



UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

Facultad de ingeniería



Computo Paralelo Y Distribuido
Docente: Jose Saul De Lira Miramontes

Actividad: Proyecto Final

Integrantes:

Danna Maribel Corral Salcedo 358147

Erick Fernando Nevarez Avila 357664

Gerardo Esteban Jurado Carrera 273880

PASOS DE LA PRACTICA

Primero instalamos Oracle en nuestro dispositivo y vamos a crear un esquema que va a corresponder a nuestra región A, por lo cual nos conectamos como administrados a ORACLE.

Nueva / Seleccionar Conexión a Base de Datos

Nombre de Cone...	Detalles de Cone...
Banco_A	banco_a@//local...
BDHeterogenea	system@//localh...

Name: BDHeterogenea

Tipo de Base de Datos: Oracle

Información de usuario: Usuario de Proxy

Tipo de autenticación: Por defecto

Usuario: system

Rol: valor por defecto

Contraseña:

Guardar Contraseña: ☐

Tipo de Conexión: Básico

Detalles: Avanzado

Nombre del Host: localhost

Puerto: 1521

☐ SID

☒ Nombre del Servicio: XEPDB1

Estado: Correcto

Ayuda Guardar Borrar Probar Conectar Cancelar

El usuario siempre es system, pero la contraseña dependerá de la que hayamos asignado en la instalación de ORACLE.

Vamos a ejecutar los siguientes querrys en orden en la hora de trabajo de system:

```
CREATE USER banco_a IDENTIFIED BY banco123;

GRANT CONNECT, RESOURCE TO banco_a;

ALTER USER banco_a DEFAULT TABLESPACE USERS;
ALTER USER banco_a QUOTA UNLIMITED ON USERS;
```

```
User BANCO_A creado.

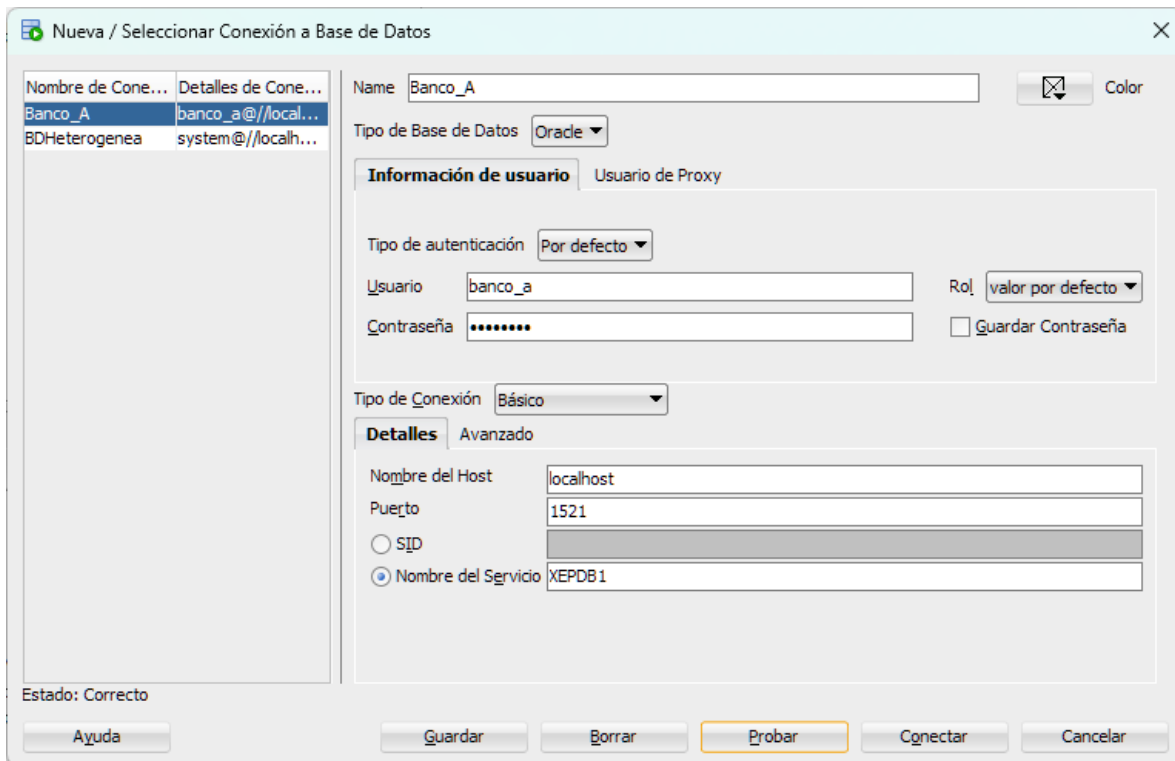
Grant correcto.

User BANCO_A alterado.

User BANCO_A alterado.
```

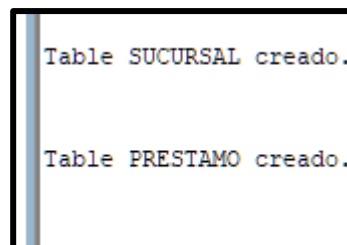
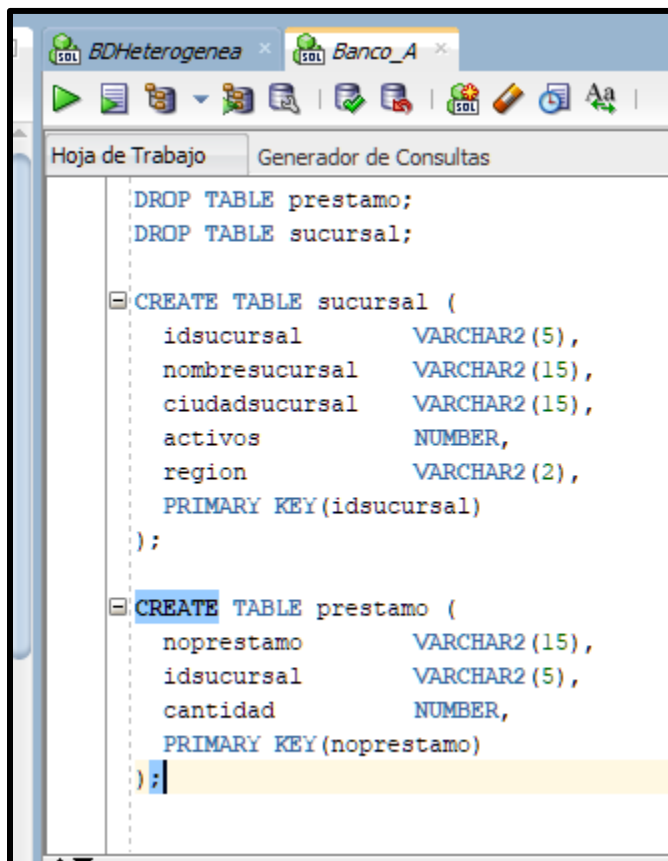
Nos debe de dar los mensajes de Banco_a creado, grant correcto y user banco_a alterado (2 veces).

Ahora creamos una conexión al Banco_A, de la misma forma que con system, solo que la contraseña ahora es la de banco123 y el usuario banco_a



Como observamos al darle Probar, aparece el estado como correcto.

Nos conectamos a dicha conexión como banco_a y vamos a ejecutar en la hora de trabajo los siguientes comandos:



Nos va a aparecer un error de que no encuentra la tabla, ya que quiere eliminarla con el comando DROP TABLE, pero después nos aparecerá el mensaje: Table Sucursal / Préstamo creado.

Después de haber creado las tablas, vamos a insertar datos en cada una de las tablas:

```

INSERT INTO sucursal VALUES (1, 'Downtown', 'Brooklyn', 900000, 'A');
INSERT INTO sucursal VALUES (2, 'Redwood', 'Palo Alto', 2100000, 'A');
INSERT INTO sucursal VALUES (3, 'Perryridge', 'Horseneck', 1700000, 'A');
INSERT INTO sucursal VALUES (4, 'Mianus', 'Horseneck', 400200, 'A');

INSERT INTO prestamo VALUES (17, 1, 1000);
INSERT INTO prestamo VALUES (23, 2, 2000);
INSERT INTO prestamo VALUES (15, 3, 1500);
INSERT INTO prestamo VALUES (14, 1, 1500);
INSERT INTO prestamo VALUES (93, 4, 500);
INSERT INTO prestamo VALUES (16, 3, 1300);

```

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

1 fila insertadas.

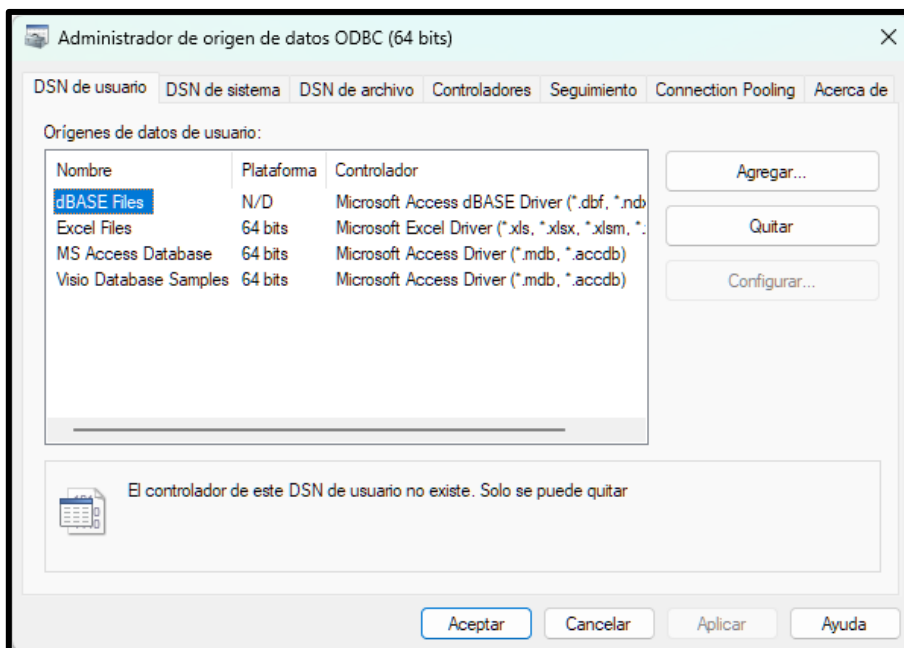
1 fila insertadas.

Nos va a dar el mensaje de que fueron insertadas diversas filas.

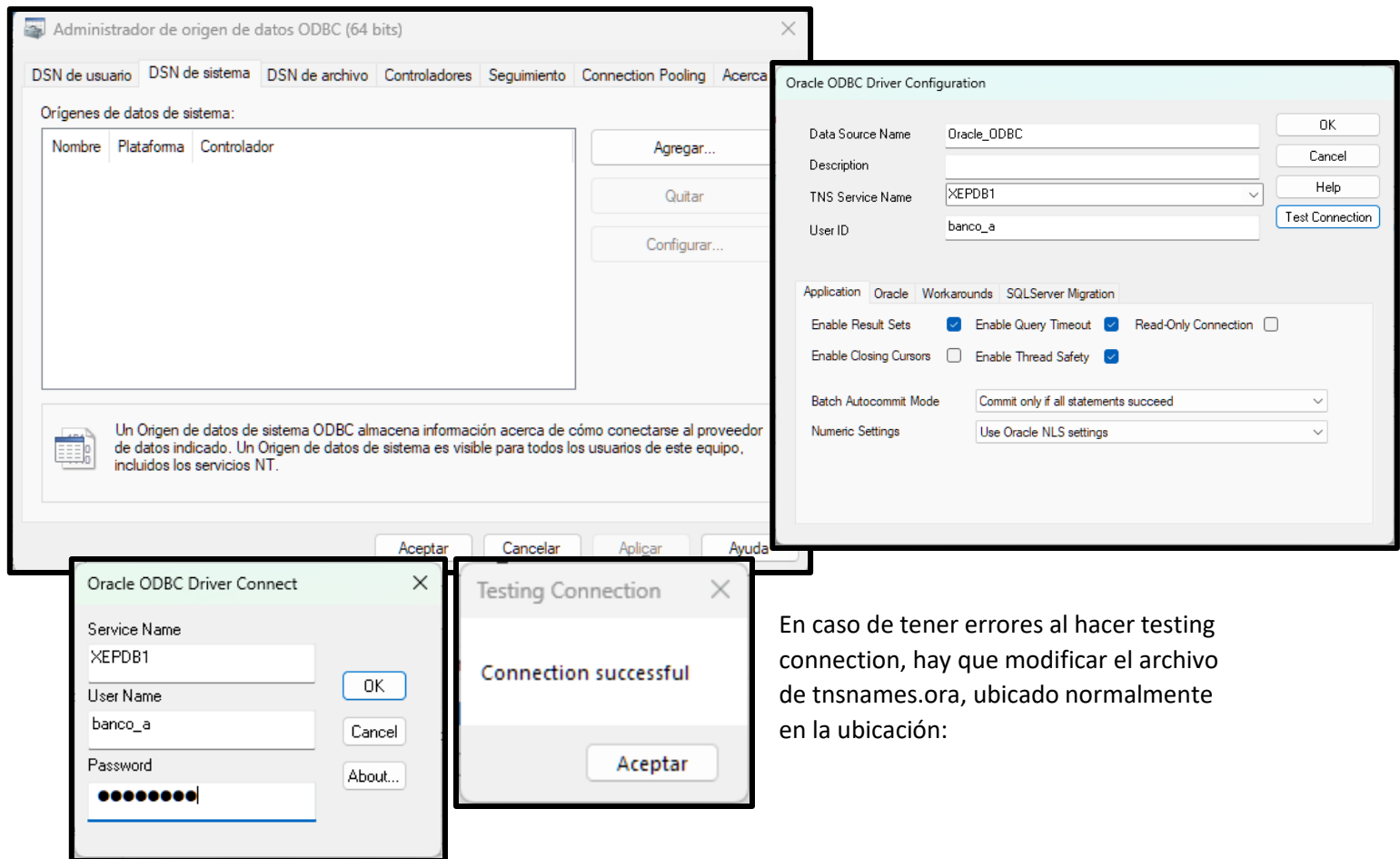
Por último, confirmamos las inserciones realizando un select a ambas tablas:

</

Ya tenemos lista nuestra base de datos en ORACLE, ahora vamos a asignar un ODBC (Open DataBase Connectivity (Conectividad Abierta a Bases de Datos)) para la conexión entre la base de datos de SQL server y la de Oracle, para ello debemos de descargar [ODBC](#) y realizar su sencilla instalación wizard, después de instalarlo, abrimos la aplicación:



Seleccionamos en DSN de sistema, damos en agregar y seleccionamos el mas parecido a OraDB21Home1, ya que la instalación puede variar. Damos click, rellenamos la información con los datos de conexión al usuario banco_a, damos en test Conection, vaciamos la contraseña banco123 y debemos de



En caso de tener errores al hacer testing connection, hay que modificar el archivo de tnsnames.ora, ubicado normalmente en la ubicación:

<C:\app\USUARIO\product\21c\homes\OraDB21Home1\network\admin\tnsnames.ora>

Si no observamos que tenemos la conexión XEADB1, La vamos a escribir quedando algo como lo siguiente:

```
# tnsnames.ora Network Configuration File: C:\app\erick\product\21c\homes\OraDB21Home1\NETWORK\ADMIN\tnsnames.ora
# Generated by Oracle configuration tools.

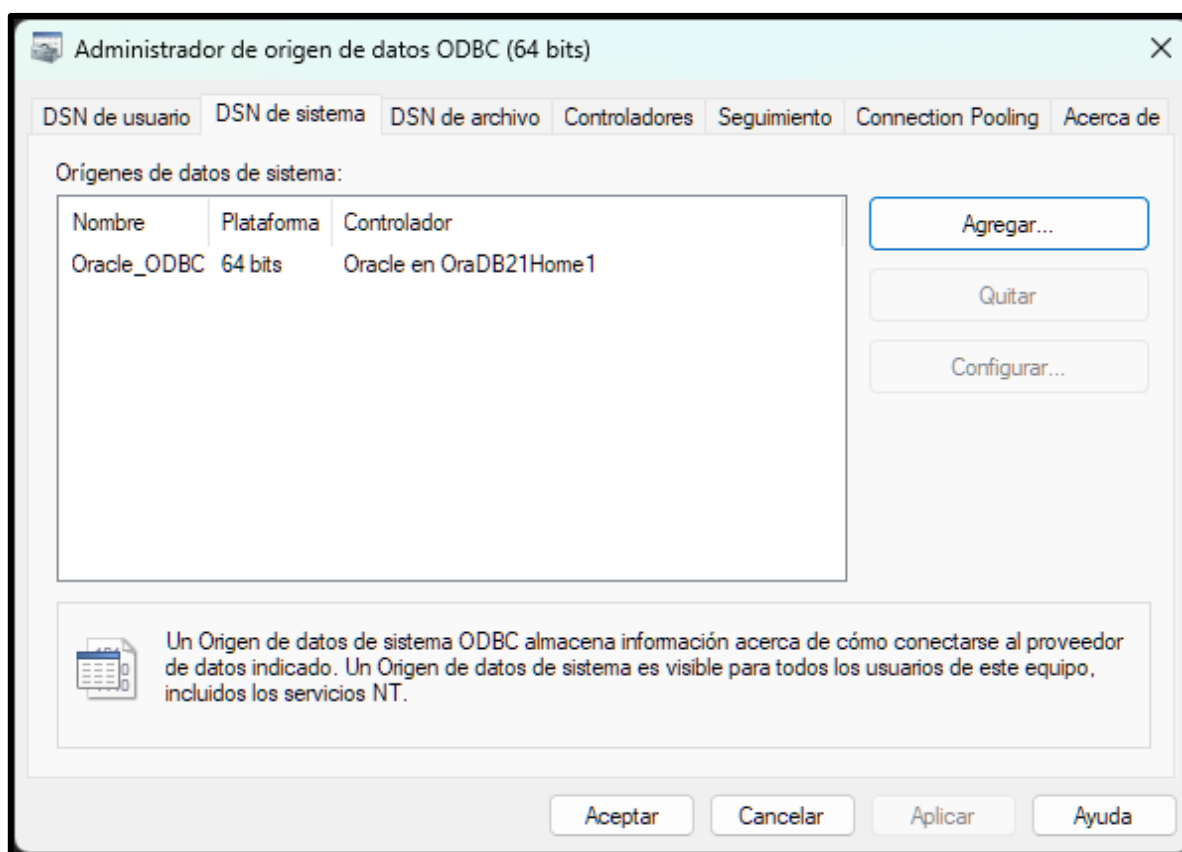
XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ErickPCComputer)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

LISTENER_XE =
  (ADDRESS = (PROTOCOL = TCP)(HOST = ErickPCComputer)(PORT = 1521))

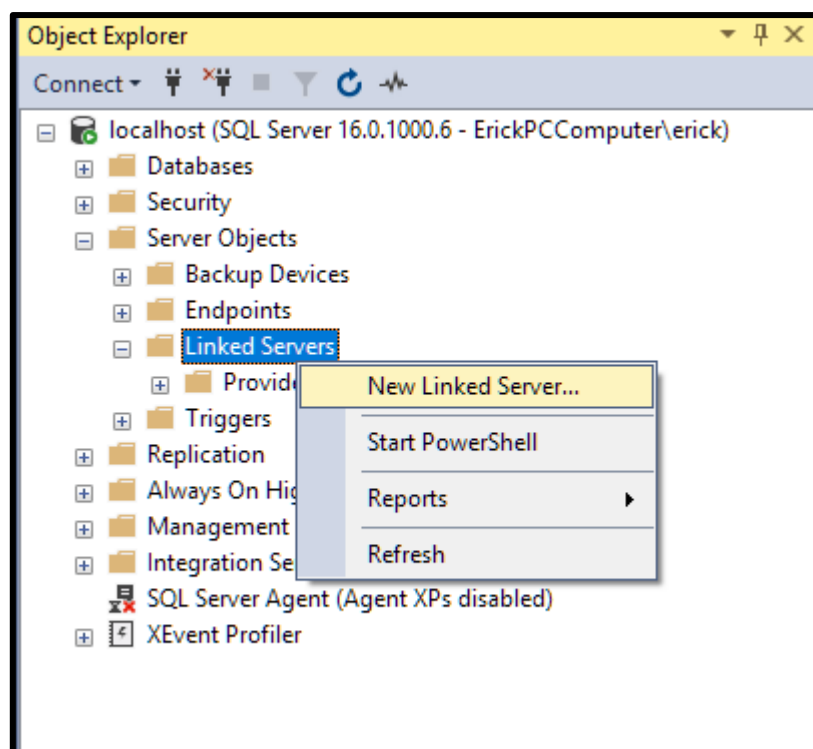
ORACLR_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
    (CONNECT_DATA =
      (SID = CLRExtProc)
      (PRESENTATION = RO)
    )
  )

XEADB1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = XEADB1)
    )
  )
```

Damos a OK y vamos a tener ahora un ODBC a Oracle que podremos utilizar en SQL server:



Ahora en Sql server, vamos a entrar a nuestra conexión local con Windows authentication:



Damos click derecho en linked servers

Y vamos a vaciar la siguiente información:

Linked server: OracleLinkedServer

Provider: Microsoft OLE DB Provider for ODBC Drivers

Product name: Oracle

Data source: Oracle_ODBC

Y en la parte de security:

Damos click en: Be made using this security context y vaciamos el login:

banco_a
banco123

Linked Server Properties - ORACLE_LINK

Select a page: General, Security, Server Options

Script Help

Linked server: ORACLE_LINK

Server type:
☐ SQL Server
☒ Other data source

Provider: Microsoft OLE DB Provider for ODBC Drivers

Product name: Oracle

Data source: Oracle_ODBC

Provider string:

Location

Catalog

Connection
Server: ErickPCComputer
Connection: ERICKPCCOMPUTER\erick
[View connection properties](#)

Progress
Ready

OK Cancel

Linked Server Properties - ORACLE_LINK

Select a page: General, Security, Server Options

Script Help

Local server login to remote server login mappings:

Local Login	Impersonate	Remote User	Remote Password
-------------	-------------	-------------	-----------------

Add Remove

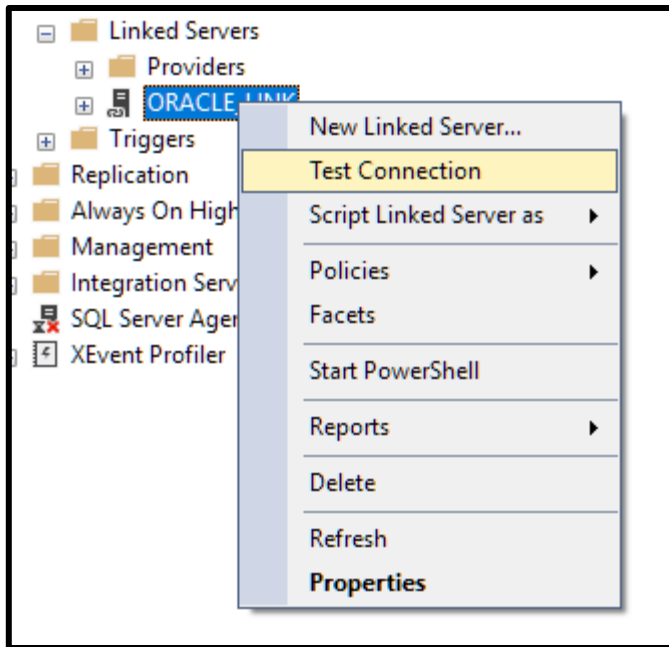
For a login not defined in the list above, connections will:

☐ Not be made
☐ Be made without using a security context
☐ Be made using the login's current security context
☒ Be made using this security context:

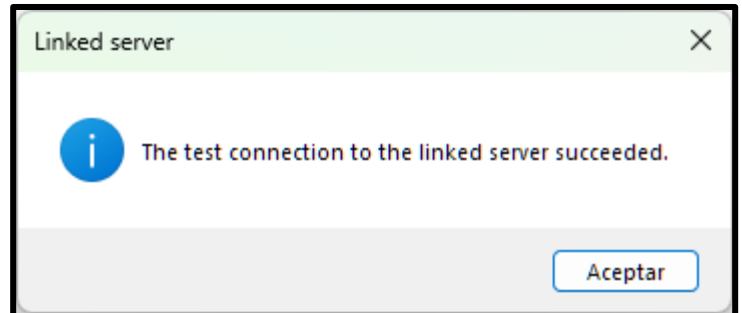
Remote login: banco_a

With password: *****

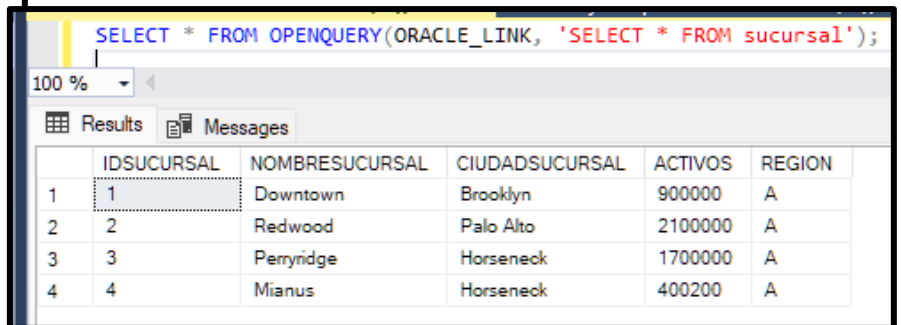
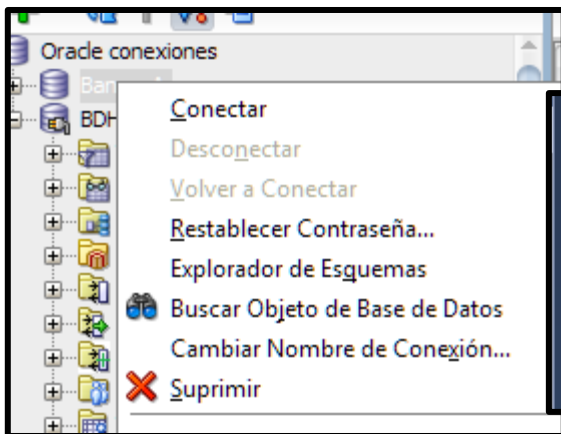
OK Cancel



Después damos click derecho a la conexión y test Connection:



Ahora si podemos consultar nuestra tabla remotamente, para ello vamos a ejecutar el siguiente query de ejemplo, ASEGURANDONOS QUE LA CONEXIÓN EN SQL DEVELOPER ESTE DESCONECTADA:



Nuestro linked con la otra base de datos es funcional.

Ahora vamos a crear 3 elementos muy importantes para la ejecución de nuestra aplicación:

- 1 trigger entre la Base de datos del nodo B y la base de datos de replicación ósea el nodo C
- Los 2 procedimientos almacenados, 1 para insert, select y delete en la base de datos B, y otro para insert, select y delete de la base de datos C:
 - o El insert y delete del nodo B condiciona el del nodo C implícitamente con el trigger
 - o Para los insert y delete del nodo A si debemos de definir que todo insert y delete se haga en ambas bases de datos.

Comenzamos con la creación de nuestras bases de datos con el siguiente query:


```

CREATE DATABASE BDHeterogenea;
GO

CREATE DATABASE BDHeterogenea_RegionB;
GO

USE BDHeterogenea;
GO

CREATE TABLE sucursal
(
    idsucursal int PRIMARY KEY,
    nombresucursal VARCHAR(15),
    ciudadsucursal VARCHAR(15),
    activos INT,
    region VARCHAR(2)
);

CREATE TABLE prestamo
(
    noprestamo int PRIMARY KEY,
    idsucursal int,
    cantidad INT,
);
GO

USE BDHeterogenea_RegionB;
GO

CREATE TABLE sucursal
(
    idsucursal int PRIMARY KEY,
    nombresucursal VARCHAR(15),
    ciudadsucursal VARCHAR(15),
    activos INT,
    region VARCHAR(2)
);

CREATE TABLE prestamo
(
    noprestamo int PRIMARY KEY,
    idsucursal int,
    cantidad INT,
);
GO

```

Estamos creando las mismas 2 tablas en ambas bases de datos, siendo BDHeterogenea nuestro nodo C, BDHeterogenea_RegionB nuestro nodo B.

Vamos a ejecutar todos los procedimientos almacenados de:

- Creación de triggers región B a región C
- Procedimiento almacenado select a prestamos
- Procedimiento almacenado select a sucursales
- Procedimiento almacenado simulando trigger region A

Todos estos procedimientos almacenados nos van a funcionar para poder ejecutar en nuestra aplicación los comandos de insert y select en nuestras diversas regiones, ya sea de sucursal o préstamos.

Por último, vamos a ejecutar el query de inserción que nos permite verificar que todo esa funcionando correctamente.

```

USE BDHeterogenea;
GO

EXEC sp_InsertarSucursal 1, 'Centro', 'Monterrey', 150000, 'A';
EXEC sp_InsertarSucursal 2, 'Norte', 'Saltillo', 800000, 'A';
EXEC sp_InsertarSucursal 3, 'Sur', 'Veracruz', 1200000, 'A';
EXEC sp_InsertarSucursal 4, 'Este', 'Tampico', 950000, 'A';
EXEC sp_InsertarSucursal 5, 'Oeste', 'Hermosillo', 1100000, 'A';
EXEC sp_InsertarSucursal 6, 'Altos', 'Guadalajara', 1400000, 'A';
EXEC sp_InsertarSucursal 7, 'Pacífico', 'Mazatlán', 750000, 'A';
EXEC sp_InsertarSucursal 8, 'Laguna', 'Torreón', 920000, 'A';
EXEC sp_InsertarSucursal 9, 'Bajío', 'León', 1350000, 'A';
EXEC sp_InsertarSucursal 10, 'Centrosur', 'Cuernavaca', 980000, 'A';

EXEC sp_InsertarPrestamo 1, 1, 10000;
EXEC sp_InsertarPrestamo 2, 2, 5000;
EXEC sp_InsertarPrestamo 3, 3, 15000;
EXEC sp_InsertarPrestamo 4, 4, 8000;
EXEC sp_InsertarPrestamo 5, 5, 12000;
EXEC sp_InsertarPrestamo 6, 6, 3000;
EXEC sp_InsertarPrestamo 7, 7, 20000;
EXEC sp_InsertarPrestamo 8, 8, 18000;
EXEC sp_InsertarPrestamo 9, 9, 6000;
EXEC sp_InsertarPrestamo 10, 10, 25000;
GO

USE BDHeterogenea_RegionB;
GO

INSERT INTO Sucursal VALUES ('Noroeste', 'Tijuana', 1600000, 'B');
INSERT INTO Sucursal VALUES ('SurEste', 'Cancún', 1450000, 'B');
INSERT INTO Sucursal VALUES ('CentroNte', 'San Luis', 1250000, 'B');
INSERT INTO Sucursal VALUES ('Sierra', 'Chihuahua', 900000, 'B');
INSERT INTO Sucursal VALUES ('Mixteca', 'Oaxaca', 700000, 'B');

INSERT INTO Prestamo VALUES (11, 13000);
INSERT INTO Prestamo VALUES (12, 17000);
INSERT INTO Prestamo VALUES (13, 11000);
INSERT INTO Prestamo VALUES (14, 9500);
INSERT INTO Prestamo VALUES (15, 10500);

```

Favor de visualizar el video de:

EXPLICACIONAPLICACION.MP4

Para conocer el funcionamiento interno de la aplicación conectada a toda base de datos.