

## Cuestionario de Entrevista - Oracle DBA Semi Sr

### INSTALACIÓN Y CONFIGURACIÓN:

- ¿Cuál es la diferencia entre una base de datos y una instancia en Oracle?

A grandes rasgos la base de datos son los archivos físicos de los datos guardados en disco como lo son los archivos de control, logs y data de la misma, mientras que la instancia es toda la lógica de monitorio de sistema, procesos, parámetros de configuración entre mas como los manejos de memoria como seria la ejecución de tareas específicas.

- ¿Qué pasos seguis para instalar Oracle Database en un servidor Linux o Windows?

Linux

1.- Verificar los requisitos mínimos de sistema en el equipo requeridos por Oracle tanto como de hardware, como la configuración del hostname del mismo, la configuración del firewall y antivirus, por último, verificar que tenemos los privilegios de administrador

2.- Descargar u obtener la versión necesaria de Oracle

3.- a) Crear usuario para oracle (grupos, DBA)

```
groupadd oinstall  
groupadd dba  
useradd -g oinstall -G dba oracle  
passwd oracle
```

b) Creación de carpetas de instalación

```
mkdir -p /u01/app/oracle/product/19.0.0/dbhome_1  
chown -R oracle:oinstall /u01  
chmod -R 775 /u01
```

4.- Configurar ambiente como variables de entorno

```
export ORACLE_BASE=/u01/app/oracle  
export ORACLE_HOME=$ORACLE_BASE/product/19.0.0/dbhome_1  
export PATH=$PATH:$ORACLE_HOME/bin  
export ORACLE_SID=orcl
```

5.- Ejecutar el instalador (estar como Oracle y no como root)

```
./runInstaller
```

6.- a) Elegir el tipo de instalación que nos ofrece la GUI

b) Definir la ubicación de la variable de entorno ORACLE\_HOME

c) Definir las contraseñas para SYS y SYSTEM

7.- Ejecutar los scripts extras solicitados por Oracle para Linux

```
/u01/app/oralInventory/orainstRoot.sh
```

```
/u01/app/oracle/product/19.0.0/dbhome_1/root.sh
```

8.- Probar la conexión con SQLPLUS

```
sqlplus / as sysdba
```

9.- Validar los servicios

```
lsnrctl status,ps -ef | grep pmon
```

- ¿Qué herramientas usás para crear una base de datos: DBCA o manualmente? ¿Por qué?

El DBCA ayuda mucho para la creación de una DB con ambiente estándar dantote rapidez y reduciendo los errores, ideal para utilizar con DOKER donde por lo regular se utilizan ambientes minimalistas aunque la instalación manual te da mas flexibilidad para el control o personalización de esta para levantamiento de varios o el total de sus procesos, recomendable para ambientes robustos o en crecimiento paulatino.

- ¿Qué parámetros iniciales configuras al crear una base de datos?

Pueden ser varios dependientes de lo requerido por el usuario o el modularidad del sistema, auditoria, recuperación, etc pero los comunes son:

- DB\_NAME Nombre
- DB\_UNIQUE\_NAME
- MEMORY\_TARGET
- SGA\_TARGET
- PGA\_AGGREGATE\_TARGET
- PROCESSES
- SESSIONS
- UNDO\_TABLESPACE
- UNDO\_MANAGEMENT
- COMPATIBLE
- CHARACTER\_SET / NLS\_\*
- DB\_BLOCK\_SIZE
- OPEN\_CURSORS
- LOG\_ARCHIVE\_DEST / LOG\_ARCHIVE\_FORMAT
- AUDIT\_TRAIL
- REMOTE\_LOGIN\_PASSWORDFILE
- DB\_RECOVERY\_FILE\_DEST / DB\_RECOVERY\_FILE\_DEST\_SIZE
- DIAGNOSTIC\_DEST

Un ejemplo sencillo seria

```
DB_NAME=orcl
```

```
DB_UNIQUE_NAME=orcldb
CONTROL_FILES=/u01/app/oracle/oradata/orcl/control01.ctl
MEMORY_TARGET=1G
UNDO_MANAGEMENT=AUTO
UNDO_TABLESPACE=UNDOTBS1
COMPATIBLE='19.0.0'
CHARACTER_SET=AL32UTF8
OPEN_CURSORS=300
PROCESSES=150
```

#### **USUARIOS Y SEGURIDAD:**

##### **- ¿Cómo creás un usuario y asignás permisos en Oracle?**

En el ejemplo para crear un usuario del tipo “Dev” mediante scrip haría esto :

a) Crear el usuario

```
CREATE USER dev_user IDENTIFIED BY DevUser123!
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp
QUOTA UNLIMITED ON users;
```

b) Dar privilegios para conectarse y crear objetos

```
GRANT CREATE SESSION TO dev_user;
GRANT CREATE TABLE TO dev_user;
GRANT CREATE VIEW TO dev_user;
GRANT CREATE SEQUENCE TO dev_user;
GRANT CREATE PROCEDURE TO dev_user;
GRANT CREATE TRIGGER TO dev_user;
GRANT CREATE TYPE TO dev_user;
GRANT CREATE SYNONYM TO dev_user;
```

d) Asignar roles opcionales

```
GRANT CONNECT TO dev_user;
```

```
GRANT RESOURCE TO dev_user;
```

E) (Opcional) Dar acceso ilimitado al espacio

```
GRANT UNLIMITED TABLESPACE TO dev_user;
```

- ¿Qué diferencias hay entre roles y privilegios de sistema?

Los roles agrupan los privilegios de sistema y estos a sus vez son permisos individuales que permiten hacer o no una acción por ejemplo la acción CRUD, los privilegios suelen granulares para otorgar capacidades específicas.

- ¿Cómo implementás políticas de contraseñas seguras (profiles)?

Lo haría controlando aspectos como caducidad de contraseña, longitud mínimo de caracteres y manejo de caracteres específicos, reintentos fallidos de conexión, tiempo de inactividad por ejemplo :

a)

```
CREATE PROFILE perfil_seguro LIMIT  
  FAILED_LOGIN_ATTEMPTS 5          -- Máximo de intentos fallidos  
  PASSWORD_LOCK_TIME 1/24         -- Tiempo de bloqueo (1 hora)  
  PASSWORD_LIFE_TIME 90           -- Cambiar cada 90 días  
  PASSWORD_REUSE_TIME 365         -- No reutilizar en un año  
  PASSWORD_REUSE_MAX 10          -- No reutilizar últimas 10 contraseñas  
  PASSWORD_GRACE_TIME 7          -- Días de gracia antes del vencimiento  
  PASSWORD_VERIFY_FUNCTION verify_function;
```

b) asignar PROFILE

```
ALTER USER juan PROFILE perfil_seguro;
```

c) Verificar la asignación

```
SELECT username, profile FROM dba_users WHERE username = 'JUAN';
```

d) ver parámetros del PROFILE

```
SELECT * FROM dba_profiles WHERE profile = 'PERFIL_SEGURO';
```

e) Se puede crear una función para verificar la complejidad de la contraseña

```

CREATE OR REPLACE FUNCTION mi_verificador (
    username VARCHAR2,
    password VARCHAR2,
    old_password VARCHAR2
) RETURN BOOLEAN IS
BEGIN
    IF LENGTH(password)< 10 THEN
        RAISE_APPLICATION_ERROR(-20001, 'La contraseña debe tener al menos 10 caracteres');
    END IF;
    RETURN TRUE;
END;

```

f) ALTER PROFILE perfil\_seguro LIMIT PASSWORD\_VERIFY\_FUNCTION mi\_verificador;

#### - ¿Cómo manejás accesos indebidos o usuarios bloqueados?

Las buenas prácticas dicen que debemos de utilizar perfiles con intentos de conexión limitados con bloqueo temporal y auditar estos eventos como lo son login fallidos o cambios de perfil y desactivar todos los accesos remotos.

1.- Primero es detectar los accesos indebidos

a) Ver usuarios bloqueados

```

SELECT username, account_status
FROM dba_users
WHERE account_status LIKE '%LOCKED%';

```

b) Revisar intentos fallidos

```

SELECT * FROM dba_audit_trail
WHERE returncode = 1017 -- código de error: usuario/contraseña inválida
ORDER BY timestamp DESC;

```

2.- Desbloquear el usuario si es legítimo

```
ALTER USER juan ACCOUNT UNLOCK;
```

3.- Restablecer contraseña si se necesita

```
ALTER USER juan IDENTIFIED BY NuevaClaveSegura123;
```

#### 4.-Prevencion para futuros bloqueos

a) ver cuantos intentos fallidos se permiten y los tiempos de bloqueo

```
SELECT * FROM dba_profiles  
WHERE resource_name IN ('FAILED_LOGIN_ATTEMPTS', 'PASSWORD_LOCK_TIME');
```

b) Modificar perfil si es necesario

```
ALTER PROFILE perfil_seguro LIMIT FAILED_LOGIN_ATTEMPTS 10;  
ALTER PROFILE perfil_seguro LIMIT PASSWORD_LOCK_TIME 0.0416;
```

c) Asilar el problema si es sospechoso mediante

```
AUDIT SESSION BY juan;
```

#### 5.- Suspender o eliminar al usuario indebido confirmado

a) Bloquear usuario de forma manual

```
ALTER USER hacker ACCOUNT LOCK;
```

b) O eliminar si no debe existir

```
DROP USER hacker CASCADE;
```

Se debe de manejar este tipo de accesos investigando el origen de este mismo, asegurándonos la legitimidad y ajustes del perfil del usuario, así como la auditoría del incidente, si se confirma como malicioso el este, se debe de bloquear o eliminar ya sea el caso.

#### **BACKUP Y RECUPERACIÓN (RMAN):**

**- ¿Qué es RMAN y por qué se prefiere sobre métodos manuales?**

Es la herramienta oficial de Oracle Recovery Manager la cual se utiliza para el respaldo y recuperación de datos, se prefiere sobre otro tipo de métodos (manuales) porque ofrece recuperación granular por ejemplo por tiempo, tablespace, incluso bloqueos corruptos, garantiza backups consistentes, se puede automatizar con scripts, Jobs, etc, tiene una recuperación precisa mediante Point-in-time, contiene registros histórico del backup.

Un ejemplo sencillo podría ser

```
rman target /  
# Backup completo  
BACKUP DATABASE PLUS ARCHIVELOG;
```

```
# Validar backup  
VALIDATE BACKUPSET;  
  
# Restaurar y recuperar  
RESTORE DATABASE;  
RECOVER DATABASE;
```

- ¿Cuál es la diferencia entre backup completo y backup incremental en RMAN?

Un backup completo copia todo, ideal como base de la cual partir

Un backup incremental es más rápido y ahorra espacio, haciendo respaldo de solo los cambios realizados, a diferencia del primero este es dependiente de la una base de la cual partir por lo cual podemos decir que para tener un incremental debemos tener primero un completo o base, siendo que este segundo será más rápido permitiéndonos optimizar tiempos.

Ejemplo:

1.- Completo

```
BACKUP DATABASE;
```

a) Nivel 0 (base)

```
BACKUP INCREMENTAL LEVEL 0 DATABASE;
```

2.- Incremental

b) Nivel 1 diferencial (más usado en backups diarios)

```
BACKUP INCREMENTAL LEVEL 1 DATABASE;
```

c) Nivel 1 acumulativo

```
BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;
```

- ¿Cómo restaurás una base de datos desde un backup RMAN?

1.- En un escenario de corrupción de DB

a) Arrancar RMAN

```
rman target /
```

b) si usas catalogo

```
rman target / catalog rman_user@catdb
```

2.- Montar la DB si se perdió , solo archivos de control

a) si se perdió , solo archivos de control

```
STARTUP MOUNT;
```

b) si se perdió el archivo de control (primero se restaura)

```
RESTORE CONTROLFILE FROM AUTOBACKUP;
```

```
ALTER DATABASE MOUNT;
```

3.- Restaurar los datafiles

```
RESTORE DATABASE;
```

4.- Recuperar la DB

```
RECOVER DATABASE;
```

5.- Abrir la DB

```
ALTER DATABASE OPEN;
```

Si hubo perdida parcial

```
ALTER DATABASE OPEN RESETLOGS;
```

Validar estado de la DB

```
LIST BACKUP;
```

```
REPORT SCHEMA;
```

En script podría quedar así:

```
rman target /
```

-- Si solo perdiste datafiles

```
STARTUP MOUNT;
```

```
RESTORE DATABASE;
```

```
RECOVER DATABASE;
```

```
ALTER DATABASE OPEN;
```

-- Si perdiste control file

```
STARTUP NOMOUNT;
```

```
RESTORE CONTROLFILE FROM AUTOBACKUP;
```

```
ALTER DATABASE MOUNT;
```

```
RESTORE DATABASE;
```

```
RECOVER DATABASE;
```

```
ALTER DATABASE OPEN RESETLOGS;
```

Con un Point-in-time Recovery:

```
rman target /  
SHUTDOWN IMMEDIATE;  
STARTUP MOUNT;  
RESTORE DATABASE;  
RECOVER DATABASE UNTIL TIME "TO_DATE('2025-07-10 23:45:00', 'YYYY-MM-DD  
HH24:MI:SS')";  
ALTER DATABASE OPEN RESETLOGS;
```

- **¿Has usado RMAN en modo ARCHIVELOG? ¿Qué significa esto?**

Si para la restauración de la DB en una fecha en específico o para recuperar ciertas tablas corruptas por procesos incompletos debidos a fallos físicos, de red, de electricidad misma, etc.

Es un modo de operación de Oracle en donde los redo logs se archivan automáticamente después de llenarse esto permite recuperar la base de datos desde un COMMIT, hacer backups sin reiniciar la DB o hacer la recuperación desde un point-in-time

Esto lo podríamos hacer de la siguiente forma, primero activando ARCHIVELOG:

1.- Poner la base en modo Mount

```
SHUTDOWN IMMEDIATE;  
STARTUP MOUNT;
```

2.- Activar el ARCHIVELOG

```
ALTER DATABASE ARCHIVELOG;
```

3.- Abrir la DB

```
ALTER DATABASE OPEN;
```

4.- Validar

```
ARCHIVE LOG LIST;
```

Hacer un backup diario:

```
BACKUP AS COMPRESSED BACKUPSET  
DATABASE PLUS ARCHIVELOG  
FORMAT '/u01/backups/diario_%d_%T_%U.bkp';  
DELETE ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-1';
```

```
CROSSCHECK BACKUP;  
DELETE NOPROMPT OBSOLETE;
```

Se puede agregar un CONFIGURE persistente en RMAN

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;  
CONFIGURE BACKUP OPTIMIZATION ON;
```

#### MONITOREO Y PERFORMANCE TUNING:

##### - ¿Cómo identificás y resolvés problemas de performance en una base de datos?

Se pueden utilizar varios tiempos de herramientas como Vistas Dinamicas o ASH (Active Session History) o el OEM ( Oracle Enterprise Manager).

Lo más común es detectar cuellos de botella en el área de CPU, I/O, SQL, Red para esto podemos implementar en SQL el agregado índices, reescribir consulta, revisar los joins y filtros, también podemos hacer ajustes de configuración al revisar la cache, o ajustes los objetos como la partición de tablas, rediseñar tablespaces, revisar el uso de redo o temp.

Para la detección podemos usar :

###### 1.- Costos en tiempo total de ejecución

```
SELECT *  
FROM (  
    SELECT sql_id, executions, elapsed_time / 1000000 AS total_secs, sql_text  
    FROM v$sql  
    WHERE executions > 0 ORDER BY elapsed_time DESC  
)WHERE ROWNUM <= 10;
```

###### 2.- Sesiones activas que están consumiendo CPU o están esperando algo

```
SELECT sid, serial#, username, status, event, wait_class, seconds_in_wait  
FROM v$session  
WHERE status = 'ACTIVE' AND username IS NOT NULL;
```

##### - ¿Qué vistas del sistema (V\$) usás para analizar el rendimiento?

Uso una combinación de vistas V\$ como V\$SESSION, V\$SQL, V\$SYSTEM\_EVENT, V\$ACTIVE\_SESSION\_HISTORY, y V\$RESOURCE\_LIMIT para detectar cuellos de botella, identificar SQL lentos, y ajustar rendimiento según los síntomas.

###### a) Sesiones activas que están haciendo algo

```
SELECT sid, username, status, wait_class, event, seconds_in_wait
```

```
FROM v$session  
WHERE status = 'ACTIVE' AND username IS NOT NULL;
```

b) Consultas más lentas

```
SELECT *  
FROM (  
    SELECT sql_id, executions, elapsed_time/1e6 AS total_seconds, sql_text  
    FROM v$sql  
    WHERE executions > 0 ORDER BY elapsed_time DESC )  
    WHERE ROWNUM <= 5;
```

c) Eventos de espera más frecuentes

```
SELECT event, total_waits, time_waited/100 AS seconds_waited  
FROM v$system_event  
WHERE event NOT LIKE 'SQL%' ORDER BY time_waited DESC FETCH FIRST 5 ROWS ONLY;
```

d) Uso de recursos

```
SELECT resource_name, current_utilization, max_utilization, limit_value  
FROM v$resource_limit  
WHERE resource_name IN ('sessions', 'processes');
```

e) Segmentos con más actividad

```
SELECT owner, object_name, statistic_name, value  
FROM v$segment_statistics  
WHERE value > 10000 ORDER BY value DESC FETCH FIRST 5 ROWS ONLY;
```

- ¿Qué es AWR y cómo lo utilizás?

Es un componente de Oracle de recopilación de análisis de rendimiento de la DB que actúa de manera automática cada unidad de tiempo, hace guardado de Datos de sesión, undo , locks, querys costosas, estadísticas de objetos como tablas, índices etc, se pueden generar reportes de rendimiento que lo podemos necesitar para hacer un baseline de rendimiento, o tras hacer un cambios de índices o hacer una comparación de una snapshots

```
SELECT snap_id, begin_interval_time, end_interval_time
```

```
FROM dba_hist_snapshot  
ORDER BY snap_id DESC  
FETCH FIRST 5 ROWS ONLY;
```

- **¿Qué técnicas usás para optimizar consultas lentas?**

1.- Analizo el plan de ejecución

```
EXPLAIN PLAN FOR <tu_consulta>;  
SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);
```

2.- Reviso uno de índices y estadísticas

```
EXEC DBMS_STATS.GATHER_TABLE_STATS('ESQUEMA', 'TABLA');
```

3.- Reescribo la consulta

4.- Aplico técnicas de tuning según el problema puede ser partición de tablas ( por algún campo fecha), reescribir consulta si algún campo lo afecta alguna función, se pueden cambiar un where implícito por un join explícito

Por ejemplo

```
SELECT * FROM pedidos WHERE TO_CHAR(fecha_pedido, 'YYYY-MM') = '2025-07';
```

Evito función en columna → impide uso de índice

```
SELECT * FROM pedidos  
WHERE fecha_pedido BETWEEN TO_DATE('2025-07-01', 'YYYY-MM-DD')  
AND TO_DATE('2025-07-31', 'YYYY-MM-DD');
```

**SQL Y PL/SQL:**

- **¿Qué diferencia hay entre una función, un procedimiento y un paquete en PL/SQL?**

A grandes rasgos una Función si o si regresa un valor, el procedimiento puede o no devolverlo y el paquete es el encapsulado que agrupa funciones y procedimientos.

Ejemplo FUNCTION:

```
CREATE OR REPLACE FUNCTION obtener_descuento(p_cliente_id NUMBER)  
RETURN NUMBER IS  
BEGIN
```

```
    RETURN 0.10;  
  
END;
```

Ejemplo PROCEDURE:

```
CREATE OR REPLACE PROCEDURE aplicar_descuento(p_cliente_id NUMBER) IS  
  
BEGIN  
  
    UPDATE clientes SET descuento = 0.10 WHERE cliente_id = p_cliente_id;  
  
END;
```

Ejemplo PACKAGE:

```
CREATE OR REPLACE PACKAGE operaciones_clientes IS  
  
    PROCEDURE alta_cliente(...);  
  
    FUNCTION calcular_descuento(...);  
  
END;
```

- **¿Qué es un trigger y en qué casos lo usarías?**

Es un bloque de código que se ejecuta automáticamente después de un evento detonante, se podría aplicar para cuando se hace un borrado físico de un objeto se registra en una tabla bitácora, se utiliza mucho para tablas de auditoria de reglas de negocio

Ejemplo:

```
CREATE OR REPLACE TRIGGER trg_auditar_pedidos  
  
AFTER INSERT OR DELETE ON pedidos  
  
FOR EACH ROW  
  
BEGIN  
  
    INSERT INTO log_pedidos (usuario, accion, fecha)  
  
    VALUES (USER, 'CAMBIO EN PEDIDOS', SYSDATE);  
  
END;
```

- **¿Cómo debugueás un procedimiento que da error?**

Lo más común que utilizo es DBMS\_OUTPUT.PUT\_LINE capturando una exception, también depende la herramienta también puedo utilizar puntos de interrupción como en cualquier analizador de código o utilizar variables que las pueda leer en tiempo real.

```
BEGIN  
  
-- Código
```

```
EXCEPTION  
WHEN OTHERS THEN  
DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);  
END;
```

**- ¿Qué es un cursor y cómo lo manejás para evitar problemas de rendimiento?**

Un cursor es un mecanismo que te permite recorrer fila por fila un resultado de una consulta.

Cuando necesitas procesar cada fila con lógica compleja (no se puede hacer con UPDATE SET directo).

En procedimientos que afectan múltiples registros con condiciones particulares.

Ejemplo:

```
DECLARE  
CURSOR c_clientes IS SELECT cliente_id FROM clientes;  
BEGIN  
FOR reg IN c_clientes LOOP  
-- Hacer algo por cliente  
DBMS_OUTPUT.PUT_LINE('Cliente: ' || reg.cliente_id);  
END LOOP;  
END;
```

**ALTA DISPONIBILIDAD Y RECUPERACIÓN:**

**- ¿Qué es Oracle Data Guard y para qué se usa?**

Oracle Data Guard es una solución de Oracle que permite tener una o más copias en espera (standby) de una base de datos en tiempo real o casi real, listas para usarse en caso de falla de la base primaria.

**1. Recuperación ante desastres (DR)**

Si el servidor principal falla, se hace failover al standby, que se convierte en la nueva primaria.

**2. Alta disponibilidad**

Se puede hacer un switchover para mantenimiento sin downtime.

**3. Consultas de solo lectura**

En modo Active Data Guard, puedes ejecutar consultas en el standby sin afectar al primary.

**4. Pruebas sin riesgos**

Puedes abrir el standby en modo snapshot para hacer testing.

**- ¿Cuál es la diferencia entre un standby físico y uno lógico?**

El standby físico aplica bloques de redo, es más confiable para failover. El lógico transforma esos cambios en SQL, lo que permite mayor flexibilidad para reporting o migraciones, pero es más propenso a errores si hay cambios estructurales en la base original.

**- ¿Tenés experiencia con Oracle RAC? ¿En qué contextos lo implementaste?**

Actualmente tengo poca experiencia práctica con Oracle RAC, pero sí tengo una base teórica clara de cómo funciona y para qué se utiliza. Entiendo que RAC permite tener varias instancias accediendo a la misma base de datos para lograr alta disponibilidad y escalabilidad horizontal. Aún no he tenido la oportunidad de implementarlo en producción, pero estoy preparado para participar en su administración, monitoreo o configuración, y me interesa mucho seguir aprendiendo sobre RAC en entornos reales.

**TROUBLESHOOTING:**

**- ¿Qué hacés si un usuario dice que "la base está lenta"?**

Sigo un enfoque sistemático ya que “Lento” no me dice nada de donde viene el problema , ya que si me dice que a cierta hora del día se nota la lentitud o que cuando un módulo o app se ejecuta me dice más información.

1.- Reviso sesiones activas y que están haciendo

```
SELECT sid, serial#, username, status, event, wait_class, seconds_in_wait
FROM v$session
WHERE status = 'ACTIVE' AND username IS NOT NULL;
```

2.- Detectar SQL lentos

```
SELECT sql_id, elapsed_time/1e6 AS segundos, sql_text
FROM v$sql
WHERE elapsed_time > 1000000
ORDER BY elapsed_time DESC  FETCH FIRST 5 ROWS ONLY;
```

3.- Detectar eventos en espera

```
SELECT event, time_waited/100 AS total_segundos
FROM v$system_event
ORDER BY time_waited DESC  FETCH FIRST 5 ROWS ONLY;
```

4.- Consultar AWR

```
@?/rdbms/admin/awrrpt.sql
```

## 5.- ver lock o bloqueos

```
SELECT * FROM v$lock WHERE block = 1;
```

### - ¿Cómo detectás y resolvés un bloqueo (deadlock)?

Un deadlock ocurre cuando dos o más sesiones están esperando recursos bloqueados entre sí. Ninguna puede continuar. Oracle detecta automáticamente el deadlock y mata una de las sesiones involucradas para liberar el recurso.

Para detectarlo:

a) Alert log busco errores como

```
ORA-00060: deadlock detected while waiting for resource
```

b) Vista v\$session y v\$locked\_object

```
SELECT s.sid, s.serial#, s.username, l.object_id, o.object_name  
FROM v$session s  
JOIN v$locked_object l ON s.sid = l.session_id  
JOIN dba_objects o ON l.object_id = o.object_id;
```

c) Ver bloqueadores

```
SELECT blocking_session, sid, serial#, wait_class, event  
FROM v$session  
WHERE blocking_session IS NOT NULL;
```

En resumen

Ver qué objetos están bloqueados

Contactar al usuario/sesión involucrada (si es app, avisar al dev).

En caso extremo:

```
ALTER SYSTEM KILL SESSION 'sid,serial#' IMMEDIATE;
```

### - ¿Qué comandos o vistas usás para ver sesiones activas o SQLs en ejecución?

a) ver sesiones activas

```
SELECT sid, serial#, username, status, event, wait_class  
FROM v$session  
WHERE status = 'ACTIVE' AND username IS NOT NULL;
```

b) ver sql actual

```
SELECT s.sid, s.username, q.sql_text  
FROM v$session s  
JOIN v$sql q ON s.sql_id = q.sql_id  
WHERE s.status = 'ACTIVE' AND s.username IS NOT NULL;
```

c) ver secciones que están bloqueando a otras

```
SELECT sid, blocking_session, event  
FROM v$session  
WHERE blocking_session IS NOT NULL;
```

- ¿Qué hacés si se llena un tablespace crítico?

Si un tablespace se llena, primero reviso si tiene autoextend habilitado. Si no, agrego espacio con un nuevo datafile o habilitando autoextend. También evalúo liberar espacio moviendo o comprimiendo objetos grandes. Todo esto lo hago monitoreando que no afecte la operación.

a) Activar autoextend si no lo tiene

```
ALTER DATABASE DATAFILE '/ruta/archivo.dbf' AUTOEXTEND ON NEXT 100M MAXSIZE  
UNLIMITED;
```

b)Agregar un nuevo datafile al tablespace

```
ALTER TABLESPACE nombre_tablespace  
ADD DATAFILE '/ruta/nuevo_archivo.dbf' SIZE 1G;
```

c) Mover el segmento a otro tablespace más grande

```
ALTER TABLE tabla  
MOVE TABLESPACE otro_tablespace;
```

Nos ayudaría saber los segmentos que ocupan mas espacio para así tenerlos monitoreados

```
SELECT owner, segment_name, segment_type, tablespace_name,  
ROUND(bytes / 1024 / 1024, 2) AS size_mb  
FROM dba_segments  
WHERE bytes > 100 * 1024 * 1024 -- Segmentos mayores a 100 MB  
ORDER BY size_mb DESC FETCH FIRST 10 ROWS ONLY;
```

## **PREGUNTAS DE CONDUCTA / EXPERIENCIA:**

### **- Contame una situación donde tuviste que restaurar una base de datos en producción.**

Me tocó participar en una restauración cuando se corrompieron unos archivos de data en un ambiente de producción. Yo no era el DBA principal, pero colaboré activamente siguiendo los pasos de recuperación con RMAN.

Recuperamos desde un backup reciente y usamos los archive logs para dejarla al punto exacto.

Aprendí mucho en ese momento, especialmente sobre la importancia de tener backups probados y monitoreo constante.

### **- ¿Cómo manejás los cambios en entornos productivos?**

Siempre trato de que cualquier cambio pase primero por ambiente de desarrollo y pruebas.

En producción, sigo una metodología de cambios controlados, documentando el paso a paso y validando ventanas de mantenimiento.

También coordinamos con el equipo funcional y avisamos a los usuarios.

Si bien no he liderado muchos cambios críticos en producción, sí he participado en implementaciones supervisadas y cada vez soy más cuidadoso con la planificación y el rollback plan.

### **- ¿Alguna vez tuviste que resolver una caída crítica? ¿Qué hiciste?**

Sí, una vez se cayó la base porque el disco donde estaban los logs se llenó. Fue un momento tenso.

Lo que hice fue mantener la calma, ver los logs de alertas, identificar el problema y liberar espacio para reactivar los procesos.

Trabajamos con el equipo de infraestructura para revisar el espacio de disco y ajustamos los parámetros de archivado.