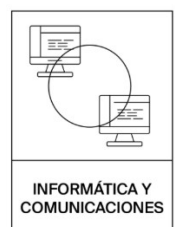
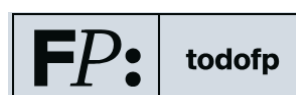


# Administración de Sistemas Gestores de Bases de Datos



## INSTALACIÓN DE UN SGBD ORACLE



# UD 1 . Instalación de un SGBD Oracle

## Resumen del contenido:

### 1. Ediciones y versiones de Oracle

- Diferentes versiones disponibles: **Enterprise, Standard, Express (gratuita), Personal, Lite, Cloud Services**.
- Explicación sobre los tipos de instalación: **Standalone, Grid Infrastructure, Real Application Cluster**.

### 2. Requisitos del sistema

- Mínimos y recomendados para Windows y Linux.
- Espacio en disco, memoria RAM, resolución gráfica, swap en Linux, etc.

### 3. Componentes principales

- Herramientas de instalación y configuración: **setup, dbca, netca, netmgr, RMAN**.
- Aplicaciones de gestión: **SQL\*Plus, SQL Developer, DBeaver, Enterprise Manager**.

### 4. Arquitectura de Oracle

- **Instancias y bases de datos:** Memoria (SGA y PGA) y procesos en segundo plano.
- **Modelo Multitenant (desde Oracle 12c):** Uso de contenedores (CDB) y bases de datos pluggables (PDB).
- **Estructura de almacenamiento** según el estándar OFA (Optimum Flexible Architecture).

### 5. Proceso de instalación

- **Windows:** Creación de usuarios en el sistema antes de ejecutar **setup.exe**.
- **Linux:** Configuración del entorno antes de la instalación (scripts, variables de entorno).
- Instalación del SGBD y creación de una base de datos (CDB y PDB).

### 6. Acceso y gestión de Oracle

- Uso de **SQL\*Plus** para conectarse como SYS y ejecutar comandos básicos.
- Uso de **SQL Developer** y **DBeaver** como interfaces gráficas.
- Administración de bases de datos mediante comandos SQL (**CREATE PLUGGABLE DATABASE, DROP PLUGGABLE DATABASE**).

### 7. Creación y eliminación de bases de datos

- **Manual y con DBCA** en Windows y Linux.
- Creación de CDBs y PDBs mediante comandos SQL.

### 8. Conexión a Oracle Database

- Acceso desde **SQL\*Plus, SQL Developer** y otros clientes.
- Uso de **ORDS** (Oracle REST Data Services) para servicios web.

## Punto 1. Ediciones y versiones de Oracle Database

Antes de instalar Oracle Database, es importante elegir la **versión y edición** adecuadas según el uso que se le dará. Oracle ofrece varias opciones con diferentes características, licencias y costos.

---

### Ediciones principales de Oracle Database

Las ediciones de Oracle se dividen en **gratuitas y de pago**. A continuación, te explico las diferencias entre ellas:

#### Ediciones de pago

##### 1. Enterprise Edition (EE)

- La más completa, con todas las características avanzadas.
- Ideal para empresas grandes con altos volúmenes de datos y requisitos de alta disponibilidad.
- Incluye **particionamiento de datos, seguridad avanzada, clustering y análisis en memoria**.

##### 2. Standard Edition (SE)

- Tiene menos características que Enterprise, pero sigue siendo potente.
- Pensada para pequeñas y medianas empresas.
- No admite **Real Application Clusters (RAC)** en versiones más recientes.

##### 3. Standard Edition One (SE1) (Descontinuada)

- Versión reducida de Standard Edition.
- Limitada a un número menor de procesadores.
- Ya no está disponible en versiones nuevas.

##### 4. Standard Edition 2 (SE2)

- Reemplaza SE y SE1 en las versiones más recientes.
- Limitada a 2 sockets de CPU, pero permite algunas funcionalidades de clustering.

##### 5. Personal Edition (PE)

- Similar a Enterprise Edition, pero para uso personal.
- No permite conexiones remotas, ideal para desarrolladores.
- Solo disponible en Windows.

##### 6. Lite Edition (LE)

- Versión reducida para dispositivos móviles y sistemas embebidos.
  - Muy poco utilizada actualmente.
-

## Ediciones gratuitas

### 1. Express Edition (XE)

- **Totalmente gratuita.**
- Limitada a 2 CPU, 2 GB de RAM y 12 GB de almacenamiento.
- Ideal para pruebas, aprendizaje y pequeñas aplicaciones.
- No incluye soporte técnico oficial.

### 2. Oracle Database Free 23ai (Nueva en Oracle 23c)

- Sustituye a XE en versiones recientes.
  - Incluye algunas funcionalidades avanzadas, pero sigue teniendo limitaciones de hardware.
- 

## Opciones en la nube (Oracle Cloud Services)

Oracle también ofrece su base de datos en la nube, con varias opciones según el rendimiento necesario:

1. **Cloud Service SE** → Versión estándar en la nube.
  2. **Cloud Service EE** → Versión empresarial con más recursos.
  3. **Cloud Service EE-High Performance** → Incluye características avanzadas como particionamiento.
  4. **Cloud Service EE-Extreme Performance** → La más potente, con todas las herramientas avanzadas de Oracle.
  5. **Exadata Cloud Service** → Ejecuta Oracle en servidores Exadata optimizados.
- 

## ¿Cuál elegir?

Dependerá de lo que necesites:

- ✓ Si es para aprender o desarrollar, usa **Express Edition (XE)** o **Oracle Database Free**.
  - ✓ Si es para una pequeña empresa, **Standard Edition 2 (SE2)** es suficiente.
  - ✓ Si es para una gran empresa con alta demanda, usa **Enterprise Edition (EE)**.
  - ✓ Si quieres pagar solo por lo que usas y no preocuparte por hardware, elige **Oracle Cloud**.
-

## Punto 2. Requisitos del sistema para instalar Oracle Database

Para instalar **Oracle Database**, es importante cumplir con ciertos requisitos de hardware y software. En el documento se presentan dos tipos de requisitos:

1. **Requisitos teóricos** (mínimos que Oracle recomienda oficialmente).
  2. **Requisitos realistas** (lo que realmente se necesita para un funcionamiento óptimo).
- 

### 2.1. Requisitos teóricos

Estos son los mínimos que Oracle exige para la instalación en **Windows y Linux**:

#### Hardware:

- **Memoria RAM:** mínimo **1GB**.
- **Espacio en disco:**
  - **1GB** en la carpeta temporal.
  - **8GB** para la instalación del SGBD.
  - Espacio adicional para las bases de datos.
- **Procesador:** No se menciona en los requisitos mínimos, pero se recomienda un CPU moderno.
- **Gráfica:** Resolución **1024x768 con 256 colores**.

#### Software:

- **Sistema operativo compatible:** Windows o Linux (con un kernel compatible).
  - **Swap en Linux:** Igual a la cantidad de RAM disponible.
  - **Conectividad a Internet** (para descargas y actualizaciones).
- 

### 2.2. Requisitos realistas

En la práctica, para que Oracle Database funcione bien, se recomienda lo siguiente:

#### Hardware:

- **Memoria RAM:** **2GB o más**, y al menos **2 núcleos de procesador**.
  - **Espacio en disco:**
    - **Windows:** Mínimo **50GB** (incluyendo el sistema operativo, Oracle y al menos una base de datos).
    - **Linux:** Similar a Windows, aunque el espacio puede variar según la configuración.
  - **Swap en Linux:** **Doble de la RAM disponible** (si tienes 2GB de RAM, necesitas 4GB de swap).
-

## Software y configuraciones adicionales:

- **Usuarios en el sistema:** Se recomienda crear un usuario "**oracle**" para la instalación y administración del SGBD.
  - **Herramientas y componentes esenciales:**
    - setup → Instalador principal.
    - dbca → Creación de bases de datos.
    - netca → Configuración del listener para conexiones remotas.
    - netmgr → Gestión de conexiones de red.
    - RMAN → Copias de seguridad.
  - **Aplicaciones cliente para gestionar Oracle:**
    - SQL\*Plus
    - SQL Developer
    - DBeaver
- 

## 2.3. Otros detalles importantes

- La instalación se debe ejecutar como **administrador** en Windows o con permisos de **root** en Linux.
  - Oracle recomienda una estructura de almacenamiento basada en su **estándar OFA (Optimum Flexible Architecture)** para organizar los archivos de la base de datos.
  - Si la instalación se hace en **servidores**, los requisitos pueden variar según la carga de trabajo y el número de usuarios concurrentes.
- 

## Conclusión

Los requisitos mínimos pueden permitir la instalación de Oracle Database, pero para un rendimiento óptimo es mejor seguir las recomendaciones realistas. La memoria RAM y el espacio en disco son factores clave para evitar problemas de rendimiento y asegurar un funcionamiento estable.

---

## Punto 3: Componentes principales de Oracle Database

Cuando instalas Oracle Database, vienen incluidos varios componentes esenciales que permiten su configuración, administración y gestión. Se pueden clasificar en dos categorías principales:

1. **Herramientas de instalación y configuración**
  2. **Aplicaciones clientes de gestión**
- 

### 3.1. Herramientas de instalación y configuración

Estas herramientas ayudan en la instalación y configuración inicial del sistema gestor de bases de datos (SGBD).

**setup** → Es el programa de instalación principal de Oracle. Se ejecuta en Windows o Linux para instalar el software del SGBD.

**dbca (Database Configuration Assistant)** → Sirve para crear, configurar y eliminar bases de datos Oracle de manera gráfica o por línea de comandos.

**netca (Network Configuration Assistant)** → Configura los "listeners" (servicios de escucha) que permiten que los clientes se conecten a la base de datos a través de la red.

**netmgr (Network Manager)** → Otra herramienta para gestionar la configuración de conexiones de red en Oracle.

**RMAN (Recovery Manager)** → Es la herramienta de copias de seguridad y recuperación de Oracle. Permite realizar backups de bases de datos y restaurarlas en caso de fallos.

---

### 3.2. Aplicaciones clientes de gestión

Estos son programas que permiten interactuar con la base de datos mediante comandos SQL o interfaces gráficas.

**SQL\*Plus** → Es una herramienta de línea de comandos que permite ejecutar consultas SQL y administrar la base de datos de manera directa.

**Ejemplo de uso:**

```
SQL> show user;  
SQL> select name from v$database;
```

**SQL Developer** → Interfaz gráfica de Oracle que facilita la gestión de bases de datos, la ejecución de consultas y la creación de objetos como tablas, índices y procedimientos almacenados.

**SQLcl (SQL Command Line)** → Es una versión moderna de SQL\*Plus que ofrece autocompletado, historial de comandos y compatibilidad con scripts.

**DBeaver** → Cliente multiplataforma para bases de datos (compatible con Oracle, MySQL, PostgreSQL, etc.), útil para administrar bases de datos con una interfaz amigable.

---

**VSCode + Extension SQL Developer de Oracle** → Extensión para Visual Studio Code, que permite interactuar con Bases de Datos de ORACLE. Permite integrar la compatibilidad de este editor e IDE con ORACLE.

**Enterprise Manager (EM)** → Actualmente ya no se utiliza en las nuevas versiones. Herramienta de administración web de Oracle que permite monitorear y gestionar bases de datos. *Nota: Está en proceso de ser reemplazada por otras soluciones más modernas.*

---

## Resumen

Este punto describe los componentes esenciales de Oracle Database, diferenciando entre herramientas necesarias para instalar y configurar el sistema y las aplicaciones que permiten su uso y administración.

---

## Punto 4: Arquitectura de Oracle

Oracle Database tiene una arquitectura estructurada en instancias y bases de datos, con la posibilidad de gestionar múltiples bases de datos dentro de un mismo sistema. Se divide en dos modelos principales: **arquitectura tradicional (non-CDB)** y **arquitectura Multitenant (CDB y PDBs)**.

---

### 4.1. Instancia y Base de Datos

Una **instancia** en Oracle es un conjunto de procesos y estructuras de memoria que gestionan una base de datos. Cuando un usuario se conecta a Oracle, lo hace a una instancia.

- **SGA (System Global Area):**
  - Es la memoria compartida que Oracle usa para almacenar datos en caché y gestionar las conexiones.
  - Se asigna cada vez que una instancia se inicia.
- **PGA (Program Global Area):**
  - Es la memoria exclusiva para cada proceso del usuario, no es compartida.
  - Maneja la información de ejecución de consultas.
- **Procesos en Background:**
  - Son procesos automáticos que gestionan la base de datos (por ejemplo, lectura y escritura en disco, gestión de conexiones, recuperación en caso de fallo).
- **Estructura de almacenamiento:**
  - La **BBDD (Base de Datos)** en Oracle está compuesta por múltiples archivos en disco.



- Contiene tablespaces (espacios lógicos), datafiles (archivos físicos) y logs de control.
- 

## 4.2. Arquitectura tradicional (non-CDB)

Antes de Oracle 12c, cada base de datos tenía su propia instancia y conjunto de procesos.

Esto implicaba más consumo de recursos si había varias bases de datos en un mismo sistema.

- Una sola base de datos por cada instancia.
  - Mayor consumo de memoria y procesos independientes.
  - Dificultad para gestionar múltiples bases de datos.
- 

## 4.3. Arquitectura Multitenant (CDB y PDBs)

Desde **Oracle 12c**, se introdujo el modelo **Multitenant**, que permite gestionar varias bases de datos dentro de un único contenedor (CDB).

- **CDB (Container Database):** Es la base de datos principal que contiene varias bases de datos pluggables.
- **PDBs (Pluggable Databases):** Son bases de datos individuales que funcionan dentro de una CDB.
- **Ventajas del modelo Multitenant:**
  - **Ahorro de recursos**, ya que las PDBs comparten la misma instancia.
  - **Mayor flexibilidad**, permitiendo mover PDBs entre servidores fácilmente.
  - **Gestión centralizada**, administrando varias bases de datos desde una única CDB.

**Estructura de una CDB:**

- **CDB\$ROOT:** Contiene los metadatos y la configuración general.
- **PDB\$SEED:** Es una plantilla para crear nuevas PDBs.
- **PDB1, PDB2, PDB3...:** Bases de datos de usuario.

**A partir de Oracle 21c, todas las bases de datos deben ser Multitenant.**

---

## 4.4. Estándar OFA (Optimal Flexible Architecture)

Oracle recomienda seguir el estándar **OFA** para organizar archivos de base de datos y mejorar la administración:

- **En Linux:**
  - `/u01/app/oracle/product/21.0.0`
- **En Windows:**
  - `C:\app\oracle\product\21.0.0`

Este estándar ayuda a:

- ✓ Organizar el software y datos en disco.
-

- ✓ Simplificar la administración de archivos.
  - ✓ Mejorar el rendimiento y la escalabilidad.
- 

## Conclusión

- La **instancia** es la combinación de memoria y procesos en ejecución.
  - La **base de datos** es la estructura de almacenamiento en disco.
  - El **modelo Multitenant (CDB y PDBs)** permite gestionar varias bases de datos de manera más eficiente.
  - **OFA facilita la organización y administración de los archivos en el sistema.**
- 

## Punto 5. Proceso de Instalación de Oracle Database

La instalación de Oracle no solo implica instalar el software del **Sistema Gestor de Bases de Datos (SGBD)**, sino también la configuración previa y la posterior creación de bases de datos.

### 5.1. Instalación en Windows

Oracle proporciona un **programa ejecutable** (`setup.exe`) que guía todo el proceso de instalación, pero antes de ejecutarlo, hay que realizar algunas configuraciones previas.

#### Pasos previos en Windows

##### 1. Crear usuarios en el sistema operativo

- Se recomienda tener al menos **tres usuarios** en Windows:
  - `admin` → Usuario principal del sistema
  - `oracle` → Usuario con permisos de administrador (para ejecutar la instalación)
  - `usuario1` → Usuario sin permisos de administrador
- El usuario `oracle` será el que ejecutará la instalación.

##### 2. Ejecutar el instalador como administrador

- Se ejecuta el archivo **`setup.exe`** con permisos de administrador.

##### 3. Opciones de instalación

- Se puede elegir instalar **solo el software** o instalarlo junto con la **creación de una base de datos**:
  - **Solo software**: Instala Oracle sin crear una base de datos.
  - **Software + CDB/PDB**: Además de instalar Oracle, crea una **Base de Datos Contenedora (CDB)** y una **Base de Datos Pluggable (PDB)**.

##### 4. Importante

- **No ejecutar `setup.exe` más de una vez simultáneamente**, ya que puede generar errores.
- 

## 5.2. Instalación en Linux

La instalación en Linux es un poco más compleja porque se deben configurar previamente el sistema y las variables de entorno.

### Pasos previos en Linux

#### 1. Preparar el sistema

- Verificar que el kernel del sistema sea compatible con Oracle.
- Configurar la memoria `swap` (debe ser el doble de la memoria RAM).
- Crear los usuarios y grupos necesarios (`oracle`, `dba`).
- Definir variables de entorno (`ORACLE_HOME`, `ORACLE_SID`).

#### 2. Ejecutar la instalación

- **Ejecutar el script de instalación** (`runInstaller` o `setup`) desde el usuario `oracle`.
- Seguir el asistente gráfico para completar la instalación.

#### 3. Post-instalación

- Configurar archivos de arranque (`.bash_profile`, `.bashrc`).
  - Verificar que los servicios de Oracle estén funcionando (`lsnrctl start`, `sqlplus / as SYSDBA`).
- 

## 5.3. Diferencia entre instalar el SGBD y crear una base de datos

- **Instalar el SGBD**: Solo instala el software de Oracle Database.
- **Crear una base de datos**: Se puede hacer automáticamente con la instalación o manualmente después usando `dbca` (Database Configuration Assistant).

Para la creación manual de bases de datos, se utiliza el comando:

`C:\Windows\system32> dbca` **(con permisos de Administrador)**

o en Linux:

`$ dbca`

---

## Resumen del punto 5

- En **Windows**, se recomienda crear usuarios antes de instalar y usar `setup.exe` con permisos de administrador.
-

- En **Linux**, es necesario preparar el sistema antes de ejecutar la instalación.
  - Se puede instalar solo el **software** o hacerlo junto con la **creación de una base de datos**.
  - Para crear bases de datos después de la instalación, se usa la herramienta **dbca**.
- 

## Punto 6. Acceso y gestión de Oracle

Una vez que **Oracle Database** está instalado, es necesario conectarse a él para administrarlo, crear bases de datos, gestionar usuarios, y ejecutar consultas SQL. Para ello, existen varias herramientas y métodos de acceso:

---

### 6.1. Uso de SQL\*Plus

#### ¿Qué es SQL\*Plus?

- **SQL\*Plus** es una herramienta de línea de comandos que permite interactuar con Oracle Database.
- Es útil para ejecutar comandos SQL y PL/SQL, crear usuarios, asignar permisos, etc.

#### Cómo acceder desde Windows

1. Abrir una terminal **CMD** como administrador.
2. Ejecutar:

```
sqlplus / as SYSDBA
```

- `sqlplus` → Lanza la herramienta.
- `/ as SYSDBA` → Accede como el usuario administrador de la base de datos.

3. Comandos básicos en SQL\*Plus:

```
show user;           -- Muestra el usuario conectado.
show con_name;       -- Muestra el nombre de la base de datos conectada.
select name from v$database; -- Consulta el nombre de la base de datos.
show pds;           -- Muestra las bases de datos pluggables (PDBs).
conn system;        -- Conecta como usuario "system".
disc;               -- Desconecta la sesión.
exit;               -- Salir de SQL*Plus.
```

4. Para cambiar entre bases de datos:

```
alter session set container=NOMBRE_PDB;
```

- Permite cambiar de una base de datos contenedora (**CDB**) a una base de datos pluggable (**PDB**).

## 6.2. Uso de SQL Developer

### ¿Qué es SQL Developer?

- **SQL Developer** es una **herramienta gráfica (GUI)** gratuita de Oracle.
- Permite conectarse a Oracle Database sin necesidad de usar la línea de comandos.
- Se utiliza para escribir y ejecutar consultas SQL, administrar usuarios, ver estructuras de bases de datos, etc.

### Cómo acceder con SQL Developer

1. Abrir **SQL Developer**.
  2. Crear una nueva conexión:
    - Nombre: **MiConexionOracle**
    - Usuario: **system**
    - Contraseña: **(la que definiste en la instalación)**
    - Hostname: **localhost**
    - Puerto: **1521**
    - SID / Service Name: **(Nombre de tu base de datos o PDB)**
  3. Hacer clic en **Probar conexión** y luego **Conectar**.
- 

## 6.3. Uso de otras herramientas gráficas

Además de SQL Developer, existen otras herramientas para gestionar Oracle:

Herramienta	Descripción
DBeaver	Cliente de bases de datos multiplataforma compatible con Oracle, MySQL, PostgreSQL, etc.
TOAD	Software avanzado para DBA y desarrolladores. Tiene versión de pago.
ORDS (Oracle REST Data Services)	Permite gestionar bases de datos a través de servicios web, similar a phpMyAdmin para MySQL.

---

## 6.4. Conexión desde Linux

Para acceder a Oracle desde un entorno Linux:

1. **Cargar las variables de entorno de Oracle:**  

```
. oraenv
```
  2. **Iniciar el listener (proceso que escucha conexiones):**
-

```
lsnrctl start
```

### 3. Abrir SQL\*Plus:

```
sqlplus / as SYSDBA
```

### 4. Ejecutar comandos básicos (igual que en Windows).

---

## Conclusión

El acceso a Oracle Database puede hacerse de varias maneras:

- **SQL\*Plus** (línea de comandos).
- **SQL Developer** (interfaz gráfica).
- **DBeaver, TOAD, ORDS** (alternativas gráficas).
- **VSCode** con extensión de SQL Developer
- **Desde Linux**, con SQL\*Plus y comandos en terminal.

Dependiendo de la tarea que necesites realizar, puedes usar la línea de comandos para configuraciones avanzadas o SQL Developer para trabajar con datos de manera más visual.

---

## Punto 7. Creación y eliminación de bases de datos en Oracle

En Oracle, se pueden gestionar dos tipos principales de bases de datos:

- **CDB (Container Database)**: Base de datos contenedora que puede alojar múltiples PDBs.
- **PDB (Pluggable Database)**: Bases de datos "enchufables" dentro de una CDB.

Se pueden crear y eliminar estas bases de datos de dos maneras:

1. **Usando DBCA** (herramienta gráfica o en línea de comandos).
  2. **Mediante comandos SQL en SQL\*Plus.**
- 

### 7.1. Creación de bases de datos con DBCA

**DBCA (Database Configuration Assistant)** es una herramienta de Oracle que permite crear, modificar o eliminar bases de datos de forma guiada.

#### ➤ Creación de una CDB (Container Database) con DBCA

##### En Windows

Desde una terminal CMD con permisos de administrador, ejecutamos:

---

```
C:\Windows\system32> dbca -silent -createDatabase -templateName
General_Purpose.dbc -gdbName NOMCDB -sid NOMCDB -sysPassword 1234
-systemPassword 1234 -emConfiguration NONE -datafileDestination C:
\app\oracle\oradata -storageType FS -characterSet AL32UTF8
```

Explicación de los parámetros:

- **-silent**: Instalación sin interfaz gráfica.
- **-gdbName**: Nombre de la base de datos contenedora (CDB).
- **-sid**: Identificador del sistema de la base de datos.
- **-sysPassword** y **-systemPassword**: Contraseñas de los usuarios administrativos.
- **-datafileDestination**: Ubicación de los archivos de datos.
- **-characterSet AL32UTF8**: Codificación de caracteres utilizada.

### En Linux

```
$ dbca -silent -createDatabase -templateName General_Purpose.dbc -gdbName
COKE.rajasekhar.com -sid COKE -sysPassword Coke123$ -systemPassword Coke@123
-emConfiguration NONE -datafileDestination /u01/app/oracle/oradata -storageType
FS -characterSet AL32UTF8
```

---

## ➤ Creación de una PDB (Pluggable Database) con DBCA

Para añadir una nueva PDB dentro de una CDB existente, ejecutamos en Windows:

```
C:\Windows\system32> dbca
```

Desde la interfaz gráfica, seguimos las opciones para crear una **Pluggable Database (PDB)** dentro de la CDB existente.

---

## 7.2. Creación de bases de datos manualmente (SQL)

También es posible crear bases de datos usando comandos SQL en **SQL\*Plus**.

### ➤ Creación manual de una PDB (Pluggable Database)

Ejecutamos en SQL\*Plus:

```
SQL> CREATE PLUGGABLE DATABASE nom_pdb
ADMIN USER pdbadmin1 IDENTIFIED BY 1234
ROLES=(DBA)
DEFAULT TABLESPACE users;
```

- **nom\_pdb**: Nombre de la PDB.
- **ADMIN USER pdbadmin1 IDENTIFIED BY 1234**: Se crea un usuario administrador para la PDB.

- **DEFAULT TABLESPACE users:** Se define el tablespace por defecto.

**Para Linux,** la creación de una PDB es similar, pero con parámetros adicionales:

```
SQL> CREATE PLUGGABLE DATABASE DATAPDB  
ADMIN USER pdbadmin1 IDENTIFIED BY 1234 ROLES=(DBA)  
PATH_PREFIX = '/opt/oracle/oradata/ORCLCDB/DATAPDB/'  
FILE_NAME_CONVERT = ('/opt/oracle/oradata/ORCLCDB/pdbseed/',  
'/opt/oracle/oradata/ORCLCDB/DATAPDB/')  
DEFAULT TABLESPACE users DATAFILE '/opt/oracle/oradata/DATAPDB' AUTOEXTEND ON;
```

- **FILE\_NAME\_CONVERT:** Ubicación de los archivos físicos en disco.
  - **AUTOEXTEND ON:** Permite que los archivos de la base de datos crezcan automáticamente.
- 

### 7.3. Eliminación de bases de datos

La eliminación de bases de datos también se puede hacer de forma manual o con DBCA.

#### ➤ Eliminación de una PDB manualmente

Desde SQL\*Plus:

```
SQL> ALTER PLUGGABLE DATABASE pdb_manual CLOSE IMMEDIATE;  
SQL> DROP PLUGGABLE DATABASE pdb_manual INCLUDING DATAFILES;
```

- **CLOSE IMMEDIATE:** Cierra la PDB antes de eliminarla.
- **INCLUDING DATAFILES:** Elimina los archivos de datos asociados.

#### ➤ Eliminación de una CDB con DBCA

Desde un CMD en Windows:

```
C:\Windows\system32> dbca -silent -deleteDatabase -sourceDB nombbdd  
-sysDBAUserName sys -sysDBAPassword 1234
```

- **-sourceDB nombbdd:** Especifica el nombre de la base de datos a eliminar.
  - **-sysDBAUserName sys:** Usuario administrador de la base de datos.
- 

### Conclusión

- **DBCA** es la forma más sencilla y guiada para crear y eliminar bases de datos.
- **SQL\*Plus** permite un control más detallado sobre la creación y administración de PDBs y CDBs.
- **Eliminar bases de datos debe hacerse con precaución**, asegurando que no haya datos importantes antes de ejecutarlo.



## Punto 8. Conexión a Oracle Database

Para conectarte a una base de datos Oracle, necesitas tener acceso a la instancia de la base de datos y conocer los detalles de conexión, como el nombre del host, el puerto, el nombre del servicio o SID, y las credenciales de usuario (nombre de usuario y contraseña).

- Host: Es la dirección IP o el nombre del servidor donde está alojada la base de datos.
- Puerto: Por defecto, Oracle usa el puerto 1521, pero puede variar.
- Service Name/SID: El nombre del servicio o el SID de la base de datos.
- Usuario/Contraseña: Las credenciales de acceso a la base de datos.

### 8.1. Métodos de Conexión a Oracle Database

Existen varios métodos para conectarse a una base de datos Oracle, ya sea desde **línea de comandos (CLI)** o mediante **interfaces gráficas (GUI)**.

#### A) Conexión desde SQL\*Plus (CLI)

SQL\*PLUS es una herramienta de línea de comandos que permite ejecutar consultas SQL y administrar bases de datos Oracle.

Para acceder desde Windows:

#### Desde la misma máquina donde está el servicio ORACLE

1. Abrir el símbolo del sistema (CMD).
2. Ejecutar:

```
sqlplus / as SYSDBA
```

Esto conecta al usuario administrador SYS con rol SYSDBA a la base de datos.

3. Una vez dentro de SQL\*PLUS, se pueden ejecutar comandos como:

```
show user;                -- Muestra el usuario conectado
show con_name;
select name from v$database; -- Muestra el nombre de la CDB
show pdbs;                -- Lista las bases de datos (PDB)
```

4. Para conectarse a una PDB específica:

```
alter session set container=NOMPDB; -- Cambia a una PDB específica
show con_name; -- Verifica que la sesión está en la PDB correcta
```

## Conexión Remota a Oracle

Si quieres conectarte desde otra máquina a un servidor con Oracle Database, debes:

1. **Configurar el archivo** `tnsnames.ora` en el cliente.
2. **Habilitar el acceso remoto** en el servidor.
3. **Comprobar que el puerto 1521 está abierto** en el firewall.
4. Tener el **listener** instalado

Ejemplo de conexión remota en SQL\*Plus:

```
sqlplus system/1234@192.168.1.100:1521/NOMBDB
```

Aquí:

- `system/1234` → Usuario y contraseña.
- `192.168.1.100` → IP del servidor Oracle.
- `1521` → Puerto del Listener.
- `NOMBDB` → Nombre de la base de datos.

## Instalar listener.

1. Muy Importante, abrir un CMD con **permisos de administrador**.
2. Ejecutar `C:\> netca`
3. Agregar un listener
4. Configurar métodos de nomenclatura

**Muy importante:** Si no se instala con permisos de administrador, la instalación terminará bien, sin dar ningún aviso de error, pero mas tarde, cuando se intente conectar a través del listener no se podrá.

---

## B) Conexión desde SQL Developer (GUI)

SQL Developer es una interfaz gráfica de Oracle que permite conectarse y administrar bases de datos de forma más intuitiva.

Prerequisitos: Tener un **listener** instalado y configurado

Pasos para conectar:

1. Abrir **Oracle SQL Developer**.
2. Crear una **nueva conexión** y completar los siguientes datos:
  - Nombre de conexión: *Ejemplo: mi\_conexion*
  - Usuario: *Ejemplo: system*
  - Contraseña: *Ejemplo: 1234*
  - Hostname: *Ejemplo: localhost*
  - Puerto: *1521* (Por defecto en Oracle)

- SID / Service Name: *Ejemplo: NOMPDB*
3. Hacer clic en "**Probar conexión**" y si todo está bien, en "**Conectar**".
- 

### C) Conexión desde SQLcl (CLI)

SQLcl (SQL Command Line) es otra herramienta de línea de comandos más moderna que SQL\*Plus, con más funciones avanzadas.

Para conectarse desde **SQLcl**:

```
sqlcl / as SYSDBA
```

Permite ejecutar los mismos comandos que SQL\*Plus pero con características mejoradas como:

- Autocompletado de comandos.
  - Resaltado de sintaxis.
  - Soporte para scripts SQL avanzados.
- 

### D) Conexión desde DBeaver (GUI)

DBeaver es otra herramienta gráfica que permite conectar con diversas bases de datos como **Oracle**, **PostgreSQL**, **MySQL**, **MariaDB**, etc.

Pasos:

1. Abrir **DBeaver**.
  2. Crear una **nueva conexión** a Oracle.
  3. Introducir los datos del servidor (usuario, contraseña, host, puerto).
  4. Probar conexión y conectar.
- 

## 8.2. Arquitectura de Conexión a Oracle

Oracle Database usa un proceso llamado **Listener** para gestionar las conexiones.

Este proceso escucha en el **puerto 1521** y redirige las conexiones a la base de datos correspondiente.

Ejemplo:

- SQL Developer o DBeaver envían la petición al **listener**.
  - El **listener** autentica al usuario y redirige la conexión a una CDB o PDB.
-

### 8.3. Uso de ORDS (Oracle REST Data Services) para servicios web

ORDS es una herramienta que permite exponer datos de una base de datos Oracle como servicios web RESTful. Esto es útil para integrar la base de datos con aplicaciones web, móviles o cualquier sistema que pueda consumir APIs REST.

- Instalación y Configuración:
  - Descarga e instala ORDS en tu servidor.
  - Configura ORDS para que se conecte a tu base de datos Oracle.
  - Define los módulos y plantillas para los servicios REST que desees exponer.
- Creación de Servicios REST:
  - Puedes crear servicios REST para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en tus tablas.
  - ORDS permite definir endpoints REST que mapean a procedimientos PL/SQL o consultas SQL.
- Ejemplo de Uso:
  - Supongamos que tienes una tabla EMPLEADOS y quieres exponer un servicio REST para obtener la lista de empleados:

```
BEGIN
ORDS.ENABLE_SCHEMA(
  p_enabled => TRUE,
  p_schema => 'TU_ESQUEMA',
  p_url_mapping_type => 'BASE_PATH',
  p_url_mapping_pattern => 'empleados',
  p_auto_rest_auth => FALSE
);
END;
```

Luego, puedes acceder a este servicio a través de una URL como:

`http://tuservidor/ords/tu_esquema/empleados/`

- Seguridad: ORDS también permite configurar autenticación y autorización para proteger tus servicios REST.

## **Conclusión**

Este apartado explica cómo conectarse a Oracle Database desde diferentes herramientas y cómo funciona el proceso de conexión. Se recomienda practicar con SQL\*Plus y SQL Developer para familiarizarse con los comandos y la gestión de bases de datos.