



海量数据学院
VASTDATA EDUCATION



ORACLE
ACE

VDOUG
VastData Oracle User Group
海量数据Oracle用户组

OMOOCU
Oracle Massive Open Online Courses Union
Oracle 慕课联盟

OCM
之家

讲师：崔旭

网名：DBstyle

海量数据学院首席讲师，院长兼总经理，Oracle ACE，中国OCM之家(OCMH)发起人，同时创办了Oracle慕课联盟(OMOOCU)，用互联网+的形式推广线上免费课程。获得Oracle 10g 11g 12c OCM认证。从业10年以上，资深Oracle数据库专家，51CTO认证讲师，在数据库领域有丰富的经验。拥有Oracle数据库，SQL Server数据库，RHCE, F5, Cisco等十余种相关技术认证。曾任职于北京神州泰岳软件股份有限公司、北京电信发展有限公司，云和恩墨的恩墨学院教学总监。负责运维全国各省客户的海量数据库，负责高可用数据库的部署实施、故障处理、性能优化，教育培训等工作。现任海量数据学院首席讲师兼院长，为多家大中型企业和多所国家211工程高等学校，提供过Oracle相关课程培训以及技术分享活动。讲课富有亲和力和感染力，擅长理论联系实际，通过华丽的操作将枯燥的技术展现出来，使学员理解技术在真实生产中的应用。至今培养OCP和OCM数千人，培训经验丰富，致力于推广和分享ORACLE技术。技术博客地址：<http://www.dbstyle.net>



ORACLE® Certified Master

Oracle Database 12c Administrator

ORACLE® Certified Master

Oracle Database 11g Administrator

ORACLE® Certified Master

Oracle Database 10g Administrator



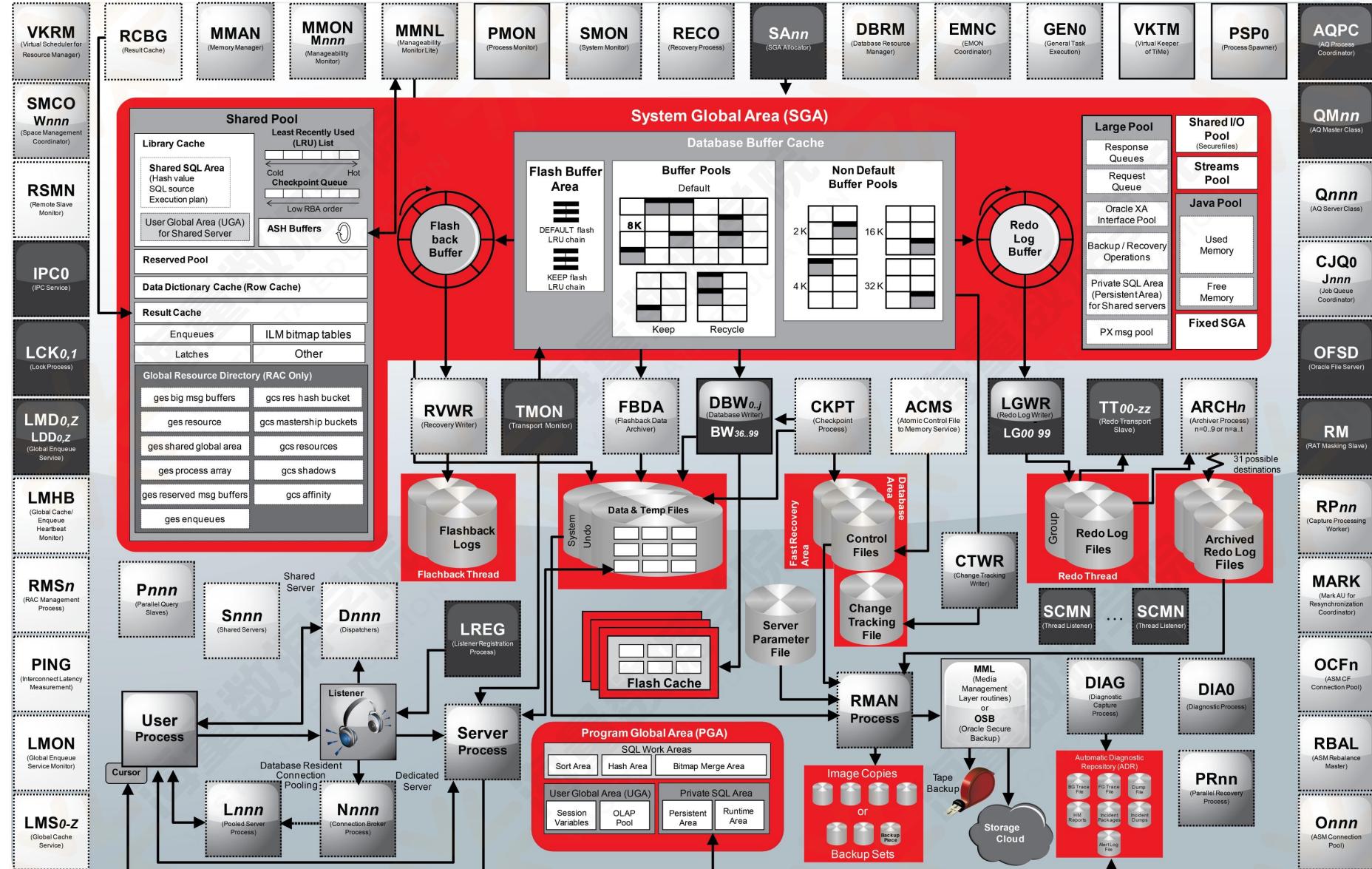
海量数据
VASTDATA EDUCATION

Examination time

- | | | |
|---|-------|---------|
| 1. General Database and
Network Administration,
and Backup Strategy | | 120 min |
| 2. Data and Performance
Management | | 90 min |
| 3. Data Guard | | 90 min |
| 4. Grid Infrastructure and
Real Application Clusters | | 90 min |

Oracle Database 12c Certified Master Upgrade Exam





Enterprise Manager Cloud Control 和其他工具

课程目标

学完本课后，应能完成以下工作：

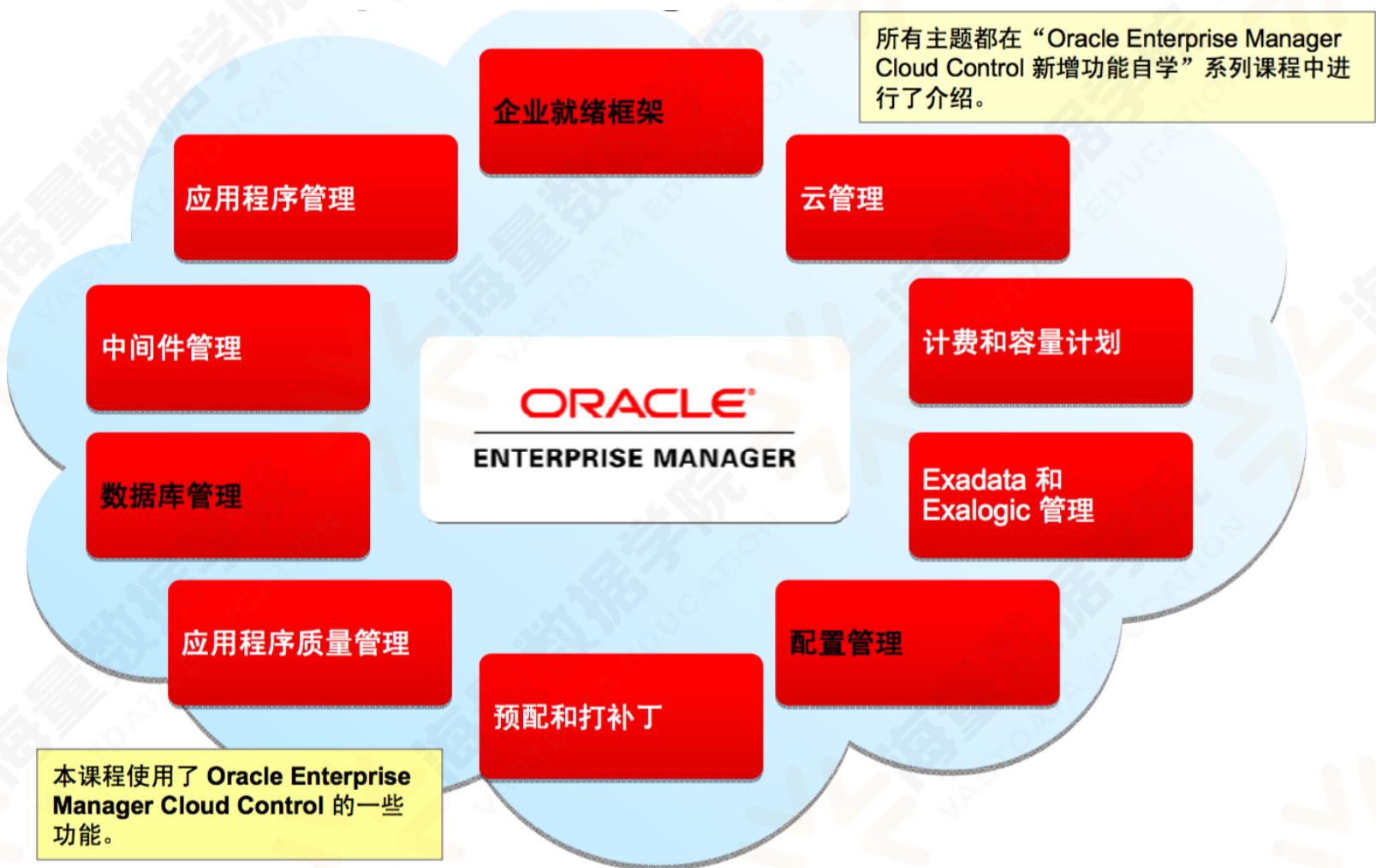
- 描述 Cloud Control 的不同组件
- 说明 Cloud Control 的体系结构
- 列出 Cloud Control 管理的目标类型
- 浏览 Oracle Enterprise Manager Cloud Control 界面
- 描述 Oracle Enterprise Manager Database Express 导航
- 使用 DBCA 创建新的数据库类型
- 浏览 Oracle SQL Developer 界面

管理员的主要难题

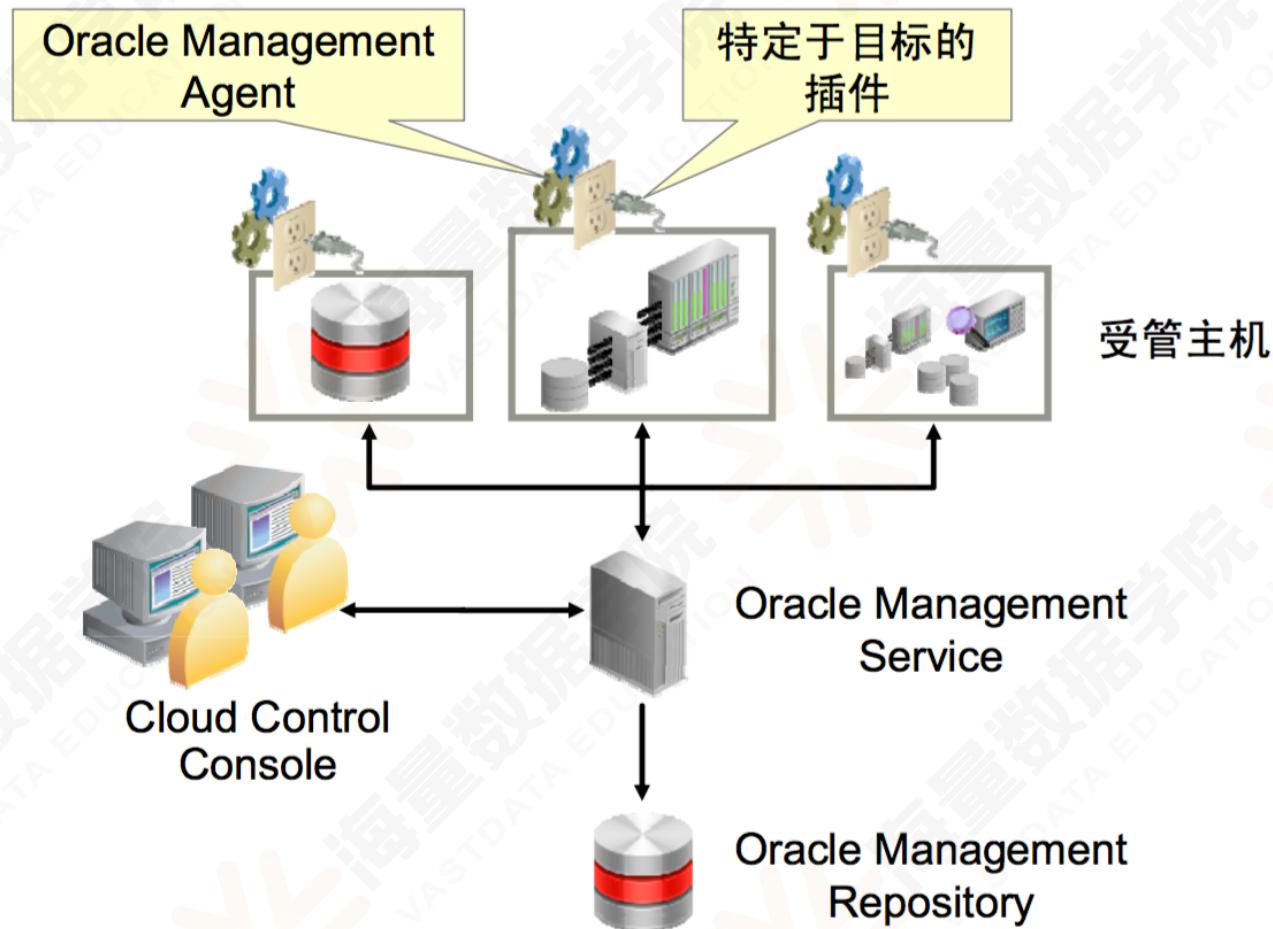
随着数据中心扩展到云环境,管理数据中心的挑战也随之增多。管理数据中心的主要挑战包括:

- 监视性能和可用性
- 快速解决问题
- 控制操作成本
- 将 IT 与业务优先级关联

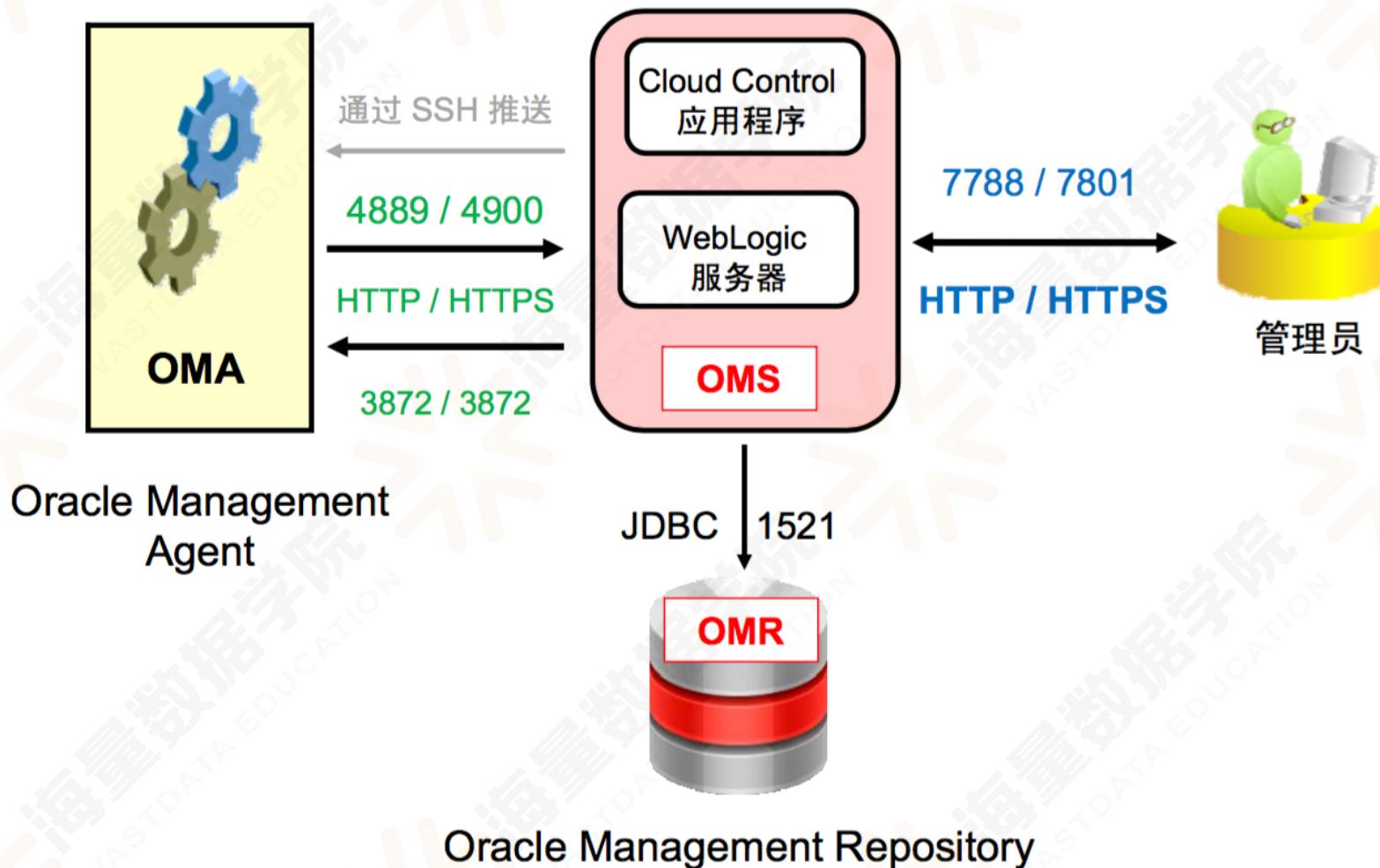
Enterprise Manager Cloud Control



Cloud Control 组件



组件和通信流



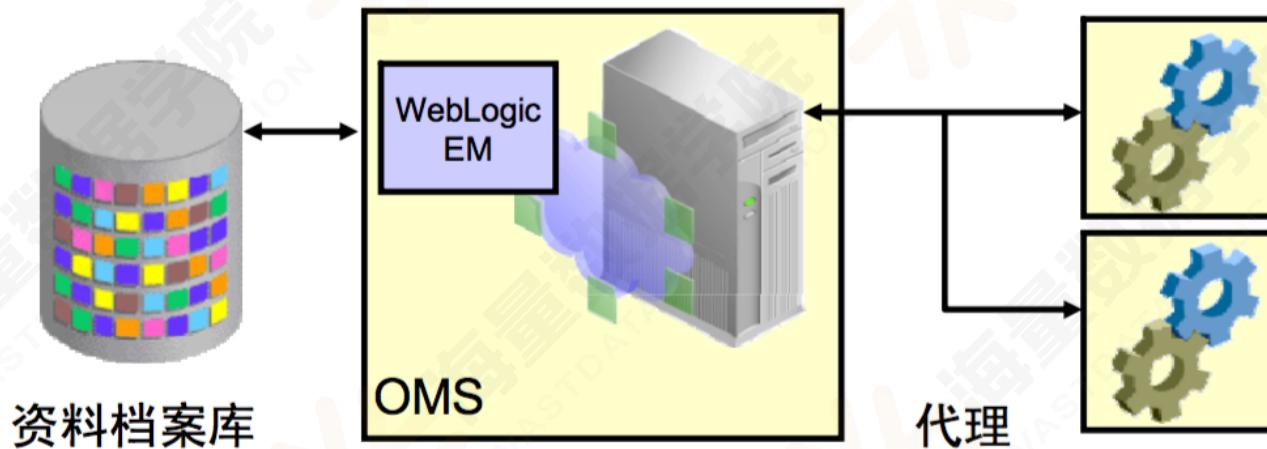
Oracle Management Repository

Oracle Management Repository (OMR):

- 位于 Oracle DB 中
- 包括属于 SYSMAN 的方案
- 必须安装在预先存在的数据库中
- 可以安装在 RAC 数据库中

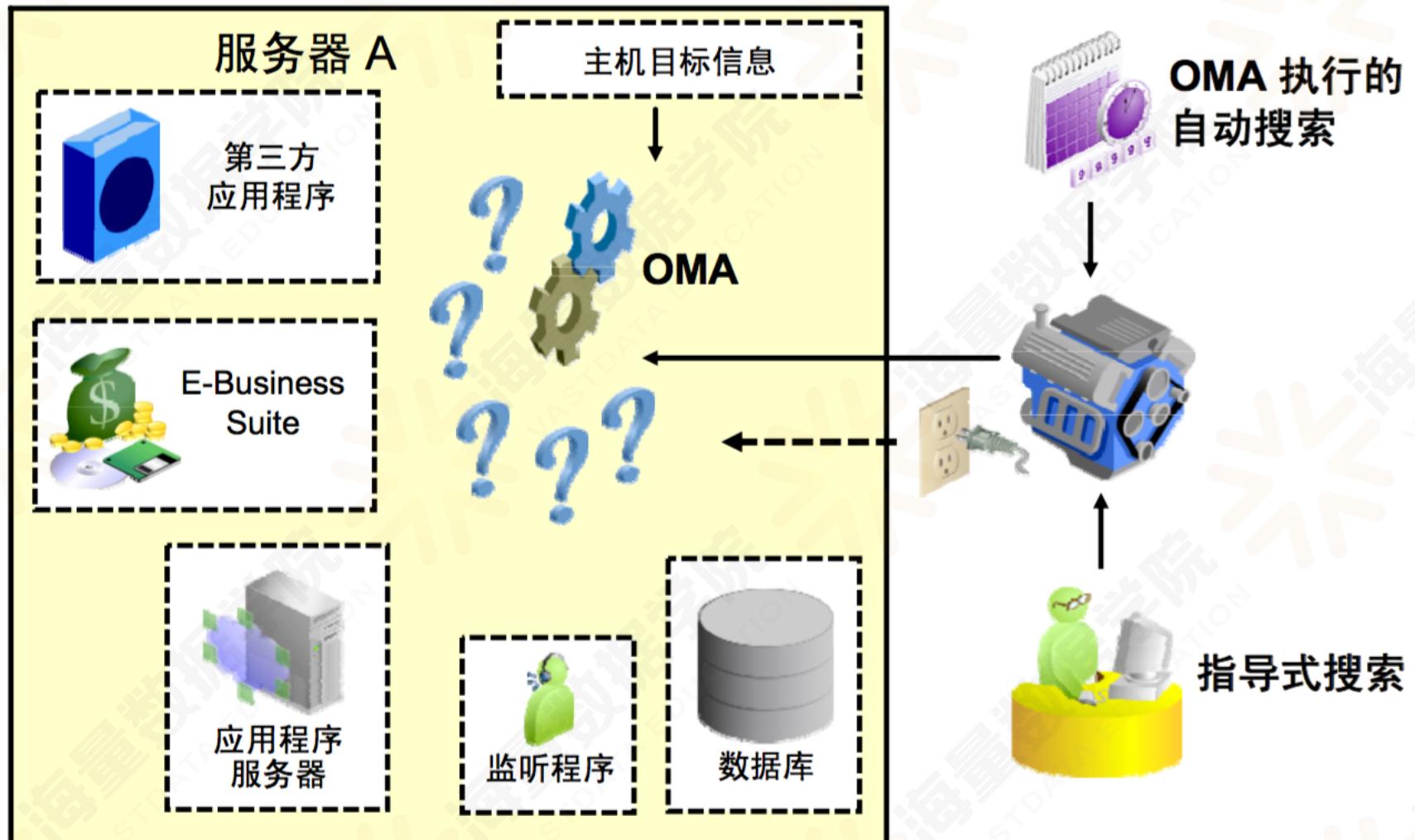
注：使用 Oracle DB 的受限使用许可证

控制 Enterprise Manager Cloud Control 框架



组件控制实用程序		
资料档案库	OMS	代理
sqlplus 或 srvctl	emctl	emctl
lsnrctl		

目标搜索



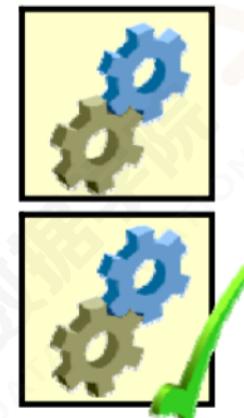
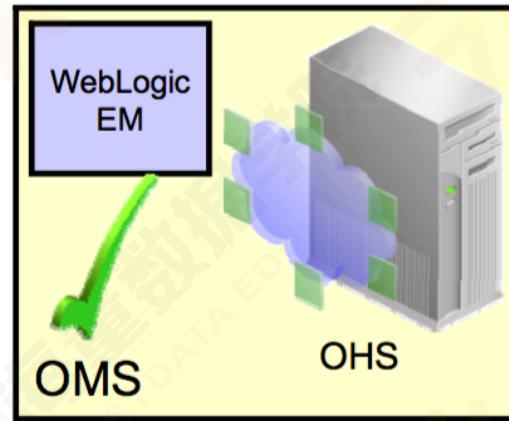
启动 Enterprise Manager Cloud Control 框架

要启动 Cloud Control 框架，请执行以下步骤：

1. 启动资料档案库数据库监听程序。
2. 启动资料档案库数据库实例。
3. 启动 OMS。
4. 启动 OMS/资料档案库服务器上的代理。
5. 启动受管服务器上的代理。



资料档案库



代理



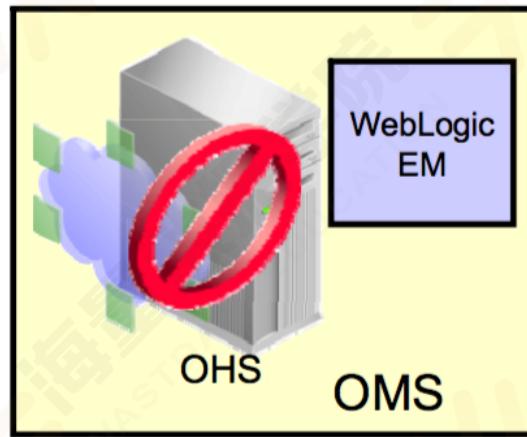
停止 Enterprise Manager Cloud Control 框架

要停止 Enterprise Manager Cloud Control 框架, 请执行以下步骤:

1. 停止受管服务器上的代理。
2. 停止 OMS/资料档案库服务器上的代理。
3. 停止 OMS。
4. 停止资料档案库数据库实例。



代理



资料档案库

不同目标类型

Enterprise Manager Cloud Control 可以监视、管理和维护许多种不同类型的目标，其中包括：

- Oracle DB
- Oracle DB 监听程序
- Oracle Fusion Middleware 产品
- Oracle Application Server
- Oracle WebLogic Server
- Oracle 应用产品，包括 E-Business Suite、SOA、Siebel 和 PeopleSoft
- Exadata 和 Exalogic
- Cloud Control 组件：OMR 和 OMS
- 第三方产品



Enterprise Manager Cloud Control

ORACLE Enterprise Manager Cloud Control 12c

Enterprise Targets Favorites History

Search Target Name Page Refreshed Aug 22, 2011 2:29:31 PM UTC

Enterprise Summary

Overview View All Targets

Targets Monitored 1420

Status

Targets with Status 1006
Targets with Pending Activation 122

Down(133)	Metric Collection	Error(16)	Agent	Unreachable(136)	Status Pending(44)	Up(677)
-----------	-------------------	-----------	-------	------------------	--------------------	---------

Incidents

Open 816
Updated in last 24 hours 333

Category	-	✗	⚠	🚩
Availability	238	13	2	-
Performance	-	2	-	-
Security	-	18	-	-
Others	-	533	23	-

Problems

Open 57 Without Service Request 57
Updated in last 24 hours 33

Jobs

Suspended Executions (last 7 days) 27 ⓘ
Problem Executions (last 7 days) 54438 ✗
Action Required Executions (last 7 days) 0 ✓

Patch Recommendations

View by Classification Target Type

Other Recommendations Security

Inventory and Usage

Show Hosts See Details

View Platform

Platform	Hosts	OS Patches
Enterprise Linux AS release 4 (October Update 8)	23	No
Enterprise Linux Server release 5.6 (Carthage)	19	No
Enterprise Linux Server release 5.4 (Carthage)	9	No
SunOS	2	No
Enterprise Linux Enterprise Linux AS release 4 (October Update 8)	1	No

Compliance Summary

Compliance Frameworks Compliance Standards

View View Trends

Name	Target Evaluations	Violations	Average Compliance Score (%)
No data to display	✗ ⚠ ⚡ ✗ ⚠ ⚡	✗ ⚠ ⚡ ✗ ⚠ ⚡	

Least Compliant Targets

View View Trends

Target Name	Target Type	Standard Evaluations	Violations	Average Compliance Score (%)
ark-2101013-1@oracle.com	Host	1 0 0 4 0 0	51	
oracle.com	Host	1 0 0 1 0 0	51	
ade.com	Host	1 0 0 4 0 0	51	
oracle.com	Host	1 0 0 1 0 0	51	
ade.com	Host	1 0 0 5 0 0	51	

Service Requests

Sign in to My Oracle Support
Enter your Single Sign-On username and password.

* User Name
* Password Go

Lost your password?

用户界面

设置主页：

- 基于角色的预定义主页
- 任意页面

基于菜单的导航：

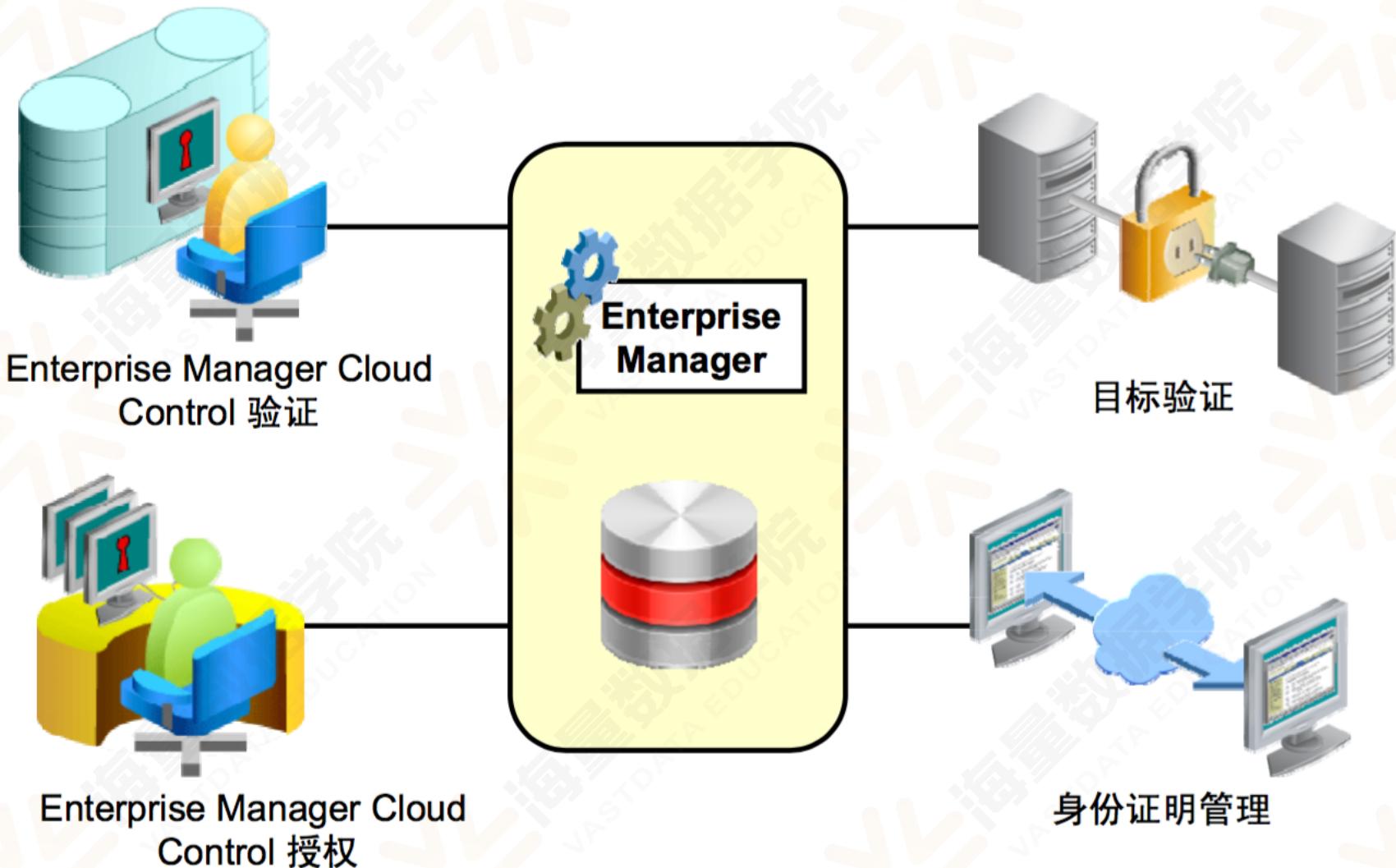
- 将任意页面加入收藏夹，以实现快速访问。

The screenshot displays the Oracle Enterprise Manager Cloud Control 12c interface. At the top, there's a navigation bar with links for Setup, Help, SYSMAN_VIEW, and Log Out. Below the navigation bar is a search bar for 'Search Target Name'.

The main area features a dashboard with various metrics and charts. On the left, a sidebar menu includes links for Enterprise, Targets, Favorites, History, Summary, Monitoring, Job, Reports, Configuration, Compliance, Provisioning and Patching, Quality Management, My Oracle Support, Infrastructure Cloud, Database Cloud, Chargeback, and Consolidation Planner. A pie chart in the center shows the status distribution: 40% Up, 12% Down, 1% Pending, and 1% Error.

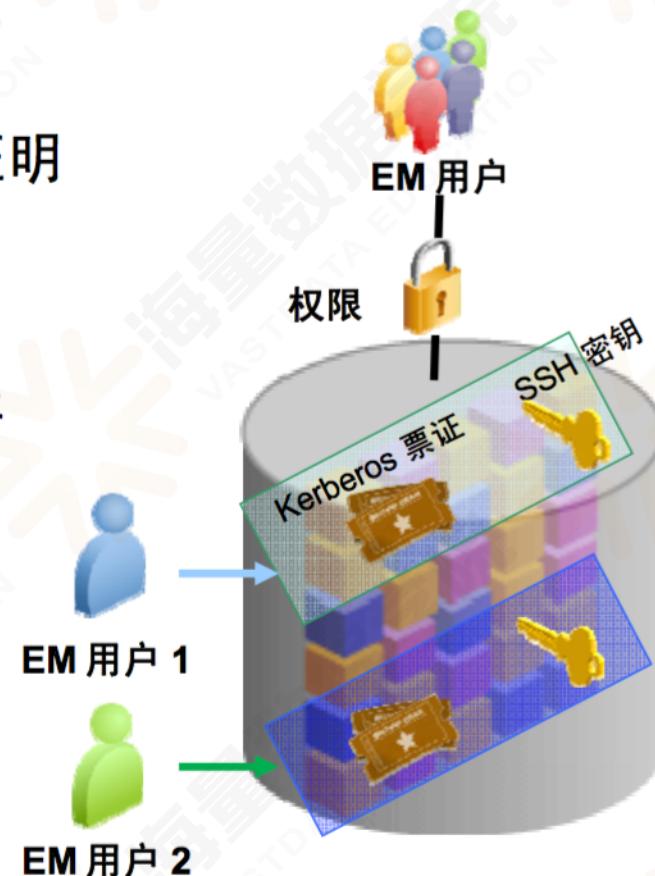
On the right, there are two main report sections: 'Inventory and Usage' and 'Compliance Summary'. The 'Inventory and Usage' section shows a table of platforms with their host counts and OS patch levels. The 'Compliance Summary' section shows a table of compliance frameworks with their target evaluations and violation counts.

安全性：概览



使用身份证明安全地进行管理

- 集中存储身份证明以易于管理
- 支持管理无口令和强验证身份证明
 - Kerberos 票证
 - SSH 密钥
- 在用户间重用和共享（而不公开身份证明的敏感内容）
- 受控制和受保护的访问
- 支持 sudo/powerbroker



区分身份证明

- 命名的身份证明
- 首选身份证明
- 默认身份证明
- 访问级别：
 - **View**（查看）：访问以使用身份证明
 - **Edit**（编辑）：更改身份证明（包括更改其名称和口令）
 - **Full**（完全）：完整访问权限（包括删除身份证明的能力）
- 使用分类：作业、集合和监视

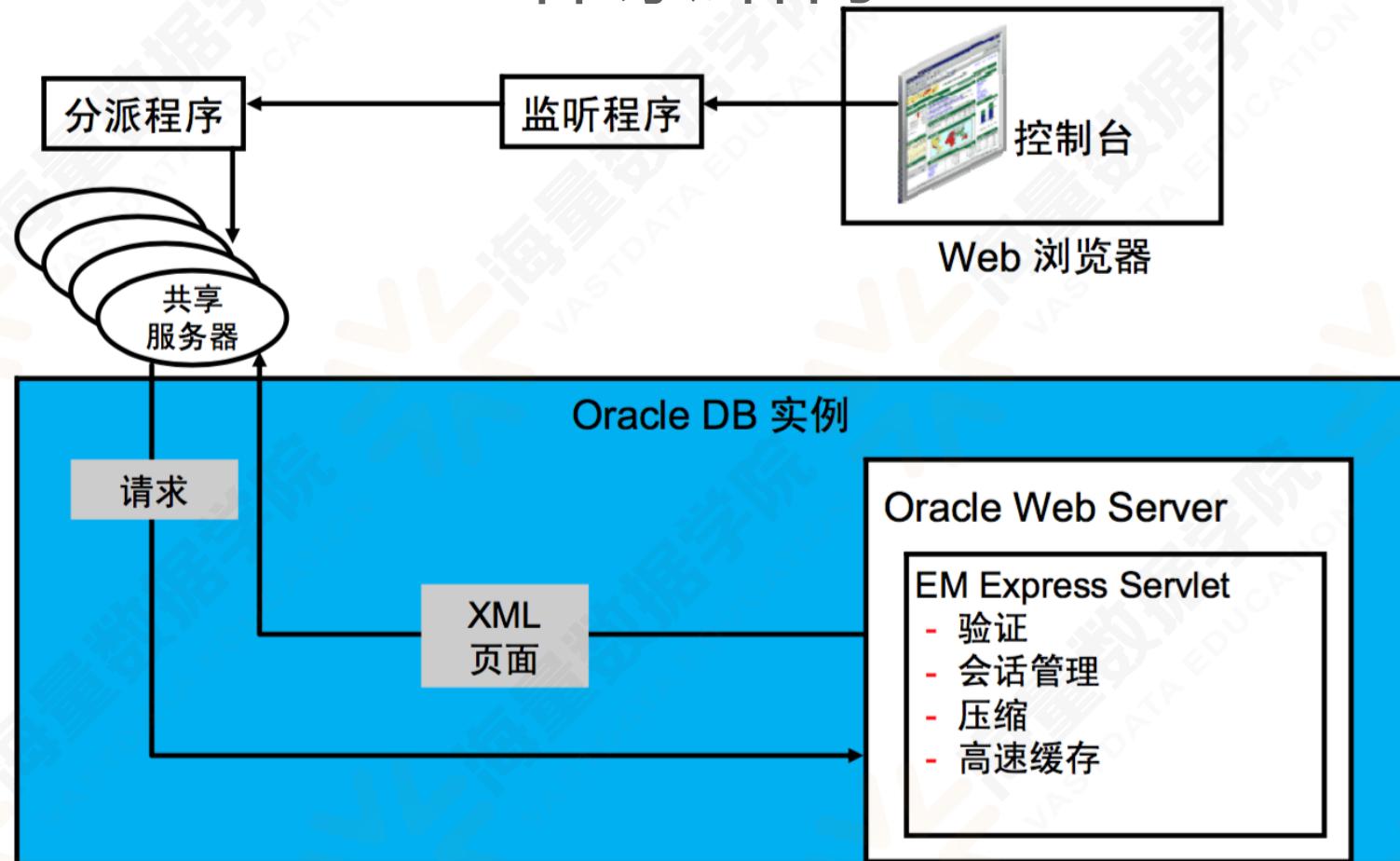
测验

可以使用 Enterprise Manager Cloud Control 管理哪些目标？

- a. 主机
- b. 数据库
- c. 应用程序服务器
- d. Web 应用程序
- e. OMS 和 OMR
- f. 以上所有项

Enterprise Manager Database Express

体系结构



配置 EM Database Express

为每个数据库实例配置 HTTP 或 HTTPS 监听程序端口：

- 验证 DISPATCHERS 参数。

```
dispatchers=(PROTOCOL=TCP) (SERVICE=sampleXDB)
```

- 使用 DBMS_XDB_CONFIG.setHTTPsPort 过程。

```
exec DBMS_XDB_CONFIG.setHTTPsPort(5500)
```

- 通过以下 URL 连接到 EM Database Express 控制台：

```
https://hostname:5500/em
```

- 对每个实例使用一个不同的端口。
- 浏览器需要 Flash 插件。

主页

ORACLE Enterprise Manager Database Express 12c

V1 (12.1.0.2.0 CDB RAC) Configuration Storage Security Performance

Help SYS Log Out slc00erz

Database Home

Status

Up Time 2 days, 8 hours, 46 minutes, 20 seconds
Type RAC - 2 instance(s) up
CDB (7 PDBs)
Version 12.1.0.2.0 Enterprise Edition
Database Name V1
Platform Name Linux x86 64-bit
Archiver Stopped

Incidents - Last 24 Hours

In...	Time	In...	Pr...	Error
1	Mon Fe...	26790	1	internal...
1	Mon Fe...	26789	1	internal...
1	Mon Fe...	26788	1	internal...
1	Mon Fe...	26787	1	internal...
1	Mon Fe...	26786	1	internal...
1	Mon Fe...	26785	1	internal...
1	Mon Fe...	26784	1	internal...

Running Jobs

I...	Cont...	O...	Na...	E...	Started
No Running Jobs					

Performance

Activity Class Services Instances Containers

SALES_PDB, HR_PDB, PDB\$SEED, NON_CDB_PLUG, CDB1_PDB2, CDB1_PDB1, CDB\$ROOT

Resources

Host CPU

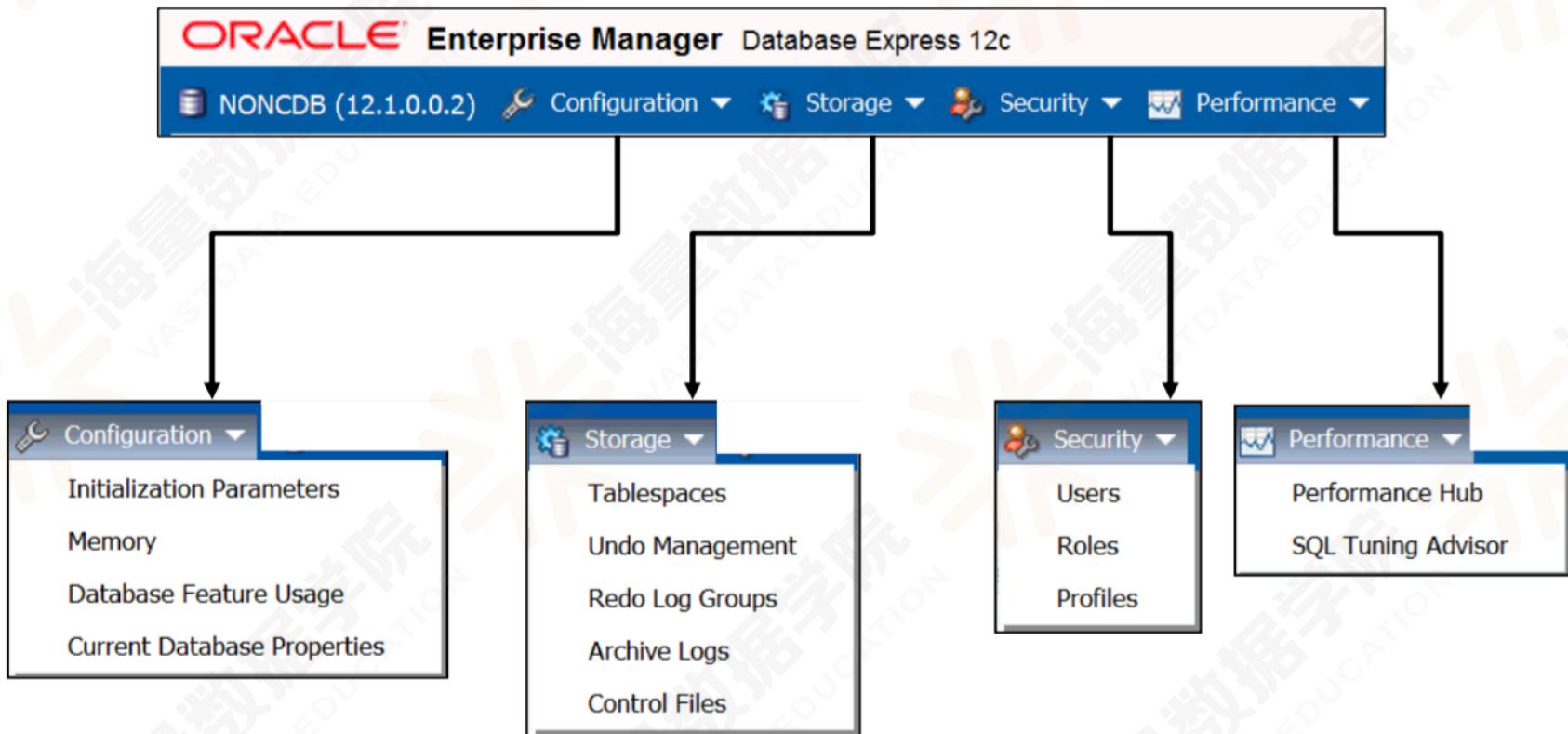
Active Sessions

Memory

SQL Monitor - Last Hour (20 max)

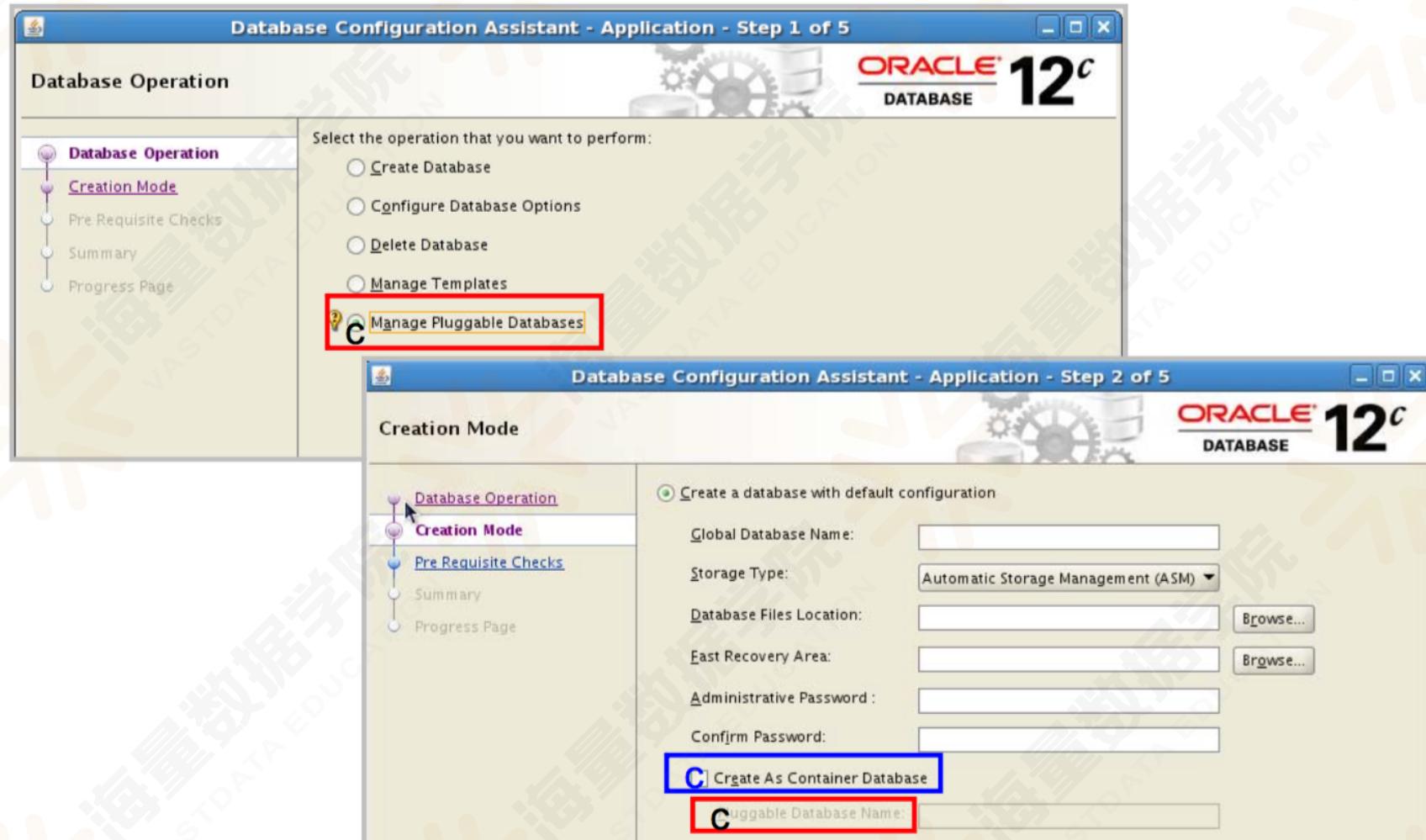
Status	Duration	T...	ID	User	Parallel	Database Ti...	SQL Te
Running	1.0s	SQL	ctv199sqz0wdg	4	0.2s	SELECT	
Running	12.0s	DBOP_DEMO2	6xb8cvj2y9dsr	2	11.6s	SELECT	
Running	12.0s	SQL	f0yr07vv8h96p	2	11.6s	DECLA	
Running	20.0s	DBOP_DEMO2	

菜单



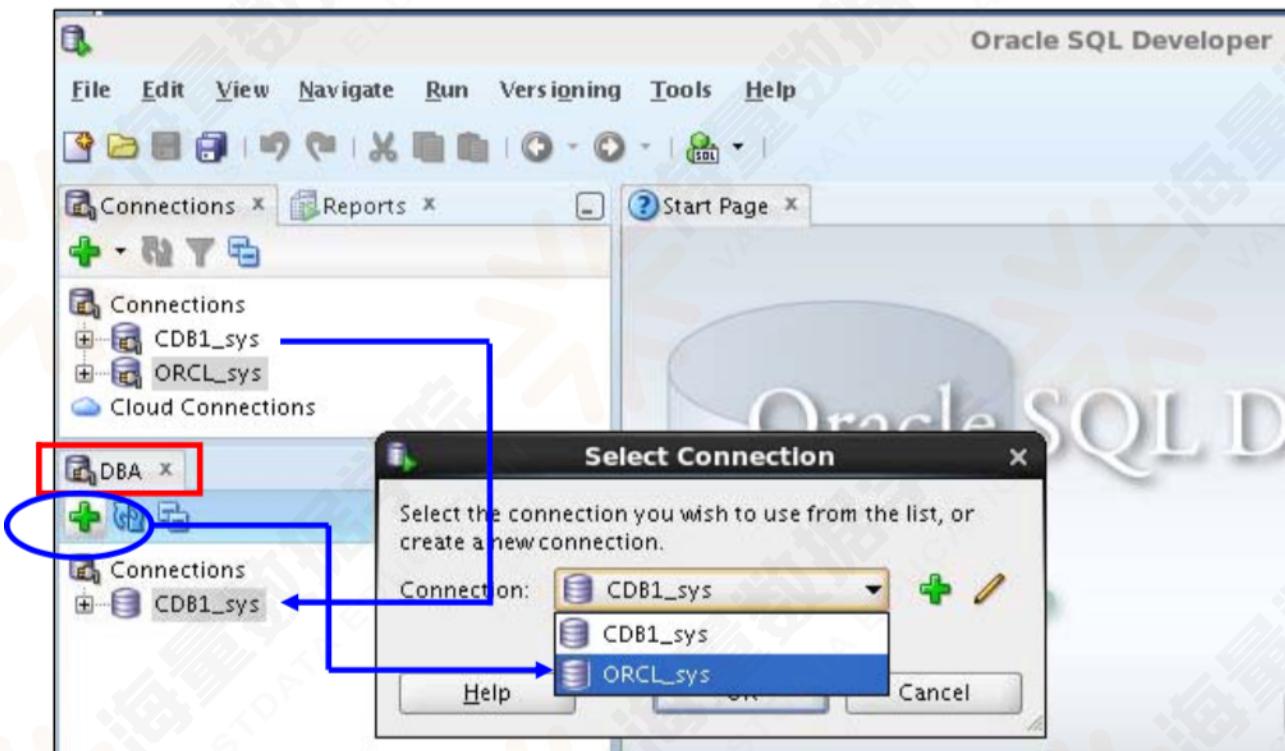
Database Configuration Assistant

多租户容器数据库和可插入数据库



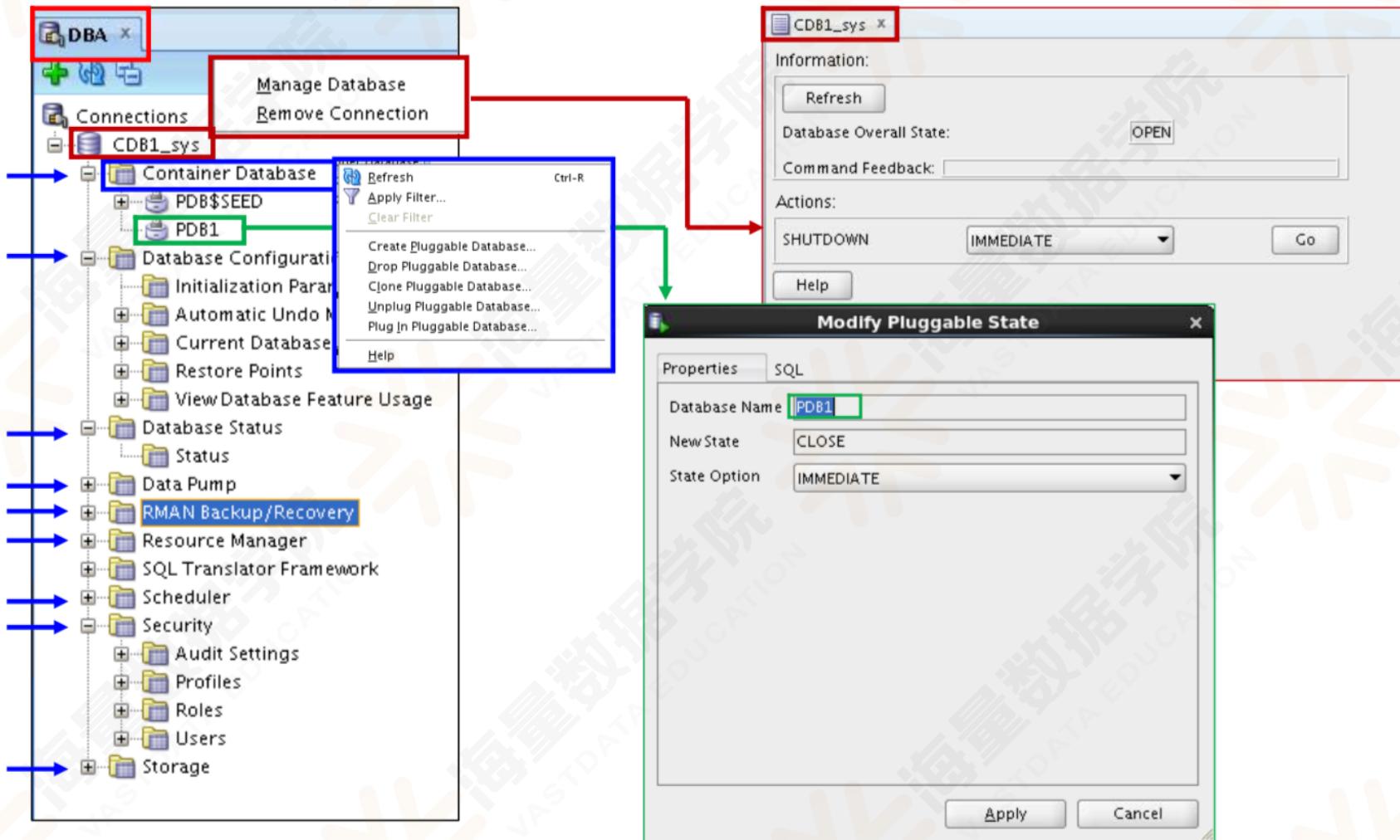
Oracle SQL Developer: 连接

使用 DBA 连接在 DBA 导航器中执行 DBA 操作:



Oracle SQL Developer: DBA 操作

通过 DBA 导航器执行 DBA 任务:

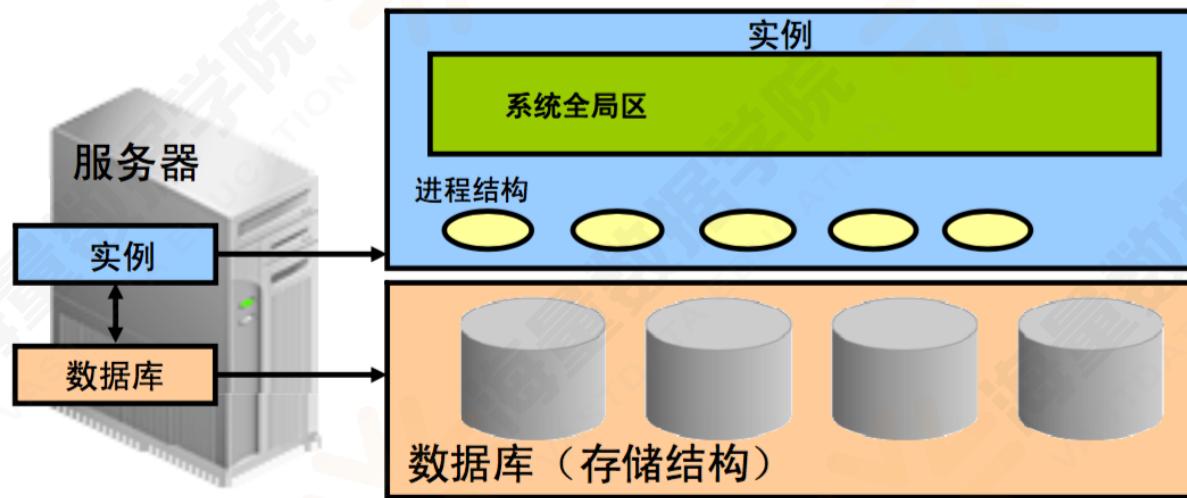


难题

许多 Oracle 客户都有大量构建于 Oracle RDBMS 上的“部门”应用程序。它们：

- 对其部署硬件的使用不占多大百分比
- 具有实例和存储空间开销，从而无法在同一物理和存储服务器上放置大量“部门”数据库
- 不是非常复杂，不需要一个全职管理员完全专注于它们
- 不需要大量时间来对所有应用程序打补丁或升级

11g 发行版 2 的 Oracle DB



多个非 CDB 不共享任何项：

- 过多后台进程
- 高共享/进程内存
- Oracle 元数据的许多副本

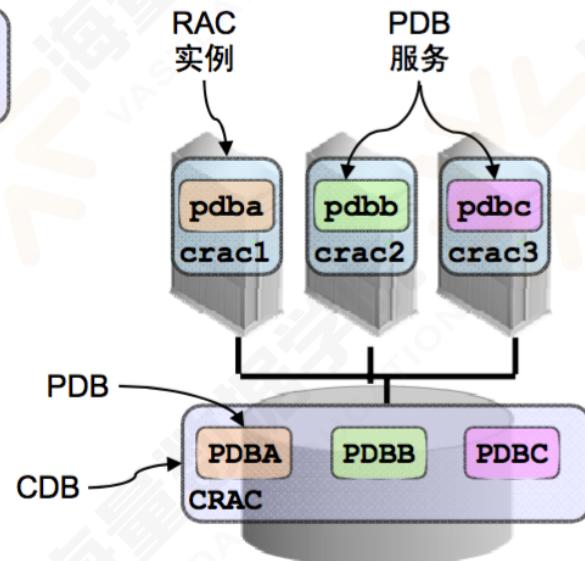
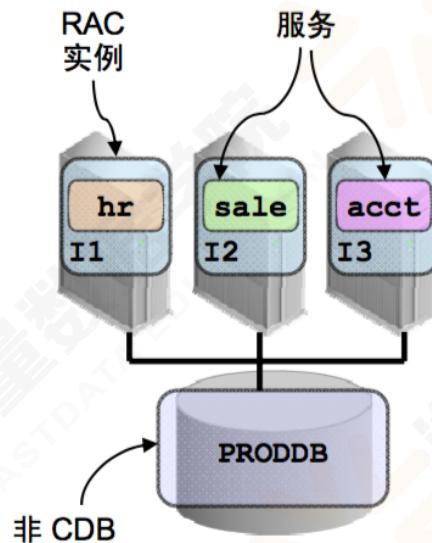
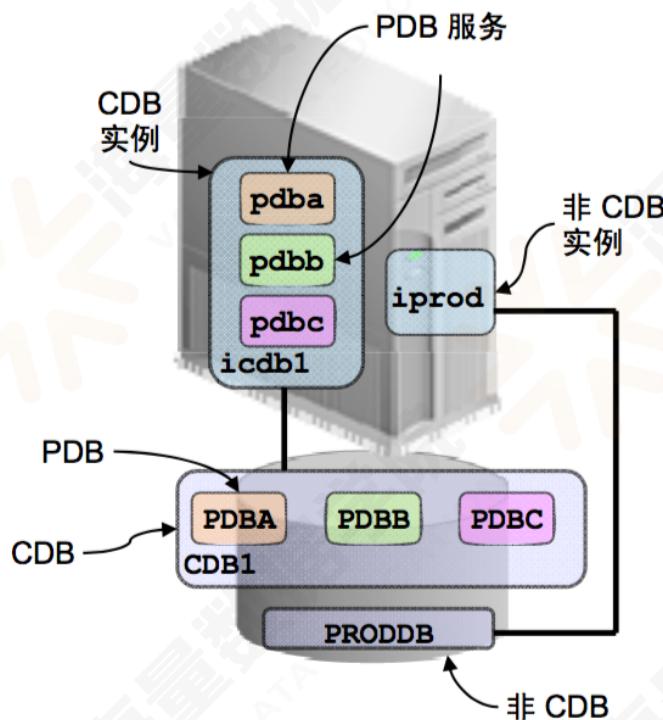
新的多租户体系结构：优点

- 以更低成本在集中管理的平台中操作多个数据库：
 - 实例开销较低
 - 存储成本较低
- 降低 DBA 资源成本并维护安全性
 - 无应用程序更改
 - 快速且易于预配
 - 节约打补丁和升级的时间
 - 将以下各项的职责分离开来：
 - 不同的应用程序管理员
 - 应用程序管理员和 DBA
 - 应用程序中的用户
- 提供隔离

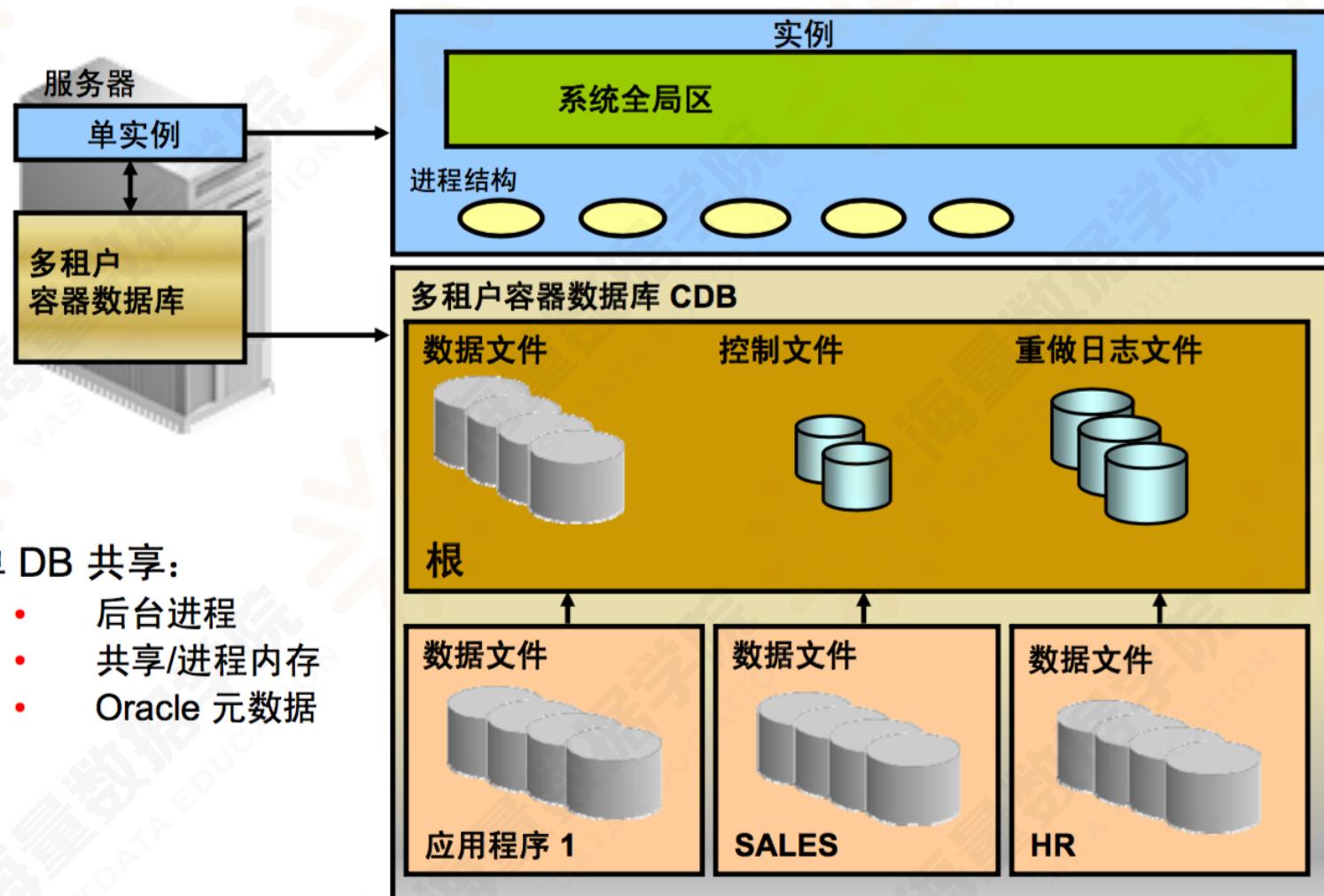
多租户体系结构的其他好处

- 确保与非 CDB 的完全向后兼容性
- 使用 RAC 和 Data Guard 进行完全操作
- 受到 Enterprise Manager 支持
- 与资源管理器集成
- 允许对多个数据库进行集中管理
 - 备份或灾难恢复
 - 打补丁和升级

配置



多租户容器数据库



单 DB 共享:

- 后台进程
- 共享/进程内存
- Oracle 元数据

原始安装

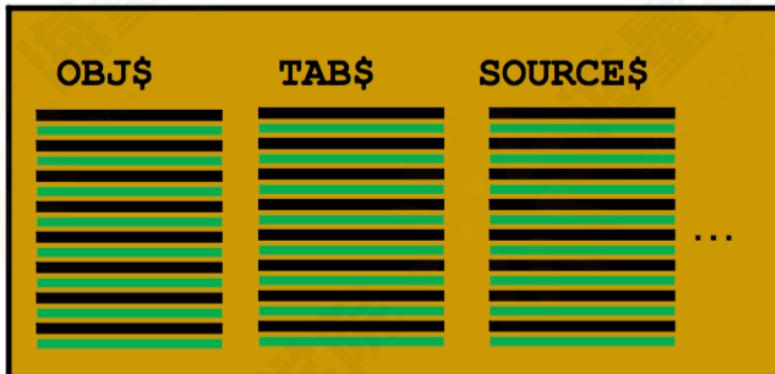
在初始数据库创建之后，唯一的对象是 Oracle 提供的对象

OBJ\$	TAB\$	SOURCE\$...

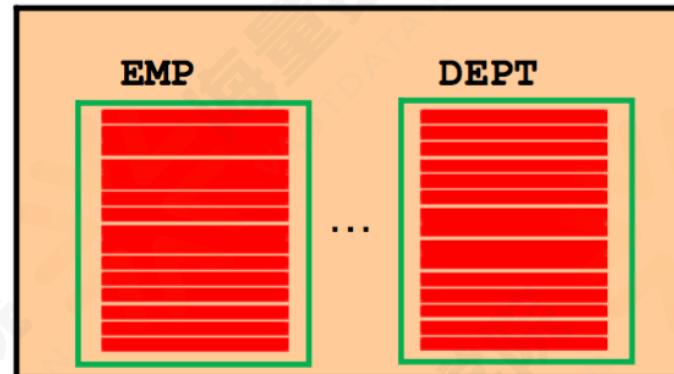
Oracle 系统数据

添加用户数据

在非 CDB 中，将添加用户数据。元数据将与数据字典中 Oracle 提供的数据混合存储。

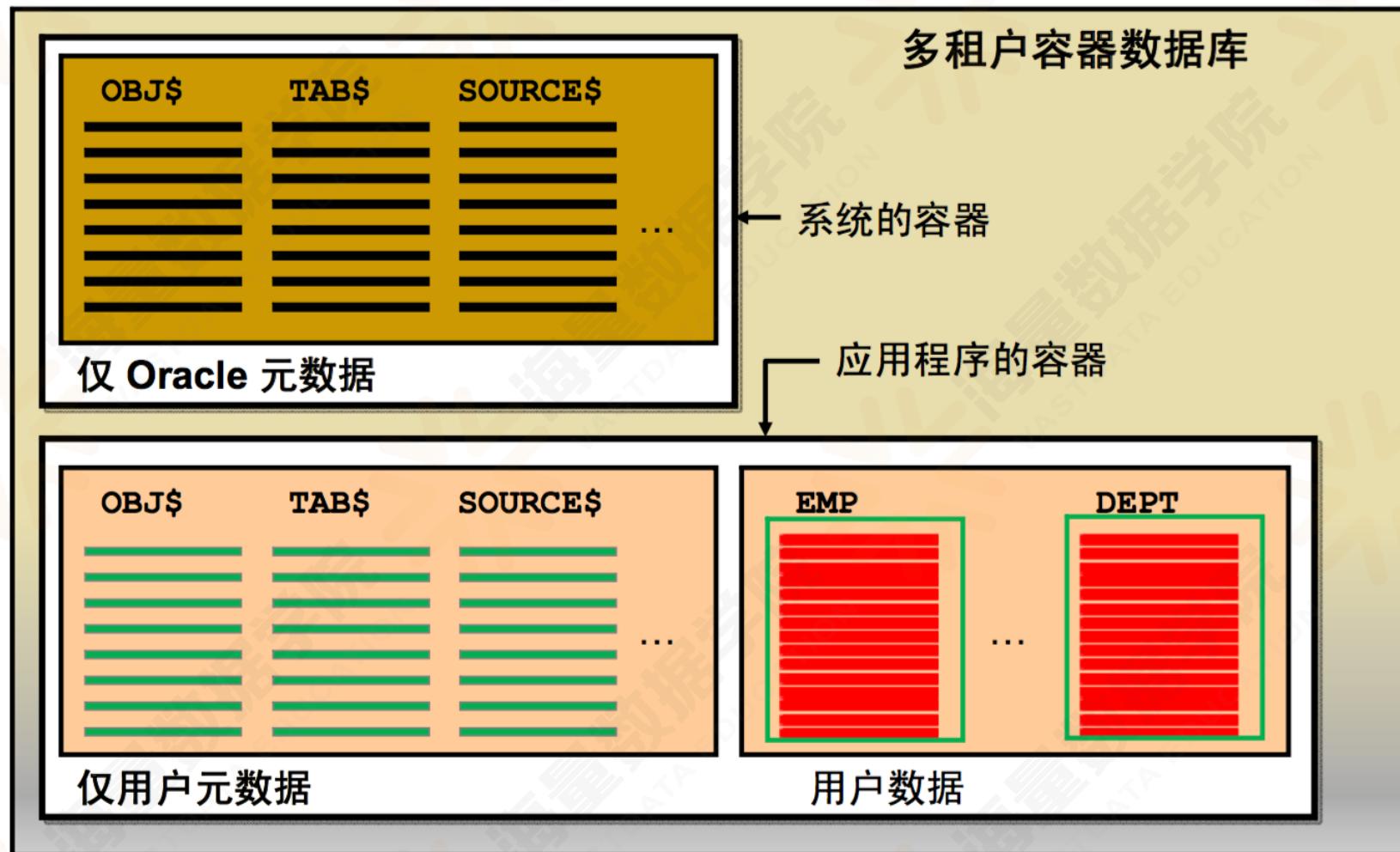


Oracle 系统数据与用户元数据混合

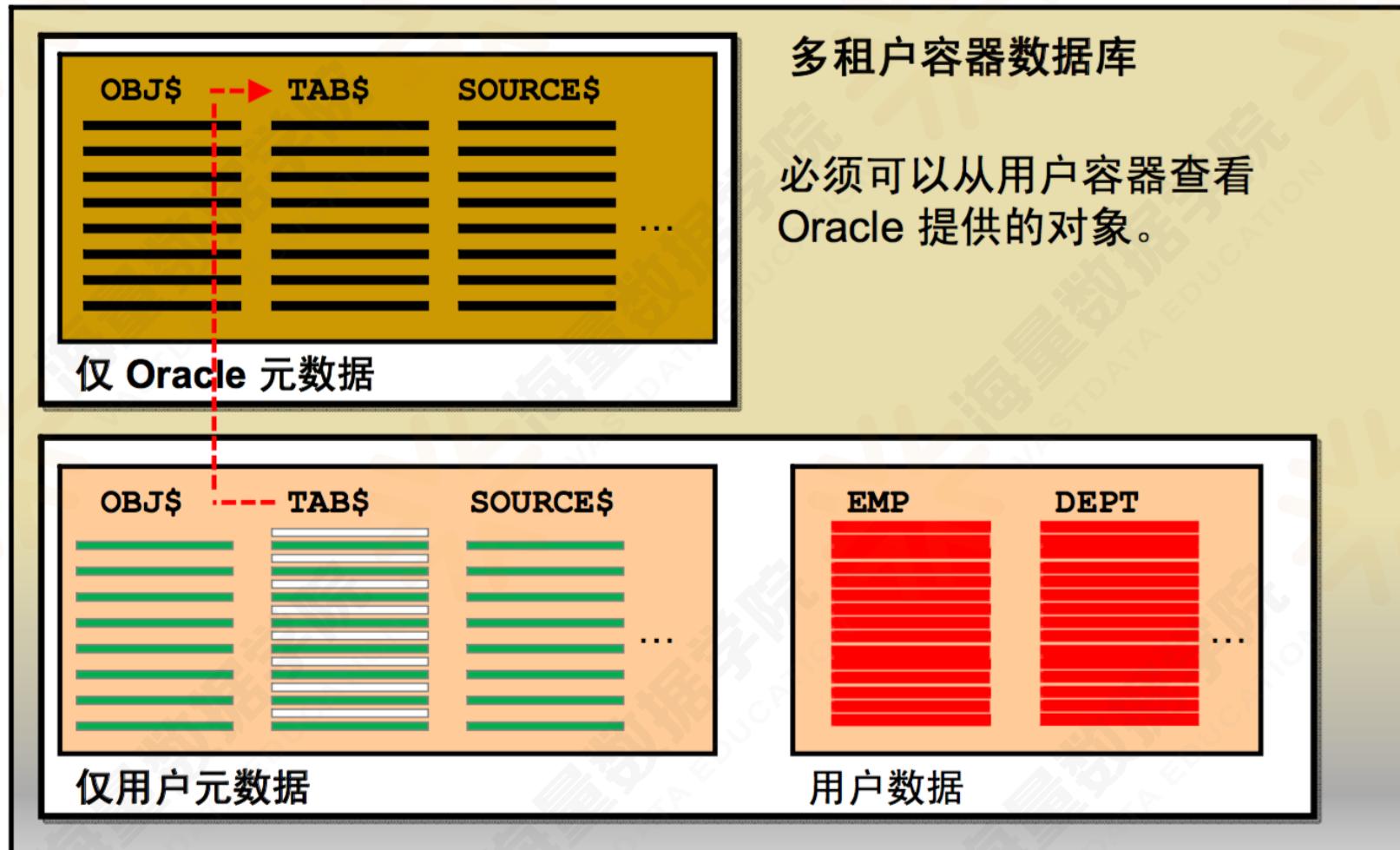


用户数据

分离 SYSTEM 和用户数据



USER 容器中的 SYSTEM 对象



命名容器

根

仅 Oracle 元数据

多租户容器数据库

- Oracle 提供的容器称为根容器。
- 用户容器是可插入数据库 (PDB)。

PDB

仅用户元数据

用户数据



海量数据
VASTDATA

海量数据学院
VASTDATA EDUCATION

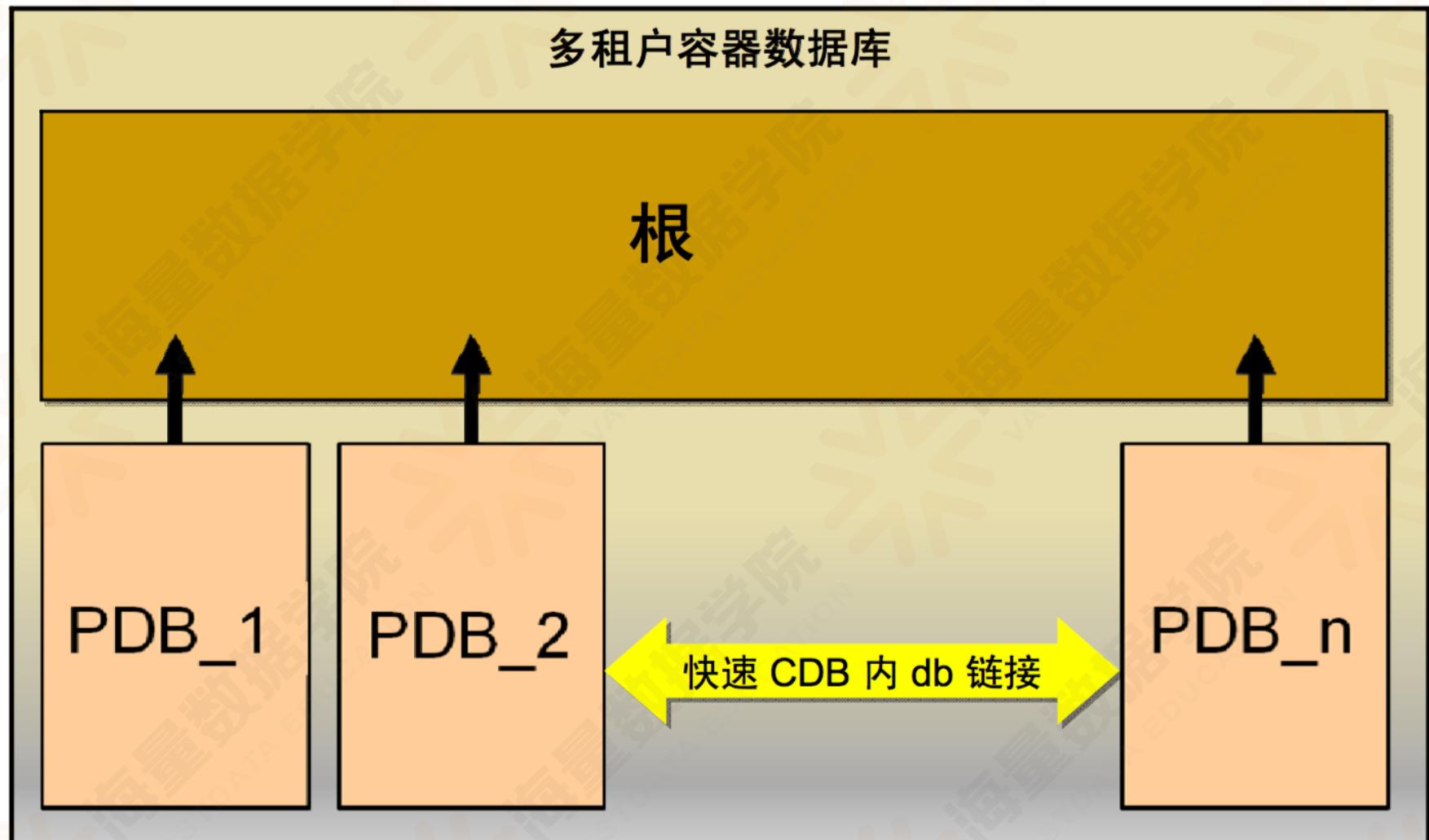
预配可插入数据库



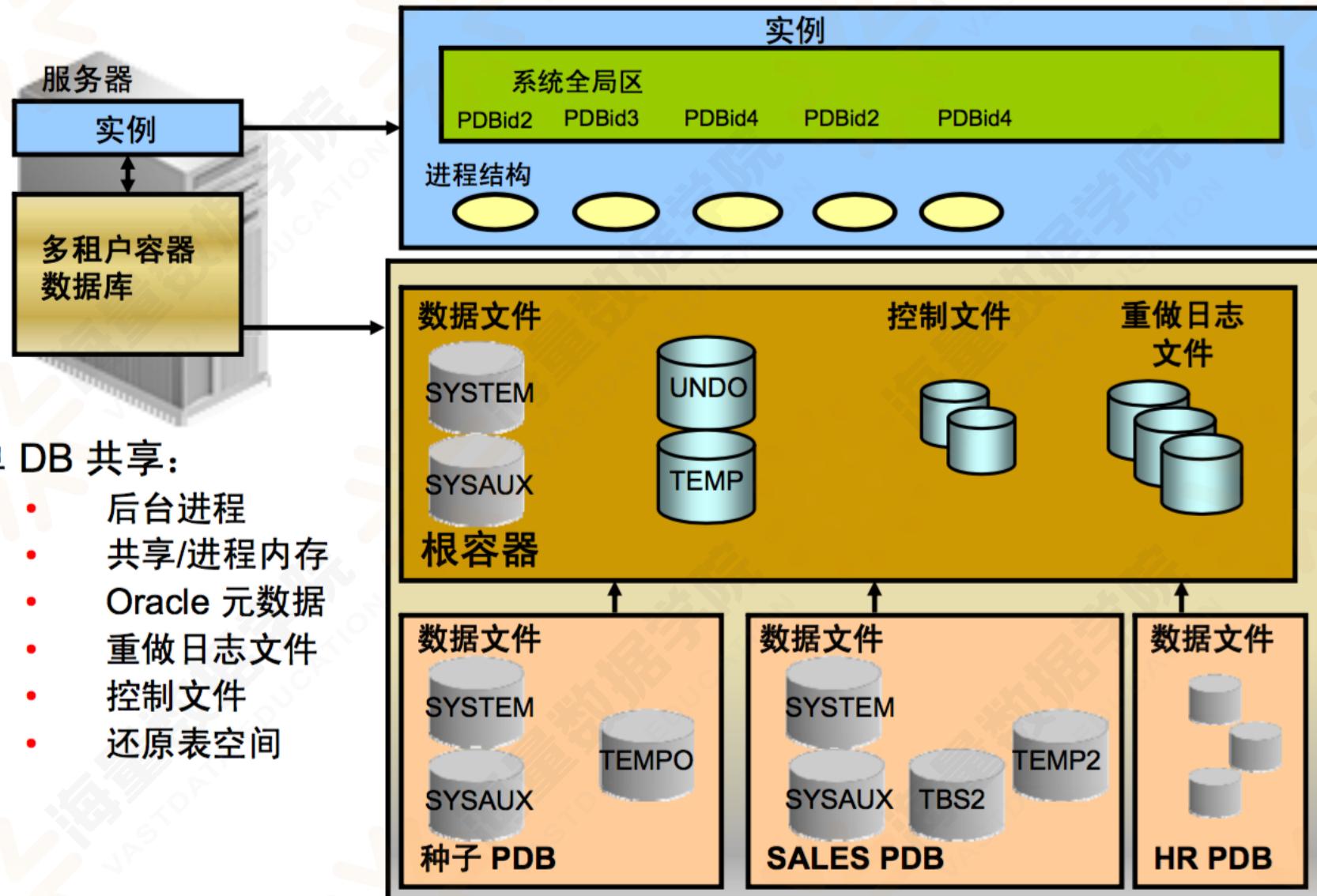
多种方法:

- 从 PDB\$SEED 可插入数据库创建 PDB。
- 插入非 CDB。
- 将非 CDB 克隆到 CDB 中，或者将 PDB 克隆到同一或其他 CDB 中。
- 将移走的 PDB 插入 CDB 中。

在多租户容器数据库内互操作



多租户容器数据库体系结构



容器

V\$CONTAINERS 中有两种类型的容器：

- 根容器：
 - 在创建 CDB 时创建的第一个容器
 - 必需
 - Oracle 系统提供的公用对象和元数据
 - Oracle 系统提供的公用用户和角色
- 可插入数据库容器 (PDB)：
 - 应用程序的容器：
 - 表空间（永久和临时）
 - 方案/对象/权限
 - 已创建/已克隆/已移走/已插入
 - 特定种子 PDB：
 - PDB\$SEED 提供新 PDB 的快速预配
 - 限制一个 CDB 中只能有 253 个 PDB（包括种子）
 - 限制一个 CDB 中只能有 1024 个服务



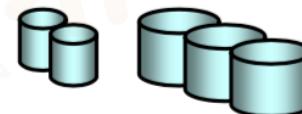


CDBA

问题：根与 PDB

什么属于 CDB 而不属于特定容器？

- 控制文件和重做日志文件



什么位于根中而不在 PDB 中？

- UNDO 表空间（对于每个实例）
- 系统提供的元数据
- 共享的 Oracle 提供的数据
 - PL/SQL Oracle 提供的程序包 (DBMS_SQL ...)
 - PDB 服务名称
- 提供 PDB 中的信息的 CDB 字典视图
- CDB 资源管理器计划



表 SYS.OBJ\$

NAME	TYPE
TAB\$	2
USER\$	2

表 SYS.SERVICE\$

NAME
PDB_SALES
PDB_HR

视图 CDB_xxx

TABLE_NAME	CON_ID
EMPLOYEES	1
TEST	2



问题：PDB 与根

什么位于某个 PDB 中而不在根中，也不在另一 PDB 中？

PDBA

- 应用程序表空间
- 本地临时表空间
- 本地用户 和本地角色



- 非共享的本地元数据

NAME	TYPE
EMPLOYEES	2
JOB\$	2

表 SYS.OBJ\$

- 不与其他 PDB 共享的应用程序数据

表 HR.EMPLOYEES

EMP_NAME
SMITH
JOHN

- PDB 资源管理器计划

术语

- 公用与本地：
 - 用户
 - 角色
 - 权限
- CDB 与 PDB 级别：
 - CDB 资源管理器计划与 PDB RM 计划
 - CDB 或 PDB 级别的统一审计
 - CDB 或 PDB 级别的 XStream

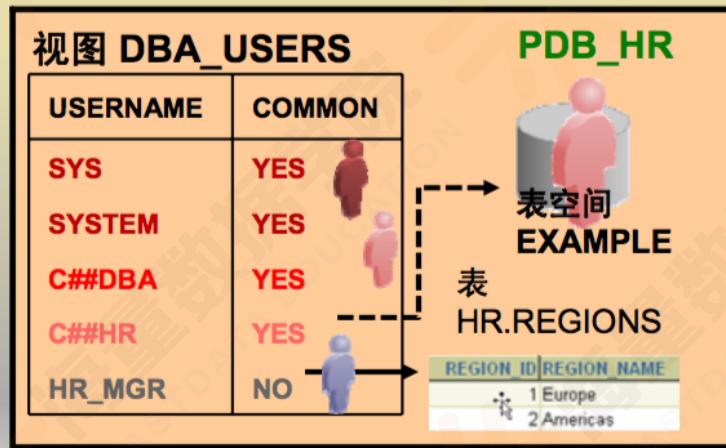
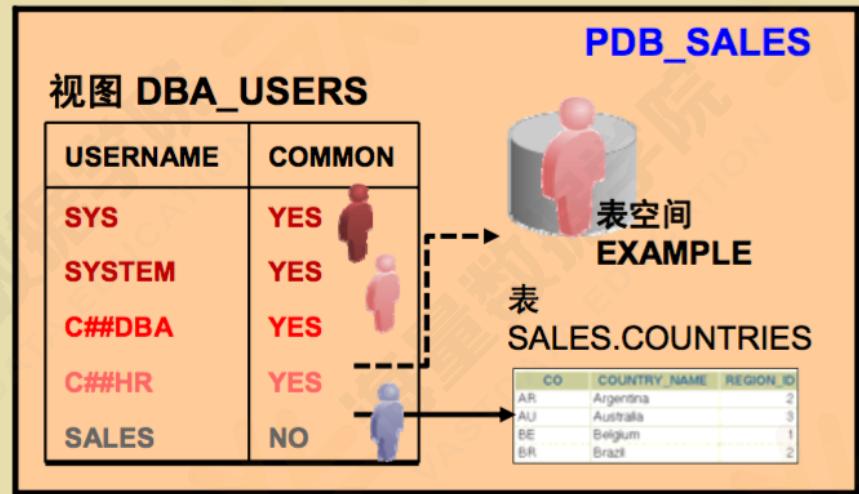
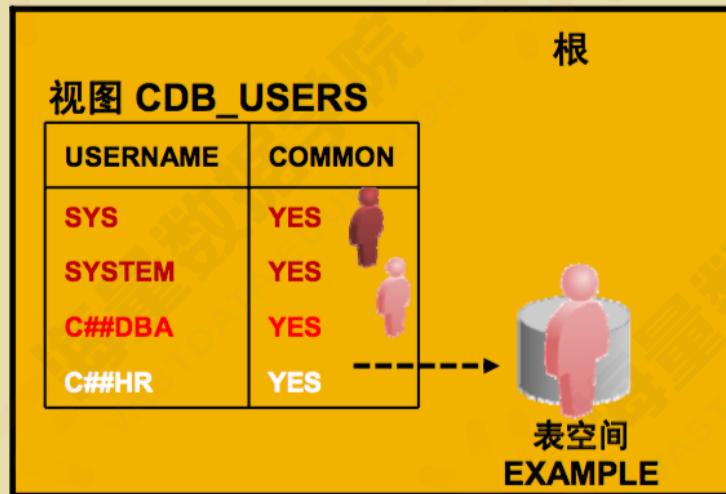


海量数据
VASTDATA

海量数据学院
VASTDATA EDUCATION

公用用户和本地用户

多租户容器数据库

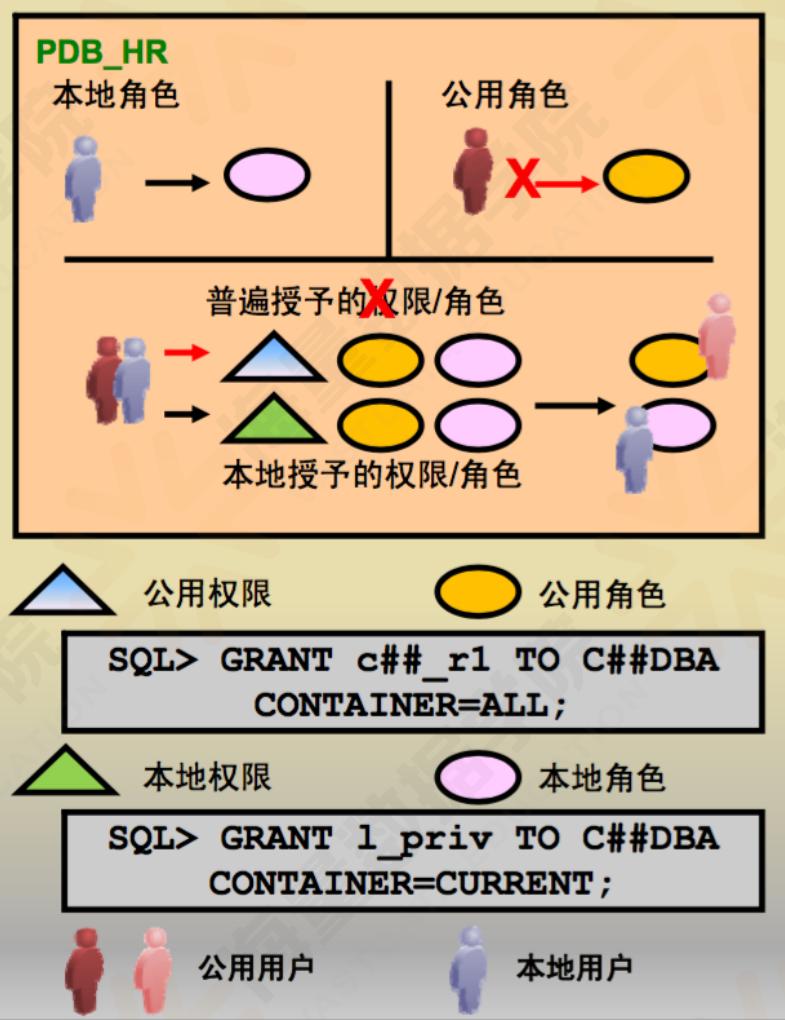
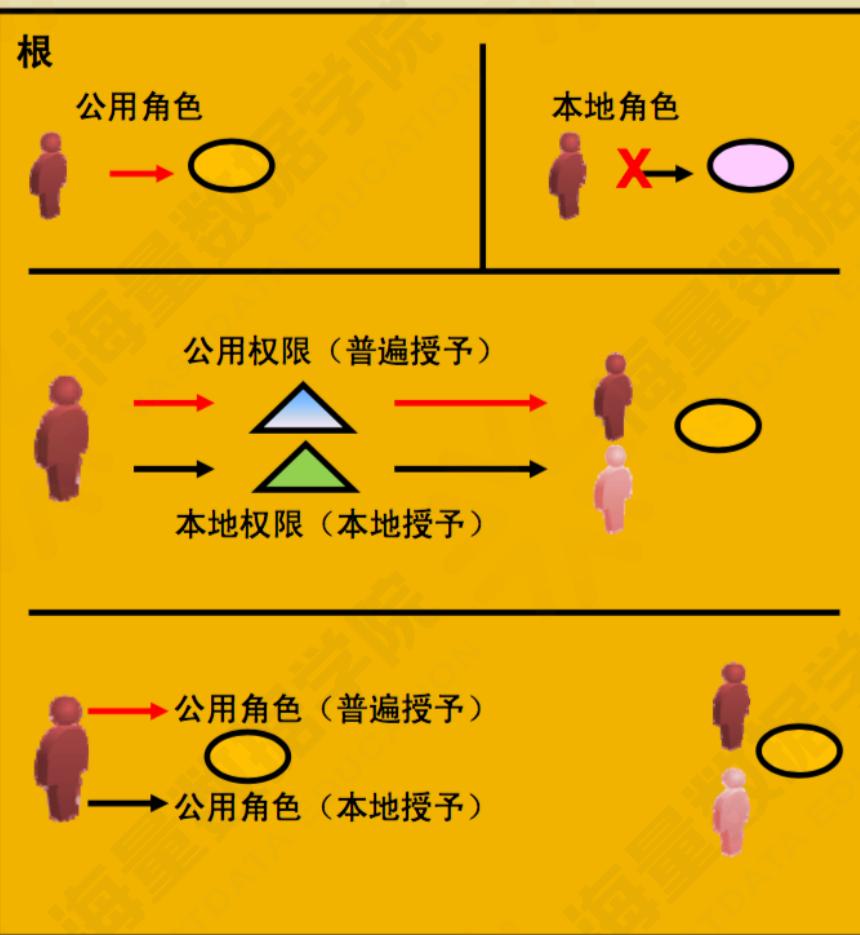


添加公用用户包括在根和每个 PDB 中添加该用户的说明。

本地用户是传统用户，仅在自己的 PDB 中为已知。

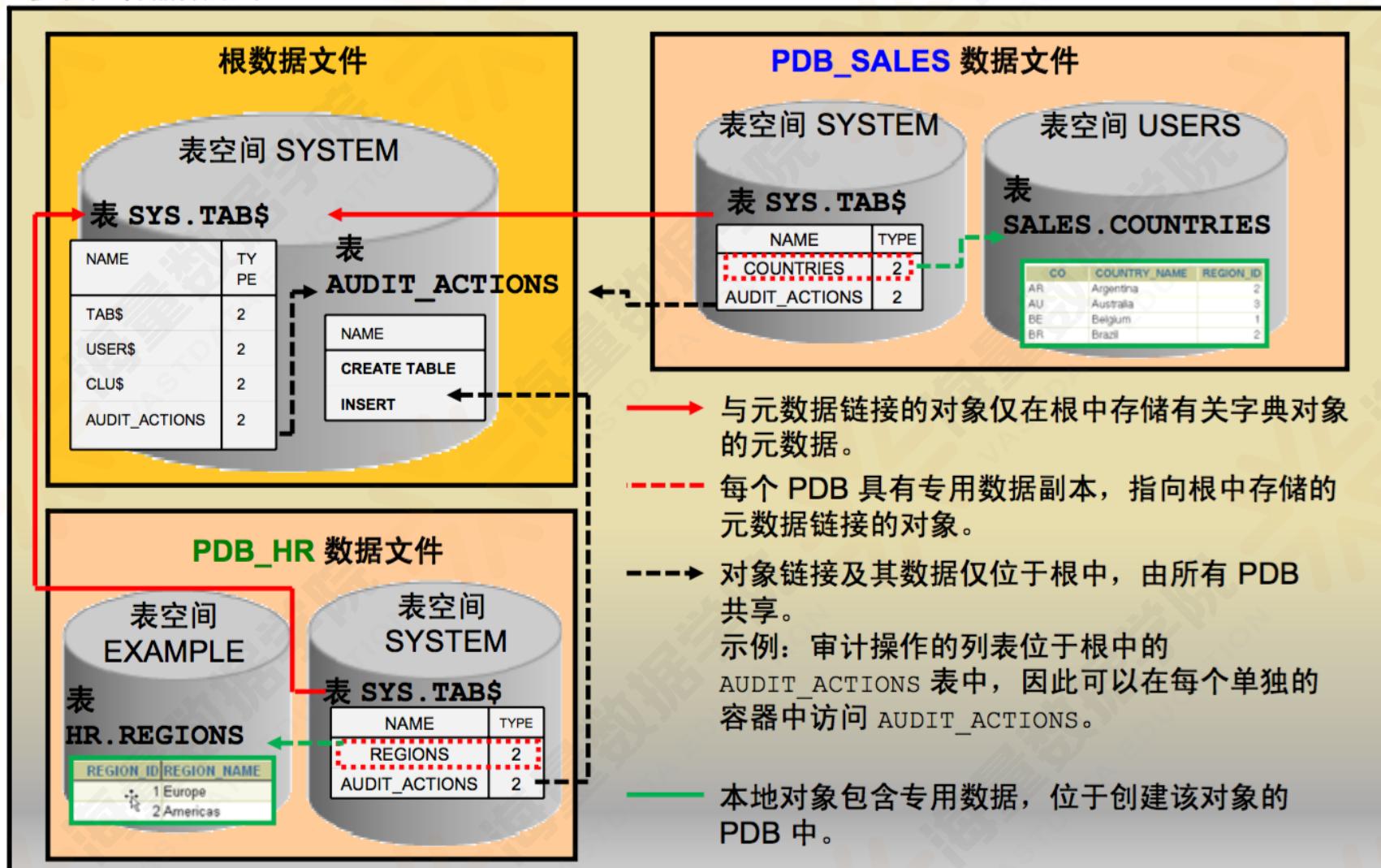
公用和本地权限和角色

多租户容器数据库



共享和非共享对象

多租户容器数据库



数据字典视图

CDB_xxx 所有 PDB 中多租户容器数据库中的所有对象

DBA_xxx 容器或可插入数据库中的所有对象

ALL_xxx 可由当前用户访问的对象

USER_xxx 当前用户拥有的对象

```
SQL> SELECT view_name FROM dba_views WHERE view_name like 'CDB%';
```

- CDB_pdbs: CDB 内的所有 PDB
- CDB_tablespaces: CDB 内的所有表空间
- CDB_users: CDB 内的所有用户（公用和本地）

提供 PDB 内信息的 DBA 字典视图：

```
SQL> SELECT table_name FROM dict WHERE table_name like 'DBA%';
```

影响

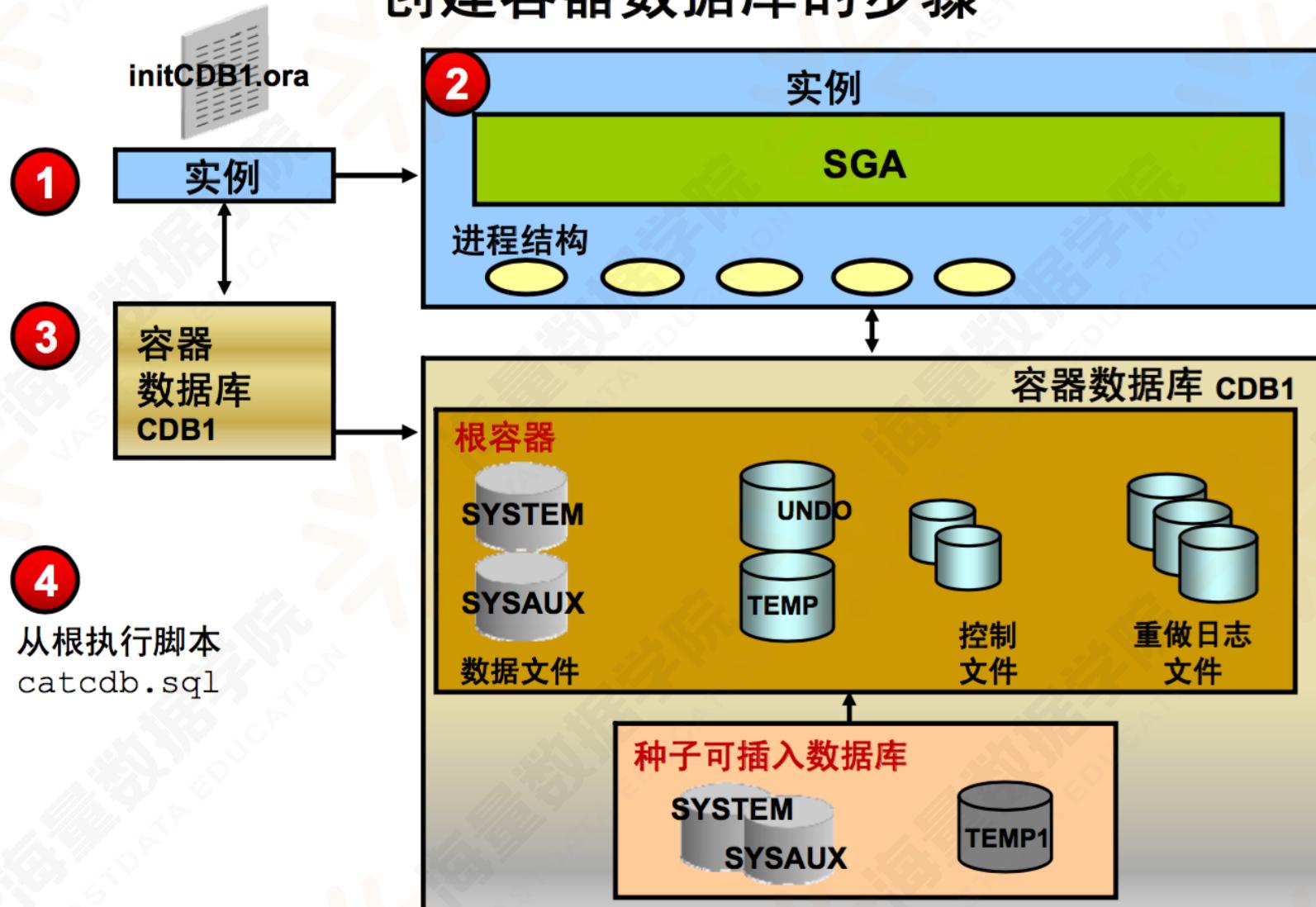
- 对所有 PDB 使用一种字符集（建议使用 Unicode）
- 多个 PDB 初始化参数，一个 SPFILE
- 无 PDB 限定数据库对象名称
 - ~~SELECT * FROM HR.apps.tab1~~
 - 使用 DB 链接: `SELECT * FROM apps.tab1@HR`
- Oracle Data Guard 位于 CDB 级别
- 仅每个 PDB 具有 Oracle Database Vault
- 每个 PDB 一个主密钥来加密 PDB 数据
- CDB 和 PDB 级别的统一审计
- Oracle Scheduler
- Oracle GoldenGate
- Oracle Streams
- CDB 和 PDB 级别的 Oracle XStream



工具

	SQL*Plus	OUI	DBCA	EM Cloud Control	EM Database Express	SQL Developer	DBUA
创建新的 CDB 或 PDB	是	是	是	是（仅限 PDB）	是（仅限 PDB）	是（仅限 PDB）	
了解 CDB 实例、体系结构和 PDB	是			是	是	是	
将 12.1 CDB 升级到 12.x CDB				是			是

创建容器数据库的步骤



创建容器数据库：使用 SQL*Plus

1. 实例启动：

- a. 设置 ORACLE_SID=CDB1
- b. 在 initCDB1.ora 中做如下设置：
 - 将 CONTROL_FILES 设置为 CDB 控制文件名。
 - 将 DB_NAME 设置为 CDB 名。
 - 将 ENABLE_PLUGGABLE_DATABASE 设置为 TRUE。

```
SQL> CONNECT / AS SYSDBA  
SQL> STARTUP NOMOUNT
```

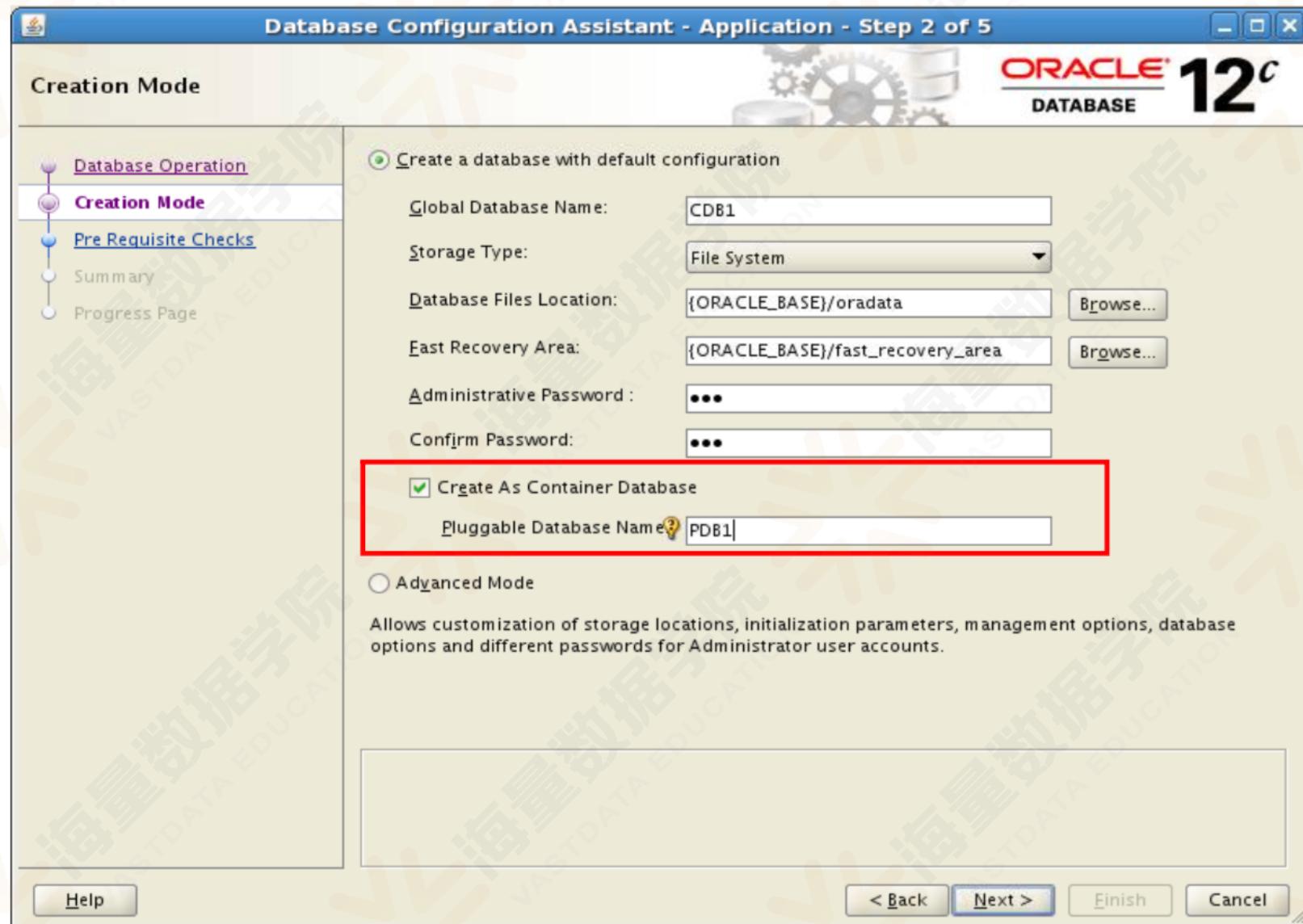
2. 创建数据库：

```
SQL> CREATE DATABASE CDB1 ENABLE PLUGGABLE DATABASE ...  
SEED FILE_NAME_CONVERT ('/oracle/dbs','/oracle/seed');
```

- CDB\$ROOT 容器
- PDB\$SEED 可插入数据库

3. 运行 catcdb.sql 脚本。

创建容器数据库：使用 DBCA



新子句：SEED FILE_NAME_CONVERT

CREATE DATABASE 新子句：

```
SQL> CREATE DATABASE cdb1
  USER SYS IDENTIFIED BY p1 USER SYSTEM IDENTIFIED BY p2
  LOGFILE GROUP 1 ('/u01/app/oradata/CDB1/redo1a.log',
                    '/u02/app/oradata/CDB1/redo1b.log') SIZE 100M,
  GROUP 2 ('/u01/app/oradata/CDB1/redo2a.log',
            '/u02/app/oradata/CDB1/redo2b.log') SIZE 100M
  CHARACTER SET AL32UTF8 NATIONAL CHARACTER SET AL16UTF16
  EXTENT MANAGEMENT LOCAL DATAFILE
            '/u01/app/oradata/CDB1/system01.dbf' SIZE 325M
  SYSAUX DATAFILE '/u01/app/oradata/CDB1/sysaux01.dbf' SIZE 325M
  DEFAULT TEMPORARY TABLESPACE tempts1
            TEMPFILE '/u01/app/oradata/CDB1/temp01.dbf' SIZE 20M
  UNDO TABLESPACE undotbs
            DATAFILE '/u01/app/oradata/CDB1/undotbs01.dbf' SIZE 200M
  ENABLE PLUGGABLE DATABASE
  SEED FILE_NAME_CONVERT =
    ('/u01/app/oradata/CDB1',
     '/u01/app/oradata/CDB1/seed');
```

新子句：ENABLE PLUGGABLE DATABASE

无 SEED FILE_NAME_CONVERT:

- OMF: DB_CREATE_FILE_DEST='/u01/app/oradata'
- 或新实例参数:

PDB_FILE_NAME_CONVERT =

'/u01/app/oradata/CDB1', '/u01/app/oradata/seed'

```
SQL> CONNECT / AS SYSDBA
SQL> STARTUP NOMOUNT

SQL> CREATE DATABASE cdb2
      USER SYS IDENTIFIED BY p1 USER SYSTEM IDENTIFIED BY p2
      EXTENT MANAGEMENT LOCAL
      DEFAULT TEMPORARY TABLESPACE temp
      UNDO TABLESPACE undotbs
      DEFAULT TABLESPACE users
      ENABLE PLUGGABLE DATABASE;
```

创建 CDB 后：CDB 的新特性

与非 CDB 相比，CDB 具有一些新特性：

- **两个容器：**
 - **根** (CDB\$ROOT)
 - **种子 PDB** (PDB\$SEED)
- **若干服务：** 每个容器一项服务
 - 根服务名 = CDB 名 (cdb1)
- **根和种子中的公用用户：** SYS、SYSTEM ...
- **向公用用户授予公用权限**
- **预定义的公用角色**
- **与每个容器相关联的表空间和数据文件：**
 - **根：** SYSTEM (系统提供的元数据和无用户数据) 、
SYSAUX、 TEMP
 - **种子：** SYSTEM、 SYSAUX、 TEMP

数据字典视图: DBA_xxx

DBA_xxx 根或可插入数据库中的所有对象

ALL_xxx 可供 PDB 中的当前用户访问的对象

USER_xxx PDB 中的当前用户拥有的对象

提供 PDB 内信息的 DBA 字典视图:

```
SQL> SELECT table_name FROM dict WHERE table_name like 'DBA%';
```

- DBA_tablespaces: PDB 的所有表空间
- DBA_data_files: PDB 的所有数据文件
- DBA_tables: PDB 中的所有表
- DBA_users: PDB 的所有公用和本地用户



数据字典视图：CDB_XXX

CDB_XXX 容器数据库中的所有对象（新列 CON_ID）

DBA_XXX 根或可插入数据库中的所有对象

ALL_XXX 可供 PDB 中的当前用户访问的对象

USER_XXX PDB 中的当前用户拥有的对象

CDB 字典视图提供各个 PDB 的信息：

```
SQL> SELECT view_name FROM dba_views WHERE view_name like 'CDB%';
```

- CDB_pdbs: CDB 中的所有 PDB
- CDB tablespaces: CDB 中的所有表空间
- CDB_data_files: CDB 中的所有数据文件
- CDB_users: CDB 中的所有用户（公用和本地）

数据字典视图：示例

- 比较：

1

```
SQL> CONNECT / AS SYSDBA  
SQL> SELECT role, common, con_id FROM cdb_roles;
```

2

```
SQL> SELECT role, common FROM dba_roles;
```

3

```
SQL> CONNECT sys@PDB1 AS SYSDBA  
SQL> SELECT role, common, con_id FROM cdb_roles;
```

4

```
SQL> SELECT role, common FROM dba_roles;
```

- 访问可显示多个 PDB 中数据的 V\$ 或 GV\$ 视图中的数据时，可使用权限提高安全性。

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

NAME	OPEN_MODE
PDB\$SEED	READ ONLY
PDB1	READ WRITE
PDB2	READ WRITE



数据字典视图：V\$xxx 视图

SGA 供所有容器访问： V\$ 视图和 CON_ID 列

```
SQL> SELECT distinct status, con_id FROM v$bh order by 2;
```

STATUS	CON_ID	
cr	1	root
free	1	
xcur	1	
xcur	2	seed PDB
cr	3	PDB1 PDB
xcur	3	

```
SQL> select OBJECT_ID, ORACLE_USERNAME, LOCKED_MODE, CON_ID  
from V$LOCKED_OBJECT;
```

OBJECT_ID	ORACLE_USERNAME	LOCKED_MODE	CON_ID	
83711	SYS	3	3	PDB1 PDB
83710	DOM	3	4	PDB2 PDB

创建 CDB 后：待办事项列表

创建 CDB 后，CDBA 必须：

- 为根和每个 PDB 设置独立的默认表空间
- 为每个容器设置默认临时表空间
- 启动监听程序
- 插入非 CDB
- 测试启动/关闭过程
- 定义默认 PDB 状态，使 PDB 自动打开
- 创建备份和恢复过程

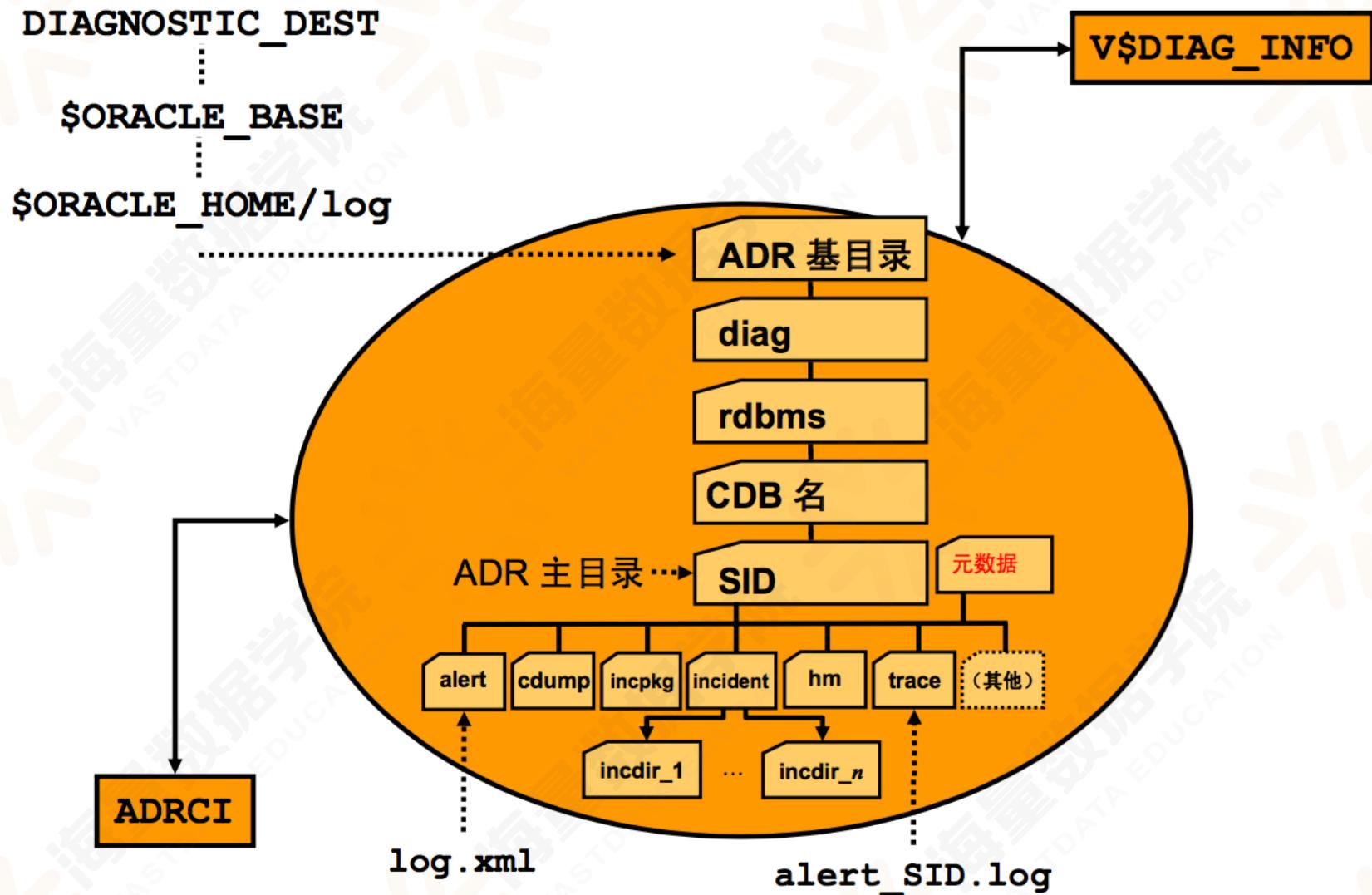
12.1.0.2



海量数据
VASTDATA

海量数据学院
VASTDATA EDUCATION

自动诊断资料档案库



自动诊断资料档案库：alert.log 文件

alert_CDB1.log 显示新的 DDL 语句。

```
CREATE DATABASE cdb1
...
ENABLE PLUGGABLE DATABASE
SEED
FILE_NAME_CONVERT=('/u01/app/oradata/CDB1','/u01/app/oradata
/seed');

CREATE PLUGGABLE DATABASE pdb1 ... ;
ALTER PLUGGABLE DATABASE pdb1 UNPLUG INTO ... ;
ALTER PLUGGABLE DATABASE ALL OPEN ;
ALTER PLUGGABLE DATABASE pdb2 CLOSE IMMEDIATE ;
```

预配新的可插入数据库

多种方法：

- 从种子 PDB 创建新的 PDB。
- 将非 CDB 插入或克隆到 CDB 中。
- 克隆：
 - 将本地 PDB 克隆到同一个 CDB 中
 - 将远程 PDB 克隆到 CDB 中
- 将移走的 PDB 插入到另一个 CDB 中。



海量数据
VASTDATA

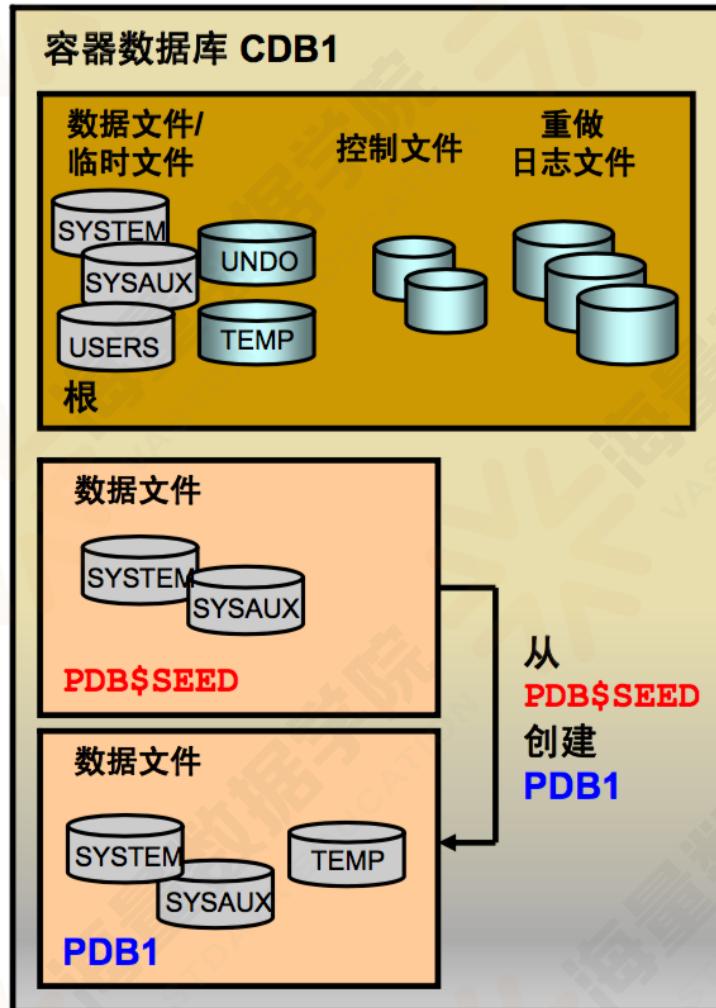
海量数据学院
VASTDATA EDUCATION

工具

要预配新的 PDB，您可以使用：

- SQL*Plus
- SQL Developer
- Enterprise Manager Cloud Control
- Enterprise Manager Database Express
- DBCA
 - 从种子复制
 - 通过移走/插入方法

方法 1：从 PDB\$SEED 创建新 PDB



- 从 PDB\$SEED 数据文件复制数据文件
- 创建 SYSTEM 和 SYSAUX 表空间
- 创建完整目录，包括指向 Oracle 提供的对象的元数据
- 创建临时表空间 TEMP
- 创建公用用户：
 - 超级用户 SYS
 - SYSTEM
- 创建授予本地 PDB_DBA 角色的本地用户 (PDBA)
- 创建新的默认服务



步骤：使用位置子句

以拥有 CREATE PLUGGABLE DATABASE 权限的公用用户的身份连接到根：

- 使用 **FILE_NAME_CONVERT**：

```
SQL> CREATE PLUGGABLE DATABASE pdb1
      ADMIN USER admin1 IDENTIFIED BY p1 ROLES=(CONNECT)
      FILE_NAME_CONVERT = ('PDB$SEEDdir', 'PDB1dir');
```

- 使用 **CREATE_FILE_DEST**：

12.1.0.2

```
SQL> CREATE PLUGGABLE DATABASE pdb2
      ADMIN USER admin2 IDENTIFIED BY p2 ROLES=(CONNECT)
      CREATE_FILE_DEST = 'PDB2dir';
```

- 使用视图验证：

```
SQL> CONNECT / AS SYSDBA
SQL> SELECT * FROM cdb_pdbs;
SQL> SELECT * FROM cdb tablespaces;
SQL> SELECT * FROM cdb data files;
SQL> CONNECT sys@pdb1 AS SYSDBA
SQL> CONNECT admin1@pdb1
```

步骤：不使用位置子句

在不使用子句的情况下从种子创建新 PDB:

- 传统 OMF 参数:

DB_CREATE_FILE_DEST = '/CDBS/CDB1/pdb1'

```
SQL> CREATE PLUGGABLE DATABASE pdb1
      ADMIN USER pdb1_admin IDENTIFIED BY p1
      ROLES=(CONNECT);
```

- 新参数: **PDB_FILE_NAME_CONVERT =**
'/u01/app/oradata/CDB1/seed','CDBS/CDB1/pdb2'

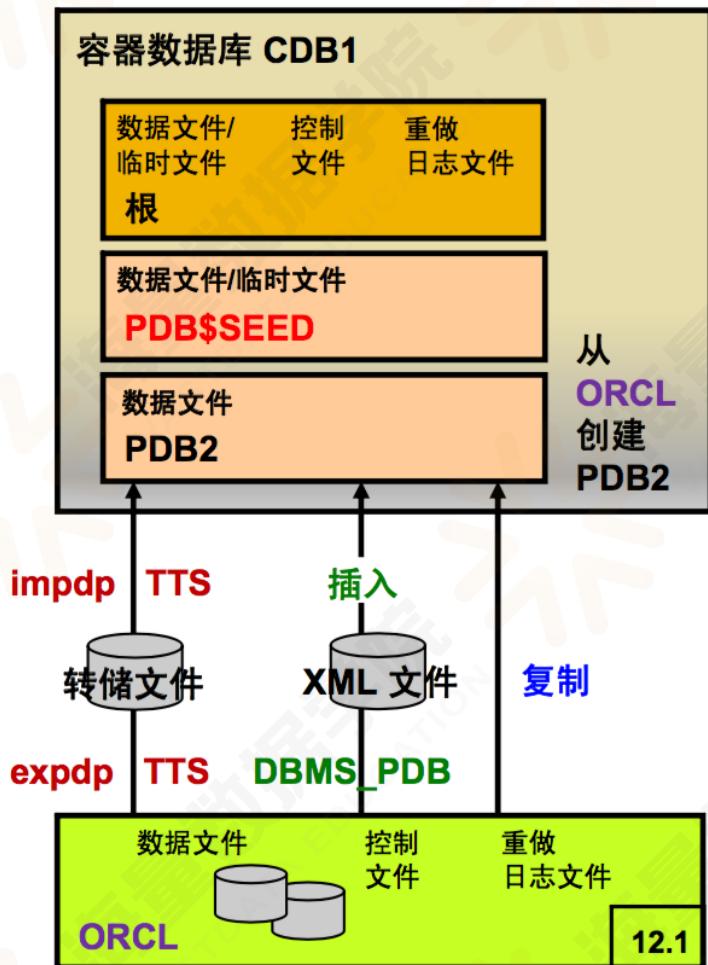
```
SQL> CREATE PLUGGABLE DATABASE pdb2
      ADMIN USER pdb2_admin IDENTIFIED BY p2
      ROLES=(CONNECT);
```

同步

- 如果 PDB 处于 MOUNTED 或 READ-ONLY 模式
 - 如果客户创建的公用用户或角色已创建/已修改/已删除
- PDB 需要与根重新同步。
- 在 READ-WRITE 模式下打开 PDB，以便同步 PDB 与目标 CDB。
 - 如果在 READ-ONLY 模式下打开 PDB，则会返回错误。
- 自动执行兼容性检查：
- 在 PDB_PLUG_IN_VIOLATIONS 视图中报告任何违规。
 - 如果无违规，PDB 状态则更改为 NORMAL。



方法 2：将非 CDB 插入 CDB



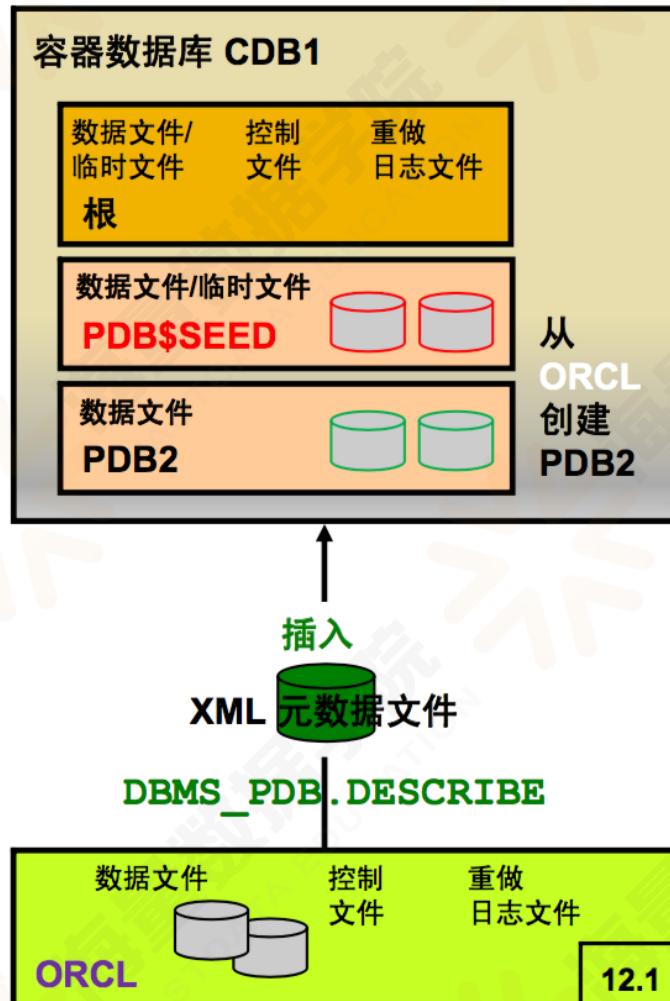
三种可行的方法：

- TTS 或 TDB 或完全导出/导入
- 使用 DBMS_PDB 构造 XML 文件定义
- 复制

新 PDB 中会创建以下实体：

- 表空间： SYSTEM、 SYSAUX、 TEMP
- 完整目录
- 公用用户： SYS、 SYSTEM
- 本地管理员 (PDBA)
- 新的默认服务

使用 DBMS_PDB 将非 CDB 插入 CDB



1. 在 READ ONLY 模式下打开 ORCL。

```
SQL> EXEC DBMS_PDB.DESCRIBE  
( '/tmp/ORCL.xml' )
```

2. 以拥有 CREATE PLUGGABLE DATABASE 权限的公用用户的身份连接到目标 CDB1 CDB。

3. 以 PDB2 的身份插入移走的 ORCL。

```
SQL> CREATE PLUGGABLE DATABASE  
PDB2 USING '/tmp/ORCL.xml';
```

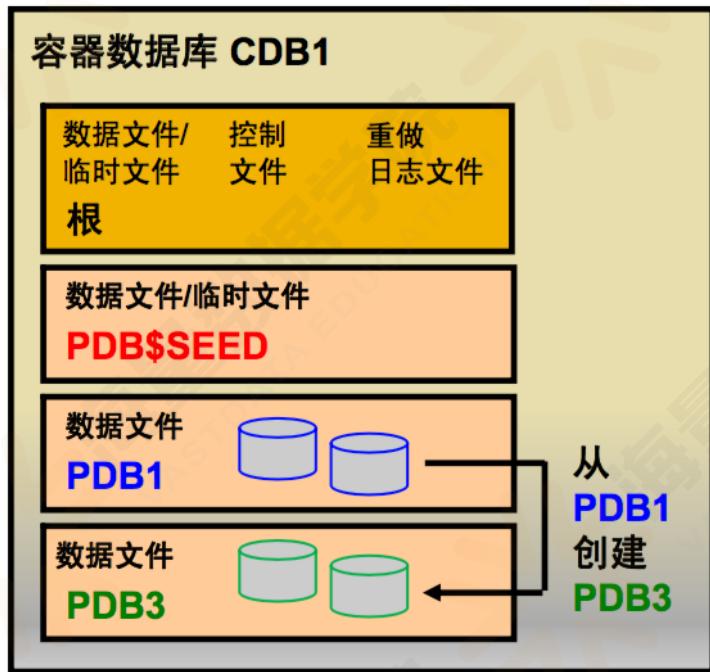
4. 运行 noncdb_to_pdb.sql 脚本。

```
SQL> CONNECT sys@PDB2 AS SYSDBA  
SQL> @$ORACLE_HOME/rdbms/admin/noncdb_to_pdb
```

5. 打开 PDB2。

```
SQL> ALTER PLUGGABLE DATABASE  
PDB2 OPEN;
```

方法 3：克隆本地 PDB



PDB3 拥有：

- SYSTEM、SYSAUX 表空间
- 完整目录
- 临时表空间
- SYS、SYSTEM 公用用户
- 同一本地管理员名
- 新服务名

1. 设置 DB_CREATE_FILE_DEST 或 DB_FILE_NAME_CONVERT 实例参数或使用 CREATE_FILE_DEST 子句。 12.1.0.2

2. 连接到根。

3. 停顿 PDB1：

```
SQL> ALTER PLUGGABLE DATABASE
      pdb1 CLOSE;
SQL> ALTER PLUGGABLE DATABASE
      pdb1 OPEN READ ONLY;
```

4. 从 PDB1 克隆 PDB3：

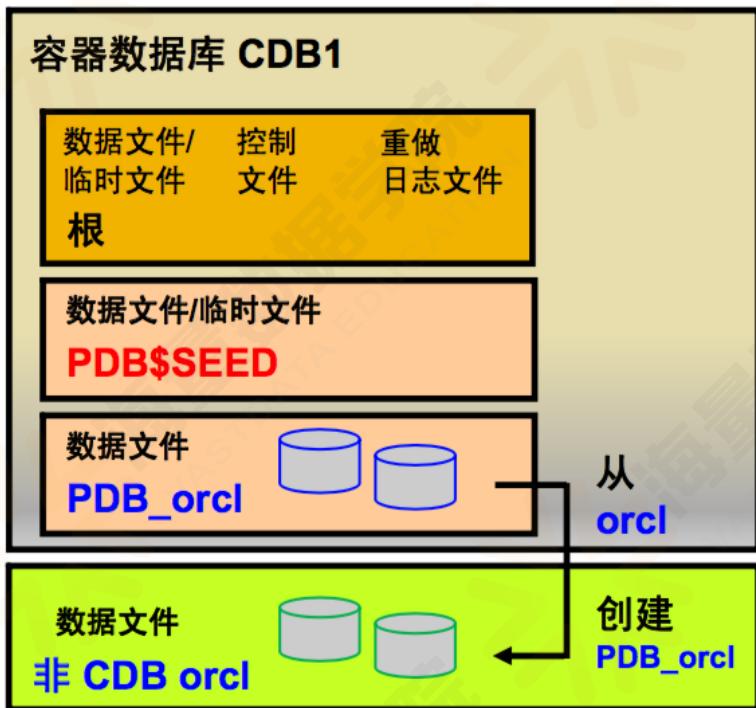
```
SQL> CREATE PLUGGABLE DATABASE
      pdb3 FROM pdb1;
```

5. 在读写模式下打开 PDB3。

```
SQL> ALTER PLUGGABLE DATABASE
      pdb3 OPEN;
```

6. 重新打开 PDB1。

方法 3：克隆非 CDB 或远程 PDB



PDB_orcl 拥有：

- SYSTEM、SYSAUX 表空间
- 完整目录
- 临时表空间
- SYS、SYSTEM 公用用户
- 新服务名

1. 将 **ORCL** 设置为只读模式。
2. 连接到 CDB，创建数据库链接：

```
SQL> CREATE DATABASE LINK  
link_orcl connect to system  
identified by ***  
using 'orcl';
```

3. 克隆非 CDB：

```
SQL> CREATE PLUGGABLE DATABASE  
pdb_orcl FROM NON$CDB@link_orcl  
CREATE_FILE_DEST = '.../PDB_orcl';
```

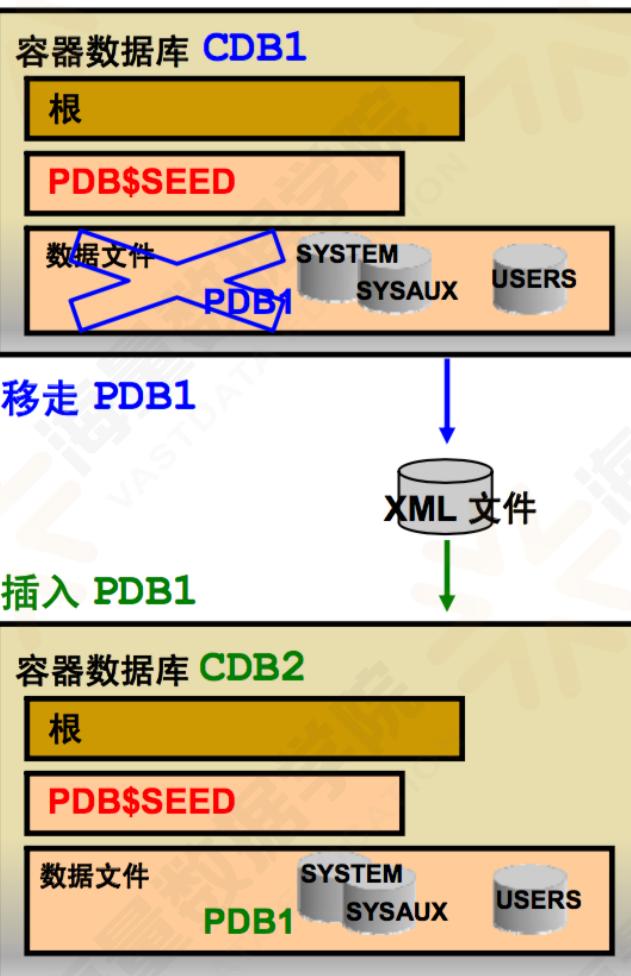
4. 运行 `noncdb_to_pdb.sql` 脚本。

```
SQL> CONNECT sys@pdb_orcl AS SYSDBA  
SQL> @$ORACLE_HOME/rdbms/admin/noncdb_to_pdb
```

5. 在读写模式下打开 **PDB_ORCL**。

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

方法 4：将移走的 PDB 插入 CDB



将 PDB1 从 CDB1 移走:

1. 以公用用户的身份连接到 CDB1。
2. 验证 PDB1 是否已关闭。
3.

```
SQL> ALTER PLUGGABLE DATABASE
      pdb1 UNPLUG INTO
      'xmlfile1.xml';
```
4. 或者, 将 PDB1 从 CDB1 删除。

将 PDB1 插入 CDB2:

1. 以公用用户的身份连接到 CDB2。
2. 使用 DBMS_PDB 程序包检查 PDB1 与 CDB2 的兼容性。
3.

```
SQL> CREATE PLUGGABLE DATABASE
      pdb1 USING 'xmlfile1.xml'
      NOCOPY;
```
4. 在读写模式下打开 PDB1。

方法 4：流

若干子句可以结合在一起使用：

新 PDB 文件是否基于用于创建 CDB 中现有 PDB 的相同文件？

如果不是，则需要使用 AS CLONE 子句，如此一来，它可以确保 Oracle DB 生成唯一的 PDB DBID、GUID 和新 PDB 所需的其他标识符。

XML 文件是否准确描述了当前的文件位置？

如果不是，则需要使用 SOURCE_FILE_NAME_CONVERT 子句。

文件是否在正确的位置？

如果不是，则指定 COPY 将文件复制到新位置，或者指定 MOVE 将它们移动到另一个位置。

如果是，则使用 NOCOPY。COPY 为默认值。

- CREATE PLUGGABLE DATABASE 语句的 FILE_NAME_CONVERT 或 CREATE_FILE_DEST 子句
- 实例 DB_CREATE_FILE_DEST 或 PDB_FILE_NAME_CONVERT 参数

是否要指定 PDB 的存储限制？

如果是，则指定 STORAGE 子句。

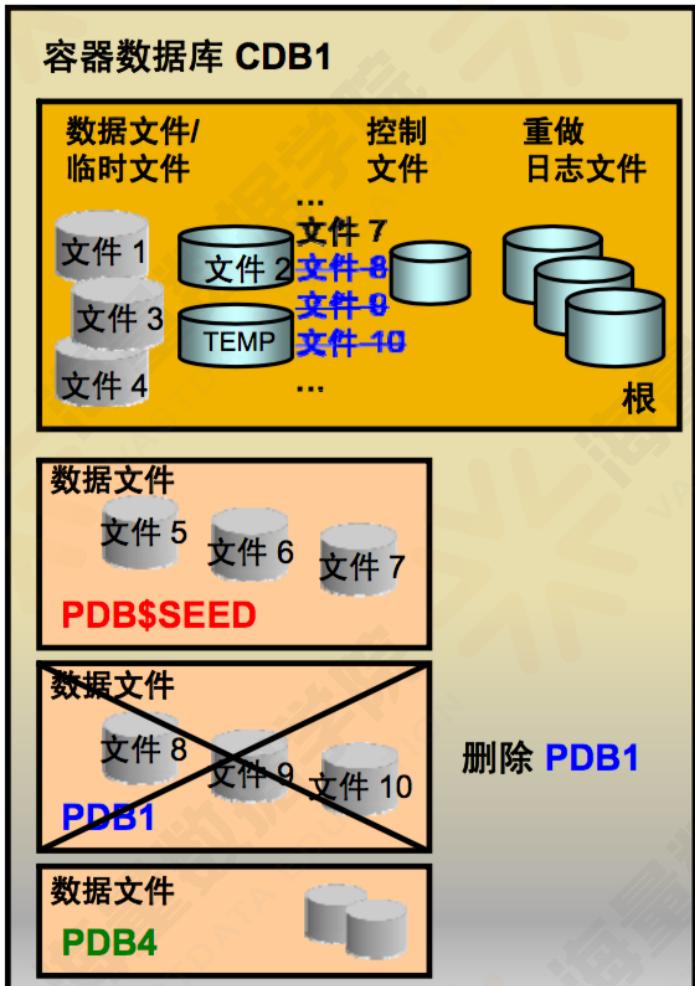
插入示例方案 PDB：使用 DBCA

The screenshot illustrates the Oracle Database Configuration Assistant (DBCA) interface across four steps:

- Step 1 of 7: Database Operation** (Green circle 1)
 - Left sidebar: Database Operation, Manage Pluggable Databases, Database List, Create Pluggable Database, Pluggable Database Options, Summary, Progress Page.
 - Main area: Select the operation that you want to perform:
 - Create a Pluggable Database
 - Create Database
 - Configure Database Options
 - Delete Database
 - Manage Templates
 - Manage Pluggable Databases
- Step 2: Select an operation that you want to perform in container database:** (Green circle 2)
 - Create a Pluggable Database
 - Unplug a Pluggable Database
 - Delete a Pluggable Database
- Step 3: Select the database in which Pluggable database needs to be created:** (Green circle 3)
 - Select Database:
 - cdb1
 - cdb2
 - orcl
 - orcl3
 - cdb3
- Step 4 of 7: Create Pluggable Database** (Green circle 4)
 - Left sidebar: Database Operation, Manage Pluggable Databases, Database List, **Create Pluggable Database**, Pluggable Database Options, Summary, Progress Page.
 - Main area:
 - Create a new Pluggable Database
 - Create Pluggable Database From PDB Archive
Pluggable Database Archive: Browse
 - Create Pluggable Database using PDB File Set
Pluggable Database Metadata File: _1/assistants/dbca/templates/sampleschema.xml
Browse
Pluggable Database Datafile Backup: _1/assistants/dbca/templates/sampleschema.fdb
Browse

A yellow callout bubble points to the "Create Pluggable Database using PDB File Set" option with the text: "使用 PDB 文件集 插入采用示例 方案的新 PDB".

删除 PDB



```
SQL> ALTER PLUGGABLE DATABASE  
      pdb1 CLOSE;  
SQL> DROP PLUGGABLE DATABASE  
      pdb1 [ INCLUDING DATAFILES ];
```

- **更新控制文件**
- **如果 INCLUDING DATAFILES :**
 - 删除 **PDB1** 数据文件
- **如果 KEEP DATAFILES (默认值) :**
 - 保留数据文件
 - 可插入另一个或同一个 CDB 中
- **要求 SYSDBA 权限**
- **不能删除种子 PDB**

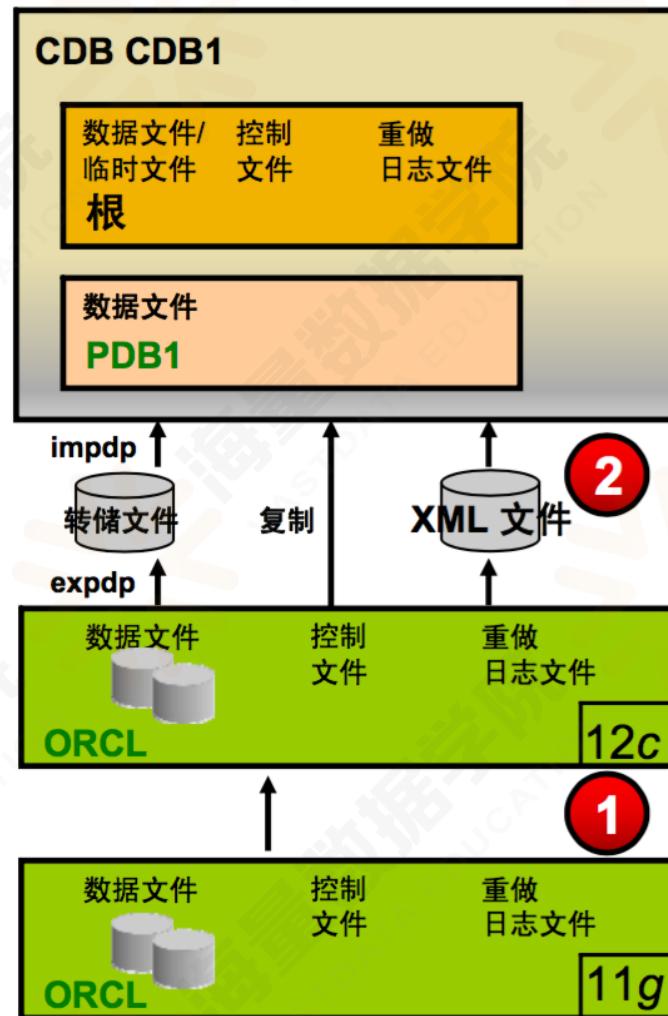
将 12.1 之前版本的数据库移植到 12.1 CDB

有两种方法：

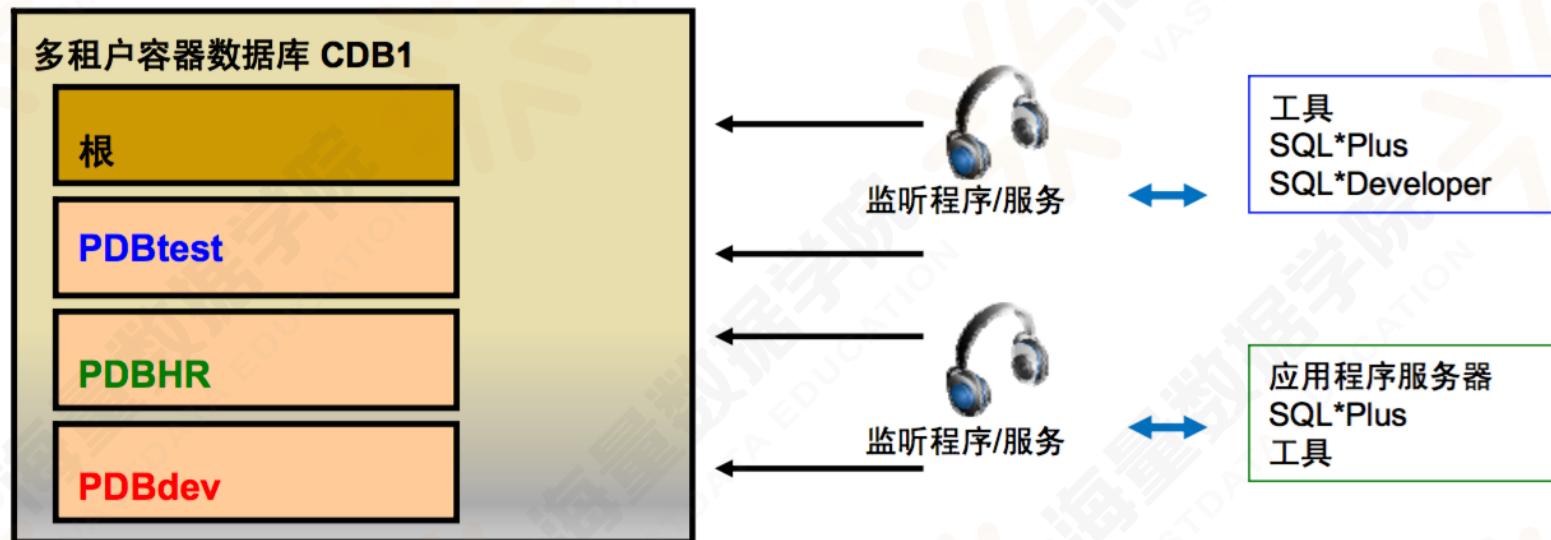
1. 将现有的 12.1 之前版本的数据库升级到 12c。
2. 将非 CDB 插入 CDB。

或

1. 在 CDB 中预创建 PDB。
2. 使用 11g expdp / 12c impdp
或者，在非 CDB 和 PDB 之间执行复制。



连接



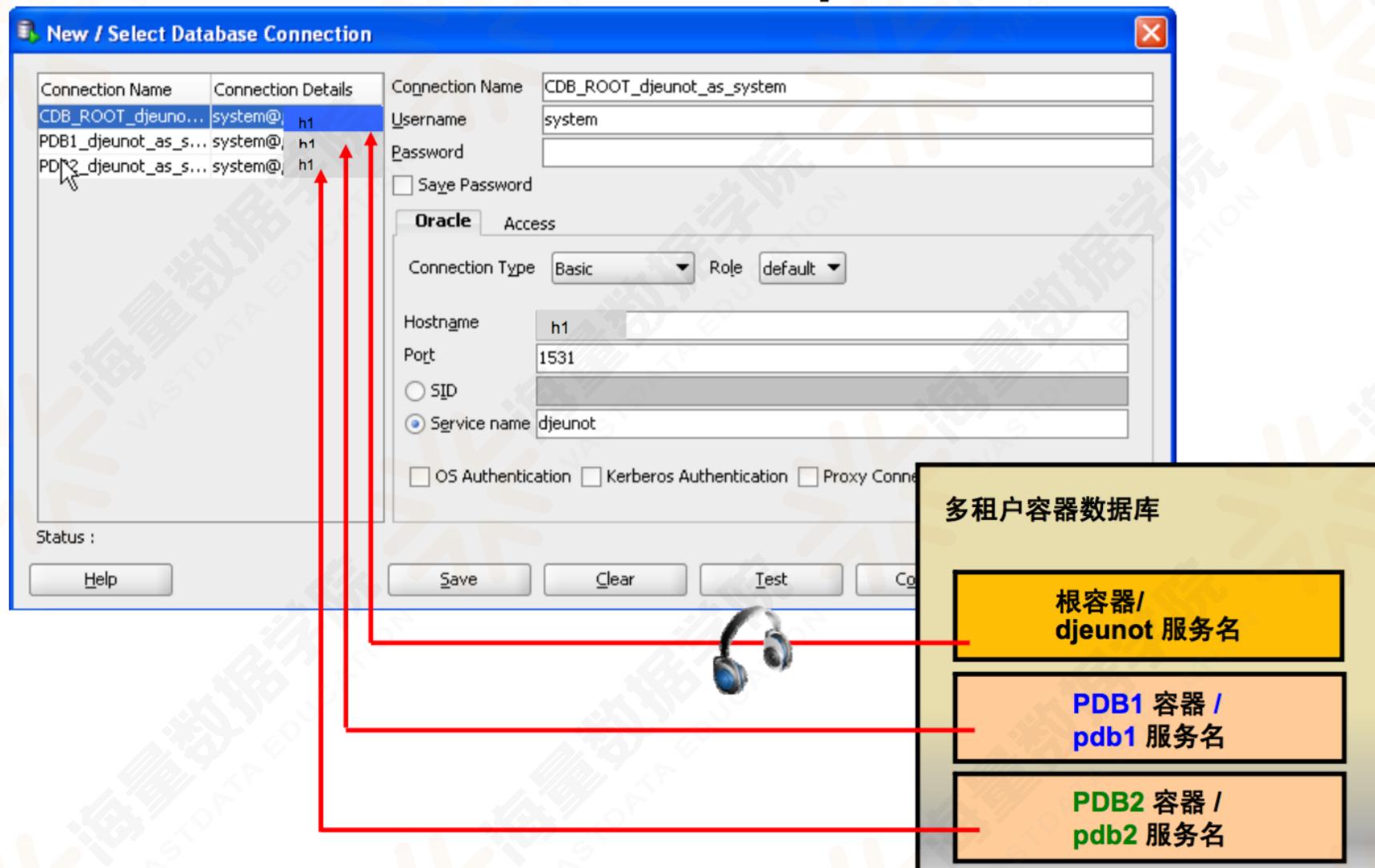
- 每个 PDB 都有一个默认服务。

```
SQL> SELECT name, pdb FROM cdb_services;
```

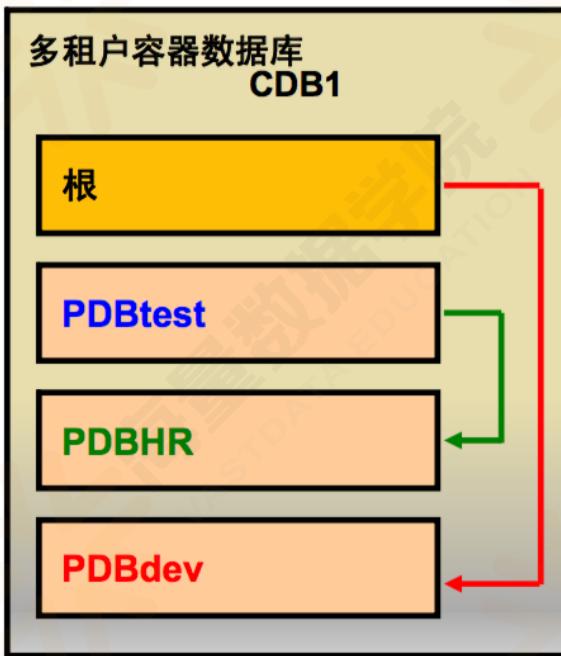
- 服务名必须在各个 CDB 中都是唯一的。

```
SQL> CONNECT / AS SYSDBA
SQL> CONNECT sys@PDBtest AS SYSDBA
SQL> CONNECT local_user1@hostname1:1525/PDBHR
SQL> CONNECT common_user2@PDBdev
SQL> SHOW CON_NAME
```

使用 SQL*Developer 连接



切换连接



在 CDB 中的容器之间切换连接的方式有两种：

- 重新连接：

```
SQL> CONNECT / AS SYSDBA  
SQL> CONNECT local_user1@PDBdev
```

- 使用 ALTER SESSION 语句：

```
SQL> CONNECT sys@PDBtest AS SYSDBA  
SQL> ALTER SESSION SET CONTAINER=PDBHR;  
SQL> SHOW CON_NAME  
SQL> ALTER SESSION SET  
CONTAINER=CDB$ROOT;
```

- 使用 CONNECT 时，能以公用用户或本地用户的身份进行连接。
- 使用 ALTER SESSION SET CONTAINER 时，只能以授予新系统权限 SET CONTAINER 的公用用户身份进行连接。
 - AFTER LOGON 触发器不触发。
 - 事务处理在切换容器后依然保持暂挂状态。

启动 CDB 实例

```
SQL> CONNECT sys@CDB1 AS SYSDBA  
SQL> STARTUP NOMOUNT
```

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

```
no rows selected
```

NOMOUNT

CDB 实例已启动

SHUTDOWN

装载 CDB

```
SQL> CONNECT sys@cdb1 AS SYSDBA  
SQL> STARTUP MOUNT
```

或

```
SQL> ALTER DATABASE cdb1 MOUNT;
```

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

NAME	OPEN_MODE
PDB\$SEED	MOUNTED
PDB1	MOUNTED
PDB2	MOUNTED

NOMOUNT

实例
已启动

SHUTDOWN

MOUNT

- CDB 控制文件已为实例打开
- 根已装载
- PDB 已装载

打开 CDB

```
SQL> STARTUP
```

```
SQL> ALTER DATABASE cdb1 OPEN;
```

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

NAME	OPEN_MODE
PDB\$SEED	READ ONLY
PDB1	MOUNTED
PDB2	MOUNTED

或

OPEN

- 根已打开
- PDB 仍然处于装载状态，但处于只读模式的种子除外

MOUNT

- CDB 控制文件已为实例打开
- 根已装载
- PDB 已装载

NOMOUNT

实例
已启动

SHUTDOWN



海量数据
VASTDATA

海量数据学院
VASTDATA EDUCATION

打开 PDB

```
SQL> CONNECT sys@cdb1 AS SYSDBA  
SQL> ALTER PLUGGABLE DATABASE pdb1 OPEN;
```

或

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

PDB OPEN

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

NAME	OPEN_MODE
PDB\$SEED	READ ONLY
PDB1	READ WRITE
PDB2	READ WRITE

PDB 在**读写模式**
下打开, 但处于只
读模式的种子除外

OPEN

- 根已打开
- PDB 仍然处于装载
状态, **但处于只读
模式的种子除外**

MOUNT

- CDB 控制文件已为实例打开
- 根已装载
- PDB 已装载

NOMOUNT

实例
已启动

SHUTDOWN



关闭 PDB

```
SQL> CONNECT / AS SYSDBA  
SQL> ALTER PLUGGABLE DATABASE pdb1  
    CLOSE IMMEDIATE;  
SQL> ALTER PLUGGABLE DATABASE  
    ALL EXCEPT pdb1 CLOSE;  
SQL> ALTER PLUGGABLE DATABASE  
    ALL CLOSE;
```

```
SQL> CONNECT sys@pdb1 AS SYSDBA  
SQL> ALTER PLUGGABLE DATABASE CLOSE;  
或者  
SQL> SHUTDOWN IMMEDIATE;
```

SHUTDOWN

实例
已启动

NOMOUNT

MOUNT

CDB OPEN

- 根已打开
- PDB 已装载，但仍然处于只读模式的种子除外

- CDB 控制文件已为实例打开
- 根已装载
- PDB 已装载

PDB CLOSE

PDB 已关闭

关闭 CDB 实例

```
SQL> CONNECT sys@CDB1 AS SYSDBA  
SQL> SHUTDOWN IMMEDIATE
```

- 所有 PDB 均已关闭（无新的具体消息）
- CDB 已关闭
- CDB 已卸装
- 实例已关闭

```
SQL> CONNECT sys@PDB1 AS SYSDBA  
SQL> SHUTDOWN IMMEDIATE
```

- PDB 已关闭

自动打开 PDB

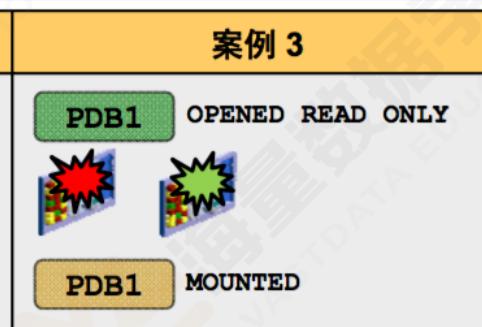
- CDB STARTUP 后自动保留 PDB 状态:

```
SQL> ALTER PLUGGABLE DATABASE pdb1 SAVE STATE;
```

案例 1	案例 2	案例 3
		

- CDB STARTUP 后自动放弃 PDB 状态:

```
SQL> ALTER PLUGGABLE DATABASE pdb1 DISCARD STATE;
```

案例 1	案例 2	案例 3
		

更改 PDB 打开模式

关闭 PDB 后，可在三种模式下打开 PDB：

- 受限读写：

```
SQL> ALTER PLUGGABLE DATABASE pdb1 OPEN RESTRICTED;
```

```
SQL> SELECT name, open_mode, restricted FROM v$pdbs;
```

NAME	OPEN_MODE	RES
PDB1	READ WRITE	YES
PDB2	READ WRITE	NO

- 只读：

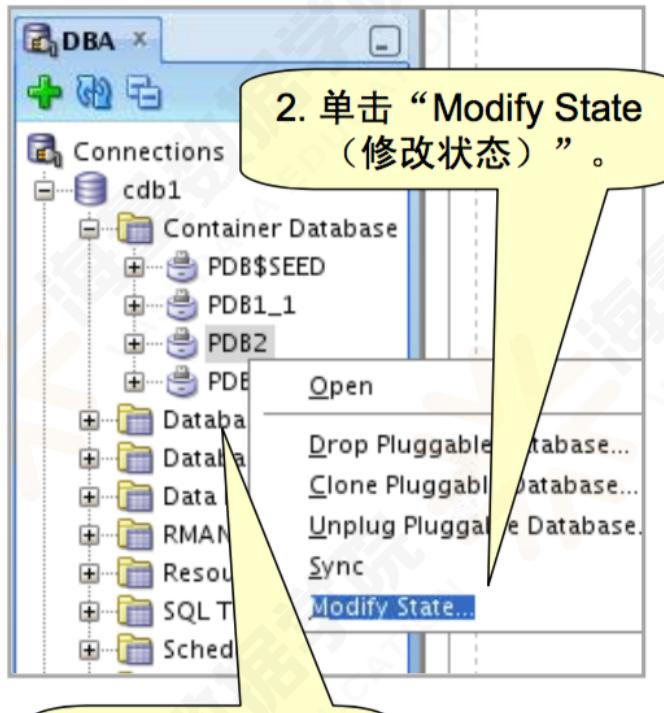
```
SQL> ALTER PLUGGABLE DATABASE ALL OPEN READ ONLY;
```

- 读写：

```
SQL> ALTER PLUGGABLE DATABASE ALL CLOSE;
```

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

更改 PDB 模式：使用 SQL Developer



修改 PDB 设置

- 使 PDB 数据文件联机。
- 更改 PDB 默认表空间。
- 更改 PDB 默认临时表空间。
- 设置 PDB 存储限制。
- 更改全局名称。

```
SQL> CONNECT sys@pdb1 AS SYSDBA
SQL> ALTER PLUGGABLE DATABASE DATAFILE '/u03/pdb1_01.dbf'
ONLINE;
```

```
SQL> ALTER PLUGGABLE DATABASE DEFAULT TABLESPACE pdb1_tbs;
```

```
SQL> ALTER PLUGGABLE DATABASE DEFAULT TEMPORARY TABLESPACE
temp_tbs;
```

```
SQL> ALTER PLUGGABLE DATABASE STORAGE (MAXSIZE 2G);
```

```
SQL> ALTER PLUGGABLE DATABASE RENAME GLOBAL_NAME TO pdbAPP1;
```

实例参数更改影响

- 每个 CDB 一个 SPFILE
- PDB 参数值更改：
 - 在 PDB 关闭后加载到内存中
 - 在 CDB 关闭后存储在字典中
 - 仅适用于 ISPDB_MODIFIABLE=TRUE 的参数

```
SQL> CONNECT sys@pdb1 AS SYSDBA
Connected.
SQL> ALTER SYSTEM SET ddl_lock_timeout=10;
System altered.
SQL> show parameter ddl_lock_timeout
```

NAME	TYPE	VALUE
ddl_lock_timeout	boolean	10

实例参数更改影响：示例

```
SQL> CONNECT sys@pdb2 AS SYSDBA

SQL> ALTER SYSTEM SET ddl_lock_timeout=20 scope=BOTH;

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

```
SQL> CONNECT / AS SYSDBA
SQL> select VALUE, ISPDB_MODIFIABLE, CON_ID
   from V$SYSTEM_PARAMETER
  where name ='ddl_lock_timeout';

      VALUE          ISPDB      CON_ID
-----  -----  -----
      0            TRUE          0
     10           TRUE          3
     20           TRUE          4
```

PDB 中的表空间

- PDB 中的表空间可以包含与且仅与一个 PDB 关联的多个对象。
- 在 CREATE DATABASE 中：
 - 利用 DBCA, USER_DATA TABLESPACE 替换自动创建 USERS 表空间。
- 每个 CDB 实例只有一个活动 UNDO 表空间。



在 CDB 中创建永久表空间

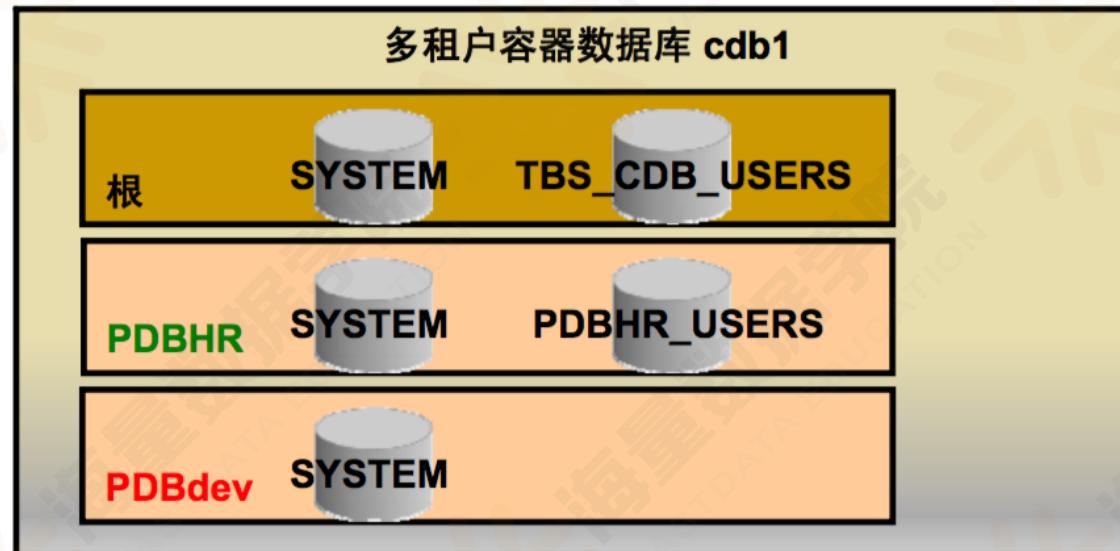
- 在 PDB 中创建永久表空间：

```
SQL> CONNECT system@PDB1
SQL> CREATE TABLESPACE tbs_PDB1_users DATAFILE
  '/u1/app/oracle/oradata/cdb/pdb1/users01.dbf'
  SIZE 100M;
```

- 在根容器中创建永久表空间：

```
SQL> CONNECT system@cdb1
SQL> CREATE TABLESPACE tbs_CDB_users DATAFILE
  '/u1/app/oracle/oradata/cdb/cdb_users01.dbf'
  SIZE 100M;
```

分配默认表空间



- 在 CDB 中：

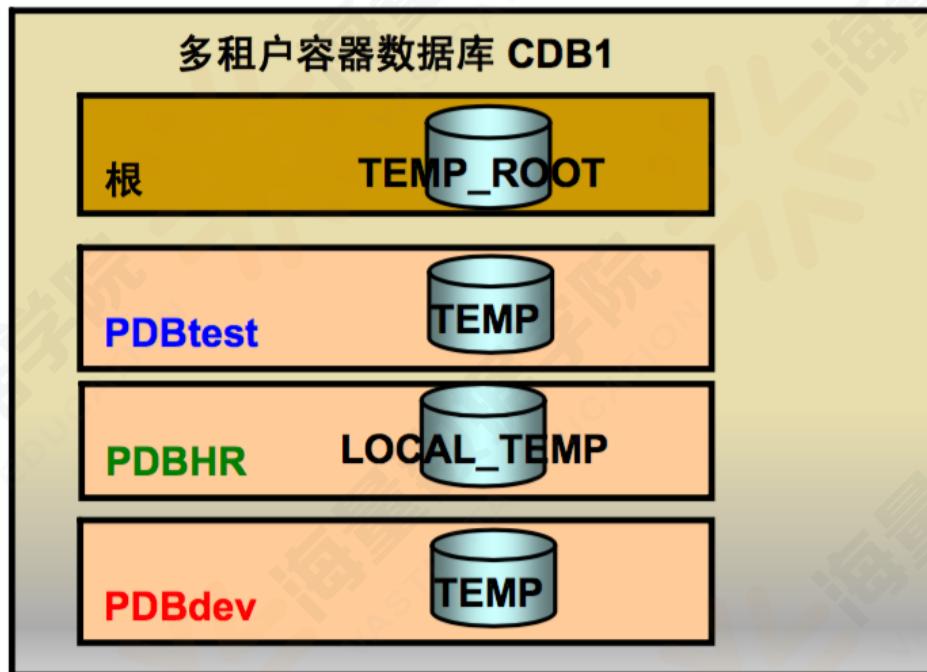
```
SQL> CONNECT system@cdb1
SQL> ALTER DATABASE
      DEFAULT TABLESPACE tbs_CDB_users;
```

- 在 PDB 中：

```
SQL> CONNECT pdb1_admin@pdahr
SQL> ALTER PLUGGABLE DATABASE
      DEFAULT TABLESPACE pdahr_users;
```

创建本地临时表空间

- 每个 CDB 或 PDB 仅允许一个默认临时表空间或表空间组。
- 每个 PDB 具有其自己的临时表空间或表空间组。



分配默认临时表空间

- 在 CDB 中：

```
SQL> CONNECT system@cdb1
SQL> ALTER DATABASE
      DEFAULT TEMPORARY TABLESPACE temp_root;
```

- 在 PDB 中：

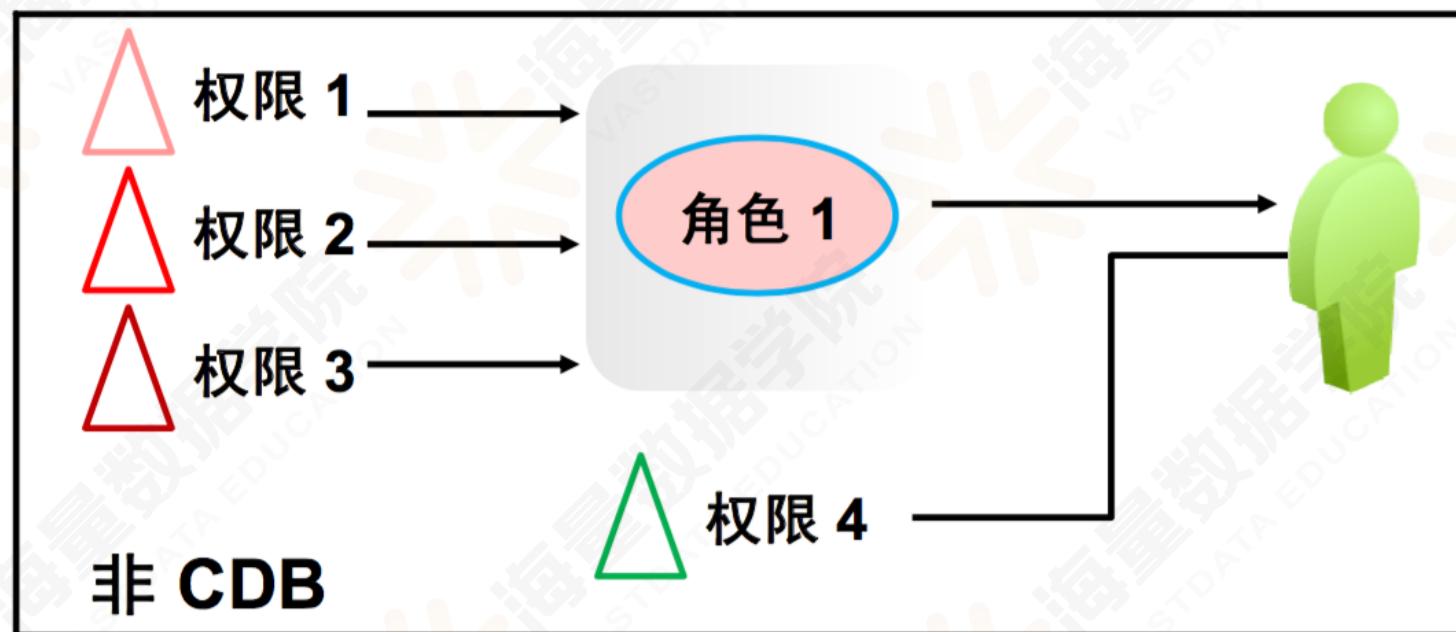
```
SQL> CONNECT pdb1_admin@pbhr
SQL> ALTER PLUGGABLE DATABASE
      DEFAULT TEMPORARY TABLESPACE local_temp;
```

或

```
SQL> CONNECT pdb1_admin@pbtest
SQL> ALTER DATABASE
      DEFAULT TEMPORARY TABLESPACE temp;
```

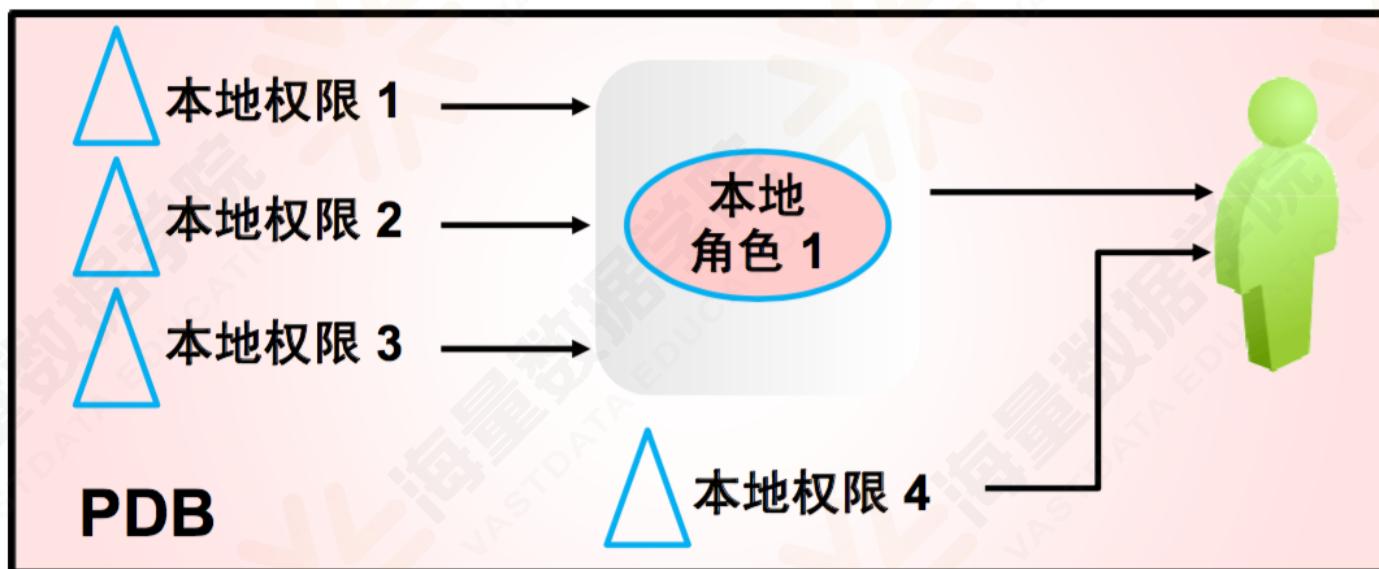
用户、角色和权限

- 每个用户可以在单个数据库的上下文中行使授予的权限。
- 角色是权限的集合。



本地用户、角色和权限

- 每个本地用户可以在单个 PDB 的上下文中行使授予的权限。
- 本地角色是在用户登录到特定的 PDB 时分配的权限集合。
- 本地权限是在单个 PDB 的上下文中授予的权限。



创建本地用户

- 具有适当权限的本地用户可以创建其他本地用户。
- 语法与在非 CDB 中的语法相同。

```
SQL> CONNECT system@pdb1  
SQL> CREATE USER george IDENTIFIED BY x;
```

- 不能在根中创建本地用户。
- 本地用户仅连接至其所在的 PDB。

```
SQL> CONNECT george@pdb1
```

公用用户

多租户容器数据库 CDB1

视图 CDB_USERS	
USERNAME	COMMON
SYS	YES
SYSTEM	YES
C##JIM	YES
DBSNMP	YES
HR_MGR	NO
SALES	NO

根

只能在根容器中创建公用用户。

```
SQL> CREATE USER C##JIM IDENTIFIED BY x  
CONTAINER=ALL;
```

视图 DBA_USERS	
USERNAME	COMMON
SYS	YES
SYSTEM	YES
C##JIM	YES
DBSNMP	YES
HR_MGR	NO

PDB_HR

视图 DBA_USERS	
USERNAME	COMMON
SYS	YES
SYSTEM	YES
C##JIM	YES
DBSNMP	YES
SALES	NO

PDB_SALES

创建用户

- 公用用户可以创建公用用户和本地用户。
- CONTAINER** 子句确定所创建的用户的类型。
- 在根容器中创建公用用户：

```
SQL> CREATE USER C##_GEORGE IDENTIFIED BY x  
      CONTAINER=ALL;
```

- 在 PDB 中创建本地用户：

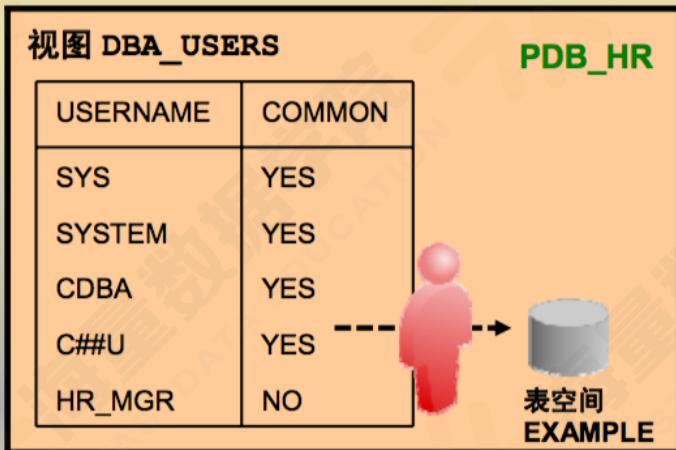
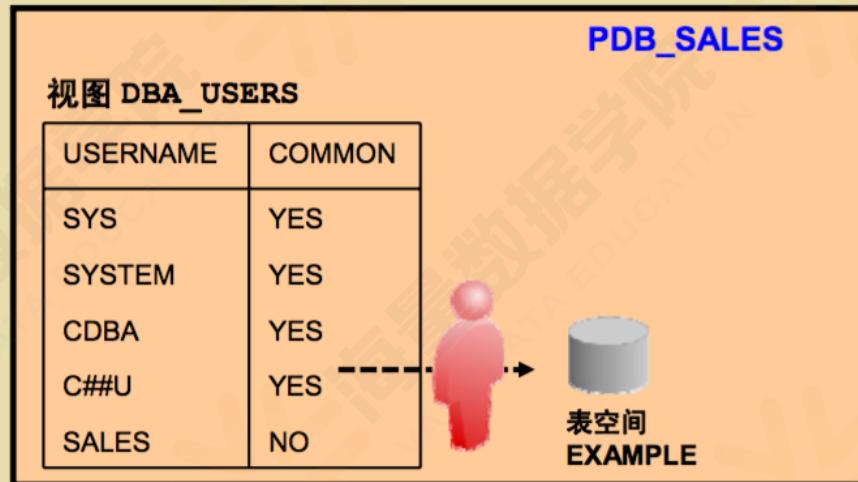
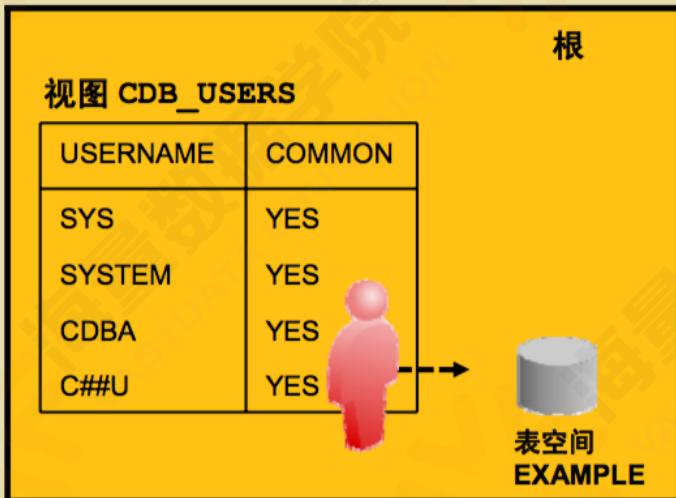
```
SQL> CREATE USER L_FRED IDENTIFIED BY y  
      CONTAINER=CURRENT;
```

- 每个容器中都存在一个公用用户：

```
SQL> CONNECT c##jim@cdb1 → jim connection in root  
SQL> CONNECT c##jim@pdb1 → jim connection in pdb1  
SQL> CONNECT c##jim@pdb2 → jim connection in pdb2
```

公用方案/用户和本地方案/用户

多租户容器数据库 CDB1



添加、变更或删除公用用户涉及在根和每个 PDB 中添加该用户的说明。



本地用户是传统用户，仅在自己的 PDB 中为已知。

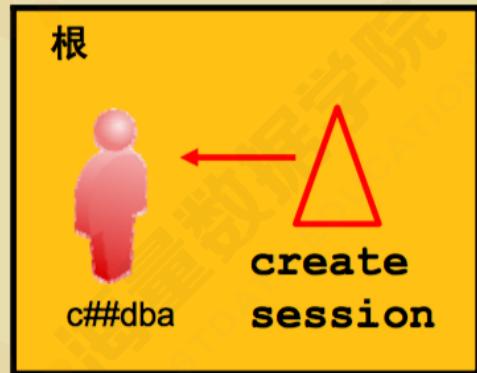


公用权限和本地权限

- 在所有容器中授予的权限是公用权限。
- 在单个 PDB 的上下文中授予的权限是本地权限。
- 本地用户只能在本地行使 PDB 的上下文中的权限。
- 公用用户只能行使所连接的 PDB 的上下文中的权限。
- 连接到根容器的公用用户可以行使跨容器权限，如创建公用用户。

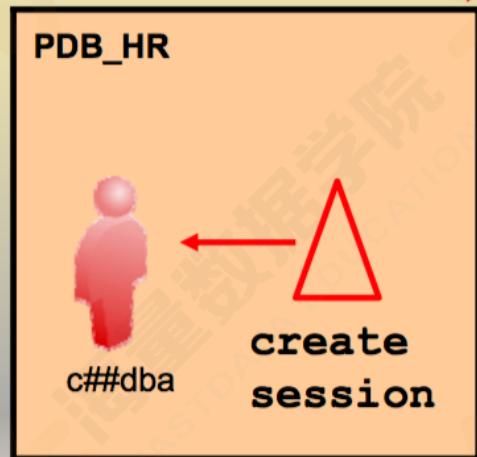
授予和撤消权限

多租户容器数据库 CDB1



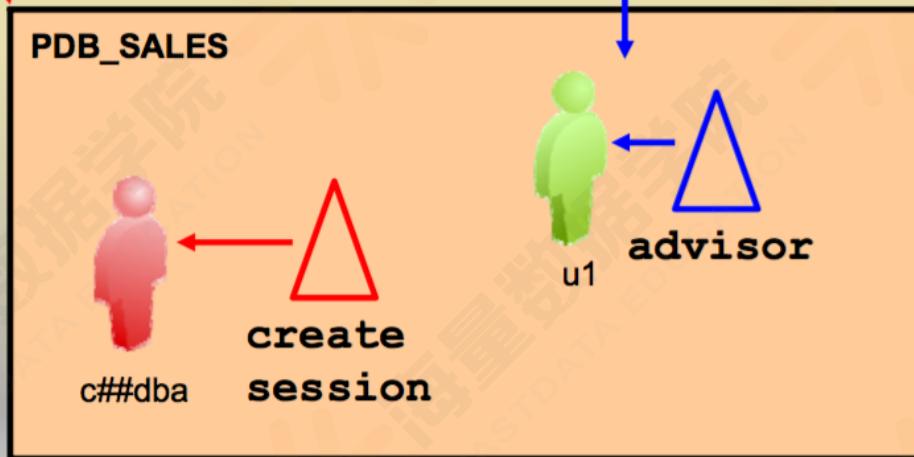
在所有容器中将公用权限授予被授权者。

```
SQL> GRANT create session TO c##dba  
2 CONTAINER=ALL;
```



在一个容器中将本地权限授予被授权者。

```
SQL> GRANT advisor TO u1;
```



创建公用角色和本地角色

多租户容器数据库 CDB1

在所有容器中创建公用角色。



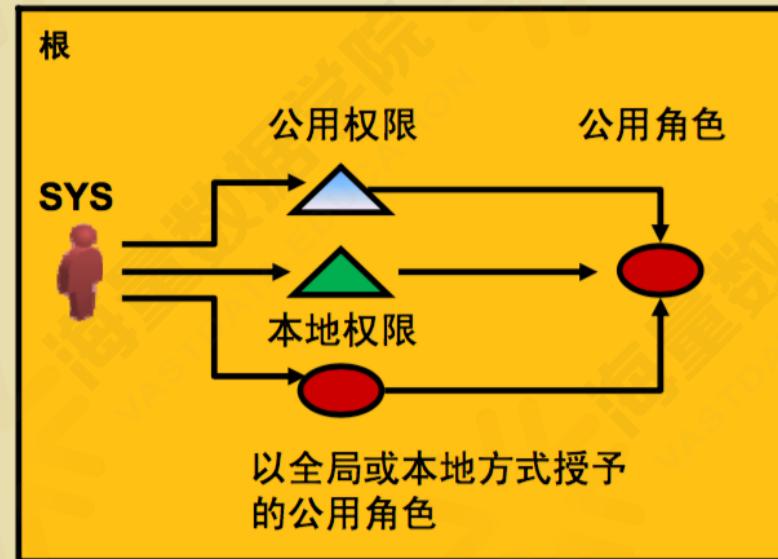
```
SQL> CREATE ROLE c##r1 CONTAINER=ALL;
```

在一个容器中创建本地角色。



向角色授予公用或本地权限/角色

多租户容器数据库 CDB1



公用权限和本地权限

```
SQL> GRANT priv TO c##_role  
CONTAINER=ALL;
```

```
SQL> GRANT priv TO C##_R2;
```

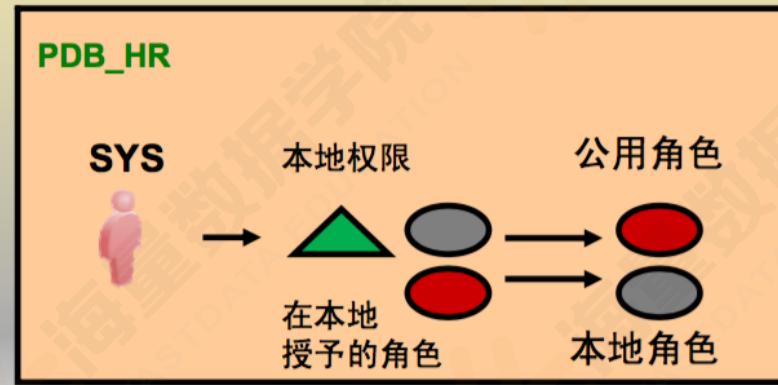
以全局方式授予的公用角色

```
SQL> GRANT c##r1 TO c##_role  
CONTAINER=ALL;
```

在本地授予的公用角色

```
SQL> GRANT c##r1 TO c##_r2;
```

PDB_HR



本地权限

```
SQL> GRANT priv  
TO MGR_ROLE  
CONTAINER=CURRENT;
```



公用用户



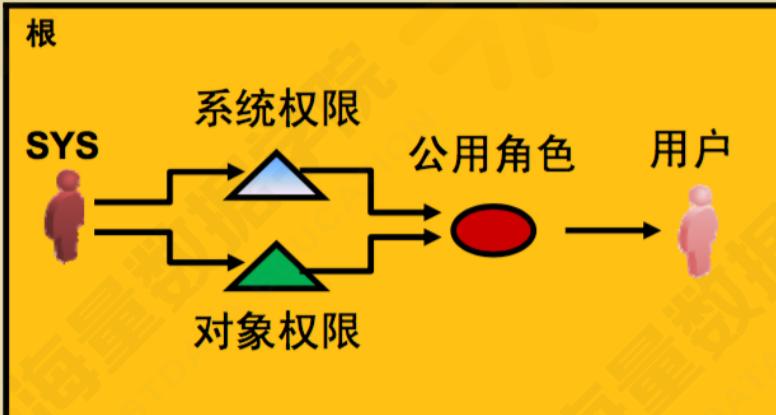
本地用户

公用角色

本地角色

向用户授予公用角色和本地角色

多租户容器数据库 CDB1



以全局方式授予的公用角色

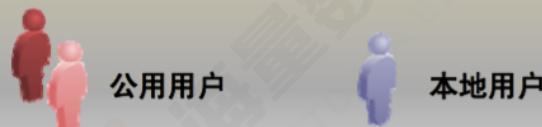
```
SQL> GRANT c##_role  
TO c##_user  
CONTAINER=ALL;
```

在本地授予的公用角色

```
SQL> GRANT c##_role  
TO l_user  
CONTAINER=CURRENT;
```

在本地授予的本地角色

```
SQL> GRANT HR_MGR_ROLE  
TO hr_user  
CONTAINER=CURRENT;
```



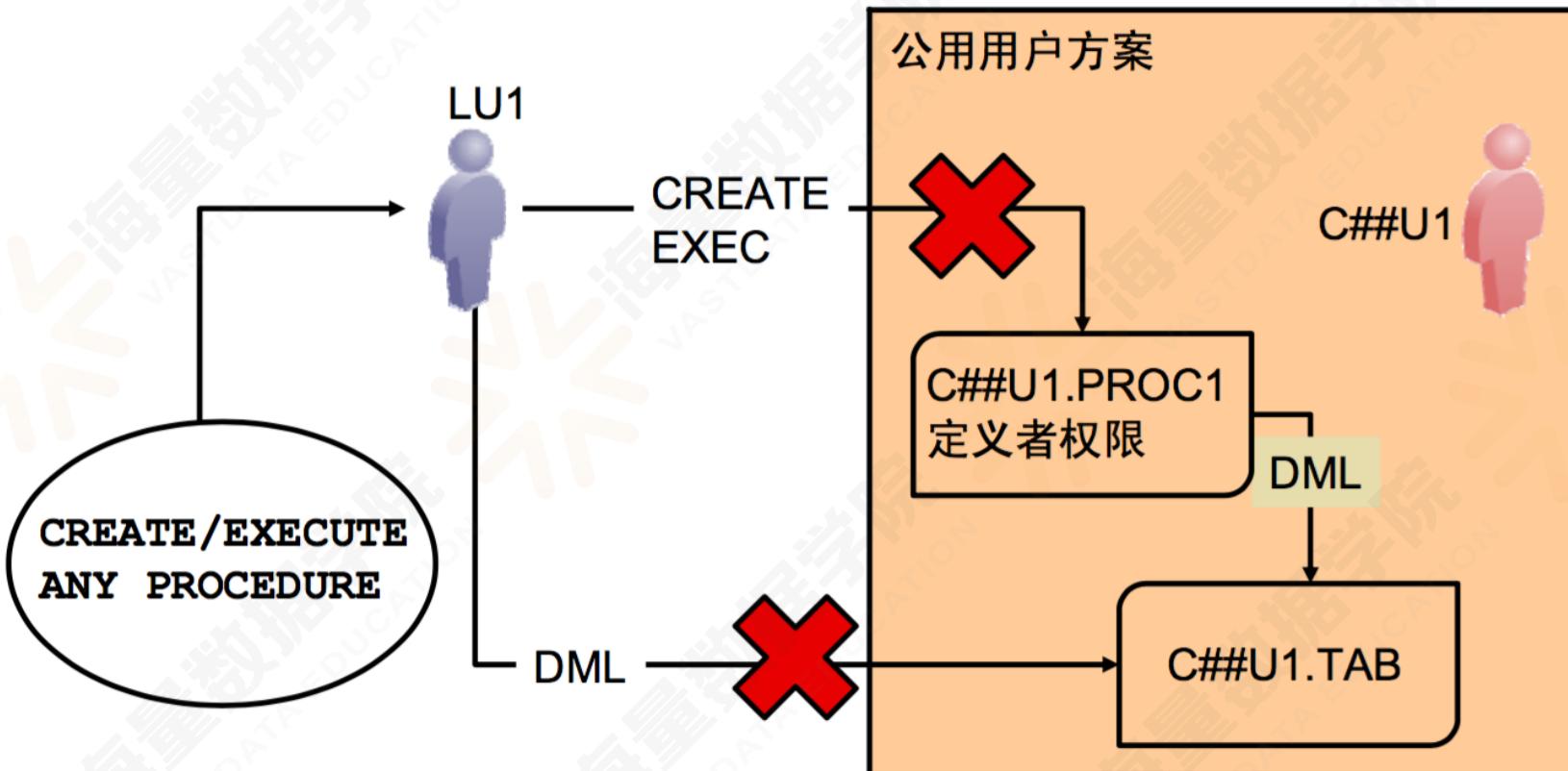
创建共享对象和非共享对象

- 公用用户和本地用户可以创建非共享表。
- 用户无法创建共享表。
- Oracle 提供的用户可以创建共享表。



定义者权限的限制

本地用户无法对公用用户的方案行使本地系统权限。



目标

- 独立备份 CDB 和应用程序：
 - CDB 级别的 ARCHIVELOG 模式
 - CDB 和 PDB 备份
 - CDB 和 PDB 级别的热备份
- 在不同级别恢复 CDB 或 PDB：
 - 实例故障：CDB 级别
 - 完全介质恢复：
 - CDB 或 PDB 临时文件
 - 控制文件/重做日志文件/根数据文件：已装载 CDB
 - PDB 数据文件
 - 不完全介质恢复：已装载 CDB 或已关闭 PDB
 - 闪回数据库：已装载 CDB

RMAN 中的新语法和子句

```
$ export ORACLE_SID=cdb1  
$ rman TARGET /
```

```
$ rman TARGET sys@sales_pdb
```

- DATABASE 关键字对所有 PDB 和根或 PDB 有效：

```
RMAN> BACKUP DATABASE;  
RMAN> RECOVER DATABASE;
```

- PLUGGABLE DATABASE 子句对单个 PDB 有效：

```
RMAN> BACKUP PLUGGABLE DATABASE hr_pdb, sales_pdb;  
RMAN> RECOVER PLUGGABLE DATABASE hr_pdb;
```

- 使用 CDB\$ROOT 关键字备份、还原、恢复根：

```
RMAN> BACKUP PLUGGABLE DATABASE "CDB$ROOT";
```

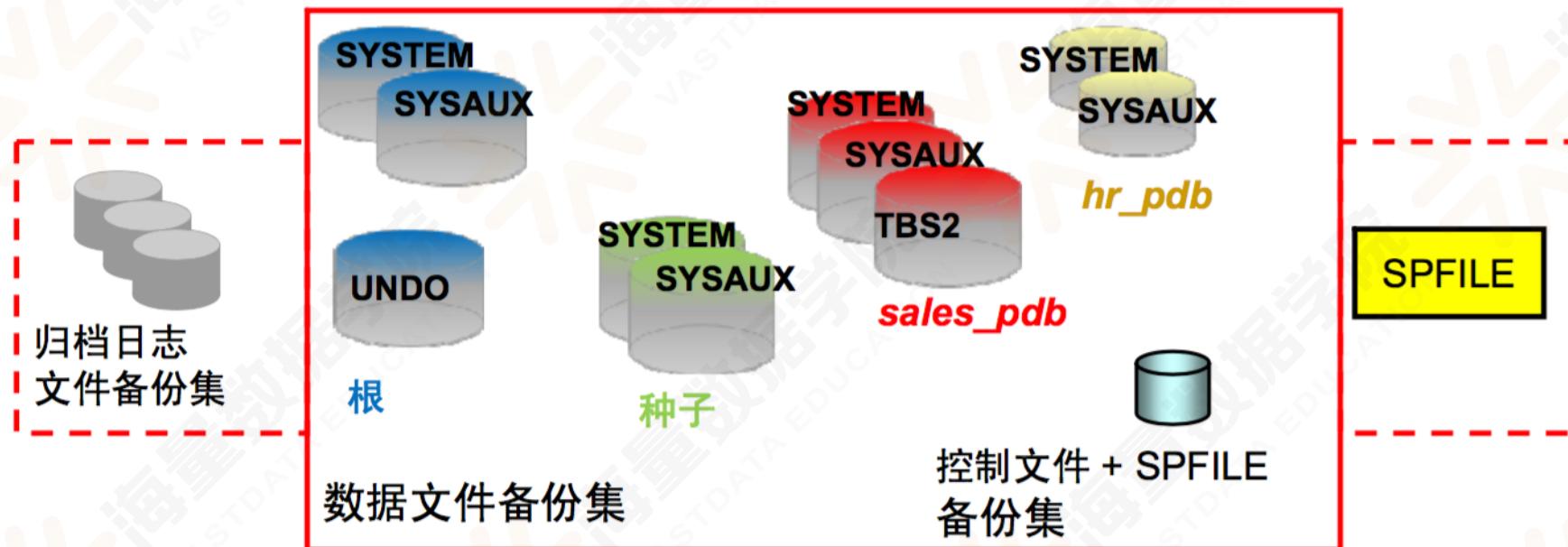
- 使用 PDB 名称限定 PDB 的表空间：

```
RMAN> BACKUP TABLESPACE sales_pdb:tbs2;  
RMAN> RESTORE TABLESPACE sysaux;
```

CDB 备份：整个 CDB 备份

备份 PDB 的所有数据文件和根文件。

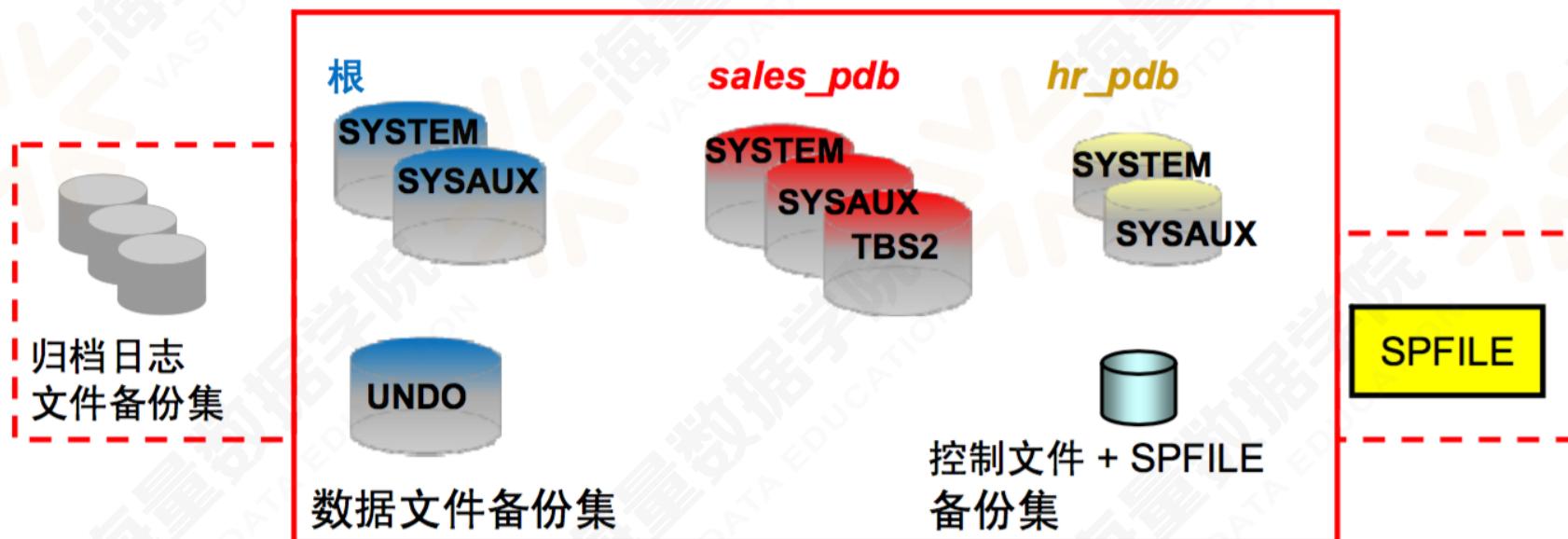
```
RMAN> CONNECT TARGET /  
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
```



CDB 备份：部分 CDB 备份

备份根和/或单独的 PDB。

```
RMAN> CONNECT TARGET /  
RMAN> BACKUP PLUGGABLE DATABASE "CDB$ROOT", sales_pdb;  
RMAN> BACKUP PLUGGABLE DATABASE hr_pdb PLUS ARCHIVELOG;
```

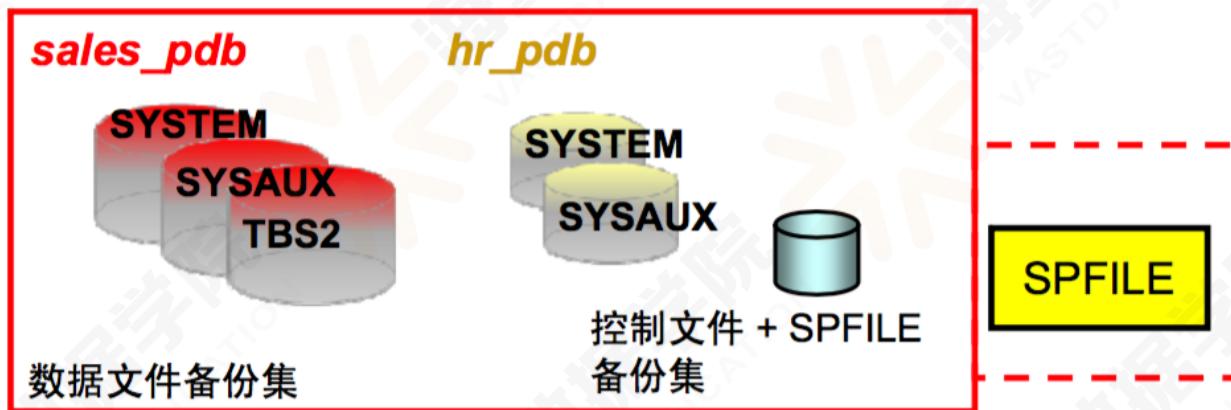


PDB 备份：整个 PDB 备份

使用 RMAN 备份整个 PDB：

```
RMAN> CONNECT TARGET /  
RMAN> BACKUP PLUGGABLE DATABASE sales_pdb;
```

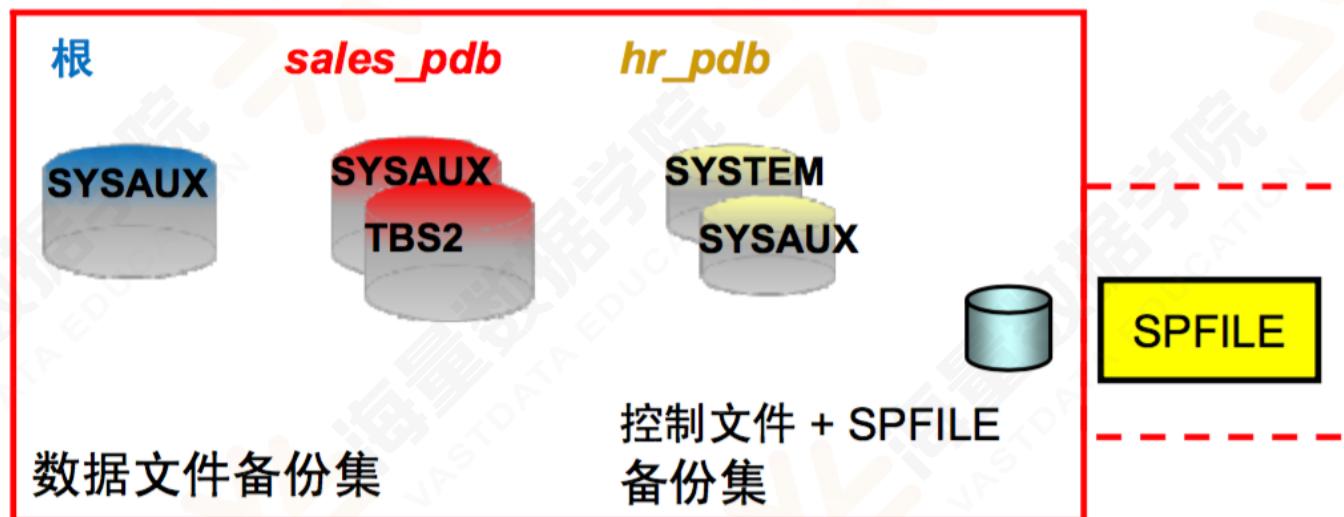
```
RMAN> BACKUP PLUGGABLE DATABASE sales_pdb, hr_pdb;
```



```
RMAN> CONNECT TARGET sys@sales_pdb  
RMAN> BACKUP DATABASE;
```

PDB 备份：部分 PDB 备份

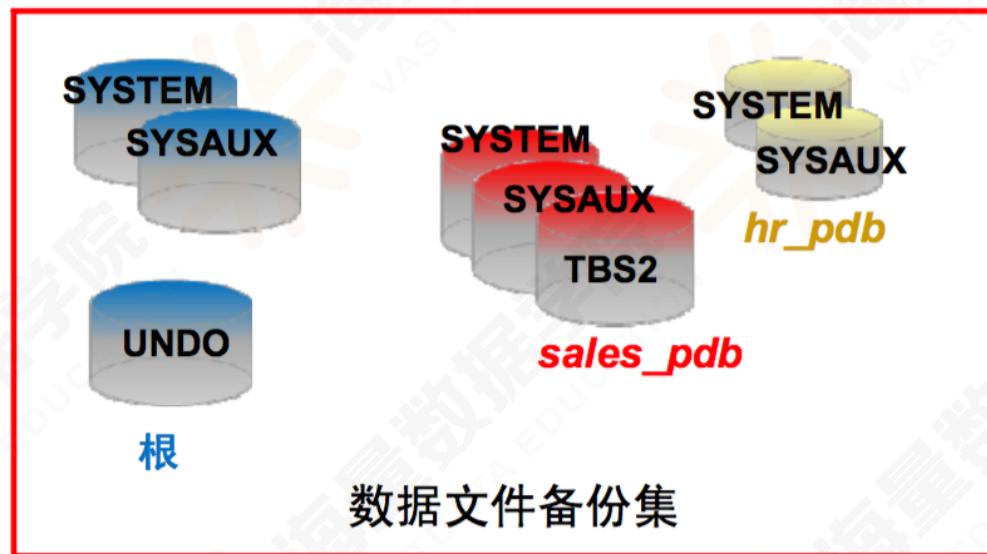
```
RMAN> CONNECT TARGET /
RMAN> REPORT SCHEMA;
RMAN> BACKUP TABLESPACE sales_pdb:tbs2;
RMAN> BACKUP TABLESPACE hr_pdb:system,
      sales_pdb:sysaux;
RMAN> BACKUP TABLESPACE sysaux, hr_pdb:sysaux;
```



PDB 备份：用户管理的热 PDB 备份

执行用户管理的热 PDB 备份。

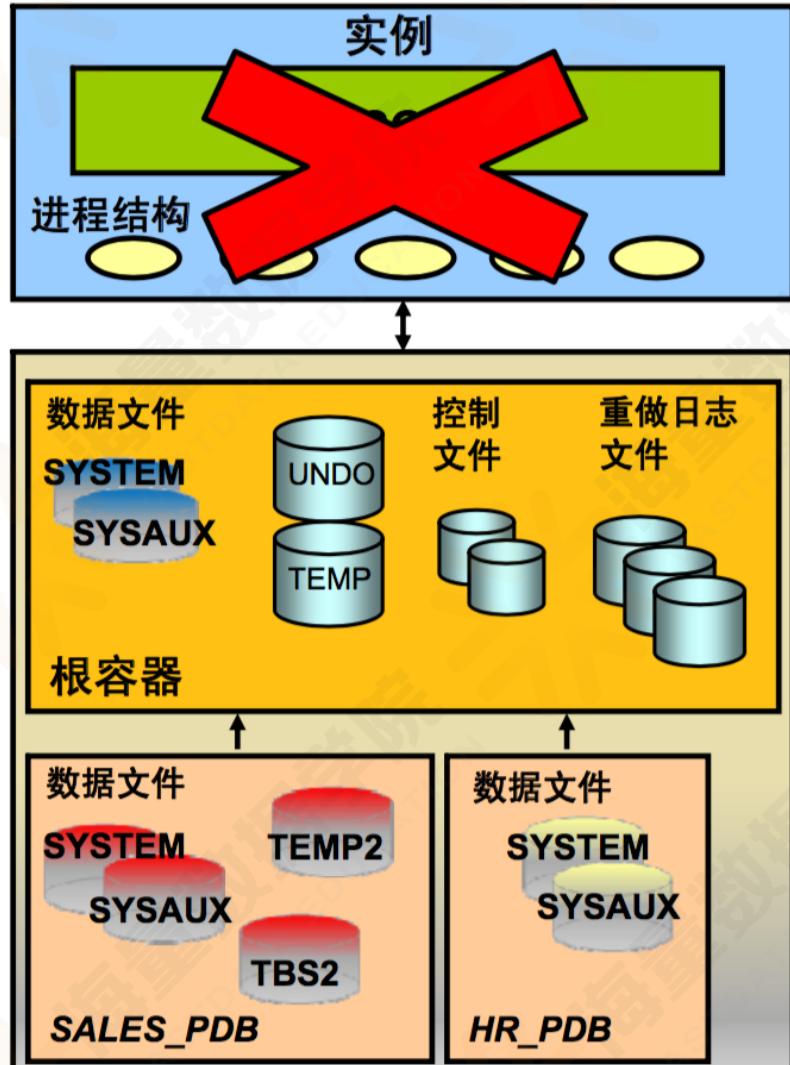
```
SQL> CONNECT system@sales_pdb
SQL> ALTER DATABASE BEGIN BACKUP;
SQL> !cp sales_pdb/datafiles /backup_dir_sales_pdb
SQL> ALTER DATABASE END BACKUP;
```



恢复

- 实例恢复：仅 CDB 级别
- 在 CDB 打开时自动重新创建缺少的临时文件
- 文件丢失或损坏后进行完全介质恢复
 - CDB：与非 CDB 相同
 - 重做日志文件、控制文件
 - 根数据文件
 - PDB 数据文件
 - 表空间：PDB 或 CDB 数据文件
- 文件丢失或损坏后进行不完全介质恢复
 - CDB：过去某个时间点的整个 CDB
 - PDB：过去某个时间点的整个 PDB
 - 任何表空间（SYSTEM、UNDO、SYSAUX 除外）的 TSPITR
- 块恢复：无更改
- 闪回数据库：已装载 CDB

实例故障



无法进行 PDB 实例恢复。

发生实例故障后：

- 连接到根
- 打开根
- 打开 PDB:

– 如果已配置保存的状态，则自动执行

```
SQL> STARTUP;
```

– 手动执行：

```
SQL> STARTUP;
SQL> ALTER PLUGGABLE DATABASE
      ALL OPEN;
```

NOARCHIVELOG 模式

如果数据库处于 NOARCHIVELOG 模式并且数据文件丢失，请执行以下任务：

- 如果实例尚未关闭，请关闭实例。
- 还原整个 CDB（包括所有数据文件和控制文件）。
- 启动实例并打开 CDB 和所有 PDB。

用户必须重新输入自上一次备份以来所做的所有更改。

介质故障：CDB 临时文件恢复

如果缺少一个临时文件，则需要临时空间的 SQL 语句可能会失败：与非 CDB 类似。

```
SQL> CONNECT / AS SYSDBA
SQL> select * from DBA_OBJECTS
      order by 1,2,3,4,5,6,7,8,9,10,11,12,13;
select * from DBA_OBJECTS order by
1,2,3,4,5,6,7,8,9,10,11,12,13
*
ERROR at line 1:
ORA-01565: error in identifying file
'/u01/app/oracle/oradata/CDB1/temp01.dbf'
ORA-27037: unable to obtain file status
Linux Error: 2: No such file or directory
```

- 在 CDB 打开时自动重新创建临时文件
- 也可以手动重新创建

介质故障：PDB 临时文件恢复

如果缺少一个临时文件，则需要临时空间的 SQL 语句可能会失败。

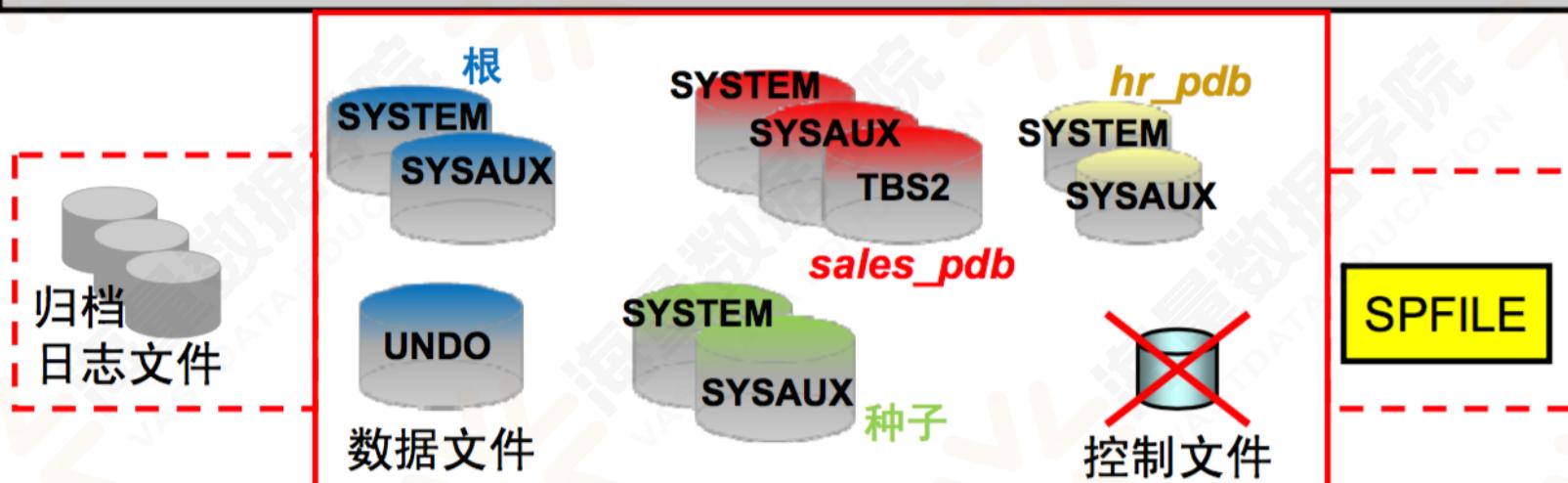
```
SQL> CONNECT local_user@HR_PDB
SQL> select * from my_table order by
1,2,3,4,5,6,7,8,9,10,11,12,13;
select * from my_table order by
1,2,3,4,5,6,7,8,9,10,11,12,13
*
ERROR at line 1:
ORA-01565: error in identifying file
'/u01/app/oracle/oradata/CDB1/HR_PDB/temp2_01.dbf'
ORA-27037: unable to obtain file status
Linux Error: 2: No such file or directory
```

- 在 **PDB 打开时**自动重新创建临时文件
- 也可以手动重新创建

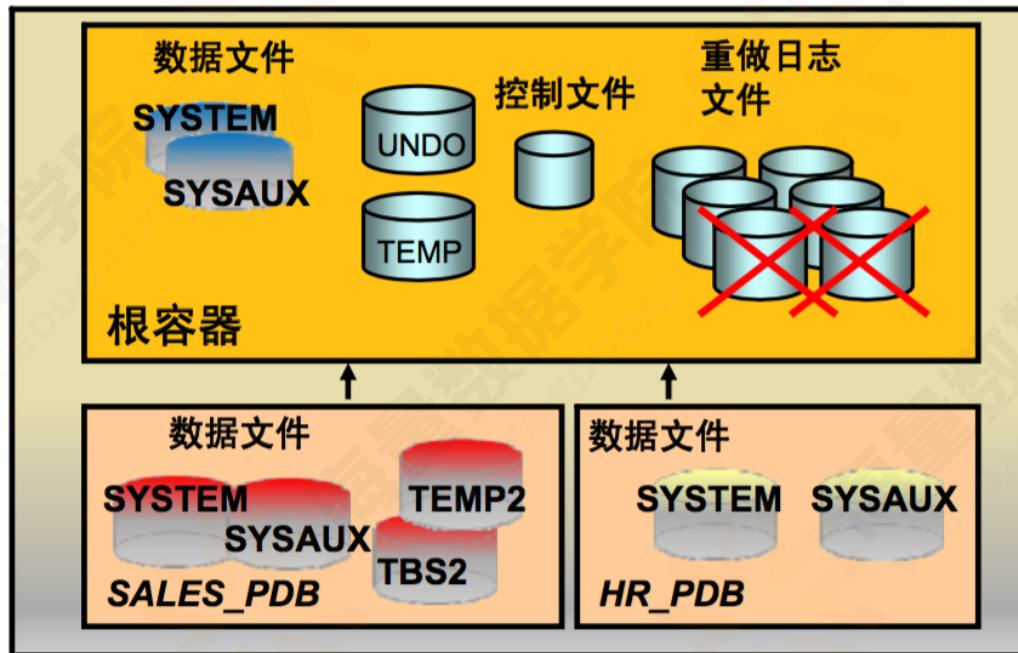
介质故障：控制文件丢失

与非 CDB 类似：已装载 CDB

```
RMAN> CONNECT TARGET /  
RMAN> STARTUP NOMOUNT;  
RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP;  
RMAN> ALTER DATABASE MOUNT;  
RMAN> RECOVER DATABASE;  
RMAN> ALTER DATABASE OPEN RESETLOGS;  
RMAN> ALTER PLUGGABLE DATABASE ALL OPEN;
```



介质故障：重做日志文件丢失



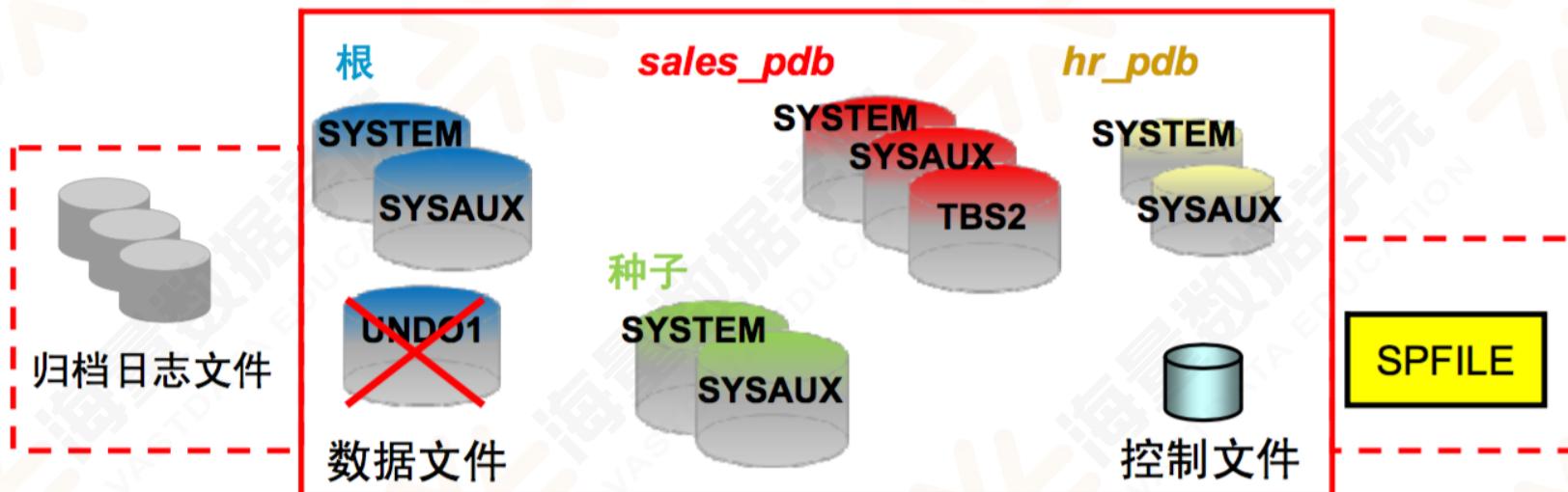
与非 CDB 过程类似：

- 连接到根容器。
- 检查丢失的文件的 STATUS: ACTIVE、INACTIVE、CURRENT。
- 按照与非 CDB 类似的方法继续操作。

介质故障：根 SYSTEM 或 UNDO 数据文件

与非 CDB 类似：已装载 CDB

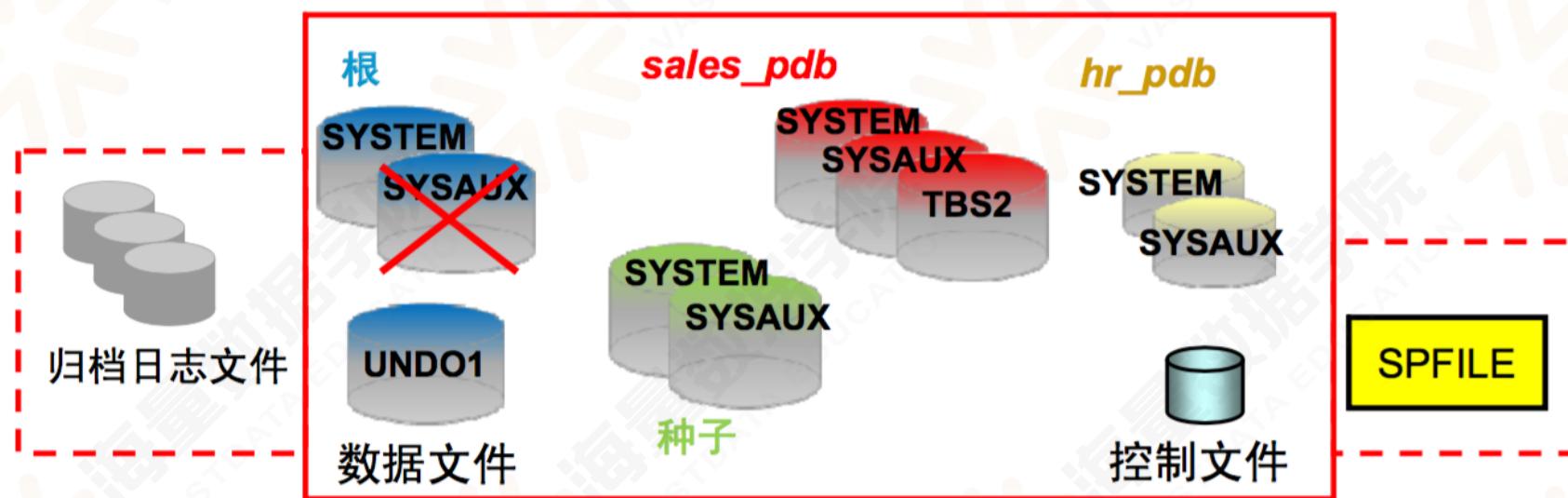
```
RMAN> STARTUP MOUNT;  
RMAN> RESTORE TABLESPACE undo1;  
RMAN> RECOVER TABLESPACE undo1;  
RMAN> ALTER DATABASE OPEN;  
RMAN> ALTER PLUGGABLE DATABASE ALL OPEN;
```



介质故障：根 SYSAUX 数据文件

与非 CDB 类似：表空间 OFFLINE

```
RMAN> ALTER TABLESPACE sysaux OFFLINE IMMEDIATE;  
RMAN> RESTORE TABLESPACE sysaux;  
RMAN> RECOVER TABLESPACE sysaux;  
RMAN> ALTER TABLESPACE sysaux ONLINE;
```

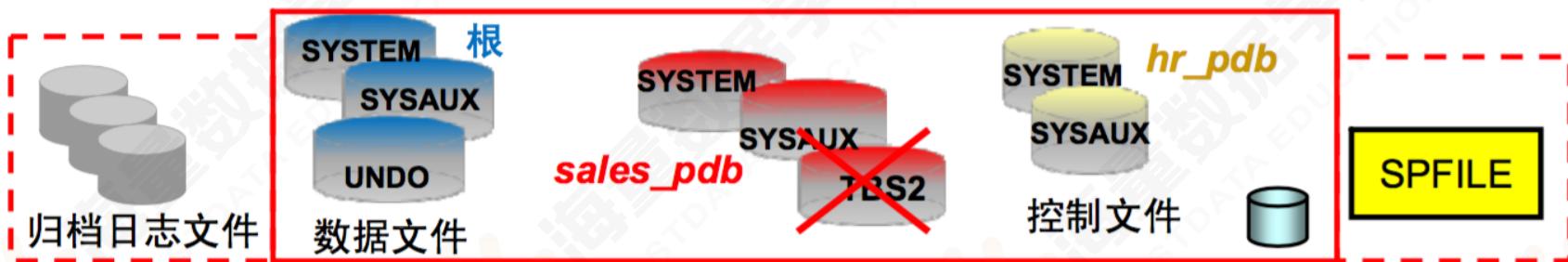


介质故障：PDB 数据文件

与非 CDB 类似：在 PDB 内执行恢复

- 连接到 PDB。
- 将表空间置于 OFFLINE 模式。
- 其他 PDB 不受影响。

```
SQL> CONNECT system@sales_pdb
SQL> ALTER TABLESPACE tbs2 OFFLINE IMMEDIATE;
RMAN> CONNECT TARGET /
RMAN> RESTORE TABLESPACE sales_pdb:tbs2;
RMAN> RECOVER TABLESPACE sales_pdb:tbs2;
SQL> ALTER TABLESPACE tbs2 ONLINE;
```



介质故障：PITR

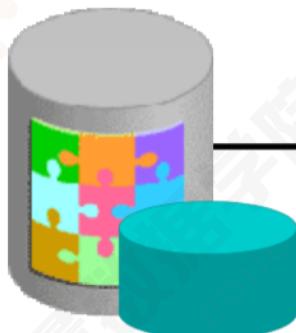
- PDB PITR

```
RMAN> ALTER PLUGGABLE DATABASE PDB1 CLOSE;
RMAN> RUN {
      SET UNTIL SCN = 1851648 ;
      RESTORE pluggable DATABASE pdb1;
      RECOVER pluggable DATABASE pdb1
      AUXILIARY DESTINATION='/u01/app/oracle/oradata';
      ALTER PLUGGABLE DATABASE pdb1 OPEN RESETLOGS;
    }
```

- PDB 表空间 PITR

```
RMAN> RECOVER TABLESPACE PDB1:TEST_TBS
          UNTIL SCN 832972
          AUXILIARY DESTINATION '/tmp/CDB1/reco';
RMAN> ALTER TABLESPACE PDB1:TEST_TBS ONLINE;
```

闪回 CDB



1. 配置 FRA。



2. 设置保留目标。



3. 启用闪回数据库。

```
SQL> SHUTDOWN IMMEDIATE  
SQL> STARTUP MOUNT  
SQL> ALTER DATABASE ARCHIVELOG;  
SQL> ALTER SYSTEM SET  
    DB_FLASHBACK_RETENTION_TARGET=2880 SCOPE=BOTH;  
SQL> ALTER DATABASE FLASHBACK ON;  
SQL> ALTER DATABASE OPEN;
```

如果 CDB 处 ARCHIVELOG 模式，则不需要重新启动该 CDB。

- 执行根闪回时会闪回整个 CDB。
- 任何闪回均不会闪回到早于执行 PDBPITR 时的时间点的时间点。

闪回 CDB

已删除某个公用用户。

1. 闪回 CDB: 已经以独占模式装载 CDB

```
SQL> STARTUP MOUNT  
SQL> FLASHBACK DATABASE TO SCN 53943;
```

2. 要复查更改, 请执行以下操作: 以 READ ONLY 方式打开 CDB 和 PDB

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

3. 要进行最终处理, 请执行以下操作: 根据需要再次闪回并使用 RESETLOGS 打开 CDB。

```
RMAN> SHUTDOWN IMMEDIATE  
RMAN> STARTUP MOUNT  
RMAN> FLASHBACK DATABASE TO SCN 10;  
RMAN> ALTER DATABASE OPEN RESETLOGS;  
RMAN> ALTER PLUGGABLE DATABASE ALL OPEN;
```

特殊情况

- 创建控制文件备份脚本:

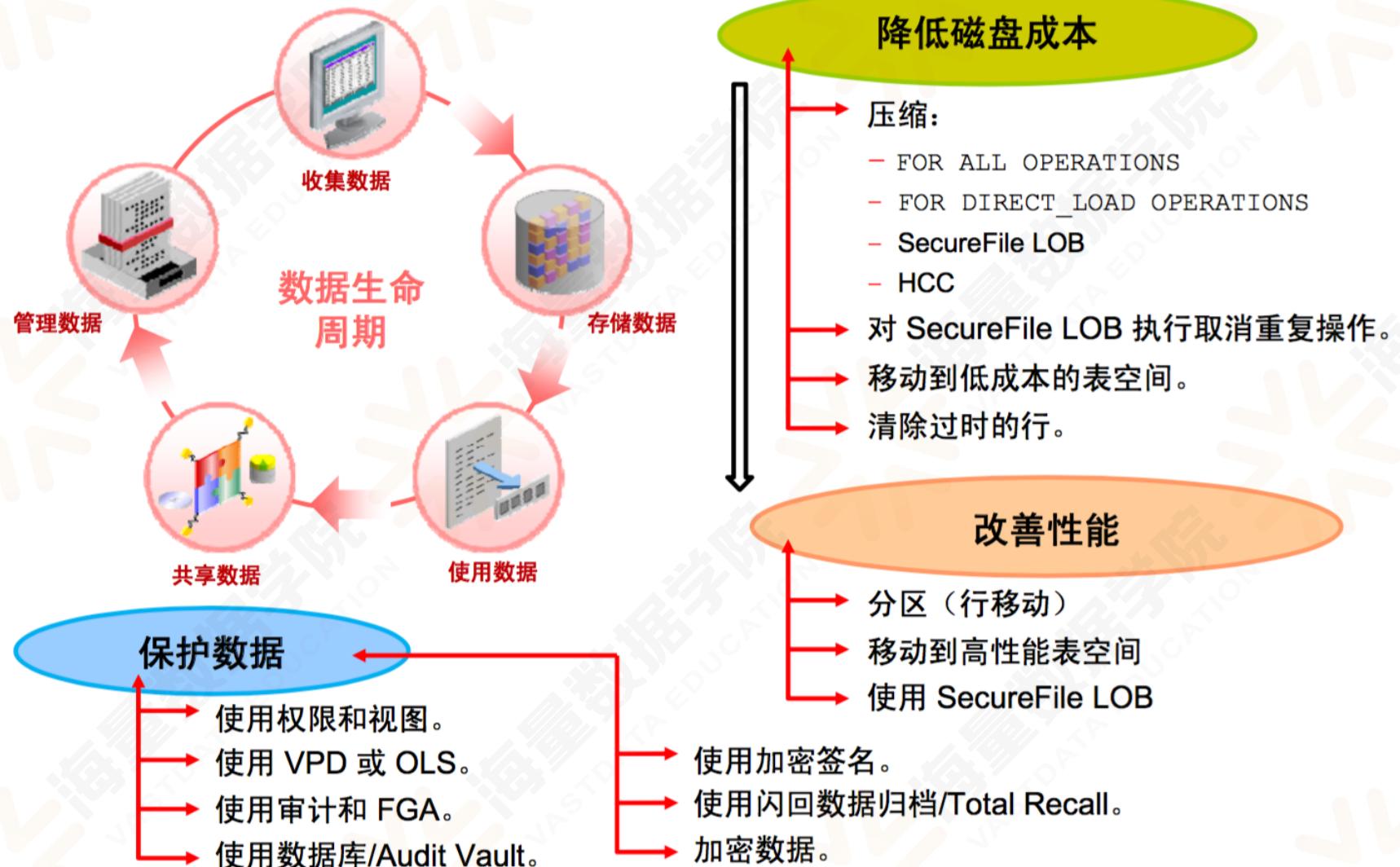
```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

```
CREATE CONTROLFILE ...
  datafile sales_pdb file1
  datafile sales_pdb file1
  ...
  datafile hr_pdb file1 ...;
```

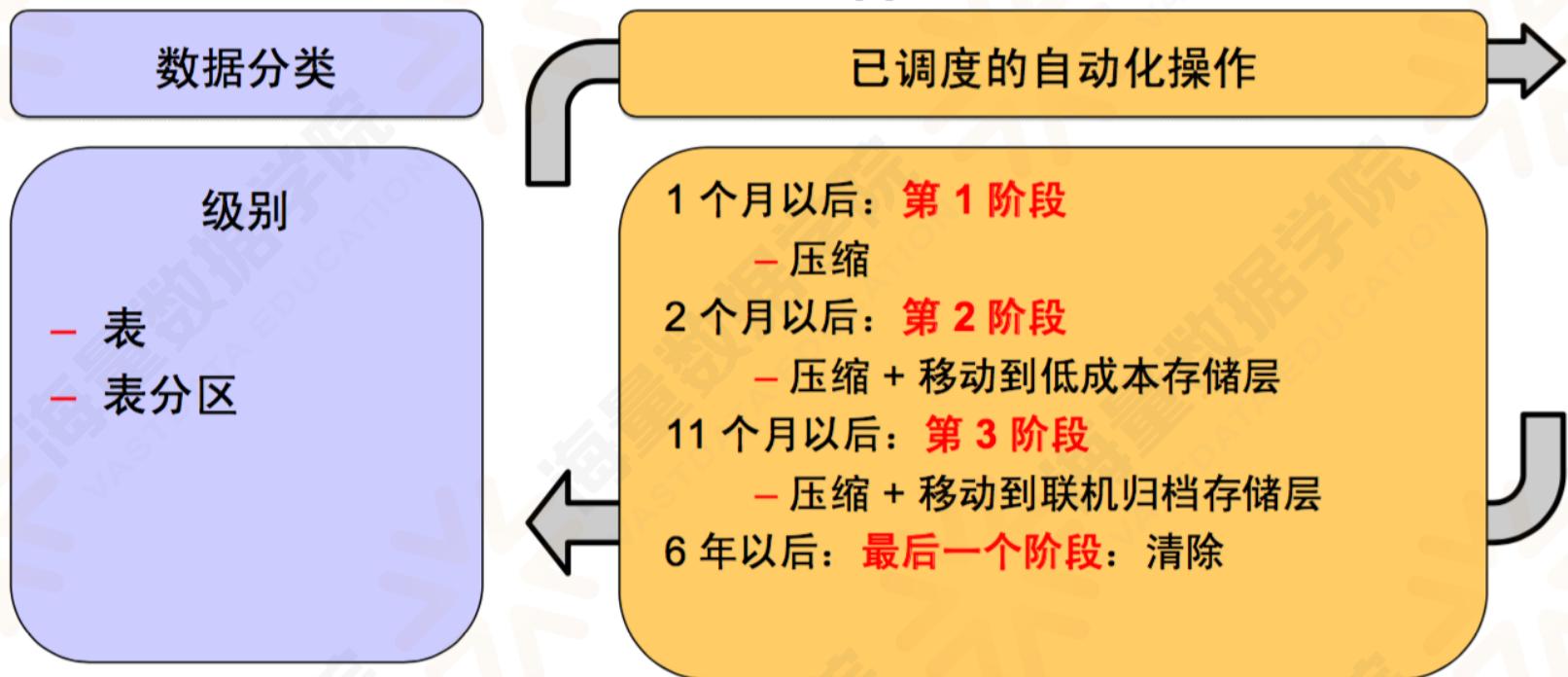
- PDB 关闭操作与 CDB 热备份不兼容:

```
SQL> ALTER DATABASE BEGIN BACKUP;
SQL> ALTER PLUGGABLE DATABASE CLOSE;
ALTER PLUGGABLE DATABASE pdb1_1 close
*
ERROR at line 1:
ORA-01149: cannot shutdown - file 10 has online backup set
ORA-01110: data file 10: '/D1/oradata/cdb1/pdb1_1/users01.dbf'
```

ILM 难题和解决方案



ILM 组件



生命周期定义:

阶段:	当前订单	以前的订单	早期订单	非常早的订单	已过时订单
层:	高性能	高性能	低成本	联机归档	
属性:	无	压缩	压缩	压缩	
保留:	当月	2 个月	11 个月	6 年	清除
时间:	最新数据 → → → 最旧数据				

ILM 难题



解决方案

降低磁盘成本:

- 采用多个不同的存储层, **更高效地管理存储。**
- 使用**压缩级别**。

改善性能:

- **使用不同压缩级别**

自动数据优化:

- 策略可以自动完成以下操作:
 - 根据用户定义的规则进行压缩
 - 在空间不足时移动数据

数据库中归档:

- 行归档
- 期间有效性

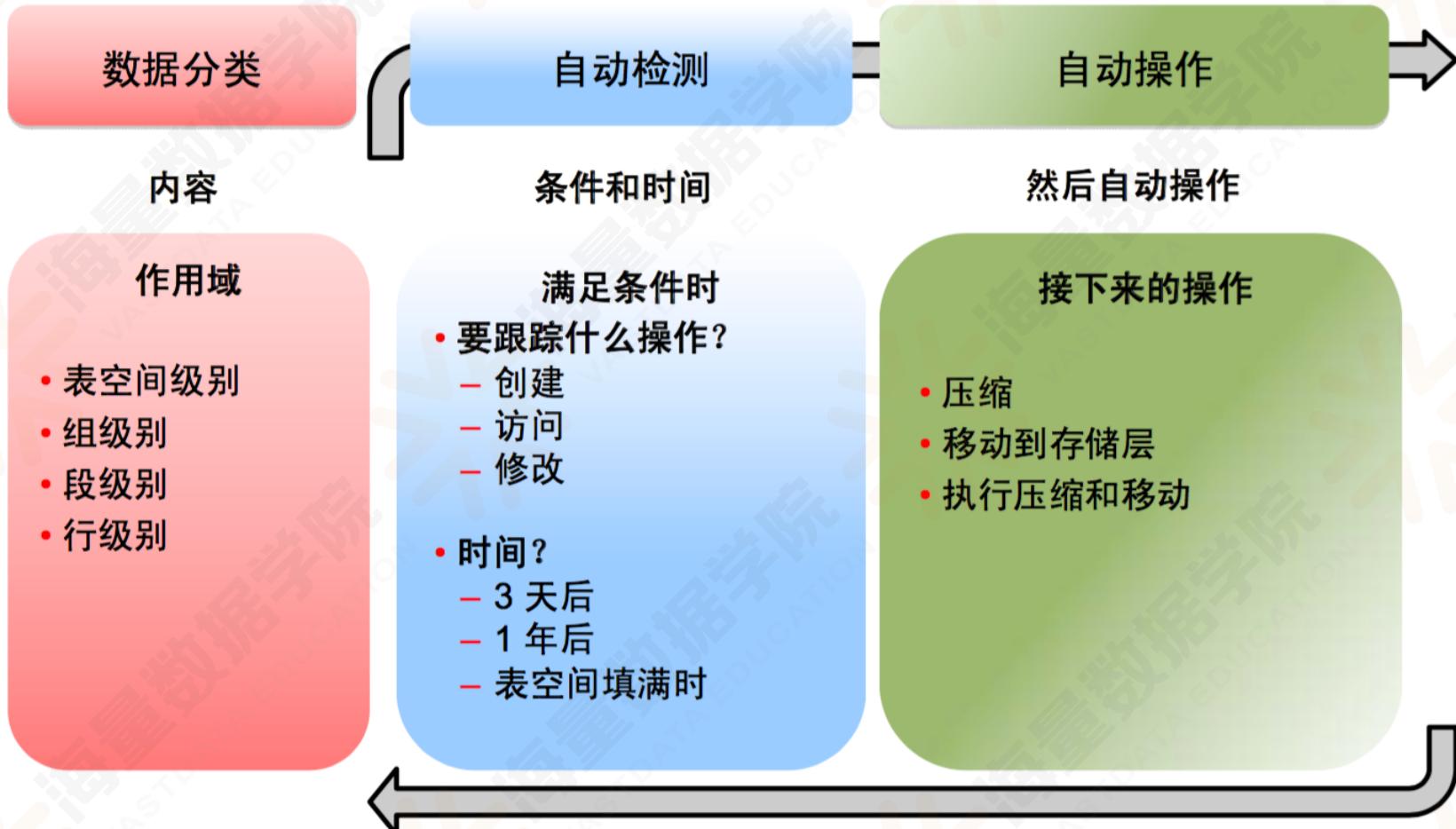
自动数据优化



- 最近插入, 频繁更新
- 高级行压缩 (2-4 倍)
- 在 DRAM 和闪存中高速缓存
- 很少访问, 经常报告
- 高压缩级别 (10 倍)
- **高性能存储**
- 很少访问
- **高压缩级别 (10 倍)**
- **低成本存储**
- 因供长期分析使用或遵守公司策略而保留
- **归档压缩 (15-50 倍)**
- **归档存储 (数据库或磁带)**

组件

热图和自动数据优化



什么是自动数据优化

- ADO 策略自动压缩符合条件的数据。
- ADO 策略根据需要自动移动段。
- ADO 依赖于热图，除非启用热图，否则无法运行。

数据分类级别

活动跟踪和自动数据优化

数据分类

内容

作用域

- 表空间级别
- 组级别
- 段级别：
 - 表/分区/子分区
 - 聚簇表
- 行级别

自动检测

条件和时间

满足条件时

- 要跟踪什么操作?
 - 创建
 - 访问
 - 修改
- 时间?
 - 3 天后
 - 1 年后
 - 表空间填满时

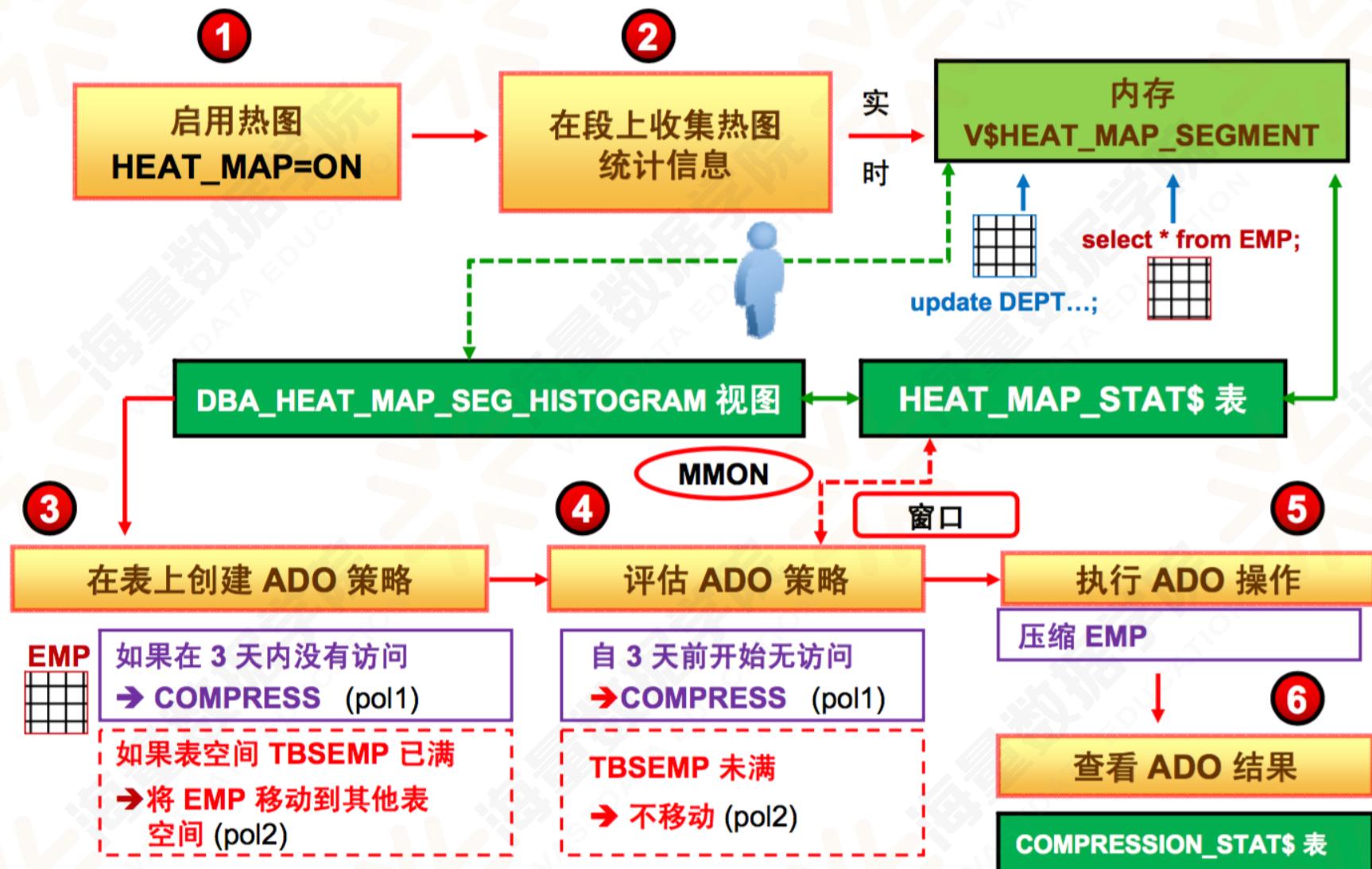
自动操作

然后执行

接下来的操作

- 压缩
 - 类型:
*ROW STORE COMPRESS
ADVANCED ...*
- 移动到其他存储
- 执行压缩和移动

热图和 ADO



启用热图段级别统计信息

1. 启用活动跟踪或热图:

```
SQL> ALTER SYSTEM SET heat_map = ON;
```

- 使用以下视图查看段级别统计信息:
 - DBA_HEAT_MAP_SEG_HISTOGRAM
 - DBA_HEAT_MAP_SEGMENT
 - V\$HEAT_MAP_SEGMENT

```
SQL> SELECT object_name, subobject_name, track_time,  
       segment_write WRI, full_scan FTS, lookup_scan LKP  
  FROM DBA_HEAT_MAP_SEG_HISTOGRAM;
```

OBJECT_NAME	SUBOBJECT_NAME	TRACK_TIME	WRI	FTS	LKP
INTERVAL_SALES		P1 02-JAN-12	YES	YES	NO
INTERVAL_SALES		P2 28-MAR-12	NO	YES	NO
INTERVAL_SALES		P3 28-MAR-12	NO	NO	YES
I_EMPNO		07-dec-12	YES	NO	YES

DBA_HEAT_MAP_SEGMENT 视图

查看最近的段活动跟踪时间：

```
SQL> SELECT object_name,
      segment_write_time WRITE_T, segment_read_time READ_T,
      full_scan FTS_T, lookup_scan LKP_T
    FROM DBA_HEAT_MAP_SEGMENT;
```

OBJECT_NAME	WRITE_T	READ_T	FTS_T	LKP_T
EMP			07-dec-12	
T1	07-dec-12		08-dec-12	
EMPLOYEE	07-dec-12		08-dec-12	08-dec-12
I_EMPNO	07-dec-12			08-dec-12

块级别统计信息

在行修改之后，使用两个新的 PL/SQL 表函数查看块级别统计信息：

- DBMS_HEAT_MAP.BLOCK_HEAT_MAP
- DBMS_HEAT_MAP.EXTENT_HEAT_MAP

```
SQL> SELECT segment_name, tablespace_name, block_id,
  writetime
    FROM table(dbms_heat_map.block_heat_map
      ('SCOTT', 'EMPLOYEE', NULL, 8, 'ASC'));
```

SEGMENT_TABLESPACE_NAME	BLOCK_ID	WRITETIME
EMPLOYEE LOW_COST_STORE	196	12-DEC-12
EMPLOYEE LOW_COST_STORE	197	12-DEC-12
EMPLOYEE LOW_COST_STORE	198	12-DEC-12

区级别统计信息

在行修改之后，视图在区级别聚集：

- 最短修改时间
- 最长修改时间

```
SQL> SELECT segment_name, block_id, blocks,
      max_writetime
    FROM table(dbms_heat_map.extent_heat_map
      ('SCOTT', 'EMPLOYEE'));
```

SEGMENT_	BLOCK_ID	BLOCKS	MAX_WRITETIME
EMPLOYEE	136	8	12-DEC-12 12:58:46
EMPLOYEE	144	8	12-DEC-12 12:58:46
EMPLOYEE	256	128	12-DEC-12 12:58:47
EMPLOYEE	384	128	12-DEC-12 12:58:47

定义自动检测条件

2. 创建 ADO 策略:

- 定义在什么条件下触发操作。
- 定义策略何时生效。



定义自动操作

2. 创建 ADO 策略以定义操作和执行级别。

	无修改	创建	无访问	表空间占满程度	定制策略
压缩（行）	✓				
压缩（段）	✓	✓	✓		✓
压缩（组）	✓	✓	✓		
压缩（表空间）	✓	✓	✓		
存储分层（段）				✓	✓
存储分层（表空间）				✓	

压缩作用域和类型

压缩作用域

- 表空间
- 组
- 段
- 行

压缩类型

- ROW STORE COMPRESS [BASIC]
- ROW STORE COMPRESS ADVANCED
- COLUMN STORE COMPRESS FOR QUERY LOW / HIGH
- COLUMN STORE COMPRESS FOR ARCHIVE LOW / HIGH



创建压缩策略表空间和组

- TABLESPACE 级别压缩策略：

```
SQL> ALTER TABLESPACE tbs1 DEFAULT ILM ADD POLICY  
      ROW STORE COMPRESS ADVANCED  
      SEGMENT AFTER 30 DAYS OF NO ACCESS;
```

- GROUP 级别压缩策略：

```
SQL> ALTER TABLE tab1 ILM ADD POLICY  
      ROW STORE COMPRESS ADVANCED  
      GROUP AFTER 90 DAYS OF NO MODIFICATION;
```

```
SQL> ALTER TABLE tab2 MODIFY PARTITION p1 ILM ADD POLICY  
      COLUMN STORE COMPRESS FOR ARCHIVE HIGH  
      GROUP AFTER 6 MONTHS OF CREATION;
```

创建压缩策略段和行

- **SEGMENT** 级别压缩策略:

```
SQL> ALTER TABLE tab4 ILM ADD POLICY  
        COLUMN STORE COMPRESS FOR QUERY HIGH  
        SEGMENT AFTER 90 DAYS OF NO MODIFICATION;
```

```
SQL> ALTER TABLE tab5 ILM ADD POLICY  
        COLUMN STORE COMPRESS FOR ARCHIVE HIGH  
        SEGMENT AFTER 6 MONTHS OF NO ACCESS;
```

```
SQL> ALTER TABLE tab6 ILM ADD POLICY  
        ROW STORE COMPRESS  
        SEGMENT AFTER 2 YEARS OF CREATION;
```

- **ROW** 级别压缩策略:

```
SQL> ALTER TABLE tab1 ILM ADD POLICY  
        ROW STORE COMPRESS ADVANCED  
        ROW AFTER 30 DAYS OF NO MODIFICATION;
```

创建存储分层策略

移动级别

— 段级别

存储分层

— 表空间

```
SQL> ALTER TABLE tab6 MODIFY PARTITION p1 ILM ADD POLICY  
TIER TO low_cost_storage [ON my_cond];
```

源表空间占满程度阈值：

```
SQL> SELECT * FROM dba_ilmparameters;
```

NAME	VALUE
TBS PERCENT USED	85
TBS PERCENT FREE	25

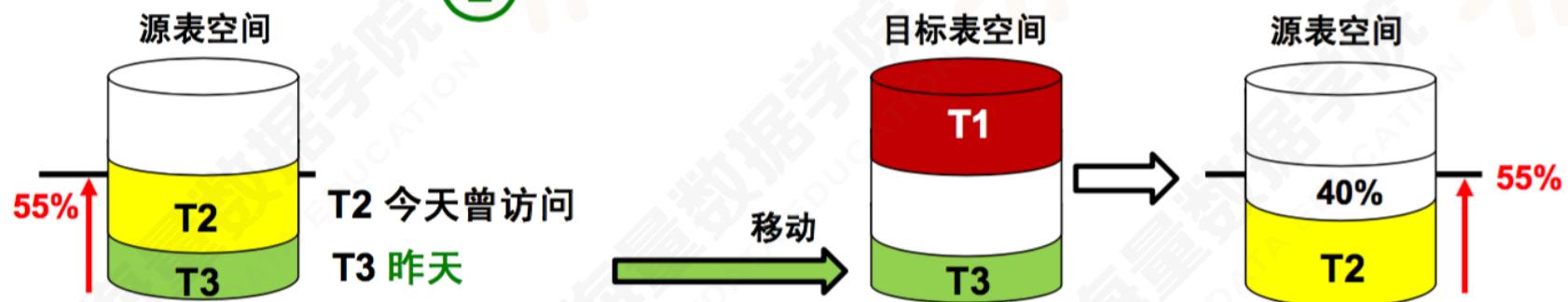
```
SQL> EXEC  
DBMS_ILM_ADMIN.CUSTOMIZE_ILM(DBMS_ILM_ADMIN.TBS_PERCENT_FREE,15)
```

存储分层：优先级

①



②



存储分层: READ ONLY

READ ONLY 层存储:

```
SQL> ALTER TABLE tab7 ILM ADD POLICY  
          TIER TO tablespace_tbs READ ONLY  
          SEGMENT ON CUSTOM_ILM_RULES;
```

```
SQL> ALTER TABLE sales MODIFY PARTITION HY_2010 ILM ADD POLICY  
          TIER TO tablespace_tbs READ ONLY  
          SEGMENT AFTER 90 DAYS OF CREATION;
```

依赖于函数的策略

定制操作：

- 创建一个返回 TRUE / FALSE 的函数：

```
SQL> CREATE FUNCTION CUSTOM_ILM_RULES  
      (objn IN NUMBER)  
      RETURN BOOLEAN ...  
      /
```

- 在策略声明中使用该函数：

```
SQL> ALTER TABLE EMP    ILM ADD POLICY  
          ROW STORE COMPRESS ADVANCED  
          SEGMENT ON CUSTOM_ILM_RULES;
```

```
SQL> ALTER TABLE sales MODIFY PARTITION before_jan_2005  
          ILM ADD POLICY  
          TIER TO history ON CUSTOM_ILM_RULES;
```

一个段的多个 SEGMENT 策略

加载新数据: 周期开始

1 天以后: 高级行压缩

1 个月以后: Query High 压缩

1 年以后: 移动到低成本表空间

2 年以后: 周期结束: 清除

```
SQL> ALTER TABLE t1 ILM ADD POLICY ROW STORE COMPRESS ADVANCED  
SEGMENT AFTER 1 DAY OF NO MODIFICATION;
```

```
SQL> ALTER TABLE t1 ILM ADD POLICY  
COLUMN STORE COMPRESS FOR QUERY HIGH  
SEGMENT AFTER 1 MONTH OF NO MODIFICATION;
```

```
SQL> ALTER TABLE t1 ILM ADD POLICY  
TIER TO history ON t1_check_after_one_year;
```

```
SQL> ALTER TABLE t1 ILM ADD POLICY COLUMN STORE COMPRESS FOR  
ARCHIVE HIGH SEGMENT AFTER 1 YEAR OF NO ACCESS;
```

```
SQL> DROP TABLE t1 PURGE;
```



一个段只能有一个 ROW 策略

- 压缩类型 (Row Store Compress Advanced) 和作用域 (row):
每个对象只有一个

```
SQL> ALTER TABLE t2 ILM ADD POLICY ROW STORE COMPRESS ADVANCED  
ROW AFTER 2 DAYS OF NO MODIFICATION;
```

```
SQL> ALTER TABLE t2 ILM ADD POLICY  
ROW STORE COMPRESS ADVANCED  
ROW AFTER 6 MONTHS OF NO MODIFICATION;
```

- 段级别策略不覆盖行级别策略

```
SQL> ALTER TABLE t2 ILM ADD POLICY COLUMN STORE COMPRESS FOR  
QUERY HIGH SEGMENT AFTER 1 YEAR OF NO MODIFICATION;
```

```
SQL> ALTER TABLE t2 ILM ADD POLICY  
COLUMN STORE COMPRESS FOR ARCHIVE HIGH  
SEGMENT AFTER 6 YEARS OF NO ACCESS;
```

策略继承

子级别策略会覆盖具有相同操作的父级别策略。

表级别策略：

- 覆盖表空间级别策略
- 在分区级别继承

继承是对策略中不同操作的补充。

```
SQL> select POLICY_NAME "POLICY", OBJECT_NAME,  
           SUBOBJECT_NAME "SUBOBJECT", OBJECT_TYPE, INHERITED_FROM  
      FROM DBA_ILMOBJECTS;
```

POLICY	OBJECT_NAME	SUBOBJECT	OBJECT_TYPE	INHERITED_FROM
P281	T1		TABLE	TABLESPACE
P341	SALES		TABLE	POLICY NOT INHERITED
P341	SALES	SALES_1994	TABLE PARTITION	TABLE
P341	SALES	SALES_1996	TABLE PARTITION	TABLE
P350	T3		TABLE	POLICY NOT INHERITED
P360	SALES	SALES_1995	TABLE PARTITION	POLICY NOT INHERITED
P610	T4		TABLE	POLICY NOT INHERITED
P281	T5		TABLE	TABLESPACE

显示策略 DBA_ILMPOLICIES / DBA_ILMDATAMOVEMENTPOLICIES

```
SQL> SELECT * FROM DBA_ILMPOLICIES;
```

POLICY_NAME	POLICY_TYPE	TABLESPACE	ENABLED
P281	DATA MOVEMENT		YES
P381	DATA MOVEMENT		YES
P400	DATA MOVEMENT	ITB	YES

```
SQL> SELECT policy_name, action_type,  
           compression_level, tier_tablespace TBS  
      FROM DBA_ILMDATAMOVEMENTPOLICIES;
```

POLICY_NAME	ACTION_TYPE	COMPRESSION_LEVEL	TBS
P281	COMPRESSION	QUERY HIGH	
P381	COMPRESSION	ADVANCED	
P400	STORAGE		ITB

显示策略

DBA_ILMDATAMOVEMENTPOLICIES

```
SQL> SELECT policy_name POL, action_type, scope,
           condition_type, condition_days
      FROM DBA_ILMDATAMOVEMENTPOLICIES;
```

POL	ACTION_TYPE	SCOPE	CONDITION_TYPE	CONDITION_DAYS
P341	COMPRESSION	GROUP	LAST MODIFICATION TIME	90
P342	COMPRESSION	ROW	LAST MODIFICATION TIME	30
P361	COMPRESSION	GROUP	LAST ACCESS TIME	180
P381	COMPRESSION	SEGMENT	LAST MODIFICATION TIME	90
P401	COMPRESSION	SEGMENT	LAST ACCESS TIME	180
P402	STORAGE	SEGMENT		0
P461	COMPRESSION	SEGMENT	USER DEFINED	0
P462	STORAGE	SEGMENT	USER DEFINED	0
P482	COMPRESSION	SEGMENT	LAST MODIFICATION TIME	30
P681	COMPRESSION	GROUP	CREATION TIME	180

准备评估和执行

段级别策略	行级别策略
在维护窗口中	由 MMON 每隔 15 分钟定期执行

- 更改 ADO 策略评估的间隔。

```
SQL> EXEC DBMS_ILM_ADMIN.CUSTOMIZE_ILM(-  
          DBMS_ILM_ADMIN.EXECUTION_INTERVAL,1)
```

- 使用 **DBMS_ILM** 程序包可以通过两种方式执行 ADO:
 - 在一组对象上立即执行 ADO 任务。

```
SQL> VAR v_taskid NUMBER  
SQL> EXEC DBMS_ILM.EXECUTE_ILM (owner =>'SCOTT', object_name  
=>'EMPLOYEE', task_id =>:v_taskid)
```

- 定制对象组，然后再执行 ADO 任务。
- 停止执行。

```
SQL> EXEC DBMS_ILM.STOP_ILM(245321)
```

定制评估和执行

- 使用 DBMS_ILM 程序包和过程来：
 - 定制一组要评估的对象
 - 调度 ADO 任务的执行

PREVIEW_ILM
→ 任务未激活

**ADD_TO_ILM /
REMOVE_FROM_ILM**
→ 任务未激活

EXECUTE_ILM_TA
→ 任务激活

DBA_ILMTASK
→ 任务完成

```
SQL> EXEC DBMS_ILM.PREVIEW_ILM (TASK_ID => :v_taskid, -  
ILM_SCOPE => DBMS_ILM.SCPE_SCHEMA)
```

```
SQL> EXEC DBMS_ILM.EXECUTE_ILM_TASK (TASK_ID => 26382, -  
EXECUTION_MODE      => DBMS_ILM.SCHEDULE_IMMEDIATE, -  
EXECUTION_SCHEDULE  => DBMS_ILM.ILM_EXECUTION_OFFLINE)
```

```
SQL> SELECT task_id, state FROM dba_ilmtasks;
```

TASK_ID	STATE
26482	ACTIVE

监视评估和执行

- DBA_ILMTASKS
- DBA_ILMEVALUATIONDETAILS
- DBA_ILMRESULTS

```
SQL> SELECT * FROM dba_ilmtasks;
```

TASK_ID	TASK_OWNER	STATE	CREATION_TIME	START_TIME	COMPLETION_TIME
18566	SCOTT	INACTIVE	12-DEC-12		
18542	SCOTT	ACTIVE	11-DEC-12		12-DEC-12

```
SQL> SELECT task_id, policy_name, object_name,  
      Selected_for_execution, job_name  
    FROM dba_ilmevaluationdetails;
```

TASK_ID	POLICY_NAME	OBJECT_NAME	SELECTED_FOR_EXECUTION	JOB_NAME
18762	P281	EMPLOYEE	POLICY DISABLED	
18542	P281	EMPLOYEE	SELECTED FOR EXECUTION	ILMJOB5002
18862	P301	EMPLOYEE	PRECONDITION NOT SATISFIED	

ADO DDL

- 启用对象上的所有策略:

```
SQL> ALTER TABLE tab1 ILM ENABLE_ALL;
```

```
SQL> ALTER TABLE tab2 MODIFY PARTITION p1 ILM ENABLE_ALL;
```

- 禁用对象上的所有策略:

```
SQL> ALTER TABLE tab1 ILM DISABLE_ALL;
```

- 删除对象上的所有策略:

```
SQL> ALTER TABLE tab1 ILM DELETE_ALL;
```

```
SQL> ALTER TABLE tab1 MODIFY PARTITION p1 ILM DELETE_ALL;
```

打开和关闭 ADO

- 关闭 ADO，但不禁用或删除所有策略：

```
SQL> exec DBMS_ILM_ADMIN.DISABLE_ILM
```

- 为所有策略重新打开 ADO：

```
SQL> exec DBMS_ILM_ADMIN.ENABLE_ILM
```

- 查看 ADO 参数：

```
SQL> select * from DBA_ILMPARAMETERS where name='ENABLED';
```

NAME	VALUE
ENABLED	1

停止活动跟踪和清除热图统计信息

- 停止会话中的活动跟踪:

```
SQL> ALTER SESSION SET HEAT_MAP = OFF;
```

- 在实例级别停止活动跟踪:

```
SQL> ALTER SYSTEM SET HEAT_MAP = OFF;
```

- 清除热图统计信息:

```
SQL> exec DBMS_ILM_ADMIN.CLEAR_HEAT_MAP_ALL;
```

活动跟踪的特定情况

- 从具有 ADO 策略的表创建 NEW_EMP 表:

```
SQL> CREATE TABLE new_emp AS SELECT * FROM emp;
```

- NEW_EMP 表不继承 ADO EMP 策略。

- 行级别 ADO 策略作业在结束前失败。
- 段级别 ADO 作业在结束前失败。



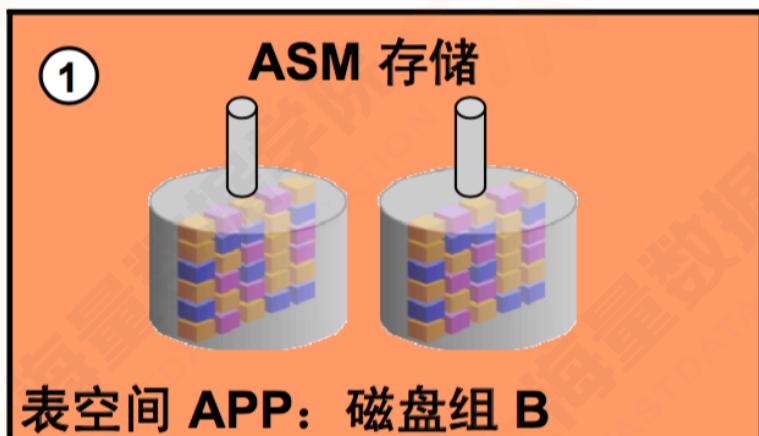
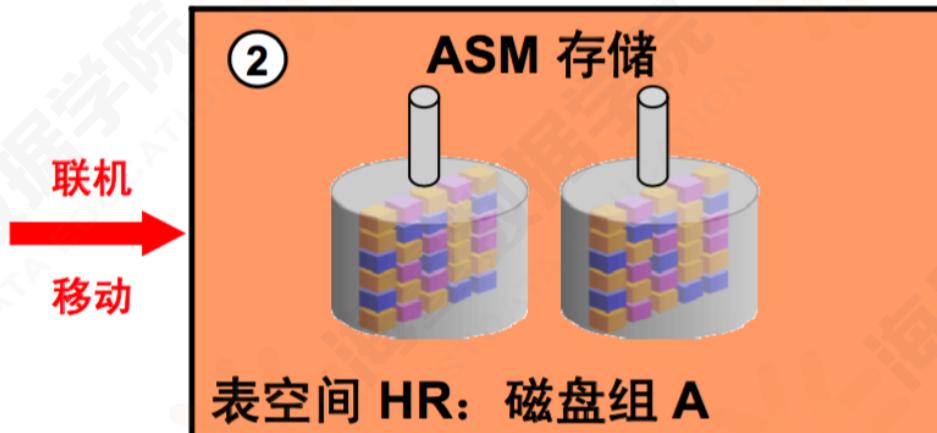
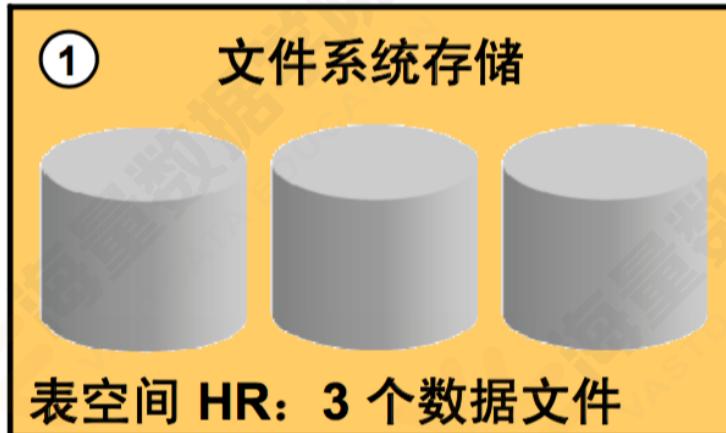
- COMPRESS 属性: 与 ADO 压缩策略兼容



```
SQL> CREATE TABLE tabcomp (c number) ROW STORE COMPRESS;  
SQL> ALTER TABLE tabcomp ILM ADD POLICY  
      ROW STORE COMPRESS ADVANCED  
      ROW AFTER 6 DAYS OF NO MODIFICATION;
```

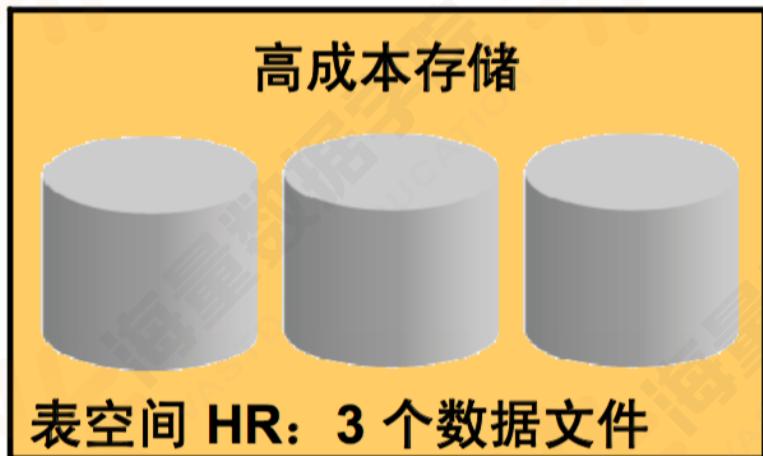
联机移动数据文件

将**数据文件**联机移动到其他类型的存储系统上。

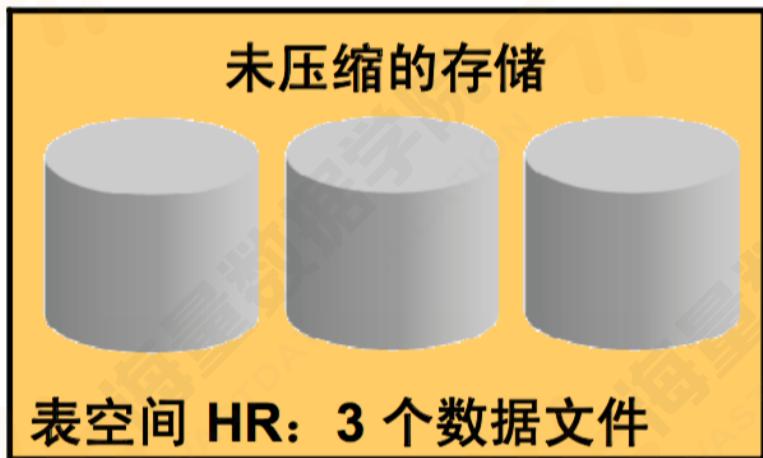
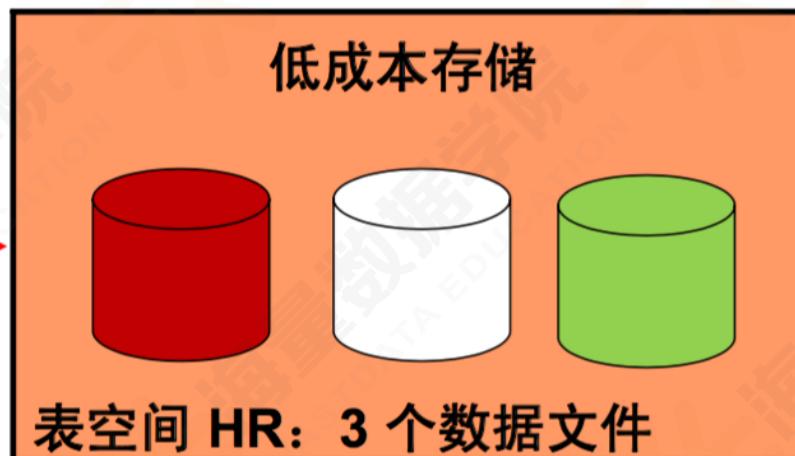


压缩

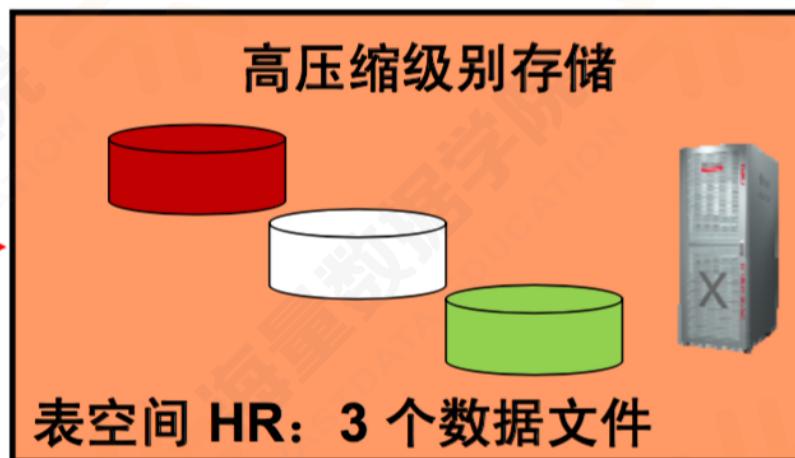
将**数据文件**联机移动到较低成本的存储：



联机
移动



联机
移动



REUSE 和 KEEP

数据库无需关闭 (SHUTDOWN) / 数据文件无需脱机 (OFFLINE)

```
SQL> alter database rename file '/u01/t1.f' to '/u02/t1.f';  
...  
ORA-01511: error in renaming log/data files  
ORA-01121: cannot rename database file 5 - file is in use or  
recovery
```

11g

```
SQL> ALTER DATABASE MOVE datafile '/disk1/myexample01.dbf'  
      TO '/disk2/myexample01.dbf' REUSE;
```

12.1

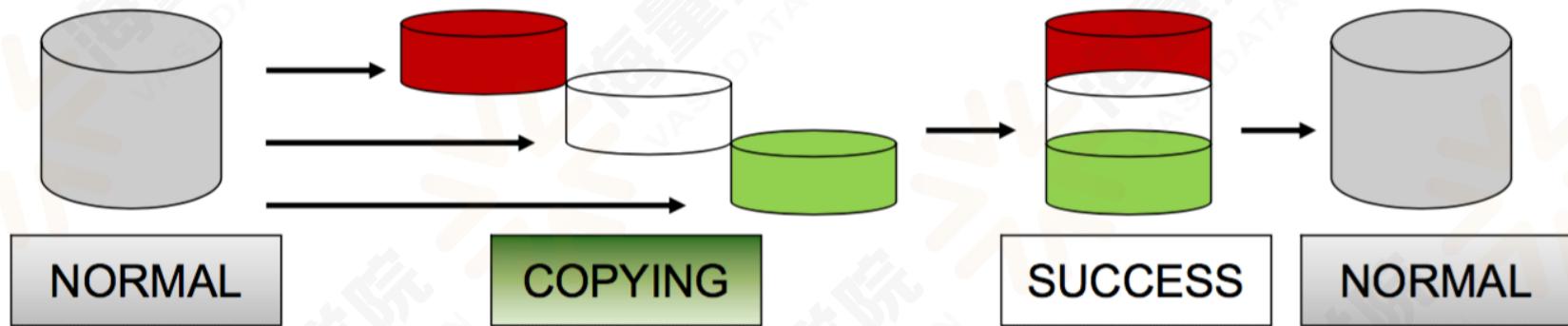
```
SQL> ALTER DATABASE MOVE datafile '/disk1/myexample01.dbf'  
      TO '+DiskGroup2' KEEP;
```

```
SQL> ALTER DATABASE MOVE datafile 5 TO '+DiskGroup3' ;
```

状态

V\$SESSION_LONGOPS 中的进度：

- 移动的每个文件都对应一行
- 迄今为止移动的块数



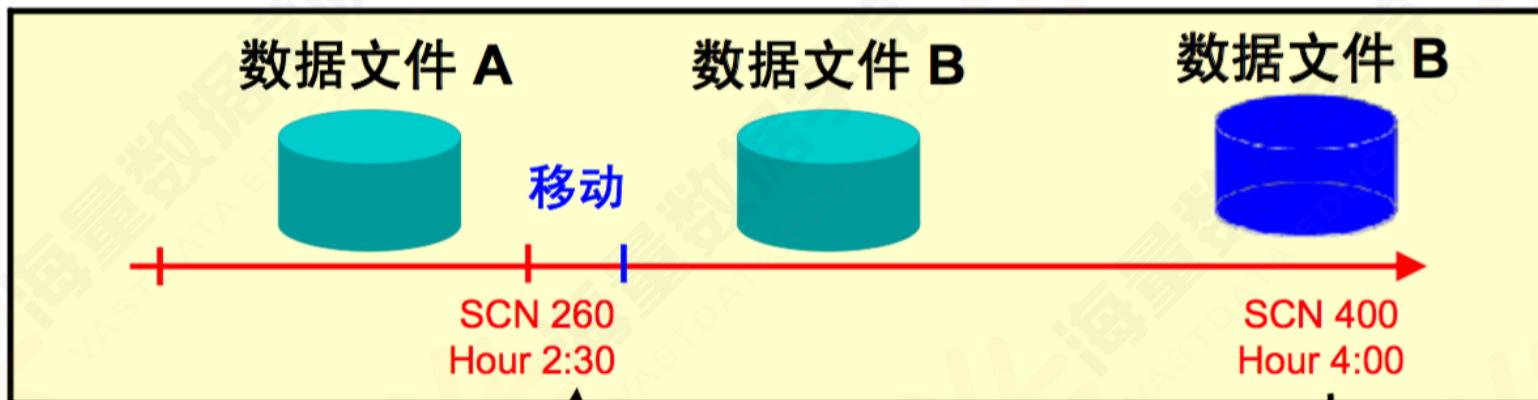
兼容性

不兼容:	<ul style="list-style-type: none">• 数据文件 OFFLINE• 并发 FLASHBACK DATABASE 操作• 并发介质恢复• data file RESIZE (收缩) 操作
兼容:	<ul style="list-style-type: none">• 块介质恢复• 成为 READ ONLY 或 READ WRITE 表空间• data file RESIZE (扩展)• 联机备份



闪回数据库

将数据库闪回到 SCN 260:

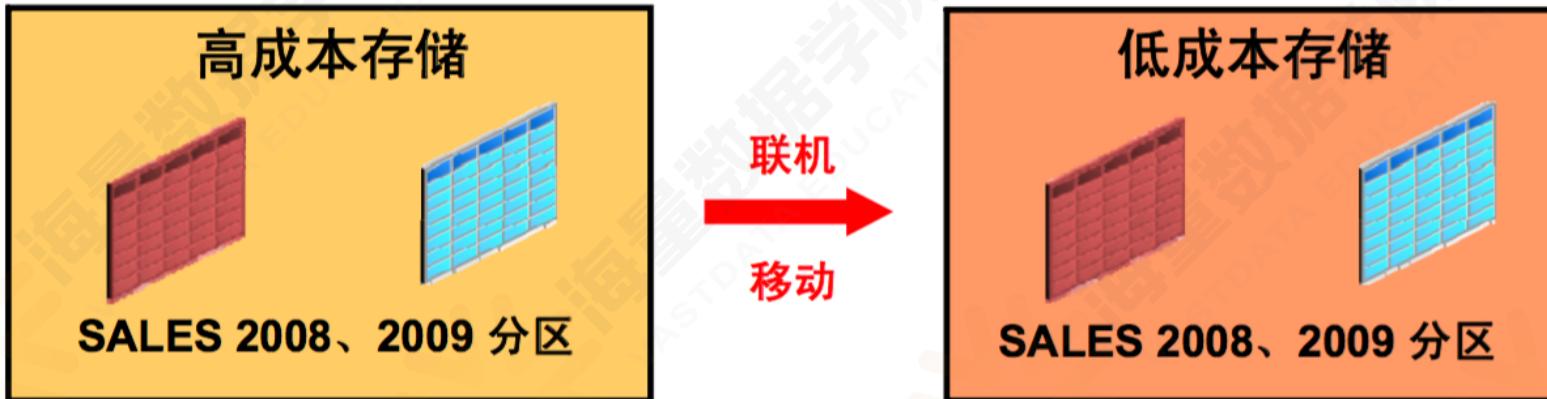


闪回数据库之后:

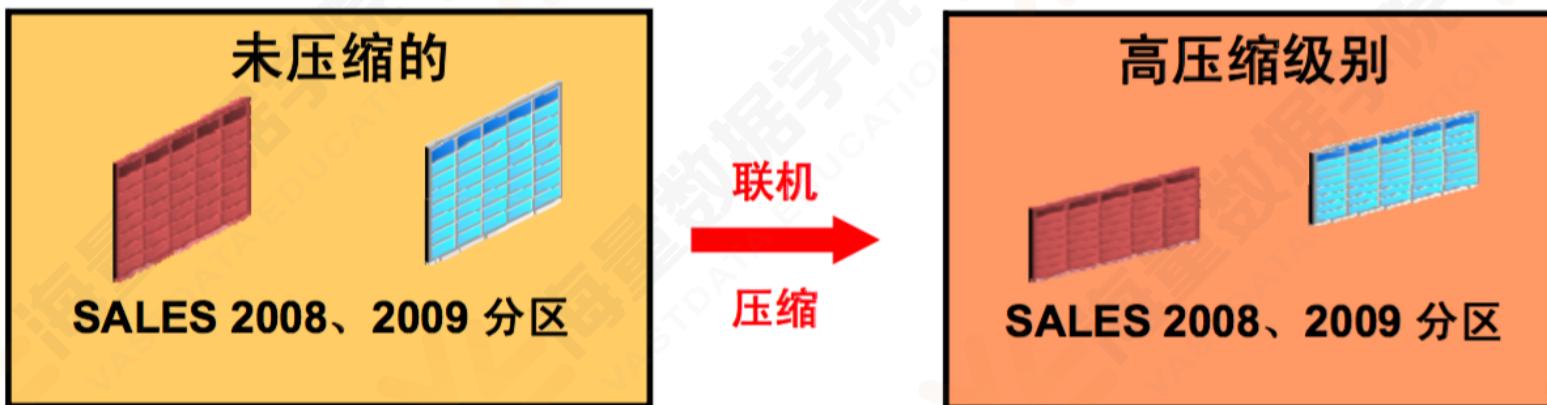


联机移动分区

- 在极少访问或更新分区和子分区的情况下，将**分区**和**子分区****联机移动/分离/合并**到低成本存储

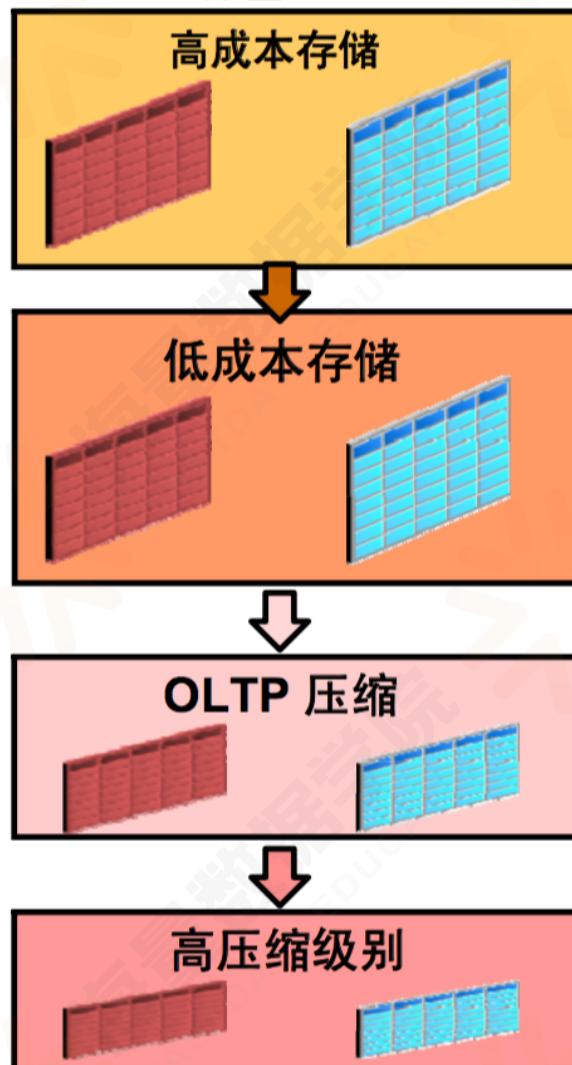


- 联机压缩分区和子分区**



联机移动分区：优点

SALES 旧分区



- 允许 **DML**, 而非 **DDL**
- 将分区联机移动、分离或合并到低
成本存储
- **保留全局和本地索引**

```
SQL1> UPDATE ORDERS  
      SET SAL_ORD_P1=SAL*2  
      WHERE DATE_P1='2001';
```

```
SQL2> ALTER TABLE ORDERS  
      MOVE PARTITION ORD_P1  
      TABLESPACE lowtbs  
      UPDATE INDEXES ONLINE;
```

- **联机压缩分区**

```
SQL> ALTER TABLE ORDERS  
      MOVE PARTITION ORD_P1  
      ROW STORE COMPRESS  
      UPDATE INDEXES ONLINE;
```

联机移动分区：压缩

MOVE 也用于压缩。

- ROW STORE COMPRESS [BASIC]

```
SQL> ALTER TABLE ORDERS MOVE PARTITION ORD_P1  
        ROW STORE COMPRESS UPDATE INDEXES ONLINE;
```

- ROW STORE COMPRESS ADVANCED

```
SQL> ALTER TABLE ORDERS MOVE PARTITION ORD_P1  
        ROW STORE COMPRESS ADVANCED UPDATE INDEXES ONLINE;
```

- COLUMN STORE COMPRESS

```
SQL> ALTER TABLE ORDERS MOVE PARTITION ORD_P1  
        COLUMN STORE COMPRESS FOR QUERY HIGH  
        UPDATE INDEXES ONLINE;
```

```
SQL> ALTER TABLE ORDERS MOVE PARTITION ORD_P1  
        COLUMN STORE COMPRESS FOR ARCHIVE HIGH  
        UPDATE INDEXES ONLINE;
```

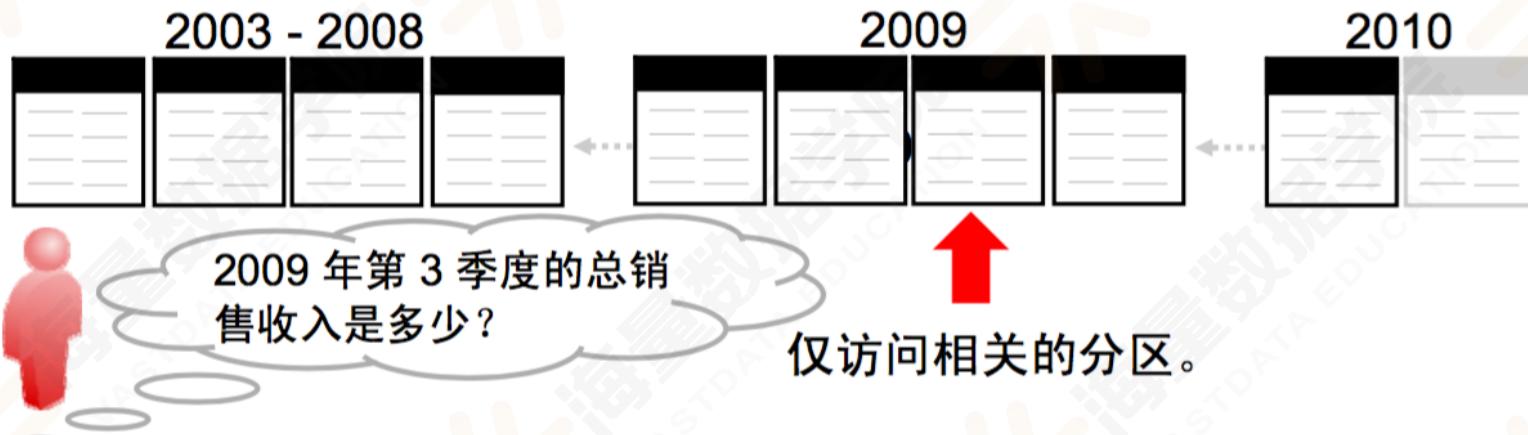
归档难题

归档中的难题包括：

- 保持归档数据在数据库中即时可用
- 编写程序以将数据从磁带重新加载到数据库中
- 备份海量数据，包括旧数据
- 删除旧数据

归档解决方案

分区



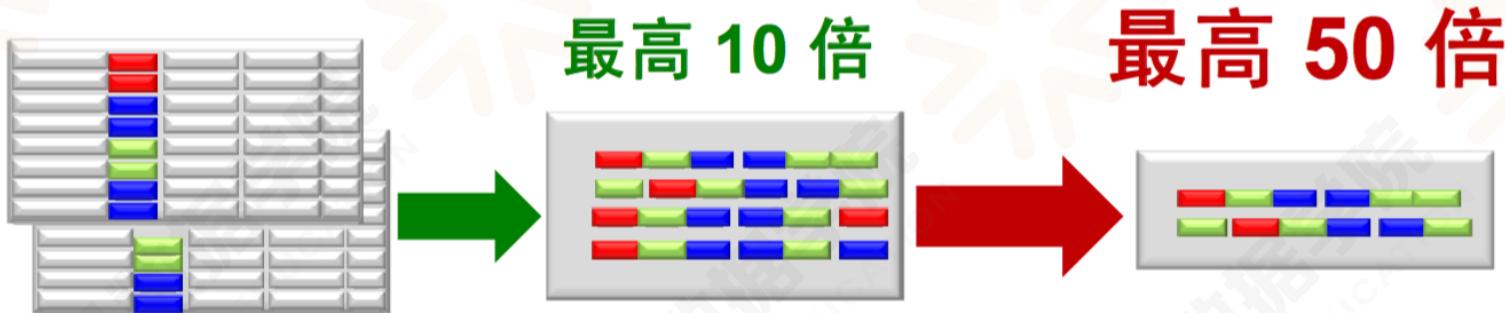
闪回数据归档



SEGMENT 级别的 ILM 联机归档

- 存储分层
- 用于归档的 OLTP/HCC 压缩

数据库中归档: HCC



- 数据仓库表的**查询模式**
 - 通常 10 倍的压缩率
- 旧数据的**归档模式**
 - 通常 15-50 倍的压缩率
- 数据按列存储, 然后进行压缩

归档难题和解决方案

难题：

- 将不活动数据和活动数据保留在同一表中。
- 仅访问活动数据。
- 减少归档存储量。
- 确保对归档数据的高性能访问。

解决方案：

- **数据库中归档**
 - 活动
 - 不活动
- **期间有效性**
 - 定义有效时间维。
 - 按有效时间列进行过滤，仅访问活动数据。



数据库中归档

- 启用行归档：添加 ORA_ARCHIVE_STATE 列。

```
SQL> CREATE TABLE emp
      (EMPNO NUMBER(7), FULLNAME VARCHAR2(40),
       JOB VARCHAR2(9), MGR NUMBER(7))
      ROW ARCHIVAL;
```

- 新建行：ORA_ARCHIVE_STATE 值 0。
- 将 ORA_ARCHIVE_STATE 更新为 1 以使用行归档。



ORA_ARCHIVE_STATE 列

- 查看 ORA_ARCHIVE_STATE 列。

```
SQL> SELECT ORA_ARCHIVE_STATE, fullname FROM emp;
```

ORA_ARCHIVE_STATE	FULLNAME
0	JEAN
1	ADAM
1	TOM
1	JIM

0	JEAN	活动
1	ADAM	不活动
1	TOM	不活动
1	JIM	不活动

- 将行设置为归档状态。

```
SQL> UPDATE emp SET ORA_ARCHIVE_STATE = 1  
WHERE empno < 100;
```

- 将行设置回活动状态。

```
SQL> UPDATE emp SET ORA_ARCHIVE_STATE = 0;
```

会话可见性控制

- 会话可见性控制：默认情况下，仅限活动行

```
SQL> select ORA_ARCHIVE_STATE, fullname from emp;
```

```
ORA_ARCHIVE_STATE FULLNAME
```

```
-----
```

```
0 JEAN
```



活动

- 启用所有行的应用程序可见性。

```
SQL> alter session set ROW ARCHIVAL VISIBILITY = ALL;
```

```
SQL> select ORA_ARCHIVE_STATE, fullname from emp;
```

```
ORA_ARCHIVE_STATE FULLNAME
```

```
-----
```

```
0 JEAN
```



活动

```
1 ADAM
```



不活动

```
1 JIM
```



不活动

禁用行归档

- 禁用行归档: ORA_ARCHIVE_STATE 列已删除

```
SQL> ALTER TABLE emp NO ROW ARCHIVAL;
```



期间有效性与期间历史记录

- 期间有效性

Emp IDc	工作	聘用日期
100	Clerk	22-Apr-11
200	Developer	12-Dec-11
400	Salesman	22-Mar-12

- 期间历史记录



PERIOD FOR 子句概念

- 将活动数据和不活动数据保留在同一表中。
- 在创建表时定义一个有效时间维。
 - 明确定义两个日期时间列。

```
SQL> CREATE TABLE emp
      ( empno number, salary number, deptid number,
        name VARCHAR2(100),
        user_time_start DATE, user_time_end DATE,
        PERIOD FOR user_time (user_time_start,user_time_end));
```

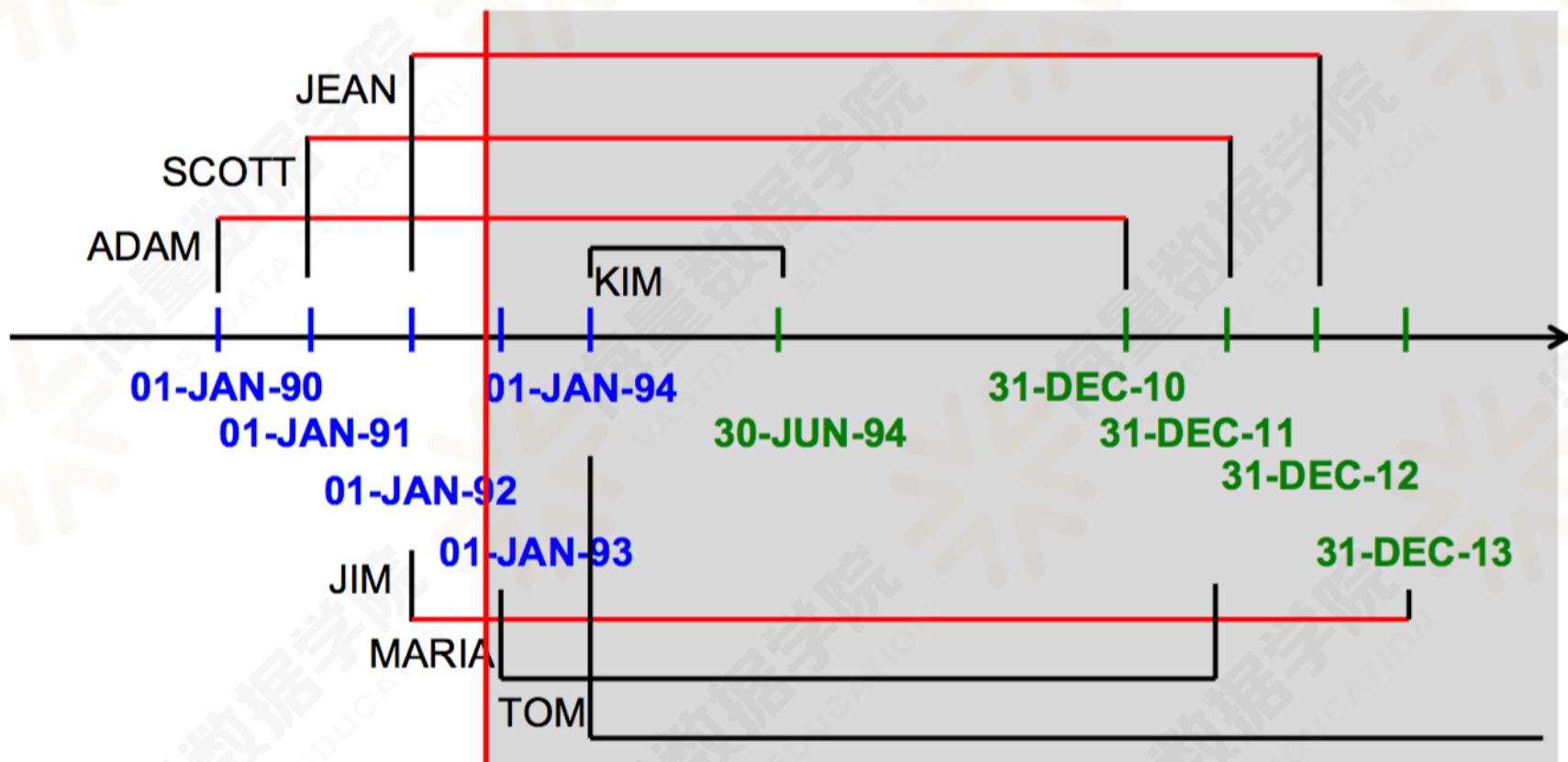
- 或者创建隐式自动有效时间列。

```
SQL> CREATE TABLE emp2
      ( empno number, salary number, deptid number,
        name VARCHAR2(100),
        PERIOD FOR user_time);
```

- 插入行时，会明确命名有效时间列。

按有效时间列进行过滤：示例 1

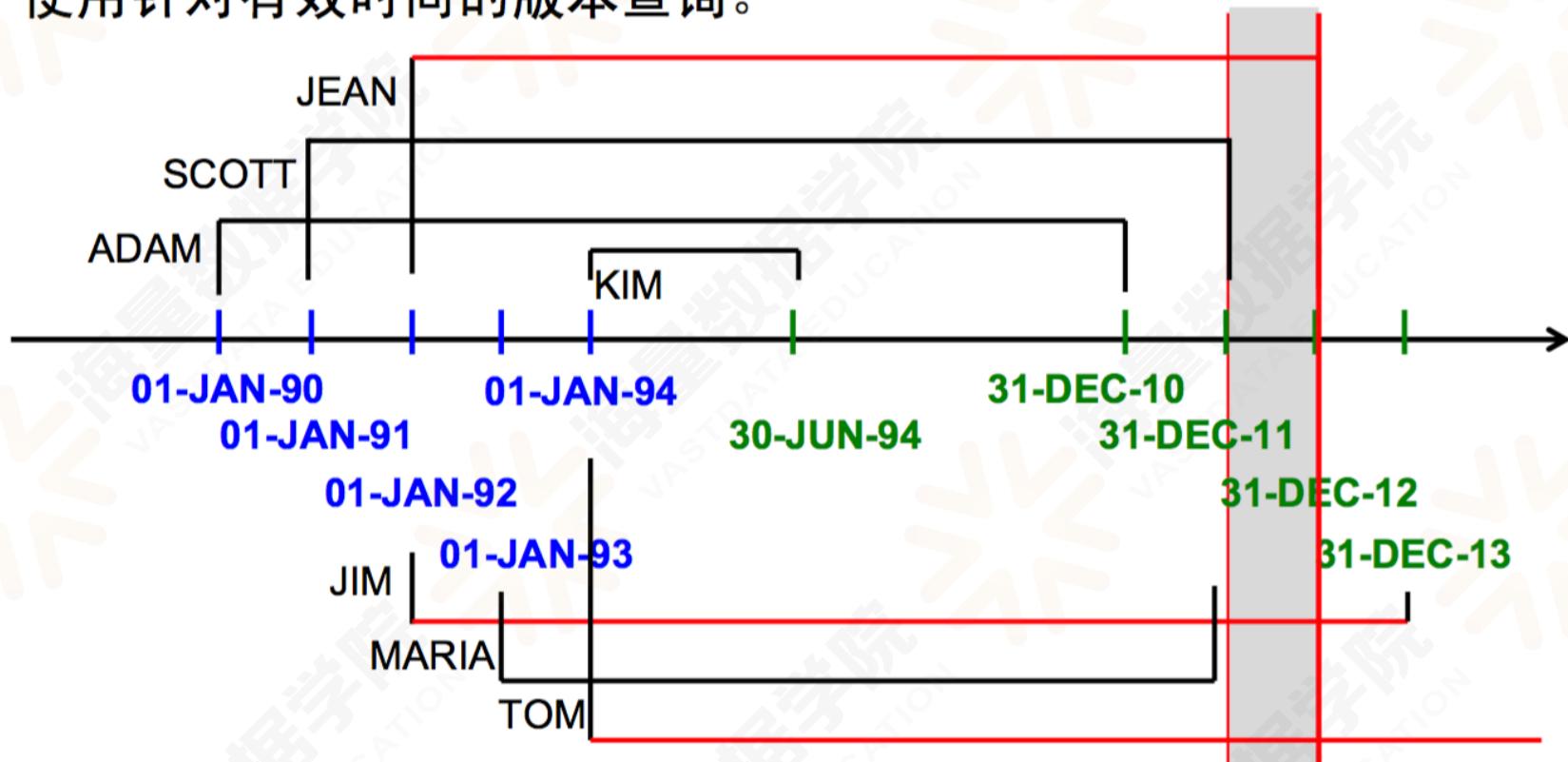
按有效时间列进行过滤，仅访问活动数据。



```
SQL> select * from hr.emp as of PERIOD FOR user_time  
      to_date('01-DEC-1992', 'dd-mon-yyyy') ;
```

按有效时间列进行过滤：示例 2

使用针对有效时间的版本查询。



```
SQL> SELECT * FROM hr.emp VERSIONS PERIOD FOR user_time  
      BETWEEN to_date('31-DEC-2011','DD-MON-YYYY')  
      AND      to_date('31-DEC-2012','DD-MON-YYYY');
```

DBMS_FLASHBACK_ARCHIVE

- 可见性控制适用于查询和 DML。
- 完整可见性适用于 DDL。
- 可见性使用 DBMS_FLASHBACK_ARCHIVE 来设置：
 - 将数据的可见性设置为在指定时间有效的数据。

```
SQL> exec DBMS_FLASHBACK_ARCHIVE.ENABLE_AT_VALID_TIME('ASOF', -  
          to_timestamp('29-SEP-10 05.44.01 PM'))
```

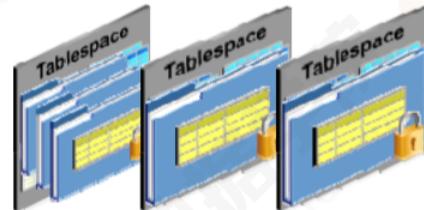
- 在会话级别将数据的可见性设置为当前在有效时段内有效的数据。

```
SQL> exec DBMS_FLASHBACK_ARCHIVE.ENABLE_AT_VALID_TIME('CURRENT')
```

- 将数据的可见性设置为全表级别。

```
SQL> exec DBMS_FLASHBACK_ARCHIVE.ENABLE_AT_VALID_TIME('ALL')
```

期间历史记录增强功能：FDA 优化



12c 之前版本

历史记录或归档表：

- 压缩和取消重复

12c

历史记录或归档表：

- 未压缩或未取消重复

```
SQL> CREATE FLASHBACK ARCHIVE fla1 TABLESPACE tbs1  
RETENTION 5 YEAR;
```

12c

历史记录或归档表：已压缩或已取消重复

```
SQL> CREATE FLASHBACK ARCHIVE fla1 TABLESPACE tbs1  
OPTIMIZE DATA RETENTION 5 YEAR;
```

期间历史记录增强功能：用户上下文元数据

在启用 FDA 的表上，事务处理的用户上下文是：

- 按属性收集：NONE（默认值）、TYPICAL、ALL
 - 使用 DBMS_FLASHBACK_ARCHIVE.SET_CONTEXT_LEVEL 设置
 - 如果为 TYPICAL，则收集数据库用户 ID、全局用户 ID、客户机标识符、服务名、模块名或主机名
- 使用 GET_SYS_CONTEXT 从 USERENV 获取

```
SQL> EXEC DBMS_FLASHBACK_ARCHIVE.SET_CONTEXT_LEVEL('TYPICAL')
PL/SQL procedure successfully completed
```

```
SQL> SELECT DBMS_FLASHBACK_ARCHIVE.GET_SYS_CONTEXT
      (VERSIONS_XID, 'USERENV', 'SESSION_USER'),
      VERSIONS_XID, VERSIONS_STARTTIME, VERSIONS_ENDTIME,
      employee_id, salary
  FROM hr.employees VERSIONS BETWEEN SCN MINVALUE
                                AND MAXVALUE ;
```

RMAN 增强功能

SYSBACKUP
RMAN SQL
RMAN DESCRIBE
多段
复制
第三方

DBA 职责分离

介绍特定于任务的最少权限

管理权限 **SYSBACKUP**

- 包括用于备份和恢复的权限（连接到关闭的数据库）
- 不包括数据访问权限，例如 SELECT ANY TABLE
- 授予在安装数据库期间创建的 SYSBACKUP 用户
- 可以由 SYSBACKUP 授权用户显示用于 RMAN 连接



```
% rman TARGET '"john@orcl AS SYSBACKUP"'  
connected to target database: ORCL (DBID=1297344416)
```

在 RMAN 中使用 SQL

- 不需要 SQL 前缀和引号
- 提供了 SQL*Plus DESCRIBE 功能
- 早期发行版：

```
RMAN> sql 'ALTER TABLESPACE users
          ADD DATAFILE ''/testdata/users02.dbf'' SIZE 10M';
```

- 使用 Oracle Database 12c 时

```
RMAN> ALTER TABLESPACE users
          ADD DATAFILE '/testdata/users02.dbf' SIZE 10M;
```

```
RMAN> SELECT NAME, DBID, LOG_MODE FROM V$DATABASE;
```

Name	Null?	Type
TEST_NAME		VARCHAR2 (128)

RMAN 增强功能

- SYSBACKUP
- RMAN SQL
- RMAN DESCRIBE
- 多段
- 复制
- 第三方

备份和还原非常大的文件

单个数据文件的多段备份：

- 由 RMAN 创建，具有指定的大小值
- 用于备份集和映像副本
- 用于完全和增量备份

优点：

- 减少映像副本创建时间
- 独立进行处理（串行或并行）
- 使 Exadata 受益

要求和限制：

- COMPATIBLE=12.0
- 不适用于控制文件或 spfile
- 不适用于大的并行度值



RMAN 增强功能

SYSBACKUP
RMAN SQL
RMAN DESCRIBE
多段
复制
第三方

RMAN 复制增强功能



活动数据库复制的新功能：

- 默认情况下对活动复制使用备份集
- 可选择压缩、段大小和加密
- 提供了用于在数据库处于已装载但未打开状态时结束数据库复制的选项
- 复制可插入数据库

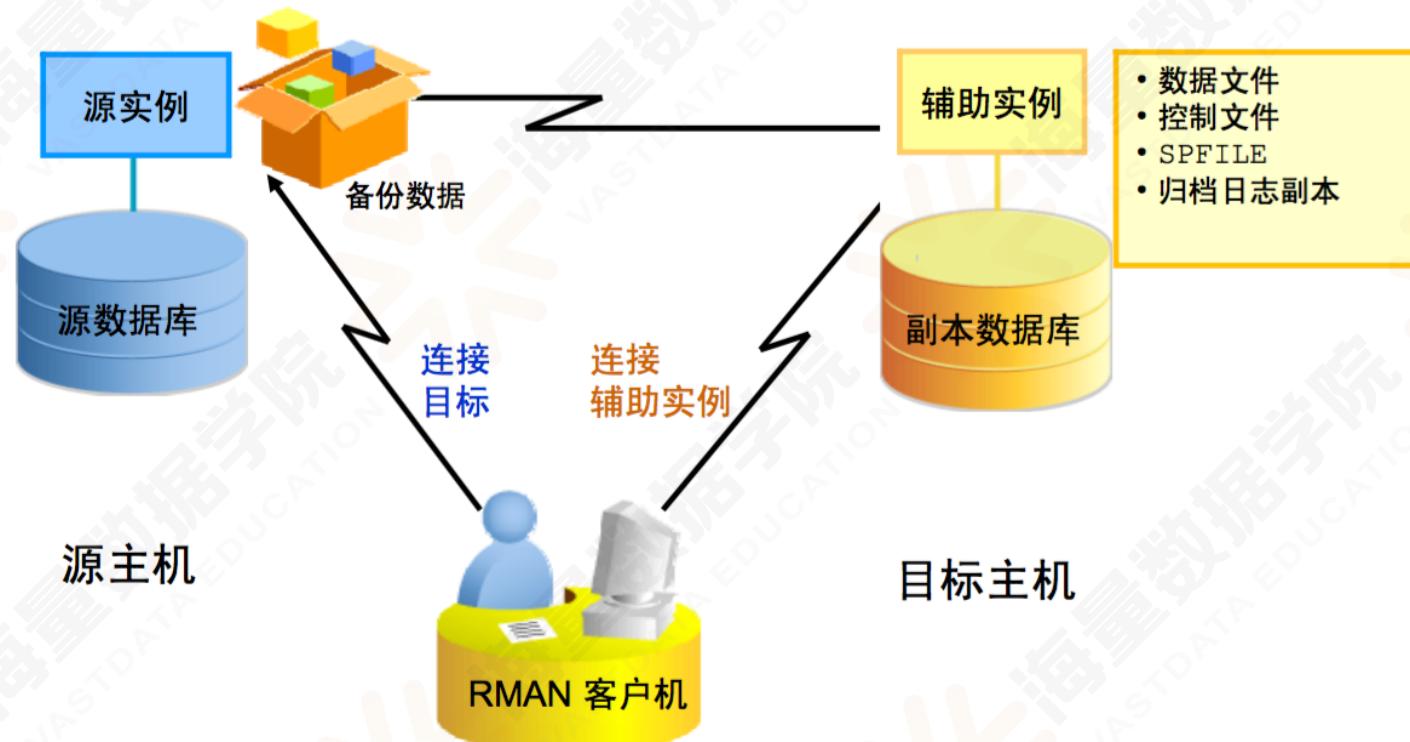


海量数据
VASTDATA

海量数据学院
VASTDATA EDUCATION

复制活动数据库

- 通过 Enterprise Manager 或 RMAN 命令行
- 通过网络（使用 FROM ACTIVE DATABASE 子句）
- 通过定制 SPFILE



新增功能

默认情况下对活动复制使用备份集：

- 原来的基于映像副本的“推送”流程变为基于**备份集**的“**拉取**”流程。
- 使用某个子句时的自动“**拉取**”流程：
 - 压缩
 - 段大小
 - 加密

数据在传输之前在源数据库上进行加密。

```
RMAN> SET ENCRYPTION ...;  
RMAN> DUPLICATE TARGET DATABASE TO orcl2  
      FROM ACTIVE DATABASE  
      [USING BACKUPSET]  
      [SECTION SIZE ...]  
      [USING COMPRESSED BACKUPSET] ...;
```

NOOPEN 选项

用于在数据库处于已装载但未打开状态时结束数据库复制的选项：

- 默认情况下，它使用 RESETLOGS 选项打开副本数据库，并且为副本数据库创建联机重做日志文件。
- 使用 NOOPEN 选项，流程将在副本数据库处于 MOUNT 模式的情况下完成。

潜在的 NOOPEN 用例：

- 移动数据库的位置（例如，移动到 ASM）
- 升级数据库（在运行升级脚本之前，禁止通过 resetlogs 打开数据库）

复制多租户容器数据库

- 单个可插入数据库:

```
RMAN> DUPLICATE DATABASE TO cdb1 PLUGGABLE DATABASE pdb1;
```

- 多个可插入数据库:

```
RMAN> DUPLICATE DATABASE TO cdb1  
PLUGGABLE DATABASE pdb1, pdb3;
```

- 除了一个可插入数据库之外的所有可插入数据库:

```
RMAN> DUPLICATE DATABASE TO cdb1  
SKIP PLUGGABLE DATABASE pdb3;
```

- 一个 PDB 和其他 PDB 的表空间:

```
RMAN> DUPLICATE DATABASE TO cdb1  
PLUGGABLE DATABASE pdb1  
TABLESPACE pdb2:users;
```

RMAN 增强功能

SYSBACKUP

SQL

DESCRIBE

多段

复制

第三方



使用第三方快照恢复数据库

- 使用最新的快照恢复数据库:

```
RMAN> RECOVER DATABASE;
```

- 使用特定的快照恢复数据库:

```
SQL> RECOVER DATABASE UNTIL TIME '02/15/2012 11:00:00'  
      SNAPSHOT TIME '02/15/2012 10:00:00';
```

- 使用归档重做日志文件执行部分恢复:

```
SQL> RECOVER DATABASE UNTIL CANCEL SNAPSHOT TIME  
      '02/15/2012 10:00:00';
```

- 快照必须满足以下要求才能在恢复操作中使用:
 - 数据库在快照创建期间处于崩溃一致状态。
 - 快照保留了每个文件的写入顺序。
 - 存储时间是快照的完成时间。

跨平台传输数据

跨平台传输数据库、数据文件和表空间：

- 在早期版本中，使用映像副本
- 在 Oracle Database 12c 中，使用备份集或映像副本
- 允许不一致的表空间备份

12c 之前版本

12c

优点：

- 减少了用于平台移植的停机时间
- 可以选择压缩和多段
- 不编入控制文件中，不用于常规还原操作



海量数据
VASTDATA

海量数据学院
VASTDATA EDUCATION

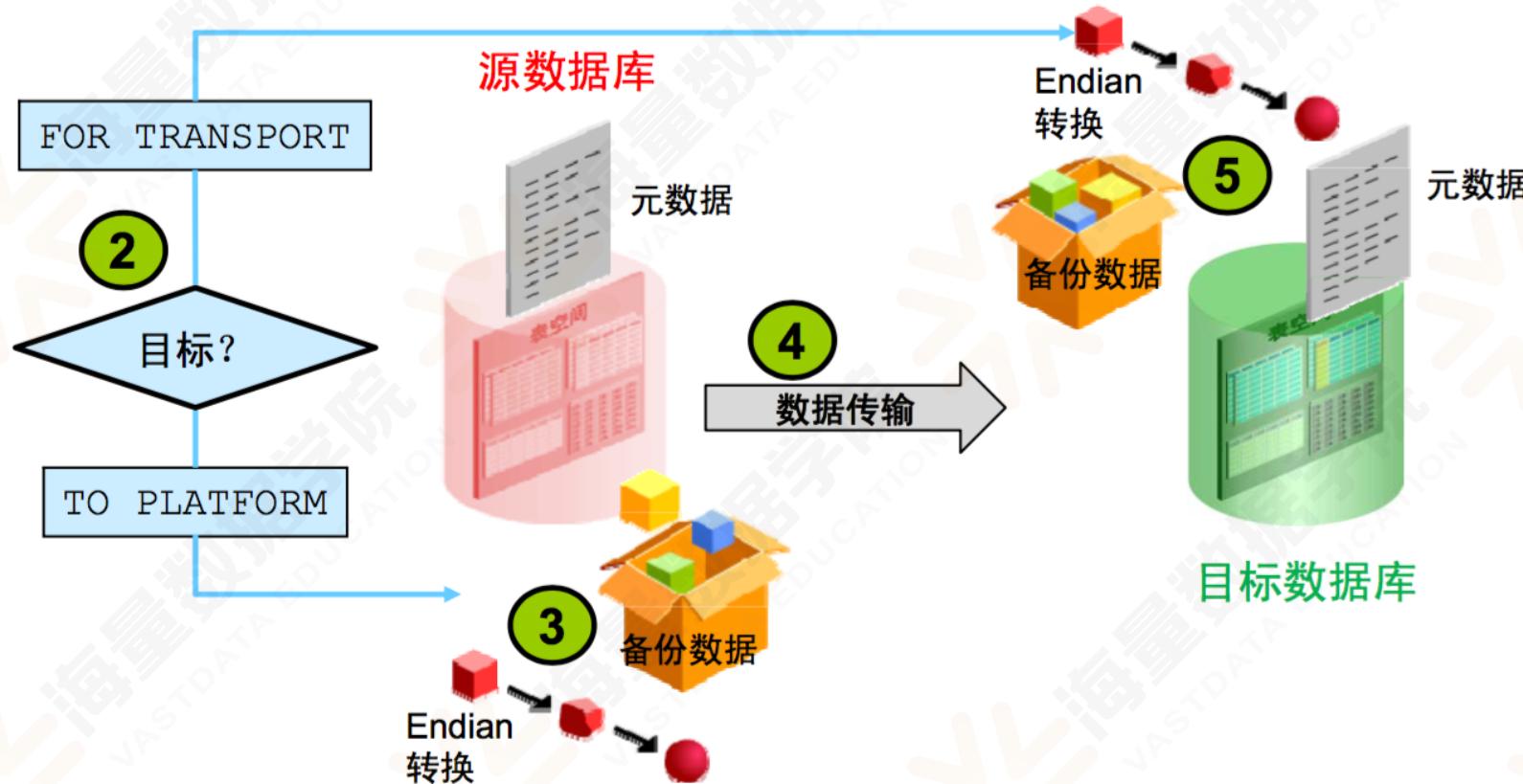
数据传输

1



先决条件

- COMPATIBLE=12.0（或更大值）
- 只读模式，用于创建跨平台数据库备份
- 处于读写模式的数据库，用于创建跨平台表空间备份



传输数据库：流程步骤 — 1



1. 验证先决条件：
 - COMPATIBLE: 大于或等于 12.0
 - OPEN_MODE: READ ONLY
2. 启动一个 RMAN 会话以连接到源数据库。
3. 从 V\$TRANSPORTABLE_PLATFORM 视图查询目标平台的确切名称。
4. 备份源数据库，包括控制文件：
 - 源主机上的转换

```
RMAN> BACKUP TO PLATFORM 'Linux x86 64-bit'  
FORMAT '/bkp_dir/transp.bck' DATABASE;
```

- 目标主机上的转换

```
RMAN> BACKUP FOR TRANSPORT FORMAT '/bkp_dir/transp.bck'  
DATABASE;
```

传输数据库：流程步骤 – 2

5. 从源数据库断开连接。
6. 将备份集移动到目标主机。
7. 连接到作为 TARGET 的目标主机。
8. 使用 RESTORE 命令还原完整备份集。
 - 当转换发生在源主机上时



```
RMAN> RESTORE  
      FROM PLATFORM 'Solaris Operating System (x86-64)'  
      ALL FOREIGN DATAFILES  
      FORMAT '/oradata/datafiles/df_%U'  
      FROM BACKUPSET '/bkp_dir/transp.bck';
```

- 当转换发生在目标主机上时

```
RMAN> ALTER SYSTEM SET DB_CREATE_FILE_DEST='/oradata/orcl';  
RMAN> RESTORE FOREIGN DATABASE TO NEW  
      FORMAT '/oradata/datafiles/df_%U';
```

传输表空间：流程步骤 — 1

1. 验证先决条件。
2. 在源数据库中启动一个 RMAN 会话。
3. 查询目标平台的确切名称。
4. 将表空间置于只读状态。



```
RMAN> ALTER TABLESPACE test READ ONLY;
```

5. 执行可跨平台传输的备份和数据泵导出。
 - 源主机上的转换

```
RMAN> BACKUP TO PLATFORM 'HP Tru64 UNIX'  
FORMAT '/bkp/test.bck'  
DATAPUMP FORMAT '/bkp/test.dmp' TABLESPACE test;
```

- 目标主机上的转换

```
RMAN> BACKUP FOR TRANSPORT FORMAT '/bkp/test.bck'  
DATAPUMP FORMAT '/bkp/test.dmp' TABLESPACE test;
```

传输表空间：流程步骤 — 2

6. 将备份集和数据泵导出转储文件移动到目标主机。
7. 作为 TARGET 连接到目标主机。
8. 还原可跨平台传输的备份和数据泵导出。
 - 当转换发生在源主机上时

```
RMAN> RESTORE FOREIGN TABLESPACE test  
FORMAT '/oracle/test.dbf' FROM BACKUPSET '/bkp/test.bck'  
DUMP FILE FROM BACKUPSET '/bkp/test.dmp' ;
```

- 当转换发生在目标主机上时

```
RMAN> RESTORE FOREIGN TABLESPACE test TO NEW  
FROM BACKUPSET '/bkp/test.bck'  
DUMP FILE FROM BACKUPSET '/bkp/test.dmp' ;
```



表恢复

- 使用 TSPITR、闪回技术进行表恢复
- 从 RMAN 备份进行表恢复
 - 不影响其余的数据库对象
 - 与之前的工作流（例如 TSPITR）相比，降低了时间和磁盘空间

12c 之前版本

12c



海量数据
VASTDATA

海量数据学院
VASTDATA EDUCATION

从备份恢复表

何时从 RMAN 备份恢复表和表分区：

- 对于下列情况，建议不要使用 TSPITR：
 - 少量表
 - 不在自包含表空间中
- 在下列情况下无法进行闪回：
 - 已清除的表（无闪回删除）
 - 超出了可用的还原（无闪回表）
 - 在进行结构性 DDL 更改后（无闪回表）



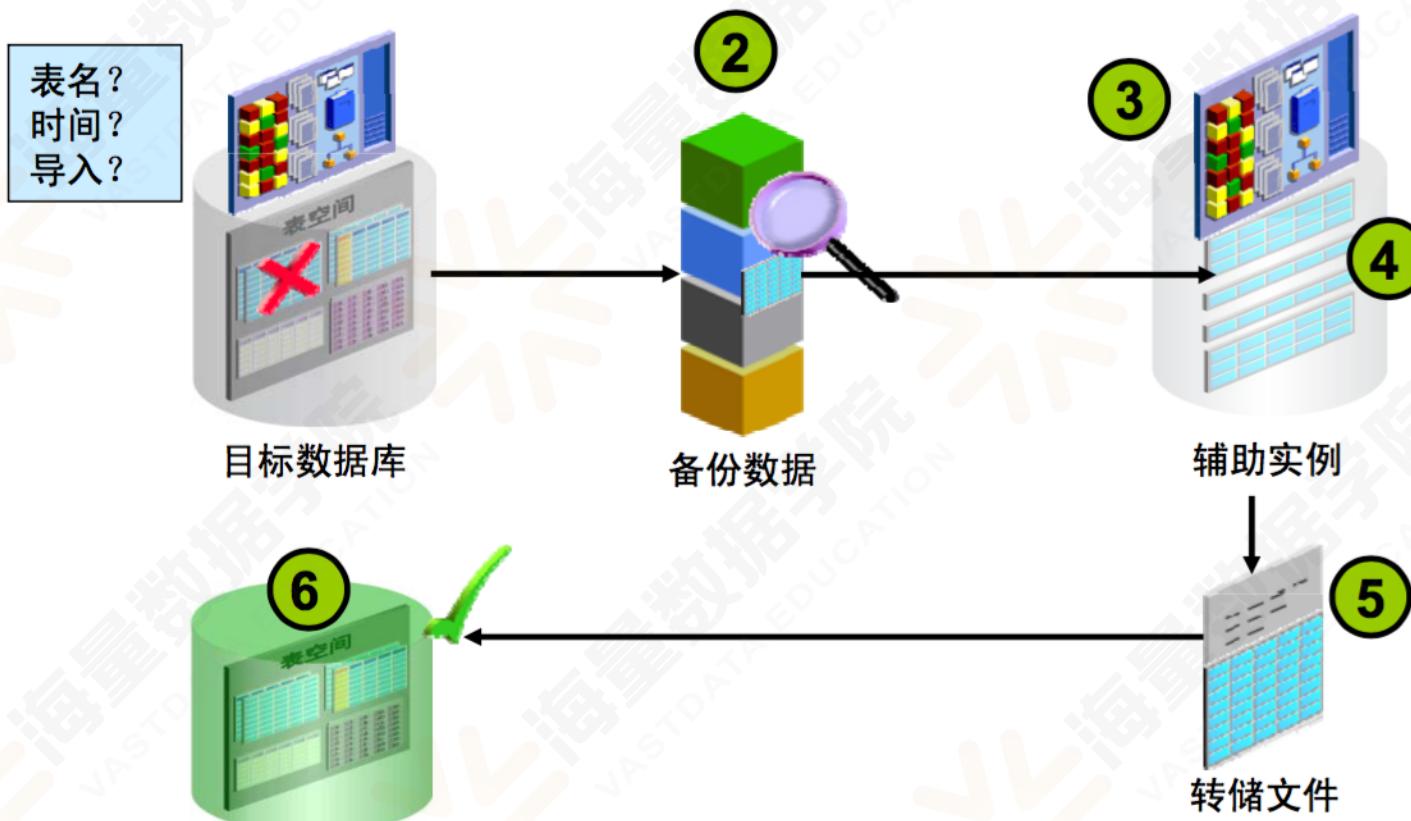
表恢复：图形化概览



先决条件满足

- COMPATIBLE=12.0 (或更大值)
- ARCHIVELOG 模式
- READ WRITE 打开模式

1



指定恢复时间点

将表和表分区恢复到通过指定以下参数确定的状态:

- UNTIL SCN *integer*: 系统更改号 (SCN)
- UNTIL TIME '*date_string*': 采用以下日期格式的时间:
 - NLS_LANG 和 NLS_DATE_FORMAT 环境变量, 或者:
 - 日期常量, 例如: SYSDATE - 5
- UNTIL SEQUENCE *integer* (THREAD *integer*): 日志序列号和线程号

表恢复流程步骤 – 1

1. 执行规划任务并通过 CONNECT TARGET 命令启动一个 RMAN 会话。
2. 输入 RECOVER TABLE 命令。
3. RMAN 根据您的指定确定备份。
4. RMAN 使用 AUXILIARY DESTINATION 子句创建一个辅助实例（如果指定了该子句）。
5. RMAN 将您的表或表分区恢复到此辅助实例，一直恢复到指定的时间点。

注：如果必须在其中创建转储文件的位置中已存在具有 DUMP FILE 指定的名称的文件，则导出将失败。
6. RMAN 使用 DUMP FILE=*name* 和 DATAPUMP DESTINATION=<OS path> 创建包含恢复的对象的数据泵导出转储文件。

定制设置

7. 您可以如下所述进行定制：

- RMAN 将恢复的对象导入到目标数据库，除非指定了 NOTABLE IMPORT。
- RMAN 可以通过 REMAP TABLE 和 REMAP TABLESPACE 子句对恢复的表或表分区进行重命名。





扫描二维码关注“海量数据学院”
助您职业生涯强力进阶
微信号：vastdataedu

联系方式：13811106886
QQ：1883926

荟萃业界名师 培养数据英才



海量数据学院
VASTDATA EDUCATION