

# 1. Arquitectura y estructura interna de un gestor de bases de datos.

## Descripción de los procesos de usuario

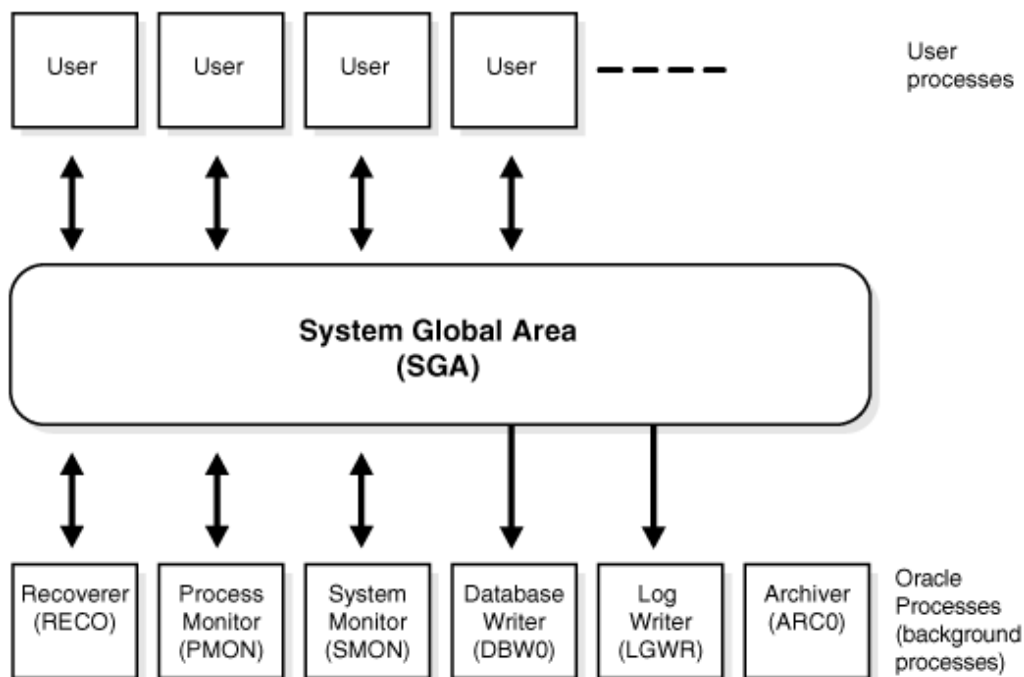
Cuando un usuario ejecuta un programa de aplicación (como un programa Pro \* C) o una herramienta de Oracle (como Oracle Enterprise Manager o SQL\*Plus), la Bases de datos Oracle crea un proceso de usuario para ejecutar la aplicación del usuario.

## Conexiones y Sesiones

Sesiones y conexiones y están estrechamente relacionadas con los procesos de usuario, pero son muy diferentes en significado.

**Conexión** es una vía de comunicación entre un usuario y un proceso de Bases de datos, ejemplo: Una vía de comunicación se establece mediante la comunicación entre los mecanismos disponibles (en un equipo que ejecutan ambos, el proceso de usuario y el de base de datos) o software de red (cuando se ejecute diferentes equipos, la aplicación y la base de datos, se comunican a través de una red).

**Sesión** es una conexión de un usuario a una base de datos a través de un proceso de usuario. Por ejemplo, cuando un usuario inicia SQL\*Plus, el usuario debe proporcionar un nombre de usuario válido y contraseña, luego es establecida una sesión para ese usuario. Una sesión dura desde el momento en que el usuario se conecta hasta el momento en que el usuario se desconecta o sale de la aplicación de base de datos.



## Descripción de los procesos de Oracle

Existen dos tipos de procesos que se ejecutan en Oracle, Los procesos de servidor y de

fondo.

### **Procesos de servidor**

Oracle crea procesos de servidor para manejar las peticiones de los usuarios conectados a los procesos de la instancia. En algunas ocasiones, cuando la aplicación y Oracle operan en el mismo equipo, es posible combinar el proceso de usuario y su correspondiente proceso de servidor en un único proceso para reducir los gastos generales del sistema. Sin embargo, cuando la aplicación y Oracle, operan de los distintos equipos, un proceso de usuario siempre se comunica con la base Oracle a través de otro proceso de servidor.

- Analiza y ejecutar comandos SQL emitido a través de la aplicación .
- Lee los bloques datos necesarios en los datafiles en disco, en los buffers compartidos de la SGA, si los bloques no están ya presentes en la SGA
- Mostrar los resultados de tal manera que la aplicación puede procesar la información

### **Procesos de fondo**

Para maximizar el rendimiento y tener muchos usuarios, un sistema Oracle multiproceso, utiliza los procesos de fondo

- Archiver Processes (ARCn)
- Checkpoint Process (CKPT)
- Database Writer Process (DBWn)
- Log Writer Process (LGWR)
- Process Monitor Process (PMON)
- Queue Monitor Processes (QMn)
- Recoverer Process (RECO)
- System Monitor Process (SMON)



cambiar el valor de este parámetro de forma dinámica para aumentar o disminuir el número ARCn processes.

### **Checkpoint Process (CKPT)**

Cuando se produce checkpoint control, Oracle debe actualizar la todas las cabeceras de los datafiles para registrar los detalles del checkpoint. Esto se hace por el proceso CKPT. El proceso CKPT, no escribir bloques en el disco; DBWn que siempre realiza el trabajo.

### **Database Writer Process (DBWn)**

Database Writer Process escribe el contenido de buffers en los datafiles. El DBWn es responsable de los procesos de escritura y modificado(sucio) del buffer caché al disco. Aunque un database writer process(DBW0) es adecuado para la mayoría de los sistemas, se puede configurar los procesos (DBW1 al DBW9 y DBWa al DBWj), para mejorar el rendimiento de escritura, si su sistema modifica altamenete los datos. Estos adicionales DBW n procesos no son útiles en sistemas monoprocesador.

### **Log Writer Process (LGWR)**

El Log Writer Process (LGWR) se encarga de la gestión-escritura de buffer redo log a un redo log file en el disco. LGWR escribe todas las entradas de redo que se han copiado en el buffer desde la última vez que escribió.

El redo log buffer es un buffer circular. Cuando LGWR escribe entradas de redo del registro de redo buffer a un archivo de registro de redo, el servidor de procesos puede copiar nuevas entradas más de las entradas en el redo log buffer que se han escrito en disco. LGWR escribe normalmente con la suficiente rapidez para garantizar que el espacio esté siempre disponible en el buffer de nuevas entradas, incluso cuando el acceso a la redo log es pesado.

### **Process Monitor Process (PMON)**

Lleva a cabo el proceso de recuperación cuando un proceso usuario falla. PMON se encarga de la limpieza del buffer cache de la base y liberar recursos para los procesos de usuario. Por ejemplo, se resetea la el estado de la tabla de operación activa, libera bloqueos y elimina los procesos de la lista de procesos activos.

PMON comprueba periódicamente la situación de control y los procesos del servidor y que se reinicie cualquier proceso que han dejado de publicarse (pero no todo lo que ha terminado intencionalmente). PMON también registra información acerca de la instancia y de los procesos de control con la red oyente.

### **Queue Monitor Processes (QMNn)**

Es un proceso de fondo opcional para Oracle Streams Avanzada Queuing, que supervisa las colas de mensajería. Se puede configurar hasta 10 procesos de la cola de monitor. Estos procesos son diferentes de otros procesos de fondo de Oracle, en caso de fracaso del proceso, no causa la caída de Oracle.

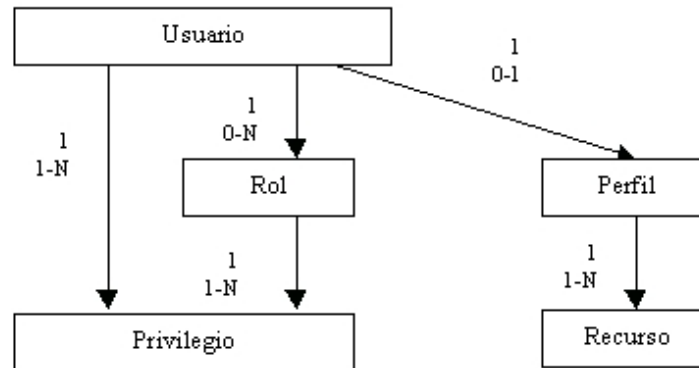
### **Recoverer Process (RECO)**

es un proceso de fondo que se utiliza con la configuración de la base de datos distribuida, resuelve automáticamente fallos de transacciones distribuidas

### **System Monitor Process (SMON)**

Lleva a cabo la recuperación, si es necesario, en el arranque de una instancia. SMON es también responsable de la limpieza de los segmentos temporales que ya no están en uso y de la fusión de los extens libres contiguos. Si alguna de las operaciones terminadas se han omitido durante la recuperación de la instancia a causa de archivos de errores de lectura o fuera de línea, SMON recupera el tablespace y se trae de nuevo en línea. SMON realiza controles periódicamente para ver si es necesario la recuperación. Otros procesos pueden llamar SMON si se detecta una necesidad.

## 2. Comando de control de acceso y roles para una base de datos.



### Conceptos de gestión de privilegios y recursos

Concepto	Significado
Privilegio	Permiso para realizar una acción, asignable a un usuario o un rol
Rol	Conjunto de privilegios, asignables a un usuario o un rol
Usuario	Colección de objetos y privilegios identificado con un nombre y password
Perfil	Conjunto de restricciones relativas al uso de recursos, y asignable a usuarios. Un usuario sólo puede tener un perfil
Recurso	Uso susceptible de ser restringido, asignable a un perfil

## Roles Predefinidos por Oracle

```
select * from dba_roles;
```

CONNECT

RESOURCE

DBA

EXP\_FULL\_DATABASE

IMP\_FULL\_DATABASE

DELETE\_CATALOG\_ROLE

EXECUTE\_CATALOG\_ROLE

SELECT\_CATALOG\_ROLE

## Recursos en Oracle

```
select * from user_resource_limits;
```

COMPOSITE\_LIMIT

SESSIONS\_PER\_USER

CPU\_PER\_SESSION

CPU\_PER\_CALL

LOGICAL\_READS\_PER\_SESSION

LOGICAL\_READS\_PER\_CALL

IDLE\_TIME

CONNECT\_TIME

PRIVATE\_SGA

## Limites en uso del espacio en disco

```
select * from dba_ts_quotas;
```

```
ALTER USER SCOTT QUOTA UNLIMITED ON USER_DATA;
```

```
ALTER USER SCOTT QUOTA 5M ON TEMPORARY_DATA;
```

```
ALTER USER SCOTT QUOTA 0 ON SYSTEM;
```

## Gestión de Privilegios

Para la gestión de:	Comando a utilizar
Privilegios	GRANT (conceder un privilegio a un usuario o a un rol) REVOKE (denegar un privilegio a un usuario o a un rol)
Roles	CREATE ROLE (crear) ALTER ROLE (modificar) DROP ROLE (borrar) SET ROLE (activar, desactivar) GRANT (conceder un permiso o un rol a un rol) REVOKE (denegar un permiso o un rol a un rol)
Usuarios	CREATE USER (crear) ALTER USER (modificar) DROP USER (borrar) GRANT (conceder un permiso o un rol a un usuario) REVOKE (denegar un permiso o un rol a un usuario)
Perfiles y Recursos	CREATE PROFILE (crear) ALTER PROFILE (modificar) DROP PROFILE (borrar) ALTER USER ... PROFILE (asignar a un usuario) CREATE USER ... PROFILE (asignar a un usuario)

## Activación de perfiles

Estado de la Base de Datos	Acción
Base de Datos Parada	RESOURCE_LIMIT = TRUE en c:\orant\database\initiorcl.ora
Base de Datos Arrancada	ALTER SYSTEM SET RESOURCE_LIMIT = TRUE;

## Vistas más relevantes

Significado	Nivel DBA	Nivel Usuario	Nivel Uusuario
<b>Usuarios</b>	dba_users	user_users	all_users
<b>Roles</b>	dba_roles		
<b>Roles asignados a roles o usuarios</b>	dba_role_privs	user_role_privs	
<b>Privilegios asignados a roles o usuarios</b>	dba_sys_privs		
<b>Permisos sobre tablas asignados a roles o usuarios</b>	dba_tab_privs		
<b>Roles asignados a roles</b>	role_role_privs		
<b>Privilegios de cada rol</b>	role_sys_privs		
<b>Límites de recursos</b>		user_resource_limits	
<b>Perfiles y sus límites de recursos asociados</b>	dba_profiles		
<b>Límites de recursos en cuanto a restricciones en claves</b>		user_password_limits	
<b>Límites de recursos en cuanto a espacio máximo en tablespaces</b>	dba_ts_quotas	user_ts_quotas	

## Ejemplos de gestión de privilegios y recursos

Crear un perfil que que sólo permita 2 conexiones concurrentes y asignar ese perfil al

## **usuario SCOTT**

```
SYSTEM>create profile solodos limit sessions_per_user 2;  
SYSTEM>alter user scott profile solodos;
```

## **Crear un rol llamado conectarse que incluya los roles connect y resource, y asignar ese rol a scott**

```
SYSTEM>create role conectarse;  
SYSTEM>grant connect, resource to conectarse;  
SYSTEM>grant conectarse to scott;
```

## **Convertir a scott en DBA**

```
SYSTEM>grant dba to scott;
```

## **Crear un usuario especificando que la password debe ser cambiada en la primera conexión**

```
SYSTEM>create user prueba1 identified by prueba1 password expire;
```

## **Cambiar el espacio reservado en disco para SCOTT**

```
SYSTEM>alter user scott quota unlimited on user_data;  
SYSTEM>alter user scott quota 5M on temporary_data;  
SYSTEM>alter user scott quota 0 on system;  
SYSTEM>select * from dba_ts_quotas;
```

## **Crear un usuario llamado PRUEBA1 y darle únicamente el permiso para conectarse (hacer log on)**

```
SYSTEM>create user prueba1 identified by prueba1;  
SYSTEM>grant create session to prueba1;
```

## **Dar al usuario PRUEBA1 permiso para crear tablas**

```
SYSTEM>grant create table to prueba1;  
SYSTEM>alter user prueba1 default tablespace system quota 10M on system;
```

```
SYSTEM>connect prueba1/prueba1
PRUEBA1>create table tabla1(campo1 number);
PRUEBA1>connect system/manager
```

### **Desde SCOTT, dar al usuario PRUEBA1 el permiso de consultar la tabla EMP**

```
SYSTEM>connect scott/tiger
SCOTT>grant select on emp to prueba1;
PRUEBA1>select * from scott.emp;
PRUEBA1>connect system/manager
```

### **Desde SCOTT, dar al usuario PRUEBA1 el permiso de insertar en la tabla EMP**

```
SYSTEM>connect scott/tiger
SCOTT>grant insert on emp to prueba1;
PRUEBA1>insert into scott.emp values ('john',999,'development');
PRUEBA1>connect system/manager
```

### **Desde SYSTEM, dar al usuario PRUEBA1 el permiso de consultar la tabla EMP de SCOTT**

```
SYSTEM>connect scott/tiger
SCOTT>grant select on emp to system with grant option;
PRUEBA1>connect system/manager
SYSTEM>grant select on scott.emp to prueba1;
SYSTEM>connect prueba1/prueba1
PRUEBA1>select * from scott.emp;
PRUEBA1>connect system/manager
```

### **Desde SYSTEM, dar al usuario PRUEBA1 el permiso de crear otros usuarios**

```
SYSTEM>grant create user to prueba1;
SYSTEM>connect prueba1/prueba1
PRUEBA1>create user prueba2 identified by prueba2;
drop user prueba2;
```

PRUEBA1>connect system/manager

**Desde SYSTEM, dar al usuario PRUEBA1 el permiso de crear otros usuarios (por ejemplo, PRUEBA2) de forma que estos, a su vez, puedan crear otros usuarios (por ejemplo, PRUEBA3)**

```
SYSTEM>grant create user to prueba1 with admin option;
SYSTEM>grant create session to prueba1 with admin option;
SYSTEM>connect prueba1/prueba1
PRUEBA1>create user prueba2 identified by prueba2;
PRUEBA1>grant create user to prueba2 with admin option;
PRUEBA1>grant create session to prueba2 with admin option;
PRUEBA1>connect prueba2/prueba2
PRUEBA2>create user prueba3 identified by prueba3;
PRUEBA2>grant create session to prueba3;
PRUEBA2>connect prueba3/prueba3
PRUEBA3>connect system/manager.
```

## objetos

### Privilegios sobre

ALTER

EXECUTE

INDEX

INSERT

READ

REFERENCES

SELECT

UPDATE

ALL ó ALL PRIVILEGES

## Sistema

### Privilegios del sistema

ALTER ANY CLUSTER

ALTER ANY INDEX

ALTER ANY PROCEDURE

ALTER ANY ROLE

ALTER ANY SEQUENCE

ALTER ANY SNAPSHOT

ALTER ANY TABLE

ALTER ANY TYPE

ALTER ANY TRIGGER

ALTER DATABASE

ALTER PROFILE

ALTER RESOURCE COST

ALTER ROLLBACK SEGMENT

ALTER SESSION

ALTER SYSTEM

ALTER TABLESPACE

ALTER USER

ANALYZE ANY

AUDIT ANY

AUDIT SYSTEM

BACKUP ANY TABLE

BECOME USER

COMMENT ANY TABLE

CREATE ANY CLUSTER

CREATE ANY DIRECTORY

CREATE ANY INDEX

CREATE ANY LIBRARY

CREATE ANY PROCEDURE

CREATE ANY SEQUENCE

CREATE ANY SNAPSHOT

CREATE ANY SYNONYM

CREATE ANY TABLE

CREATE ANY TRIGGER

CREATE ANY TYPE

CREATE ANY VIEW

CREATE CLUSTER

CREATE DATABASE LINK

CREATE ANY LIBRARY

CREATE PROCEDURE

CREATE PROFILE

CREATE PUBLIC DATABASE LINK

CREATE PUBLIC SYNONYM

CREATE ROLE

CREATE ROLLBACK SEGMENT

CREATE SEQUENCE

CREATE SESSION

CREATE SNAPSHOT

CREATE SYNONYM

CREATE TABLE

CREATE TABLESPACE

CREATE TRIGGER

CREATE TYPE

CREATE USER

CREATE VIEW

DELETE ANY TABLE

DROP ANY CLUSTER

DROP ANY DIRECTORY

DROP ANY INDEX

DROP ANY LIBRARY

DROP ANY PROCEDURE

DROP ANY ROLE

DROP ANY SEQUENCE

DROP ANY SNAPSHOT

DROP ANY SYNONYM

DROP ANY TABLE

DROP ANY TRIGGER

DROP ANY TYPE

DROP ANY VIEW

DROP LIBRARY

DROP PROFILE

DROP PUBLIC DATABASE LINK

DROP PUBLIC SYNONYM

DROP ROLLBACK SEGMENT

DROP TABLESPACE

DROP USER

EXECUTE ANY PROCEDURE

EXECUTE ANY TYPE

FORCE ANY TRANSACTION

FORCE TRANSACTION

GRANT ANY PRIVILEGE

GRANT ANY ROLE

INSERT ANY TABLE

LOCK ANY TABLE

MANAGE TABLESPACE

RESTRICTED SESSION

SELECT ANY SEQUENCE

SELECT ANY TABLE

SYSDBA

SYSOPER

UNLIMITED TABLESPACE

UPDATE ANY TABLE

### 3. Estructuras de almacenamiento

Almacenamiento	Descripción
Archivos de Control	<p>Contiene información de control y seguimiento imprescindible para el funcionamiento de la base de datos.</p> <p>Es recomendable tener varias copias del fichero de control para poder arrancar la base de datos si una de ellas se estropea, y durante la creación se crean tres ficheros.</p> <p>Es una estructura lógica que agrupa uno más ficheros de datos o ficheros temporales (tempfiles) y que permiten almacenar los objetos creados por los usuarios.</p>
Tablespaces	<p>Hay varios tipos de tablespaces:</p> <p>Undo tablespaces. Se utilizan para gestionar poder deshacer las transacciones incompletas. En versiones anteriores se utilizaban los segmentos de Rollback para realizar esta tarea.</p> <p>Temporary tablespaces. Permiten entre otras tareas mejorar el tiempo de respuesta en las ordenaciones</p> <p>Permanent tablespaces. Este tipo de tablespaces serán asignados a uno o varios usuarios y contendrán los objetos de su esquema (tablas, procedimientos, índices, ...)</p>

4. Identifica los estados de inicio y cierre una base de datos
5. Identifica las etapas para instalación del motor de base de datos
6. Identifica las etapas para la creación de una base de datos manual y automática
7. Describe el uso de los: datafiles, tablespaces, redo log, segmentos, extensiones, bloques y segmentos de rollback
8. Configura un servidor de bases de datos restringiendo el acceso sólo a las aplicaciones clientes.
9. Respalda los datos de la base de datos.
10. Restaura los datos de base de datos.
11. Recupera una falla en un sistema de bases de datos, en una situación simulada o extraída
12. de un caso real.
13. Transporta datos entre bases de datos.

