

PROYECTO FINAL  
BDD empleada para administrar el negocio de la empresa NetMAX – PARTE 2

### 1.1. CONFIGURACIÓN DE SERVICIOS DE RED.

El siguiente paso a desarrollar es la creación de los objetos de base de datos necesarios en todos los sitios.

#### 1.1.1. Configuración del archivo `/etc/hosts`

- Recordar configurar este archivo para evitar problemas de conectividad. Se recomienda configurar ambas IPs en ambos nodos. La idea es que en los archivos de configuración de Oracle se empleará siempre el nombre de host (hostname) del sitio para evitar la configuración de IPs de forma directa.
- El único punto de configuración de IPs (mapeo hostname-IP) se realizará en este archivo.
- Como primer paso ejecutar el comando `ip a` para verificar que la IP obtenida sea la misma a configurar en este archivo.
- En este documento se consideran los siguientes nodos como ejemplo:

Nodo 1:

- Hostname: `jrc-ora-pc.fi.unam`
- IP: `192.168.0.4`

Nodo 2:

- Hostname: `arc-ora-pc.fi.unam`
- IP: `192.168.0.51`

- Las siguientes imágenes muestran la configuración del archivo:

- Configuración en `jrc-ora-pc`

```
#maquina local
127.0.0.1      jrc-ora-pc.fi.unam      jrc-ora-pc
192.168.0.4    jrc-ora-pc.fi.unam      jrc-ora-pc

#maquina remota
192.168.0.51   arc-ora-pc.fi.unam      arc-ora-pc
```

- Configuración en `arc-ora-pc`

```
#maquina local
127.0.0.1      arc-ora-pc.fi.unam      arc-ora-pc
192.168.0.51   arc-ora-pc.fi.unam      arc-ora-pc

#maquina remota
192.168.0.4    jrc-ora-pc.fi.unam      jrc-ora-pc
```

- Configurar el archivo `/etc/hosts` de forma similar para ambos nodos.
- No olvidar editar el archivo `/etc/hostname` para configurar el nombre del host.
- Verificar la comunicación bidireccional ejecutando en comando `ping`

#### Ejemplo:

- `ping jrc-ora-pc desde arc-ora-pc`
- `ping arc-ora-pc desde jrc-ora-pc`

**Nota:** Si las maquinas no están configuradas con una IP fija, cada vez que se prenda el equipo se deberá hacer una validación de IPs, y en caso de haber cambiado, se deberán actualizar las IPs en ambos nodos.

#### 1.1.2. Revisión del archivo `/etc/profile`.

- Revisar la configuración del archivo, verificar que todas las variables tengan valor adecuado, en especial `ORACLE_SID`, `ORACLE_HOSTNAME`.

Ejemplo:

Configuración para arc-ora-pc

```
#variables para Oracle 12c
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_HOSTNAME=arc-ora-pc.fi.unam
export ORACLE_UNQNAME=arcbd
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/12.1.0.2/db_1
export ORACLE_SID=arcbd
export TNS_ADMIN=$ORACLE_HOME/network/admin
export PATH=/usr/sbin:$PATH
export PATH=$ORACLE_HOME/bin:$PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
```

**1.1.3. Configuración del archivo listener.ora**

- En caso de ser necesario, actualizar el valor de “Hostname” en este archivo con base al valor asignado en /etc/hosts para ambos nodos.
- En el siguiente ejemplo se muestra la configuración del archivo en el nodo arc-ora-pc el cual apunta a localhost (127.0.0.1). Realizar lo mismo para el otro nodo.

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = arc-ora-pc.fi.unam)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )
```

- Levantar el listener en ambos nodos.

**1.1.4. Configuración del archivo tnsnames.ora**

Configurar el archivo tnsnames.ora en *ambos nodos*. El archivo deberá contener la definición de 5 servicios: contenedor ROOT local, 2 PBDs locales y 2 PBDs remotos. Emplear netmgr para completar la configuración en cada Nodo.

Ejemplo:

Archivo tnsnames.ora en arc-ora-pc

```
ARCBD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = arc-ora-pc.fi.unam)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = arcbd.fi.unam)
    )
  )

ARCBD_S1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = arc-ora-pc.fi.unam)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = arcbd_s1.fi.unam)
    )
  )

ARCBD_S2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = arc-ora-pc.fi.unam)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = arcbd_s2.fi.unam)
    )
  )
```

```

JRCBD_S1 =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP) (HOST = jrc-ora-pc.fi.unam) (PORT = 1521))
  )
  (CONNECT_DATA =
    (SERVICE_NAME = jrcbd_s1.fi.unam)
  )
)

JRCBD_S2 =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP) (HOST = jrc-ora-pc.fi.unam) (PORT = 1521))
  )
  (CONNECT_DATA =
    (SERVICE_NAME = jrcbd_s2.fi.unam)
  )
)

LISTENER_ARCBD =
(ADDRESS = (PROTOCOL = TCP) (HOST = arc-ora-pc.fi.unam) (PORT = 1521))

```

- Observar en la configuración anterior. El hostname y el nombre de servicio local se marca con azul, el host y el nombre de servicio remoto se marcan con rojo.
- Para verificar que la comunicación entre ambos nodos sea correcta, se puede emplear el comando `tnsping` (Ejecutar en ambos nodos).
- Nota: No olvidar levantar las instancias y revisar que las PDBs estén en status OPEN.

#### Ejemplo:

```

tnsping ARCBD_S1
tnsping ARCBD_S2
tnsping JRCBD_S1
tnsping JRCBD_S2

```

## 1.2. CREACIÓN DE OBJETOS EN LA BDD

Antes de iniciar con la creación y ejecución de scripts SQL, considerar las siguientes reglas:

- Los scripts NO deberán ser creados con el usuario `oracle` del sistema operativo.
- Los scripts NO deberán ser ejecutados con el usuario `oracle` del sistema operativo.
- Los scripts NO deberán ser ejecutados con el usuario `SYS` o `SYSTEM` a menos que se traten de sentencias que lo requieran.
- Todos los scripts deberán hacer uso de la sentencia `connect` para cambiarse de usuario o de PDB. Lo anterior implica que al entrar a `Sql *Plus` se debe emplear la instrucción `sqlplus /nolog`.

**Omitir cualquiera de estas reglas causará puntos menos en la calificación.**

### 1.2.1. Creación de usuarios.

Crear un solo script SQL `s-01-netmax-usuario.sql` El Script deberá conectarse a cada una de las 4 PDBs y realizar las siguientes acciones:

- Crear un usuario llamado `netmax_bdd`
- Otorgar privilegios necesarios. No otorgar mas de los permisos requeridos.

#### Ejemplo:

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación:
--@Descripción:     Creación de usuarios en las 4 PDBs

whenever sqlerror exit rollback;
--jrcbd_s1
prompt Conectando a jrcbd_s1 como sys
prompt Proporcione el password del usuario sys
connect sys@jrcbd_s1 as sysdba

create user netmax_bdd identified by netmax_bdd quota unlimited on users;
grant create session, create table, create sequence, create procedure,
create trigger, create synonym, create view, create database link, create type
to netmax_bdd;

-- jrcbd_s2
-- hacer lo mismo para las otra pdds.
. . .
Prompt Listo.
```

- No olvidar el encabezado a cada script como se muestra en la figura anterior.
- Ejecutar el script en cada nodo.
- Observar la instrucción `whenever sqlerror exit rollback;` Esta instrucción detendrá la ejecución del script al encontrar el primer error. En cada script se deberá incluir para facilitar la detección de errores.

### 1.2.2. Creación de ligas.

- Generar un solo script SQL llamado `s-02-netmax-ligas.sql`
- El script deberá contener las instrucciones necesarias para crear las ligas en todos los nodos. No es necesario especificar datos de autenticación ya el usuario existe en los 4 nodos.
- La comunicación debe ser bidireccional por lo que se requieren crear 3 ligas por nodo, 12 en total.
- Emplear el nombre global de cada PDB como nombre de la liga:

#### Ejemplo:

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación:
--@Descripción:     Creación de ligas en los 4 nodos.
clear screen
whenever sqlerror exit rollback;

--Ligas en la pdb jrcbd_s1
Prompt Creando ligas en jrcbd_s1
connect netmax_bdd/netmax_bdd@jrcbd_s1
-- PDB local
create database link jrcbd_s2.fi.unam using 'JRCBD_S2';
--PDB remotas
create database link arcdb_s1.fi.unam using 'ARCBD_S1';
Create database link arcdb_s2.fi.unam using 'ARCBD_S2';

--Ligas en la pdb jrcbd_s2
Prompt Creando ligas en jrcbd_s2
connect netmax_bdd/netmax_bdd@jrcbd_s2

. . .

--Ligas en la pdb arcdb_s1

. . .

--Ligas en la pdb arcdb_s2

. . .
Prompt Listo!
```

### 1.2.3. Creación de fragmentos.

El siguiente punto es la creación de todas las tablas que representan los fragmentos en cada sitio. Para ello, generar un archivo por cada PDB llamado s-03-netmax-<pdb>-ddl.sql. Ejemplo: s-03-netmax-jrc-s1-ddl.sql. Considerar los siguientes puntos.

- Se recomienda emplear Er-Studio para generar el código SQL.
- Hacer copias del modelo global y realizar los cambios correspondientes empleando la tabla de fragmentos generada en el documento anterior
- Las tablas se deberán crear en orden de dependencias.
- Cada tabla deberá contener la definición de sus restricciones (restricciones de llave primaria, restricciones de integridad: NOT NULL, restricciones de referencia).
- Debido al diseño de fragmentación, algunas restricciones de referencia incluidas en el script anterior no se podrán verificar de manera automática. Las únicas referencias a conservar son las llaves foráneas que provienen de catálogos y las fragmentaciones horizontales derivadas.
- Para identificar las restricciones que se deben **conservar**, revisar las reglas de fragmentación, y llenar la siguiente tabla.
- Las restricciones de referencia que no pueden conservarse deberán ser eliminadas desde ER -Studio conservando únicamente el campo en la tabla hija.
  - Para eliminar una restricción de referencia en ER-Studio sin eliminar el campo en la tabla hija, seleccionar la relación, presionar “Supr” o en su defecto hacer clic derecho y seleccionar “Delete Relationship”.
  - Aparecerá una ventana que solicita la confirmación de la eliminación.
  - Observar que el campo “Make Foreign Keys Native” debe ser habilitado para que el campo se conserve en la tabla hija y solo se elimine la restricción de referencia.



- Crear un solo script sql llamado s-03-netmax-main-ddl.sql. El script deberá conectarse a cada PDB e invocar a los 4 archivos creados anteriormente:

Ejemplo:

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación:
--@Descripción:    Creación de fragmentos en los 4 nodos
clear screen
whenever sqlerror exit rollback;

prompt =====
prompt Creando fragmentos para jrcbd_s1
prompt =====
connect netmax_bdd/netmax_bdd@jrcbd_s1
@s-03-netmax-jrc-s1-ddl.sql

prompt =====
prompt Creando fragmentos para jrcbd_s2
prompt =====
connect netmax_bdd/netmax_bdd@jrcbd_s2
@s-03-netmax-jrc-s2-ddl.sql

prompt =====
prompt Creando fragmentos para arcdb_s1
prompt =====
connect netmax_bdd/netmax_bdd@arcdb_s1
@s-03-netmax-arc-s1-ddl.sql

prompt =====
prompt Creando fragmentos para arcdb_s2
prompt =====
connect netmax_bdd/netmax_bdd@arcdb_s2
@s-03-netmax-arc-s2-ddl.sql

Prompt Listo!
```

- Ejecutar el script.
- Continuar con la siguiente parte del proyecto.