**1.-Consultar el valor del parámetro de la auditoría básica o estándar.**

SHOW PARAMETER AUDIT\_TRAIL;

**2.-Consultar si el parámetro de la auditoría sobre las cuentas de administración es estático o dinámico.**

SELECT NAME, ISSYS\_MODIFIABLE

FROM V$PARAMETER WHERE NAME='audit\_sys\_operations';

**3.-Escribe los comandos para activar que los registros de auditorías se almacenen en un fichero de tipo XML.**

ALTER SYSTEM SET AUDIT\_TRAIL = XML SCOPE=SPFILE;

Debemos reiniciar la base de datos para que el cambio tenga efecto, pues AUDIT\_TRAIL es parámetro estático, debemos conectarnos SLQ \*Plus con el usuario SYS AS SYSDBA, y ejecutar los comandos:

SHUTDOWN IMMEDIATE;

STARTUP;

**4.-Consulta la vista para obtener información sobre los registros de auditorías unificadas.**

SELECT \* FROM UNIFIED\_AUDIT\_TRAIL;

**5.-Asigna al usuario COMPRAR el privilegio de consultar la vista con información sobre los registros de auditorías finas. Se supone que existe el usuario COMPRAR.**

GRANT AUDIT\_ADMIN, AUDIT\_VIEWER TO COMPRAR;

**6.-Consulta la ruta y el fichero que almacena los registros de las auditorías.** SHOW PARAMETER AUDIT\_FILE\_DEST;

**7.-Activa auditar solo las conexiones fallidas.**

AUDIT SESSION WHENEVER NOT SUCCESSFUL;

**8.-Activa auditar la modificación de la tabla ALUMNOS del usuario SECRETARIO para todas las veces que se modifique dicha tabla.**

AUDIT ALTER ON SECRETARIO.ALUMNOS BY ACCESS;

**9.-Activa auditar todas las acciones sobre las tablas del usuario CENTROFOR**.

AUDIT TABLE BY CENTROFOR;

**10.-Activa auditar las consultas sobre la tabla PROVEEDOR del usuario TIENDAIF solo para la primera vez que se consulta.**

AUDIT SELECT ON TIENDAINF.PROVEEDOR BY SESSION;

**11.-Desactiva auditar las operaciones DML de la tabla ALUMNOS del usuario SECRETARIO.**

NOAUDIT INSERT, UPDATE, DELETE, SELECT ON SECRETARIO.ALUMNOS;

**12.-Consulta los valores de las columnas: fecha y hora del evento de auditoría, usuario que efectúa la acción y nombre de la máquina donde sucede el evento de auditoría. De todas las vistas que contienen información sobre las auditorías.**

SELECT TIMESTAMP, USERNAME, USERHOST FROM DBA\_AUDIT\_TRAIL; SELECT TIMESTAMP, USERNAME, USERHOST FROM DBA\_AUDIT\_TRAIL; SELECT TIMESTAMP, USERNAME, USERHOST FROM DBA\_AUDIT\_STATEMENT; SELECT TIMESTAMP, USERNAME, USERHOST FROM DBA\_AUDIT\_SESSION; SELECT TIMESTAMP, USERNAME, USERHOST FROM DBA\_AUDIT\_OBJECT

**13.-Concede al usuario VENDER que pueda crear auditorías finas. Se supone que existe el usuario VENDER.**

GRANT AUDIT\_ADMIN TO VENDER;

**14A.-Crea un índice sobre las columnas NOMPRO, SALPRO, ESPPRO de la tabla PROFESORES del usuario SECRETARIO con sus dos primeras columnas comprimidas.**

CREATE INDEX IX\_PROFESORES ON

PROFESORES (NOMPRO,SALPRO,ESPPRO) COMPRESS 2;

**14B.-Consulta que índices del usuario SECRETARIO están comprimidos.**

SELECT COMPRESSION FROM USER\_INDEXES;

**15.-Consulta sobre la vista que contiene todos los índices de la base de datos la información sobre el índice IX\_PROFESORES.**

SELECT \* FROM dba\_INDEXES WHERE INDEX\_NAME = 'IX\_PROFESORES'; **16.-Consulta las columnas que tienen todos los índices de la base de datos.**

SELECT \* FROM dba\_IND\_COLUMNS;

**17.-Crea una vista basada en SECRETARIO llamada PROF\_TOT\_HOR que muestre el número del profesor, su nombre y el total de horas de los diferentes cursos impartidos. Sólo deben aparecer en la vista los profesores con salario mayor que 1900.**

CREATE OR REPLACE VIEW PROF\_TOT\_HOR (NUMERO, NOMBRE, HORASTOTALES) AS SELECT PROFESORES.NUMPRO, PROFESORES.NOMPRO, SUM(CURSOS.HORCUR) FROM PROFESORES, CURSOS

WHERE PROFESORES.NUMPRO=CURSOS.NUMPRO AND

PROFESORES.SALPRO > 1900

GROUP BY PROFESORES.NUMPRO, PROFESORES.NOMPRO;

**18.-Crea el usuario DIRECTOR con contraseña director, en el espacio de tabla DATOS (se supone que está creada), espacio de tabla temporal TEMP, una cuota de 300 megabytes en DATOS y su cuenta está bloqueada.**

CREATE USER DIRECTOR IDENTIFIED BY director

DEFAULT TABLESPACE DATOS

TEMPORARY TABLESPACE TEMP (se puede omitir)

QUOTA 300M ON DATOS

ACCOUNT LOCK;

**19.-Desbloquea la cuenta del usuario DIRECTOR.**

ALTER USER DIRECTOR

ACCOUNT UNLOCK;

**20.-Consulta el estado de todas las cuentas, mostrando las columnas que indiquen el estado de la cuenta y el nombre del usuario.**

SELECT USERNAME, ACCOUNT\_STATUS FROM DBA\_USERS;

**21.-Asigna los privilegios al usuario VENDER:**

**● Crear roles.**

**● Bloquear una tabla en cualquier esquema.**

**● Poner online u offline en cualquier tablespace.**

**● Eliminar disparadores en cualquier esquema.**

GRANT CREATE ROLE, LOCK ANY TABLE, MANAGE TABLESPACE,

DROP ANY TRIGGER TO VENDER;

**22.-Asigna el privilegio SELECT sobre la tabla PROFESORES del usuario SECRETARIO al usuario DIRECTOR. El usuario DIRECTOR podrá conceder este privilegio a otros usuarios.**

GRANT SELECT ON SECRETARIO.PROFESORES TO DIRECTOR WITH GRANT OPTION;

**23.-Consulta los privilegios del sistema otorgados a los usuarios DIRECTOR, VENDER.**

SELECT \* FROM DBA\_SYS\_PRIVS

WHERE GRANTEE IN ('DIRECTOR','VENDER’);

**24.-Consulta la lista de privilegios del sistema que permite ORACLE**

SELECT \* FROM SYSTEM\_PRIVILEGE\_MAP

ORDER BY NAME;

**25A.-Realiza un programa PL/SQL que pida una cadena por teclado y la muestre en mayúsculas indicando la posición del carácter dentro de la cadena:**

*Ejemplo:*

*La cadena leída es*

*ESTAMOS BIEN*

*La mostrará*

*1E2S3T4A5M6O7S9B10I11E12N*

SET SERVEROUTPUT ON

DECLARE

vcadena VARCHAR2(75);

vcar CHAR;

vposcad PLS\_INTEGER;

vespacio PLS\_INTEGER;

BEGIN

vcadena:=UPPER('&vcadena');

vposcad:=1;

LOOP

vcar:=substr(vcadena,vposcad,1);

if vcar != ' '

then dbms\_output.PUT(vposcad || vcar);

end if;

vposcad:=vposcad+1;

EXIT WHEN vposcad > LENGTH(vcadena);

END LOOP;

dbms\_output.put\_line (' ');

dbms\_output.put\_line (' ');

dbms\_output.put\_line(' ');

END;

**25B.-Realiza un programa que pida un DNI por teclado y muestre si tiene todas las asignaturas aprobadas. El DNI leído de teclado estará formado sólo por números, mostrando un error en caso contrario, también mostrará un error si no se ha encontrado ningún alumno con ese DNI. Se supone que estamos trabajando con el usuario CENTROFOR.**

SET SERVEROUTPUT ON

DECLARE

vcodmatr PLS\_INTEGER;

vnumasig PLS\_INTEGER;

vnumaprob PLS\_INTEGER;

vnumsusp PLS\_INTEGER;

vdniusuario VARCHAR2 (8);

vdniusuarionum PLS\_INTEGER;

BEGIN

vdniusuario:='&vdniusuario';

vdniusuarionum:=TO\_NUMBER(vdniusuario);

SELECT codmatr INTO vcodmatr FROM matricula

WHERE dni=vdniusuario;

SELECT COUNT(\*) INTO vnumasig FROM lineamatricula

WHERE codmatr=vcodmatr;

SELECT COUNT(\*) INTO vnumaprob FROM lineamatricula

WHERE codmatr=vcodmatr AND nota>=5;

vnumsusp:=vnumasig-vnumaprob;

IF vnumsusp=0 THEN

dbms\_output.put\_line('Todo aprobado.');

end if;

EXCEPTION

WHEN value\_error THEN

dbms\_output.put\_line('DNI no numérico');

WHEN no\_data\_found THEN

dbms\_output.put\_line('No hay ningún alumno con el DNI indicado.');

END;

**26.-Diseña un disparador para la tabla ASIGNATURA del usuario CENTROFOR. El disparador se deberá activar antes que se modifique la columna NUMHORAS de la tabla ASIGNATURA. El disparador deberá mostrar uno de los siguientes mensajes**

*La asignatura NOMBREASIGNATURA tiene ahora más horas que antes. La asignatura NOMBREASIGNATURA tiene ahora menos horas que antes.*

*En el caso de introducir un nuevo número de horas igual al que ya tiene la asignatura, no se llevará a cabo la modificación de la columna NUMHORAS de la tabla ASIGNATURA. Igualmente en caso de introducir un nuevo número de horas negativo, no se llevará a cabo la modificación de la columna NUMHORAS de la tabla ASIGNATURA. En ambos casos, se mostrará un error que lo indique.*

CREATE OR REPLACE NONEDITIONABLE TRIGGER CAMBIOHORAS BEFORE UPDATE OF NUMHORAS ON ASIGNATURA

FOR EACH ROW

DECLARE

VTEXTO VARCHAR2(5);

BEGIN

IF :NEW.NUMHORAS > :OLD.NUMHORAS THEN

VTEXTO:='más';

END IF;

IF :NEW.NUMHORAS < :OLD.NUMHORAS THEN

VTEXTO:='menos';

END IF;

IF :NEW.NUMHORAS = :OLD.NUMHORAS THEN

RAISE\_APPLICATION\_ERROR (-20700, 'MISMAS HORAS, DATO SIN MODIFICAR');

END IF;

IF :NEW.NUMHORAS < 0 THEN

RAISE\_APPLICATION\_ERROR (-20701, 'NUEVO NÚMERO NEGATIVO'); END IF;

DBMS\_OUTPUT.PUT\_LINE('La asignatura '||:OLD.NOMBRE|| ' tiene ahora ' ||VTEXTO||' horas que antes.');

END;

**27.-Consulta la información sobre todos los disparados mostrando las columnas de su propietario, la tabla a la que afecta y el estado del disparador.**

SELECT owner, trigger\_name, table\_name,

trigger\_type, triggering\_event, status

FROM DBA\_TRIGGERS;