.2 远程克隆（源库non-CDB）：

第一步，登录CDB1 root或application root：

```
# su – oracle
$ export ORACLE_SID=CDB1
$ sqlplus / as sysdba
```

第二步，CDB1中创建到远程Non-CDB的database link：

```
SQL> CREATE DATABASE LINK ncdb1_link CONNECT TO system IDENTIFIED BY welcome1 USING
'HOST:PORT/ncdb1';
```

第三步，CDB1中执行远程克隆操作：

```
SQL> CREATE PLUGGABLE DATABASE pdb4 FROM ncdb1@ncdb1_link;
或：
SQL> CREATE PLUGGABLE DATABASE pdb4 FROM NON$CDB@ncdb1_link;
```

第四步，CDB1中运行noncdb_to_pdb.sql脚本：

```
在首次打开数据库之前需要执行noncdb_to_pdb.sql脚本：
SQL> @$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```

第五步，CDB1打开数据库到读写状态：

```
SQL>ALTER PLUGGABLE DATABASE ncdb open [read write];
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

5. 插拔PDB:

5.1 拔出PDB:

第一步，登录PDB所在的root容器：

```
# su – oracle
$ export ORACLE_SID=CDB1
$ sqlplus / as sysdba
```

第二步，关闭PDB：

```
SQL> ALTER PLUGGABLE DATABASE salespdb close immediate [instances=all];
```

第三步，拔出PDB：

方式一，拔出到XML Metadata File （此文件描述pdb及关联的files）：

```
SQL> ALTER PLUGGABLE DATABASE salespdb UNPLUG INTO '/oracle/data/salespdb.xml';
```

方式二，拔出到Archive File （包含XML Metadata File 和PDB files）：

```
SQL> ALTER PLUGGABLE DATABASE salespdb UNPLUG INTO '/oracle/data/sales.pdb';
```

注意：PDB拔出后，仍然存在CDB中，使用cdb_pdbs视图查看拔出pdb的status为UNPLUGGED状态，要完全从CDB中删除PDB，则需要使用drop命令，而drop也是拔出pdb上唯一支持的操作，drop pdb命令：SQL> DROP PLUGGABLE DATABASE *salespdb* [KEEP DATAFILES]|INCLUDING DATAFILES;

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

5. 插拔PDB:

5.2 插入已拔出PDB:

第一步，登录要插入pdb的root容器：

```
# su – oracle
$ export ORACLE_SID=CDB2
$ sqlplus / as sysdba
```

第二步，检查兼容性：

```
SQL>SET SERVEROUTPUT ON
        DECLARE
          compatible CONSTANT VARCHAR2(3) :=
            CASE DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
              pdb_descr_file => '/disk1/usr/salespdb.xml'，或'/oracle/data/sales.pdb',
              pdb_name => 'SALESPDB')
             WHEN TRUE THEN 'YES'
             ELSE 'NO'
        END;
        BEGIN
          DBMS_OUTPUT.PUT_LINE(compatible);
        END;
         /
```

如果输出为YES，则PDB兼容，您可以继续下一步。
如果输出为NO，则 PDB 不兼容，检查PDB_PLUG_IN_VIOLATIONS视图以了解不兼容的原因

第三步，插入（创建）PDB：

在原CDB上执行插入需要先drop原pdb

```
方式一，使用XML Metadata File （此文件描述pdb及关联的files）：
SQL> CREATE PLUGGABLE DATABASE salespdb USING '/disk1/usr/salespdb.xml' NOCOPY TEMPFILE REUSE;

方式二，使用Archive File （包含XML Metadata File 和PDB files）：
SQL> CREATE PLUGGABLE DATABASE salespdb USING '/oracle/data/sales.pdb' STORAGE (MAXSIZE 100G);
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

5. 插拔PDB:

5.3 插入Non-CDB:

第一步，确保non-CDB处于事务一致性状态并启动到只读模式：

```
SQL>select name,open_mode from v$database;
       NAME      OPEN_MODE
       ---------  ----------------------------------
       NCDB       READ ONLY
```

第二步，创建XML file：

```
SQL>BEGIN DBMS_PDB.DESCRIBE(
       pdb_descr_file => '/disk1/oracle/ncdb.xml');
       END;
       /
```

第三步，检查兼容性：

```
SQL>SET SERVEROUTPUT ON
       DECLARE
          compatible CONSTANT VARCHAR2(3) :=
             CASE DBMS_PDB.CHECK_PLUG_COMPATIBILITY(
                pdb_descr_file => '/disk1/usr/salespdb.xml',
                pdb_name => 'SALESPDB')
              WHEN TRUE THEN 'YES'
              ELSE 'NO'
        END;
        BEGIN
           DBMS_OUTPUT.PUT_LINE(compatible);
        END;
         /
```

如果输出为YES，则PDB兼容，您可以继续下一步。
如果输出为NO，则 PDB 不兼容，检查PDB_PLUG_IN_VIOLATIONS视图以了解不兼容的原因

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

5. 插拔PDB:

5.3 插入Non-CDB:

第四步，关闭non-CDB：

```
SQL> shutdown immediate;
```

第五步，利用XML file创建PDB：

```
SQL> CREATE PLUGGABLE DATABASE ncdb USING '/disk1/oracle/ncdb.xml'
        COPY
        FILE_NAME_CONVERT = ('/disk1/oracle/dbs/', '/disk2/oracle/ncdb/');
```

第六步，运行noncdb_to_pdb.sql脚本：

在首次打开数据库之前需要执行noncdb_to_pdb.sql脚本：

```
SQL> @$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql
```

第七步，打开数据库到读写状态：

```
SQL>ALTER PLUGGABLE DATABASE ncdb open [read write];
```

第八步，备份数据库：

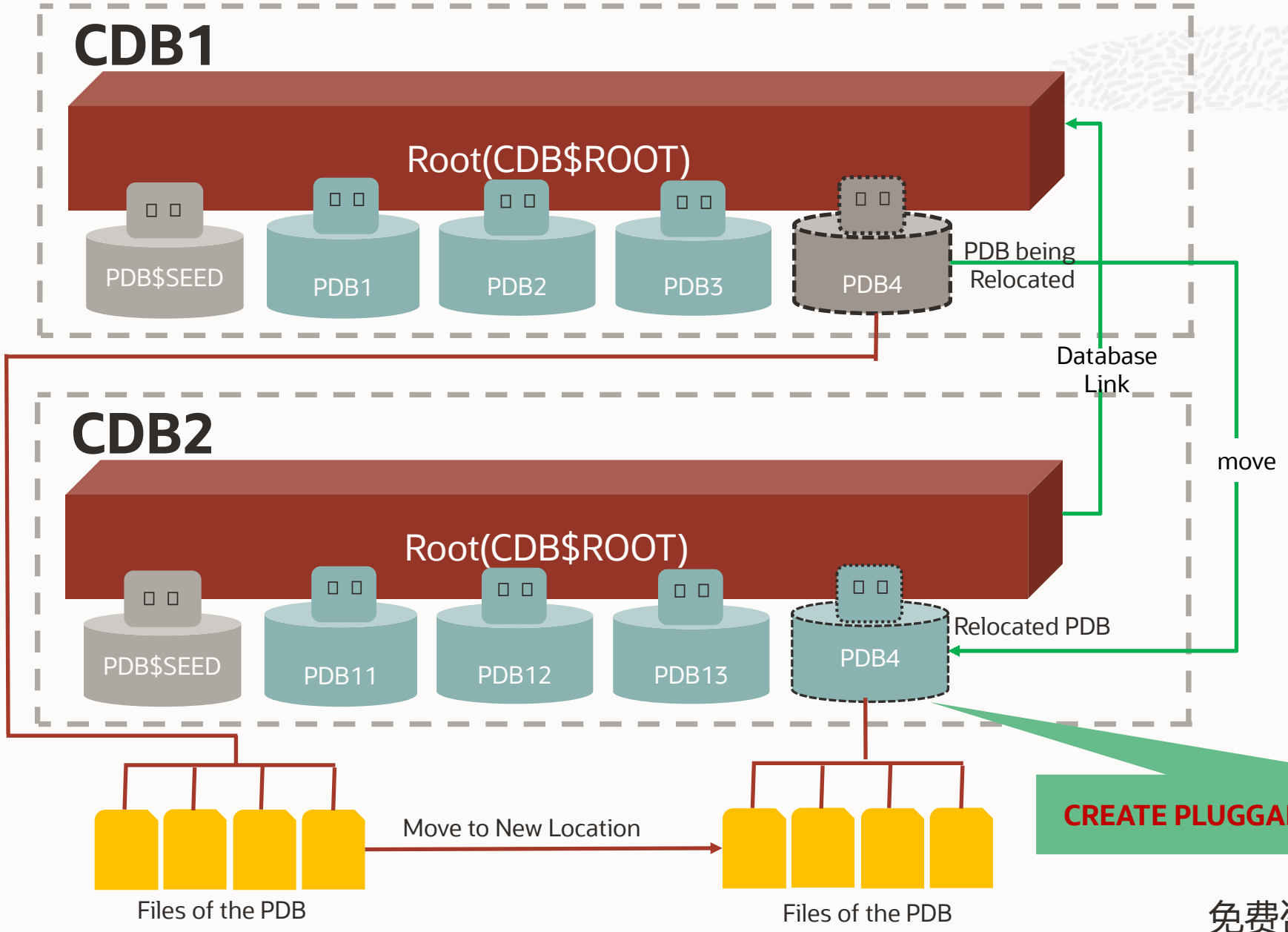
```
RMAN> BACKUP PLUGGABLE DATABASE  ncdb;
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

6. 迁移PDB

6.1 迁移PDB(Relocate PDB):



- ✓ 您可以将PDB迁移到不同的CDB或应用程序容器。
- ✓ 在迁移期间，源PDB可以在读/写模式下打开，并且完全可用。
- ✓ 当源PDB打开并有活动会话时，PDB迁移时在线执行源PDB数据文件、REDO、UNDO块级别复制。当目标PDB使用ALTER PLUGABLE DATABASE OPEN打开时，Oracle DATABASE将终止活动会话并关闭源PDB。
- ✓ 这项技术是迁移PDB的最快方法，只需最少或没有停机时间。而拔出源PDB则需要中断PDB，直到PDB插入目标CDB为止。

CREATE PLUGGABLE DATABASE ... FROM ... RELOCATE

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

6. 迁移PDB:

6.2 迁移PDB使用SQL语句（CDB1中pdb4迁移到CDB2中）：

第一步，CDB1中创建用于dblink的公共用户：

```
SQL>GRANT CREATE SESSION,CREATE PLUGGABLE DATABASE,SYSOPER TO c##mvpdb IDENTIFIED BY
welcome1 CONTAINER=ALL;
```

第二 步，登录CDB2 root或application root：

```
# su – oracle
$ export ORACLE_SID=CDB2
$ sqlplus / as sysdba
```

第三步，CDB2创建到远程CDB1的database link：

```
SQL> CREATE DATABASE LINK pdb4_link CONNECT TO c##mvpdb IDENTIFIED BY welcome1 USING
'HOST:PORT/CDB1';
```

第四步，CDB2中执行迁移操作：

```
SQL> CREATE PLUGGABLE DATABASE pdb4 FROM pdb4@pdb4_link RELOCATE AVAILABILITY [MAX|NORMAL];
```

第五步，CDB2中以读写方式打开迁移PDB：

```
SQL> ALTER PLUGGABLE DATABASE pdb4 open [read write];
```

open之后pdb4就不会存在于CDB1中

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

6. 迁移PDB:

6.3 迁移PDB使用DBCA命令行工具:

第一步，CDB1中创建用于dblink的公共用户：

```
SQL>GRANT CREATE SESSION,CREATE PLUGGABLE DATABASE,SYSOPER TO c##mvpdb IDENTIFIED BY
welcome1 CONTAINER=ALL;
```

第二步，在CDB2主机上执行，迁移CDB1中pdb4到CDB2：

```
$ dbca -silent \
-relocatePDB \
-sourceDB cdb2 \
-remotePDBName pdb4 \
-remoteDBConnString cdb1_host:1521/cdb1  \
-remoteDBSYSDBAUserName cdb1_SYS \
        -remoteDBSYSDBAUserPassword cdb1_syspwd \
-dbLinkUsername c##mvpdb \
        -dbLinkUserPassword welcome1 \
-sysDBAUserName cdb2_SYS \
        -sysDBAPassword cdb2_syspwd \
-pdbName pdb4
```

命令执行完成pdb4在CDB2中处于READ WRITE状态

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

7. 可刷新PDB:

7.1 可刷新PDB介绍:

- ✓ 生产PDB的克隆操作可能需要相当长的时间。为了尽量减少对生产系统库的影响，克隆操作不能频繁执行，这样克隆的数据就会过时。一个可刷新的克隆PDB就可以解决这个问题。
- ✓ 当一个可刷新的克隆PDB过时后，您可以关闭它，然后用最近的REDO来刷新它。不刷新时，可刷新的克隆PDB能够以只读方式打开。
- ✓ 一种典型的做法是维护生产PDB的“golden master”可刷新克隆，获取PDB级别的快照，然后从PDB快照创建克隆以进行开发和测试，还可以利用可刷新PDB来迁移生产库。
- ✓ 可以使用ALTER PLUGABLE DATABASE … SWITCH OVER语句切换源和克隆PDB的角色，此功能在以下情况下很有用:
  - 计划内切换:

    源PDB所在的CDB可能比克隆PDB所在的CDB拥有更多的开销时，要实现负载均衡，可以切换角色，使克隆PDB成为新的源PDB，使源PDB成为新的克隆PDB。
  - 意外切换:

    源PDB可能会发生意外故障，在这种情况下，可以使克隆PDB成为新的源PDB，并恢复正常工作。

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

7. 可刷新PDB:

7.2 刷新模式:

通过REFRESH MODE来配置克隆PDB的刷新模式:

✓ REFRESH MODE NONE：默认值，创建不可刷新的PDB。
通过在ALTER PLUGABLE DATABASE语句中包含REFRESH MODE NONE子句，然后以读/写模式打开PDB，可以将可刷新的克隆PDB更改为普通PDB。您不能将普通PDB更改为可刷新的克隆PDB。可刷新的克隆PDB转换为普通PDB后，无法将其更改回可刷新PDB。

✓ REFRESH MODE MANUAL：创建必须手动刷新的可刷新PDB，刷新时要求关闭PDB。

✓ REFRESH MODE EVERY number_of_minute MINUTES：创建可自动刷新的PDB，该PDB在指定分钟后会自动刷新，使用自动刷新的可刷新PDB也可以进行手动刷新，刷新时要求关闭PDB。

创建可自动刷新PDB:

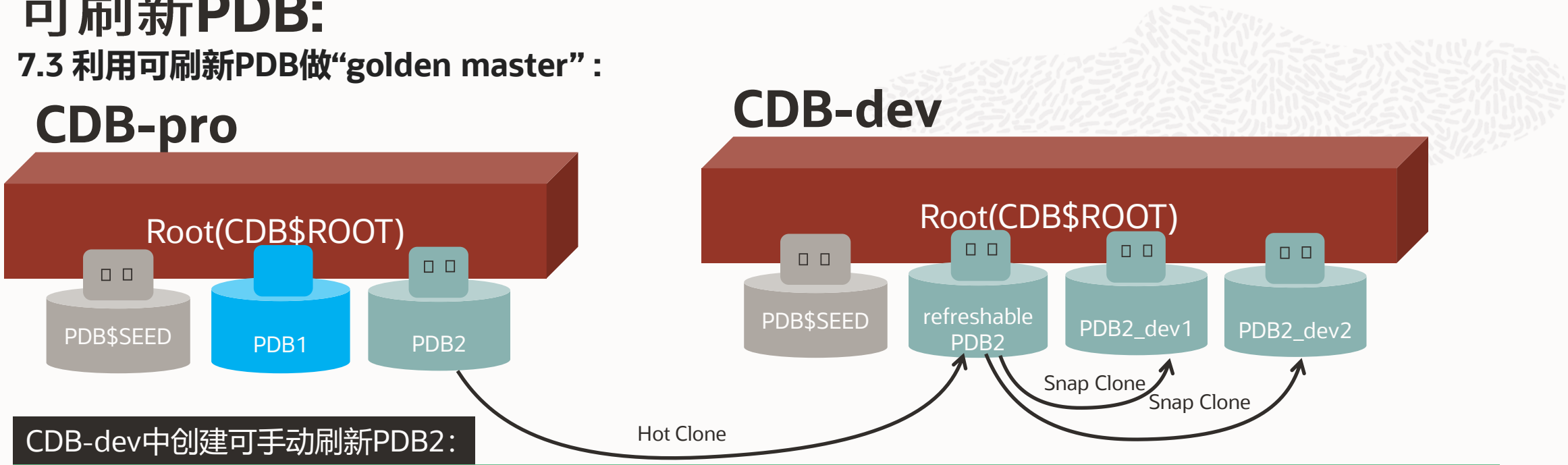
```
SQL> CREATE PLUGGABLE DATABASE pdb4 FROM pdb4@pdb4_link REFRESH MODE EVERY 60 MINUTES;
SQL> SELECT pdb_name,refresh_mode,refresh_interval FROM cdb_pdbs WHERE pdb_name='PDB4';
```

 Copyright © 2023, Oracle and/or its affiliates

免费咨询热线：400-699-8888

7. 可刷新PDB:

7.3 利用可刷新PDB做“golden master”：



CDB-dev中创建可手动刷新PDB2：

```
SQL> CREATE PLUGGABLE DATABASE pdb2 FROM pdb2@pdb2_link REFRESH MANUAL;
```

CDB-dev中创建快照并克隆新库:

```
SQL> alter session set container=PDB2;
SQL> alter pluggable database snapshot pdb2_snap1;
SQL> CREATE PLUGGABLE DATABASE pdb2_dev1 FROM PDB2 USING SNAPSHOT pdb2_snap1;
SQL> CREATE PLUGGABLE DATABASE pdb2_dev2 FROM PDB2 USING SNAPSHOT pdb2_snap1;
```

CDB-dev中创建两个Snapshot copy PDB：

```
SQL> CREATE PLUGGABLE DATABASE pdb2_dev1 FROM PDB2 SNAPSHOT COPY;
SQL> CREATE PLUGGABLE DATABASE pdb2_dev2 FROM PDB2 SNAPSHOT COPY;
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

7. 可刷新PDB:

7.4 利用可刷新PDB迁移数据库(CDB1中源pdb4迁移到目标CDB2中):

第一步，CDB1中创建用于dblink的公共用户：

```
SQL>GRANT CREATE SESSION,CREATE PLUGGABLE DATABASE,SYSOPER TO c##refpdb IDENTIFIED BY welcome1 CONTAINER=ALL;
```

第二 步，登录目标CDB2 root:

```
# su – oracle
$ export ORACLE_SID=CDB2
$ sqlplus / as sysdba
```

第三步，CDB2中创建到远程CDB1的database link:

```
SQL> CREATE DATABASE LINK pdb4_link CONNECT TO c##refpdb IDENTIFIED BY welcome1 USING 'HOST:PORT/CDB1';
```

第四步，CDB2中创建可刷新PDB：

```
SQL> CREATE PLUGGABLE DATABASE pdb4 FROM pdb4@pdb4_link PARALLEL 8  REFRESH MODE MANUAL;
```

第五步，CDB2中多次手动刷新或修改为自动刷新：

```
SQL>ALTER PLUGGABLE DATABASE pdb4 REFRESH;
或：SQL>ALTER PLUGGABLE DATABASE pdb4 REFRESH MODE EVERY 60 MINUTES;
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

7. 可刷新PDB:

7.4 利用可刷新PDB迁移数据库(CDB1中源pdb4迁移到目标CDB2中):

第六步，登录源库CDB1 root，关闭源PDB：

```
SQL>ALTER PLUGGABLE DATABASE pdb4 close [instances=all];
```

第七步，登录目标CDB2 root，执行最后一次手动同步：

```
SQL>ALTER PLUGGABLE DATABASE pdb4 REFRESH;
检查alert日志，查看介质恢复情况，恢复到最后的SCN号。
检查v$datafile_header视图中checkpoint_change#是否一致（正常都是alert日志中恢复到的最后的SCN号）：
SQL>select con_id,file#,checkpoint_change# from v$datafile_header where con_id=6;
```

第八步，停止refresh，与源库脱离关系:

```
SQL>ALTER PLUGGABLE DATABASE pdb4 REFRESH MODE NONE;
检查新pdb状态：
SQL>select pdb_name,status,refresh_mode,refresh_interval from dba_pdbs;
```

STATUS： NEW

第九步，把新PDB以读写方式打开::

```
SQL>ALTER PLUGGABLE DATABASE pdb4 open [instances=all];
检查新pdb状态：
SQL>select pdb_name,status,refresh_mode,refresh_interval from dba_pdbs;
```

STATUS： NORMAL且不可刷新

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

7. 可刷新PDB:

7.5 可刷新PDB角色切换(CDB1中源库pdb4，CDB2中可刷新克隆库pdb4_ref):

第一步，在CDB1和CDB2中创建相同的公共用户及密码：

```
SQL> CREATE USER c##u1 IDENTIFIED BY welcome1;
SQL> GRANT CREATE SESSION, RESOURCE, CREATE ANY TABLE, UNLIMITED TABLESPACE TO c##u1 CONTAINER=ALL;
SQL> GRANT CREATE PLUGGABLE DATABASE TO c##u1 CONTAINER=ALL;
SQL> GRANT SYSOPER TO c##u1 CONTAINER=ALL;
```

第二步，在CDB2中创建database link：

```
SQL> CREATE DATABASE LINK cdb1_datalink CONNECT TO c##u1 IDENTIFIED BY welcome1 USING 'HOST:PORT/CDB1';
```

第三步，在CDB2中创建可刷新PDB：

```
SQL> CREATE PLUGGABLE DATABASE pdb4_ref FROM pdb4@cdb1_datalink REFRESH MODE MANUAL;
```

第四步，在CDB2中手动刷新pdb4_ref：

```
SQL> ALTER PLUGGABLE DATABASE pdb4_ref REFRESH;
```

第五步，在CDB2中启动pdb4_ref到只读状态：

```
SQL> ALTER PLUGGABLE DATABASE pdb4_ref OPEN READ ONLY;
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

7. 可刷新PDB:

7.5 可刷新PDB角色切换(CDB1中源库pdb4，CDB2中可刷新克隆库pdb4_ref):

第六步，在CDB1中创建database link：

```
SQL> CREATE DATABASE LINK cdb2_datalink CONNECT TO c##u1 IDENTIFIED BY welcome1 USING  'HOST:PORT/CDB2';
```

第七步，在CDB1中执行切换：

```
SQL> ALTER PLUGGABLE DATABASE pdb4 REFRESH MODE MANUAL FROM pdb4_ref@cdb2_datalink SWITCHOVER;
```

第八步，在CDB1中检查pdb4状态：

```
SQL> SELECT PDB_NAME,STATUS,REFRESH_MODE,REFRESH_INTERVAL FROM DBA_PDBS;
PDB_NAME           STATUS            REFRESH_MODE       REFRESH_INTERVAL
------------------ -------------     ---------- ---------       --------------------
PDB4               REFRESHING        MANUAL
...
```

第九步，在CDB2中启动pdb4_ref状态：

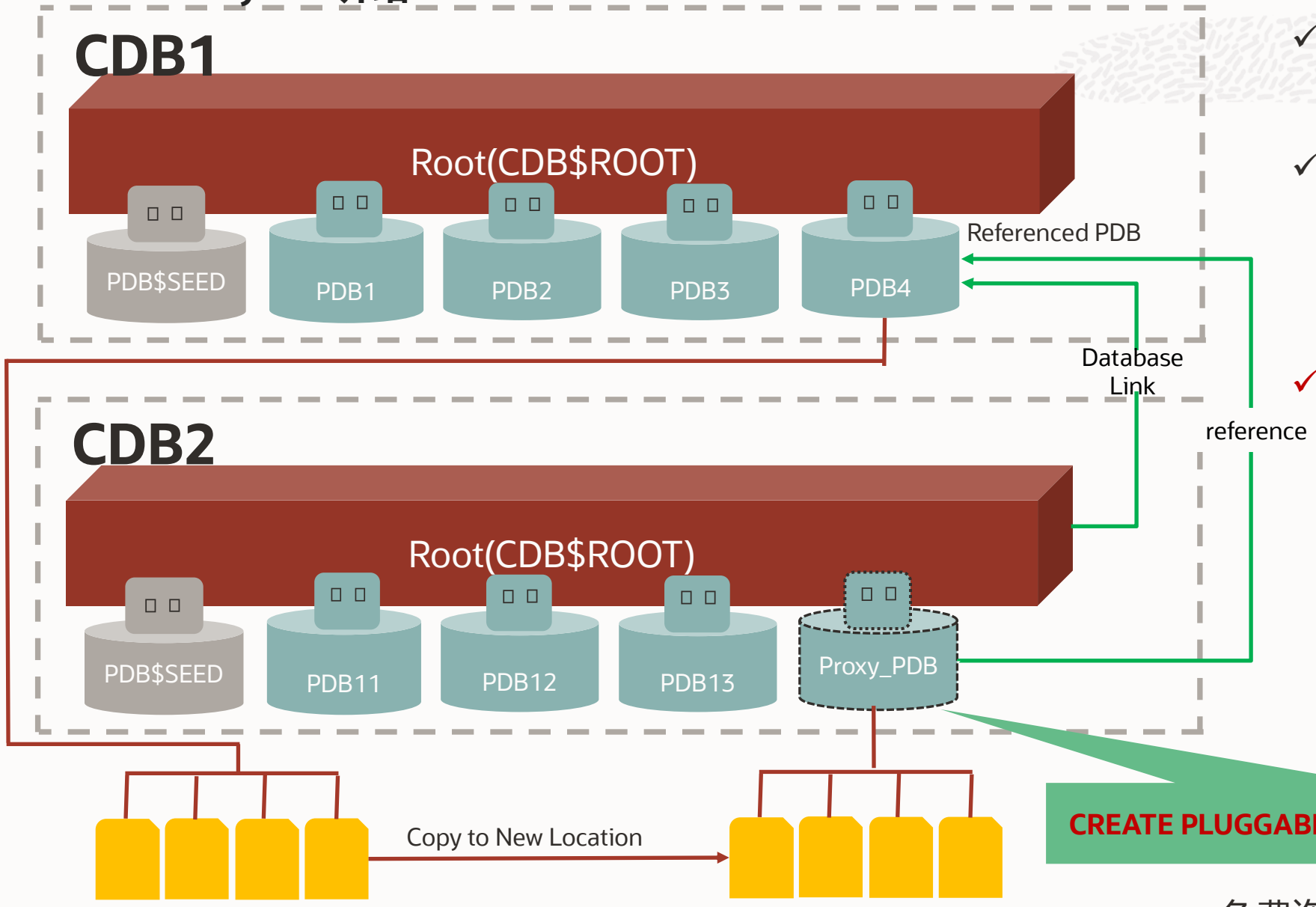
```
SQL> show pdbs
CON_ID  CON_NAME           OPEN MODE     RESTRICTED
--------  ------------------     --------------   --------------
 6         PDB4_ref             READ WRITE     NO
...
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

8. Proxy PDB

8.1 Proxy PDB介绍：



- ✓ Proxy PDB提供对远程CDB中PDB的访问。它类似于一个符号链接。
- ✓ 当Proxy PDB是当前容器时，在Proxy PDB中执行的SQL语句会在Referenced PDB中被执行。远程执行的结果被返回给Proxy PDB。
- ✓ 例外情况：当Proxy PDB 是当前容器时，执行 ALTER PLUGGABLE DATABASE 和 ALTER DATABASE 语句时，这些语句仅影响Proxy PDB，它们不会发送到Referenced PDB执行。 同样，当前容器是 Proxy PDB 所属的root时，ALTER PLUGGABLE DATABASE 语句仅影响Proxy PDB。

CREATE PLUGGABLE DATABASE ... AS PROXY FROM ...

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

8. Porxy PDB:

8.2 创建Proxy PDB:

第一步，登录CDB1中创建公共用户

```
SQL> GRANT CREATE SESSION,CREATE PLUGGABLE DATABASE,SYSOPER TO C##PROXY IDENTIFIED BY Welcome1 CONTAINER=ALL;
```

第二步，登录CDB2 root或application root：

```
# su - oracle
$ export ORACLE_SID=CDB2
$ sqlplus / as sysdba
```

第三步，创建到远程CDB1的database link:

```
SQL> CREATE DATABASE LINK pdb4_link CONNECT TO C##PROXY IDENTIFIED BY Welcome1 USING 'HOST:PORT/pdb4';
```

第四步，创建Proxy PDB:

```
SQL> CREATE PLUGGABLE DATABASE pdb4_proxy AS PROXY FROM pdb4@pdb4_link;
```

第五步，以读写方式打开Proxy PDB:

```
SQL> ALTER PLUGGABLE DATABASE pdb4_proxy open [read write];
```

说明：创建Proxy PDB时需要数据库链接。创建Proxy PDB后，Proxy PDB将不再使用创建时指定的数据库链接，而直接与Referenced PDB进行通信。

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

8. Porxy PDB:

8.3 Proxy PDB示例:

```
SQL> SELECT pdb_name, is_proxy_pdb FROM cdb_pdbs where pdb_name='PDB4_PROXY';
        PDB_NAME         IS_PROXY_PDB
        ---------------------   ---------------------
         PDB4_PROXY     YES

SQL> SELECT name, proxy_pdb FROM v$pdbs where name='PDB4_PROXY';
         PDB_NAME       IS_PROXY_PDB
         ----------------       ----------------------
         PDB4_PROXY    YES

SQL> show pdbdetails
                    APP                                                  APP
                   ROOT                          APP     APP    APP     ROOT      PXY
         CONID CONID CON_NAME                    ROOT    PDB    SEED    CLONE     PDB      OPEN MODE    REST
        --------- --------- -----------------     --------- ------- -------  ---------  ---------  -------------------  -------
           2              PDB$SEED          NO       NO     NO       NO          NO       READ ONLY     NO
           6              PDB4_PROXY        NO       NO     NO       NO          YES      READ WRITE    NO

SQL> SELECT con_id,target_port,target_host,target_service,target_user  FROM  v$proxy_pdb_targets;
           CON_ID TARGET_PORT    TARGET_HOST          TARGET_SERVICE                                    TARGET_USER
           ---------  ----------------------   -------------------------   ------------------------------------------------  --------------------
           6          1521                     x9mdbadm01          0159c0e192466ca3e063047f710a0993
```

YES表示此PDB是proxy pdb

引用数据库port

引用数据库host

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

8. Porxy PDB:

8.3 创建Proxy PDB示例:

登录**操作系统认证方式**切换到proxy pdb后检查数据文件:

```
$ export ORACLE_SID=cdb2
$ sqlplus / as sysdba
SQL> alter session set container=pdb4_proxy;

SQL> SELECT name FROM v$database;
        NAME
        ---------
        CDB2

SQL> show con_name
        CON_NAME
        ----------------
        PDB4_PROXY

SQL> select name from v$datafile union select name from v$tempfile;
        NAME
        --------------------------------------------------------------------------------
       +DATAC1/CDB2/015B282301852CFEE063047F710A4EF0/DATAFILE/sysaux.861.1143194201
       +DATAC1/CDB2/015B282301852CFEE063047F710A4EF0/DATAFILE/system.860.1143194201
       +DATAC1/CDB2/015B282301852CFEE063047F710A4EF0/DATAFILE/undotbs1.862.1143194201
       +DATAC1/CDB2/015B282301852CFEE063047F710A4EF0/TEMPFILE/temp.863.1143194209
```

当前数据库是proxy pdb所在的CDB

当前容器是proxy pdb

下面是proxy pdb的数据及临时文件

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

8. Porxy PDB:

8.3 Proxy PDB示例:

通过**服务方式**登录proxy pdb检查数据文件:

```
$ sqlplus system/"We1come$"@host:1521/cdb2
  SQL> alter session set container=pdb4_proxy;
或: $ sqlplus sys/“We1come$”@host:1521/pdb4_proxy as sysdba
SYS@host:1521/pdb4_proxy> show con_name
      CON_NAME
      ------------
      PDB4
SYS@host:1521/pdb4_proxy> SELECT name FROM v$database;
      NAME
     ---------
      CDB1
SYS@host:1521/pdb4_proxy> select name from v$datafile union select name from v$tempfile;
      NAME
      ------------------------------------------------------------------------
      +DATAC1/CDB1/0159C0E192466CA3E063047F710A0993/DATAFILE/sysaux.774.1143188355
      +DATAC1/CDB1/0159C0E192466CA3E063047F710A0993/DATAFILE/system.773.1143188355
      +DATAC1/CDB1/0159C0E192466CA3E063047F710A0993/DATAFILE/undotbs1.775.1143188355
      +DATAC1/CDB1/0159C0E192466CA3E063047F710A0993/DATAFILE/test_ts.866.1143196715
      +DATAC1/CDB1/0159C0E192466CA3E063047F710A0993/DATAFILE/ts1.858.1143188645
      +DATAC1/CDB1/0159C0E192466CA3E063047F710A0993/DATAFILE/ts2.859.1143188651
      +DATAC1/CDB1/0159C0E192466CA3E063047F710A0993/DATAFILE/users.772.1143188355
      +DATAC1/CDB1/0159C0E192466CA3E063047F710A0993/TEMPFILE/temp.778.1143188363
```

当前容器是引用PDB

当前数据库是引用数据库所在的CDB

下面是引用pdb的数据及临时文件

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

8. Porxy PDB:

8.3 Proxy PDB示例:

登录proxy pdb进行操作:

```
$ sqlplus sys/“We1come$”@host:1521/pdb4_proxy as sysdba
SYS@host:1521/pdb4_proxy> CREATE TABLESPACE test_ts2;

SYS@host:1521/pdb4_proxy> CREATE USER u2 IDENTIFIED BY welcome1
                          DEFAULT TABLESPACE test_ts2  QUOTA UNLIMITED ON test_ts2;
SYS@host:1521/pdb4_proxy> GRANT dba TO u2;

SYS@host:1521/pdb4_proxy> CONN u2/welcome1@host:1521/pdb4_proxy

U2@host:1521/pdb4_proxy> CREATE TABLE t1 (id int);
U2@host:1521/pdb4_proxy> INSERT INTO t1 VALUES (1);
U2@host:1521/pdb4_proxy> COMMIT;
```

通过服务方式登录Proxy PDB

对Proxy PDB执行DML操作，操作会发送到引用PDB中执行并返回结果

直接登录到引用PDB检查数据:

```
$ sqlplus sys/"We1come$"@host:1521/pdb4 as sysdba
   SYS@10.113.127.4:1521/pdb4>select * from u2.t1;
       ID
       ----------
       1
```

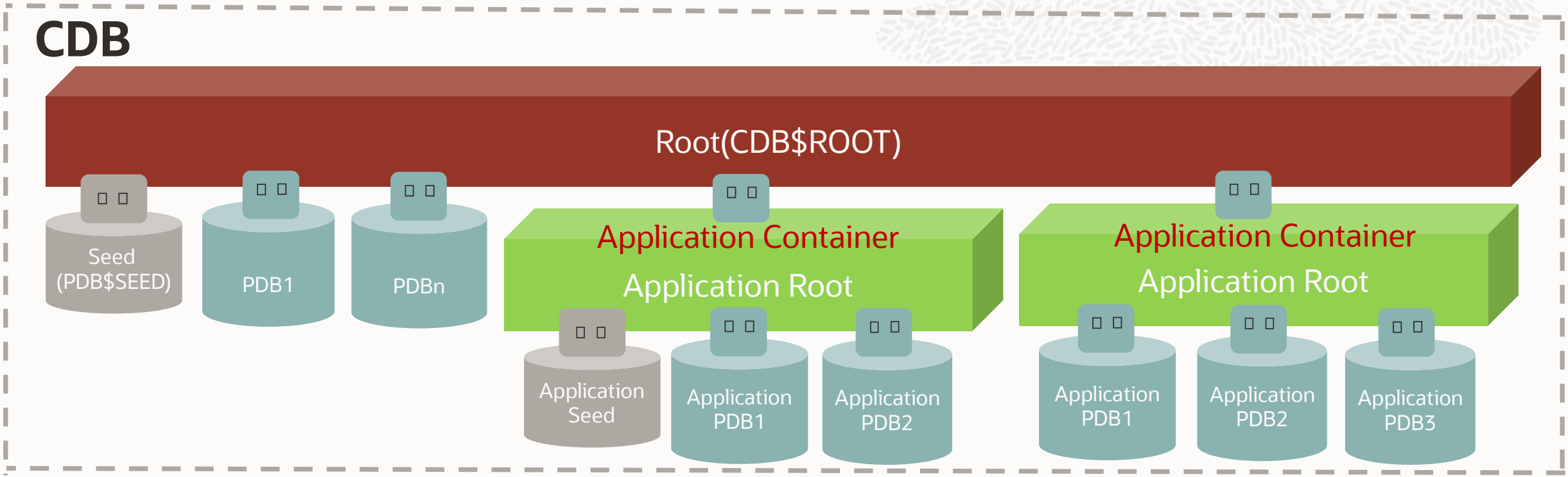
数据在引用数据库中已插入

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.1 应用容器介绍:



- ✓ 在application container中，一个application是存储在application root中的一组命名的、版本化的公共数据和元数据。
- ✓ 在应用容器的上下文中，术语“application”的意思是“master application definition”。例如， application可能包括表、视图和包的定义。
- ✓ application container是一个可选的、用户创建的CDB组件，用于存储一个或多个application后端的数据和元数据。CDB包含零个或多个应用容器。
- ✓ 像CDB本身一样， application container可以包括多个应用PDB，并使这些PDB能够共享元数据和数据。

 Copyright © 2023, Oracle and/or its affiliates

免费咨询热线：400-699-8888

9. 应用容器(application container):

9.2 应用公共对象（Application Common Objects）：

✓ 要创建Application Common Objects，需连接到Application root，然后create语句中指定sharing属性。

✓ 只能在Application安装、升级或打补丁中创建或更改Application Common Objects，可通过以下方式共享：

- DEFAULT_SHARING初始化参数
  该设置是所有支持类型的数据库对象的默认共享属性。
- SHARING子句
  可以在CREATE语句本身中指定此子句。
  SHARING子句优先于DEFAULT_SHARING初始化参数中指定的值。
  可能的值包括METADATA、DATA、EXTENDED DATA和NONE。

✓ 下表显示了Application Common Objects的类型，以及数据和元数据的存储位置。

Object Type	SHARING Value	Metadata Storage	Data Storage
Data-Linked	DATA	Application root	Application root
Extended Data-Linked	EXTENDED DATA	Application root	Application root and application PDB
Metadata-Linked	METADATA	Application root	Application PDB

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.2 应用公共对象（Application Common Objects）：

Sharing = Metadata

- ✓ 在Application Root中定义，并在所有Application PDB中可用，如Tables、Views、PL/SQL程序等
- ✓ 数据将会以私有形式存在于每个Application PDB
- ✓ 如果有需要也可以在Application Root中存放数据
- ✓ 参数default_sharing默认设置为metadata（可以修改）
- ✓ 大多数应用的对象是以元数据链接形式存在

Sharing = Data

- ✓ 定义和数据均存在于Application Root中，如公共的引用数据
- ✓ 通常用于存储共享给所有Application PDB的通用数据（并不是复制）
- ✓ 数据仓库中的维度表特别适合这种模式
- ✓ Application PDB只能访问数据，但不可以修改

Sharing = Extended Data

- ✓ 部分数据存在于Application Root中，在所有Application PDB间共享
- ✓ 每个PDB在访问Application Root上公共的数据同时，还可以创建各自的私有数据，存在于自己的PDB中
- ✓ Application PDB可以修改自己私有数据

 Copyright © 2023, Oracle and/or its affiliates

免费咨询热线：400-699-8888

9. 应用容器(application container):

9.3 创建Application Container：

创建application root：

```
SQL>CREATE PLUGGABLE DATABASE app_con1 AS APPLICATION CONTAINER
      ADMIN USER app1adm IDENTIFIED BY manager;
```

打开application root：

```
SQL>ALTER PLUGGABLE DATABASE app_con1 OPEN;
```

在application root创建application seed：

```
SQL> ALTER SESSION SET CONTAINER=app_con1;
SQL> CREATE PLUGGABLE DATABASE AS SEED ADMIN USER seadm1 IDENTIFIED BY password;
```

注意：application seed的名称为：*application_container_name*$SEED（此处名为APP_CON1$SEED）

在application root创建application pdb：

```
SQL> ALTER SESSION SET CONTAINER=app_con1;
SQL> CREATE PLUGGABLE DATABASE app_pdb1 ADMIN USER appdbadm1 IDENTIFIED BY password;
SQL> alter PLUGGABLE DATABASE app_pdb1 open;
SQL> CREATE PLUGGABLE DATABASE app_pdb2 FROM app_pdb1;
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.4 应用维护：应用安装

第一步，登录application root：

```
SQL> ALTER SESSION SET CONTAINER=app_con1;
```

第二步，开始安装application：

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app BEGIN INSTALL '1.0';
```

第三步，执行安装application语句：

```
SQL> CREATE TABLESPACE ts_tbs DATAFILE SIZE 100M AUTOEXTEND ON NEXT 10M MAXSIZE 200M;
SQL> CREATE USER app_u1 IDENTIFIED BY welcome1 DEFAULT TABLESPACE ts_tbs CONTAINER=ALL;
SQL> GRANT CREATE SESSION, DBA TO app_u1;
SQL> CREATE TABLE app_u1.sales_mlt SHARING=METADATA
       (YEAR NUMBER(4),REGION VARCHAR2(10),QUARTER VARCHAR2(4), REVENUE NUMBER);
```

第四步，结束安装application：

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app END INSTALL '1.0';
```

第五步，在application pdb中同步application：

```
SQL> ALTER SESSION SET CONTAINER=app_pdb1;
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app SYNC;
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.5 应用维护：应用升级

第一步，登录application root：

```
SQL> ALTER SESSION SET CONTAINER=app_con1;
```

第二步，开始application升级：

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app BEGIN UPGRADE '1.0' to '2.0';
```

第三步，执行升级application语句：

```
SQL> CREATE TABLE app_u1.countries_dlt SHARING=DATA (country_id NUMBER,country_name VARCHAR2(20));
SQL> INSERT INTO app_u1.countries_dlt VALUES(86, 'China');
SQL> CREATE TABLE app_u1.zipcodes_edt SHARING=EXTENDED DATA (code VARCHAR2(5),country_id NUMBER,
region VARCHAR2(10));
SQL> INSERT INTO app_u1.zipcodes_edt VALUES ('08820','1','East');
SQL> COMMIT;
```

第四步，结束application升级：

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app END UPGRADE TO '2.0';
```

第五步，在application pdb中同步application：

```
SQL> ALTER SESSION SET CONTAINER=app_pdb1;
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app SYNC;
```

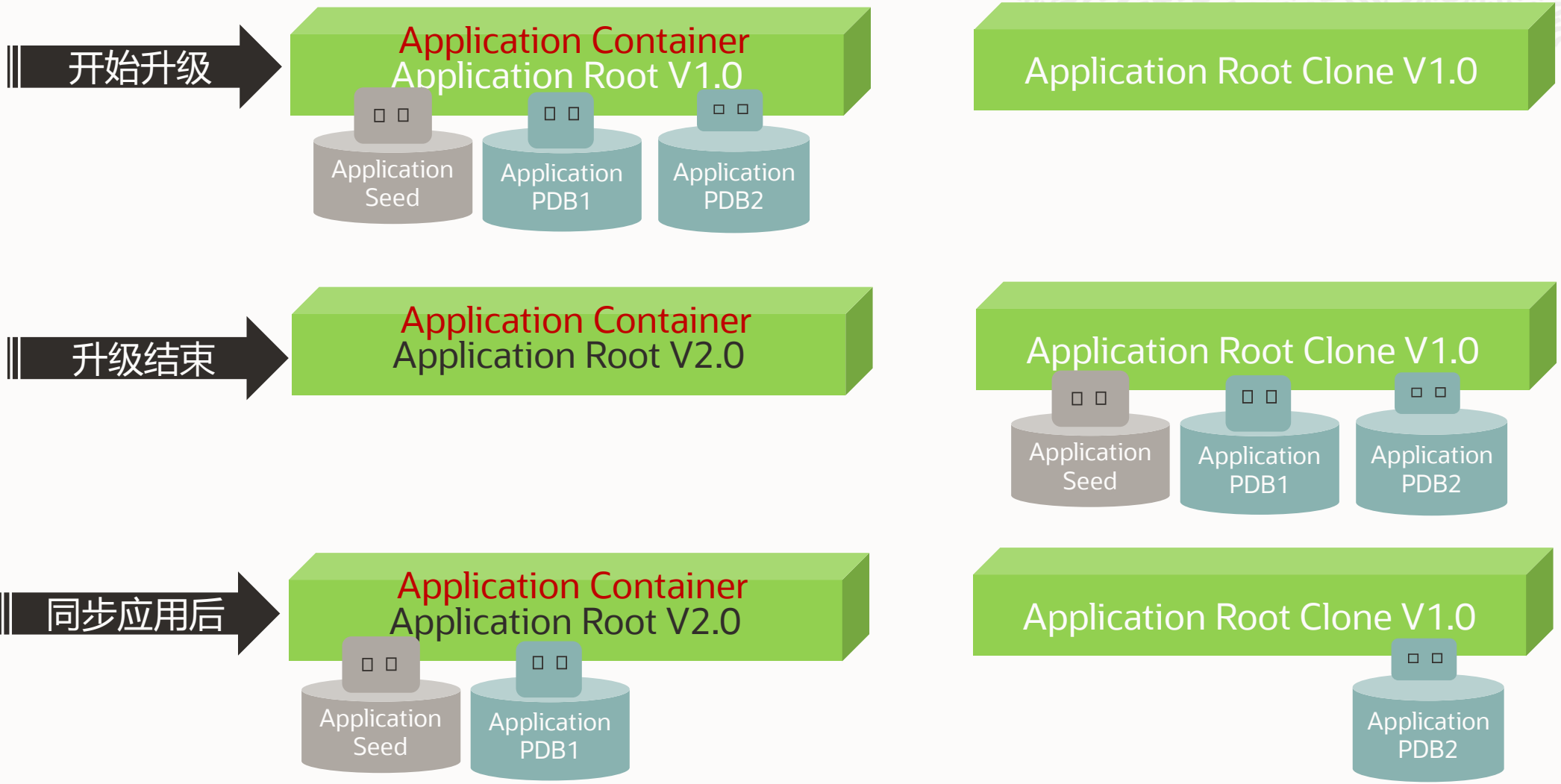
免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.5 应用维护：应用升级实现过程

在应用升级期间，应用仍然可用。Oracle数据库通过克隆Application root来实现此可用性。



免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.6 应用维护：应用打补丁

第一步，登录application root：

```
SQL> ALTER SESSION SET CONTAINER=app_con1;
```

第二步，开始application打补丁：

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app BEGIN PATCH 101 MINIMUM VERSION '1.0';
```

第三步，执行application打补丁语句：

```
SQL> CREATE FUNCTION app_u1.get_total_revenue SHARING=METADATA (p_year IN NUMBER)
         RETURN SYS_REFCURSOR
         AS
         c1_cursor SYS_REFCURSOR;
         BEGIN
         OPEN c1_cursor FOR
               SELECT a.year,sum(a.revenue) FROM containers(app_u1.sales_mlt) a WHERE a.year = p_year GROUP BY a.year;
          RETURN c1_cursor;
        END;
         /
```

第四步，结束application打补丁：

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app END PATCH 101;
```

第五步，在application pdb中同步application：

```
SQL> ALTER SESSION SET CONTAINER=app_pdb1;
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app SYNC;
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.7 应用维护：应用升级与补丁区别

应用升级

- ✓ 应用升级是对已安装应用程序的重大更改。
- ✓ 升级会更改应用的物理体系结构。例如，升级可能会添加新的用户、表和包，或者更改现有对象的定义。
- ✓ 升级会克隆Application root。

应用补丁

- ✓ 应用补丁是对应用程序的微小更改。
- ✓ 应用补丁的典型示例包括bug修复和安全补丁。
- ✓ 一个补丁中允许有函数和程序包。
- ✓ 一般情况下，不允许进行破坏性操作。例如，补丁中不能包含DROP语句，也不能包含删除列或更改数据类型的ALTER TABLE语句。
- ✓ 就像Oracle数据库打补丁过程中限制了Oracle数据库补丁中允许的操作类型一样，应用打补丁过程中也限制应用补丁中所允许的操作。如果补丁中包含引发**“operation not supported in an application patch”**错误的操作，请改为执行应用升级。
- ✓ 打补丁不会克隆Application root。

注意：当另一个应用打补丁或升级正在进行时，无法进行应用打补丁。

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.8 应用维护：卸载应用

第一步，登录application root：

```
SQL> ALTER SESSION SET CONTAINER=app_con1;
```

第二步，开始application卸载：

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app BEGIN UNINSTALL;
```

第三步，执行application卸载语句：

```
SQL> DROP USER APP_U1 CASCADE;
SQL> DROP TABLESPACE TS_TBS INCLUDING CONTENTS AND DATAFILES;
```

第四步，结束application卸载：

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app END UNINSTALL;
```

第五步，在application pdb中同步application：

```
SQL> ALTER SESSION SET CONTAINER=app_pdb1;
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app SYNC;
```

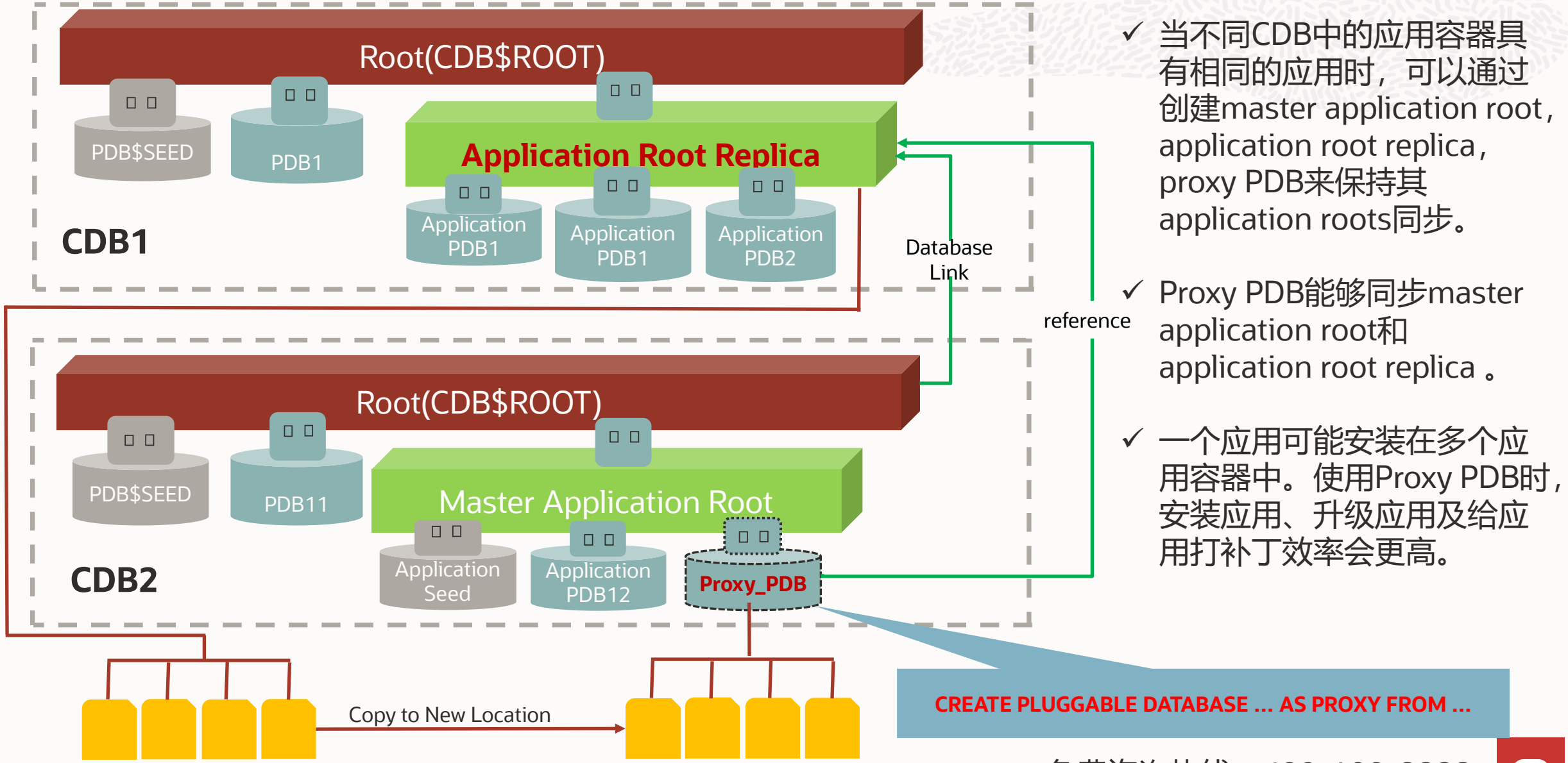
说明：卸载会克隆Application root。

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.9 通过Proxy PDB实现跨CDB操作



- ✓ 当不同CDB中的应用容器具有相同的应用时，可以通过创建master application root，application root replica，proxy PDB来保持其application roots同步。
- ✓ Proxy PDB能够同步master application root和application root replica 。
- ✓ 一个应用可能安装在多个应用容器中。使用Proxy PDB时，安装应用、升级应用及给应用打补丁效率会更高。

58 Copyright © 2023, Oracle and/or its affiliates

免费咨询热线：400-699-8888

9. 应用容器:

9.10 容器映射（container map）介绍

容器映射（container map）使连接到Application Root的会话发出的SQL语句能够路由到正确的PDB，具体取决于SQL语句中使用的谓词的值。

使用container map的关键组件:

元数据链接表（Metadata-linked table）

- ✓ 此表是要使用容器映射进行查询的表
- ✓ 此表在表级别并没有进行物理分区

映射表（Map table）

- ✓ 此表是Application Root中创建的一个按list、hash或range分区的单列分区表
- ✓ 此表允许使用容器映射启用的分区策略来查询元数据链接表
- ✓ 此表中的分区名称必须与应用容器中的Application PDB名称匹配

容器映射（Container map）

- ✓ 容器映射是指定映射表的数据库属性
- ✓ 要设置该属性，需连接到Application Root并执行如下语句:

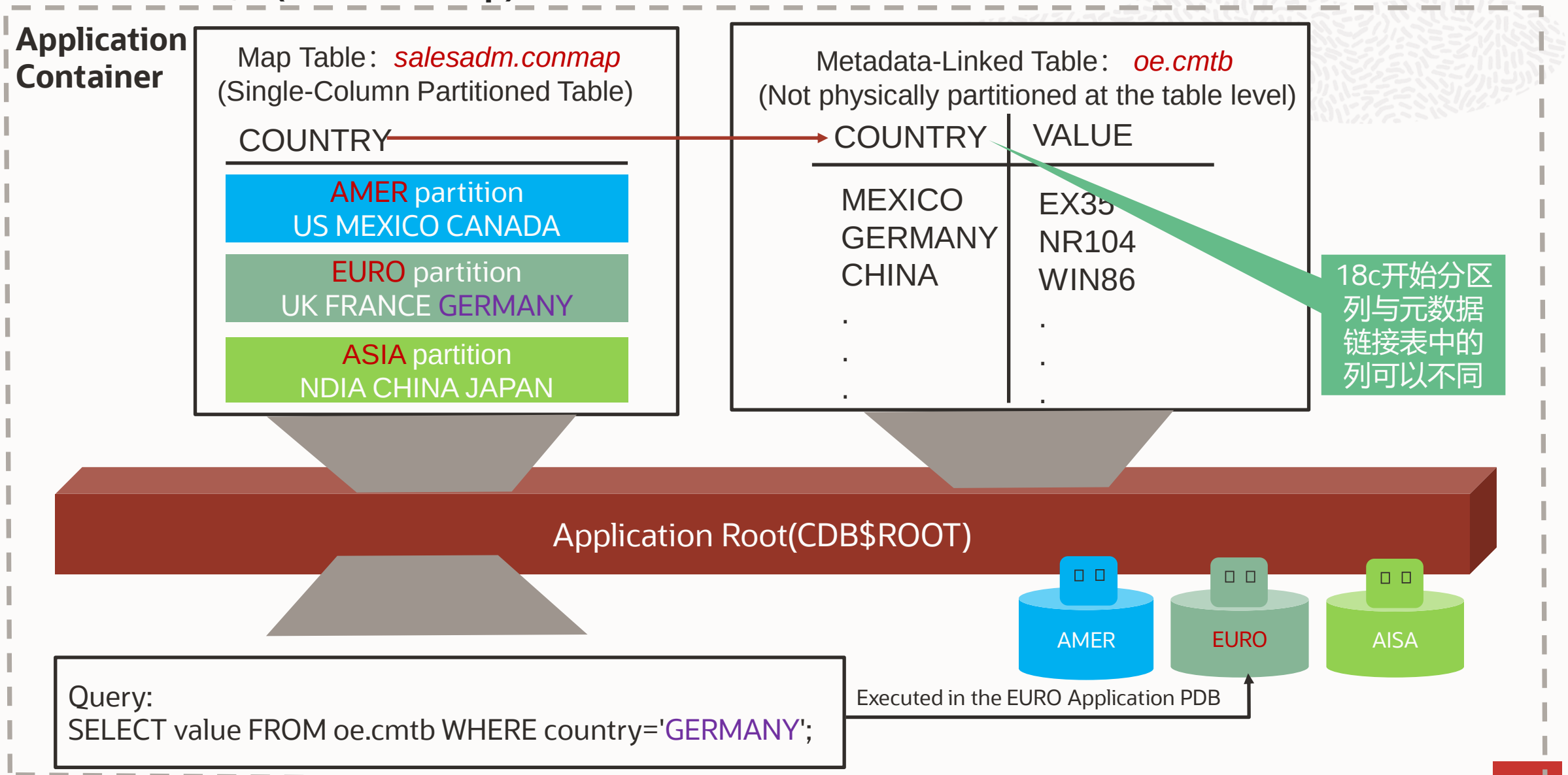
```
SQL> ALTER PLUGGABLE DATABASE SET CONTAINER_MAP=map_table;
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.10 容器映射（container map）示意图



 Copyright © 2023, Oracle and/or its affiliates

免费咨询热线：400-699-8888

9. 应用容器(application container):

9.10 创建及使用容器映射（container map）

第一步，登录application root：

提前创建好应用容器包含三个应用PDB、salesadm和oe用户

```
SQL> ALTER SESSION SET CONTAINER=app_con1;
```

第二步，创建映射表：

```
SQL> CREATE TABLE salesadm.conmap (country VARCHAR2(30) NOT NULL)
       PARTITION BY LIST (country) (
       PARTITION AMER VALUES ('US','MEXICO','CANADA'),
       PARTITION EURO VALUES ('UK','FRANCE','GERMANY'),
       PARTITION ASIA VALUES ('INDIA','CHINA','JAPAN'));
```

第三步，设置数据库属性CONTAINER_MAP ：

```
SQL> ALTER PLUGGABLE DATABASE SET CONTAINER_MAP='salesadm.conmap';
```

第四步，开始应用安装：

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION salesapp BEGIN INSTALL '1.0';
```

第五步，创建元数据链接表：

```
SQL> CREATE TABLE oe.cmtb SHARING=METADATA ( value VARCHAR2(30),country VARCHAR2(30));
```

第六步，为要查询元数据链接表启用容器映射：

```
SQL> ALTER TABLE oe.cmtb ENABLE CONTAINER_MAP;
```

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.10 创建及使用容器映射（container map）

第七步，确保要查询的元数据链接表启用了 CONTAINERS 子句：

```
SQL> ALTER TABLE oe.cmtb ENABLE CONTAINERS_DEFAULT;
```

第八步，结束应用安装：

```
SQL> ALTER PLUGGABLE DATABASE APPLICATION salesapp END INSTALL '1.0';
```

第九步，同步每个Application PDB：

```
SQL> ALTER SESSION SET CONTAINER=amer;
SQL> ALTER PLUGGABLE DATABASE APPLICATION salesapp SYNC;
SQL> ALTER SESSION SET CONTAINER=euro;
SQL> ALTER PLUGGABLE DATABASE APPLICATION salesapp SYNC;
SQL> ALTER SESSION SET CONTAINER=asia;
SQL> ALTER PLUGGABLE DATABASE APPLICATION salesapp SYNC;
```

第十步，根据分区策略将值插入到每个应用程序PDB中的oe.cmtb表中：

```
SQL> ALTER SESSION SET CONTAINER=amer;
SQL> INSERT INTO oe.cmtb VALUES ('AMER VALUE','US');
SQL> INSERT INTO oe.cmtb VALUES ('AMER VALUE','MEXICO');
SQL> INSERT INTO oe.cmtb VALUES ('AMER VALUE','CANADA');
SQL> COMMIT;
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.10 创建及使用容器映射（container map）

第十步（续），根据分区策略将值插入到每个应用程序PDB中的oe.cmtb表中：

```
SQL> ALTER SESSION SET CONTAINER=euro;
SQL> INSERT INTO oe.cmtb VALUES ('EURO VALUE','UK');
SQL> INSERT INTO oe.cmtb VALUES ('EURO VALUE','FRANCE');
SQL> INSERT INTO oe.cmtb VALUES ('EURO VALUE','GERMANY');
SQL> COMMIT;
SQL> ALTER SESSION SET CONTAINER=asia;
SQL> INSERT INTO oe.cmtb VALUES ('ASIA VALUE','INDIA');
SQL> INSERT INTO oe.cmtb VALUES ('ASIA VALUE','CHINA');
SQL> INSERT INTO oe.cmtb VALUES ('ASIA VALUE','JAPAN');
SQL> COMMIT;
```

第十一步，切换到Application Root中使用容器映射查询数据：

```
SQL> ALTER SESSION SET CONTAINER=app_con1;
SQL> SELECT value FROM oe.cmtb WHERE country='MEXICO';
SQL> SELECT value FROM oe.cmtb WHERE country='GERMANY';
SQL> SELECT value FROM oe.cmtb WHERE country='JAPAN';
```

查询会自动路由到对应的PDB将结果返回

执行： SQL> select * from containers(oe.cmtb); 或 select * from oe.cmtb; 将返回所有PDB中的全部数据。

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.11 应用维护示例：app_pdb2数据查询

pdb2同步application后查询数据：

```
SQL> ALTER SESSION SET CONTAINER=app_pdb2;
SQL> ALTER PLUGGABLE DATABASE APPLICATION sales_app SYNC;
SQL> select * from app_u1.sales_mlt;
          no rows selected

SQL> select * from app_u1.countries_dlt;
          COUNTRY_ID COUNTRY_NAME
          ---------- --------------------
          86         China

SQL> select * from app_u1.zipcodes_edt;
          CODE     COUNTRY_ID REGION
          -------- ---------- ---------
          08820    1          East

SQL> desc app_u1.get_total_revenue;
          FUNCTION app_u1.get_total_revenue RETURNS REF CURSOR
          Argument Name          Type                 In/Out   Default?
          ---------------------- -------------------- -------- --------
          P_YEAR                 NUMBER               IN
```

app_pdb2同步应用

同步后元数据链接表无数据

同步后数据链接表有只读数据

同步后扩展数据链接表有数据

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.11 应用维护示例：app_pdb2数据插入

```
SQL> ALTER SESSION SET CONTAINER=app_pdb2;
        Session altered.

SQL> INSERT INTO app_u1.sales_mlt VALUES(2023, 'EAST', 'Q1',100000);
        1 row created.
SQL> commit;
        Commit complete.

SQL> INSERT INTO app_u1.countries_dlt VALUES(44, 'UK');
        INSERT INTO app_u1.countries_dlt VALUES(44, 'UK')
                    *
        ERROR at line 1:
        ORA-65097: DML into a data link table is outside an application action

SQL> INSERT INTO app_u1.zipcodes_edt VALUES ('94065','1','West');
        1 row created.
SQL> commit;
        Commit complete.
```

切换到app_pdb2并插入数据

app_pdb2中向元数据链接表插入数据

app_pdb2中无法向数据链接表插入数据

app_pdb2中扩展数据链接表插入数据

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.11 应用维护示例：app_pdb2查询插入数据

```
SQL> ALTER SESSION SET CONTAINER=app_pdb2;
          Session altered.

SQL> select * from app_u1.sales_mlt;
        YEAR    REGION     QUAR    REVENUE
        ---------- -------------   ----------  ---------------
        2023      EAST         Q1         100000

SQL> select * from app_u1.countries_dlt;
        COUNTRY_ID   COUNTRY_NAME
        -------------------  ------------------------
        86                   China

SQL> select * from app_u1.zipcodes_edt;
         CODE    COUNTRY_ID  REGION
          --------   ------------------  -------------
          08820    1                   East
          94065    1                   West
```

切换到app_pdb2查询插入数据

由app_pdb2向元数据链接表插入的数据

Application root中数据链接表的数据

Application root中扩展数据链接表的数据

由app_pdb2向扩展数据链接表插入的数据

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.11 应用维护示例：application root数据查询

```
SQL> alter session set container=APP_CON1;
        Session altered.

SQL> select * from app_u1.sales_mlt;
         no rows selected


SQL> select * from app_u1.countries_dlt;
        COUNTRY_ID    COUNTRY_NAME
        ----------    ------------
        86            China


SQL> select * from app_u1.zipcodes_edt;
         CODE    COUNTRY_ID   REGION
         -----   ----------   ------
         08820   1            East
```

切换到Application root中查询数据

Application root中元数据链接表无数据，只有表定义

Application root中数据链接表中数据对于其pdb是只读数据

Application root中扩展数据链接表中数据对于其pdb是只读数据，每个PDB可以插入自己的数据

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.11 应用维护示例：app_pdb1数据插入

```
SQL> ALTER SESSION SET CONTAINER=app_pdb1;
        Session altered.

SQL> INSERT INTO app_u1.sales_mlt VALUES(2023, 'EAST', 'Q2',200000);
        1 row created.

SQL> INSERT INTO app_u1.countries_dlt VALUES(10, 'Beijing');
        INSERT INTO app_u1.countries_dlt VALUES(10, 'Beijing')
                    *
        ERROR at line 1:
        ORA-65097: DML into a data link table is outside an application action

SQL> INSERT INTO app_u1.zipcodes_edt VALUES ('08823','2','East2');
        1 row created.

SQL> commit;
        Commit complete.
```

切换到app_pdb1并插入数据

app_pdb1中向元数据链接表插入数据

app_pdb1中无法向数据链接表插入数据

app_pdb1中向扩展数据链接表插入数据

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

9. 应用容器(application container):

9.11 应用维护示例：数据查询及containers()操作

```
SQL> ALTER SESSION SET CONTAINER=app_pdb1;
        Session altered.
SQL> select * from app_u1.sales_mlt;
        YEAR    REGION      QUAR    REVENUE
       -------  ----------- --------- -----------
        2023    EAST        Q2        200000
SQL> select * from app_u1.countries_dlt;
        COUNTRY_ID    COUNTRY_NAME
        ---------------  -----------------
        86                China
SQL> select * from app_u1.zipcodes_edt;
        CODE    COUNTRY_ID    REGION
        -------  -------------   -----------
        08820       1             East
        08823       2             East2
SQL> alter session set container=APP_CON1;
        Session altered.
SQL> select * from CONTAINERS(app_u1.sales_mlt);
        YEAR      REGION    QUAR   REVENUE    CON_ID
        --------- ---------- --------- -----------  ----------
        2023      EAST       Q1       100000       14
        2023      EAST       Q2        200000      12
```

切换到app_pdb1检查插入数据

由app_pdb1向元数据链接表插入的数据

Application root中数据链接表的数据

由app_pdb1向扩展数据链接表插入的数据

切换到Application root执行containers()操作

数据来自app_pdb2

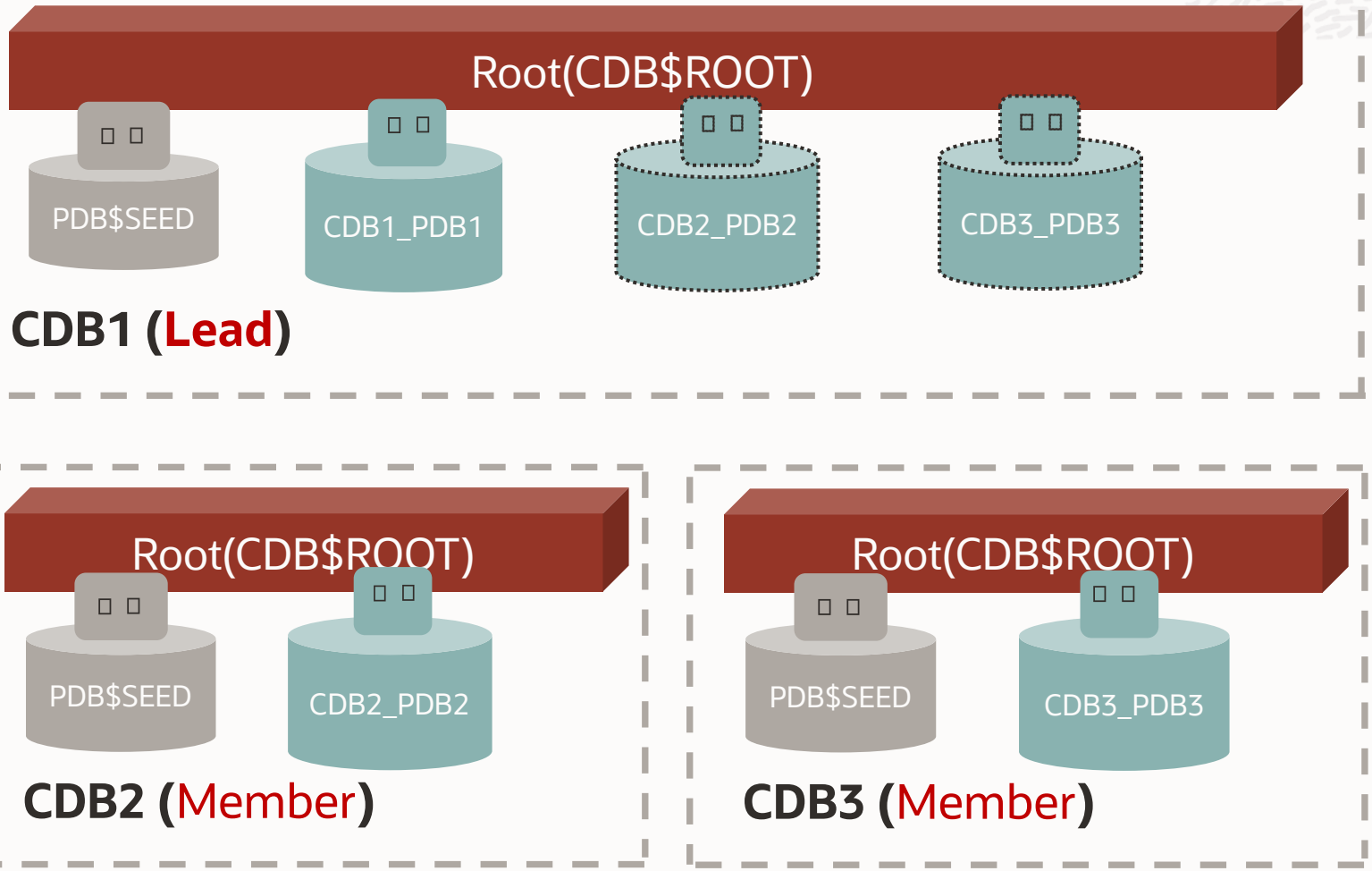
数据来自app_pdb1

 Copyright © 2023, Oracle and/or its affiliates

免费咨询热线：400-699-8888

10. CDB舰队(CDB FLEET):

10.1 CDB舰队架构:



- ✓ CDB舰队是多个CDB及其包含的所有PDB的集合，可以将其作为一个逻辑CDB进行统一管理。
- ✓ CDB舰队给多个CDB的可扩展性和集中管理提供了数据库基础架构。
- ✓ 舰队中的CDB支持所有Oracle数据库功能，如Oracle RAC、RMAN、基于时间点的恢复和闪回功能等。
- ✓ CDB船队中所有CDB的每个PDB名称必须唯一。
- ✓ CDB舰队包括:
  - Lead CDB(设置LEAD_CDB=TRUE)
  - CDB Fleet Member(设置LEAD_CDB_URI)

说明: 配置CDB舰队后，来自不同CDB的PDB信息将与Lead CDB同步。各个CDB中的所有PDB在Lead CDB中当前是“可见”的，使您能够把Lead CDB作为单个的逻辑CDB来访问舰队中的所有PDB。

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

10. CDB舰队(CDB FLEET):

10.2 创建CDB舰队：设置Lead CDB

Lead CDB是监控和管理舰队中CDB的中心位置。

第一步，登录CDB1 root：

```
SQL> ALTER SESSION SET CONTAINER = CDB$ROOT;
```

第二步，检查LEAD_CDB数据库属性：

```
SQL> SELECT PROPERTY_VALUE FROM DATABASE_PROPERTIES WHERE PROPERTY_NAME='LEAD_CDB';
或： SQL> SELECT sys_context('userenv', 'is_lead_cdb') LEAD_CDB FROM dual;
```

第三步，设置LEAD_CDB属性：

```
SQL>ALTER DATABASE SET LEAD_CDB = TRUE;
```

第四步，检查数据库alert日志：

```
The role of current CDB in the Fleet is: LEAD
Completed: ALTER DATABASE SET LEAD_CDB = TRUE
```

第五步，创建用于Member连接的用户：

```
SQL>CREATE USER C##CF1 IDENTIFIED BY welcome1 CONTAINER=ALL;
SQL>GRANT CREATE SESSION TO C##CF1 CONTAINER=ALL;
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

10. CDB舰队(CDB FLEET):

10.3 创建CDB舰队：指定舰队Member

CDB Fleet Member是CDB舰队中指向Lead CDB的其它CDB。

第一步，登录CDB2|CDB3 root：

```
SQL> ALTER SESSION SET CONTAINER = CDB$ROOT;
```

第二步，检查'LEAD_CDB_URI数据库属性：

```
SQL>SELECT PROPERTY_VALUE FROM DATABASE_PROPERTIES WHERE PROPERTY_NAME='LEAD_CDB_URI';
或：SQL> SELECT sys_context('userenv', 'is_member_cdb') MEMBER_CDB FROM dual;
```

第三步，创建到Lead CDB的database link：

```
SQL> CREATE PUBLIC DATABASE LINK lead_link CONNECT TO C##CF1 IDENTIFIED BY welcome1 USING
'host:port/cdb1';
```

第四步，设置LEAD_CDB_URI属性：

```
SQL>ALTER DATABASE SET LEAD_CDB_URI = 'dblink:LEAD_LINK';
```

第五步，检查数据库alert日志：

```
The role of current CDB in the Fleet is: MEMBER
Completed: ALTER DATABASE SET LEAD_CDB_URI = 'dblink:LEAD_LINK'
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

10. CDB舰队(CDB FLEET):

10.4 CDB舰队：示例

➢ **创建CDB舰队前状态:**

```
SYS@cdb1> show pdbs
    CON_ID CON_NAME                       OPEN MODE        RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY        NO
         3 CDB1_PDB1                      READ WRITE       NO

SYS@cdb2 > show pdbs
    CON_ID CON_NAME                       OPEN MODE        RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY        NO
         3 CDB2_PDB2                      READ WRITE       NO

SYS@cdb3 > show pdbs
    CON_ID CON_NAME                       OPEN MODE        RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY        NO
         3 CDB3_PDB3                      READ WRITE       NO
```

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

10. CDB舰队(CDB FLEET):

10.4 CDB舰队：示例

➢ **创建CDB舰队后状态：**

```
SYS@cdb1> show pdbdetails
            APP                                                  APP
           ROOT                        APP     APP    APP      ROOT      PXY
CONID CONID CON_NAME                   ROOT    PDB    SEED     CLONE     PDB      OPEN MODE    REST
--------- --------- ------------------ ------- ------ -------  --------- -------- ------------ -------
   2                  PDB$SEED         NO      NO     NO       NO        NO       READ ONLY    NO
   3                 CDB1_PDB1         NO      NO     NO       NO        NO       READ WRITE   NO
   5                 CDB2_PDB2         NO      NO     NO       NO        YES      MOUNTED
   7                 CDB3_PDB3         NO      NO     NO       NO        YES      MOUNTED

SYS@cdb1 > select con_id,pdb_name,status,is_proxy_pdb from cdb_pdbs;
CON_ID   PDB_NAME          STATUS     IS_PROXY_PDB
---------   ------------------   ------------   ----------------------
       2    PDB$SEED         NORMAL     NO
       3    CDB1_PDB1        NORMAL     NO
       4    CDB2             STUB       YES
       5    CDB2_PDB2        STUB       YES
       6    CDB3             STUB       YES
       7    CDB3_PDB3        STUB       YES
```

只能在PDB物理存在的CDB中OPEN

STUB表明是Fleet Member

免费咨询热线：400-699-8888

 Copyright © 2023, Oracle and/or its affiliates

Q&A

谢 谢!

免费咨询热线：400-699-8888

深入了解Oracle审计仓库与数据库防火墙(AVDF)

数据安全实战演练系列(二)

钟远

- 资深数据安全专家
- 10年以上系统安全运维和管理库管理经验
- 电信行业背景，在智能运维与数据安全架构方面经验丰富

内容简介

深入了解Oracle审计仓库和数据库防火墙(AVDF)，理解AVDF的原理，架构，功能，以及如何部署使用等.通过现场DEMO演示掌握AVDF的使用，主要包括：

- 如何做用户权限审计
- 如何做敏感数据审计
- 如何查看审计关键表值的前/后变化
- 如何创建审计告警策略
- 如何拦截SQL语句注入

Zoom直播

直播时间：8月4日 11:00 - 12:00
扫描二维码进入直播
Zoom ID：957 9669 6723
密码：20212023

微信扫一扫预约

数据库和云讲座群

20-21

甲骨文云技术公众号

技术专家1V1深入交流