ENTRENAMIENTO
EXPERIENCIA

GALAXY
TRAINING

SESION 03

ARQUITECTURA MULTITENANT - PDB

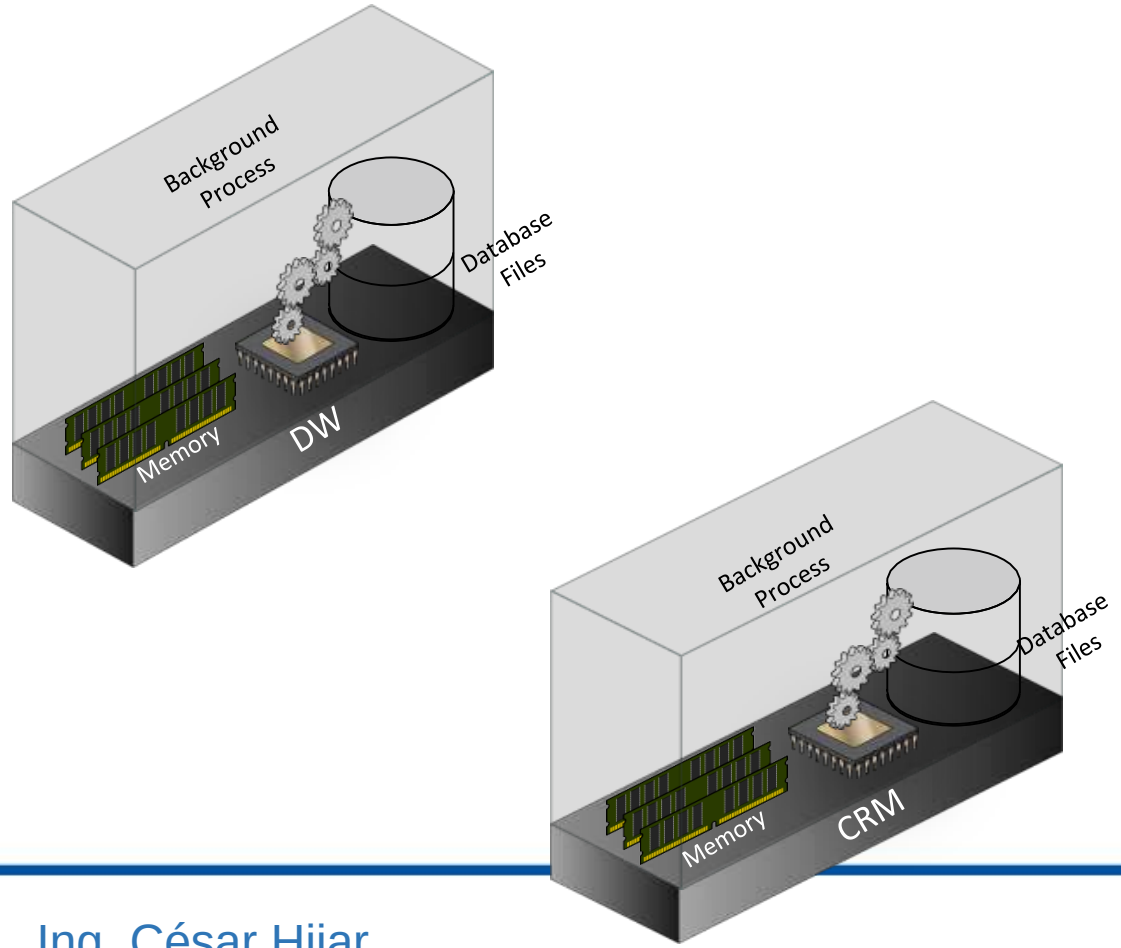
Ing. Cesar Hijar
Instructor

chijar@Gmail.com

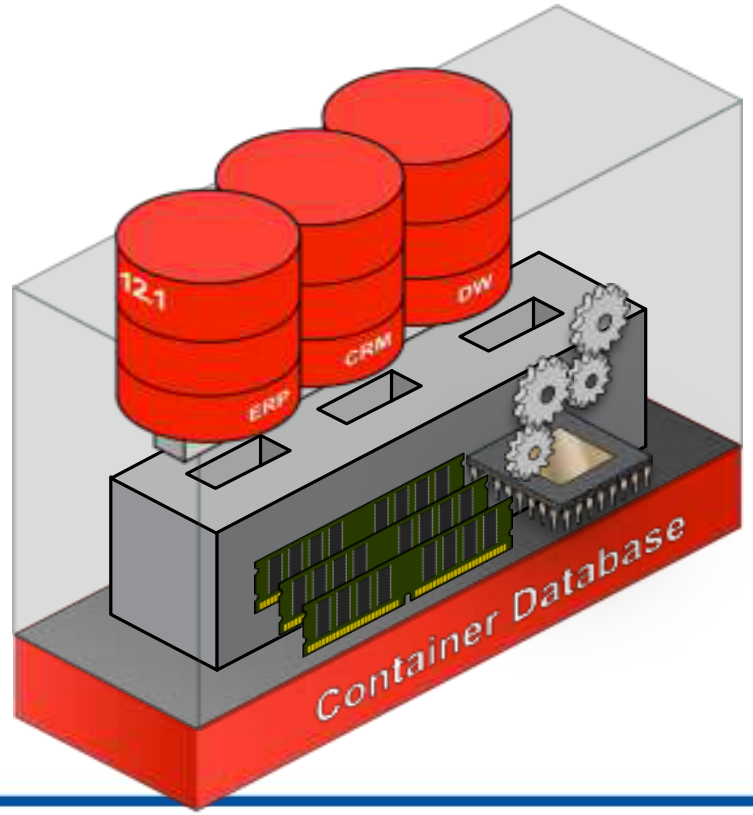
INTRODUCCIÓN A LA BASE DE DATOS 12c (MULTITENANT)

¿Qué es Arquitectura Multitenant?

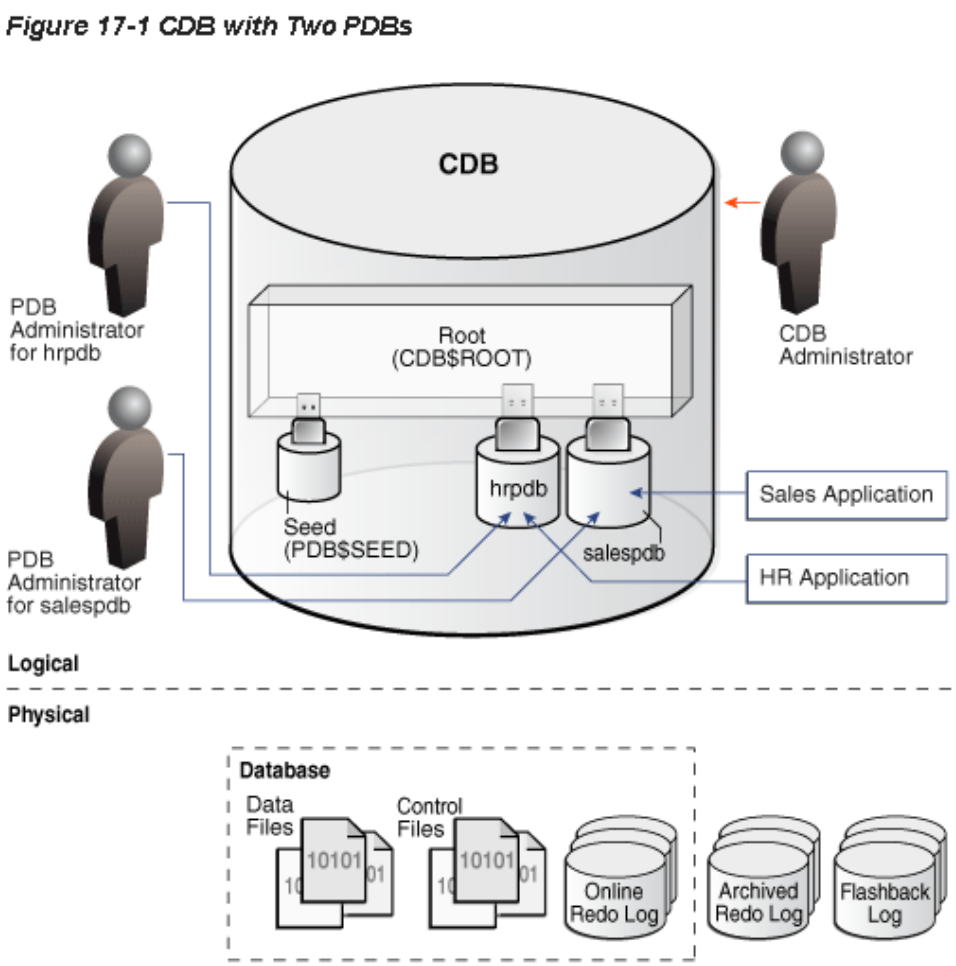
Arquitectura Tradicional

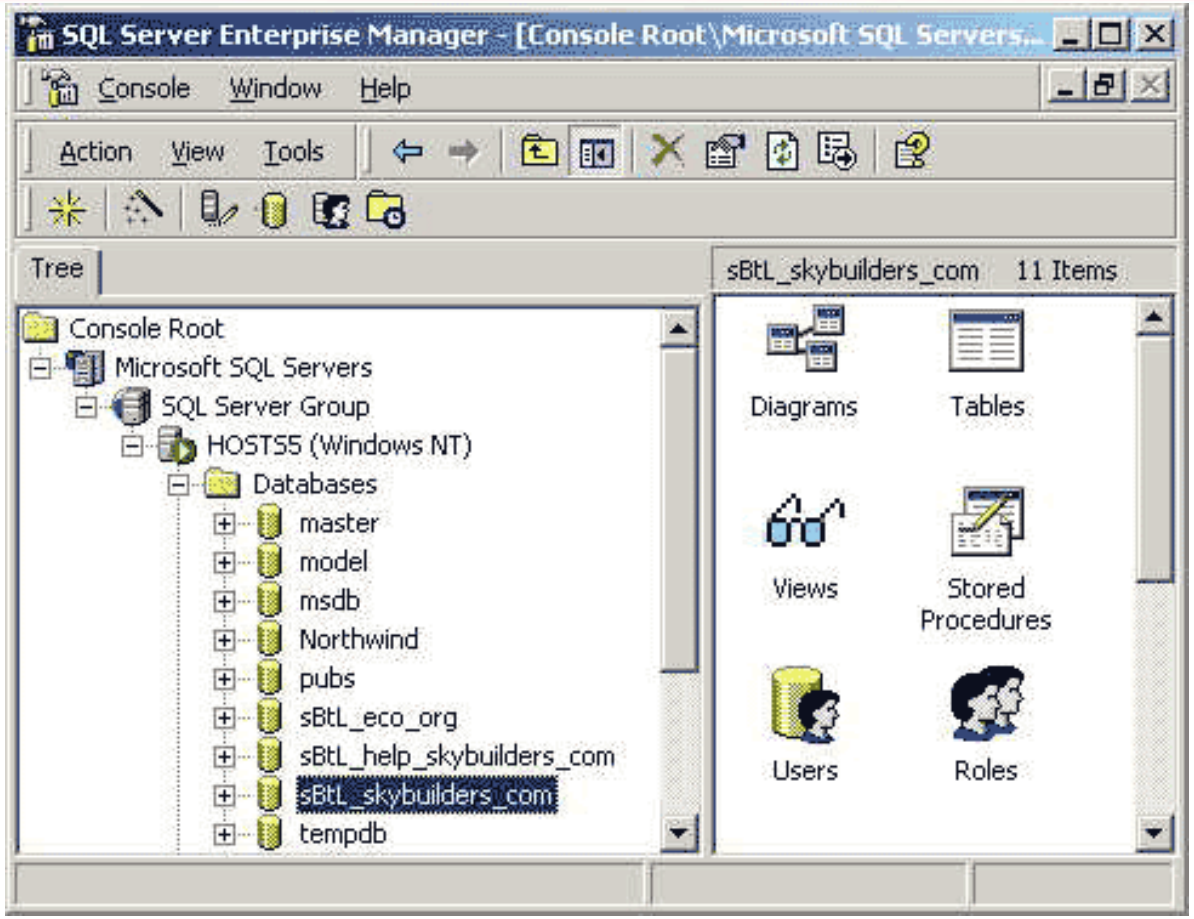


Arquitectura Multitenant

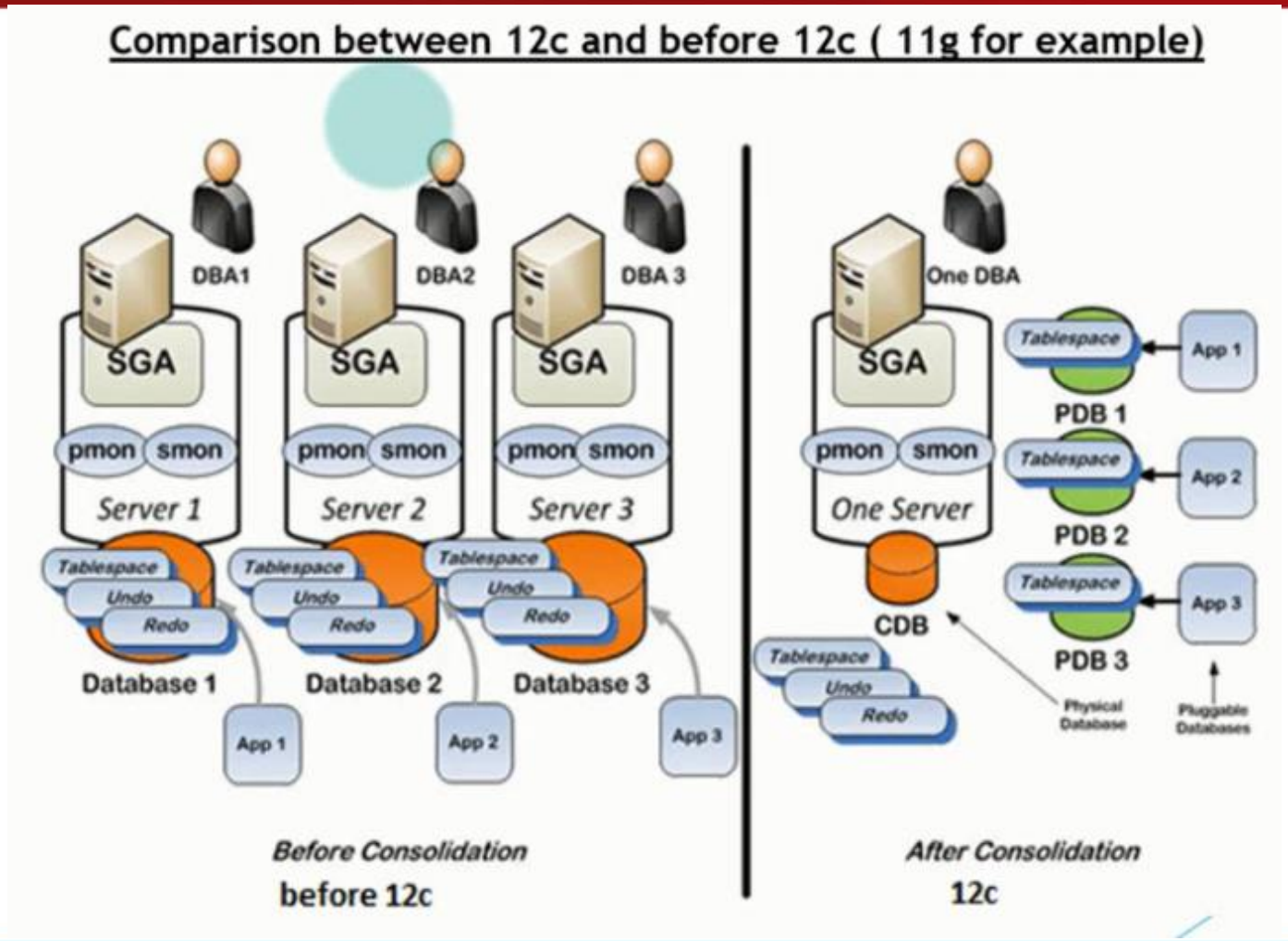


Oracle Multitenant





Oracle Multitenant



Oracle Multitenant

- Aumenta la escalabilidad y la utilización del servidor
- Administra varias bases de datos como si fuera una
- Cumple las expectativas de nivel de servicio con administración de recursos integrados de carga de trabajo

INSTALACIÓN Y CONFIGURACIÓN DE BASE DE DATOS 12c

Planning Your Installation

- What Oracle software are you installing?
- Does the hardware involved meet the minimum required specifications?
- Is there a recommended order of installation when multiple products are involved?
- Are there prerequisite steps that must be performed by someone other than the DBA?

New with 12c

Configuring Oracle Linux with Oracle RDBMS Pre-Install RPM

To configure a server using Oracle Linux and the Oracle RDBMS Pre-Install RPM:

1. Install Oracle Linux.
2. Register your Linux distribution with Oracle Unbreakable Linux Network (ULN), or download and configure the yum repository for your system.
3. Install the RPM:
   - Oracle RDBMS Pre-Install RPM for Oracle Linux 6
   - Oracle Validated RPM with the RPM for Oracle Linux 5

New with 12c

Operating System Groups and Users

Oracle Pre-Install RPM configures:

- `oracle`: Oracle Database installation owner
- `oinstall`: Oracle Inventory group
- `dba`: Oracle administrative privileges group

Environment Variables

The following Oracle environment variables are suggested by the OUI during installation:

- `ORACLE_BASE`: Base of the Oracle directory structure
- `ORACLE_HOME`: Oracle Grid Infrastructure home directory or Oracle Database home directory depending on the product that is being installed

Configuring the Oracle Software Owner Environment

Prior to installing the Oracle software, configure the Oracle software owner environment:

- Set the default file mode creation mask (`umask`) to `022` in the shell startup file.
- Unset Oracle environment variables (`ORACLE_HOME`, `ORACLE_BASE`, `ORACLE_SID`, and `TNS_ADMIN`).
- Remove `$ORACLE_HOME/bin` from your `PATH` environment variable.

Using Oracle Universal Installer (OUI)

Creating Operating System Groups and Users

Create custom configuration groups and users based on job role separation:

- Groups:
  - Oracle Inventory group (`oinstall`)
  - Oracle Grid Infrastructure groups for job role separation:
    - OSDBA (`asmdba`)
    - OSASM (`asmadmin`)
    - OSOPER (`asmoper`)
- Users (software owners):
  - Oracle Grid Infrastructure/Oracle Restart: `grid`

Oracle Database Installation: System Requirements

- Memory requirements for Linux:
  - Minimum 1 GB (2 GB recommended) for the database instance
  - Swap space:
    - 1 GB – 2 GB RAM, swap space = 1.5 times the size of RAM
    - 2 GB – 16 GB RMAN, swap space = size of RAM
    - 16 GB + RAM, swap space = 16 GB
- Disk space requirements for Linux:
  - 6.4 GB for the Oracle Database software (Enterprise Edition) and data files
  - 1 GB of disk space in the `/tmp` directory
  - 4 GB (default) for the fast recovery area (optional)

Types of Installations

- Software-only installation
  - Copies the binaries to the specified location
  - Use Database Configuration Assistant (DBCA) to create the database
- Installation of the software and creation of a database
- Upgrade an existing database

Creando una base de datos con DBCA

Planning the Database

As a DBA, you must plan:

- The logical storage structure of the database and its physical implementation:
  - How many disk drives do you have? What type of storage is being used?
  - How many data files will you need? (Plan for growth.)
  - How many tablespaces will you use?
  - What types of information will be stored?
  - Are there any special storage requirements due to type or size?
- Overall database design
- Database backup strategy

Creando una base de datos con DBCA

Types of Databases

General purpose or transaction processing:

- Online transaction processing (OLTP) system, for example a retail billing system for a software house or a nursery

Custom:

- Multipurpose database (perhaps combined OLTP and data warehouse functionality)

Data warehouse:

- Research and marketing data
- State or federal tax payments
- Professional licensing (doctors, nurses, and so on)

Choosing the Appropriate Character Set

Creando una base de datos con DBCA

Oracle Database supports different classes of character-encoding schemes:

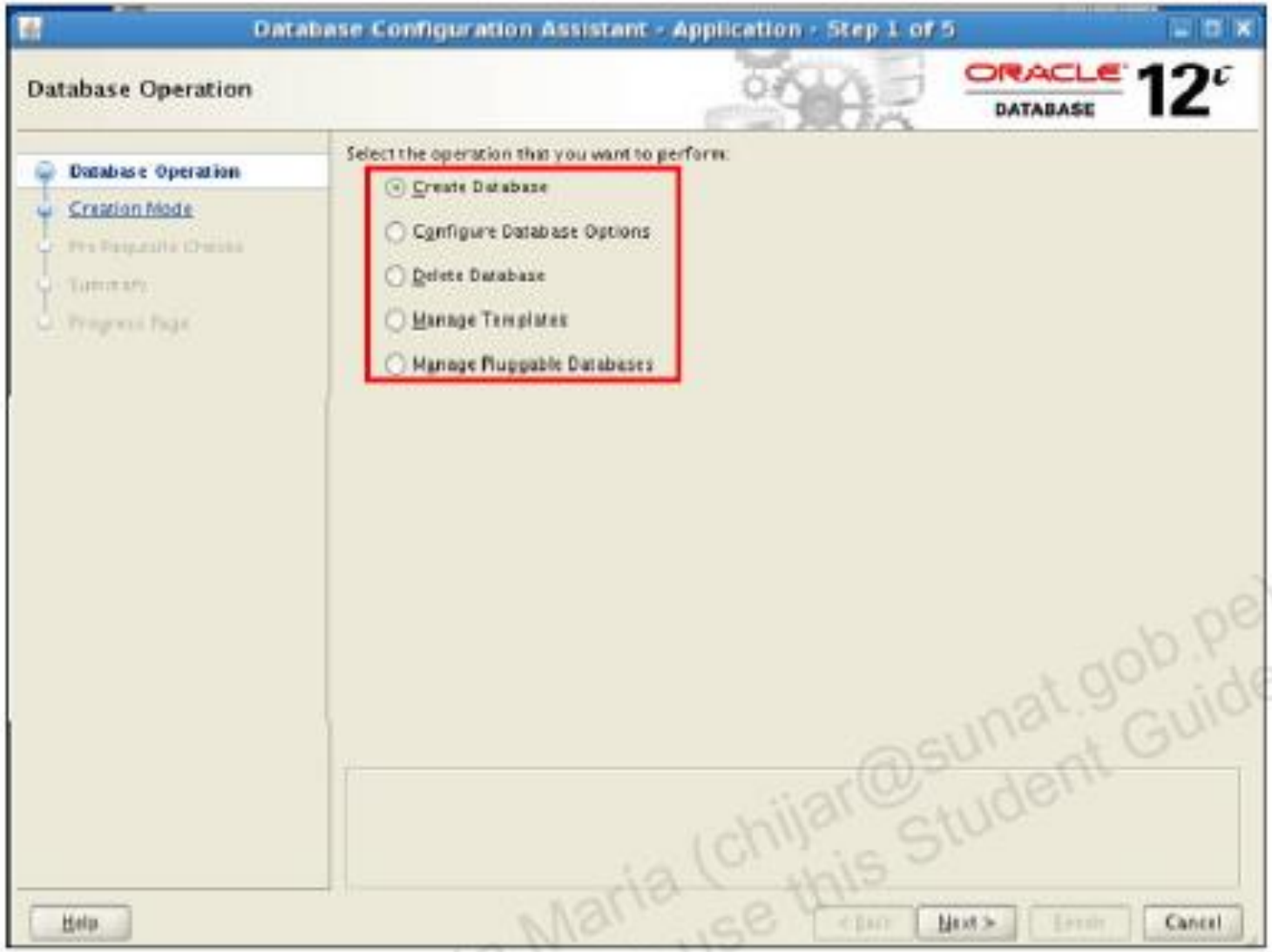
- Single-byte character sets
  - 7-bit
  - 8-bit
- Multibyte character sets, including Unicode

The character set is chosen at the time of database creation. Choose the character set that best meets your business requirements now and in the future because it can be difficult to change character sets later on.

In general Unicode is recommended because it is the most flexible character set.

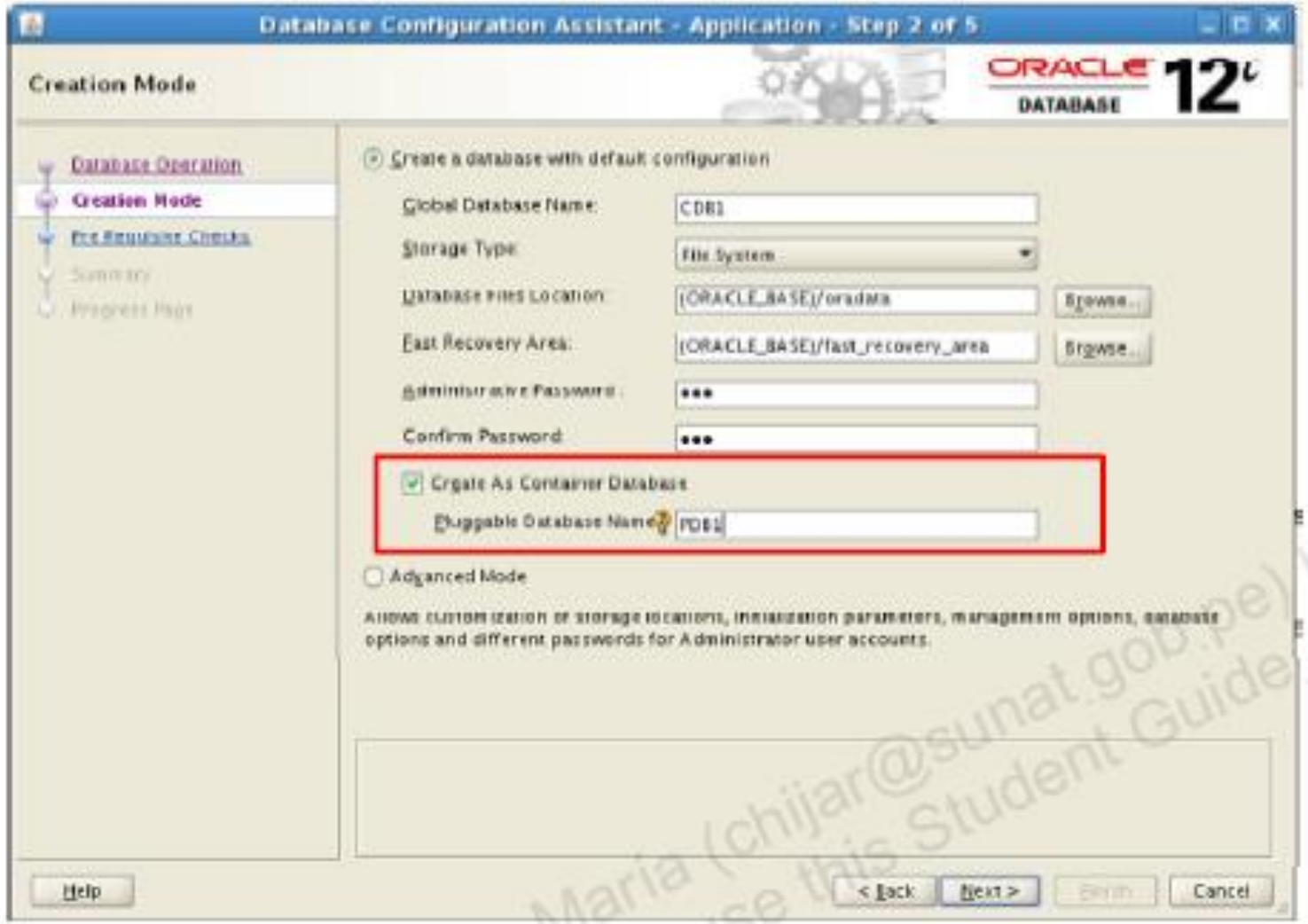
Using the DBCA to Create a Database

Creando una base de datos con DBCA



Creating a Container Database by Using DBCA

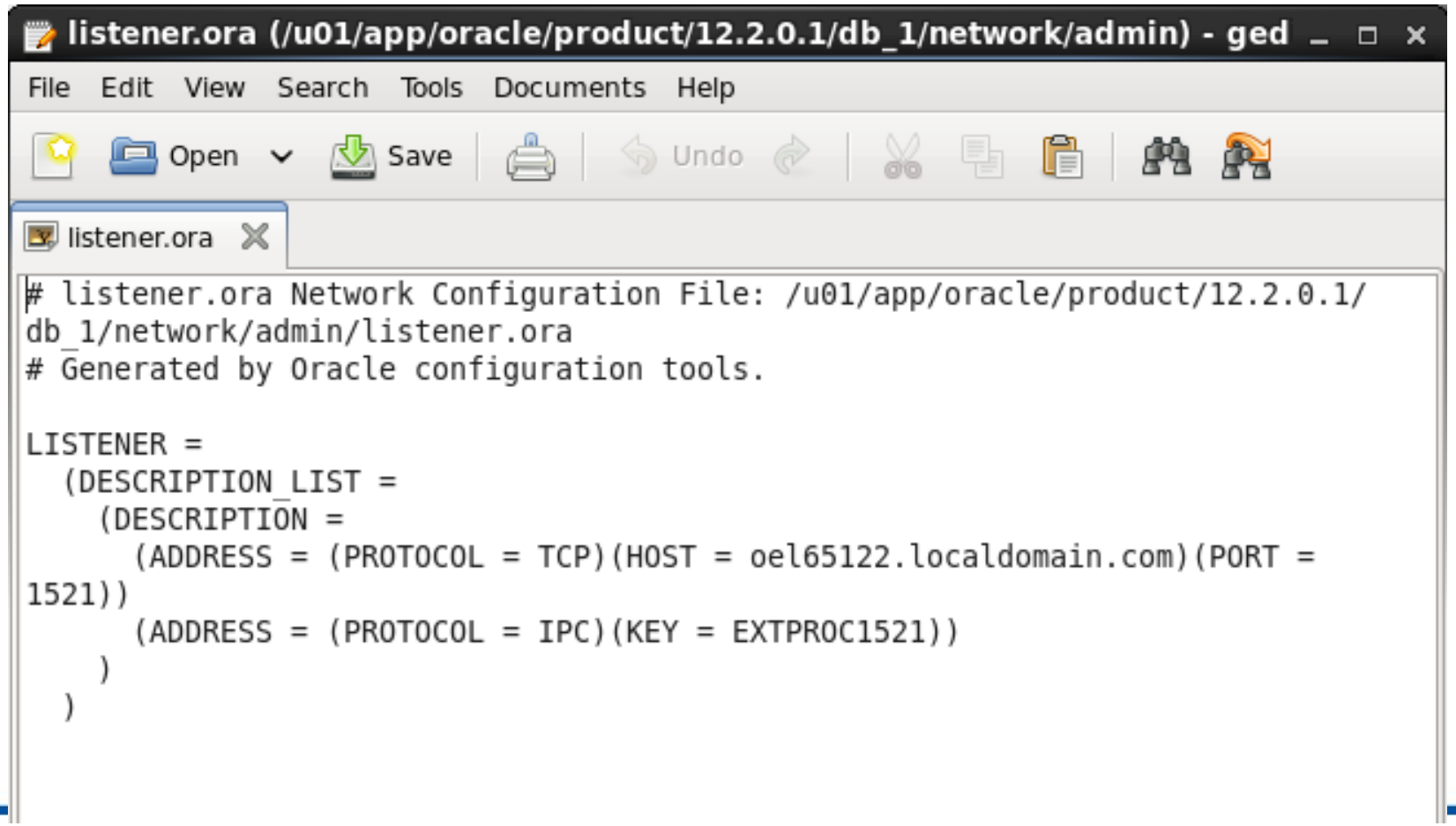
Creando una base de datos con DBCA



CREACION DE BASE DE DATOS PDB

Revisar el archivo listener.ora ubicado en $ORACLE_HOME/network/admin y verifique que el parámetro HOST esté el nombre del hostname

El nombre del host de su equipo se puede visualizar ejecutando desde la ventana de comandos: hostname <ENTER>

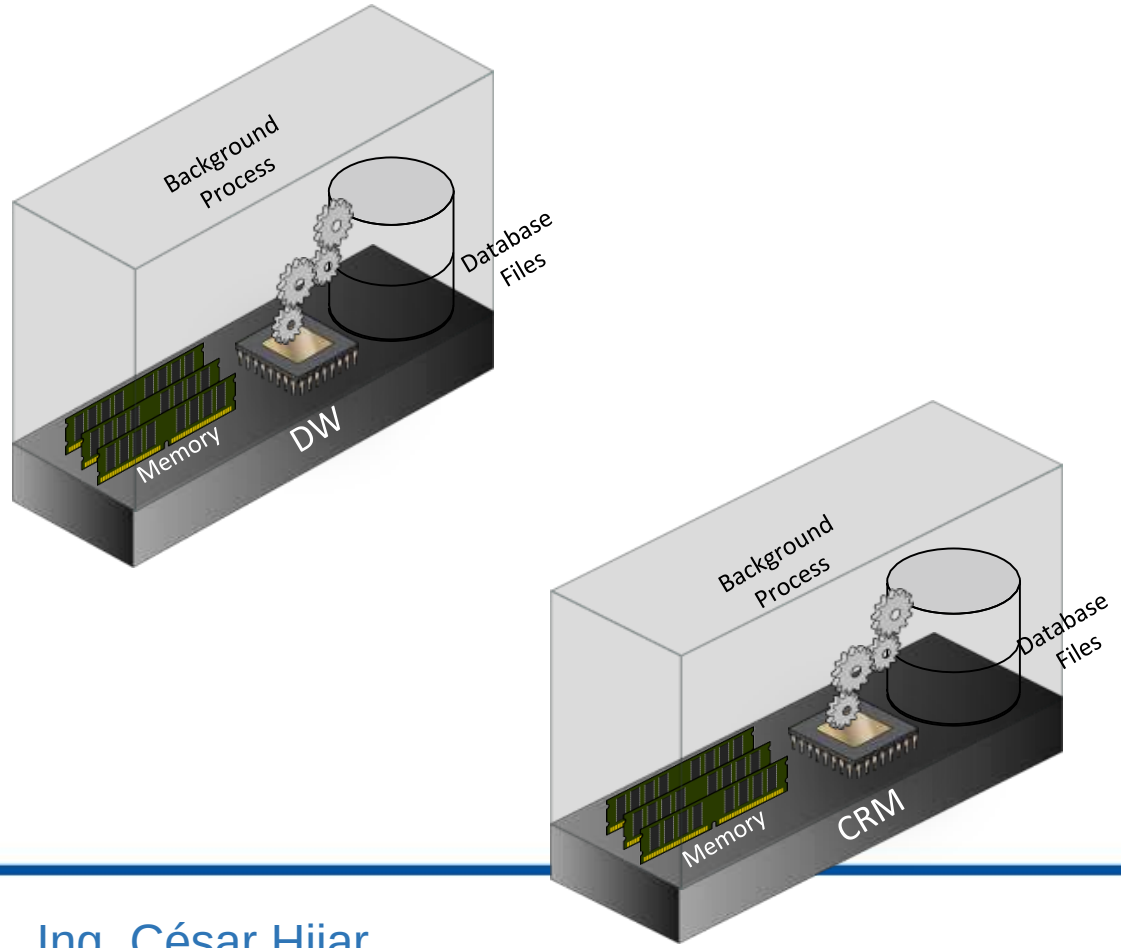


Revisar el estado del listener

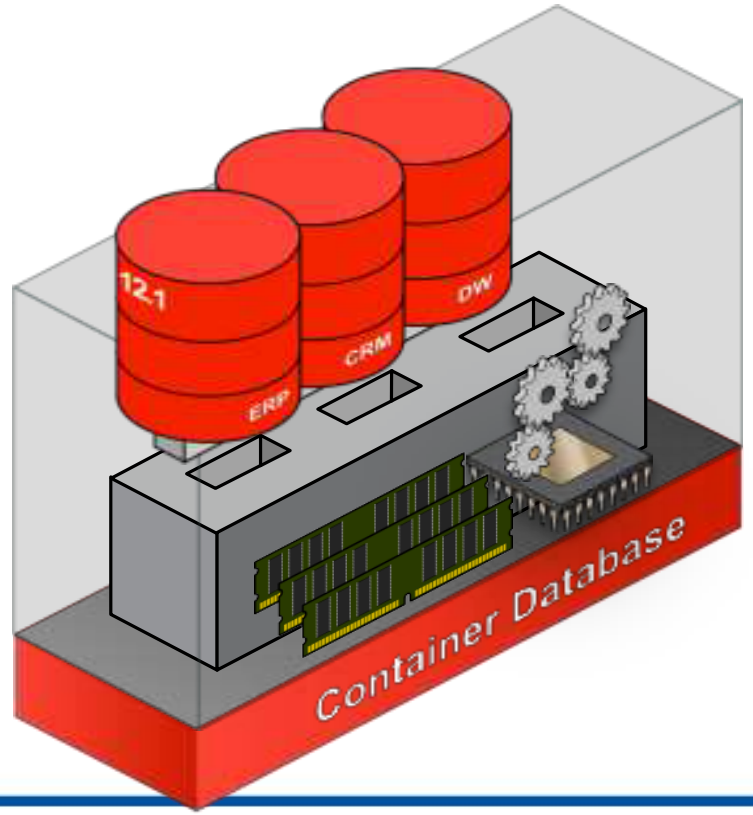
lsnrctl status
lsnrctl start
lsnrctl stop

¿Qué es Arquitectura Multitenant?

Arquitectura Tradicional

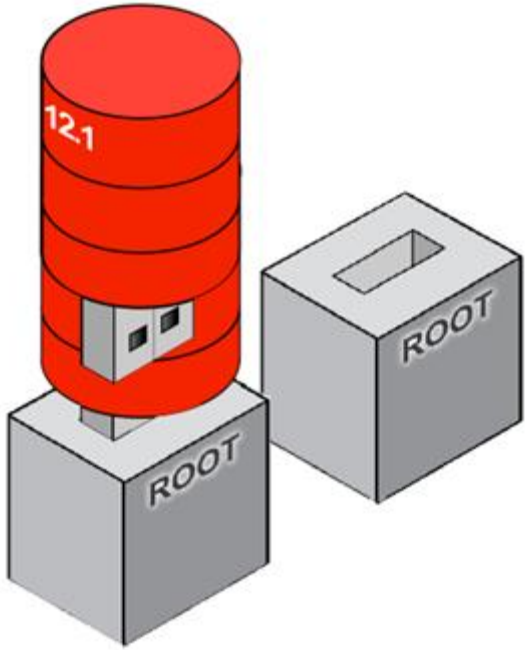


Arquitectura Multitenant

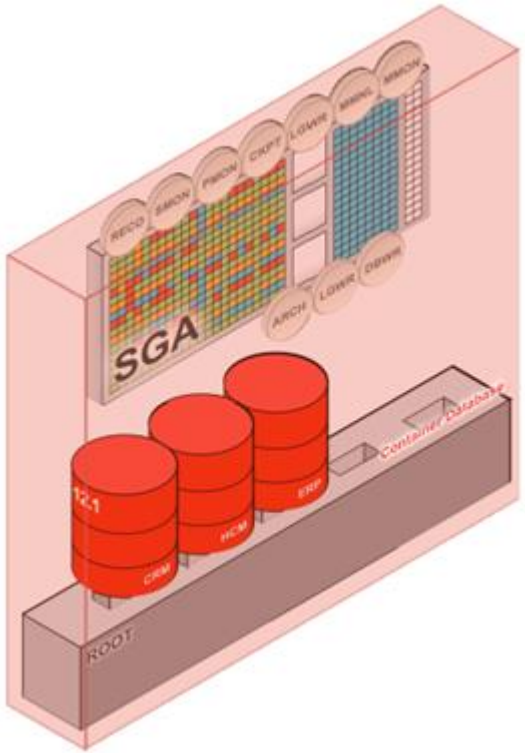


Pluggable Database

- Una PDB es una colección de esquemas, objetos de esquema y objetos sin esquema
- Un contenedor o CDB es también un PDB raíz también llamado root
- Cada CDB tiene los siguientes contenedores:
  - Un root (root container)
  - Un PDB semilla (seed PDB)
  - Ninguna o varias PDBs creadas por usuario
  - La arquitectura soporta hasta 252 PDBs



Arquitectura de una Pluggable Database



- Las PDBs comparten una única SGA
- Las sesiones externas solo pueden ver la PDB a la que se conectan
- Una PDB es totalmente compatible con bases de datos pre 12.1 ordinarias

Comandos para iniciar una PDB

- Sintaxis para iniciar una PDB:
- `alter pluggable database <nombre_pdb> start;`
- Ejemplo:
- `alter pluggable database contabilidad start;`
- Sintaxis para iniciar todas las PDBs
- `alter pluggable database all open;`

Comandos para detener una PDB

- Sintaxis:
- `alter pluggable database <nombre_pdb> stop;`
- Ejemplo:
- `alter pluggable database recursos stop;`
- Sintaxis para detener todas las PDBs
- `alter pluggable database all close;`

Comandos para ver el estado de las PDBs

- Sintaxis:

- `show pdbs;`

```
    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 ORCL                           MOUNTED
```

Inicio automático de una PDB

- Cuando se reinicia el servicio de la CDB, las PDBs quedan en estado MOUNT. Por defecto las PDBs no quedan en estado READ WRITE.
- Para hacer que una PDB se abra automáticamente, se debe de grabar el estado de la misma.
- Sintaxis:
- `alter pluggable database <pdb> save state;`

Nota: si es estado de la PDB es READ WRITE, en el siguiente reinicio se abrirá automáticamente, si el estado está en MOUNTED el siguiente reinicio quedará en ese estado.

Creación de base de datos PDB

1. Ejecuta el DBCA.
2. Seleccionar “Crear Base de Datos de Conexión”
3. Seleccionar la base de datos de contenedor “CDB”
4. Seleccionar “Crear Nueva Base de Datos de Conexión
5. Llenar los datos de Identificación de la nueva base de datos de conexión
6. Se muestra la pantalla de “Resumen”, presionar “Terminar” para iniciar la creación.
7. Se mostrará la pantalla de finalización

Pueden variar algunas opciones o pantallas

Creación de base de datos PDB

- Se puede crear PDB de forma manual, ejemplos:

-
```
CREATE PLUGGABLE DATABASE PDB_0101
ADMIN USER pdb0101 IDENTIFIED BY pdb0101 ROLES=(CONNECT)
file_name_convert = ('pdbseed', 'pdb_0101');
```

-
```
CREATE PLUGGABLE DATABASE test
ADMIN USER test IDENTIFIED BY test ROLES=(CONNECT)
file_name_convert = ('pdbseed', 'test');
```

REVISIÓN DE TAREAS POST-INSTALACION

Revisar…

- Revisión de los servicios registrados en el listener.
- Configurar el TNSNAMES.ORA
- Conexiones y estados de las bases de datos CDB y PDB.

Revisar…

- Revisión de los servicios registrados en el listener. (ejemplo)

```
LSNRCTL> status listener
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=WIN-5TSC2967CPP)(PORT=16
00)))
STATUS of the LISTENER
------------------------
Alias                     listener
Version                   TNSLSNR for 64-bit Windows: Version 12.1.0.2.0 - Produ
ction
Start Date                20-MAY-2017 10:34:57
Uptime                    0 days 0 hr. 5 min. 55 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   C:\app\oracle\product\12.1.0\dbhome_1\network\admin\li
stener.ora
Listener Log File         C:\app\oracle\product\12.1.0\dbhome_1\log\diag\tnslsnr
\WIN-5TSC2967CPP\listener    \alert\log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=WIN-5TSC2967CPP)(PORT=1600)))
The listener supports no services
The command completed successfully
LSNRCTL>
```

Revisar…

- Configurar el TNSNAMES.ORA

Se creó la PDB: orcl. Terminada la creación, editar el archivo $ORACLE_HOME\NETWORK\ADMIN\tnsnames.ora y agregar la siguiente entrada:

```
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = WIN-5TSC2967CPP)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
```

Revisar…

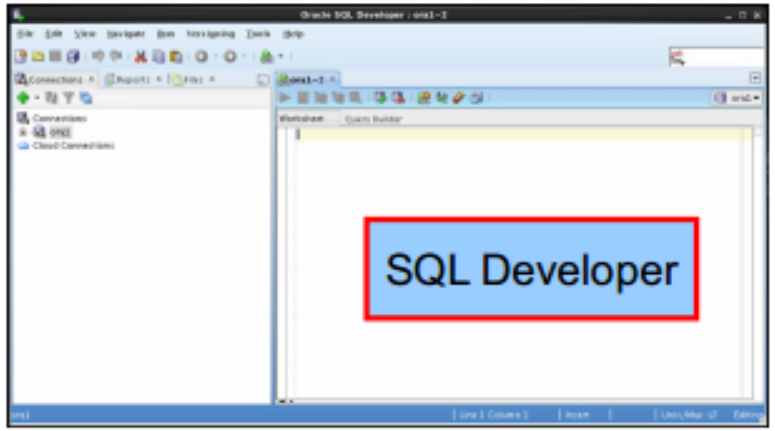
- Conexiones a las bases de datos CDB y PDB:
  - Verificar status de las PDBs
  - Realizar las conexiones.

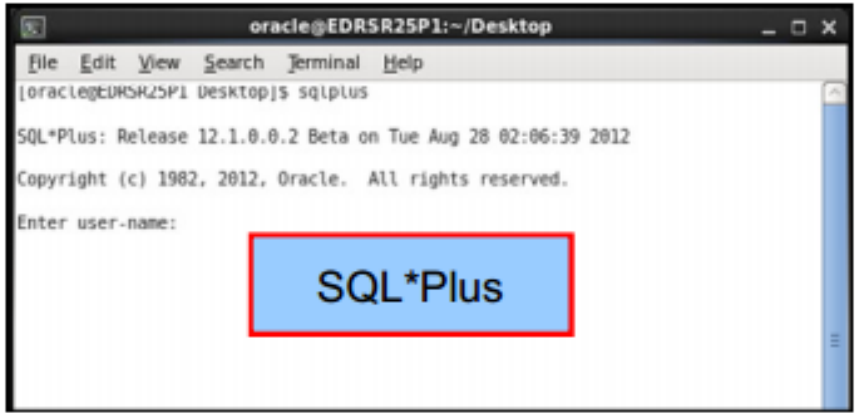
```
SQL> connect sys/oracle@PDB_LIMA as sysdba;
Connected.
SQL> select file_name, tablespace_name, file_id,con_id from cdb_data_files order
 by tablespace name;

FILE_NAME                                          TABLESPACE_NAME  FILE_ID     CON_ID
-------------------------------------------------- ---------------- ---------- ------
----
C:\APP\ORACLE\ORADATA\CDB\PDB_LIMA\SYSAUX01.DBF    SYSAUX                    8          3
C:\APP\ORACLE\ORADATA\CDB\PDB_LIMA\SYSTEM01.DBF    SYSTEM                    7          3
C:\APP\ORACLE\ORADATA\CDB\PDB_LIMA\PDB_LIMA_USERS0 USERS                     9          3
1.DBF
```

Development Environments for SQL

There are two development environments for this course:

- The primary tool is Oracle SQL Developer.
- SQL*Plus command-line interface can also be used.





Comandos Básicos – SQL*Plus

- Para ejecutar la herramienta SQL*Plus debemos cargar la interfaz de comandos del S.O. y ejecutar la siguiente aplicación, según sintaxis:

- SQLPLUS [/nolog]
- SQLPLUS [<usr>/<pwd>[@<identf>] [AS SYSDBA]]

- Ejemplo:
- SQLPLUS /nolog
- SQLPLUS sys/oracle@orcl AS SYSDBA

Comandos Básicos – SQL*Plus

- **CONNECT:**
- Realiza la conexión a una base de datos activa. Su sintaxis es:

- connect <usr>/<pwd>[@<identf>] [AS SYSDBA]
- conn <usr>/<pwd>[@<identf>] [AS SYSDBA]

- **DESCRIBE (DESC):**
- Permite ver la estructura de una tabla. Ejemplo:

- DESC dba_tables;

Comandos Básicos – SQL*Plus

- **SHOW SGA:**
- Permite ver la información de la instancia (SGA).
- Ejemplo:

- sqlplus> show sga

- **SHOW USER:**
- Permite ver con que usuario estamos conectados. Ejemplo:

- sqlplus> show user

Comandos Básicos – SQL*Plus

- **DISCONNECT:**
- Permite cerrar la conexión con el usuario actual. Tener en cuenta que la base de datos no es cerrada. Ejemplo:

- sqlplus> disconnect ó disc

SQL*Plus File Commands

- SAVE *filename*
- GET *filename*
- START *filename*
- @ *filename*
- EDIT *filename*
- SPOOL *filename*
- EXIT

Command	Description
SAV[E] *filename* [.ext] [REP[LACE]APP[END]]	Saves the current contents of the SQL buffer to a file. Use APPEND to add to an existing file; use REPLACE to overwrite an existing file. The default extension is .sql.
GET *filename* [.ext]	Writes the contents of a previously saved file to the SQL buffer. The default extension for the file name is .sql.
STA[RT] *filename* [.ext]	Runs a previously saved command file
@ *filename*	Runs a previously saved command file (same as START)
ED[IT]	Invokes the editor and saves the buffer contents to a file named afiedt.buf
ED[IT] [*filename*[.ext]]	Invokes the editor to edit the contents of a saved file
SPO[OL] [*filename*[.ext]|OFF|OUT]	Stores query results in a file. OFF closes the spool file. OUT closes the spool file and sends the file results to the printer.
EXIT	Quits SQL*Plus

Comandos Básicos con SQL*Plus

Base de datos ejemplo