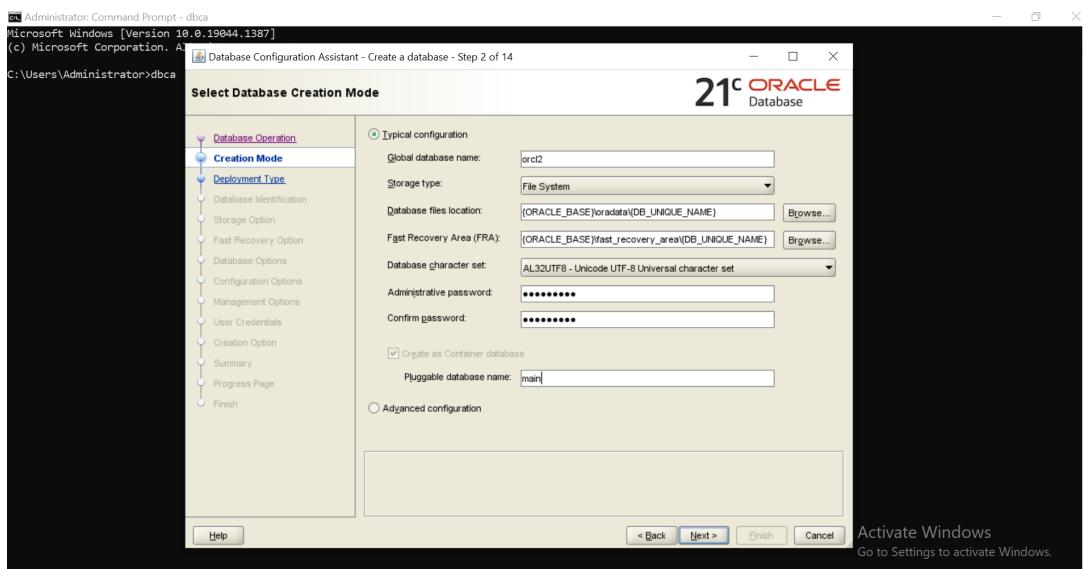
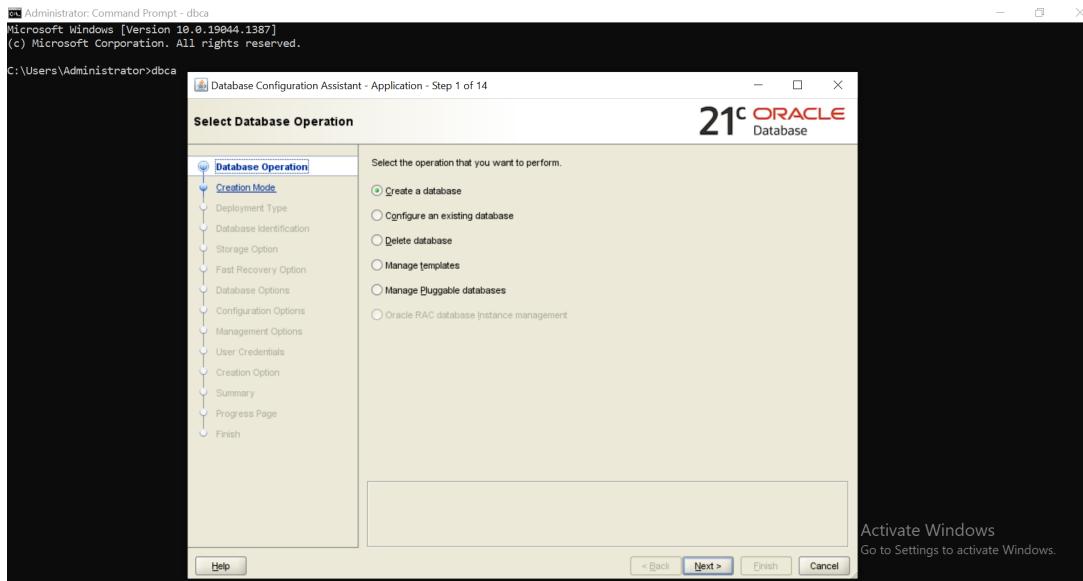


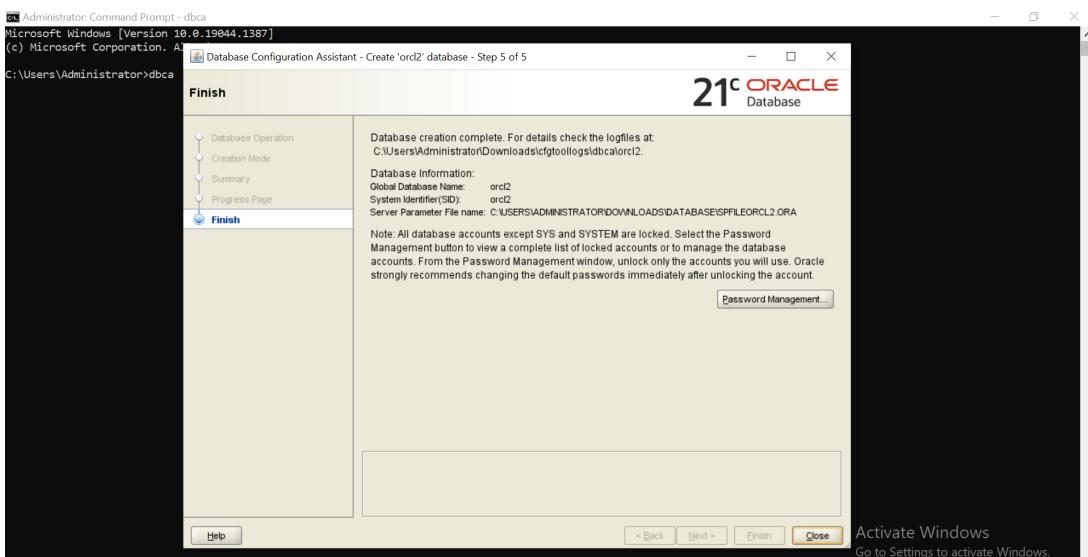
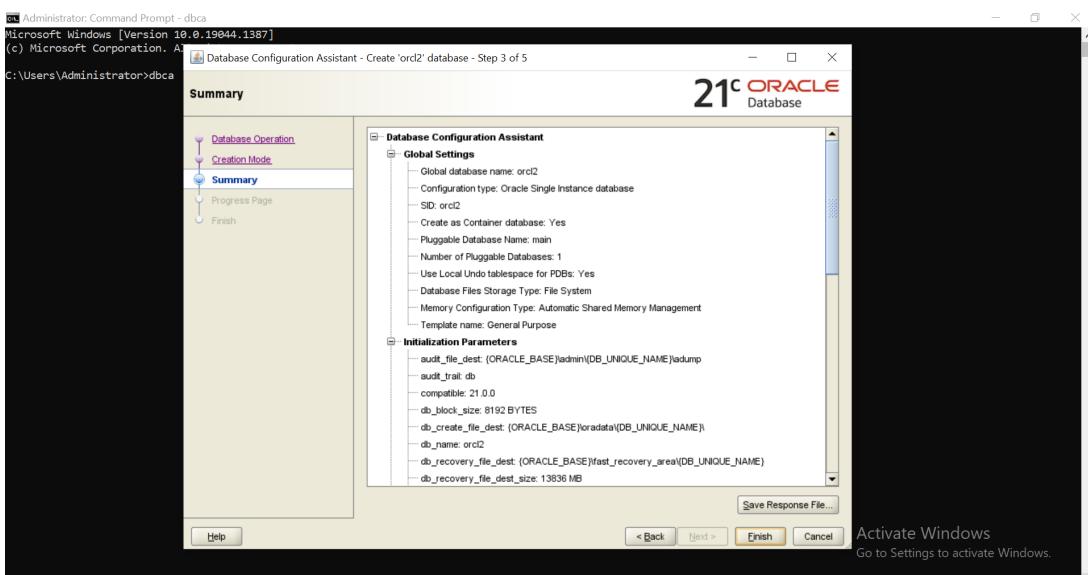
# Executer Plusieur Instance D'ORACLE TP2

## Creation D'une Nouvelle Instance

On va executer la commande `dbca` pour lancer l'application java **Database Configuration Assistant** qui va nous permettre de creer une nouvelle instance d'oracle :

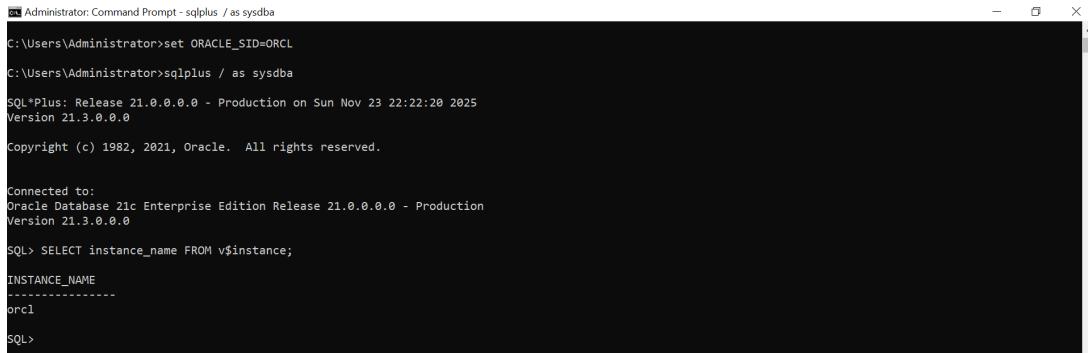
1. Choisir **Create a database** et cliquer sur **next**.
2. Choisir **Typical configuration**.
3. Donner un identifiant unique a l'instance **Global database name**.
4. Donne un mot de passe administratif **Administrative password**.
5. Donne un nom a la **Pluggable Database** (represente une base de donne de l'instance).
6. Cliquer sur **next** puis **finish**.
7. Attendre le telechargement puis cliquer sur **close**.





## Executer Plusieur Instance

1. Apres avoir creer les instance on va ouvrir plusieur instance du terminal et on va utiliser la commande `set ORACLE_SID = <nom_instance>` qui va temporairement changer la variable d'environnement `ORACLE_SID` pour selectionner l'instance dans notre session terminal.
2. On lance oracle avec `sqlplus / as sysdba`.
3. Pour verifier l'instance on va consulter la view `v$instance` : `select instance_name from v$instance;`

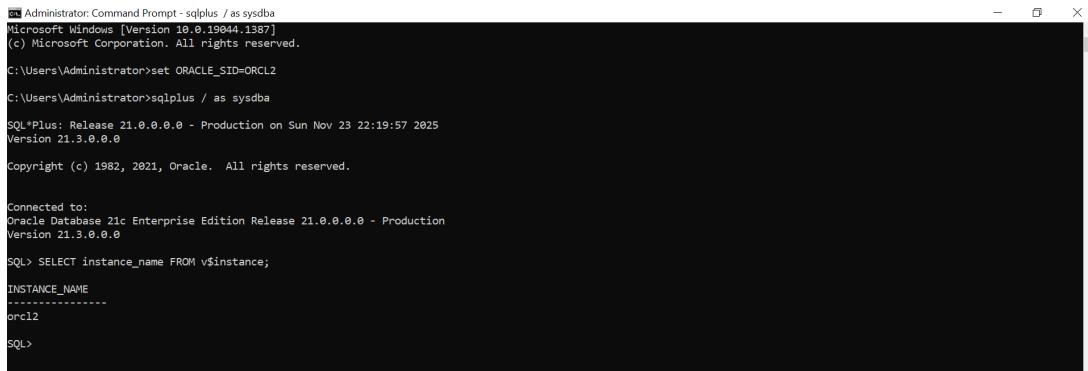


```
C:\Users\Administrator>set ORACLE_SID=ORCL
C:\Users\Administrator>sqlplus / as sysdba
SQL*Plus: Release 21.0.0.0.0 - Production on Sun Nov 23 22:22:20 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> SELECT instance_name FROM v$instance;
INSTANCE_NAME
-----
orcl
SQL>
```



```
C:\Users\Administrator>set ORACLE_SID=ORCL2
C:\Users\Administrator>sqlplus / as sysdba
SQL*Plus: Release 21.0.0.0.0 - Production on Sun Nov 23 22:19:57 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> SELECT instance_name FROM v$instance;
INSTANCE_NAME
-----
orcl2
SQL>
```

## Lien

1. Pour La Creation de lien on va utiliser la requete suivant :

```
CREATE DATABASE LINK NOM_LIEN
CONNECT TO NOM_UTILISATEUR IDENTIFIED BY MOTDEPASSE_INSTANCE
USING 'NOM_INSTANCE';
```

2. Acceder a une table avec le lien : `nom_table@nom_lien`
3. Pour consulter les liens de l'utilisateur courrant on utilise la view `user_db_links`
4. Au lieu de changer la variable d'environnement `ORACLE_SID` on peut indiquer a quelle instance on veut se connecter dans la commande `sqlplus` comme suit : `sqlplus userName/password@SID`

Avant de creer les liens on va creer un user dans chaque instances et leur donne tous les privileges :

</>

### Sql Program 1: Creation De L'utilisateur userORCL dans ORCL

</>

```
-- creation de l'utilisateur
create user C##userORCL identified by Nanno1234;

-- affectation des privileges
grant all privileges to C##userORCL;
```

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt - sqlplus / as sysdba". The session starts with setting the Oracle SID to "orcl1" and connecting to the database. Then, it creates a user named "C##userORCL" with the password "Nanno1234" and grants all privileges to this user.

```
C:\Users\Administrator>set ORACLE_SID=orcl1
C:\Users\Administrator>sqlplus / as sysdba
SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 9 15:27:21 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> select instance_name from v$instance;
INSTANCE_NAME
-----
orcl1

SQL> create user c##userORCL identified by Nanno1234;
User created.

SQL> grant all privileges to c##userORCL;
Grant succeeded.
```

</>

### Sql Program 2: Creation De L'utilisateur userORCL2 dans ORCL2

</>

```
-- creation de l'utilisateur
create user C##userORCL2 identified by Nanno1234;

-- affectation des privileges
grant all privileges to C##userORCL2;
```

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt - sqlplus / as sysdba". The session starts with setting the Oracle SID to "orcl2" and connecting to the database. Then, it creates a user named "C##userORCL2" with the password "Nanno1234" and grants all privileges to this user.

```
C:\Users\Administrator>set ORACLE_SID=orcl2
C:\Users\Administrator>sqlplus / as sysdba
SQL*Plus: Release 21.0.0.0 - Production on Tue Dec 9 15:30:08 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0 - Production
Version 21.3.0.0.0

SQL> select instance_name from v$instance;
INSTANCE_NAME
-----
orcl2

SQL> create user c##userORCL2 identified by Nanno1234;
User created.

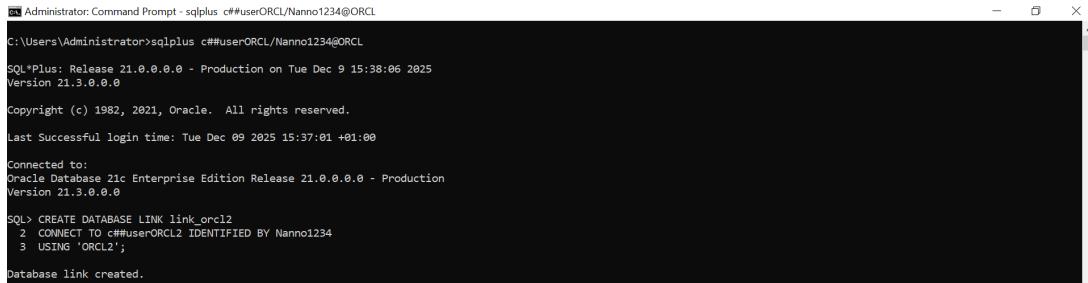
SQL> grant all privileges to c##userORCL2;
Grant succeeded.
```

Creation des liens :

```
</>          Sql Program 3: Creation Du Lien Vers ORCL2 Dans ORCL          </>
-- creation du lien
CREATE DATABASE LINK link_orcl2
CONNECT TO c##userORCL2 IDENTIFIED BY Nanno1234
USING 'ORCL2';

-- description de la view des liens de l'utilisateur courrant
desc db_user_links;

-- afficher le nom du lien et son SID
select host,db_link from db_user_links;
```



```
C:\Administrator: Command Prompt - sqlplus c##userORCL/Nanno1234@ORCL
C:\Users\Administrator>sqlplus c##userORCL/Nanno1234@ORCL

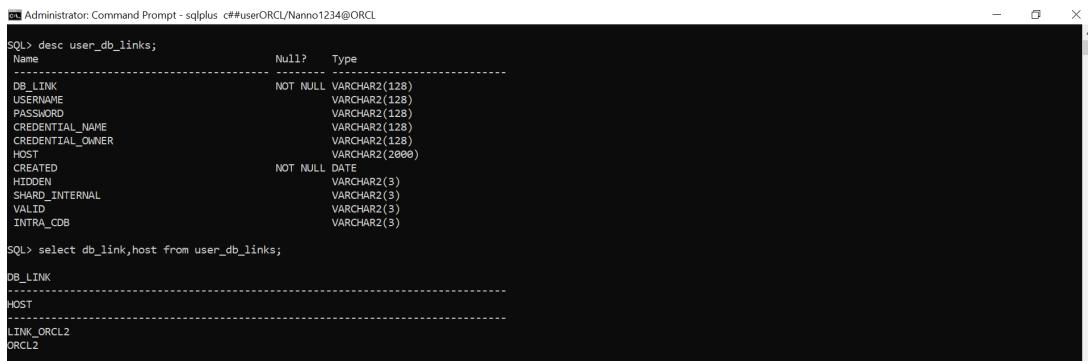
SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 9 15:38:06 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Last Successful login time: Tue Dec 09 2025 15:37:01 +01:00

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> CREATE DATABASE LINK link_orcl2
  2  CONNECT TO c##userORCL2 IDENTIFIED BY Nanno1234
  3  USING ORCL2';
Database link created.
```

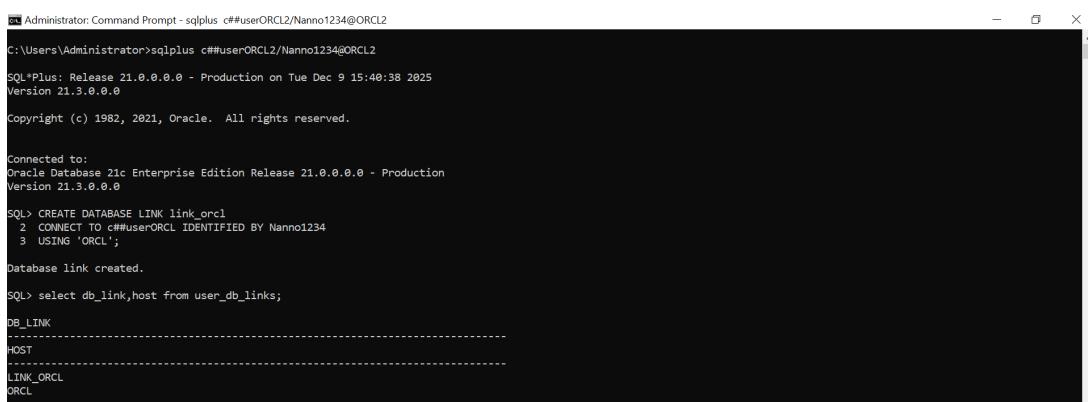


```
C:\Administrator: Command Prompt - sqlplus c##userORCL/Nanno1234@ORCL
SQL> desc user_db_links;
Name                           Null?    Type
-----                         -----
DB_LINK                        NOT NULL VARCHAR2(128)
USERNAME                       VARCHAR2(128)
PASSWORD                       VARCHAR2(128)
CREDENTIAL_NAME                VARCHAR2(128)
CREDENTIAL_OWNER               VARCHAR2(128)
HOST                           VARCHAR2(2000)
CREATED                        NOT NULL DATE
HIDDEN                         VARCHAR2(3)
SHARD_INTERNAL                 VARCHAR2(3)
VALID                          VARCHAR2(3)
INTRACDB                       VARCHAR2(3)

SQL> select db_link,host from user_db_links;
DB_LINK
-----
HOST
-----
LINK_ORCL2
ORCL2
```

```
</>          Sql Program 4: Creation Du Lien Vers ORCL Dans ORCL2          </>
-- creation du lien
CREATE DATABASE LINK link_orcl
CONNECT TO c##userORCL IDENTIFIED BY Nanno1234
USING 'ORCL';

-- afficher le nom du lien et son SID
select host,db_link from db_user_links;
```



```
C:\Administrator: Command Prompt - sqlplus c##userORCL2/Nanno1234@ORCL2
C:\Users\Administrator>sqlplus c##userORCL2/Nanno1234@ORCL2

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 9 15:40:38 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> CREATE DATABASE LINK link_orcl
  2  CONNECT TO c##userORCL IDENTIFIED BY Nanno1234
  3  USING ORCL';
Database link created.

SQL> select db_link,host from user_db_links;
DB_LINK
-----
HOST
-----
LINK_ORCL
ORCL
```

Test Pour Confirmer le lien :

```
Administrator: Command Prompt - sqlplus c##userORCL2/Nanno1234@ORCL2
C:\Users\Administrator>sqlplus c##userORCL/Nanno1234@ORCL
SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 9 15:42:32 2025
Version 21.3.0.0.0
Copyright (c) 1982, 2021, Oracle. All rights reserved.
Last Successful login time: Tue Dec 09 2025 15:38:06 +01:00
Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL> create table test1 (a int);
Table created.

SQL> exit
Disconnected from Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
C:\Users\Administrator>sqlplus c##userORCL2/Nanno1234@ORCL2
SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 9 15:42:58 2025
Version 21.3.0.0.0
Copyright (c) 1982, 2021, Oracle. All rights reserved.
Last Successful login time: Tue Dec 09 2025 15:48:49 +01:00
Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL> desc test1@link_orcl;
Name          Null?    Type
-----        -----   -----
A             NUMBER(38)

SQL>
```

## Creation Et Insertion Des Table Dans ORCL Avec userORCL

```
</>                               Sql Program 5: Table Participant </>

-- creation de la table
-- participant(idParticipant, nomParticipant, ville, age)

CREATE TABLE participant (
    idParticipant NUMBER,
    nomParticipant VARCHAR2(50),
    ville VARCHAR2(50),
    age NUMBER,
    CONSTRAINT pk_participant PRIMARY KEY (idParticipant)
);

-- Insertion Automatique Avec PL/SQL

DECLARE
    v_idparticip NUMBER;
    v_nomparticip VARCHAR2(50);
    v_ville VARCHAR2(50);
    v_age NUMBER;
BEGIN
    FOR i IN 1..10 LOOP -- Remplir 10 enregistrement
        v_idparticip := i;
        v_nomparticip := 'Participant' || i;
        v_ville := CASE ROUND(DBMS_RANDOM.VALUE(1, 4))
            WHEN 1 THEN 'ALGER'
            WHEN 2 THEN 'ORAN'
            WHEN 3 THEN 'TIZI OUZOU'
            ELSE 'ANNABA'
        END;
        v_age := ROUND(DBMS_RANDOM.VALUE(18, 60));

        INSERT INTO participant (idParticipant, nomParticipant, ville, age)
        VALUES (v_idparticip, v_nomparticip, v_ville, v_age);
    END LOOP;
    COMMIT;
END;
/
```

```
-- creation de la table
--randonnee(idRando, nomRando, region, distance, denivele)

CREATE TABLE randonnee (
    idRando NUMBER,
    nomRando VARCHAR2(50),
    region VARCHAR2(50),
    distance NUMBER,
    denivele NUMBER,
    CONSTRAINT pk_randonnee PRIMARY KEY (idRando)
);
```

-- Insertion Automatique Avec PL/SQL

```
DECLARE
    v_idRando NUMBER;
    v_nomRando VARCHAR2(50);
    v_region VARCHAR2(50);
    v_distance NUMBER;
    v_denivele NUMBER;

BEGIN
    FOR i IN 1..10 LOOP -- Remplir 10 enregistrements, ajustez selon vos besoins
        v_idRando := i;
        v_nomRando := 'Randonnee' || i;
        v_region := CASE ROUND(DBMS_RANDOM.VALUE(1, 4))
            WHEN 1 THEN 'ALGER'
            WHEN 2 THEN 'ORAN'
            WHEN 3 THEN 'TIZI OUZOU'
            ELSE 'ANNABA'
        END;
        v_distance := ROUND(DBMS_RANDOM.VALUE(5, 20), 2);
        v_denivele := ROUND(DBMS_RANDOM.VALUE(500, 1000));

        INSERT INTO randonnee (idRando, nomRando, region, distance, denivele)
        VALUES (v_idRando, v_nomRando, v_region, v_distance, v_denivele);
    END LOOP;
    COMMIT;
END;
/
```

```

-- creation de la table
-- sortie(idRando, idParticipant, dateSortie, dureeSortie)

CREATE TABLE sortie (
    idRando NUMBER,
    idParticipant NUMBER,
    dateSortie DATE,
    dureeSortie NUMBER,
    CONSTRAINT pk_sortie PRIMARY KEY (idRando, idParticipant, dateSortie),
    CONSTRAINT fk_sortie_rando FOREIGN KEY (idRando) REFERENCES randonnee(idRando),
    CONSTRAINT fk_sortie_participant FOREIGN KEY (idParticipant) REFERENCES
        participant(idParticipant)
);

-- Insertion Automatique Avec PL/SQL

DECLARE
    v_idRando NUMBER;
    v_idParticipant NUMBER;
    v_dateSortie DATE;
    v_dureeSortie NUMBER;

BEGIN
    FOR i IN 1..10 LOOP -- Remplir 10 enregistrements, -- Sélection aléatoire d'idRando et
        -- idParticipant existants
        SELECT idRando INTO v_idRando FROM (
            SELECT idRando
            FROM randonnee
            ORDER BY DBMS_RANDOM.VALUE
        )
        WHERE ROWNUM = 1;

        SELECT idParticipant INTO v_idParticipant FROM (
            SELECT idParticipant
            FROM participant
            ORDER BY DBMS_RANDOM.VALUE
        )
        WHERE ROWNUM = 1;

        -- Génération de valeurs aléatoires pour les autres colonnes
        v_dateSortie := TO_DATE('2023-01-01', 'YYYY-MM-DD') + ROUND(DBMS_RANDOM.VALUE(1,
            365));
        v_dureeSortie := ROUND(DBMS_RANDOM.VALUE(1, 8));

        -- Insertion dans la table sortie
        INSERT INTO sortie (idRando, idParticipant, dateSortie, dureeSortie)
        VALUES (v_idRando, v_idParticipant, v_dateSortie, v_dureeSortie);
    END LOOP;
    COMMIT;
END;
/

```

```
-- creation de la table
-- Projet(numj, nomj, ville)

CREATE TABLE Projet (
    numj NUMBER,
    nomj VARCHAR2(50),
    ville VARCHAR2(50),
    CONSTRAINT pk_projet PRIMARY KEY (numj)
);

-- Insertion Automatique Avec PL/SQL

DECLARE
    v_numj NUMBER;
    v_nomj VARCHAR2(50);
    v_ville VARCHAR2(50);

BEGIN
    FOR i IN 1..10 LOOP -- Remplir 10 enregistrements, ajustez selon vos besoins
        v_numj := i;
        v_nomj := 'Projet' || i;
        v_ville := CASE ROUND(DBMS_RANDOM.VALUE(1, 3))
            WHEN 1 THEN 'ALGER'
            WHEN 2 THEN 'ORAN'
            ELSE 'TIZI OUZOU'
        END;

        INSERT INTO Projet (numj, nomj, ville)
        VALUES (v_numj, v_nomj, v_ville);
    END LOOP;
    COMMIT;
END;
/
```

```
-- creation de la table
-- Produit(num_p, nom_p, couleur, poids, ville)

CREATE TABLE Produit (
    num_p NUMBER,
    nom_p VARCHAR2(50),
    couleur VARCHAR2(50),
    poids NUMBER,
    ville VARCHAR2(50),
    CONSTRAINT pk_produit PRIMARY KEY (num_p)
);

-- Insertion Automatique Avec PL/SQL

DECLARE
    v_num_p NUMBER;
    v_nom_p VARCHAR2(50);
    v_couleur VARCHAR2(50);
    v_poids NUMBER;
    v_ville VARCHAR2(50);

BEGIN
    FOR i IN 1..10 LOOP -- Remplir 10 enregistrements, ajustez selon vos besoins
        v_num_p := i;
        v_nom_p := 'Produit' || i;
        v_couleur := CASE ROUND(DBMS_RANDOM.VALUE(1, 3))
            WHEN 1 THEN 'Rouge'
            WHEN 2 THEN 'Vert'
            ELSE 'Bleu'
        END;
        v_poids := ROUND(DBMS_RANDOM.VALUE(1, 10), 2);
        v_ville := CASE ROUND(DBMS_RANDOM.VALUE(1, 3))
            WHEN 1 THEN 'ALGER'
            WHEN 2 THEN 'ORAN'
            ELSE 'TIZI OUZOU'
        END;

        INSERT INTO Produit (num_p, nom_p, couleur, poids, ville)
        VALUES (v_num_p, v_nom_p, v_couleur, v_poids, v_ville);
    END LOOP;
    COMMIT;
END;
/

```

```
-- Creation De La Table  
-- Fournisseur(numf, nomf, statut, ville)
```

```
CREATE TABLE Fournisseur (  
    numf NUMBER,  
    nomf VARCHAR2(50),  
    statut VARCHAR2(50),  
    ville VARCHAR2(50),  
    CONSTRAINT pk_fournisseur PRIMARY KEY (numf)  
)
```

```
-- Insertion Automatique Avec PL/SQL
```

```
DECLARE  
    v_numf NUMBER;  
    v_nomf VARCHAR2(50);  
    v_statut VARCHAR2(50);  
    v_ville VARCHAR2(50);  
  
BEGIN  
    FOR i IN 1..10 LOOP -- Remplir 10 enregistrements, ajustez selon vos besoins  
        v_numf := i;  
        v_nomf := 'Fournisseur' || i;  
        v_statut := CASE ROUND(DBMS_RANDOM.VALUE(1, 3))  
            WHEN 1 THEN 'Actif'  
            WHEN 2 THEN 'Inactif'  
            ELSE 'Suspendu'  
        END;  
        v_ville := CASE ROUND(DBMS_RANDOM.VALUE(1, 3))  
            WHEN 1 THEN 'ALGER'  
            WHEN 2 THEN 'ORAN'  
            ELSE 'TIZI OUZOU'  
        END;  
  
        INSERT INTO Fournisseur (numf, nomf, statut, ville)  
        VALUES (v_numf, v_nomf, v_statut, v_ville);  
    END LOOP;  
    COMMIT;  
END;  
/
```

```
-- creation de la table
-- Commande(numf, nump, numj, qte), où qte est la quantité du produit

CREATE TABLE Commande (
    numf NUMBER,
    nump NUMBER,
    numj NUMBER,
    qte NUMBER,
    CONSTRAINT pk_commande PRIMARY KEY (numf, nump, numj),
    CONSTRAINT fk_commande_fournisseur FOREIGN KEY (numf) REFERENCES Fournisseur(numf),
    CONSTRAINT fk_commande_produit FOREIGN KEY (nump) REFERENCES Produit(nump),
    CONSTRAINT fk_commande_projet FOREIGN KEY (numj) REFERENCES Projet(numj)
);

-- Insertion Automatique Avec PL/SQL
```

```
DECLARE
    v_numf NUMBER;
    v_nump NUMBER;
    v_numj NUMBER;
    v_qte NUMBER;

BEGIN
    FOR i IN 1..10 LOOP -- Remplir 10 enregistrements, ajustez selon vos besoins
        -- Sélection aléatoire d'un fournisseur, produit et projet existants
        SELECT numf INTO v_numf
        FROM (
            SELECT numf
            FROM Fournisseur
            ORDER BY DBMS_RANDOM.RANDOM
        )
        WHERE ROWNUM = 1;

        SELECT nump INTO v_nump
        FROM (
            SELECT nump
            FROM Produit
            ORDER BY DBMS_RANDOM.RANDOM
        )
        WHERE ROWNUM = 1;

        SELECT numj INTO v_numj
        FROM (
            SELECT numj
            FROM Projet
            ORDER BY DBMS_RANDOM.RANDOM
        )
        WHERE ROWNUM = 1;

        -- Génération d'une quantité aléatoire
        v_qte := ROUND(DBMS_RANDOM.VALUE(1, 100));

        -- Insertion dans la table Commande
        INSERT INTO Commande (numf, nump, numj, qte)
        VALUES (v_numf, v_nump, v_numj, v_qte);
    END LOOP;
    COMMIT;
END;
/
```

# Copier La Table Sortie De L'ORCL Dans ORCL2 (Et La Supprimer Dans ORCL)

## Contrainte D'intégrité

Les contraintes d'intégrités sont isolées entre instances donc si on fait une copie de la table sortie de ORCL vers ORCL2 elle va copier que le contenu de la table sans les contraintes (clé étrangère, clé primaire,...etc). donc on doit :

- Redefinir la contrainte de la clé primaire dans ORCL2.
- Faire un trigger avant l'insertion pour vérifier la contrainte clé étrangère avec les tables randonne, participant qui sont dans ORCL.

```
</>          Sql Program 12: Copie Table Sortie Dans ORCL2 </>
-- creation de la copie
create table sortie as select * from sortie@link_orcl;

-- ajouter la contrainte de la cle etrangere
alter table sortie add CONSTRAINT pk_sortie PRIMARY KEY (idRando, idParticipant, dateSortie);

-- affichage
select * from sortie;
```

```
C:\Users\Administrator>sqlplus c##userORCL2/Nanno1234@ORCL2
SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 9 17:11:41 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Last Successful login time: Tue Dec 09 2025 15:42:58 +01:00

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> create table sortie as select * from sortie@link_orcl;
Table created.

SQL> alter table sortie add CONSTRAINT pk_sortie PRIMARY KEY (idRando, idParticipant, dateSortie);
Table altered.

SQL> select * from sortie;
-----+
IDRANDO IDPARTICIPANT DATESORTIE DUREESORTIE
-----+
      8           8 11-FEB-23          6
      2           3 11-JAN-23          4
      8           3 11-AUG-23          3
      9           10 27-JUL-23         8
      5           7 11-MAR-23          7
      7           5 23-MAR-23          2
     10           7 12-MAR-23          6
      7           1 25-JAN-23          6
      6           2 08-JAN-23          7
      5           10 30-MAY-23         3
-----+
10 rows selected.

Activate Windows
Go to Settings to activate Windows.
```

&lt;/&gt;

## Sql Program 13: Drop La Table Sortie Dans ORCL

&lt;/&gt;

```
-- suppression de la table sortie
drop table sortie;
```

```
c:\Users\Administrator>sqlplus c##userORCL/Nanno1234@ORCL
C:\Users\Administrator>SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 9 17:15:10 2025
Version 21.3.0.0.0
Copyright (c) 1982, 2021, Oracle. All rights reserved.
Last Successful login time: Tue Dec 09 2025 17:13:42 +01:00
Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL> drop table sortie;
Table dropped.
SQL>
```

&lt;/&gt;

## Sql Program 14: Creation Du Trigger Dans ORCL2

&lt;/&gt;

```
create or replace trigger check_site1_constraint
before insert on sortie -- avant insertion
for each row -- besoin de :new

declare
existrandonne int;
existparticipant int;

begin

select count(*) into existrandonne from randonnee@link_orcl where idRando = :new.idRando;
select count(*) into existparticipant from participant@link_orcl where idParticipant = :new.idParticipant;

if (existrandonne = 0 OR existparticipant = 0) then -- si l'un des id n'existe pas
-- on fait un raise application error
    raise_application_error(-20001,'violation de la contrainte de la cle etranger');
end if;

end;
/
```

```
c:\Users\Administrator>sqlplus c##userORCL2/Nanno1234@ORCL2
C:\Users\Administrator>SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 9 17:18:43 2025
Version 21.3.0.0.0
Copyright (c) 1982, 2021, Oracle. All rights reserved.
Last Successful login time: Tue Dec 09 2025 17:16:00 +01:00
Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL> create or replace trigger check_site1_constraint
2   before insert on sortie
3
4   for each row
5
6   declare
7       existrandonne int;
8       existparticipant int;
9
10  begin
11
12      select count(*) into existrandonne from randonnee@link_orcl where idRando = :new.idRando;
13      select count(*) into existparticipant from participant@link_orcl where idParticipant = :new.idParticipant;
14
15      if (existrandonne = 0 OR existparticipant = 0) then
16          raise_application_error(-20001,'violation de la contrainte de la cle etranger');
17      end if;
18
19  end;
20  /
Trigger created.
SQL>
```

```
-- afficher id randonnee
select idrando from randonnee@link_orcl;

-- afficher id participant
select idparticipant from participant@link_orcl;

-- 8 et 1 exist insertion normal
insert into sortie values (8,1,TO_DATE('01/06/2018', 'DD/MM/YYYY'),10);

-- 8 exist mais participant 11 n'existe donc raise application error
insert into sortie values (8,11,TO_DATE('01/06/2018', 'DD/MM/YYYY'),10);
```

The screenshot shows a Command Prompt window titled "Administrator: Command Prompt - sqlplus c##userORCL2/Nanno1234@ORCL". It displays two SELECT statements:

```
SQL> select idrando from randonnee@link_orcl;
IDRANDO
-----
1
2
3
4
5
6
7
8
9
10
10 rows selected.

SQL> select idparticipant from participant@link_orcl;
IDPARTICIPANT
-----
1
2
3
4
5
6
7
8
9
10
10 rows selected.
```

The screenshot shows a Command Prompt window titled "Administrator: Command Prompt - sqlplus c##userORCL2/Nanno1234@ORCL". It displays two INSERT statements followed by an error message:

```
SQL> insert into sortie values (8,1,TO_DATE('01/06/2018', 'DD/MM/YYYY'),10);
1 row created.

SQL> insert into sortie values (8,11,TO_DATE('01/06/2018', 'DD/MM/YYYY'),10);
insert into sortie values (8,11,TO_DATE('01/06/2018', 'DD/MM/YYYY'),10)
*
ERROR at line 1:
ORA-20001: violation de la contrainte de la clé étrangère
ORA-06512: at "C##USERORCL2.CHECK_SITE1_CONSTRAINT", line 11
ORA-04088: error during execution of trigger
'C##USERORCL2.CHECK_SITE1_CONSTRAINT'
```

# Creation Des Fragments Et Synonymes

## Synonymes

- un synonyme est un alias qu'on attribue à une table avec la commande suivante :  
`create or replace public synonym <alias> for <nom_table>;`
- Si il existe un alias et un nom d'une table avec les mêmes noms Oracle va prioriser le nom de la table.
- On ne peut pas accéder à un synonyme d'une autre instance.

## Fragmentation

- ORCL :
  - Fournisseur1 =  $\sigma_{ville \text{ in } ('ORAN', 'ANNABA')}$  (Fournisseur)
  - Produit1 =  $\prod_{(nom_p, couleur, poids)}$  (Produit)
  - Commande1 = Commande  $\bowtie$  Fournisseur1
- ORCL2 :
  - Fournisseur2 =  $\sigma_{ville \text{ not in } ('ORAN', 'ANNABA')}$  (Fournisseur)
  - Produit2 =  $\prod_{(ville)}$  (Produit)
  - Commande2 = Commande  $\bowtie$  Fournisseur2

```
-- fragmentation horizontal sur l'attribut ville
CREATE TABLE Fournisseur1 AS
SELECT * FROM Fournisseur WHERE ville IN ('ORAN', 'ANNABA');

-- fragmentation vertical
CREATE TABLE Produit1 AS
SELECT nump, nomp, couleur, poids FROM Produit;

-- fragmentation horizontal derive de fournisseur1
CREATE TABLE Commande1 AS
SELECT * FROM Commande
WHERE numf IN (SELECT numf FROM Fournisseur1);

CREATE OR REPLACE PUBLIC SYNONYM Fournisseur FOR Fournisseur1;

-- affiche la table fournisseur pas le fragment
-- car le nom d'une table plus prioritaire que un alias
select * from fournisseur;

CREATE OR REPLACE PUBLIC SYNONYM Produit FOR Produit1;
CREATE OR REPLACE PUBLIC SYNONYM Commande FOR Commande1;
ALTER TABLE Fournisseur RENAME TO Fournisseur_orig;
ALTER TABLE Produit RENAME TO Produit_orig;
ALTER TABLE Commande RENAME TO Commande_orig;

-- affiche le fragment car on a changer le nom de la table original
select * from fournisseur;
```

```
Administrator: Command Prompt - sqlplus c##userORCL/Nanno1234@ORCL
SQL> CREATE TABLE Fournisseur1 AS
  2  SELECT * FROM Fournisseur WHERE ville IN ('ORAN', 'ANNABA');
Table created.

SQL> CREATE TABLE Produit1 AS
  2  SELECT nump, nomp, couleur, poids FROM Produit;
Table created.

SQL> CREATE TABLE Commande1 AS
  2  SELECT * FROM Commande
  3  WHERE numf IN (SELECT numf FROM Fournisseur1);
Table created.
```

```
Administrator: Command Prompt - sqlplus c##userORCL/Nanno1234@ORCL
SQL> CREATE OR REPLACE PUBLIC SYNONYM Fournisseur FOR Fournisseur1;
Synonym created.

SQL> select * from fournisseur;
      NUMF NOMF
      -----
      STATUT
      VILLE
      -----
      1 Fournisseur1
Suspendu
ORAN

      2 Fournisseur2
Actif
ALGER

      NUMF NOMF
      -----
      STATUT
      VILLE
      -----
      3 Fournisseur3
Actif
ORAN

      4 Fournisseur4
Suspendu
```

```
Administrator: Command Prompt - sqlplus c##userORCL/Nanno1234@ORCL
SQL> ALTER TABLE Fournisseur RENAME TO Fournisseur_orig;
Table altered.

SQL> ALTER TABLE Produit RENAME TO Produit_orig;
Table altered.

SQL> ALTER TABLE Commande RENAME TO Commande_orig;
Table altered.
```

```
[Administrator: Command Prompt - sqlplus c##userORCL/Nanno1234@ORCL] -- C

SQL> select * from fournisseur;

        NUMF NOMF
        -----
        STATUT
        VILLE
        -----
          1 Fournisseur1
Suspendu
ORAN
        -----
          3 Fournisseur3
Actif
ORAN
        -----
        NUMF NOMF
        -----
        STATUT
        VILLE
        -----
          5 Fournisseur5
Suspendu
ORAN
        -----
          8 Fournisseur8
Inactif
        -----
        NUMF NOMF
        -----
        STATUT
        VILLE
        -----
ORAN
```

## Sql Program 17: Fragment ORCL2

```
-- fragmentation horizontal sur l'attribut ville
CREATE TABLE Fournisseur2 AS
SELECT * FROM Fournisseur_orig@link_orcl WHERE ville NOT IN ('ORAN', 'ANNABA');

-- fragmentation vertical
CREATE TABLE Produit2 AS
SELECT nump, ville FROM Produit_orig@link_orcl;

-- fragmentation horizontal derive de fournisseur1
CREATE TABLE Commande2 AS
SELECT * FROM Commande_orig@link_orcl
WHERE numf IN (SELECT numf FROM Fournisseur2);

CREATE OR REPLACE PUBLIC SYNONYM Fournisseur FOR Fournisseur2;
CREATE OR REPLACE PUBLIC SYNONYM Produit FOR Produit2;
CREATE OR REPLACE PUBLIC SYNONYM Commande FOR Commande2;
```

```
Administrator: Command Prompt - sqlplus c##userORCL2/Nanno1234@ORCL2

SQL> CREATE TABLE Fournisseur2 AS
 2  SELECT * FROM Fournisseur_orig@link_orcl WHERE ville NOT IN ('ORAN', 'ANNABA');

Table created.

SQL> CREATE TABLE Produit2 AS
 2  SELECT nump, ville FROM Produit_orig@link_orcl;

Table created.

SQL> CREATE TABLE Commande2 AS
 2  SELECT * FROM Commande_orig@link_orcl
 3  WHERE numf IN (SELECT numf FROM Fournisseur2);

Table created.

SQL> CREATE OR REPLACE PUBLIC SYNONYM Produit FOR Produit2;

Synonym created.

SQL> CREATE OR REPLACE PUBLIC SYNONYM Commande FOR Commande2;

Synonym created.

SQL> CREATE OR REPLACE PUBLIC SYNONYM Fournisseur FOR Fournisseur2;

Synonym created.

SQL>
```

## View Et Trigger Pour Insertion

&lt;/&gt;

Sql Program 18: Vue Et Trigger

&lt;/&gt;

```
-- creation de la vue union des fragments horizontal
CREATE OR REPLACE VIEW vw_Fournisseur AS
SELECT * FROM Fournisseur1
UNION ALL
SELECT * FROM Fournisseur2@link_orcl2;

-- trigger pour inserer dans la vue(impossible a cause de l'union)
CREATE OR REPLACE TRIGGER trg_insert_fournisseur
INSTEAD OF INSERT ON vw_Fournisseur
FOR EACH ROW
BEGIN
    IF :NEW.ville IN ('ORAN', 'ANNABA') THEN
        -- si oran ou annaba insert dans fragment 1
        INSERT INTO Fournisseur1 VALUES (:NEW.numf, :NEW.nomf, :NEW.statut, :NEW.ville);
    ELSE
        -- sinon insert dans fragment 2
        INSERT INTO Fournisseur2@link_orcl2 VALUES (:NEW.numf, :NEW.nomf, :NEW.statut,
                                         :NEW.ville);
    END IF;
END;
/

```

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt - sqlplus c##userORCL/Nanno1234@ORCL". The window displays the following SQL\*Plus session:

```
C:\Users\Administrator>sqlplus c##userORCL/Nanno1234@ORCL
SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 9 19:31:46 2025
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle. All rights reserved.

Last Successful login time: Tue Dec 09 2025 18:59:53 +01:00

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> CREATE OR REPLACE VIEW vw_Fournisseur AS
  2  SELECT * FROM Fournisseur1
  3  UNION ALL
  4  SELECT * FROM Fournisseur2@link_orcl2;
View created.

SQL> CREATE OR REPLACE TRIGGER trg_insert_fournisseur
  2  INSTEAD OF INSERT ON vw_Fournisseur
  3  FOR EACH ROW
  4  BEGIN
  5      IF :NEW.ville IN ('ORAN', 'ANNABA') THEN
  6          INSERT INTO Fournisseur1 VALUES (:NEW.numf, :NEW.nomf, :NEW.statut, :NEW.ville);
  7      ELSE
  8          INSERT INTO Fournisseur2@link_orcl2 VALUES (:NEW.numf, :NEW.nomf, :NEW.statut, :NEW.ville);
  9      END IF;
10  END;
11 /
Trigger created.
```

```
-- affichage avant insertion
select nomf from fournisseur;
select nomf from fournisseur2@link_orcl2;

-- insertion dans la vue
insert into vw_fournisseur values (99,'Fournisseur X','Actif','ALGER');
insert into vw_fournisseur values (100,'Fournisseur Y','Actif','ORAN');

-- affichage apres insertion
select nomf from fournisseur;
select nomf from fournisseur2@link_orcl2;
```

Administrator: Command Prompt - sqlplus c##userORCL/Nanno1234@ORCL

```
SQL> select nomf from fournisseur2@link_orcl2;
NOMF
-----
Fournisseur2
Fournisseur4
Fournisseur6
Fournisseur7
Fournisseur9
Fournisseur10
6 rows selected.

SQL> select nomf from fournisseur;
NOMF
-----
Fournisseur1
Fournisseur3
Fournisseur5
Fournisseur8

SQL> desc fournisseur;
Name          Null?    Type
-----          -----    -----
NUMF           NUMBER
NOMF           VARCHAR2(50)
STATUT         VARCHAR2(50)
VILLE          VARCHAR2(50)
```

Administrator: Command Prompt - sqlplus c##userORCL/Nanno1234@ORCL

```
SQL> insert into vw_fournisseur values (99,'Fournisseur X','Actif','ALGER');
1 row created.

SQL> insert into vw_fournisseur values (100,'Fournisseur Y','Actif','ORAN');
1 row created.

SQL> select nomf from fournisseur;
NOMF
-----
Fournisseur1
Fournisseur3
Fournisseur5
Fournisseur8
Fournisseur Y

SQL> select nomf from fournisseur1;
NOMF
-----
Fournisseur1
Fournisseur3
Fournisseur5
Fournisseur8
Fournisseur Y

SQL> select nomf from fournisseur2@link_orcl2;
NOMF
-----
Fournisseur2
Fournisseur4
Fournisseur6
Fournisseur7
Fournisseur9
Fournisseur10
Fournisseur X
7 rows selected.
```

Activate Windows  
Go to Settings to activate Windows.