



UNIVERSITÉ ABDELMALEK ESSAADI
FACULTÉ DES SCIENCES ET TECHNIQUES
DEPARTEMENT GENIE INFORMATIQUE
TANGER



MODULE : Administration des Bases de Données SQL SERVER & ORACLE

PROJET DE FIN DU MODULE

Réalisé par : **El Msaouri Tarik**

Encadré par : EL Amrani Chaker

Année Universitaire 2020-2021

Sommaire

Introduction :.....	3
Virtual Machine: WINDOWS SERVER 2019	4
Configurer le service Active Directory :.....	4
Création d'utilisateur de la Domaine	6
Intégration dans le domaine :	10
Se connecter à SQL SERVER 2019 avec l'Authentification Windows	10
Création de la base donnée : BD1	11
Machine Virtuelle - LINUX POSTGRESQL.....	11
Intigration dans le domaine :	12
Autoriser Windows server à communiquer avec notre serveur postgre.....	14
Machine physique WINDOWS 10.....	15
Intégration dans le domaine tarikelsaouri.local [192.168.1.20]	15
Se connecter à SQL SERVER 2019 avec l'Authentification mixte	16
Création de la base donnée : BD3	16
Activation du protocole de la communication TCP/IP dans le port 1433, dans Le SQL Server Manager	17
System oracle: Oracle 11g Express Edition.....	18
Autoriser l'utilisation du port TCP 1521 qui sera utilisé par l'instance de SQL Server de notre machine dans Le Windows Firewall :	19
Création de la table dans Postgresql:.....	19
Liaison des serveurs Dans SQL Server Management Studio & création de la vue partitionnée :.....	20
Vérification de la liaison de la sql server avec les autres serveurs.....	25
Création de la vue partitionne basée sur les 4 bases de données :	25
Exportation des données de BD1 vers BD2, BD3, BD4, Fichier Texte en utilisant SSIS.....	26
Creation d'un Data flow Task dans onglet Conrol flow	26
Configuration des Components de SSIS :	26
BD1 SQL SERVER WINDOWS SERVER	26
BD3 SQL SERVER dans Machine physique Windows 10	28

BD2 POSTGRESQL dans Vbox Ubuntu	28
BD4 ORACLE dans Machine physique Windows 10	30
Vérification que les données ont reçues	31
Exportation de la vue partitionnée vers une table vide créée dans les bases de données destinations	33
Conclusion :	33

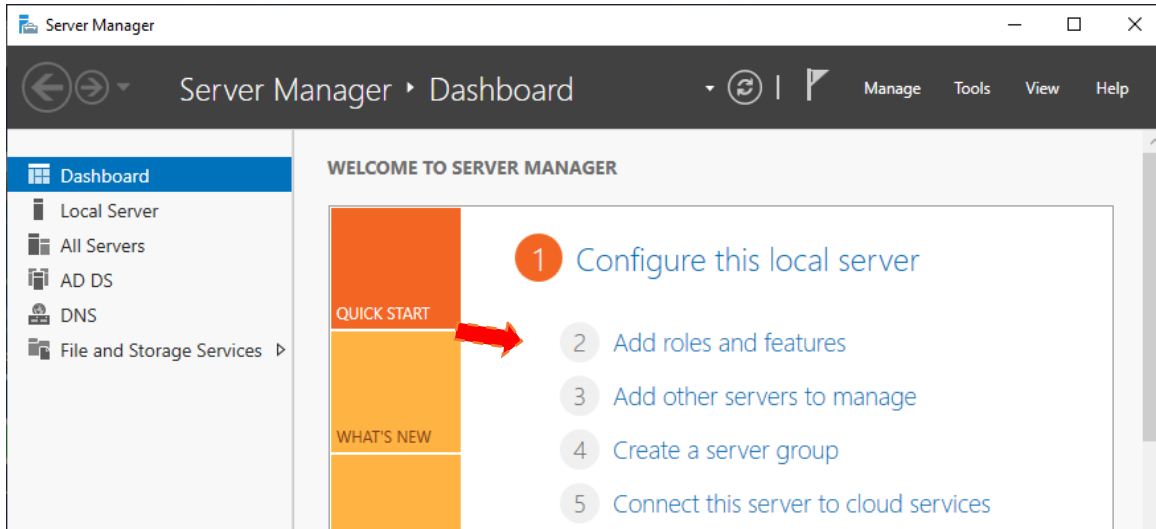
Introduction :

Le but de ce projet est de réaliser un système qui peut transférer et conserver les données d'une base de données SQL server et on transfère ces données vers plusieurs systèmes. Comme linux, Windows server et vers plusieurs bases de données comme POSTGRE, MS ACCESS, ORACLE, SQL SERVER. Et vers même des fichiers textes.

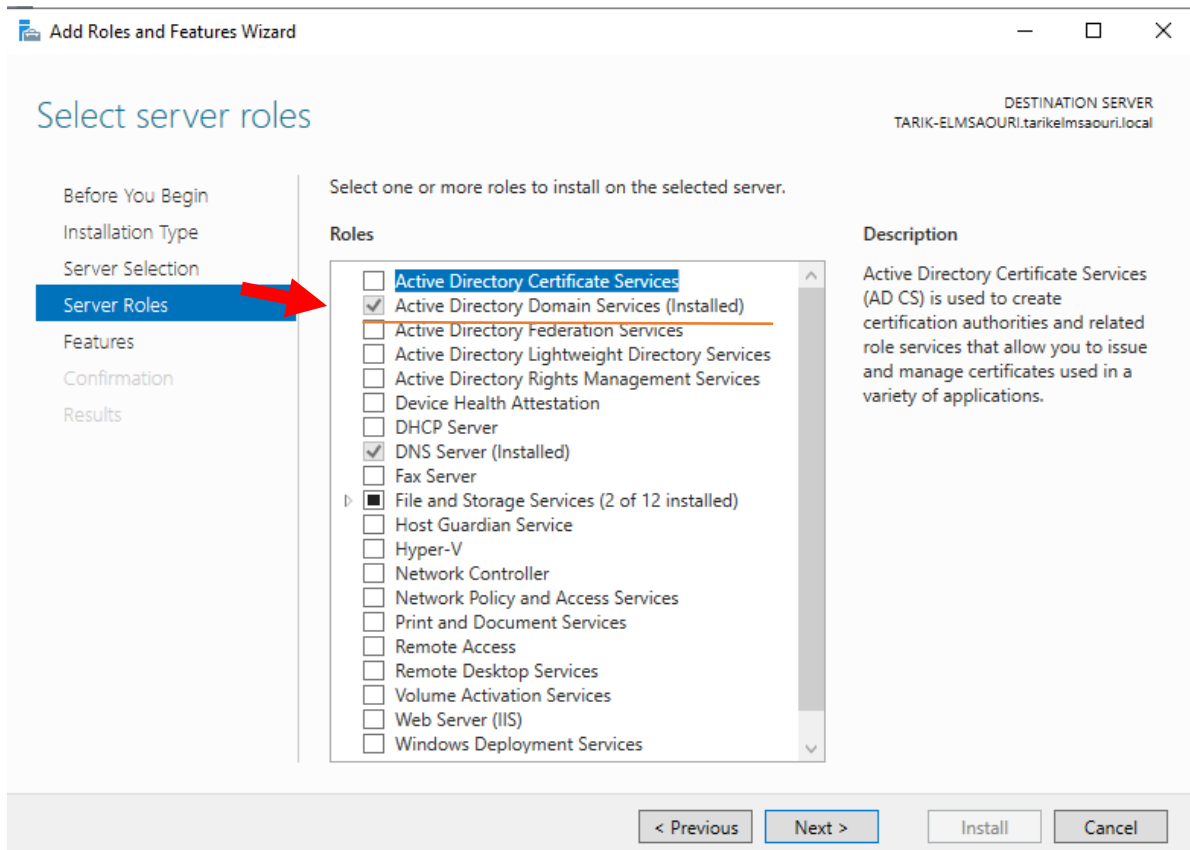
Virtual Machine: WINDOWS SERVER 2019

Configurer le service Active Directory :

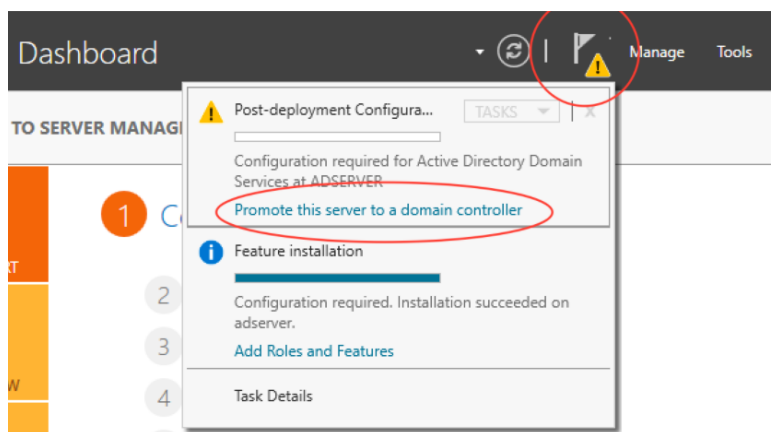
On ouvre le server manager et on clique sur Add roles and features



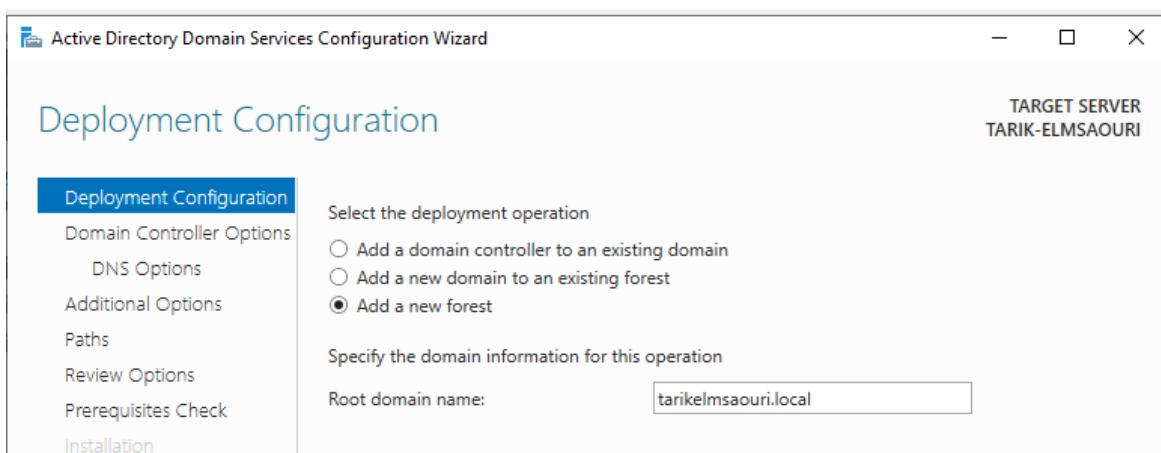
- On continue les etapes de configuration en cliquant



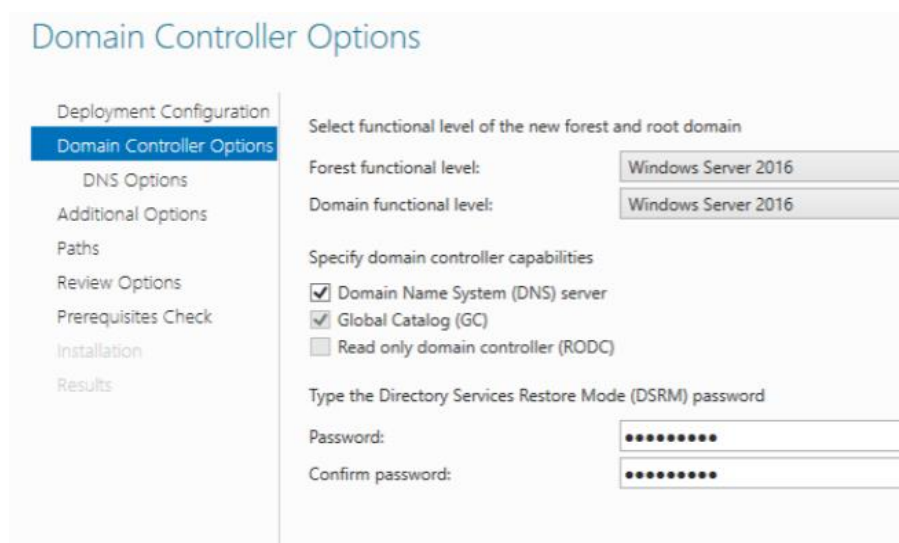
- Après l'activation de Directory, On clique sur promote this server to a domain controller pour créer notre domaine



- On choisit le nom de domaine : dans mon cas, j'ai choisi : **tarikelmsaouri.local**



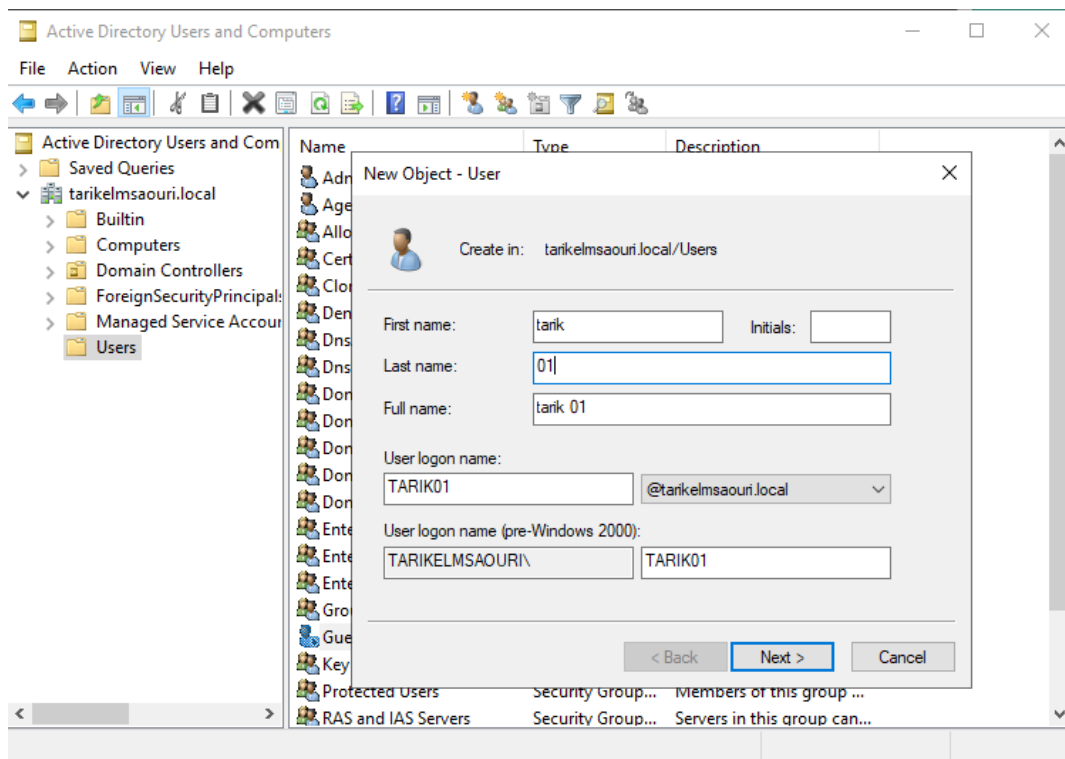
-Puis on choisit le mot de passe



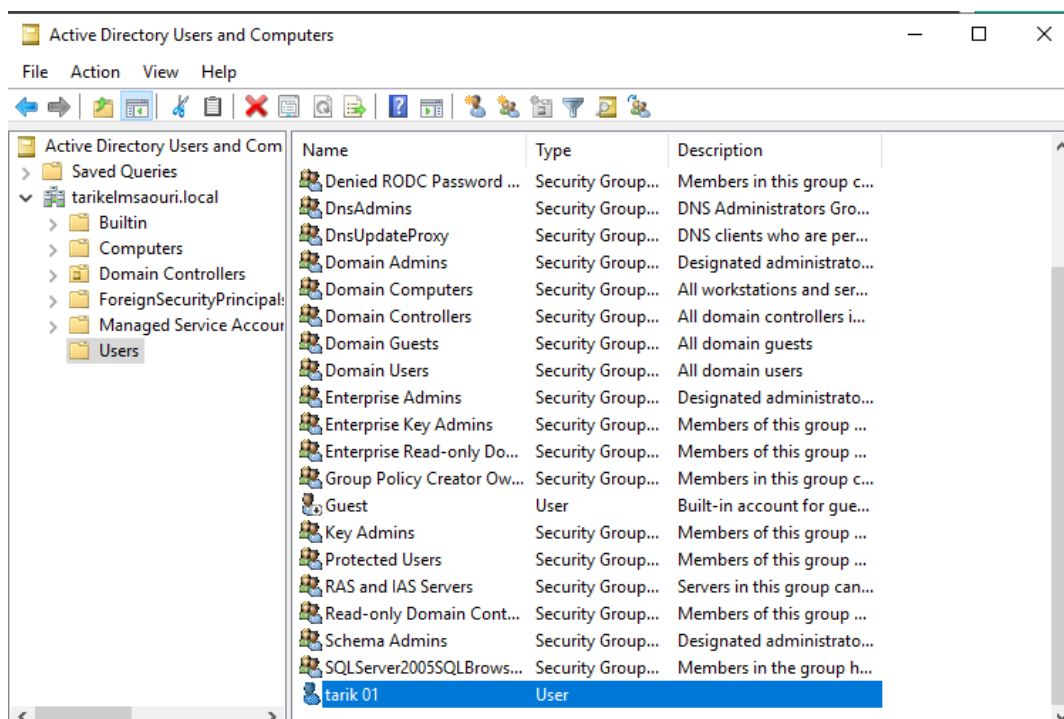
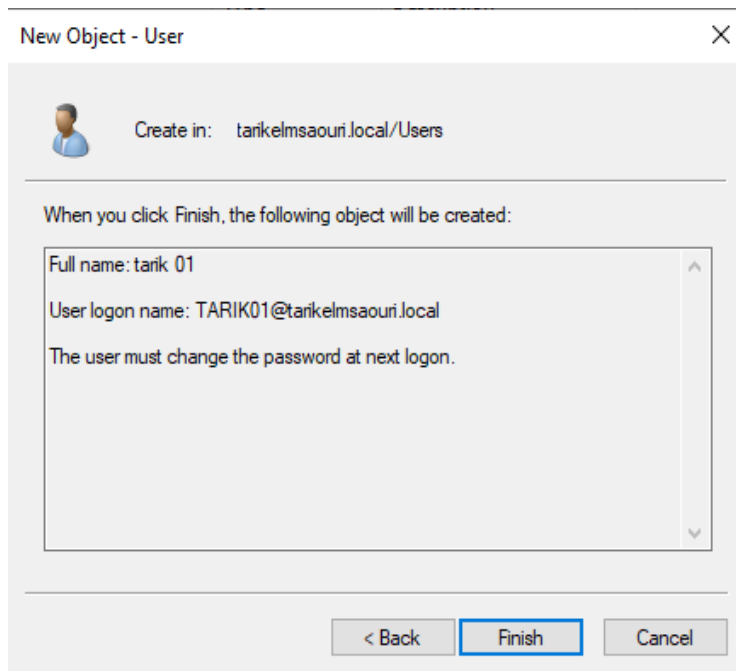


- On clique **next** pour tous les etapes suivants jusqu'à la création d nouveau domaine finir.

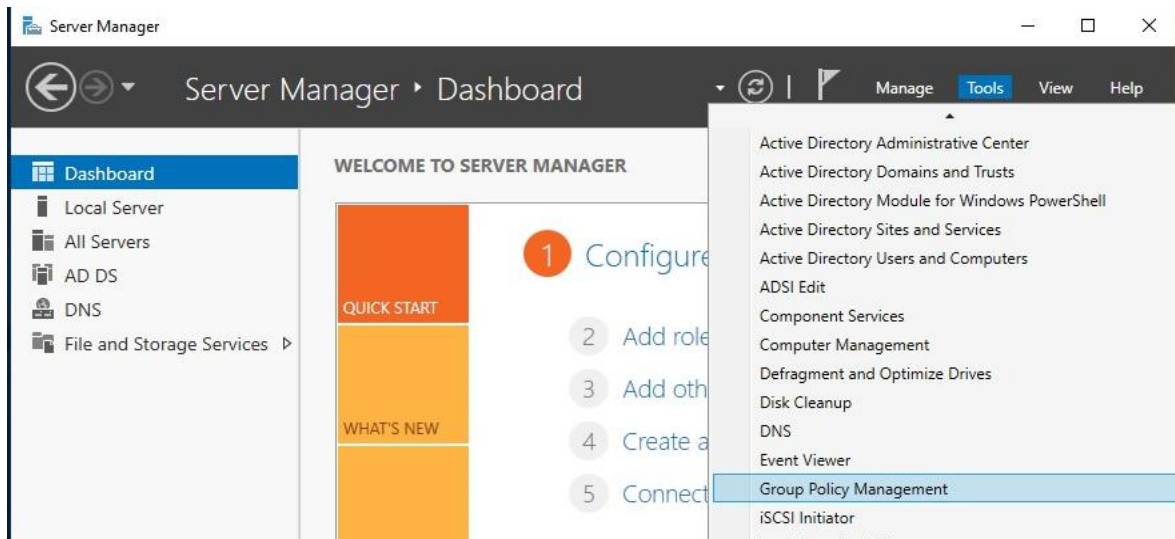
Création d'utilisateur de la Domaine



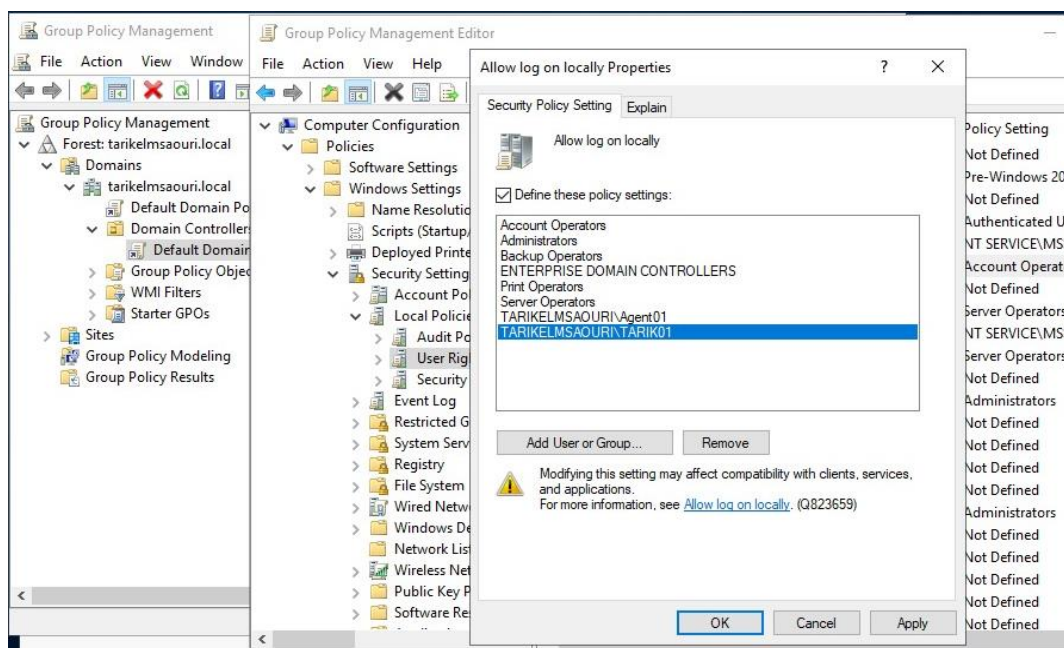
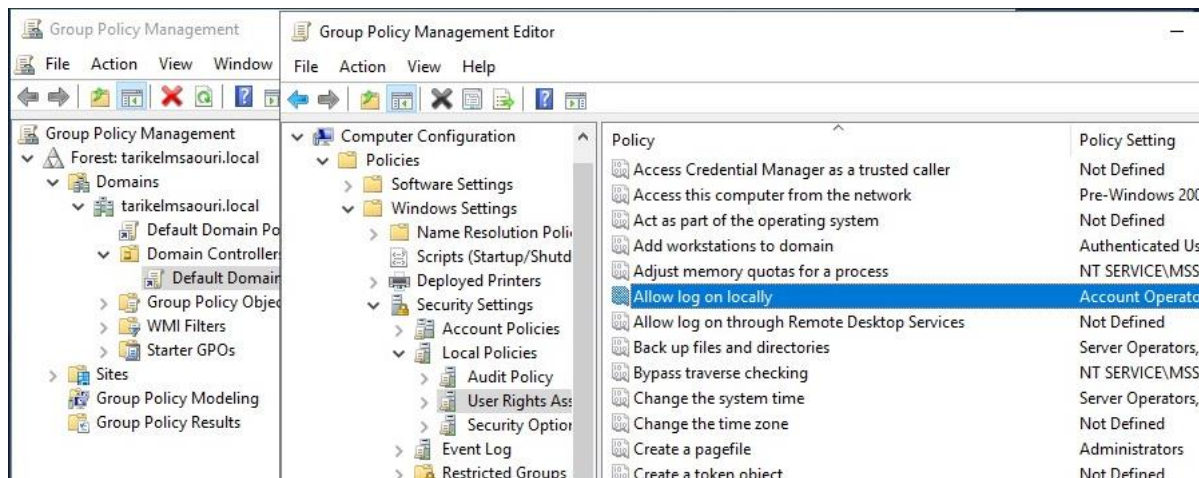
Les informations liées à la création d'un nouveau User



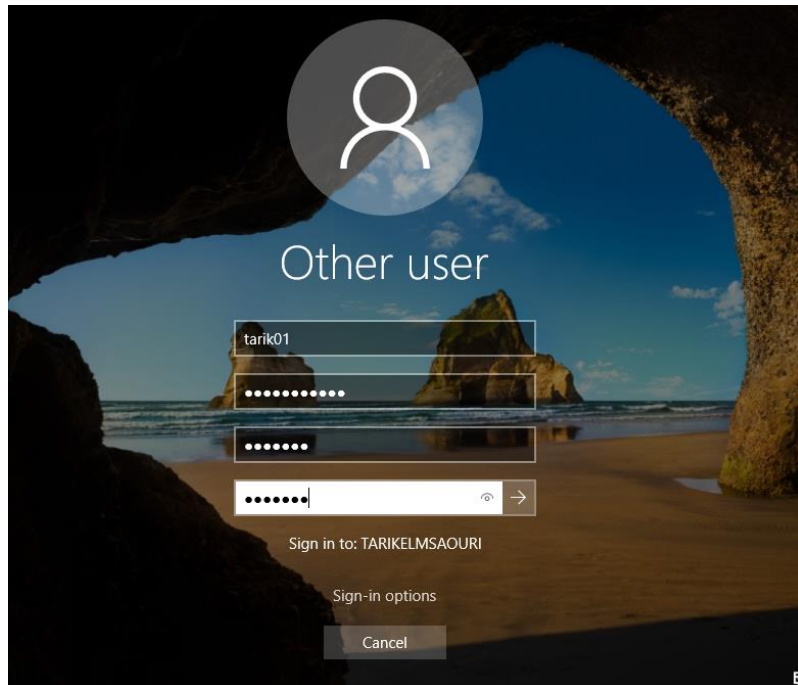
- Maintenant que l'utilisateur est créé, on a besoin de lui accorder la permission de se connecter à notre serveur. Pour cela, on va accéder à l'onglet Group Policy Management.



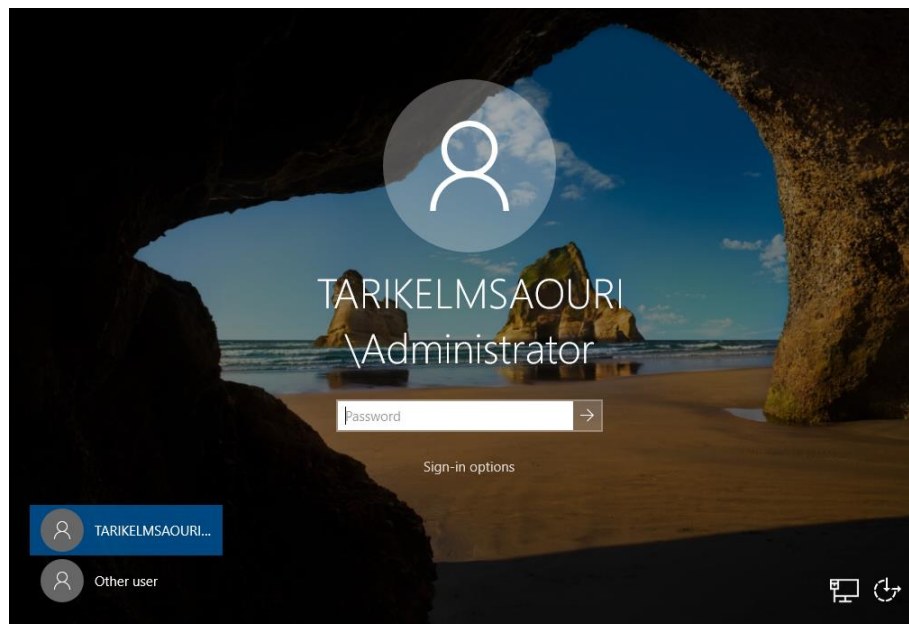
Allow log on locally



- Après avoir fini la configuration du domaine on redémarre notre machine. Maintenant on peut s'authentifier avec deux modes de connexion qui appartiennent au domaine (nouvel utilisateur tarik01) ou à partir de l'utilisateur Administrateur.

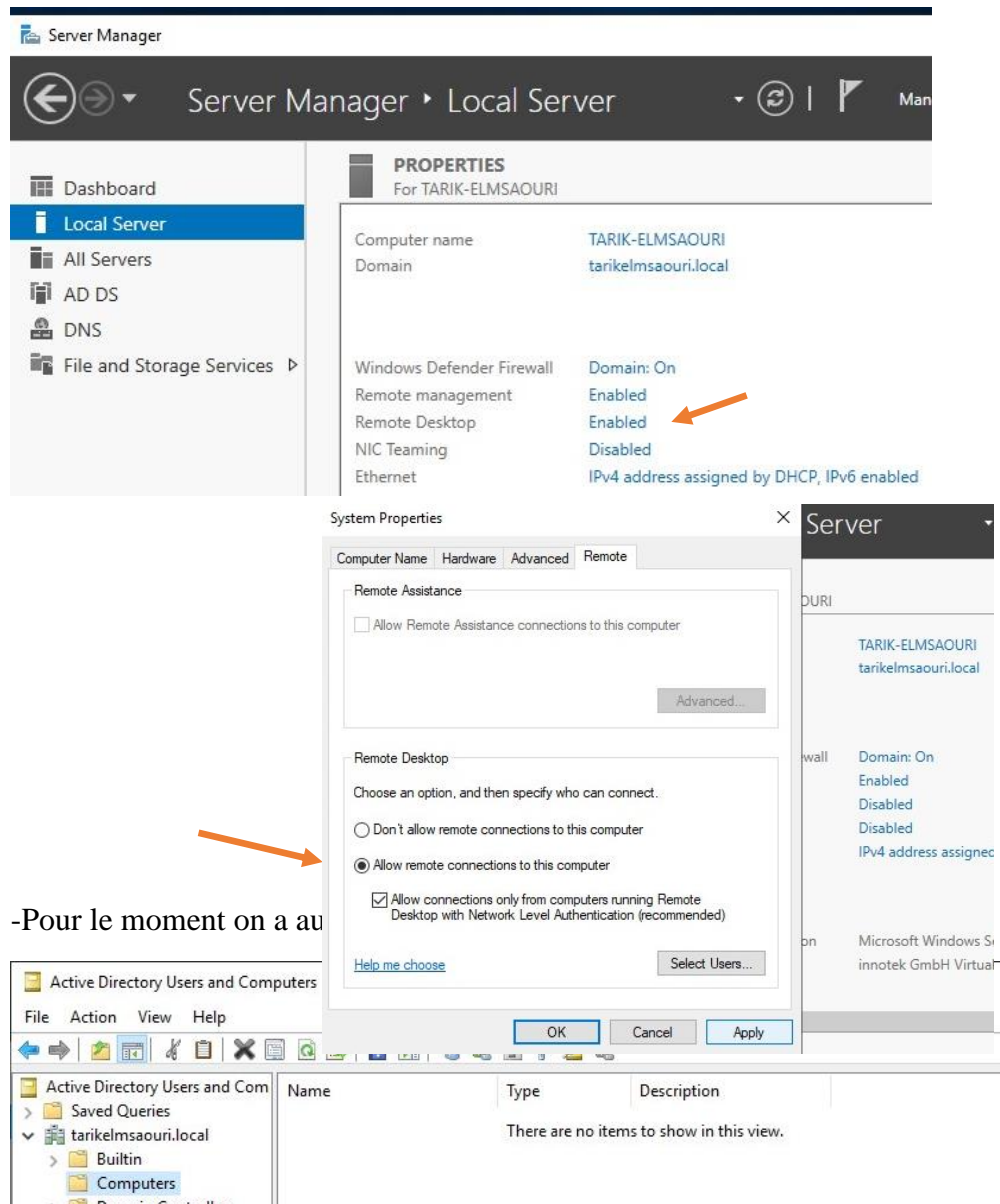


- Pour mon cas je vais travailler avec l'utilisateur Administrateur pendant tout ce projet



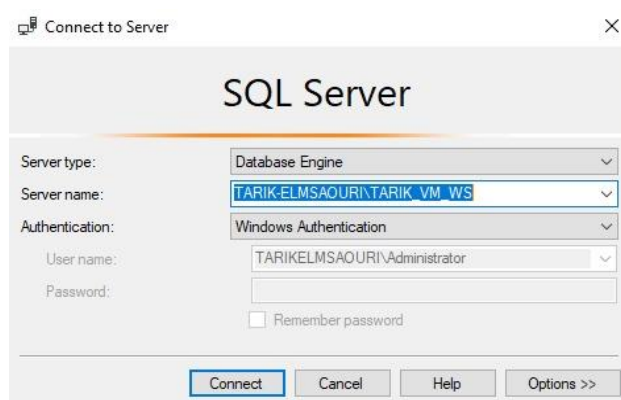
Intégration dans le domaine :

-Pour interagir avec les autres machines et établir les serveurs lies on active l'option Remote Desktop

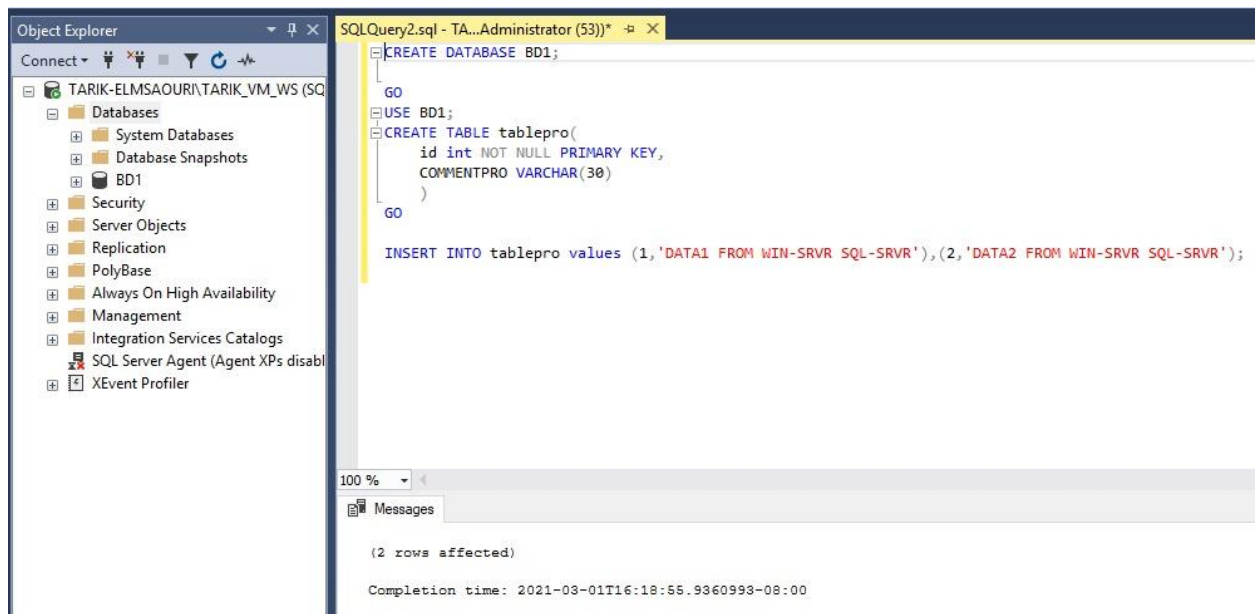


-Pour le moment on a au

Se connecter à SQL SERVER 2019 avec l'Authentification Windows



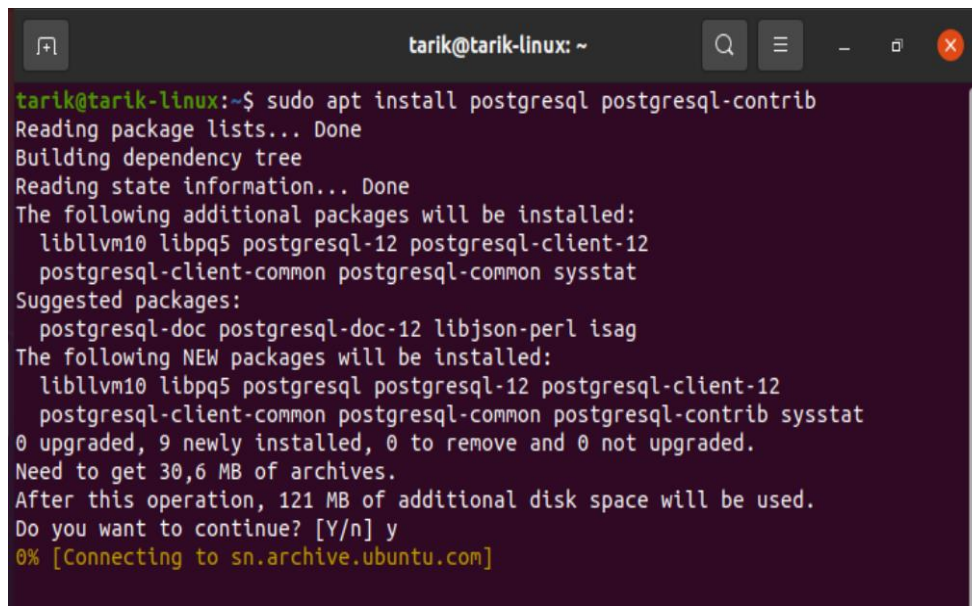
Création de la base donnée : BD1



Machine Virtuelle - LINUX POSTGRESQL

-Installation et configuration de PostgreSQL :

On Install par la commande : **sudo apt install postgresql postgres**



- Création du nom de connexion :

On crée l'utilisateur **tarik** qui va se connecter à la base donnée BD2.

On lui affecter le rôle createuser qui doit avoir le même nom d'utilisateur qui va se connecter au bdd postgre :

```
Terminal fee 19 13:30
tarik@tarik-linux: ~
tarik@tarik-linux:~$ sudo -u postgres psql
[sudo] password for tarik:
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# create user tarik WITH PASSWORD 'post@gres'
postgres=# createuser --interactive tarik
postgres=# createdb DB2
postgres=#
```

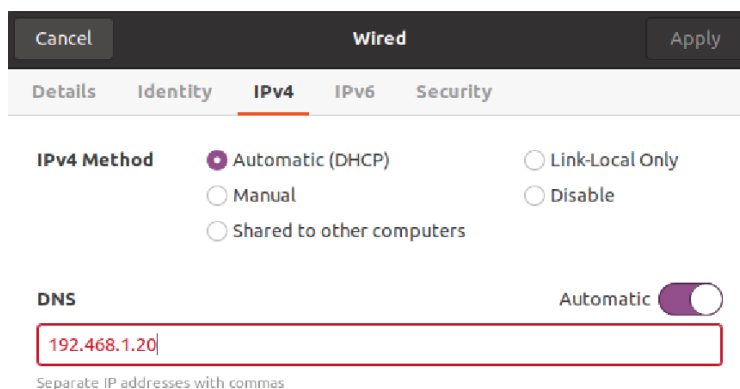
- Création de la base donnée BD2
- Creation et l'insertion dans la table 'tablepro'.

```
tarik@tarik-linux: ~
tarik@tarik-linux:~$ sudo -u postgres createdb BD2
tarik@tarik-linux:~$ sudo -i -u tarik
tarik@tarik-linux:~$ psql -d BD2
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))
Type "help" for help.

BD2=# \conninfo
You are connected to database "BD2" as user "tarik" via socket in "/var/run/postgresql" at port "5432".
BD2=# create table tablepro(ID INT PRIMARY KEY not null,COMMENTPRO TEXT);
CREATE TABLE
BD2=# INSERT INTO tablepro values (6,'Vbox Postgresql 1');
INSERT 0 1
BD2=# INSERT INTO tablepro values (7,'Vbox Postgresql 2');
INSERT 0 1
BD2=#
```

Intigration dans le domaine :

-Affecter au DNS l'adresse IP de notre Windows Server

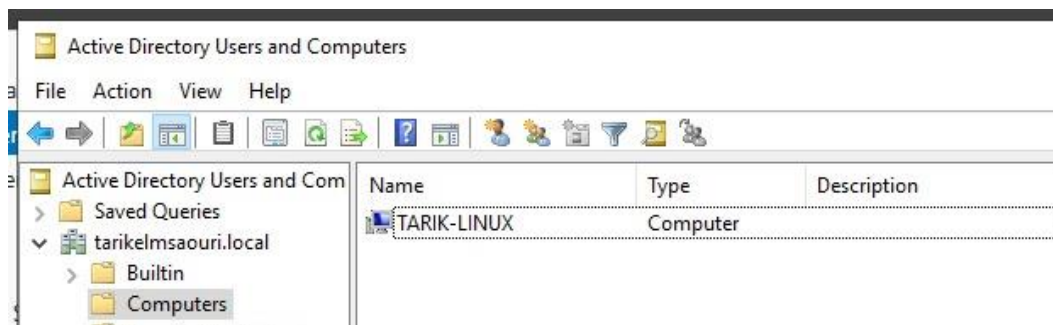


-On découvrir les informations sur notre domaine on utilisant realm

```
tarik@tarik-linux: ~  
tarik@tarik-linux:~$ realm discover tarikelmsaouri.local  
tarikelmsaouri.local  
type: kerberos  
realm-name: TARIKELMSAOURI.LOCAL  
domain-name: tarikelmsaouri.local  
configured: no  
server-software: active-directory  
client-software: sssd  
required-package: sssd-tools  
required-package: sssd  
required-package: libnss-sss  
required-package: libpam-sss  
required-package: adcli  
required-package: samba-common-bin  
tarik@tarik-linux:~$
```

On intigre au domaine par la commande :

```
tarik@tarik-linux:~$ realm join tarikelmsaouri.local  
Password for Administrator:  
* Installing necessary packages: sssd-tools adcli sssd libnss-sss libpam-sss
```



On vérifie si on est bien connecté à notre domaine :

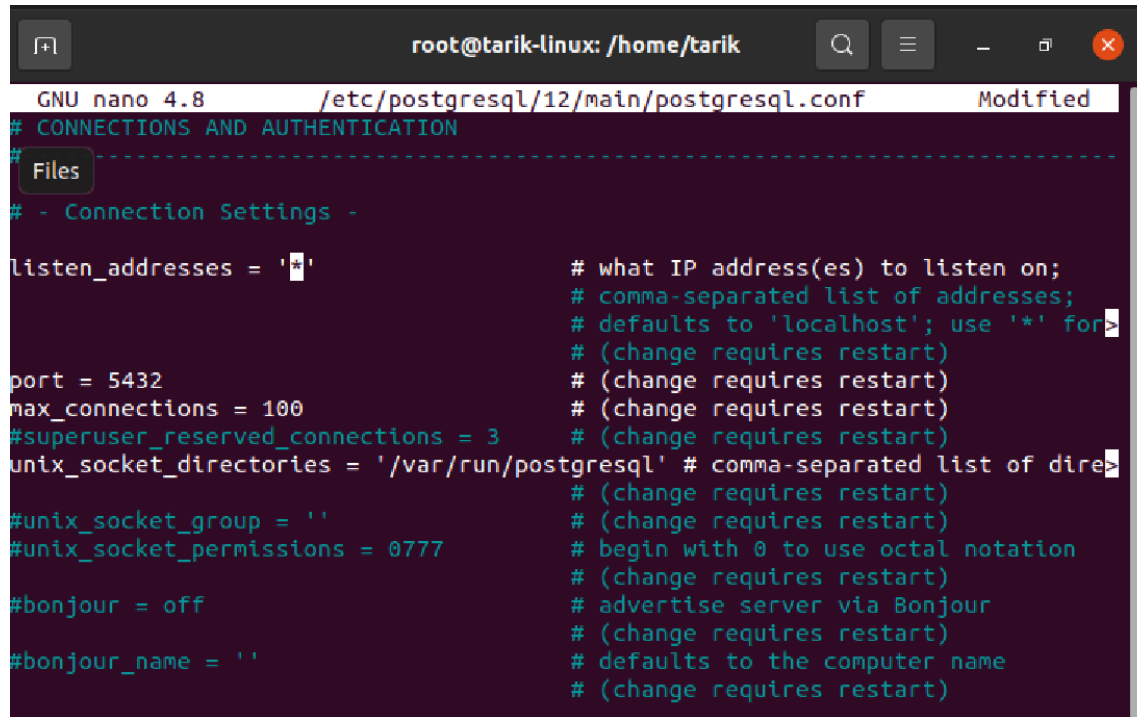
```
tarik@tarik-linux: ~  
tarik@tarik-linux:~$ ping 192.168.1.20  
PING 192.168.1.20 (192.168.1.20) 56(84) bytes of data.  
4 bytes from 192.168.1.20: icmp_seq=1 ttl=128 time=1.21 ms  
4 bytes from 192.168.1.20: icmp_seq=2 ttl=128 time=0.540 ms  
4 bytes from 192.168.1.20: icmp_seq=3 ttl=128 time=0.872 ms  
4 bytes from 192.168.1.20: icmp_seq=4 ttl=128 time=0.559 ms  
4 bytes from 192.168.1.20: icmp_seq=5 ttl=128 time=1.25 ms  
4 bytes from 192.168.1.20: icmp_seq=6 ttl=128 time=0.547 ms
```


Autoriser Windows server à communiquer avec notre serveur postgre

On modifie le premier fichier avec la commande :

sudo nano /etc/postgresql/12/main/postgresql.conf

listen_addresses='*' veut dire qu'on va laisser le serveur postgre le droit d'écouter à tous les adresses du réseau



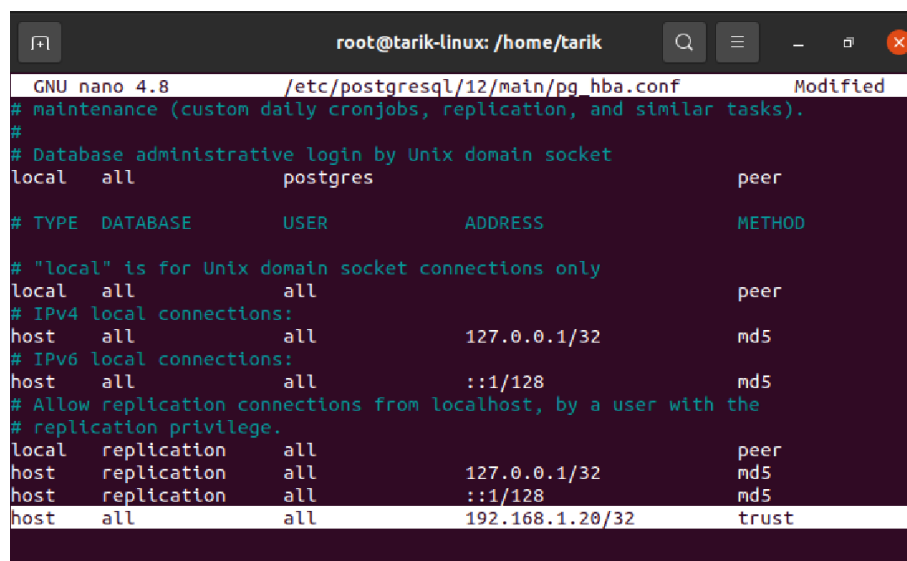
```
GNU nano 4.8 /etc/postgresql/12/main/postgresql.conf Modified
# CONNECTIONS AND AUTHENTICATION
# -----
# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for
                                # (change requires restart)
port = 5432                     # (change requires restart)
max_connections = 100           # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of dire
                                # (change requires restart)
#unix_socket_group = ''         # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
                                # (change requires restart)
#bonjour = off                  # advertise server via Bonjour
                                # (change requires restart)
#bonjour_name = ''              # defaults to the computer name
                                # (change requires restart)
```

-On modifie le deuxième fichier avec la commande :

sudo nano /etc/postgresql/12/main/pg_hba.conf

- On autoriser l'adresse 192.168.1.20 qui est l'adresse statique de notre machine windows server à communiquer et interagir avec notre serveur postgre



```
GNU nano 4.8 /etc/postgresql/12/main/pg_hba.conf Modified
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local all postgres peer

# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 md5
host replication all ::1/128 md5
host all all 192.168.1.20/32 trust
```

Machine physique WINDOWS 10

Intégration dans le domaine tarikelsaouri.local [192.168.1.20]

-On a l'adresse IP de windows server est :192.168.1.20

```
CA: Select Administrator: Command Prompt

C:\Users\Administrator>whoami
tarikelsaouri\administrator

C:\Users\Administrator>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : Home
    IPv6 Address. . . . . : fd0b:d9d8:9e3d:0:9daa:752f:9a98:b2fc
    Link-local IPv6 Address . . . . . : fe80::9daa:752f:9a98:b2fc%4
    IPv4 Address. . . . . : 192.168.1.20
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
```

-On vérifie si on est vraiment connecté à notre domaine :

```
CA: Command Prompt

Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

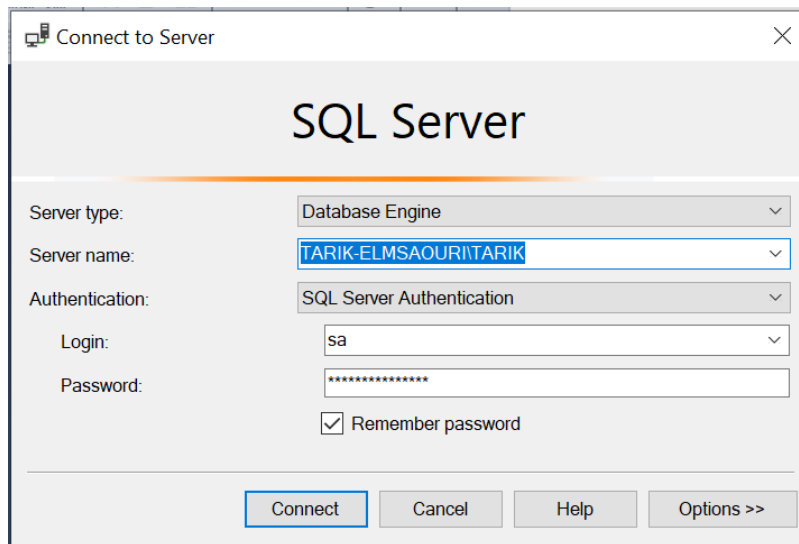
C:\Users\Tarik>ping 192.168.1.20

Pinging 192.168.1.20 with 32 bytes of data:
Reply from 192.168.1.20: bytes=32 time<1ms TTL=128
Reply from 192.168.1.20: bytes=32 time<1ms TTL=128
Reply from 192.168.1.20: bytes=32 time<1ms TTL=128
Reply from 192.168.1.20: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Tarik>
```

Se connecter à SQL SERVER 2019 avec l'Authentification mixte



Connect to Server

SQL Server

Server type: Database Engine

Server name: TARIK-ELMSAOURI\TARIK

Authentication: SQL Server Authentication

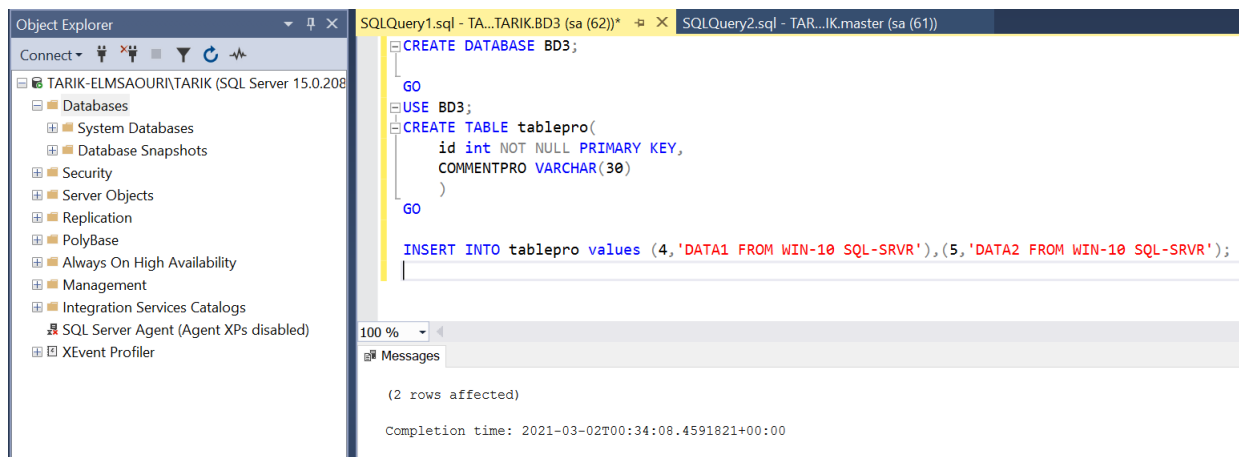
Login: sa

Password: *****

☒ Remember password

Connect Cancel Help Options >>

Création de la base donnée : BD3



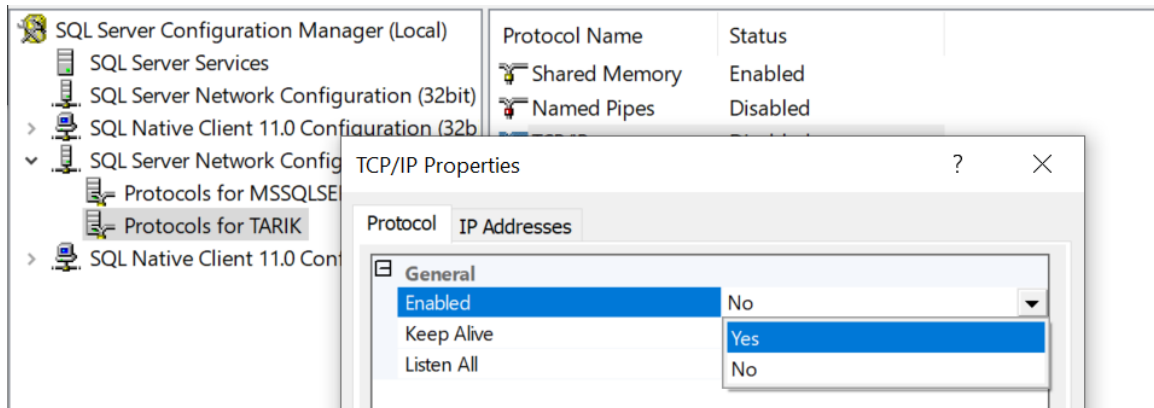
```
CREATE DATABASE BD3;
GO
USE BD3;
CREATE TABLE tablepro(
    id int NOT NULL PRIMARY KEY,
    COMMENTPRO VARCHAR(30)
)
GO
INSERT INTO tablepro values (4, 'DATA1 FROM WIN-10 SQL-SRVR'), (5, 'DATA2 FROM WIN-10 SQL-SRVR');
```

(2 rows affected)

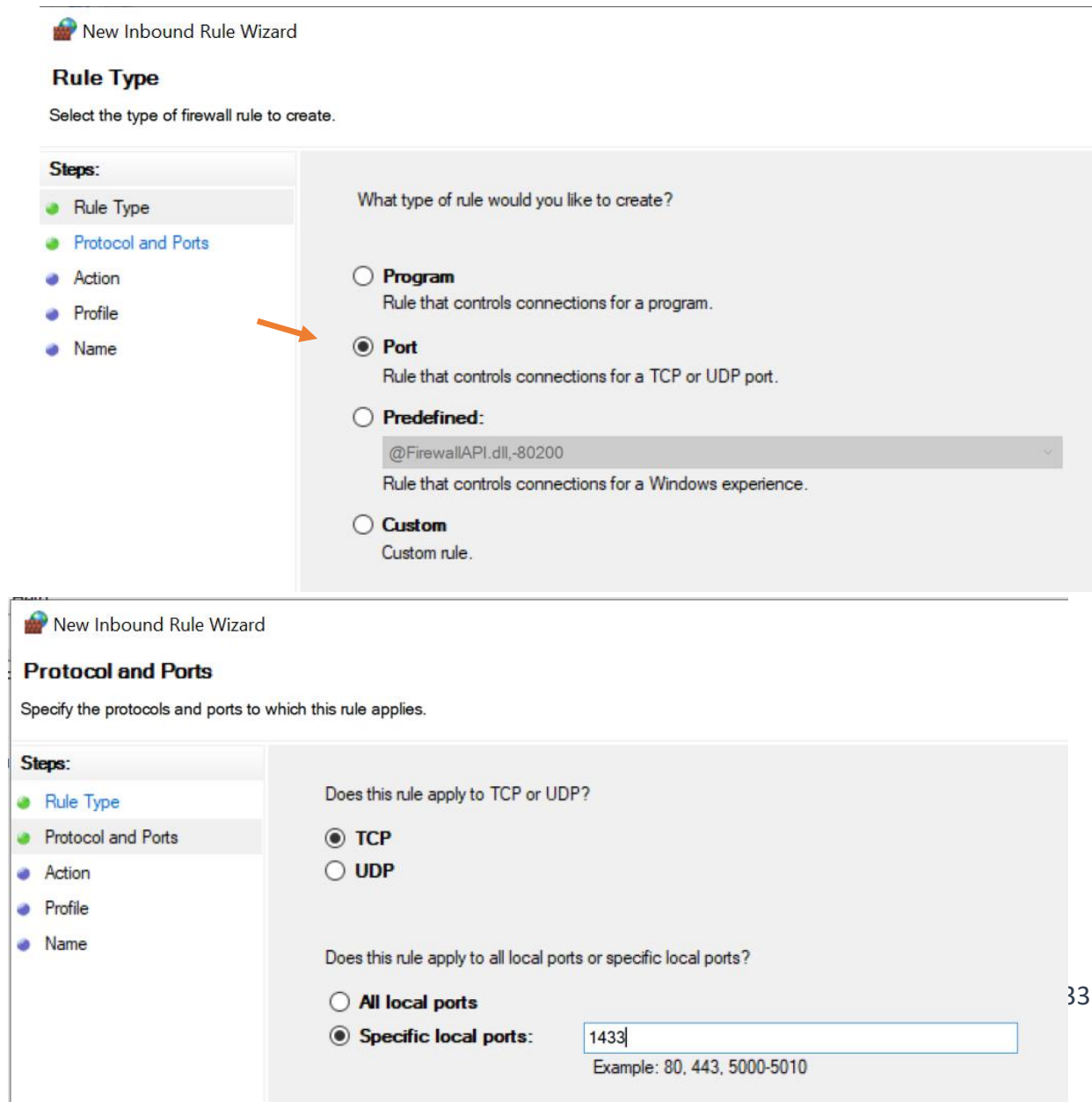
Completion time: 2021-03-02T00:34:08.4591821+00:00

Accès à SQL SERVER VIA Windows server

Activation du protocole de la communication TCP/IP dans le port 1433, dans Le SQL Server Manager



Autoriser l'utilisation du port TCP 1433 qui sera utilisé par l'instance de SQL Server de notre machine dans Le Windows Firewall :



New Inbound Rule Wizard

×

Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

Steps:

- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

What action should be taken when a connection matches the specified conditions?

☒ **Allow the connection**
 This includes connections that are protected with IPsec as well as those are not.

☐ **Allow the connection if it is secure**
 This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

☐ **Block the connection**

✓	spotify.exe	Public	Yes	Allow	
✗	spotify.exe	Private	Yes	Block	
✓	spotify.exe	Public	Yes	Allow	
✓	allow-using-port1433-tarik	All	Yes	Allow	
✓	TechSmith Camtasia 2019	All	Yes	Allow	
✓	@{Microsoft.MicrosoftStickyNotes_3.7.142....	@{Microsoft.MicrosoftSticky...	Domai...	Yes	Allow
✓	@{Microsoft.Windows.CloudExperienceHo...	@{Microsoft.Windows.Cloud...	Domai...	Yes	Allow
✓	@{microsoft.windowscommunicationsapp...	@{microsoft.windowscommu...	All	Yes	Allow

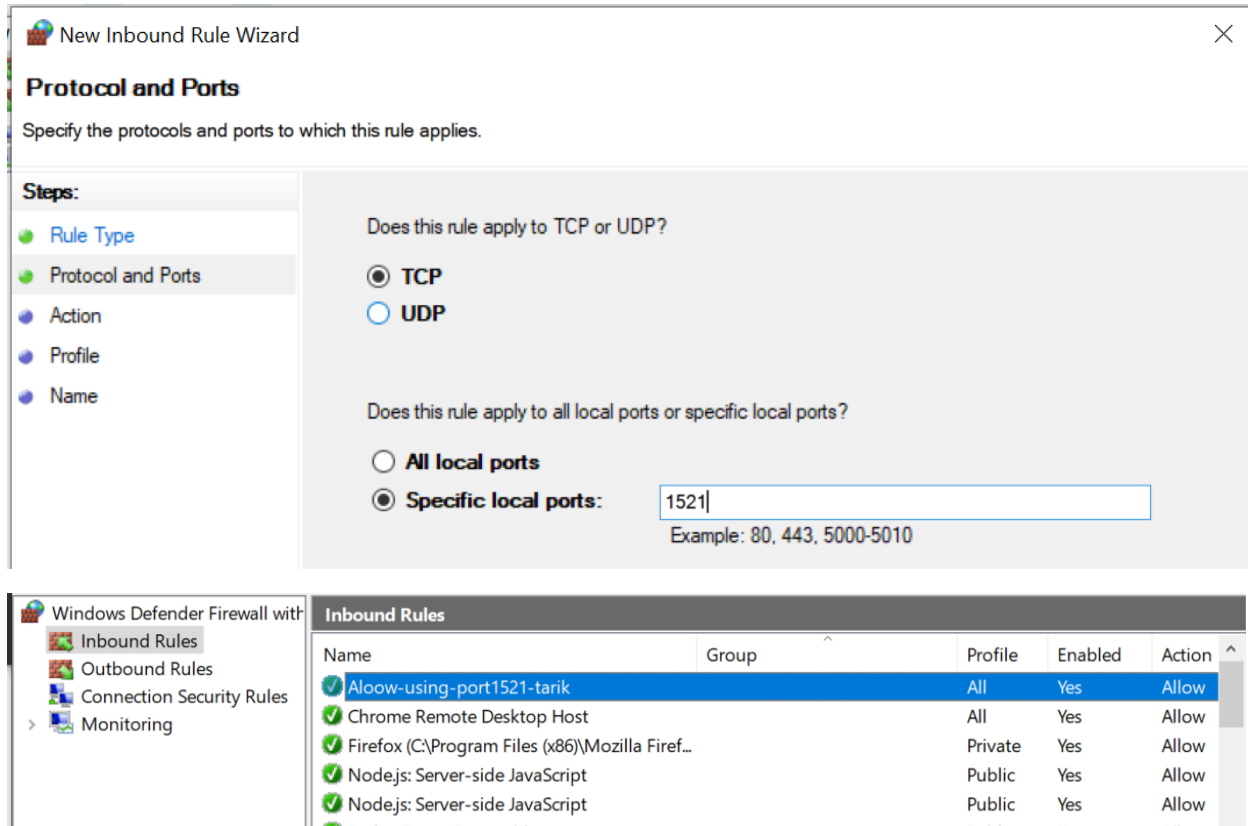
System oracle: Oracle 11g Express Edition

Pour pouvoir utiliser le Service ORCL de base de donnée, on doit ajouter une ligne qui contient l'adresse IP de notre machine qui contient L'instance oracle dans le fichier C:\oraclexe\app\oracle\product\11.2.0\server\network\ADMIN\listener.ora

```

*listener.ora - Notepad
File Edit Format View Help
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
  )
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
      (ADDRESS = (PROTOCOL = TCP)(HOST = TARIK-ELMSAOURI)(PORT = 1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.6)(PORT = 1521))
    )
  )
DEFAULT_SERVICE_LISTENER = (XE)
  
```

Autoriser l'utilisation du port TCP 1521 qui sera utilisé par l'instance de SQL Server de notre machine dans Le Windows Firewall :



Création de la table dans Postgresql:

```
SQL*Plus: Release 11.2.0.2.0 Production on Mon Mar 1 23:58:29 2021
Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL>
SQL>
SQL> connect system/oracle2020
Connected.
SQL> select host_name from v$instance;

HOST_NAME
-----
TARIK-ELMSAOURI

SQL> CREATE TABLE tablepro( id number not null,commentpro varchar(30),PRIMARY KEY(id
));

Table created.

SQL>
```

```
SQL> INSERT INTO tablepro values (8,'DATA1 FROM ORACLE');  
1 row created.  
  
SQL> INSERT INTO tablepro values (9,'DATA2 FROM ORACLE');  
1 row created.
```

Liaison des serveurs Dans SQL Server Management Studio & création de la vue partitionnée :

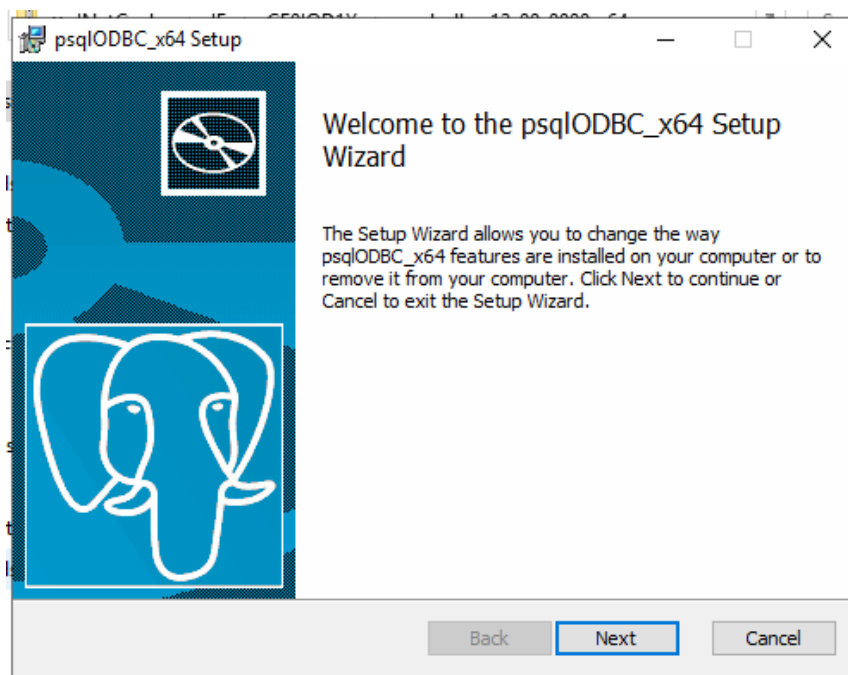
- ***Liaison de la BD2 Postgresql :***

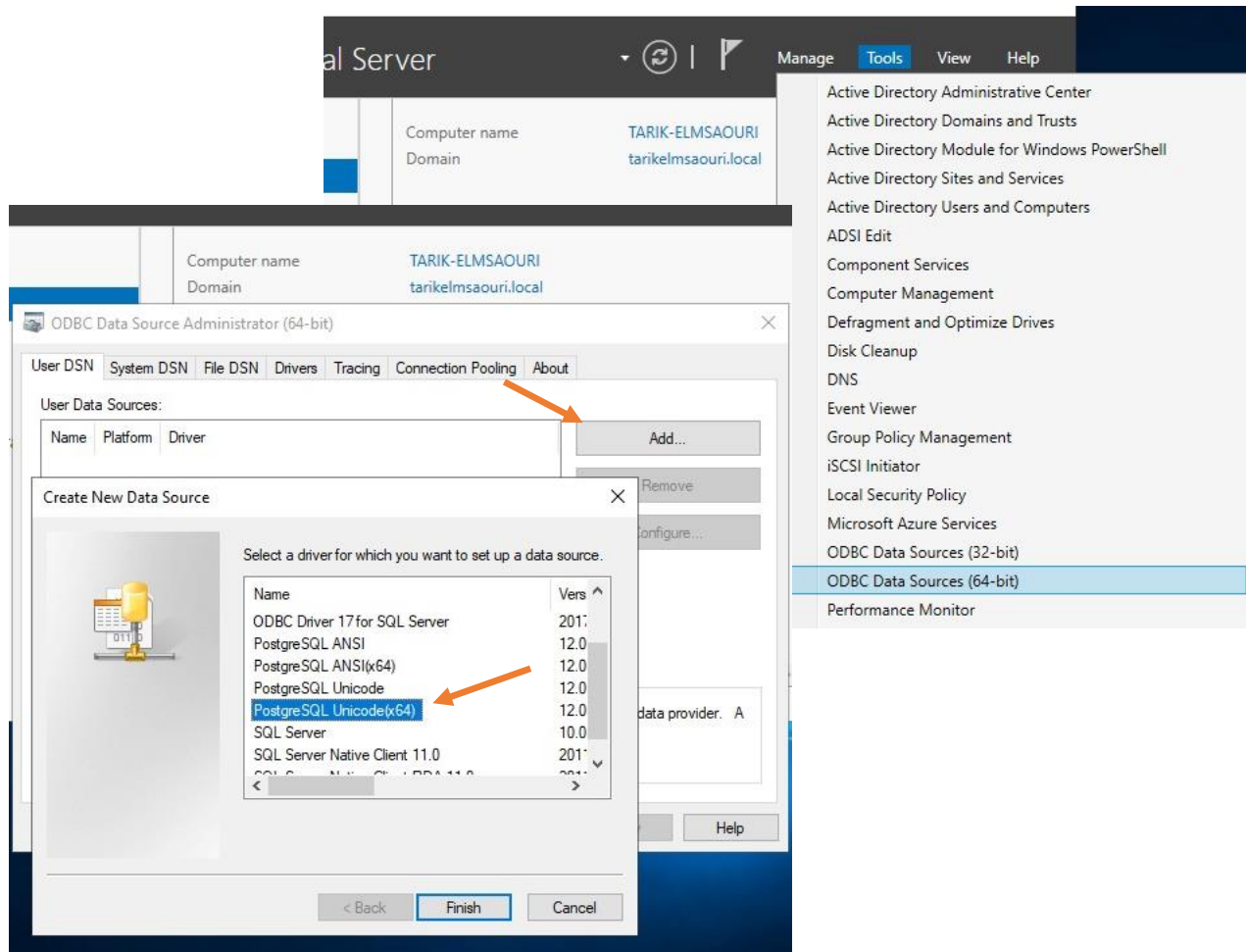
Puisque OLE DB de PostgreSQL est un outil payant, on va utiliser ODBC Postgre comme alternatif de liaison.

On télécharge l'ODBC depuis le site officiel du postgresql :

<https://www.postgresql.org/ftp/odbc/versions/msi/>

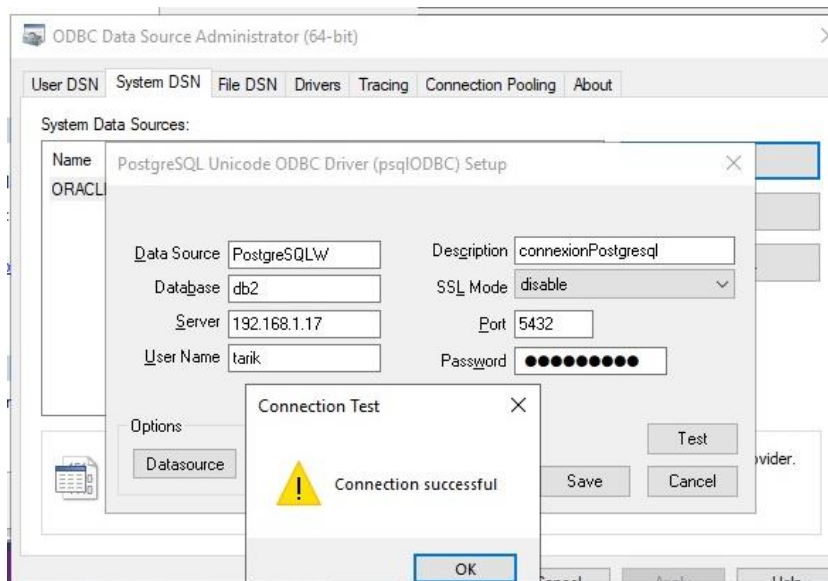
On l'installe sur notre machine Windows serveur.





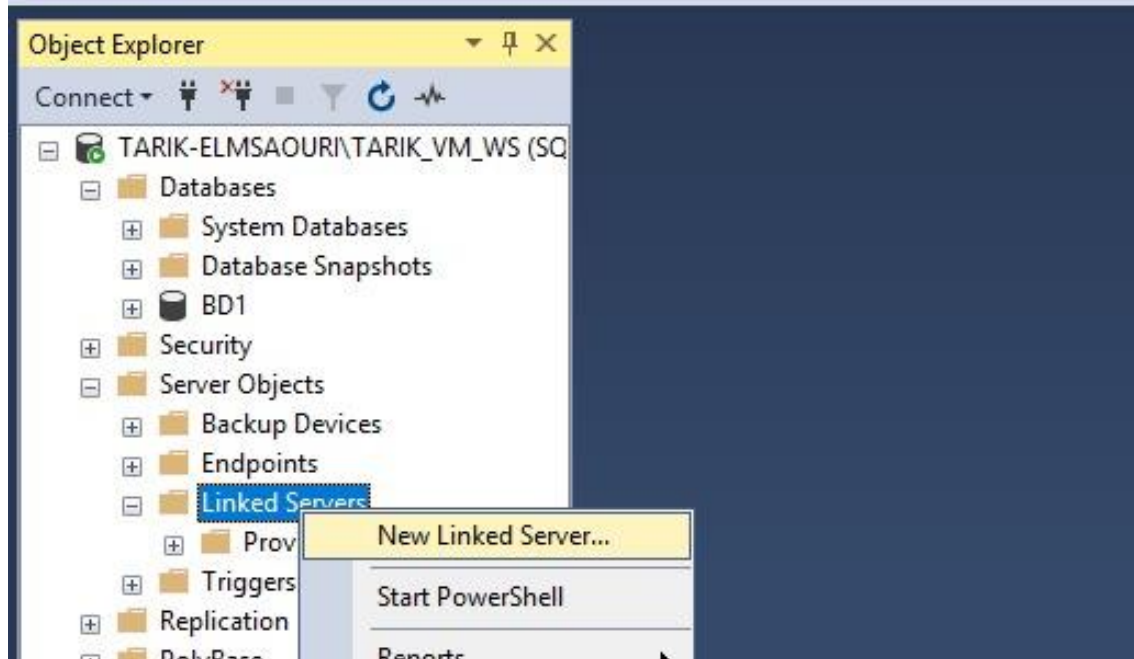
-On configure notre ODBC, avec PostgreSQLW est le nom qu'on va utiliser pour le lier avec le SQL serveur du Windows server.

192.168.1.17 est l'adresse IP de la machine Ubuntu qui contient le serveur Postgre



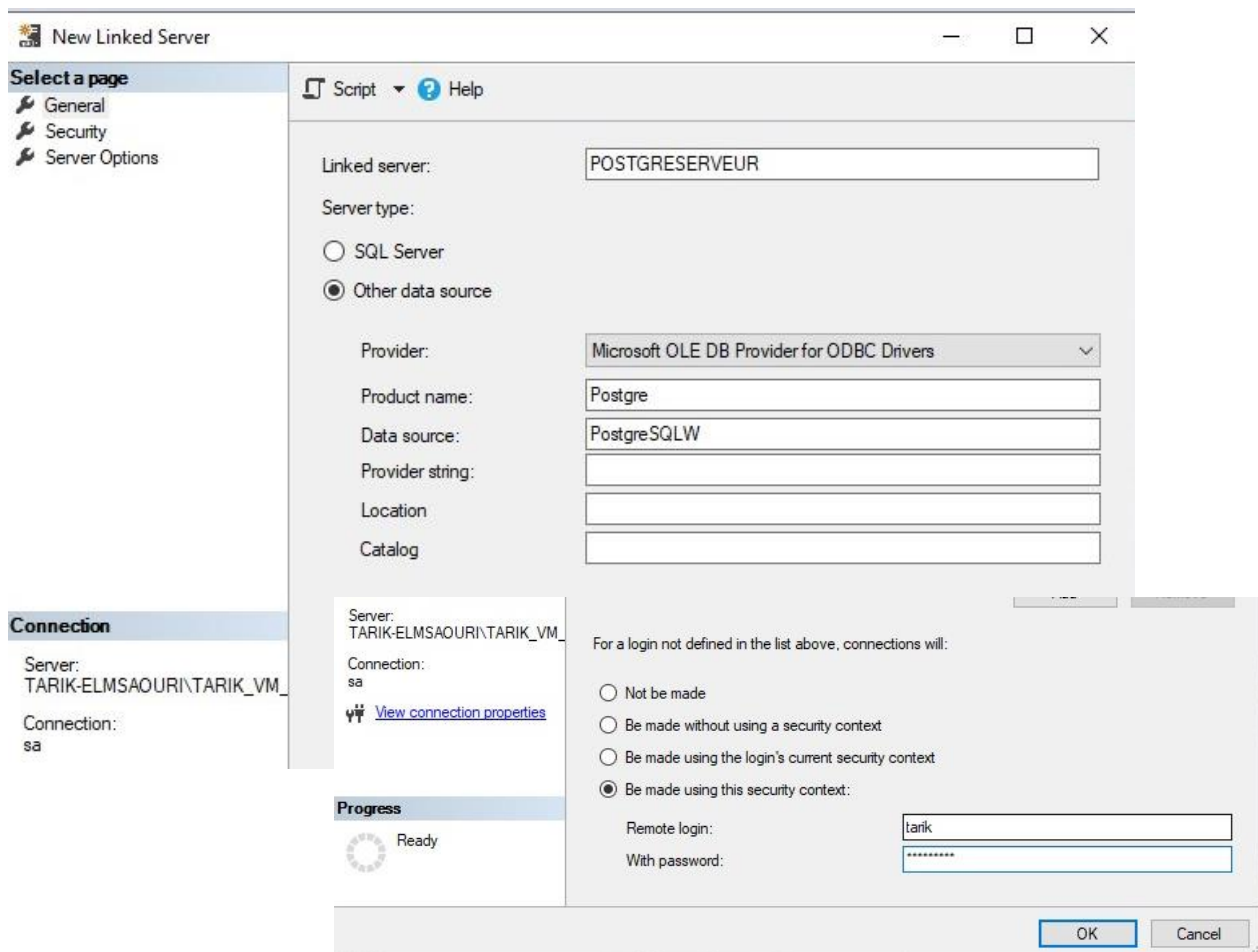
- **Linked Server :**

Allant à SQL SERVER MANAGEMENT STUDIO :



- **Liaison avec la BD2 Postgre :**

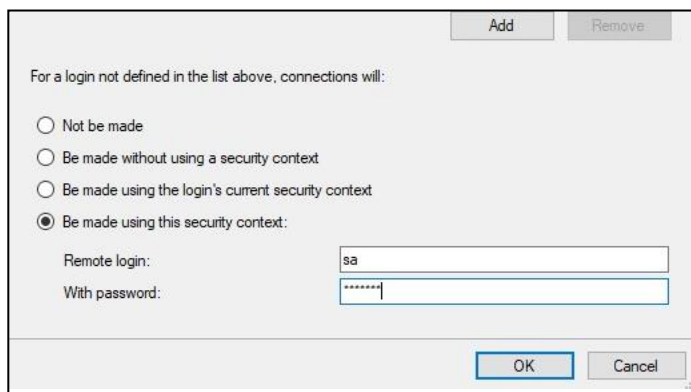
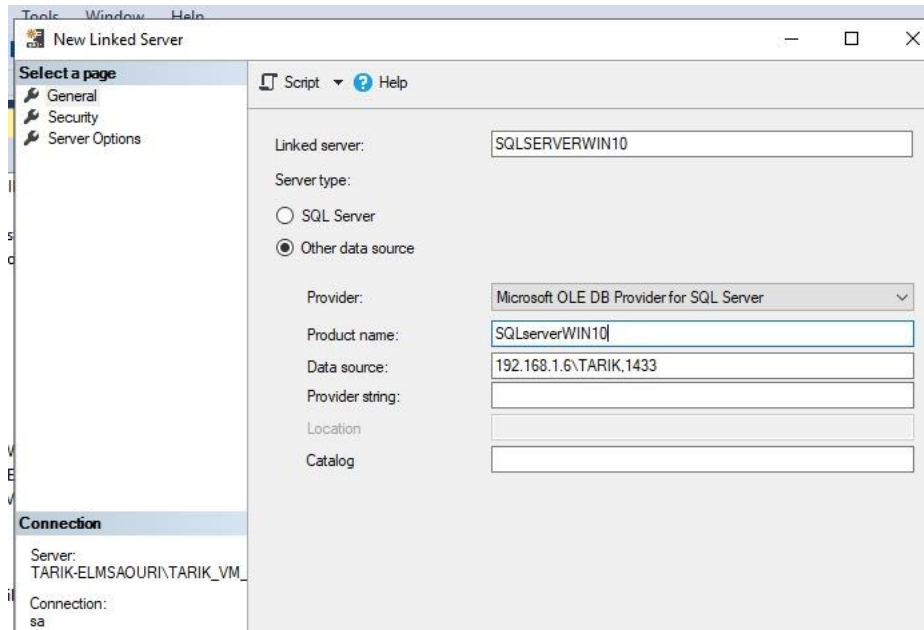
En utilisant ODBC DRIVER Qu'on déjà installé et le configuré : **PostgreSQLW**



- ***Liaison avec la BD3 SQL Server :***

En utilisant OLE DB Server et on a lié notre SQL SERVEUR (Principale) avec celui qui est dans la machine physique et qui a le nom d'instance TARIK et il utilise le port 1433.

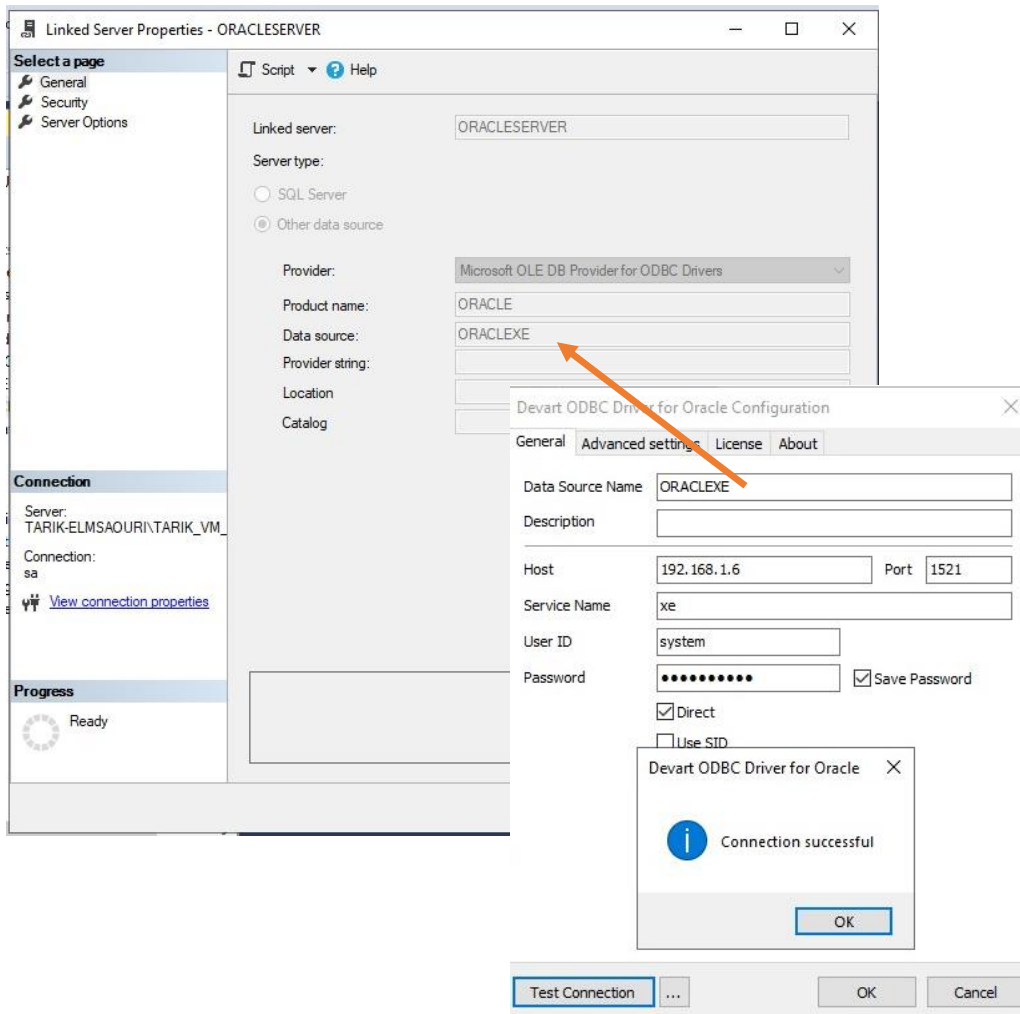
Et on n'oublie pas de spécifier le mot de passe pour se connecter sur cette instance.



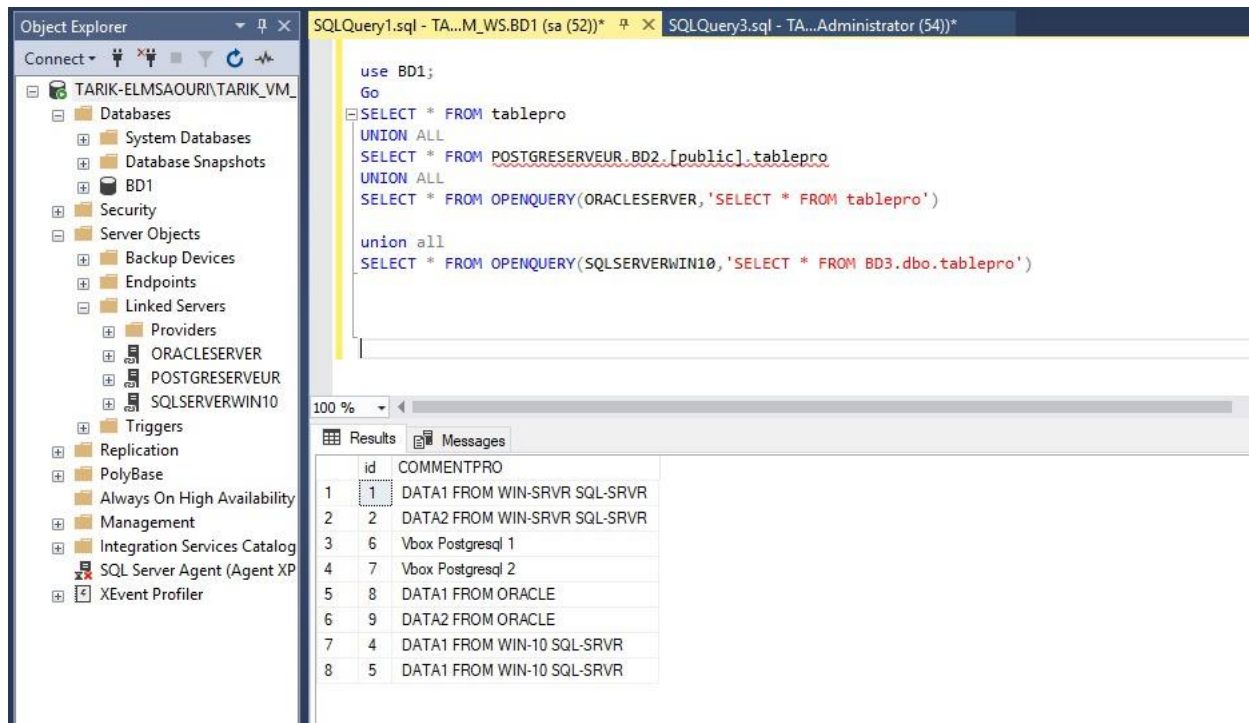
- ***Liaison avec la BD4 ORACLE :***

En utilisant ODBC DRIVER Server et on a lié notre SQL SERVEUR avec Oracle DB qui est dans la machine physique et qu'elle utilise le port 1521.

Et on n'oublie pas de spécifier le mot de passe pour se connecter sur cette instance.



Vérification de la liaison de la sql server avec les autres serveurs.

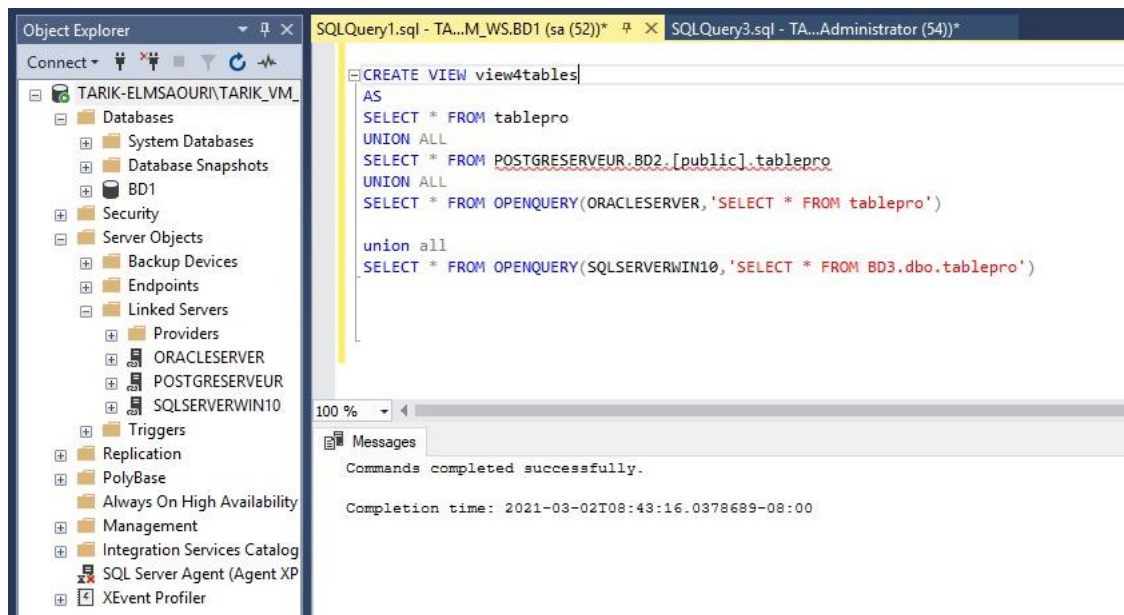


```
use BD1;
Go
SELECT * FROM tablepro
UNION ALL
SELECT * FROM POSTGRESERVEUR.BD2.[public].tablepro
UNION ALL
SELECT * FROM OPENQUERY(ORACLESERVER,'SELECT * FROM tablepro')

union all
SELECT * FROM OPENQUERY(SQLSERVERWIN10,'SELECT * FROM BD3.dbo.tablepro')
```

id	COMMENTPRO
1	DATA1 FROM WIN-SRVR SQL-SRVR
2	DATA2 FROM WIN-SRVR SQL-SRVR
3	Vbox Postgresql 1
4	Vbox Postgresql 2
5	DATA1 FROM ORACLE
6	DATA2 FROM ORACLE
7	DATA1 FROM WIN-10 SQL-SRVR
8	DATA1 FROM WIN-10 SQL-SRVR

Création de la vue partitionne basée sur les 4 bases de données :



```
CREATE VIEW view4tables
AS
SELECT * FROM tablepro
UNION ALL
SELECT * FROM POSTGRESERVEUR.BD2.[public].tablepro
UNION ALL
SELECT * FROM OPENQUERY(ORACLESERVER,'SELECT * FROM tablepro')

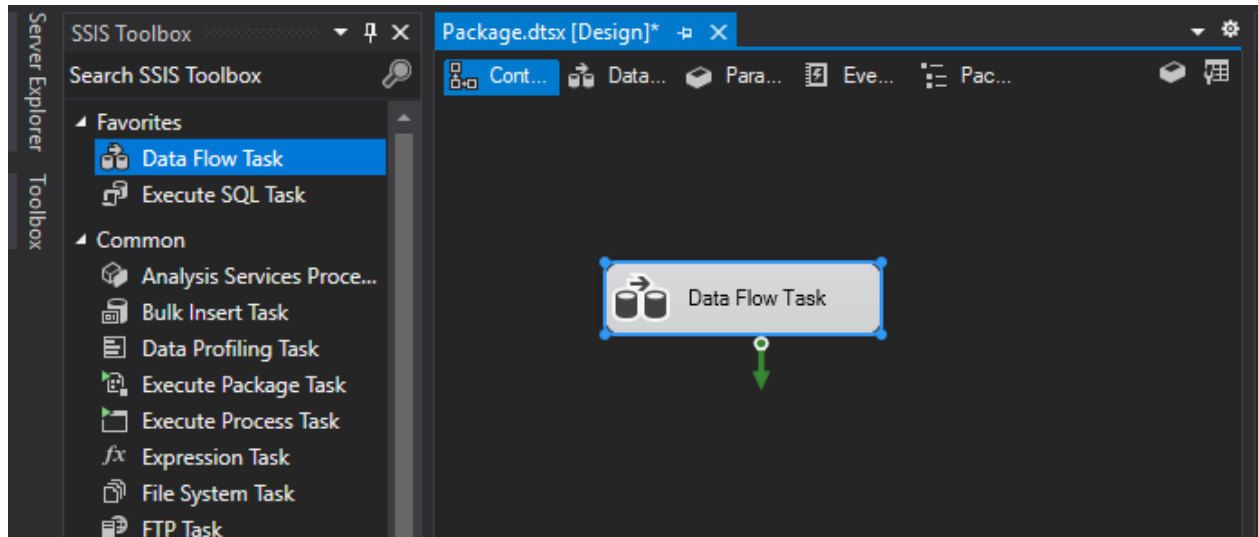
union all
SELECT * FROM OPENQUERY(SQLSERVERWIN10,'SELECT * FROM BD3.dbo.tablepro')
```

Commands completed successfully.

Completion time: 2021-03-02T08:43:16.0378689-08:00

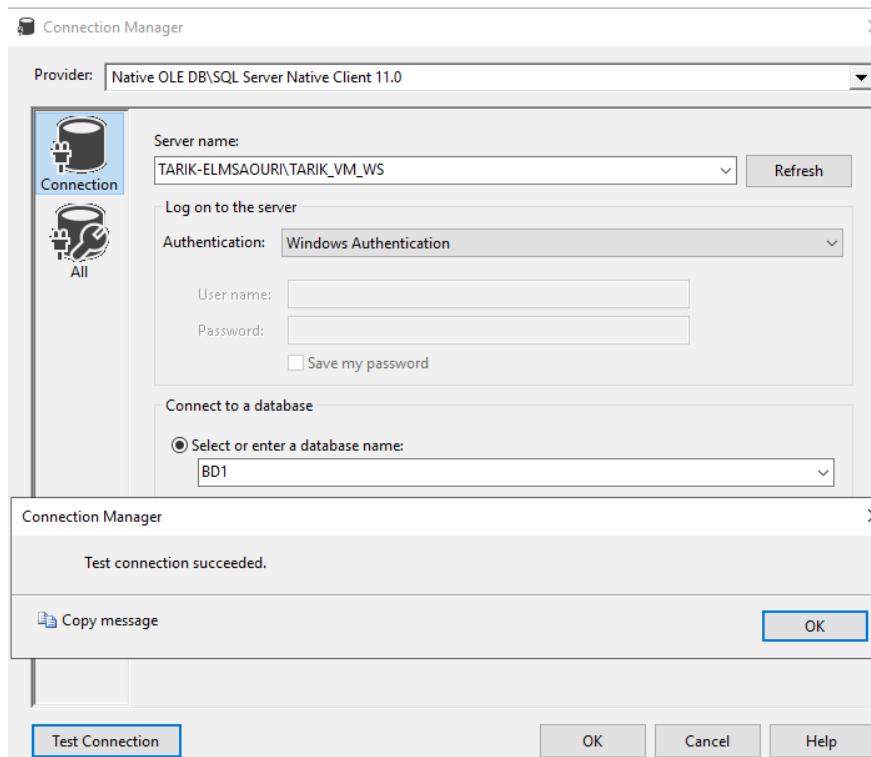
Exportation des données de BD1 vers BD2, BD3, BD4, Fichier Texte en utilisant SSIS.

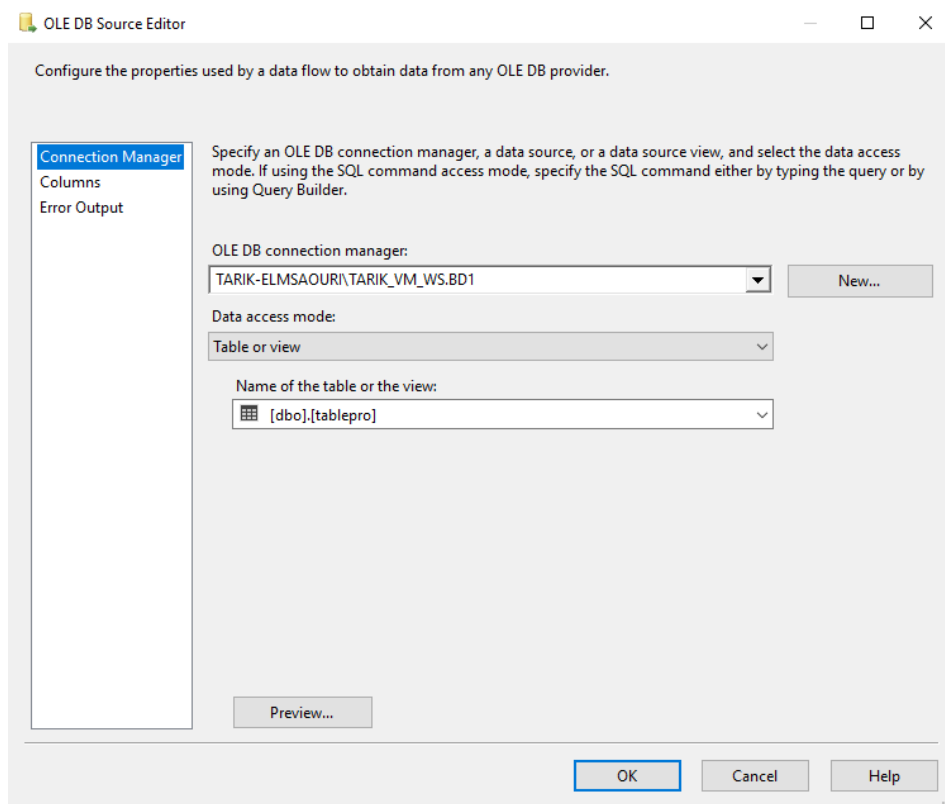
Creation d'un Data flow Task dans onglet Control flow



Configuration des Components de SSIS :

BD1 SQL SERVER WINDOWS SERVER





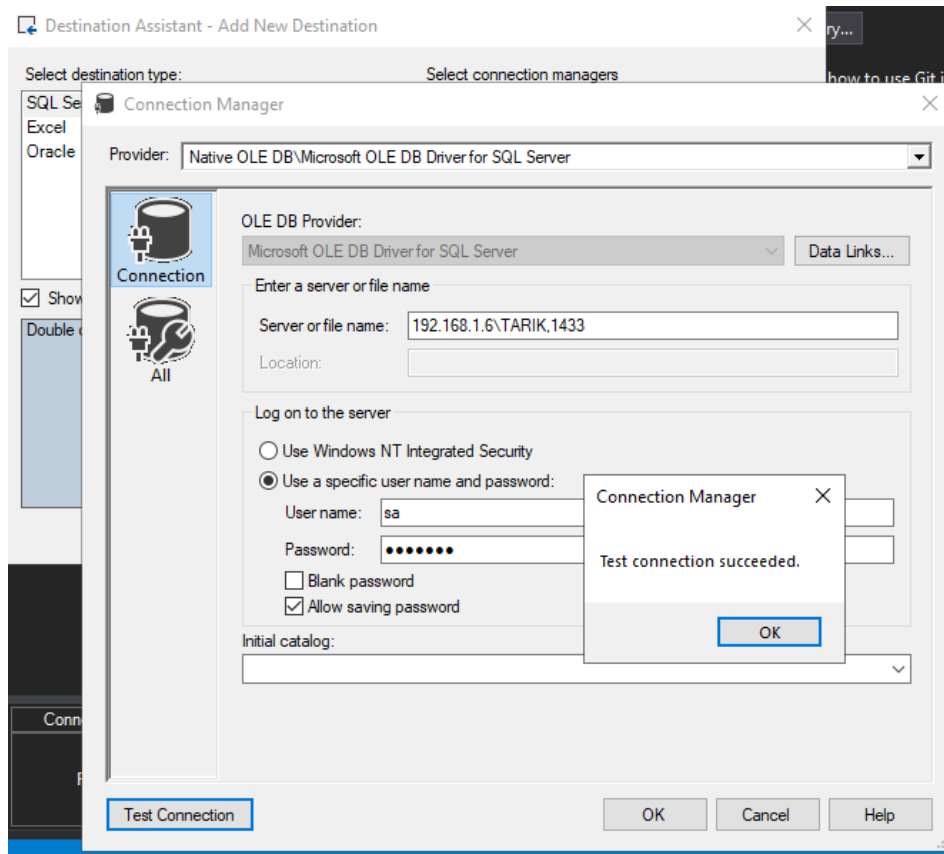
Contenu de la table:

Preview Query Results

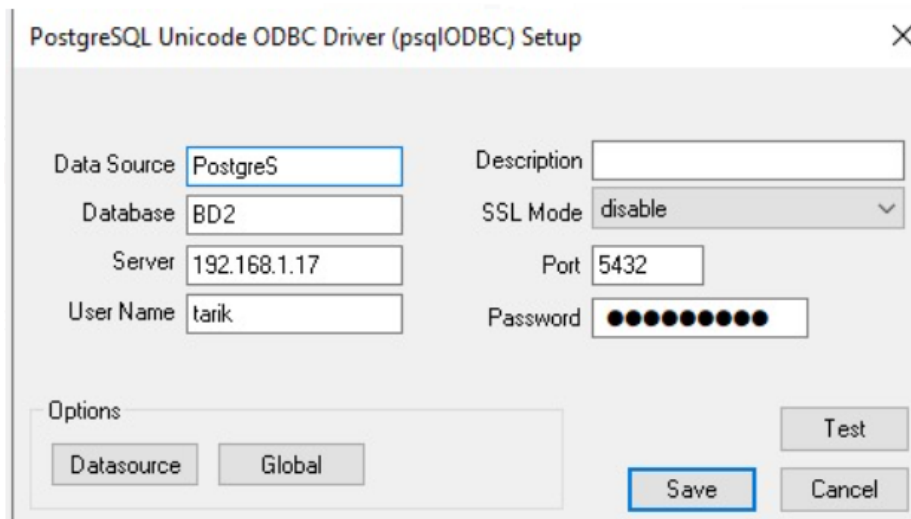
Query result (up to the first 200 rows):

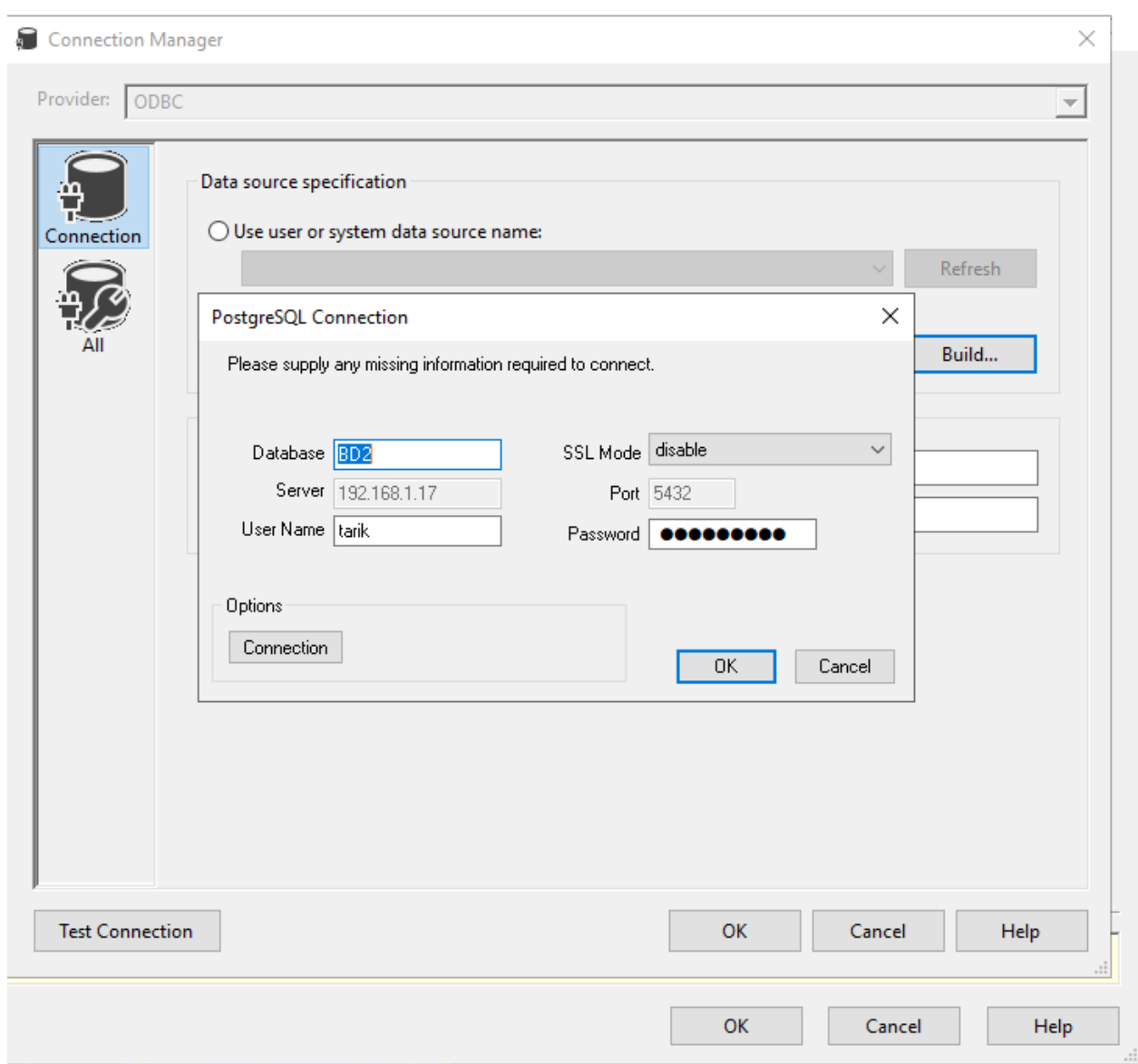
id	COMMENTPRO
1	DATA1 FROM WIN-SRVR. SQL-SRVR
2	DATA2 FROM WIN-SRVR. SQL-SRVR

BD3 SQL SERVER dans Machine physique Windows 10



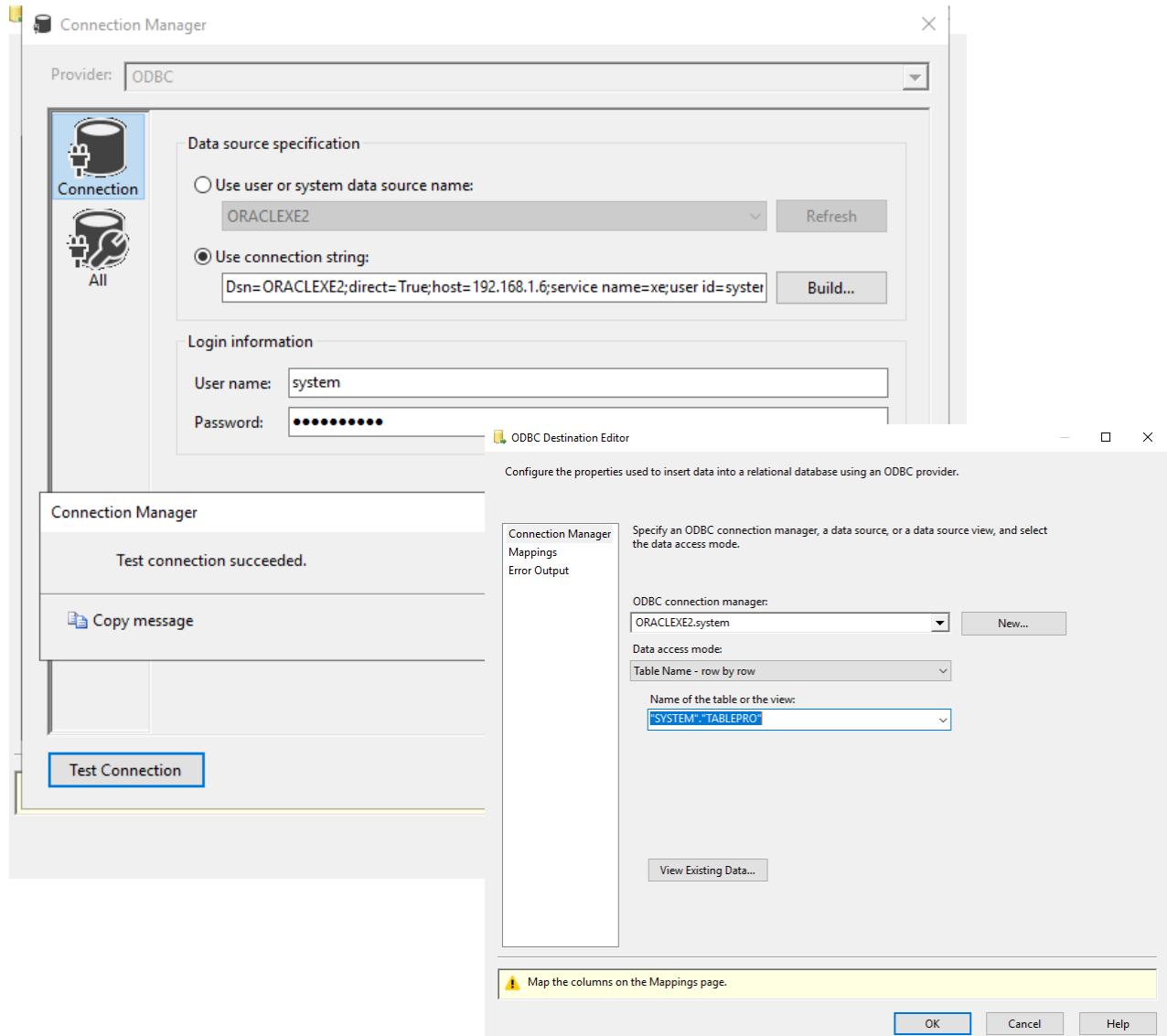
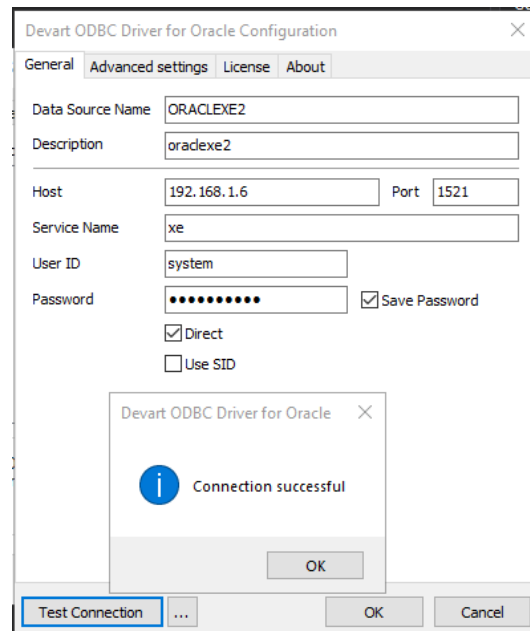
BD2 POSTGRESQL dans Vbox Ubuntu



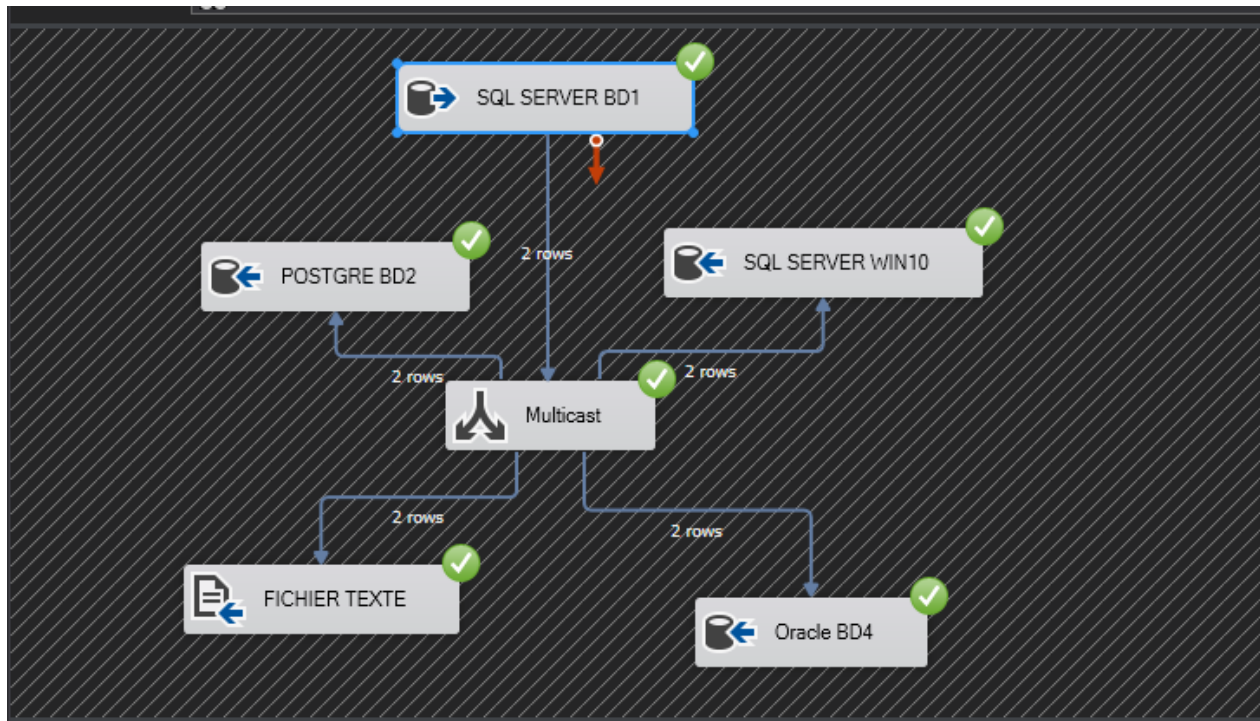


BD4 ORACLE dans Machine physique Windows 10

En utilisant l'ODBC d'oracle on a déjà installé car il fonctionne bien avec le package SSIS.



Exécution de l'exportation des données de la table 'tablepro'



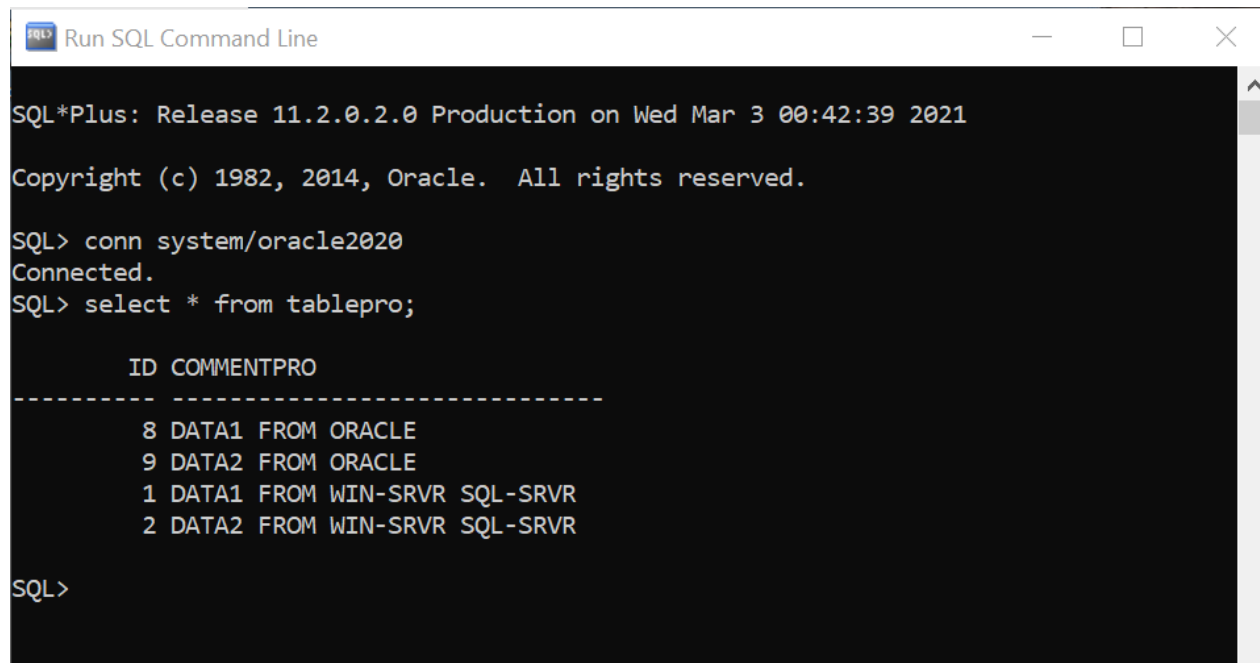
Vérification que les données ont été reçues

BD2 - POSTGRESQL:

```
tarik@tarik-linux: ~  
tarik@tarik-linux:~$ psql -d BD2  
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))  
Type "help" for help.  
  
BD2=# select * from tablepro;  
 id |      commentpro        
----+-----  
  6 | Vbox Postgresql 1  
  7 | Vbox Postgresql 2  
  1 | DATA1 FROM WIN-SRVR SQL-SRVR  
  2 | DATA2 FROM WIN-SRVR SQL-SRVR  
(4 rows)  
  
BD2=#
```

BD3 – SQL SERVER WIN10:

BD4 – ORACLE WIN10:



```
SQL*Plus: Release 11.2.0.2.0 Production on Wed Mar 3 00:42:39 2021

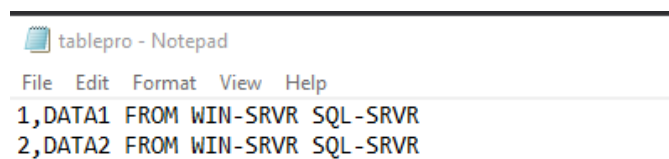
Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> conn system/oracle2020
Connected.
SQL> select * from tablepro;

      ID COMMENTPRO
-----
      8 DATA1 FROM ORACLE
      9 DATA2 FROM ORACLE
      1 DATA1 FROM WIN-SRVR SQL-SRVR
      2 DATA2 FROM WIN-SRVR SQL-SRVR

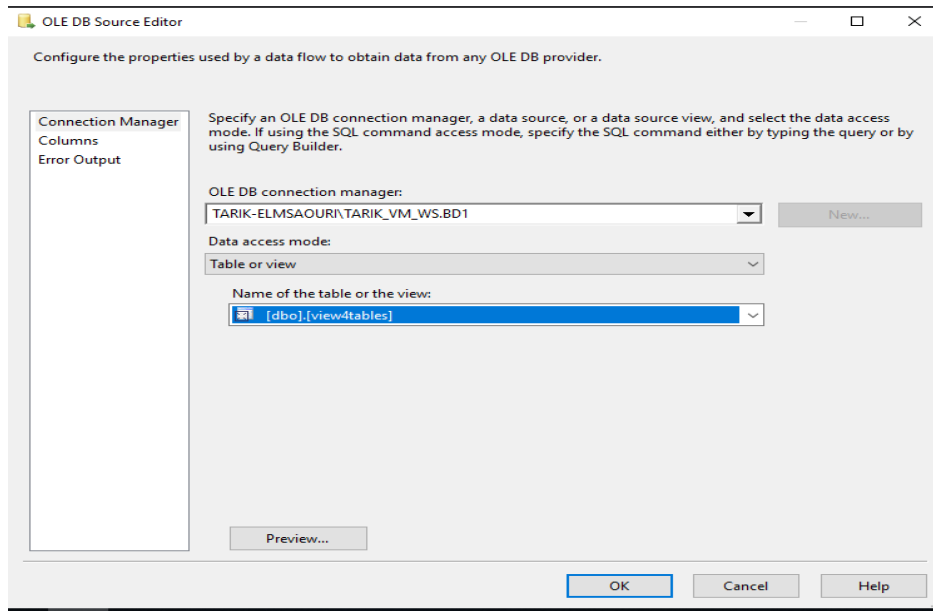
SQL>
```

FICHER TEXE:



```
1,DATA1 FROM WIN-SRVR SQL-SRVR
2,DATA2 FROM WIN-SRVR SQL-SRVR
```


Exportation de la vue partitionnée vers une table vide créée dans les bases de données destinations



Conclusion :

A travers ce projet, nous avons pu mettre en pratique nos connaissances en termes de l'administration de base données. Nous avons aussi acquis une compréhension forte sur le fonctionnement des différents systèmes des bases données comme : SQL SERVER, ORACLE et POSTGRE.

En plus, On a appris comment on peut créer et intégrer un système au domaine, comment lier plusieurs serveurs de différents types.

Finalement, ce projet nous aura aussi permis d'approfondir nos notions de gestion de projet.