PROYECTO FINAL
BDD empleada para administrar el negocio de la empresa NetMAX – PARTE 4

1.1. CARGA DE DATOS.

- En esta última parte se realizará la carga de datos para validar los conceptos de transparencia implementados anteriormente.
- Para realizar esta actividad se proporciona un archivo zip llamado `carga-inicial.zip`. Este archivo contiene una carpeta llamada `carga-inicial`. En dicha carpeta se encuentra una serie de scripts SQL con sentencias `insert` así como una muestra de imágenes empleadas para validar el soporte de datos BLOB.
- Se recomienda colocar la carpeta `carga-inicial` dentro del mismo directorio donde se encuentran todos los scripts SQL del proyecto. Esta convención permitirá usar las rutas que se muestran en este documento y no se necesitará realizar modificaciones a las rutas o nombres de directorios.
- Adicional al archivo zip, se incluyen algunos scripts SQL para realizar la validación de resultados. Copiar estos archivos al directorio donde se encuentran todos los scripts SQL del proyecto.

1.1.1. Preparar el ambiente para cargar un archivo BLOB de prueba.

- Crear un script llamado `s-07-netmax-configuracion-soporte-blobs.sql` El archivo deberá ser ejecutado en todos los nodos para configurar y crear los objetos necesarios para poder leer datos BLOB/CLOB del sistema de archivos local. El script deberá realizar lo siguiente:
  - Crear un objeto tipo `directory` llamado `proy_final_pdfs_dir` que apunte a un directorio del sistema de archivos. Este directorio contendrá archivos PDF de prueba para ser insertados en la tabla `archivo_programa`. Por simplicidad se elige al directorio `/tmp/bdd/proyecto-final/pdfs`.
  - Crear un objeto tipo `directory` llamado `proy_final_trailers_dir` que apunte a un directorio del sistema de archivos. Este directorio contendrá archivos de video para ser insertados en la tabla `documental`. Por simplicidad se elige al directorio `/tmp/bdd/proyecto-final/trailers`.
  - Para que el usuario `netmax` pueda crear, leer y escribir en el directorio, este deberá contar con el privilegio `create any directory`. Se recomienda agregar este privilegio en el script `s-01-netmax-usuario.sql`
  - Crear una función llamada `fx_carga_blob` que se empleará en instrucciones `insert` para leer un archivo binario y guardarlo en una columna `BLOB`. El código del script se muestra a continuación.

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación:  dd/mm/yyyy
--@Descripción:     Script empleado para configurar el
--                  Soporte de datos BLOB.

Prompt Creando objetos para leer datos BLOB

Prompt creando directorio
-- el usuario netmax debe tener el privilegio create any directory

--Objeto tipo Directory para representar al campo archivo_programa.archivo
create or replace directory proyecto_final_pdfs_dir as '/tmp/bdd/proyecto-final/pdfs';

--Objeto tipo Directory empleado para almacenar los trailers de un documental.
create or replace directory proyecto_final_trailers_dir as '/tmp/bdd/proyecto-final/trailers';

Prompt creando funcion para leer datos BLOB
create or replace function fx_carga_blob(
 p_directory_name      in varchar2,
 p_src_file_name       in varchar2 ) return blob is

--variables
v_src_blob bfile:=bfilename(upper(p_directory_name),p_src_file_name);
v_dest_blob blob := empty_blob();
v_src_offset number := 1;
v_dest_offset number :=1;
v_src_blob_size number;
```

```
begin
    if dbms_lob.fileexists(v_src_blob) =0 then
        raise_application_error(-20001, p_src_file_name
            ||'  El archivo '
            ||p src file name
            ||' no existe en '
            ||p_directory_name
        );
    end if;

    --abre el archivo
    if dbms lob.isopen(v src blob) = 0 then
        dbms_lob.open(v_src_blob,dbms_lob.LOB_READONLY);
    end if;

 v src blob size := dbms lob.getlength(v src blob);

    --crea un objeto lob temporal
    dbms_lob.createtemporary(
        lob_loc => v_dest_blob
      , cache   => true
      , dur     => dbms lob.call
    );

    --lee el archivo y escribe en el blob
    dbms_lob.loadblobfromfile(
        dest_lob    => v_dest_blob,
        src bfile   => v src blob,
        amount      => dbms_lob.getlength(v_src_blob),
        dest offset => v dest offset,
        src_offset  => v_src_offset
    );
    --cerrando blob
    dbms lob.close(v src blob);

    if v_src_blob_size<> dbms_lob.getlength(v_dest_blob) then
      raise_application_error(-20104,
        'Numero de byes leidos VS escritos no coinciden: '
        ||v src blob size||', actual: '|| dbms lob.getlength(v dest blob));
    end if;
    return v dest blob;
end;
/
show errors
```

- Crear un script llamado `s-07-netmax-main-soporte-blobs.sql` Este script deberá ejecutar el script anterior en cada nodo.

Ejemplo:

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación:
--@Descripción:     Script principal empleado para configurar el soporte
--                  de datos BLOB en los 4 nodos.

Prompt configurando directorios y otorgando registros.

--jrcbd_s1
Prompt configurando  soporte BLOB  para jrcbd_s1
connect netmax bdd/netmax bdd@jrcbd s1
@s-07-netmax-configuracion-soporte-blobs.sql

--jrcbd_s2
Prompt configurando  soporte BLOB  para jrcbd s2
connect netmax bdd/netmax bdd@jrcbd s2
@s-07-netmax-configuracion-soporte-blobs.sql

--arcbd_s1
Prompt configurando  soporte BLOB  para arcbd s1
connect netmax_bdd/netmax_bdd@arcbd_s1
@s-07-netmax-configuracion-soporte-blobs.sql

--arcbd_s2
Prompt configurando  soporte BLOB  para arcbd_s2
connect netmax_bdd/netmax_bdd@arcbd_s2
@s-07-netmax-configuracion-soporte-blobs.sql

Prompt Listo !
```

1.2. PRESENTACIÓN DEL PROYECTO.

Para realizar la presentación del proyecto, se deberán crear y ejecutar los siguientes scripts.

1.2.1. Presentación 1: Creación de la BDD

- Crear un `script s-08-netmax-presentacion-1.sql` encargado de ejecutar todos los scripts que generan la BDD.

Ejemplo:

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación: dd/mm/yyyy
--@Descripción:    Script encargado de crear la BDD

clear screen
whenever sqlerror exit rollback;
Prompt Iniciando con la creacion de la  BDD.

@s-01-netmax-main-usuario.sql
@s-02-netmax-ligas.sql
@s-03-netmax-main-ddl.sql
@s-04-netmax-main-sinonimos.sql
@s-05-netmax-main-vistas.sql
@s-06-netmax-main-triggers.sql
@s-07-netmax-main-soporte-blobs.sql

Prompt Listo !
exit
```

1.2.2. Presentación 2: Carga de datos por copia manual

- Crear un script llamado `s-08-netmax-presentacion-2.sql` El script se conectará a cada PDB y realizará la inserción de los datos en donde no existe fragmentación, ni esquema de replicación. En este caso, el script contendrá la carga para la tabla `status_programa`

Ejemplo:

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación:  dd/mm/yyyy
--@Descripción:     Archivo de carga inicial.
clear screen
whenever sqlerror exit rollback;

Prompt ======================================
Prompt Cargando catalogos replicados en jrcbd_s1
Prompt ======================================
connect netmax_bdd/netmax_bdd@jrcbd_s1
delete from status_programa;
@carga-inicial/netmax-carga-inicial-status-programa.sql
commit;

Prompt ======================================
Prompt Cargando catalogos replicados en jrcbd_s2
Prompt ======================================
connect netmax_bdd/netmax_bdd@jrcbd_s2
delete from status_programa;
@carga-inicial/netmax-carga-inicial-status-programa.sql
commit;

Prompt ======================================
Prompt Cargando catalogos replicados en arcbd_s1
Prompt ======================================
connect netmax bdd/netmax bdd@arcbd s1
delete from status_programa;
@carga-inicial/netmax-carga-inicial-status-programa.sql
commit;

Prompt ======================================
Prompt Cargando catalogos replicados en arcbd s2
Prompt ======================================
connect netmax bdd/netmax bdd@arcbd s2
delete from status_programa;
@carga-inicial/netmax-carga-inicial-status-programa.sql
commit;

Prompt Carga de datos replicados exitosa!.
exit
```

1.2.3. Presentación 3: Carga de datos con transparencia de distribución.

- Crear un script llamado `s-08-netmax-presentacion-3.sql` Este archivo contendrá la carga inicial de todas las tablas globales así como la ejecución de algunas consultas para validar el correcto funcionamiento.
- Este script a su vez invoca a un script llamado `s-08-netmax-presentacion-3.sh` Este archivo contiene un pequeño programa (Shell script) que realiza la copia de las imágenes en los directorios configurados anteriormente. El código de este script se muestra a continuación. Leer su contenido y en caso de requerir, actualizar rutas.

Ejemplo:

- Shell Script.

```
#!/bin/bash
#@Autor:          Jorge A. Rodríguez C
#@Fecha creación:  dd/mm/yyyy
#@Descripción:     Copia archivos binarios

#Si ocurre un error, el programa termina.
set -e
set -o pipefail

#En caso de no econtrar el directorio, extrae el contenido del archivo zip
if [ ! -d "/tmp/bdd/proyecto-final/pdfs" ]; then

  echo "Copiando archivos pdf de muestra "
  mkdir -p /tmp/bdd/proyecto-final/pdfs
  unzip carga-inicial/pdfs.zip -d /tmp/bdd/proyecto-final

else
  echo "=> Los archivos PDF de muestra ya fueron copiados"
fi

if [ ! -d "/tmp/bdd/proyecto-final/trailers" ]; then
  echo "Copiando archivos de video  de muestra"
  mkdir -p /tmp/bdd/proyecto-final/trailers
  unzip carga-inicial/trailers.zip -d /tmp/bdd/proyecto-final
else
  echo "=> Los archivos de video de muestra ya fuero copiados."
fi

#actualiza permisos
chmod -R 755 /tmp/bdd/proyecto-final
```

Ejemplo:

- Script SQL

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación:  dd/mm/yyyy
--@Descripción:     Archivo de carga inicial - fragmentos
clear screen
--Para visualizar export NLS_LANG=SPANISH_SPAIN.WE8ISO8859P1

Prompt ======================================
Prompt Preparando carga de Datos
Prompt ======================================

Prompt => Seleccionar la PDB LOCAL para insertar datos
Prompt => Para seleccionar una PDB remota, asegurarse que los archivos existen.

connect netmax bdd/netmax bdd@&pdb

Prompt Personalizando el formato de fechas
alter session set nls_date_format='yyyy-mm-dd hh24:mi:ss';

Prompt => Al ocurrir un error se saldrá del programa y se hará rollback
whenever sqlerror exit rollback

Pause  => Presionar Enter para Iniciar con la extracción de datos binarios, Ctrl-C para cancelar

--Invoca a un shell script para realizar la extracción y copia de archivos
!sh s-08-netmax-presentacion-3.sh

Prompt ================================================
Prompt  ¿ Listo para Iniciar con la carga ?
Prompt ================================================
Pause  => Presionar Enter para Iniciar, Ctrl-C para cancelar
```

```
Prompt => Realizando limpieza inicial ....
set feedback off
Prompt Eliminando datos de PLAYLIST
delete from playlist;

Prompt Eliminando datos de USUARIO
delete from usuario;

Prompt Eliminando datos de SERIE
delete from serie;

Prompt Eliminando datos de PELICULA
delete from pelicula;

Prompt Eliminando datos de DOCUMENTAL
delete from documental;

Prompt Eliminando datos de HISTORICO
delete from historico_status_programa;

Prompt Eliminando datos de ARCHIVO_PROGRAMA
delete from archivo programa;

Prompt Eliminando datos de PROGRAMA
delete from programa;

Prompt Eliminando datos de PAIS
delete from pais;

Prompt Eliminando datos de TIPO CUENTA
delete from tipo_cuenta;

Prompt Eliminando datos de TIPO_SERIE
delete from tipo serie;

Prompt => Realizando Carga de datos ....

Prompt Cargando PAIS
@carga-inicial/netmax-carga-inicial-pais.sql

Prompt Cargando TIPO SERIE
@carga-inicial/netmax-carga-inicial-tipo-serie.sql

Prompt Cargando TIPO_CUENTA
@carga-inicial/netmax-carga-inicial-tipo-cuenta.sql

Prompt Cargando USUARIO
@carga-inicial/netmax-carga-inicial-usuario.sql

Prompt Cargando PROGRAMA (DOCUMENTAL)
@carga-inicial/netmax-carga-inicial-programa-documental.sql

Prompt Cargando PROGRAMA (PELICULA)
@carga-inicial/netmax-carga-inicial-programa-pelicula.sql

Prompt Cargando PROGRAMA (SERIE)
@carga-inicial/netmax-carga-inicial-programa-serie.sql

Prompt Cargando HISTORICO_STATUS_PROGRAMA
@carga-inicial/netmax-carga-inicial-historico-status-prog.sql

Prompt Cargando DOCUMENTAL (con datos BLOB)
@carga-inicial/netmax-carga-inicial-documental-empty-blob.sql
--@carga-inicial/netmax-carga-inicial-documental.sql

Prompt Cargando SERIE
@carga-inicial/netmax-carga-inicial-serie.sql

Prompt Cargando PELICULA
@carga-inicial/netmax-carga-inicial-pelicula.sql

Prompt Cargando PLAYLIST
@carga-inicial/netmax-carga-inicial-playlist.sql

Prompt cargando ARCHIVO_PROGRAMA (con datos BLOB)
--@carga-inicial/netmax-carga-inicial-archivo-programa.sql
@carga-inicial/netmax-carga-inicial-archivo-programa-empty-blob.sql

Prompt Carga de datos  replicados exitosa, haciendo commit!.
commit;
```

```
exit
```

Observar las partes en Negritas. Se muestran 2 versiones del script. Uno de ellos contiene el prefijo `empty_blob` Esto significa que la tabla contiene columnas BLOB y que el valor de la columna está vacía, es decir, se está haciendo uso de la función `empty_blob`.
El script que se proporciona es justamente el script que contiene la función `empty_blob`. Lo anterior implica que se deben realizar las siguientes acciones:

- Renombrar el archivo para que coincida con el valor esperado, es decir, remover el prefijo ***`empty_blob`***.
- Editar el archivo para que ya no haga uso de la función `empty_blob`. En su lugar se deberá invocar a la función `fx_carga_blob` que fue creada anteriormente. Esta función será la encargada de leer el contenido de un archivo binario y regresar un objeto BLOB que será insertado en las tablas `documental` y `archivo_programa`.
- La función recibe 2 parámetros:
    - El nombre del objeto tipo DIRECTORY que apunta al directorio donde se encuentran los archivos binarios. Recordando, anteriormente se crearon 2 objetos DIRECTORY llamados `PROYECTO_FINAL_PDFS_DIR` y `PROYECTO_FINAL_TRAILERS_DIR`.
    - El nombre del archivo cuyo contenido será guardado en la BDD. Se deberá seleccionar un nombre de archivo aleatorio del directorio. Los archivos están numerados iniciando en 1. Se cuenta con un total de **50** archivos de muestra (videos) para simular a un tráiler de un documental, y un total de **65** archivos pdf que se emplean para simular archivos de un programa. Con estas características, es posible crear una cadena que contenga un número aleatorio entre estos rangos. Por ejemplo: `'sample'||num_aleatorio||'.mp4'` o `'sample'||num_aleatorio||'.pdf'` Se recomienda emplear la función `dbms_random.value` y `round` para generar un número aleatorio.
- Con estos cambios, cada sentencia `insert` incluirá un objeto BLOB. Se recomienda dejar esta actividad hasta el final para reducir los tiempos de carga de datos.
- Realizar este cambio al menos a **100** registros de cada script.
- Para hacer el reemplazo, se recomienda emplear un editor de texto y realizar una operación de "Find & replace".

1.2.4. Presentación 4: Validación de resultados – Insert y datos replicados.

- Ejecutar el script `s-08-netmax-presentacion-4.plb` Se recomienda ejecutar este script en cada una de las 4 PDBs para validar que todo esté funcionando correctamente.

1.2.5. Presentación 5: Eliminación de datos.

- Crear un script `s-08-netmax-presentacion-5.sql` Este se encargará de verificar que la transparencia de eliminación funciona correctamente. El script deberá eliminar los datos de todas las tablas. Hacer uso de un procedimiento almacenado que invoque instrucciones delete en orden correcto para realizar el vaciado de los datos. Se crea una transacción distribuida para realizar esta operación. Cualquier error que ocurra durante el borrado de datos provocará que se haga un `rollback` de todo el proceso.

Ejemplo:

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación:  dd/mm/yyyy
--@Descripción:    Script de eliminación de datos

Prompt Seleccionar la PDB para realizar la eliminación de datos

connect netmax_bdd/netmax_bdd@&pdb
set serveroutput on
Prompt Eliminando datos ...
declare
 v_formato varchar2(50) := 'yyy-mm-dd hh24:mi:ss';

begin
 dbms_output.put_line(to_char(sysdate,v_formato)
  || ' Eliminando datos de playlist');
 delete from playlist;

--completar

commit;
```

```
exception
 when others then
  dbms_output.put_line('Errores detectados al realizar la eliminacion');
  dbms_output.put_line('Se hara rollback');
  rollback;
  raise;
end;
/
Prompt Listo!
exit
```

1.2.6. Presentación 6: Validación de resultados - delete.

- Ejecutar el script `s-08-netmax-presentacion-6.plb` Se recomienda ejecutar este script en cada una de las 4 PDBs para validar que todo esté funcionando correctamente. No olvidar volver a cargar datos antes de realizar la prueba. El script verifica que la transparencia de distribución para instrucciones `delete` funciona correctamente.

FIN!