

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_dir` que apunte a un directorio del sistema de archivos. Por simplicidad se elige al directorio `/tmp/bdd/proyecto-final`. 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
create or replace directory proy_final_dir as '/tmp/bdd/proyecto-final';
```

Prompt creando funcion para leer datos BLOB

```
create or replace function fx_carga_blob(
  v_directory_name      in varchar2,
  v_src_file_name       in varchar2 ) return blob is

--variables
v_src_blob bfile:=bfilename(v_directory_name,v_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, v_src_file_name
    ||' El archivo no existe ');
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
    dbms_output.put_line('done '|| v_src_blob_size || ' bytes read.' );
else
    raise_application_error(-20104,'Invalid blob size. Expected: '
        ||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 durante la demostración del proyecto.

1.2.1. Creación de la BDD

- Crear un script llamado `s-00-netmax-drop.sql` Este script será empleado para borrar el usuario existente.

Ejemplo:

```
--@Autor:           Jorge A. Rodríguez C
--@Fecha creación:  dd/mm/yyyy
--@Descripción:     eliminación del usuario netmax_bdd y sus objetos
--                  Notar que se usa un programa para validar si existe.
```

```
Prompt eliminando al usuario netmax_bdd
accept syspass char prompt 'Proporcione el password de sys: ' hide
```

```
--jrcbd_s1
prompt eliminando usuario netmax_bdd
connect sys/&&syspass@jrcbd_s1 as sysdba
set serveroutput on
declare
  v_count number(1,0);
begin
  select count(*) into v_count
  from dba_users
  where lower(username) = 'netmax_bdd';
  if v_count > 0 then
    execute immediate 'drop user netmax_bdd cascade';
  else
    dbms_output.put_line('El usuario no existe');
  end if;
end;
/
```

```
--jrcbd_s2
prompt Conectando a jrcbd_s2 como sys
connect sys/&&syspass@jrcbd_s2 as sysdba
set serveroutput on
declare
  v_count number(1,0);
begin
  select count(*) into v_count
  from dba_users
  where lower(username) = 'netmax_bdd';
  if v_count > 0 then
    execute immediate 'drop user netmax_bdd cascade';
  else
    dbms_output.put_line('El usuario no existe');
  end if;
end;
/
```

```
--arcbd_s1
prompt Conectando a arcbd_s1 como sys
connect sys/&&syspass@arcbd_s1 as sysdba
set serveroutput on
declare
  v_count number(1,0);
begin
  select count(*) into v_count
  from dba_users
  where lower(username) = 'netmax_bdd';
  if v_count > 0 then
    execute immediate 'drop user netmax_bdd cascade';
  else
    dbms_output.put_line('El usuario no existe');
  end if;
end;
/
```

```
--arcbd_s2
prompt Conectando a arcbd_s2 como sys
connect sys/&&syspass@arcbd_s2 as sysdba
set serveroutput on
declare
  v_count number(1,0);
begin
  select count(*) into v_count
  from dba_users
  where lower(username) = 'netmax_bdd';
  if v_count > 0 then
    execute immediate 'drop user netmax_bdd cascade';
  else
    dbms_output.put_line('El usuario no existe');
  end if;
end;
/
```

- 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-00-netmax-drop.sql
@s-01-netmax-usuario.sql
@s-02-netmax-ligas.sql
@s-03-netmax-main-ddl.sql
@s-04-netmax-main-sinonimos.sql
@s-05-netmax-vistas-main.sql
@s-06-netmax-main-trigger.sql
@s-07-netmax-main-soporte-blobs.sql
```

```
Prompt Listo !
exit
```

1.2.1. Carga de datos replicados

- 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 (datos replicados).

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 pais;
delete from tipo_serie;
delete from tipo_cuenta;
delete from status_programa;
@carga-inicial/netmax-carga-inicial-pais.sql
@carga-inicial/netmax-carga-inicial-tipo-serie.sql
@carga-inicial/netmax-carga-inicial-tipo-cuenta.sql
@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 pais;
delete from tipo_serie;
delete from tipo_cuenta;
delete from status_programa;
@carga-inicial/netmax-carga-inicial-pais.sql
@carga-inicial/netmax-carga-inicial-tipo-serie.sql
@carga-inicial/netmax-carga-inicial-tipo-cuenta.sql
@carga-inicial/netmax-carga-inicial-status-programa.sql
commit;

Prompt =====
Prompt Cargando catalogos replicados en arcdb_s1
Prompt =====
connect netmax_bdd/netmax_bdd@arcdb_s1
delete from pais;
delete from tipo_serie;
delete from tipo_cuenta;
delete from status_programa;
@carga-inicial/netmax-carga-inicial-pais.sql
@carga-inicial/netmax-carga-inicial-tipo-serie.sql
@carga-inicial/netmax-carga-inicial-tipo-cuenta.sql
@carga-inicial/netmax-carga-inicial-status-programa.sql
commit;
```

```
Prompt =====
Prompt Cargando catalogos replicados en arcdb_s2
Prompt =====
connect netmax_bdd/netmax_bdd@arcdb_s2
delete from país;
delete from tipo_serie;
delete from tipo_cuenta;
delete from status_programa;
@carga-inicial/netmax-carga-inicial-pais.sql
@carga-inicial/netmax-carga-inicial-tipo-serie.sql
@carga-inicial/netmax-carga-inicial-tipo-cuenta.sql
@carga-inicial/netmax-carga-inicial-status-programa.sql
commit;

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

1.2.2. 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 incluyendo tablas con datos BLOB.
- Se asume que la carpeta carga-inicial se encuentra en el mismo directorio donde se encuentran todos los scripts del proyecto. De lo contrario se deberá actualizar el siguiente script, revisar cuidadosamente.

Ejemplo:

```
--@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

Prompt =>Copiando archivos pdf y videos al directorio /tmp/bdd/proyecto-final
Prompt =>Creando directorio y actualizando permisos
!rm -rf /tmp/bdd/proyecto-final
!mkdir -p /tmp/bdd/proyecto-final
!cp carga-inicial/pdfs/sample*.pdf /tmp/bdd/proyecto-final
!cp carga-inicial/trailers/sample*.mp4 /tmp/bdd/proyecto-final
!chmod 777 /tmp/bdd/proyecto-final/sample*

Prompt =====
Prompt Listo para Iniciar ?
Prompt =====
Pause Presionar Enter para Iniciar, Ctrl-C para cancelar
Prompt Insertando datos ....

begin

@carga-inicial/netmax-carga-inicial-usuario.sql
@carga-inicial/netmax-carga-inicial-programa-documental.sql
@carga-inicial/netmax-carga-inicial-programa-pelicula.sql
@carga-inicial/netmax-carga-inicial-programa-serie.sql
@carga-inicial/netmax-carga-inicial-historico-status-prog.sql
@carga-inicial/netmax-carga-inicial-documental.sql
@carga-inicial/netmax-carga-inicial-serie.sql
--este script contiene carga de datos BLOB
@carga-inicial/netmax-carga-inicial-pelicula.sql
@carga-inicial/netmax-carga-inicial-playlist.sql
--este script contiene carga de datos BLOB
@carga-inicial/netmax-carga-inicial-archivo-programa.sql

dbms_output.put_line('Carga de datos completa!');
```

```

commit;

exception
when others then
dbms_output.put_line('Errores detectados al realizar la carga');
dbms_output.put_line('Se hara rollback');
rollback;
raise;
end;
/

Prompt Carga de datos exitosa!.
exit

```

1.2.3. Validación de la carga de datos.

- Ejecutar el script (se encuentra junto con el archivo zip) s-08-netmax-presentacion-4.sql Este archivo realizará algunas consultas para mostrar la cantidad de datos cargados en el paso anterior.

1.2.4. Eliminación de datos.

- Una vez que se han validado los datos, se procede con la eliminación. Crear un script s-08-netmax-presentacion-5.sql Este archivo se encargará de eliminar **todos** los registros de la BDD. Notar que todas las instrucciones delete deberán estar incluidas en una transacción distribuida para garantizar integridad.

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 ...

begin

    --completar

dbms_output.put_line('Elimimacion concluida, haciendo commit');
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

```

1.2.5. Validación de eliminación de datos.

Finalmente, ejecutar el script (Se encuentra junto con el archivo zip) s-08-netmax-presentacion-6.sql. El script realizará algunas acciones para validar que la eliminación de los datos fue exitosa.