

DBA Oracle

Architecture Multitenant

Plan

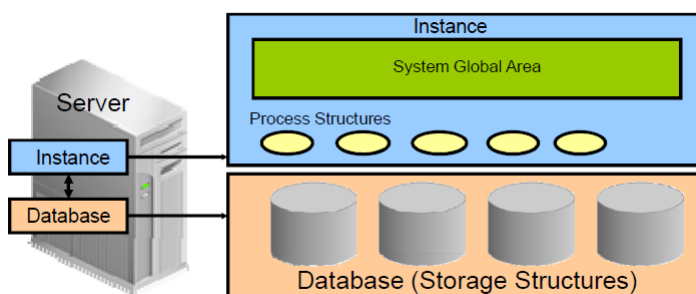
1. Architecture Multi-tenant
2. Séparation entre système et données utilisateur
3. Les utilisateurs communs et locaux
4. Structure des bases de données pluggables (PDB)

Architecture Multi-tenant

- Avant la version 12c, il existait 2 types d'architectures possibles :
 - ▣ La première est la plus utilisée et consiste en la création d'une seule base de données accessible via une seule instance ou bien ce qu'on nomme architecture mono-instance
 - ▣ La deuxième architecture utilise le RAC (Real Application Cluster) qui consiste à mutualiser les ressources de plusieurs serveur pour travailler avec une base de données

Architecture Multi-tenant

Avant la version 12c



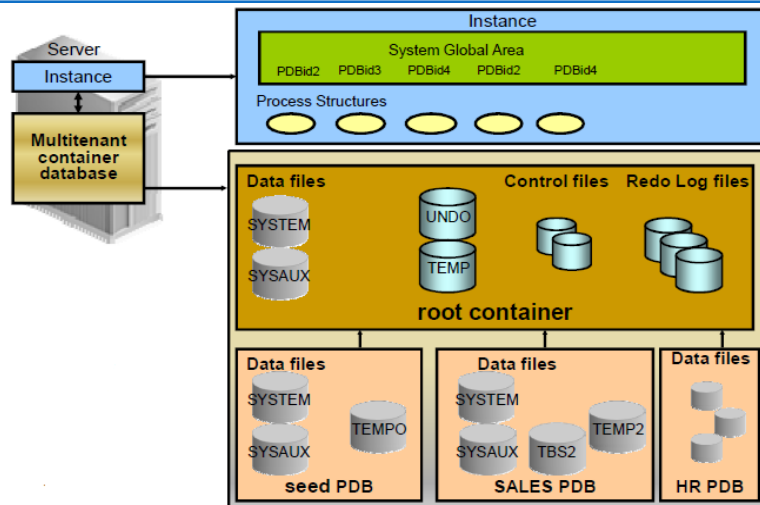
Architecture Multi-tenant

- ❑ Dans la pratique et pour des raisons de structuration, de sécurité et d'applicatif, les entreprises optent pour le choix de plusieurs base de données avec une architecture le plus souvent mono-instance.
- ❑ En effet, si plusieurs applications utilisent la même base de données, il ya un risque de conflit dans les schémas des applications, des utilisateurs, des synonymes publiques...

Architecture Multi-tenant

- ❑ Avec l'arrivée de la version 12c d'Oracle, une architecture appelée Architecture Multitenant est proposée pour régler tous ces problèmes
- ❑ L'idée de l'architecture consiste à créer plusieurs base de données en une seule base de données conteneur
- ❑ Cette architecture permet à Oracle de placer un ensemble de schémas appartenant à une application dans une base de données séparée nommée **PDB (Pluggable Database)** ou en français Base de données Insérée), cette dernière est logée dans une base de données conteneur nommée **CDB (Container Database)**

Architecture Multi-tenant



Source: <https://mkrishnadba.blogspot.com/2013/09/12c.html>

Séparation entre système et données utilisateur

- Dans l'architecture Multitenant, chaque base de données conteneur(CDB) contient les éléments suivants :
 - La base de données nommée **CDB\$ROOT** qui contient le dictionnaire de données global contenant les informations sur les PDBs
 - Le modèle type **PDB\$SEED** utilisé comme modèle de création des PDBs
 - L'ensemble des **PDBs** créées dans le but de gérer les données des applications utilisateurs d'une manière séparée et autonome; Ainsi chaque PDBs possède son propre dictionnaire de données et les objets de ces utilisateurs

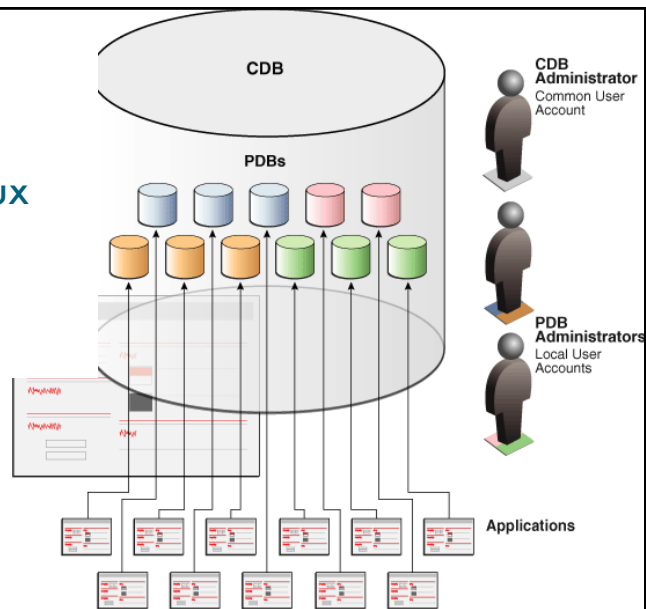
Les utilisateurs communs et locaux

- Un utilisateur local est un utilisateur de la PDB
 - Il n'est pas connu des autres PDBs
 - Il possède des privilèges au sein de sa PDB
 - Il possède son propre schéma
 - On ne peut pas créer d'utilisateur local à un CDB
- Un utilisateur commun est défini dans le dictionnaire de données de la CDB\$ROOT
 - Il est connu dans toutes les PDBs appartenant au CDB
 - Il peut réaliser des tâches d'administration tel que Insertion ou non des PDBs, Démarrage de la CDB lorsqu'il possède les privilèges adéquats
 - Son login commence par c##

Les utilisateurs communs et locaux

- Rôle local est un rôle défini au sein d'une PDB et il n'est pas connu hors cette PDB
- Un rôle commun est défini au niveau de la CDB et il peut être dupliqué sur l'ensemble des PDBs. Il est affecté aux utilisateurs communs

Les utilisateurs communs et locaux



Source : <https://docs.oracle.com/database/121/CNCPT/cdbovrvw.htm#CNCPT89240>

Structure des bases de données pluggables (PDB)

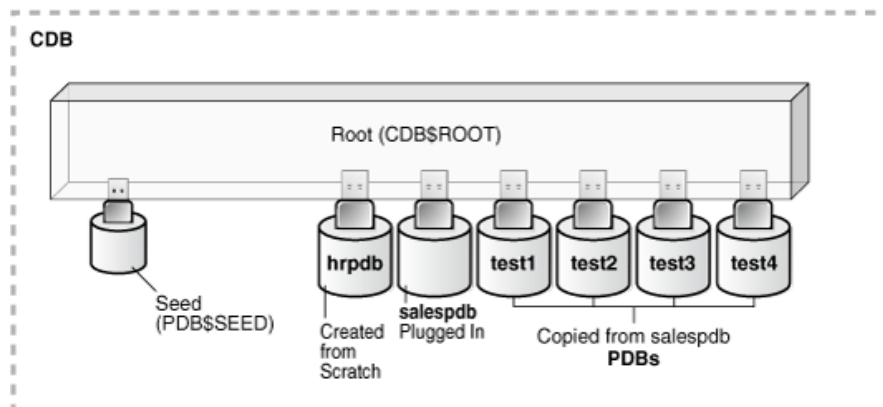
- ❑ Chaque PDB est autonome et fonctionne du point de vue de l'utilisateur comme une base de données normale
- ❑ Lorsqu'un utilisateur est connecté à une PDB, il n'a pas de visibilité sur le CDB ou les autres PDBs
- ❑ Chaque PDB contient ses propres tablespaces : SYSTEM(Dictionnaire de données propre), SYSAUX et TEMP

Structure des bases de données pluggables (PDB)

- La PDB peut être mis hors ligne dans la CDB, les autres PDBs ne sont pas influencées
- Il ya un seul fichier de contrôle, un seul fichier d'alertes, un seul tablespace undo et plusieurs fichiers de journalisation pour l'ensemble des PDBs

Structure des bases de données pluggables (PDB)

CDB avec Six PDBs



Source : <https://docs.oracle.com/database/121/CNCPT/cdbovrvw.htm#CNCPT89240>

Vues CDB_* et DBA_*

- Pour chaque vue DBA_*, une vue CDB_* est définie. À la racine d'une base de données conteneur mutualisée (CDB), les vues CDB_* peuvent être utilisées pour obtenir des informations sur les tables, les tablespaces, les utilisateurs, les privilèges, les paramètres, etc. contenus dans la racine et dans les bases de données enfilables (PDB).
- Les vues CDB_* peuvent être interrogées uniquement par les utilisateurs disposant du privilège ou du rôle SYSDBA, ou par les utilisateurs disposant de privilèges directs qui leur sont accordés

Vues CDB_* et DBA_*

- Dans un PDB, les vues CDB_* affichent uniquement les objets visibles via une vue DBA_* correspondante.
- Toutes les colonnes trouvées dans une vue DBA_* donnée, la vue CDB_* correspondante contient également la colonne CON_ID qui identifie un conteneur dont les données représentent une ligne CDB_* donnée.
- Dans un "non-CDB", la valeur de CON_ID sera 0.