BANCO DE DADOS
Grupos, Usuários e Privilégios

Ma. Simone Maria Viana Romano

GRUPOS, USUARIOS E PRIVILÉGIOS

O que o usuário quer?

O que o usuário precisa?

Qual seria o nível mínimo de segurança aceitável? E o máximo?

Exigem regras na empresa que restrinjam a utilização do banco pelo usuário?

Exibe outro usuário que queira ou necessite do mesmo?

GRUPOS, USUARIOS E PRIVILÉGIOS

CONCEITO	EXPLICAÇÃO
USUARIO	Indivíduo que se conecta ao banco de dados e utiliza objetos nele. Também pode criar objetos e permitir que outros usuários manipulem.
SCHEMA	Conjuntos de objetos (tabelas, índices, visões, etc) de um usuário. Pode ser listado com o comando: SELECT * FROM cat;
PRIVILÉGIO	Direito que um usuário recebe para fazer algo. Existem duas categorias: - sistema: realiza conexão até eliminar qualquer tabela - objeto: desde ler até eliminar
PAPEL (role)	Conjunto de privilégios agrupados e com um nome. Facilita bastante a gerência de privilégios.

11g

Traditional Architecture (Non-CDB)

12c

Multitenant Architecture

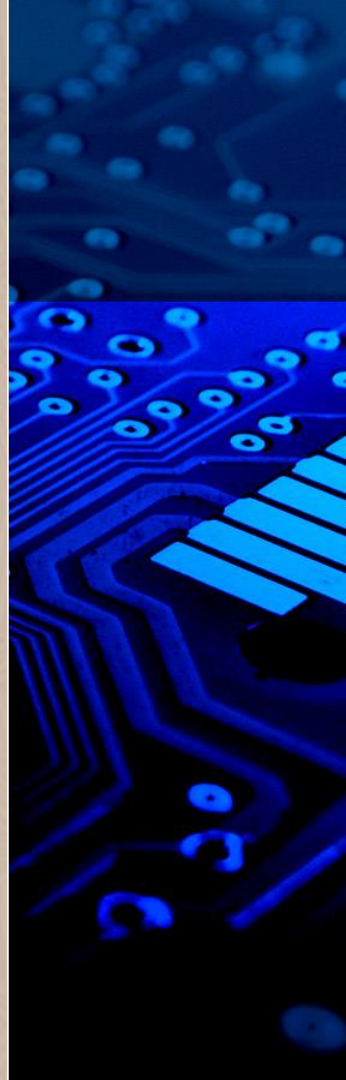
One PDB per CDB
(Single-Tenant)
without Multitenant License

19c

Three PDBs per CDB
(Multi-Tenant)
without Multitenant License

21c

Multitenant Architecture (CDB/PDB) is the **ONLY available Architecture**



VERSÃO ORACLE 10G, 11G

USUARIOS

CRIANDO USUARIOS:

Sintaxe: CREATE USER nome
IDENTIFIED BY senha
DEFAULT TABLESPACE users
QUOTA UNLIMITED ON users;

CONSULTAR USUÁRIOS EXISTENTES:

SELECT * FROM ALL_USERS;

PARA ALTERAR A SENHA DE ACESSO:

ALTER USER nome IDENTIFIED BY nova_senha;

EXCLUINDO UM USUÁRIO: DROP USER nome CASCADE;

Obs. O espaço (cota) pode ser ilimitado(UNLIMITED), 1M (Mbytes) ou 1K (Kbytes).

VERSÃO ORACLE 12C e seguintes

ORACLE MULTITENANT (Vários Inquilinos)

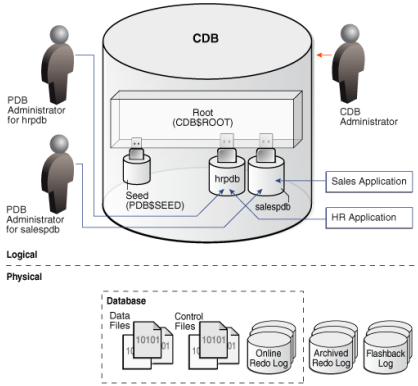
Consolidar em um contêiner de banco de dados, ao mesmo tempo mantendo todos eles como bancos de dados separados.

Vantagens: consolidação, desempenho, capacidade e perspectiva operacional.

Trabalha com todos os recursos: *Oracle Database como Real Application Clusters, Partitioning, Data Guard, Compression, Automatic Storage Management, Real Application Testing, Transparent Data Encryption, Database Vault,* entre outros.

CDB: Banco de Dados de Contêiner multilocatário (CDB).

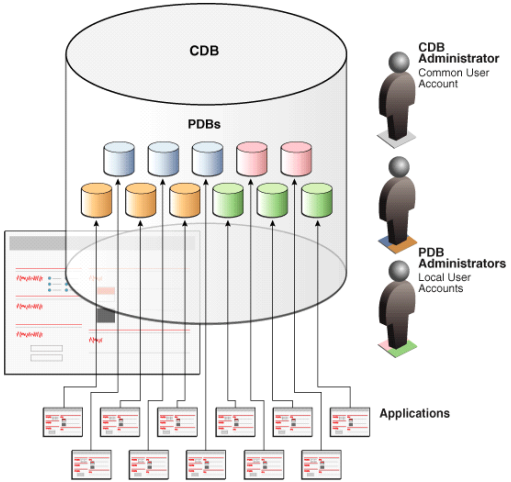
- Coleção de esquemas, objetos e estruturas relacionadas;
- Dentro de um CDB, cada contêiner tem um ID e um nome exclusivos.
- A partir do Oracle Database 21c, um banco de dados de contêiner multilocatário é a única arquitetura suportada.
- Um container chamado “ROOT” (CDB$ROOT) possue as “tablespaces”: SYSTEM, SYSAUX, UNDO e TEMP e com isso, os “controlfiles” e os arquivos de “redo log”.
- Um container chamado “SEED” (PDB$SEED) possui as “tablespaces”: SYSTEM, SYSAUX, TEMP e EXAMPLE, usados como base para criar novos PDBs.



AMBIENTE MULTITENANT

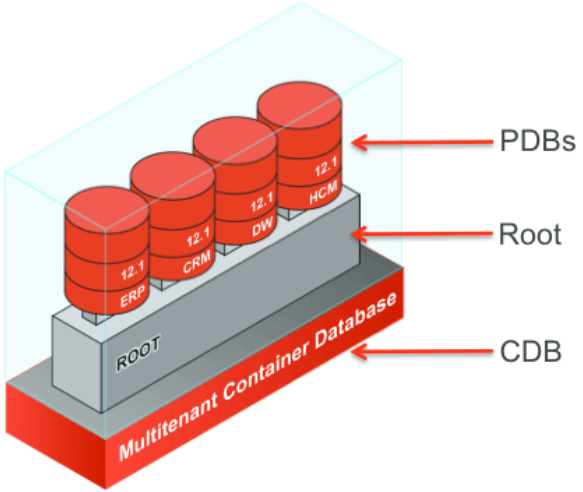
Há 2 tipos de usuários e roles:

- **Common User:** usuário que está presente em todos os containers (container root e nos PDBs);
- **Local User:** usuário que está presente em apenas um PDB específico;
- **Common Role:** role que está presente em todos os containers (container root e nos PDBs);
- **Local Role:** role que está presente em apenas um PDB específico.



CDB x PDB

- **Banco de Dados do tipo Container (CDB):** capacidade de armazenar logicamente diversos bancos de dados internos (*Pluggable Database* (PDB));
- **Banco de Dados do tipo "Pluggable" (PDB):** conjunto de esquemas de bancos de dados que são apresentados aos usuários e aplicações como uma representação de um banco de dados separado e independente.



Criando **Common Users**

Deve estar conectado no container root e o prefixo deve respeitar o parâmetro de banco “common_user_prefix”:

```
SELECT name, value
FROM v$parameters
WHERE name LIKE '%common%’
```

Criando Common Users

CREATE USER c##usuario1 IDENTIFIED BY senha CONTAINER=ALL;

- **Por padrão, quando estamos conectados no container root, ele irá utilizar a cláusula CONTAINER=ALL**

CREATE USER c##usuario2 IDENTIFIED BY senha;

- **Não é possível utilizar a clausula CONTAINER=CURRENT quando estamos conectados no root:**

CREATE USER c##usuario3 IDENTIFIED BY senha CONTAINER=CURRENT;

- **Verificando os usuários criados:**

SELECT username, conn_id FROM cdb_users
WHERE username LIKE 'C##%' ORDER BY 2,1;

Criando Local Users

- **Mesma forma que era realizado no 11G:**

```
ALTER SESSION SET CONTAINER=pdb1;

CREATE USER usuario1 IDENTIFIED BY senha CONTAINER=CURRENT;
```

- **Por padrão, quando estamos conectados em um PDB, ele irá utilizar a cláusula CONTAINER=CURRENT:**

```
CREATE USER usuario2 IDENTIFIED BY senha;
```

- **Verificando os usuários criados:**

```
SELECT username, conn_id FROM cdb_users
 WHERE username LIKE 'USUARIO%' ORDER BY 2,1;
```

Criando Common Roles

- **Criando common roles:**

```
CREATE ROLE c##role1 CONTAINER=ALL;
```

- **Por padrão, quando estamos conectados no container root, ele irá utilizar a cláusula CONTAINER=ALL:**

```
CREATE ROLE c##role2;
```

- **Não é possível utilizar a cláusula CONTAINER=CURRENT quando estamos conectados no root:**

```
CREATE ROLE c##role3 CONTAINER=CURRENT;
```

- **Verificando roles criadas:**

```
SELECT role, con_id FROM cdb_roles WHERE role LIKE 'C##%'
ORDER BY 2,1;
```

Criando Local Roles

- **Mesma forma que era feito até a versão 11G:**

```
ALTER SESSION SET CONTAINER=pdb1;

CREATE ROLE role1 CONTAINER=CURRENT;
```

- **Por padrão, quando estamos conectados em um PDB, ele irá utilizar a cláusula CONTAINER=CURRENT:**

```
CREATE ROLE role2;
```

- **Verificando as roles criadas:**

```
SELECT role, con_id FROM cdb_roles WHERE role LIKE 'ROLE%'
ORDER BY 2,1;
```

PRIVILÉGIOS

Há dois tipos de privilégio:

- **SISTEMA:** CREATE SESSION; CREATE TABLE; CREATE SEQUENCE; CREATE *VIEW*; CREATE USER e DROP USER.

- **OBJETO:**

PRIVILÉGIOS DE OBJETO

- Privilégios de objeto servem, basicamente para que outros usuários que não o dono, ou aqueles que tenham privilégios de sistema do tipo ANY possam ler, alterar ou até mesmo apagar linhas de tabelas.

Privilégio	Tabela	View	Sequence	Procedure
ALTER	X		X	
DELETE	X	X		
EXECUTE				X
INDEX	X			
INSERT	X	X		
REFERENCES	X			
SELECT	X	X	X	
UPDATE	X	X		
ALL	X	X	X	X

PRIVILÉGIOS DE OBJETO

PARA CONCEDER PRIVILÉGIOS

GRANT privilegio1, privilegio2 ON Objeto

TO Usuário ou Grupo ou PUBLIC

[WITH GRANT OPTION];

Observação:

- **WITH GRANT OPTION** permite ao grantee conceder os privilégios de objetos a outros usuários.
- **PUBLIC** o privilégio será dado a todos os usuários.

Atribuindo Privilégios: Common Users e Common Roles

- **Atribuindo privilégios a um Common User:**

```
GRANT CREATE SESSION TO c##usuario1 CONTAINER=ALL;
```

- **Atribuindo privilégios a um Common User apenas no container corrente:**

```
GRANT CREATE SESSION TO c##usuario2
CONTAINER=CURRENT;
```

- **Verificando os privilégios atribuídos:**

```
SELECT * FROM cdb_sys_privs WHERE grantee LIKE
'C##U%' ORDER BY 1,5;
```

Atribuindo Privilégios: Common Users e Common Roles

A atribuição de privilégios para Roles funciona da mesma forma.

- **Atribuindo privilégios a uma Common Role em todos containers:**

```
GRANT CREATE SESSION TO c##role2 CONTAINER=ALL;
```

- **Atribuindo privilégios a uma Common Role apenas no container corrente:**

```
GRANT CREATE SESSION TO c##role1 CONTAINER=CURRENT;
```

- **Verificando os privilégios atribuídos:**

```
SELECT * FROM cdb_sys_privs WHERE grantee LIKE 'C##ROLE%’
ORDER BY 1,5;
```

Atribuindo Privilégios: Common Users e Common Roles

Atribuindo Privilégios a Local Users e Local Roles

- **Atribuindo privilégios a um Local User:**

ALTER SESSION SET CONTAINER=pdb1;

- **Verificando os privilégios atribuídos:**

GRANT CREATE SESSION TO usuario1;

- **Atribuindo privilégios a um Local Role:**

SELECT * FROM cdb_sys_privs WHERE grantee LIKE 'USUARIO%’
ORDER BY 1,5;

- **Verificando os privilégios atribuídos:**

GRANT CREATE SESSION TO role1;

Atribuindo Privilégios: Common Users e Common Roles

- **Atribuindo uma Common Role para um Local User:**

- **Atribuindo uma Local Role para um Common User:**

GRANT c##role1 TO usuario2;

PRIVILÉGIOS DE OBJETO

PARA VISUALIZAR OS PRIVILÉGIOS DE OBJETO

- SELECT * FROM USER_TAB_PRIVS_MADE;

SELECT * FROM USER_COL_PRIVS_MADE;

COLUNAS	DESCRIÇÃO
GRANTEE	Usuário a quem o privilégio foi concedido;
TABLE_NAME	Nome do objeto no qual o privilégio foi concedido;
COLUMN_NAME	Nome da coluna no qual o privilégio foi concedido;
GRANTOR	Usuário que concedeu o privilégio;
PRIVILEGE	Privilégio de objeto;
GRANTABLE	Beneficiado pode conceder ou não o privilégio a outro
HIERARCHY	Privilégio faz parte de uma hierarquia.

PRIVILÉGIOS A GRUPOS

.

CONCEDER PRIVILEGIO PARA UMA ROLE:

GRANT privilégio TO nome_role;

PRIVILÉGIOS:

Consultar Privilégios Concedidos

Privilégio	Observações
ROLE_SYS_PRIVS	Privilégios de sistema concedidos a atribuições
ROLE_TAB_PRIVS	Privilégios de tabela concedidos a atribuições
USER_ROLE_PRIVS	Atribuições acessíveis ao usuário
USER_TAB_PRIVS_MADE	Privilégios de objeto concedidos para os objetos de usuário
USER_COL_PRIVS_RECD	Privilégios concedidos ao usuário em colunas específicas
USER_SYS_PRIVS	Privilégios de sistema concedidos ao usuário

PRIVILÉGIOS: Retirar ou Revogar

RETIRAR PRIVILÉGIO DE SISTEMA:

REVOKE nome_privilégio FROM usuário ou grupo;

RETIRAR PRIVILÉGIO DE OBJETO:

REVOKE nome_privilégio ON objeto

FROM usuário ou grupo;

PRIVILÉGIOS: Observações

Privilégio de Sistema deve estar logado como DBA para conceder ou revogar.

Privilégio de Objeto deve estar logado com o *owner* (proprietário do objeto) para conceder ou revogar.

SAIBA MAIS

- https://www.oracle.com/br/technical-resources/articles/creating-managing-user-roles.html
- https://www.oracle.com/br/database/multitenant/