



Modulo 1.- CONFIDENCIALIDAD DE LOS DATOS

El objetivo de esta parte de la práctica es comprender los distintos mecanismos que se pueden utilizar en ORACLE 10g para gestionar la confidencialidad de los datos en una base de datos (BD): Usuarios, Permisos, Roles y Perfiles. Para ello, se trabajará con tres usuarios diferentes.

Primeramente conectarnos a la instancia ORACLE mediante SQL*Plus. En vez de entrar con usuario, contraseña e instancia, simplemente tecleamos donde el usuario */nolog* y después ejecutamos:

```
connect system/administrador;
```



- 1.1.1.- Si al conectarse, no has podido ¿qué indica el error? ¿te has conectado a una instancia inactiva?, ¿qué hay que hacer para levantar la instancia? ¿Con qué usuario puedes hacerlo? Realiza la operación.
- 1.1.2.- Si al conectarse, te has conectado a una instancia activa, realiza la operación de bajar y volver a levantar la instancia. ¿Con que usuario puedes hacerlo?
- 1.1.3.- Con la instancia activa conectarse como usuario *SYSTEM*.

1.2.- Creación de un espacio virtual de almacenamiento

1.2.1.- Seguidamente se van a crear un espacio de tablas (en el 1.2.2), y antes hay que saber localizar donde se encuentran los ficheros de datos. Para ello existe una vista que da esta información: *DBA_DATA_FILES*, sacar por pantalla sólo los campos de la vista que nos interesa para obtener esta información.

1.2.2.- Crear un espacio de tablas temporal de 10M que se llamará *TEMP_USERXXX* (denotando las letras x vuestro número de usuario). Se creará con almacenamiento gestionado localmente con un tamaño máximo de 20M extensible automáticamente en tamaños de 640k.

1.2.3.- Crear un espacio de tablas permanente llamado *DESARROLLO*, con un tamaño mínimo de 3M y máximo de 10M, con extensiones de 1280K y gestionado automáticamente.

	Diseño de BBDD: Practica Obligatoria	Curso: 2007-2008	
	Segunda Parte: Administración	Convocatoria: Febrero	

1.3.- Creación de un usuario

1.3.1.- Cread un nuevo usuario *JEFE* que tenga por defecto asignado como espacio de tablas virtual temporal *TEMP_USERXXXX* para el almacenamiento y para datos el *DESARROLLO*.

1.3.2.- Comprobar abriendo otra sesión de SQL*Plus si os podéis conectar con este nuevo usuario. ¿Qué pasa?

1.4.- Otorgar Permisos a un Usuario.

1.4.1.- Tras crear el usuario *JEFE*, y para que este pueda conectarse a una instancia de base de datos, le hemos de otorgar permisos para crear una sesión (*CREATE SESSION*). El privilegio *CONNECT* incluye este permiso del sistema. Además debemos proporcionarle permisos para trabajar con los objetos del esquema mediante el privilegio *RESOURCE*.

1.4.2.- Comprobar en otra sesión de SQL*Plus que ya sí se puede conectar.

1.5.- Modificar un Usuario

(Se supone que el usuario *USERXXXX* ya esta creado y tiene privilegios de *CONNECT* y *RESOURCE*)



1.5.1.- Asignar al usuario *USERXXXX* como espacio de tablas temporal *TEMP_USERXXXX* y de datos *DESARROLLO* con cuota ilimitada para *DESARROLLO*.

1.6.- Creación de Objetos en un Esquema

1.6.1.- En la sesión del usuario *USERXXXX*, cargaremos el esquema proporcionado sobre PROYECTOS DE INVESTIGACIÓN EN LA UC3M en el espacio de tablas que tiene asociado. Ejecuta el script *INVESTIGACION.SQL*.

1.7.- Inserción de Datos en la BD

1.7.1.- Una vez creadas las tablas en el espacio de tablas que tenemos asociados, realizamos la inserción de los datos a partir de los archivos (*INSERT*.SQL*) con las sentencias *INSERT*. Es posible que quizás sea necesario deshabilitar algunas restricciones de clave ajena que impidan la carga de datos.

	Diseño de BBDD: Practica Obligatoria	Curso: 2007-2008	
	Segunda Parte: Administración	Convocatoria: Febrero	

1.8 - Creación de Roles.

SYSTEM creara un rol que se asigna posteriormente a usuarios.

1.8.1.- Cread un rol llamado *ROL_JEFEINVESTIGACIÓN* con privilegios para que se pueda realizar consulta sobre qué profesores y a qué grupos de investigación pertenecen, y qué proyectos de investigación se dan en cada uno de los grupos del esquema cargado en *USERXXXX*.

1.8.2.- Asignad este rol al usuario *JEFE*.

1.8.3.- Asignar el privilegio *CREATE SYNONYM* al usuario *JEFE*.

1.8.4.- Como usuario administrador y utilizando alguna de las siguientes vistas: *DBA_ROLES*, *DBA_ROLE_PRIVS*, *DBA_SYS_PRIVS*, *DBA_TAB_PRIVS*, *ROLE_ROLE_PRIVS*, *ROLE_SYS_PRIVS* comprobar lo siguiente:

- Que el rol *ROL_JEFEINVESTIGACIÓN* ha sido definido.

- Que el usuario *JEFE* tiene asignado dicho rol y el privilegios *CREATE SYNONYM*

Indica qué vista se utiliza en cada caso.

1.8.5.- Como usuario *JEFE* haz la siguiente consulta. ¿Qué pasa?

```
Select cod_proyecto, nombre_proyecto
from proyecto
where nombre_grupo='Grupo Investigacion_19'.
```



1.8.6.- ¿Cómo lo solucionarías para poder hacer esta consulta desde *JEFE*? hazlo.

1.8.7.- Crear un sinónimo para la tabla *PROYECTO* llamado *PROYECTO_JEFE* para el usuario *JEFE* de tal forma que pueda referirse a la tabla sin necesidad de usar el prefijo.

1.8.8.- Realiza la consulta del apartado 1.8.5. desde el *JEFE* utilizando sinónimos sobre la tabla *PROYECTO* de *USERXXXX* que hay que crear.

1.8.9.- Desde sesión de usuario *JEFE* haz la siguiente inserción:

```
INSERT INTO userxxxx.grupo
VALUES
('Grupo Investigacion_prueba', 'Area_32',
'Derecho Social e Internacional Privado',
'022088405');
```

	Diseño de BBDD: Practica Obligatoria	Curso: 2007-2008	
	Segunda Parte: Administración	Convocatoria: Febrero	

¿Por qué no puede?, si se han dado privilegios a *USERXXXX* sobre la tabla *GRUPO*.

1.8.10.- Dar permisos desde el usuario *USERXXXX* para que el *JEFE* sí pueda realizar inserciones.

1.8.11.- En la sesión de *JEFE* crear un sinónimo para *USERXXXX.GRUPO* que se llame *GRUPO_JEFE* y comprobar (después de haber dado permisos en 1.8.10) que ya sí se puede realizar la inserción anterior.

1.9.- Creación de perfiles

Por último, en cuanto a confidencialidad se refiere, como usuario administrador vamos a crear un perfil que sea capaz de limitar los recursos a los que tiene acceso un usuario cuando se conecta a la BD y de gestionar todo lo referente a la contraseña del usuario.

1.9.1.- Las características que deseamos que posea nuestro perfil son las siguientes:

- Nombre = COORDINADOR
- Intentos de conexión fallidos (error en la contraseña) = 3

1.9.2.- Modifica este perfil para que además:



- Tiempo de inactividad de la conexión = 2 minuto
- Tiempo máximo de sesión = 90 minutos
- Límite de sesiones concurrentes del usuario = 2
- Límite de tiempo de acceso a CPU por llamada = 6000 segundos

1.9.3. Asignad dicho perfil al usuario *USERXXXX*.

Modulo 2.- BACKUP Y RECUPERACIÓN.

El objetivo de este módulo es aplicar los conocimientos dados en teoría poniéndolos en práctica. En este modulo se hace hincapié en los conceptos de "checkpoint", "redo" y "archiver". Una vez aplicados estos conceptos se pasará a aplicar un backup físico y lógico y su restauración.

2.1. Como primer apartado deberemos verificar en qué modo esta la base de datos, y ponerla en modo de "archivelog".

	Diseño de BBDD: Practica Obligatoria	Curso: 2007-2008	
	Segunda Parte: Administración	Convocatoria: Febrero	

2.1.1.- Para ello nos conectaremos con el usuario apropiado y consultaremos la vista *V\$DATABASE*, hacer la consulta con sólo los campos de la vista que nos interesa para obtener esta información. ¿En que modo se encuentra la base de datos?

2.1.2.- Si la instancia esta en modo no "archivelog" vamos a cambiar a modo "archivelog". Para ello se deberá bajar la instancia. Montar la base de datos, ¿Con qué usuario? ¿Qué significa montar la base de datos?. Ejecutamos: `Alter database archivelog`, ¿Qué significa poner la base de datos en "archivelog"? Y por ultimo abrir o levantar la base de datos.

2.1.3.- Verificar el estado de la base de datos comprobando que ya esta en modo "archivelog".

2.2. Forzar checkpoint en la Base de Datos

2.2.1.- En la sesión de SQLPlus apuntar el número de "checkpoint" consultando la vista *V\$DATABASE* (sólo con los campos de la vista que nos interesa para obtener la información pedida). ¿Qué es un checkpoint?.

2.2.2.- Forzar a realizar un "checkpoint", ejecutando la sentencia: `alter system checkpoint`.

2.2.3.- Consultar otra vez el número de "checkpoint". ¿Qué ha pasado? ¿Qué indica ese cambio?

2.3. Consultar información sobre ficheros de logfile, datafile y controlfile.



2.3.1.- Realizar tres consultas para encontrar el nombre y lugar donde residen los ficheros de "redo" (utilizando al vista *V\$LOGFILE*, sólo los campos que interese), los "datafiles" (utilizando al vista *V\$DATAFILE*, sacar sólo los campos que interese) y los ficheros de control (utilizando la vista *V\$CONTROLFILE*, sacar sólo los campos que interese).

2.4. Sentencias para comprobación con usuario *USERXXXX*

2.4.1.- Consultar el número de datos de la tabla *PROGRAMA*.

2.4.2- Crea la siguiente tabla *DESPACHO* e inserta 5 registros:

```
create table despacho (
id      number(5) primary key,
numero_despacho  varchar(50));
```

	Diseño de BBDD: Practica Obligatoria	Curso: 2007-2008	
	Segunda Parte: Administración	Convocatoria: Febrero	

2.5. Realización de backup.

2.5.1.- ¿Cual es el espacio de tablas por defecto del usuario *USERXXXX*? Hacer una consulta a la vista *DBA_TABLESPACES*.

2.5.2. Realizaremos con el usuario indicado un backup del espacio de tablas por defecto (*DESARROLLO*) del usuario *USERXXXX* mediante la sentencia:

```
alter tablespace DESARROLLO begin backup.
```

2.5.3.- Realizar la copia del fichero de datos *DESARROLLO.ORA* ejemplo mediante el comando "copy" de Windows a través de SQLPlus pero anteponiendo la palabra clave "host" y lo copiáis en "c:\temp", sería:

```
"host                                     copy
C:\oracle\product\10.2.0\oradata\orcl\DESARROLLO.ora
c:\temp\DESARROLLO.ORA"
```

2.5.4.- Comprobar que el fichero efectivamente se ha copiado en la carpeta "c:\temp", dar la imagen de la captura de la pantalla del explorador de Windows.

2.5.5.- Finalizar backup mediante: `alter tablespace DESARROLLO end backup.`

2.5.6.- Desde el *USERXXXX* insertar más datos en la tabla *PROGRAMA* (de 3 a 10 registros).



2.5.7.- Consultar el número de datos de la tabla *PROGRAMA*.

2.6. Realización de proceso de exportación

2.6.1.- Hay que realizar un "export" solamente del usuario *USERXXXX* (utilizar parámetro "owner=") mediante la consola de comandos Windows, dejando el resultado del fichero de datos en c:\temp\exportXXXX.dmp y el de log igualmente en c:\temp\exportXXXX.log. (Para mas información en otn de oracle o en la consola de comandos Windows poniendo "exp help=yes").

2.6.2.- Con el usuario *USERXXXX* borra la tabla *DESPACHO*.

2.7.- Simulación de un fallo de sistema con necesidad de proceso de recuperación. En esta apartado vamos suponer que ha un espacio de tablas ha sufrido daños por problemas diversos, caída de disco, corrupción, etc. Para simular esta caída como Oracle el "datafile" actual lo sustituimos por el antiguo, por el del backup copiado en "c:/temp" y pasamos a recuperar. Para ello deberemos desconectar

	Diseño de BBDD: Practica Obligatoria	Curso: 2007-2008	
	Segunda Parte: Administración	Convocatoria: Febrero	

todas las sesiones, incluida la que tengáis como SYSTEM, ya que si hay algún usuario conectado no podremos bajar la instancia de base de datos.

2.7.1. Bajar/parar la instancia.

2.7.2. Renombrar el fichero *DESARROLLO.ORA* donde se encuentran los "datafiles" de la instancia por el *DESARROLLO_COPIA1.ORA.old*.

2.7.3.- Copiar el fichero guardado en "c:\temp*DESARROLLO.ORA*" en la localización de los "datafiles" de la instancia.

2.7.4.- Levantar la instancia. ¿Qué mensajes dan? ¿Por qué?

2.8.- Proceso de recuperación

2.8.1.- Recuperar el datafile con `recover datafile <fichero a recuperar>`.

2.8.2.- Abrir la base de datos. Volver a conectarse con el usuario *USERXXXX* y verificar el numero de tuplas en las tablas *PROGRAMA* y *DESPACHO*.

2.8.3.-Según lo realizado en el apartado 2.7.3. la tabla programa debería tener 15 tuplas, ¿por qué no es así? ¿En que ha consistido el proceso de recuperación? ¿Ha habido perdida de información después del fallo de sistema simulado en el apartado 2.7.? ¿Si estuviéramos en modo "no archiveolog", cuantas tuplas tendría la tabla *PROGRAMA*?

2.9.- Proceso de importación a partir de la exportación (2.6.).



2.9.1.- Mostrar con una consulta el número de tuplas de la tabla *DESPACHO*.

2.9.2. Como no existe la tabla *DESPACHO*. Recuperarla con un "import" a partir del fichero de exportación generado en el apartado 2.6.(Mirar la misma ayuda que con el "export").

2.9.3. - Una vez recuperada, mostrar con una consulta el número de tuplas de la tabla *DESPACHO*.

Módulo 3.-OPTIMIZACIÓN DE CONSULTAS

El objetivo de esta parte es comprender cuál es el mecanismo de ajuste y optimización de consultas en ORACLE y cómo modificando la estructura física de la base de datos se puede influir en el tiempo de

	Diseño de BBDD: Practica Obligatoria	Curso: 2007-2008	
	Segunda Parte: Administración	Convocatoria: Febrero	

ejecución de una consulta. Estos ejemplos tienen un carácter ilustrativo de los principales conceptos que se manejan en la optimización de consultas así como de la manera de proceder. Sin embargo en un caso real, será necesario considerar la frecuencia de las consultas, actualizaciones, etc. así como contar con una cantidad de datos considerable para que realmente notemos mejoras significativas.

El optimizador de consultas de ORACLE funciona en dos modos:

- El optimizador basado en reglas utiliza heurísticas generadas por expertos para la generación de los planes de ejecución física.

- El optimizador por costes calcula el coste aproximado de las operaciones de acceso físico (y opcionalmente de CPU). Para ello se basa en estadísticas obtenidas a partir de los datos de las tablas. Estas estadísticas son por ejemplo el número de filas y longitud de las filas, el número de claves distintas, el tamaño de los índices, etc. La recolección de estadísticas debe de ser programada o realizada por el administrador de forma que el estado dinámico de la base de datos se refleje a la hora de calcular los costes.

3.1.- Recolección de estadísticas.



Para la recolección de estadísticas se utiliza el paquete de procedimientos *DBMS_STATS* que proporciona herramientas para la gestión de estadísticas de las tablas. Los resultados del análisis de las tablas y sus estadísticas se almacenan en el diccionario de datos en vistas como **_TABLES* o **_INDEXES*.

3.1.1.- Para recoger las estadísticas usamos el procedimiento *GATHER_SCHEMA_STATS* que analiza todos los objetos de un esquema (el valor NULL del parámetro esquema indica que se realice el análisis sobre el esquema actual y la opción *CASCADE* permite extenderlo a todos los índices asociados a una tabla). Ejecutar:

```
Execute DBMS_STATS.GATHER_SCHEMA_STATS(NULL, CASCADE
=> TRUE);
```

3.2.- Consulta de estadísticas

3.2.1.- Consultad el número de filas, bloques y otras estadísticas que se consideren interesantes sobre las tablas e índices del esquema usando la vistas del diccionario *USER_TABLES*, *USER_INDEXES* (en las consultas solo seleccionar los campos de la vistas necesarios para dar la información pedida). ¿Qué tablas y qué índices son los más grandes?

	Diseño de BBDD: Practica Obligatoria	Curso: 2007-2008	
	Segunda Parte: Administración	Convocatoria: Febrero	

¿Cuál es la altura/profundidad de estos índices? ¿Cuántos bloques a nivel de hoja tienen?

3.3.- Desde *USERXXXX*. Consulta de planes de ejecución

3.3.1. Para poder consultar y visualizar los planes de ejecución es necesario que creamos una nueva tabla adicional que almacena los resultados temporales del optimizador. Esta tabla por defecto se llama *PLAN_TABLE* y podemos crearla y consultarla ayudándonos de unos scripts que proporciona ORACLE. El script de creación se llama *UTLXPLAN.SQL* situado junto a otros scripts de administración de ORACLE en el directorio *C:\oracle\product\10.2.0\db_1\RDBMS\ADMIN* en el virtualPC de la instalación. Ejecutar el script.

3.3.2.- Ahora vamos a proceder a consultar cuál es el plan de ejecución para una consulta sencilla:

```
Select dni, dedicacion from participa;
```

Generad el plan de ejecución de la consulta con la cláusula *EXPLAIN PLAN* que genera el plan y lo almacena en la tabla *PLAN_TABLE*.

3.3.3. Visualizad el plan consultando la tabla *PLAN TABLE*. El script *UTLXPLS.SQL* os puede ayudar a visualizar el plan de ejecución con más claridad. ¿Qué coste aproximado da la consulta?, ¿Podéis explicar qué es lo que aparece en el plan de ejecución?

3.3.4. Sin embargo si realizamos la consulta similar: `Select dni from participa;`

¿Qué coste tiene? ¿Cuál es la principal diferencia en el plan de ejecución con la consulta del apartado anterior? ¿Por qué?



3.3.5. ¿Se te ocurre alguna forma de mejorar el coste que tarda en realizar la consulta del apartado 3.3.3. mediante un índice? En caso afirmativo, creadlo y después, generar y visualizar el plan de la ejecución de la consulta del apartado 3.3.3 nuevamente. ¿Habéis reducido bajar el coste? ¿Qué diferencia hay en el plan de ejecución con el del apartado 3.3.3.? ¿Por qué?

3.4.- Diseño físico para optimización de consultas

(Borrar el índice creado en el apartado anterior)

3.4.1.- Generad y explicad el plan de ejecución de la siguiente consulta.

```
SELECT DEDICACION, COD_PROYECTO FROM PARTICIPA, PROFESOR
WHERE NOT EXISTS
(SELECT DNI FROM profesor
```

	Diseño de BBDD: Practica Obligatoria	Curso: 2007-2008	
	Segunda Parte: Administración	Convocatoria: Febrero	

```

WHERE PARTICIPA.COD_PROYECTO= '21392' AND PARTICIPA.DNI=PROFESOR.DNI)
AND
PROFESOR.NOMBRE_DPTO='Física'
GROUP BY COD_PROYECTO, DEDICACION
HAVING DEDICACION >40;

```

3.4.2.- ¿Se te ocurre alguna forma de mejorar el coste que se tarda en realizar esta consulta mediante un índice? Volver a generar el plan de ejecución. ¿Cómo ha cambiado el plan de ejecución? ¿Podéis explicar el nuevo plan?