

كلية العلوم  
ⵜⴰⵎⴻⵔⴰⵏⵜ ⴰⵎⴰⵔⴰⵏⵜ  
FACULTÉ DES SCIENCES



## UNIVERSITE IBN ZOHR FACULTE DES SCIENCES

---

### **Mini Projet BDR : Gestion des étudiants marocains et étrangers au Maroc**

---

**Base de Données Réparties**

**Présenté par : EN-NAHEL Aissam**

**Systèmes Informatiques Distribués & Big Data**

**Année universitaire 2022-2023**



# Table des matières

Table des matières .....	1
Chapitre 1 : Introduction .....	3
1. Introduction au mini projet : .....	3
Chapitre 2 : Création de la base .....	3
1. Les tables dans la base de données : .....	3
2. Fragmentation de la base de données : .....	4
3. Utilisation de Database Links : .....	4
4. Utilisation de Synonymes : .....	5
Chapitre 3 : Implémentation des sites .....	7
1. Site de l'Administrateur des bourses .....	8
- Création de la base de données .....	8
- Création les tables et les séquences.....	8
- Ajoutant des exemples de données .....	10
- Création des Database Links.....	14
- Création des synonymes .....	15
- Création des triggers .....	18
- Création des procédures .....	39
2. Site du Modérateur des bourses des étudiants marocains .....	50
- Copie des données.....	50
- Création de database link .....	52
- Création des Synonymes.....	53
- Ajoute des contraintes de base .....	54
3. Site du Modérateur des bourses des étudiants étrangers .....	55
- Copie des données.....	55
- Création de database link .....	57
- Création des Synonymes.....	58
- Ajoute des contraintes de base .....	59
Chapitre 4 : Intégration avec JDBC .....	61
L'architecture de projet JAVA JDBC .....	61
Interface console .....	62
Le code complète .....	64
Main.java.....	64
Admin_Bourse.java.....	66
Bourses_Etudiants_Marocains.java .....	77
Bourses_Etudiants_Etrangers.java .....	81

Chapitre 5 : Utilisation de SQLJ .....	86
Architecture SQLJ .....	86
Interface console .....	86
Le code complète .....	90
admin_connect.properties .....	90
maroc_connect.properties .....	91
etranger_connect.properties .....	91
MiniProjet_sqlj.sqlj .....	91

# Chapitre 1 : Introduction

## 1. Introduction au mini projet :

L'objectif principal est de développer un système de gestion des bourses pour les étudiants marocains et étrangers en implémentant une architecture de base de données réparties pour la gestion de base de données.

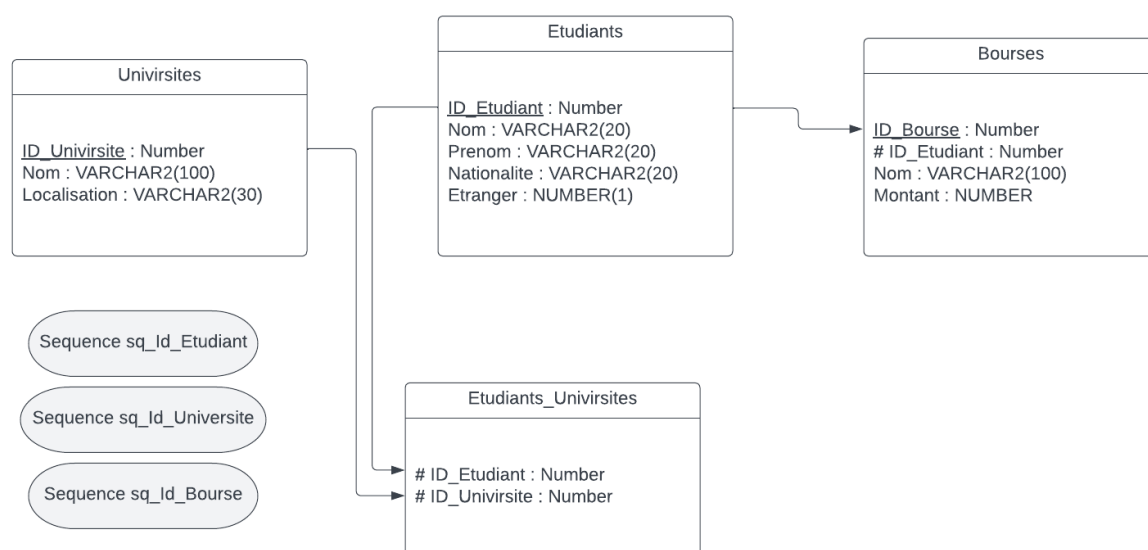
## Chapitre 2 : Création de la base

Ce chapitre se concentre sur la création de la base de données répartie et aborde différentes étapes telles que la création des tables, la fragmentation de la base de données, l'utilisation de Database Links et l'utilisation de Synonymes.

### 1. Les tables dans la base de données :

Dans le cadre de ce mini projet, j'ai procédé à la création des tables suivantes :

- Table "**Etudiants**" : cette table contient les informations relatives aux étudiants.
- Table "**Universités**" : cette table contient les informations des universités.
- Table "**Etudiants\_Universités**" : cette table est utilisée pour établir une liaison entre un étudiant et une université.
- Table "**Bourses**" : cette table permet d'établir une correspondance entre un étudiant et la bourse qui lui est attribuée.



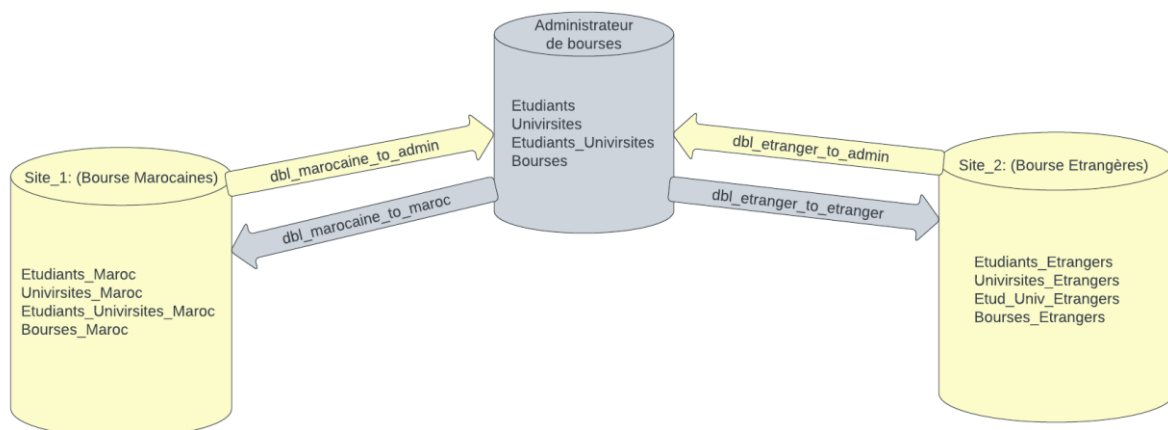
En plus de cela, j'ai également créé les séquences suivantes afin de garantir l'unicité des identifiants :

- Séquence "**sq\_Id\_Etudiant**" : cette séquence est utilisée pour générer des identifiants uniques pour les étudiants.
- Séquence "**sq\_Id\_Universite**" : cette séquence est utilisée pour générer des identifiants uniques pour les universités.

- Séquence "sq\_Id\_Bourse" : cette séquence est utilisée pour générer des identifiants uniques pour les bourses.

## 2. Fragmentation de la base de données :

Dans le cadre de ce mini projet, j'ai choisi d'opter la technique de fragmentation horizontale en fonction du type d'étudiant, à savoir les étudiants marocains et les étudiants étrangers. Je vais diviser les données des tables du site **Administrateur des Bourses** et les répartir sur les sites **Modérateur des Bourses des Etudiants Marocains** et **Modérateur des Bourses des Etudiants Etrangers**.



## 3. Utilisation de Database Links :

Dans le cadre de ce projet, je vais configurer les Database Links afin de permettre la communication entre les 3 sites, nous aurons besoin de 4 Database Links:

- Database Link "**dbl\_marocaine\_to\_admin**" : Ce lien sera établi depuis le site 2 "Modérateur des Bourses des Etudiants Marocains" vers le site 1 "Administrateur de Bourses". Il permettra au site 2 de communiquer avec le site 1.
- Database Link "**dbl\_marocaine\_to\_maroc**" : Ce lien sera établi depuis le site 1 "Administrateur de Bourses" vers le site 2 "Modérateur des Bourses des Etudiants Marocains". Il permettra au site 1 de communiquer avec le site 2.
- Database Link "**dbl\_etrange\_to\_admin**" : Ce lien sera établi depuis le site 3 "Modérateur des Bourses des Etudiants Etrangers" vers le site 1 "Administrateur de Bourses". Il permettra au site 3 de communiquer avec le site 1.
- Database Link "**dbl\_etrange\_to\_etrange**" : Ce lien sera établi depuis le site 1 "Administrateur de Bourses" vers le site 3 "Modérateur des Bourses des Etudiants Etrangers". Il permettra au site 1 de communiquer avec le site 3.

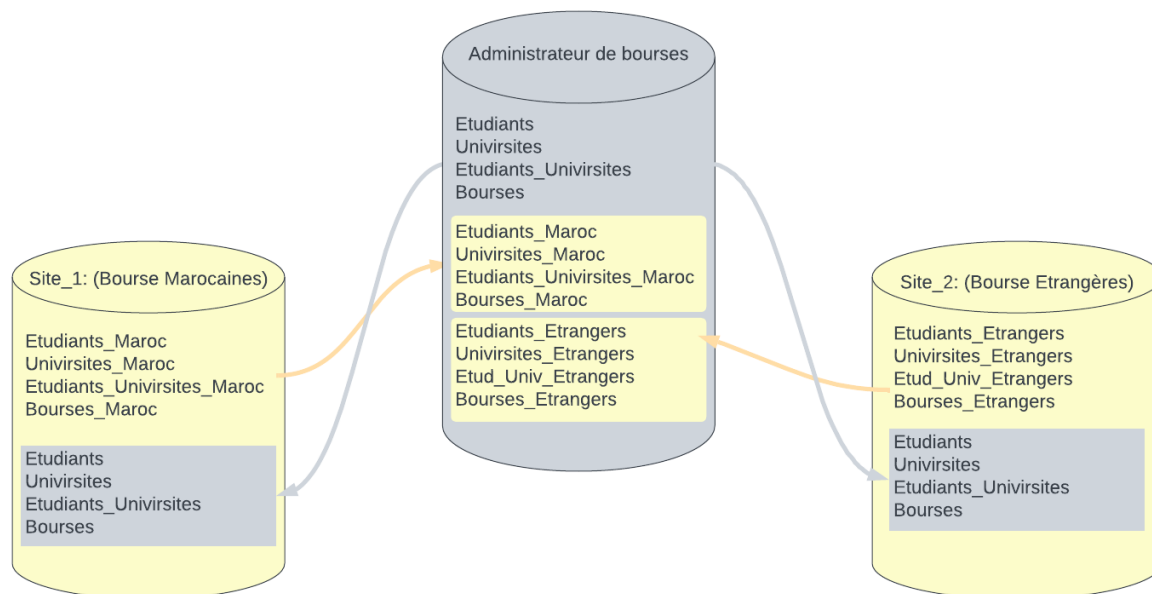
## 4. Utilisation de Synonymes :

Dans le site 1 "Administrateur de Bourses", je vais créer les synonymes suivants pour faciliter l'accès aux tables situées sur le site 2 via le Database Link "dbl\_marocaine\_to\_maroc", et accès aux tables situées sur le site 3 via le Database Link "dbl\_etrananger\_to\_etrange" :

- Synonyme "**Etudiants\_Maroc**" : Pour accéder à la table "Etudiants\_Maroc" située sur le site 2.
- Synonyme "**Universites\_Maroc**" : Pour accéder à la table "Universites\_Maroc" située sur le site 2.
- Synonyme "**Etud\_Univ\_Maroc**" : Pour accéder à la table "Etud\_Univ\_Maroc" située sur le site 2.
- Synonyme "**Bourses\_Maroc**" : Pour accéder à la table "Bourses\_Maroc" située sur le site 2.
- Synonyme "**Etudiants\_Etrangers**" : Pour accéder à la table "Etudiants\_Etrangers" située sur le site 3.
- Synonyme "**Universites\_Etrangers**" : Pour accéder à la table "Universites\_Etrangers" située sur le site 3.
- Synonyme "**Etud\_Univ\_Etrangers**" : Pour accéder à la table "Etud\_Univ\_Etrangers" située sur le site 3.
- Synonyme "**Bourses\_Etrangers**" : Pour accéder à la table "Bourses\_Etrangers" située sur le site 3.

Dans le site 2 "Modérateur des bourses des étudiants marocains" et le site 3 "Modérateur des bourses des étudiants étrangers", je vais créer les synonymes suivants pour faciliter l'accès aux tables et aux séquences situées sur le site 1 via le Database Link "dbl\_marocaine\_to\_admin" le Database Link "dbl\_etrananger\_to\_admin":

- Synonyme "**Bourses\_Etrangers**" : Pour accéder à la table "Bourses\_Etrangers" située sur le site 1.
- Synonyme "**Universites**" : Pour accéder à la table "Universites" située sur le site 1.
- Synonyme "**Etudiants\_Universites**" : Pour accéder à la table "Etudiants\_Universites" située sur le site 1.
- Synonyme "**Bourses**" : Pour accéder à la table "Bourses" située sur le site 1
- Synonyme "**sq\_Id\_Etudiant**" : Pour accéder à la séquence "sq\_Id\_Etudiant" située sur le site 1.
- Synonyme "**sq\_Id\_Univetsite**" : Pour accéder à la séquence "sq\_Id\_Univetsite" située sur le site 1.
- Synonyme "**sq\_Id\_Bourse**" : Pour accéder à la séquence "sq\_Id\_Bourse" située sur le site 1.





## Chapitre 3 : Implémentation des sites

Dans ce chapitre, je vais procéder à l'implémentation des 3 sites de la base de données, je vais mettre en place chaque site en commençant par la création de la base de données, la copie des données, puis la création des synonymes, des déclencheurs et des procédures.

Tout d'abord, dans le système de base de données Oracle, je vais créer trois utilisateurs qui représenteront les trois sites principaux : "**Administrateur de Bourses**", "**Modérateur des bourses des étudiants marocains**" et "**Modérateur des bourses des étudiants étrangers**".

sqlplus system/aissam

```
CREATE tablespace tbs_mini_prj datafile  
'C:\mini_project_BDR\tbs_mini_prj.dbf' size 500M;
```

```
CREATE USER admin_bourse identified by aissam default tablespace  
tbs_mini_prj quota unlimited on tbs_mini_prj;
```

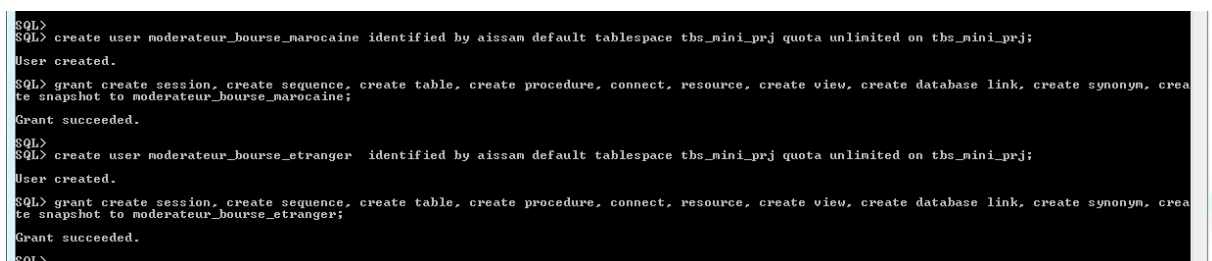
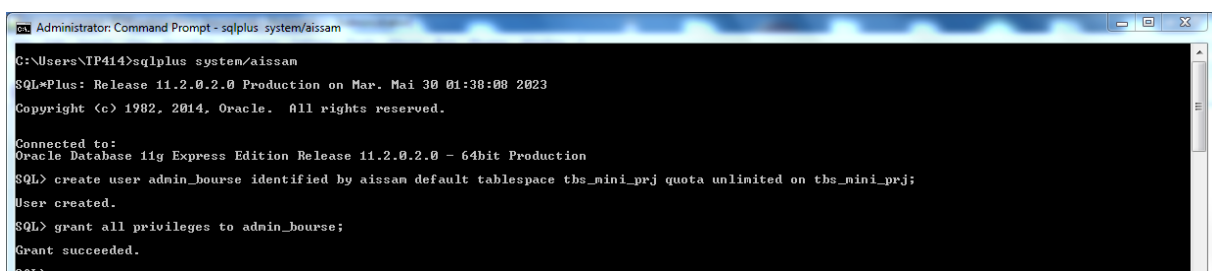
```
Grant all privileges to admin_bourse;
```

```
CREATE user modérateur_bourse_marocaine identified by aissam  
default tablespace tbs_mini_prj quota unlimited on tbs_mini_prj;
```

```
Grant CREATE session, CREATE sequence, CREATE table, CREATE  
procedure, connect, resource, CREATE view, CREATE database link,  
CREATE synonym, CREATE snapshot to modérateur_bourse_marocaine;
```

```
CREATE user modérateur_bourse_etranger identified by aissam default  
tablespace tbs_mini_prj quota unlimited on tbs_mini_prj;
```

```
Grant CREATE session, CREATE sequence, CREATE table, CREATE  
procedure, connect, resource, CREATE view, CREATE database link,  
CREATE synonym, CREATE snapshot to modérateur_bourse_etranger;
```



## 1. Site de l'Administrateur des bourses

### - Création de la base de données

Dans le site de l'Administrateur des bourses, je vais commencer par la création de la base de données. Je vais créer les tables nécessaires, ainsi que les séquences qui seront utilisées pour générer des identifiants uniques, ensuite, je vais implémenter les tables en ajoutant des exemples de données.

Par la suite, je vais créer les deux Database Links ("dbl\_marocaine\_to\_admin" et "dbl\_etranger\_to\_admin") pour établir une connexion vers les autres sites pour accéder aux données situées sur les autres sites, en plus je vais créer des synonymes.

Enfin, je vais mettre en place des triggers et des procédures pour gérer des opérations spécifiques, telles que l'insertion et la suppression des données.

### - Création des tables et des séquences

sqlplus admin\_bourse/aissam  
SELECT \* FROM cat;

```
SQL> connect admin_bourse/aissam
Connected.
SQL> select * from cat;

no rows selected

SQL>
```

```
CREATE TABLE Etudiants(
  ID_etudiant NUMBER primary key,
  Nom VARCHAR2(20),
  Prenom VARCHAR2(20),
  Nationalite VARCHAR2(20),
  Etranger NUMBER(1) CHECK(Etranger IN(0, 1))
);
```

```
CREATE TABLE Universites(
  ID_universite NUMBER primary key,
  Nom VARCHAR2(100),
  Localisation VARCHAR2(30)
);
```

```
CREATE TABLE Etudiants_Universites(
  ID_etudiant NUMBER,
  ID_universite NUMBER,
  CONSTRAINT PK_Etudiant_Universite PRIMARY KEY(ID_etudiant,
  ID_universite),
  CONSTRAINT FK_EtudUniv_Etudiants FOREIGN KEY(ID_etudiant)
  REFERENCES Etudiants(ID_etudiant),
  CONSTRAINT FK_EtudUniv_Universite FOREIGN KEY(ID_universite)
  REFERENCES Universites(ID_universite)
```

);

```
CREATE TABLE Bourses(
  ID_bourse NUMBER primary key,
  ID_etudiant NUMBER,
  Nom VARCHAR2(100),
  Montant NUMBER,
  CONSTRAINT FK_Brs_Etud FOREIGN KEY(ID_etudiant) REFERENCES
Etudiants(ID_etudiant)
);
```

```
SQL>
SQL> CREATE TABLE Etudiants(
2   ID_etudiant NUMBER primary key,
3   Nom VARCHAR2(20),
4   Prenom VARCHAR2(20),
5   Nationalite VARCHAR2(20),
6   Etranger NUMBER(1) CHECK(Etranger IN(0, 1))
7  );
Table created.
SQL>
SQL> CREATE TABLE Universites(
2   ID_universite NUMBER primary key,
3   Nom VARCHAR2(100),
4   Localisation VARCHAR2(30)
5  );
Table created.
SQL>
SQL> CREATE TABLE Etudiants_Universites(
2   ID_etudiant NUMBER,
3   ID_universite NUMBER,
4   CONSTRAINT PK_Etudiant_Universite PRIMARY KEY(ID_etudiant, ID_universite),
5   CONSTRAINT FK_EtudUniv_Etudiants FOREIGN KEY(ID_etudiant) REFERENCES Etudiants(ID_etudiant),
6   CONSTRAINT FK_EtudUniv_Universite FOREIGN KEY(ID_universite) REFERENCES Universites(ID_universite)
7  );
Table created.
SQL>
SQL> CREATE TABLE Bourses(
2   ID_bourse NUMBER primary key,
3   ID_etudiant NUMBER,
4   Nom VARCHAR2(100),
5   Montant NUMBER,
6   CONSTRAINT FK_Brs_Etud FOREIGN KEY(ID_etudiant) REFERENCES Etudiants(ID_etudiant)
7  );
Table created.
SQL>
SQL> select * from cat;
```

SELECT \* FROM cat;

```
SQL>
SQL> select * from cat;
TABLE_NAME                                TABLE_TYPE
-----
BOURSES                                  TABLE
ETUDIANTS                               TABLE
ETUDIANTS_UNIVERSITES                   TABLE
UNIVERSITES                             TABLE
SQL>
```

CREATE sequence **sq\_Id\_Etudiant** increment by 1 start with 1;  
CREATE sequence **sq\_Id\_Univetsite** increment by 1 start with 1;  
CREATE sequence **sq\_Id\_Bourse** increment by 1 start with 1;

```

SQL>
SQL> create sequence sq_Id_Etudiant increment by 1 start with 1;
Sequence created.
SQL>
SQL> create sequence sq_Id_Univetsite increment by 1 start with 1;
Sequence created.
SQL>
SQL> create sequence sq_Id_Bourse increment by 1 start with 1;
Sequence created.
SQL>

```

### - Ajoutant des exemples de données

```

INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'EN-NAHEL',
'Aissam', 'Marocain', 0);

```

```

INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'Amina', 'Ben-
Ali', 'Marocaine', 0);

```

```

INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'Karim', 'Ben-
Moussa', 'Marocain', 0);

```

```

INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'Fatima', 'Ben-
Mohammed', 'Marocaine', 0);

```

```

INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'Elon', 'Musk',
'Americaïn', 1);

```

```

INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'John', 'Smith',
'Britannique', 1);

```

```

INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'Mark',
'Zuckerberg', 'Chinois', 1);

```

```

set lines 150
SELECT * FROM Etudiants;

```

```

Administrator: Command Prompt - sqlplus system/aissam

SQL> insert into Etudiants values(sq_Id_Etudiant.nextval, 'EN-NAHEL', 'Aissam', 'Marocain', 0);
1 row created.
SQL> insert into Etudiants values(sq_Id_Etudiant.nextval, 'Amina', 'Ben-Ali', 'Marocaine', 0);
1 row created.
SQL> insert into Etudiants values(sq_Id_Etudiant.nextval, 'Karim', 'Ben-Moussa', 'Marocain', 0);
1 row created.
SQL> insert into Etudiants values(sq_Id_Etudiant.nextval, 'Fatima', 'Ben-Mohammed', 'Marocaine', 0);
1 row created.
SQL> insert into Etudiants values(sq_Id_Etudiant.nextval, 'Elon', 'Musk', 'Américain', 1);
1 row created.
SQL> insert into Etudiants values(sq_Id_Etudiant.nextval, 'John', 'Smith', 'Britannique', 1);
1 row created.

SQL> insert into Etudiants values(sq_Id_Etudiant.nextval, 'Mark', 'Zuckerberg', 'Chinois', 1);
1 row created.
SQL> set lines 150
SQL> select * from Etudiants;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1

```

7 rows selected.

```

INSERT INTO Universites VALUES(sq\_Id\_Univetsite.NEXTVAL, 'Universite Ibn Zohr', 'Agadir');

INSERT INTO Universites VALUES(sq\_Id\_Univetsite.NEXTVAL, 'Universite Cadi Ayyad', 'Marrakech');

INSERT INTO Universites VALUES(sq\_Id\_Univetsite.NEXTVAL, 'Universite Hassan II', 'Casablanca');

INSERT INTO Universites VALUES(sq\_Id\_Univetsite.NEXTVAL, 'Universite Mohammed V', 'Rabat');

INSERT INTO Universites VALUES(sq\_Id\_Univetsite.NEXTVAL, 'Universite Moulay Ismail', 'Meknes');

INSERT INTO Universites VALUES(sq\_Id\_Univetsite.NEXTVAL, 'Universite Al Quaraouiyine', 'Fes');

col Nom for A40

SELECT \* FROM Universites;

```
SQL> insert into Universites values(sq_Id_Univetsite.nextval, 'Université Ibn Zohr', 'Agadir');
1 row created.
SQL> insert into Universites values(sq_Id_Univetsite.nextval, 'Université Cadi Ayyad', 'Marrakech');
1 row created.
SQL> insert into Universites values(sq_Id_Univetsite.nextval, 'Université Hassan II', 'Casablanca');
1 row created.
SQL> insert into Universites values(sq_Id_Univetsite.nextval, 'Université Mohammed U', 'Rabat');
1 row created.
SQL> insert into Universites values(sq_Id_Univetsite.nextval, 'Université Moulay Ismail', 'Meknès');
1 row created.
```

```
SQL> insert into Universites values(sq_Id_Univetsite.nextval, 'Université Al Quaraouiyine', 'Fès');
1 row created.
SQL> col Nom for A40
SQL> select * from Universites;
```

ID_UNIVERSITE	NOM	LOCALISATION
1	Université Ibn Zohr	Agadir
2	Université Cadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed U	Rabat
5	Université Moulay Ismail	Meknès
6	Université Al Quaraouiyine	Fès

```
6 rows selected.
SQL>
```

```
INSERT INTO Etudiants_Universites VALUES(1, 1);
INSERT INTO Etudiants_Universites VALUES(2, 2);
INSERT INTO Etudiants_Universites VALUES(3, 3);
INSERT INTO Etudiants_Universites VALUES(4, 2);
INSERT INTO Etudiants_Universites VALUES(5, 4);
INSERT INTO Etudiants_Universites VALUES(6, 5);
INSERT INTO Etudiants_Universites VALUES(7, 6);
SELECT * FROM Etudiants_Universites;
```

```
SQL> insert into Etudiants_Universites values(1, 1);
1 row created.
SQL> insert into Etudiants_Universites values(2, 2);
1 row created.
SQL> insert into Etudiants_Universites values(3, 3);
1 row created.
SQL> insert into Etudiants_Universites values(4, 2);
1 row created.
SQL> insert into Etudiants_Universites values(5, 4);
1 row created.
SQL> insert into Etudiants_Universites values(6, 5);
1 row created.
SQL> insert into Etudiants_Universites values(7, 6);
1 row created.
SQL> select * from Etudiants_Universites;
ID_ETUDIANT ID_UNIVERSITE
-----
1          1
2          2
3          3
4          2
5          4
6          5
7          6
7 rows selected.
SQL>
```

INSERT INTO Bourses VALUES(sq\_Id\_Bourse.NEXTVAL, 1, 'Bourses Minhaty', 1500);

INSERT INTO Bourses VALUES(sq\_Id\_Bourse.NEXTVAL, 2, 'Bourse Istihqaq', 3000);

INSERT INTO Bourses VALUES(sq\_Id\_Bourse.NEXTVAL, 3, 'Bourses FME', 2000);

INSERT INTO Bourses VALUES(sq\_Id\_Bourse.NEXTVAL, 4, 'Bourse AMCI', 4000);

INSERT INTO Bourses VALUES(sq\_Id\_Bourse.NEXTVAL, 5, 'Bourses Minhaty', 1500);

INSERT INTO Bourses VALUES(sq\_Id\_Bourse.NEXTVAL, 6, 'Bourse AMCI', 4000);

INSERT INTO Bourses VALUES(sq\_Id\_Bourse.NEXTVAL, 7, 'Bourses Minhaty', 1500);

SELECT \* FROM Bourses;

commit;

```
SQL>
SQL> insert into Bourses values(sq_Id_Bourse.nextval, 1, 'Bourses Minhaty', 1500);
1 row created.
SQL> insert into Bourses values(sq_Id_Bourse.nextval, 2, 'Bourse Istihqaq', 3000);
1 row created.
SQL> insert into Bourses values(sq_Id_Bourse.nextval, 3, 'Bourses FME', 2000);
1 row created.
SQL> insert into Bourses values(sq_Id_Bourse.nextval, 4, 'Bourse AMCI', 4000);
1 row created.
SQL> insert into Bourses values(sq_Id_Bourse.nextval, 5, 'Bourses Minhaty', 1500);
1 row created.
SQL> insert into Bourses values(sq_Id_Bourse.nextval, 6, 'Bourse AMCI', 4000);
1 row created.
```

```
SQL> insert into Bourses values(sq_Id_Bourse.nextval, 7, 'Bourses Minhaty', 1500);
1 row created.
SQL> select * from Bourses;
  ID_BOURSE ID_ETUDIANT NOM                                MONTANT
-----
         1          1 Bourses Minhaty                                1500
         2          2 Bourse Istihqaq                                3000
         3          3 Bourses FME                                2000
         4          4 Bourse AMCI                                4000
         5          5 Bourses Minhaty                                1500
         6          6 Bourse AMCI                                4000
         7          7 Bourses Minhaty                                1500

7 rows selected.
SQL>
SQL> commit;
Commit complete.
```

### - Création des Database Links

SELECT \* FROM Etudiants\_Maroc;

CREATE **database link** **dbl\_marocaine\_to\_maroc** connect to  
moderateur\_bourse\_marocaine identified by aissam using 'xe';

col db\_link for A30

col host for A20

SELECT \* FROM USER\_DB\_LINKS;

SELECT \* FROM Etudiants\_Maroc@**dbl\_marocaine\_to\_maroc**;



```

Administrator: Command Prompt - sqlplus system/aissam

SQL> select * from Etudiants_Maroc;
select * from Etudiants_Maroc
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> create database link dbl_marocaine_to_maroc connect to modérateur_bourse_marocaine identified by aissam using 'xe';
Database link created.

SQL>
SQL> col db_link for a30
SQL> col host for a20
SQL> select * from USER_DB_LINKS;

```

DB_LINK	USERNAME	PASSWORD	HOST	CREATED
DBL_MAROCAINE_TO_MAROC	MODÉRATEUR_BOURSE_MAROCAINE		xe	30/05/23

```

SQL>
SQL> select * from Etudiants_Maroc@dbl_marocaine_to_maroc;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karin	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0

```

SQL>

```

SELECT \* FROM Etudiants\_Etrangers;

CREATE **database link dbl\_etranger\_to\_etranger** connect to modérateur\_bourse\_etranger identified by aissam using 'xe';

SELECT \* FROM USER\_DB\_LINKS;

SELECT \* FROM Etudiants\_Etrangers@**dbl\_etranger\_to\_etranger**;

```

SQL>
SQL> select * from Etudiants_Etrangers;
select * from Etudiants_Etrangers
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> create database link dbl_etranger_to_etranger connect to modérateur_bourse_etranger identified by aissam using 'xe';
Database link created.

```

```

Select Administrator: Command Prompt - sqlplus system/aissam

Database link created.
SQL> select * from USER_DB_LINKS;

```

DB_LINK	USERNAME	PASSWORD	HOST	CREATED
DBL_ETRANGER_TO_ETRANGER	MODÉRATEUR_BOURSE_ETRANGER		xe	30/05/23
DBL_MAROCAINE_TO_MAROC	MODÉRATEUR_BOURSE_MAROCAINE		xe	30/05/23

```

SQL>
SQL> select * from Etudiants_Etrangers@dbl_etranger_to_etranger;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1

```

SQL>

```

## - Création des synonymes

CREATE OR REPLACE **synonym Etudiants\_Maroc** FOR modérateur\_bourse\_marocaine.Etudiants\_Maroc@dbl\_marocaine\_to\_maroc;  
 SELECT \* FROM **Etudiants\_Maroc**;

```
CREATE OR REPLACE synonym Universites_Maroc FOR
moderateur_bourse_marocaine.Universites_Maroc@db1_marocaine_to_ma
roc;
SELECT * FROM Universites_Maroc;
```

```
SQL>
SQL>
SQL> create or REPLACE synonym Etudiants_Maroc FOR moderateur_bourse_marocaine.Etudiants_Maroc@db1_marocaine_to_maroc;
Synonym created.
SQL> select * from Etudiants_Maroc;
ID_ETUDIANT  NOM                                PRENOM                                NATIONALITE                                ETRANGER
-----
1  EN-NAHEL                                Aissam                                Marocain                                0
2  Amina                                Ben-Ali                                Marocaine                                0
3  Karim                                Ben-Moussa                                Marocain                                0
4  Fatima                                Ben-Mohammed                                Marocaine                                0
SQL>
SQL> create or REPLACE synonym Universites_Maroc FOR moderateur_bourse_marocaine.Universites_Maroc@db1_marocaine_to_maroc;
Synonym created.
SQL> select * from Universites_Maroc;
ID_UNIVERSITE  NOM                                LOCALISATION
-----
1  Université Ibn Zohr                                Agadir
2  Université Cadi Ayyad                                Marrakech
3  Université Hassan II                                Casablanca
4  Université Mohammed V                                Rabat
5  Université Moulay Ismail                                Meknès
6  Université Al Quaraouiyine                                Fès
```

```
CREATE OR REPLACE synonym Etud_Univ_Maroc FOR
moderateur_bourse_marocaine.Etud_Univ_Maroc@db1_marocaine_to_mar
oc;
SELECT * FROM Etud_Univ_Maroc;
```

```
CREATE OR REPLACE synonym Bourses_Maroc FOR
moderateur_bourse_marocaine.Bourses_Maroc@db1_marocaine_to_maroc
;
SELECT * FROM Bourses_Maroc;
```

```
SQL>
SQL> create or REPLACE synonym Etud_Univ_Maroc FOR moderateur_bourse_marocaine.Etud_Univ_Maroc@db1_marocaine_to_maroc;
Synonym created.
SQL> select * from Etud_Univ_Maroc;
ID_ETUDIANT  ID_UNIVERSITE
-----
1            1
2            2
3            3
4            2
SQL>
SQL> create or REPLACE synonym Bourses_Maroc FOR moderateur_bourse_marocaine.Bourses_Maroc@db1_marocaine_to_maroc;
Synonym created.
SQL> select * from Bourses_Maroc;
ID_BOURSE  ID_ETUDIANT  NOM                                MONTANT
-----
1            1  Bourses Minhaty                                1500
2            2  Bourse Istihqag                                3000
3            3  Bourses FME                                    2000
4            4  Bourse AMCI                                    4000
SQL>
```

```
CREATE OR REPLACE synonym Etudiants_Etrangers FOR
moderateur_bourse_etranger.Etudiants_Etrangers@db1_etranger_to_etran
ger;
SELECT * FROM Etudiants_Etrangers;
```

```
SQL>
SQL> create or REPLACE synonym Etudiants_Etrangers FOR modérateur_bourse_etranger.Etudiants_Etrangers@db1_etranger_to_etranger;
Synonym created.
SQL> select * from Etudiants_Etrangers;
```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1

```
SQL>
```

CREATE OR REPLACE **synonym Universites\_Etrangers** FOR  
modérateur\_bourse\_etranger.Universites\_Etrangers@db1\_etranger\_to\_etranger;  
SELECT \* FROM **Universites\_Etrangers**;

CREATE OR REPLACE **synonym Etud\_Univ\_Etrangers** FOR  
modérateur\_bourse\_etranger.Etud\_Univ\_Etrangers@db1\_etranger\_to\_etranger;  
SELECT \* FROM **Etud\_Univ\_Etrangers**;

CREATE OR REPLACE **synonym Bourses\_Etrangers** FOR  
modérateur\_bourse\_etranger.Bourses\_Etrangers@db1\_etranger\_to\_etranger;  
SELECT \* FROM **Bourses\_Etrangers**;

```
SQL>
SQL> create or REPLACE synonym Universites_Etrangers FOR modérateur_bourse_etranger.Universites_Etrangers@db1_etranger_to_etranger;
Synonym created.
SQL> select * from Universites_Etrangers;
```

ID_UNIVERSITE	NOM	LOCALISATION
1	Université Ibn Zohr	Agadir
2	Université Cadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed V	Rabat
5	Université Moulay Ismail	Meknès
6	Université Al Quaraouiyine	Fès

```
6 rows selected.
SQL>
SQL> create or REPLACE synonym Etud_Univ_Etrangers FOR modérateur_bourse_etranger.Etud_Univ_Etrangers@db1_etranger_to_etranger;
Synonym created.
SQL> select * from Etud_Univ_Etrangers;
```

ID_ETUDIANT	ID_UNIVERSITE
5	4
6	5
7	6

```
SQL>
SQL> create or REPLACE synonym Bourses_Etrangers FOR modérateur_bourse_etranger.Bourses_Etrangers@db1_etranger_to_etranger;
Synonym created.
SQL> select * from Bourses_Etrangers;
```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
5	5	Bourses Minhaty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhaty	1500

```
SQL>
```

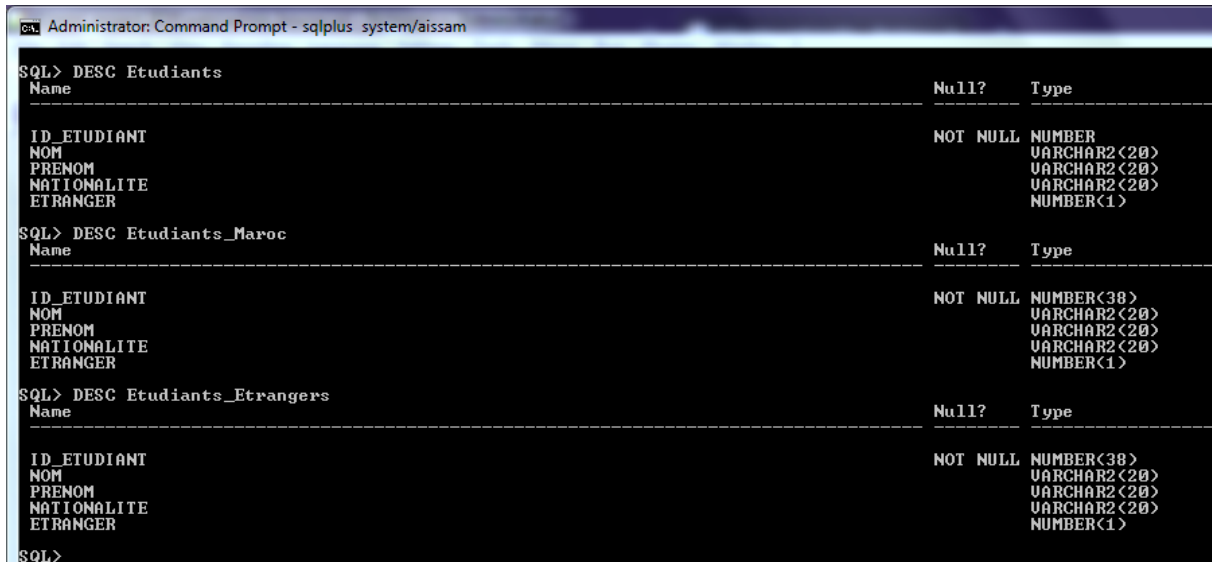
## - Création des triggers

### Création du trigger TRIGGER\_INSERT\_etudiant :

DESC Etudiants

DESC Etudiants\_Maroc

DESC Etudiants\_Etrangers



Administrator: Command Prompt - sqlplus system/aissam

```
SQL> DESC Etudiants
Name                                         Null?    Type
-----
ID_ETUDIANT                                NOT NULL NUMBER
NOM                                         VARCHAR2(20)
PRENOM                                     VARCHAR2(20)
NATIONALITE                                VARCHAR2(20)
ETRANGER                                   NUMBER(1)

SQL> DESC Etudiants_Maroc
Name                                         Null?    Type
-----
ID_ETUDIANT                                NOT NULL NUMBER(38)
NOM                                         VARCHAR2(20)
PRENOM                                     VARCHAR2(20)
NATIONALITE                                VARCHAR2(20)
ETRANGER                                   NUMBER(1)

SQL> DESC Etudiants_Etrangers
Name                                         Null?    Type
-----
ID_ETUDIANT                                NOT NULL NUMBER(38)
NOM                                         VARCHAR2(20)
PRENOM                                     VARCHAR2(20)
NATIONALITE                                VARCHAR2(20)
ETRANGER                                   NUMBER(1)

SQL>
```

CREATE OR REPLACE **TRIGGER TRIGGER\_INSERT\_etudiant**

BEFORE INSERT ON Etudiants

FOR EACH ROW

DECLARE

    n NUMBER;

    etudiant\_duplique EXCEPTION;

BEGIN

    SELECT COUNT(\*) INTO n FROM Etudiants WHERE Nom = :NEW.Nom AND  
    Prenom = :NEW.Prenom;

    IF n = 0 THEN

        IF :NEW.Etranger = 0 THEN

            -- verifier si la ligne inseree est pour un etudiant marocain

            -- Insérer la ligne dans la table Etudiants\_Maroc

            INSERT INTO Etudiants\_Maroc VALUES (:NEW.ID\_etudiant, :NEW.Nom,  
:NEW.Prenom, :NEW.Nationalite, :NEW.Etranger);

        ELSE

            -- si la ligne inseree est pour un etudiant etranger

            -- Insérer la ligne dans la table Etudiants\_Etranger

            INSERT INTO Etudiants\_Etrangers VALUES (:NEW.ID\_etudiant,  
:NEW.Nom, :NEW.Prenom, :NEW.Nationalite, :NEW.Etranger);

        END IF;

    ELSE

        -- si une ligne avec le même nom et prenom existe déjà

        RAISE etudiant\_duplique;

    END IF;

EXCEPTION

    WHEN etudiant\_duplique THEN

        RAISE\_APPLICATION\_ERROR(-20001, '!!! L''etudiant existe déjà !!!');

END;  
/

```
SQL>
SQL>
SQL> CREATE OR REPLACE TRIGGER TRIGGER_INSERT_etudiant
2 BEFORE INSERT ON Etudiants
3 FOR EACH ROW
4 DECLARE
5     n NUMBER;
6     etudiant_duplique EXCEPTION;
7 BEGIN
8     SELECT COUNT(*) INTO n FROM Etudiants WHERE Nom = :NEW.Nom AND Prenom = :NEW.Prenom;
9     IF n = 0 THEN
10        IF :NEW.Etranger = 0 THEN
11            -- verifier si la ligne inseree est pour un etudiant marocain
12            -- Insérer la ligne dans la table Etudiants_Maroc
13            INSERT INTO Etudiants_Maroc VALUES (:NEW.ID_etudiant, :NEW.Nom, :NEW.Prenom, :NEW.Nationalite, :NEW.Etranger);
14        ELSE
15            -- si la ligne inseree est pour un etudiant etranger
16            -- Insérer la ligne dans la table Etudiants_Etranger
17            INSERT INTO Etudiants_Etrangers VALUES (:NEW.ID_etudiant, :NEW.Nom, :NEW.Prenom, :NEW.Nationalite, :NEW.Etranger);
18        END IF;
19    ELSE
20        -- si une ligne avec le même nom et prenom existe deja
21        RAISE etudiant_duplique;
22    END IF;
23 EXCEPTION
24 WHEN etudiant_duplique THEN
25     RAISE_APPLICATION_ERROR(-20001, '!!! L''etudiant existe deja !!!');
26 END;
27 /
Trigger created.
```

### Test de trigger TRIGGER INSERT etudiant :

```
SELECT * FROM Etudiants;
SELECT * FROM Etudiants_Maroc;
SELECT * FROM Etudiants_Etrangers;
```

```
INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'EN-NAHEL', 'Aissam',
'Marocain', 0);
```

```
SQL>
SQL>
SQL> SELECT * FROM Etudiants;
```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1
9	EN-NAHEL	Hamza	Marocain	0

8 rows selected.

```
8 rows selected.
SQL> SELECT * FROM Etudiants_Maroc;
```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
9	EN-NAHEL	Hamza	Marocain	0

```
SQL> SELECT * FROM Etudiants_Etrangers;
```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1

```
SQL>
SQL>
SQL>
SQL> INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'EN-NAHEL', 'Aissam', 'Marocain', 0);
INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'EN-NAHEL', 'Aissam', 'Marocain', 0)
*
ERROR at line 1:
ORA-20001: !!! L''etudiant existe deja !!!
ORA-06512: at "ADMIN_BOURSE.TRIGGER_INSERT_ETUDIANT", line 22
ORA-04088: error during execution of trigger 'ADMIN_BOURSE.TRIGGER_INSERT_ETUDIANT'
```

```
INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'EN-NAHEL', 'Hassan',
'Marocain', 0);
```

```

SELECT * FROM Etudiants;
SELECT * FROM Etudiants_Maroc;
SELECT * FROM Etudiants_Etrangers;

```

```

SQL>
SQL> INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'EN-NAHEL', 'Hassan', 'Marocain', 0);
1 row created.

SQL>
SQL> SELECT * FROM Etudiants;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0

```

9 rows selected.

```

```

SQL> SELECT * FROM Etudiants_Maroc;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0

```

6 rows selected.

SQL> SELECT * FROM Etudiants_Etrangers;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1

```

INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'Bill', 'Gates', 'Allemand', 1);

```

```

SELECT * FROM Etudiants;
SELECT * FROM Etudiants_Maroc;
SELECT * FROM Etudiants_Etrangers;

```

```

SQL>
SQL> INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'Bill', 'Gates', 'Allemand', 1);
1 row created.

SQL>
SQL> SELECT * FROM Etudiants;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0
12	Bill	Gates	Allemand	1

```

10 rows selected.

```

```
SQL> SELECT * FROM Etudiants_Maroc;
```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amira	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0

6 rows selected.

```
SQL> SELECT * FROM Etudiants_Etrangers;
```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1
12	Bill	Gates	Allemand	1

```
SQL>
```

### Création du trigger TRIGGER\_INSERT universite:

DESC Universites

DESC Universites\_Maroc

DESC Universites\_Etrangers

```
SQL> DESC Universites
```

Name	Null?	Type
ID_UNIVERSITE	NOT NULL	NUMBER
NOM		VARCHAR2(100)
LOCALISATION		VARCHAR2(30)

```
SQL> DESC Universites_Maroc
```

Name	Null?	Type
ID_UNIVERSITE	NOT NULL	NUMBER(38)
NOM		VARCHAR2(100)
LOCALISATION		VARCHAR2(30)

```
SQL> DESC Universites_Etrangers
```

Name	Null?	Type
ID_UNIVERSITE	NOT NULL	NUMBER(38)
NOM		VARCHAR2(100)
LOCALISATION		VARCHAR2(30)

```
SQL>
```

CREATE OR REPLACE **TRIGGER TRIGGER\_INSERT\_universite**

BEFORE INSERT ON Universites

FOR EACH ROW

DECLARE

n NUMBER;

universites\_duplique EXCEPTION;

BEGIN

SELECT COUNT(\*) INTO n FROM Universites WHERE Nom = :NEW.Nom;

IF n = 0 THEN

-- verifier si aucune universite avec le même nom n'existe deja

-- Insérer la ligne dans les tables Universites\_Maroc et Universites\_Etrangers

INSERT INTO Universites\_Maroc VALUES (:NEW.ID\_universite, :NEW.Nom, :NEW.Localisation);

INSERT INTO Universites\_Etrangers VALUES (:NEW.ID\_universite, :NEW.Nom, :NEW.Localisation);

ELSE

-- si une universite avec le même nom existe deja

RAISE universites\_duplique;

END IF;

EXCEPTION

WHEN universites\_duplique THEN

```

        RAISE_APPLICATION_ERROR(-20002, '!!! L"universites "'||:NEW.Nom||"'
est deja exist !!!');
END;
/

```

```

SQL>
SQL> CREATE OR REPLACE TRIGGER TRIGGER_INSERT_universite
2 BEFORE INSERT ON Universites
3 FOR EACH ROW
4 DECLARE
5     n NUMBER;
6     universites_duplique EXCEPTION;
7 BEGIN
8     SELECT COUNT(*) INTO n FROM Universites WHERE Nom = :NEW.Nom;
9     IF n = 0 THEN
10        -- verifier si aucune universite avec le même nom n'existe deja
11        -- Insérer la ligne dans les tables Universites_Maroc et Universites_Etrangers
12        INSERT INTO Universites_Maroc VALUES (<:NEW.ID_universite, :NEW.Nom, :NEW.Localisation);
13        INSERT INTO Universites_Etrangers VALUES (<:NEW.ID_universite, :NEW.Nom, :NEW.Localisation);
14    ELSE
15        -- si une universite avec le même nom existe deja
16        RAISE universites_duplique;
17    END IF;
18 EXCEPTION
19     WHEN universites_duplique THEN
20        RAISE_APPLICATION_ERROR(-20002, '!!! L"universites "'||:NEW.Nom||"' est deja exist !!!');
21 END;
22 /
Trigger created.

```

### Test de trigger TRIGGER\_INSERT universite :

```

SELECT * FROM Universites;
SELECT * FROM Universites_Maroc;
SELECT * FROM Universites_Etrangers;

```

```

INSERT INTO Universites VALUES(sq_Id_Univetsite.NEXTVAL, 'Université Ibn
Zohr', 'Fes');

```

```

SQL>
SQL> SELECT * FROM Universites;
ID_UNIVERSITE NOM LOCALISATION
-----
1 Université Ibn Zohr Agadir
2 Université Cadi Ayyad Marrakech
3 Université Hassan II Casablanca
4 Université Mohammed V Rabat
5 Université Moulay Ismail Meknès
6 Université Al Quaraouiyine Fès
6 rows selected.

SQL> SELECT * FROM Universites_Maroc;
ID_UNIVERSITE NOM LOCALISATION
-----
1 Université Ibn Zohr Agadir
2 Université Cadi Ayyad Marrakech
3 Université Hassan II Casablanca
4 Université Mohammed V Rabat
5 Université Moulay Ismail Meknès
6 Université Al Quaraouiyine Fès
6 rows selected.

SQL> SELECT * FROM Universites_Etrangers;
ID_UNIVERSITE NOM LOCALISATION
-----
1 Université Ibn Zohr Agadir
2 Université Cadi Ayyad Marrakech
3 Université Hassan II Casablanca
4 Université Mohammed V Rabat
5 Université Moulay Ismail Meknès
6 Université Al Quaraouiyine Fès
6 rows selected.

SQL>
SQL>
SQL> INSERT INTO Universites VALUES(sq_Id_Univetsite.NEXTVAL, 'Université Ibn Zohr', 'Fes');
INSERT INTO Universites VALUES(sq_Id_Univetsite.NEXTVAL, 'Université Ibn Zohr', 'Fes')
*
ERROR at line 1:
ORA-20002: !!! L"universites "Université Ibn Zohr" est deja exist !!!
ORA-06512: at "ADMIN_BOURSE.TRIGGER_INSERT_UNIVERSITE", line 17
ORA-04088: error during execution of trigger 'ADMIN_BOURSE.TRIGGER_INSERT_UNIVERSITE'
SQL>

```

```

INSERT INTO Universites VALUES(sq_Id_Univetsite.NEXTVAL, 'Université Hassan
3', 'Fes');

```

```

SELECT * FROM Universites;

```



```
SELECT * FROM Universites_Maroc;
SELECT * FROM Universites_Etrangers;
```

```
SQL>
SQL> INSERT INTO Universites VALUES(sq_Id_Univetsite.NEXTVAL, 'Université Hassan 3', 'Fes');
1 row created.
SQL>
SQL> SELECT * FROM Universites;
ID_UNIVERSITE NOM                                LOCALISATION
-----
22 Université Hassan 3                          Fes
1 Université Ibn Zohr                           Agadir
2 Université Cadi Ayyad                        Marrakech
3 Université Hassan II                         Casablanca
4 Université Mohammed V                        Rabat
5 Université Moulay Ismail                     Meknès
6 Université Al Quaraouiyine                   Fès
7 rows selected.
SQL> SELECT * FROM Universites_Maroc;
ID_UNIVERSITE NOM                                LOCALISATION
-----
1 Université Ibn Zohr                           Agadir
2 Université Cadi Ayyad                        Marrakech
3 Université Hassan II                         Casablanca
4 Université Mohammed V                        Rabat
5 Université Moulay Ismail                     Meknès
6 Université Al Quaraouiyine                   Fès
22 Université Hassan 3                          Fes
7 rows selected.
SQL> SELECT * FROM Universites_Etrangers;
ID_UNIVERSITE NOM                                LOCALISATION
-----
1 Université Ibn Zohr                           Agadir
2 Université Cadi Ayyad                        Marrakech
3 Université Hassan II                         Casablanca
4 Université Mohammed V                        Rabat
5 Université Moulay Ismail                     Meknès
6 Université Al Quaraouiyine                   Fès
22 Université Hassan 3                          Fes
7 rows selected.
SQL>
```

### Création du trigger TRIGGER\_INSERT\_etud\_univ:

```
DESC Etudiants_Universites
DESC Etud_Univ_Maroc
DESC Etud_Univ_Etrangers
```

```
Administrator: Command Prompt - sqlplus admin_bourse/aissam
SQL> DESC Etudiants_Universites
Name                                Null?    Type
-----
ID_ETUDIANT                        NOT NULL NUMBER
ID_UNIVERSITE                      NOT NULL NUMBER
SQL> DESC Etud_Univ_Maroc
Name                                Null?    Type
-----
ID_ETUDIANT                        NOT NULL NUMBER(38)
ID_UNIVERSITE                      NOT NULL NUMBER(38)
SQL> DESC Etud_Univ_Etrangers
Name                                Null?    Type
-----
ID_ETUDIANT                        NOT NULL NUMBER(38)
ID_UNIVERSITE                      NOT NULL NUMBER(38)
SQL>
```

```
CREATE OR REPLACE TRIGGER TRIGGER_INSERT_etud_univ
BEFORE INSERT ON Etudiants_Universites
FOR EACH ROW
DECLARE
    n NUMBER;
    temp_etranger NUMBER := -1;
    etudiant_n_exist EXCEPTION;
    etudiant_deja EXCEPTION;
BEGIN
```

```

SELECT COUNT(*) INTO n FROM Etudiants_Universites WHERE ID_etudiant =
:NEW.ID_etudiant;
IF n = 0 THEN
    SELECT Etranger INTO temp_etranger FROM Etudiants WHERE
Etudiants.ID_etudiant = :NEW.ID_etudiant;
    IF temp_etranger = 0 THEN
        -- verifier si l'etudiant exist et de nationalite marocaine
        -- Insérer la ligne dans la table Etud_Univ_Maroc
        INSERT INTO Etud_Univ_Maroc VALUES (:NEW.ID_etudiant,
:NEW.ID_universite);
    ELSIF temp_etranger = 1 THEN
        -- si l'etudiant est etranger
        -- Insérer la ligne dans la table Etud_Univ_Etrangers
        INSERT INTO Etud_Univ_Etrangers VALUES (:NEW.ID_etudiant,
:NEW.ID_universite);
    ELSE
        -- si l'etudiant n'existe pas
        RAISE etudiant_n_exist;
    END IF;
ELSE
    -- si l'etudiant existe déjà
    RAISE etudiant_deja;
END IF;
EXCEPTION
    WHEN etudiant_n_exist THEN
        RAISE_APPLICATION_ERROR(-20003, '!!! L"etudiant n"existe pas!!!');
    WHEN etudiant_deja THEN
        RAISE_APPLICATION_ERROR(-20004, '!!! L"etudiant est déjà exist !!!');
END;
/

```

```

SQL>
SQL> CREATE OR REPLACE TRIGGER TRIGGER_INSERT_etud_univ
2 BEFORE INSERT ON Etudiants_Universites
3 FOR EACH ROW
4 DECLARE
5     n NUMBER;
6     temp_etranger NUMBER := -1;
7     etudiant_n_exist EXCEPTION;
8     etudiant_deja EXCEPTION;
9 BEGIN
10 SELECT COUNT(*) INTO n FROM Etudiants_Universites WHERE ID_etudiant = :NEW.ID_etudiant;
11 IF n = 0 THEN
12 SELECT Etranger INTO temp_etranger FROM Etudiants WHERE Etudiants.ID_etudiant = :NEW.ID_etudiant;
13 IF temp_etranger = 0 THEN
14     -- verifier si l'etudiant exist et de nationalite marocaine
15     -- Insérer la ligne dans la table Etud_Univ_Maroc
16     INSERT INTO Etud_Univ_Maroc VALUES (:NEW.ID_etudiant, :NEW.ID_universite);
17 ELSIF temp_etranger = 1 THEN
18     -- si l'etudiant est etranger
19     -- Insérer la ligne dans la table Etud_Univ_Etrangers
20     INSERT INTO Etud_Univ_Etrangers VALUES (:NEW.ID_etudiant, :NEW.ID_universite);
21 ELSE
22     -- si l'etudiant n'existe pas
23     RAISE etudiant_n_exist;
24 END IF;
25 ELSE
26     -- si l'etudiant existe déjà
27     RAISE etudiant_deja;
28 END IF;
29 EXCEPTION
30 WHEN etudiant_n_exist THEN
31     RAISE_APPLICATION_ERROR(-20003, '!!! L"etudiant n"existe pas!!!');
32 WHEN etudiant_deja THEN
33     RAISE_APPLICATION_ERROR(-20004, '!!! L"etudiant est déjà exist !!!');
34 END;
35 /
Trigger created.

```

## Test de trigger TRIGGER INSERT etud univ:

Clear Screen

```
SELECT * FROM Etudiants;  
SELECT * FROM Universites;
```

```
SELECT * FROM Etudiants_Universites;  
SELECT * FROM Etud_Univ_Maroc;  
SELECT * FROM Etud_Univ_Etrangers;
```

```
SELECT * FROM Etudiants
```

```
INSERT INTO Etudiants_Universites VALUES(55, 22);
```

```
SQL>  
SQL> SELECT * FROM Etudiants;  
ID_ETUDIANT  NOM                PRENOM              NATIONALITE          ETRANGER  
-----  
1  EN-NAHEL         Aïssam             Marocain             0  
2  Amina            Ben-Âli            Marocaine            0  
3  Karim             Ben-Moussa         Marocain             0  
4  Fatima            Ben-Mohammed       Marocaine            0  
5  Elon              Musk               Américain            1  
6  John              Smith              Britannique           1  
7  Mark              Zuckerberg          Chinois              1  
9  EN-NAHEL         Hamza              Marocain             0  
11 EN-NAHEL         Hassan             Marocain             0  
12 Bill            Gates              Allemand             1  
  
10 rows selected.  
  
SQL> SELECT * FROM Universites;  
ID_UNIVERSITE NOM                LOCALISATION  
-----  
22 Université Hassan 3 Fes  
1 Université Ibn Zohr Agadir  
2 Université Cadi Ayyad Marrakech  
3 Université Hassan II Casablanca  
4 Université Mohammed V Rabat  
5 Université Moulay Ismail Meknès  
6 Université Al Quaraouiyine Fès  
  
7 rows selected.  
  
SQL>  
SQL> SELECT * FROM Etudiants_Universites;  
ID_ETUDIANT  ID_UNIVERSITE  
-----  
1            1  
2            2  
3            3  
4            2  
5            4  
6            5  
7            6  
  
7 rows selected.  
-----  
  
SQL> SELECT * FROM Etud_Univ_Maroc;  
ID_ETUDIANT  ID_UNIVERSITE  
-----  
1            1  
2            2  
3            3  
4            2  
  
SQL> SELECT * FROM Etud_Univ_Etrangers;  
ID_ETUDIANT  ID_UNIVERSITE  
-----  
5            4  
6            5  
7            6  
  
SQL>  
SQL> SELECT * FROM Etudiants  
2  
SQL> INSERT INTO Etudiants_Universites VALUES(55, 22);  
INSERT INTO Etudiants_Universites VALUES(55, 22)  
*  
ERROR at line 1:  
ORA-01403: no data found  
ORA-06512: at "ADMIN_BOURSE.TRIGGER_INSERT_ETUD_UNIU", line 9  
ORA-04088: error during execution of trigger 'ADMIN_BOURSE.TRIGGER_INSERT_ETUD_UNIU'
```

```

INSERT INTO Etudiants_Universites VALUES(1, 1);
INSERT INTO Etudiants_Universites VALUES(11, 22);
INSERT INTO Etudiants_Universites VALUES(12, 22);

```

```

SQL>
SQL> INSERT INTO Etudiants_Universites VALUES(1, 1);
INSERT INTO Etudiants_Universites VALUES(1, 1)
      *
ERROR at line 1:
ORA-20004: ??? L'etudiant est deja exist ???
ORA-06512: at "ADMIN_BOURSE.TRIGGER_INSERT_ETUD_UNIU", line 30
ORA-04088: error during execution of trigger 'ADMIN_BOURSE.TRIGGER_INSERT_ETUD_UNIU'

SQL> INSERT INTO Etudiants_Universites VALUES(11, 22);
1 row created.

SQL> INSERT INTO Etudiants_Universites VALUES(12, 22);
1 row created.

SQL>

```

```

SELECT * FROM Etudiants_Universites;
SELECT * FROM Etud_Univ_Maroc;
SELECT * FROM Etud_Univ_Etrangers;

```

```

SQL>
SQL> SELECT * FROM Etudiants_Universites;

ID_ETUDIANT  ID_UNIVERSITE
-----
          1              1
          2              2
          3              3
          4              2
          5              4
          6              5
          7              6
         11             22
         12             22

9 rows selected.

SQL> SELECT * FROM Etud_Univ_Maroc;

ID_ETUDIANT  ID_UNIVERSITE
-----
          1              1
          2              2
          3              3
          4              2
         11             22

SQL> SELECT * FROM Etud_Univ_Etrangers;

ID_ETUDIANT  ID_UNIVERSITE
-----
          5              4
          6              5
          7              6
         12             22

SQL>
SQL>

```

### Création du trigger TRIGGER\_INSERT\_bourses:

```

DESC Bourses
DESC Bourses_Maroc
DESC Bourses_Etrangers

```

```

SQL> DESC Bourses
Name                                     Null?   Type
-----
ID_BOURSE                               NOT NULL NUMBER
ID_ETUDIANT                             NOT NULL NUMBER
NOM                                       VARCHAR2(100)
MONTANT                                  NUMBER

SQL> DESC Bourses_Maroc
Name                                     Null?   Type
-----
ID_BOURSE                               NOT NULL NUMBER(38)
ID_ETUDIANT                             NOT NULL NUMBER(38)
NOM                                       VARCHAR2(100)
MONTANT                                  NUMBER(38)

SQL> DESC Bourses_Etrangers
Name                                     Null?   Type
-----
ID_BOURSE                               NOT NULL NUMBER(38)
ID_ETUDIANT                             NOT NULL NUMBER(38)
NOM                                       VARCHAR2(100)
MONTANT                                  NUMBER(38)

SQL>

```

```

CREATE OR REPLACE TRIGGER TRIGGER_INSERT_bourses
BEFORE INSERT ON Bourses
FOR EACH ROW
DECLARE
    n NUMBER;
    temp_etrange NUMBER;
    etudiant_deja EXCEPTION;
BEGIN
    SELECT COUNT(*) INTO n FROM Bourses WHERE ID_etudiant =
:NEW.ID_etudiant;
    IF n = 0 THEN
        SELECT Etranger INTO temp_etrange FROM Etudiants WHERE
Etudiants.ID_etudiant = :NEW.ID_etudiant;
        IF temp_etrange = 0 THEN
            -- verifier si l'etudiant exist et de nationalite marocaine
            -- Insérer la ligne dans la table Bourses_Maroc
            INSERT INTO Bourses_Maroc VALUES (:NEW.ID_bourse,
:NEW.ID_etudiant, :NEW.Nom, :NEW.Montant);
        ELSE
            -- si l'etudiant est etrange
            -- Insérer la ligne dans la table Bourses_Etrangers
            INSERT INTO Bourses_Etrangers VALUES (:NEW.ID_bourse,
:NEW.ID_etudiant, :NEW.Nom, :NEW.Montant);
        END IF;
    ELSE
        -- si l'etudiant exist deja
        RAISE etudiant_deja;
    END IF;
EXCEPTION
    WHEN etudiant_deja THEN
        RAISE_APPLICATION_ERROR(-20005, '!!! L"etudiant est deja exist !!!');
END;
/

```

```

SQL>
SQL> CREATE OR REPLACE TRIGGER TRIGGER_INSERT_bourses
2 BEFORE INSERT ON Bourses
3 FOR EACH ROW
4 DECLARE
5     n NUMBER;
6     temp_etrange NUMBER;
7     etudiant_deja EXCEPTION;
8 BEGIN
9     SELECT COUNT(*) INTO n FROM Bourses WHERE ID_etudiant = :NEW.ID_etudiant;
10    IF n = 0 THEN
11        SELECT Etranger INTO temp_etrange FROM Etudiants WHERE Etudiants.ID_etudiant = :NEW.ID_etudiant;
12        IF temp_etrange = 0 THEN
13            -- verifier si l'etudiant exist et de nationalite marocaine
14            -- Insérer la ligne dans la table Bourses_Maroc
15            INSERT INTO Bourses_Maroc VALUES (<:NEW.ID_bourse, :NEW.ID_etudiant, :NEW.Nom, :NEW.Montant>);
16        ELSE
17            -- si l'etudiant est etrange
18            -- Insérer la ligne dans la table Bourses_Etrangers
19            INSERT INTO Bourses_Etrangers VALUES (<:NEW.ID_bourse, :NEW.ID_etudiant, :NEW.Nom, :NEW.Montant>);
20        END IF;
21    ELSE
22        -- si l'etudiant exist deja
23        RAISE etudiant_deja;
24    END IF;
25 EXCEPTION
26    WHEN etudiant_deja THEN
27        RAISE_APPLICATION_ERROR(-20004, '!!! L''etudiant est deja exist !!!');
28 END;
29 /
Trigger created.
SQL>

```

### Test de trigger TRIGGER INSERT bourses:

SELECT \* FROM Etudiants;

SELECT \* FROM Bourses;

SELECT \* FROM Bourses\_Maroc;

SELECT \* FROM Bourses\_Etrangers;

INSERT INTO Bourses VALUES(sq\_Id\_Bourse.NEXTVAL, 1, 'Bourses Minhaty', 5000 );

```

SQL>
SQL> SELECT * FROM Etudiants;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0
12	Bill	Gates	Allemand	1

```

10 rows selected.
SQL>

```

```

SQL>
SQL> SELECT * FROM Bourses;

```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
1	1	Bourses Minhaty	1500
2	2	Bourse Istihqag	3000
3	3	Bourses FME	2000
4	4	Bourse AMCI	4000
5	5	Bourses Minhaty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhaty	1500

```

7 rows selected.
SQL> SELECT * FROM Bourses_Maroc;

```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
1	1	Bourses Minhaty	1500
2	2	Bourse Istihqag	3000
3	3	Bourses FME	2000
4	4	Bourse AMCI	4000

```

SQL> SELECT * FROM Bourses_Etrangers;

```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
5	5	Bourses Minhaty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhaty	1500

```

SQL>
SQL>
SQL> INSERT INTO Bourses VALUES(sq_Id_Bourse.NEXTVAL, 1, 'Bourses Minhaty', 5000 );
INSERT INTO Bourses VALUES(sq_Id_Bourse.NEXTVAL, 1, 'Bourses Minhaty', 5000 )
*
ERROR at line 1:
ORA-20004: !!! L'etudiant est deja exist !!!
ORA-06512: at "ADMIN_BOURSE.TRIGGER_INSERT_BOURSES", line 24
ORA-04088: error during execution of trigger 'ADMIN_BOURSE.TRIGGER_INSERT_BOURSES'
SQL>

```

```

INSERT INTO Bourses VALUES(sq_Id_Bourse.NEXTVAL, 9, 'Bourses Minhaty',
5000 );
INSERT INTO Bourses VALUES(sq_Id_Bourse.NEXTVAL, 11, 'Bourses Minhaty',
5000 );
INSERT INTO Bourses VALUES(sq_Id_Bourse.NEXTVAL, 12, 'Bourses Minhaty',
5000 );

```

```

SELECT * FROM Bourses;
SELECT * FROM Bourses_Maroc;
SELECT * FROM Bourses_Etrangers;

```

```

SQL>
SQL> INSERT INTO Bourses VALUES(sq_Id_Bourse.NEXTVAL, 9, 'Bourses Minhaty', 5000 );
1 row created.
SQL> INSERT INTO Bourses VALUES(sq_Id_Bourse.NEXTVAL, 11, 'Bourses Minhaty', 5000 );
1 row created.
SQL> INSERT INTO Bourses VALUES(sq_Id_Bourse.NEXTVAL, 12, 'Bourses Minhaty', 5000 );
1 row created.

```

```

SQL>
SQL> SELECT * FROM Bourses;

```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
30	9	Bourses Minhaty	5000
31	11	Bourses Minhaty	5000
32	12	Bourses Minhaty	5000
1	1	Bourses Minhaty	1500
2	2	Bourse Istihqag	3000
3	3	Bourses FME	2000
4	4	Bourse AMCI	4000
5	5	Bourses Minhaty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhaty	1500

```

10 rows selected.
SQL> SELECT * FROM Bourses_Maroc;

```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
1	1	Bourses Minhaty	1500
2	2	Bourse Istihqag	3000
3	3	Bourses FME	2000
4	4	Bourse AMCI	4000
30	9	Bourses Minhaty	5000
31	11	Bourses Minhaty	5000

```

6 rows selected.
SQL> SELECT * FROM Bourses_Etrangers;

```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
5	5	Bourses Minhaty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhaty	1500
32	12	Bourses Minhaty	5000

```

SQL>
SQL> SELECT * FRO

```

**Note:** Dans la création des tables, je n'ai pas ajouté "ON DELETE CASCADE" aux clés étrangères, donc avant de créer les triggers **TRIGGER\_DELETE\_etudiant**, **TRIGGER\_DELETE\_universite**, **TRIGGER\_DELETE\_bourse** et **TRIGGER\_DELETE\_etud\_univ**, il est nécessaire de modifier les contraintes des clés étrangères en ajoutant l'option ON DELETE CASCADE, pour éviter l'erreur suivante :

```
SQL> delete from ETUDIANTS where ID_ETUDIANT = 21;
delete from ETUDIANTS where ID_ETUDIANT = 21
*
ERROR at line 1:
ORA-02055: distributed update operation failed; rollback required
ORA-02292: integrity constraint (ADMIN_BOURSE.FK_ETUDUNIV_ETUDIANTS) violated - child record found
```

Table de Etudiants Universites de site 1 :

```
alter table Etudiants_Universites drop constraint FK_EtudUniv_Etudiants;
```

```
alter table Etudiants_Universites drop constraint FK_EtudUniv_Universite;
```

```
alter table Etudiants_Universites add CONSTRAINT FK_EtudUniv_Etudiants
FOREIGN KEY (ID_etudiant) REFERENCES Etudiants(ID_etudiant) ON DELETE
CASCADE;
```

```
alter table Etudiants_Universites add CONSTRAINT FK_EtudUniv_Universite
FOREIGN KEY (ID_universite) REFERENCES Universites(ID_universite) ON
DELETE CASCADE;
```

```
SQL> alter table Etudiants_Universites drop constraint FK_EtudUniv_Etudiants;
Table altered.
SQL> alter table Etudiants_Universites drop constraint FK_EtudUniv_Universite;
Table altered.
SQL>
SQL> alter table Etudiants_Universites add CONSTRAINT FK_EtudUniv_Etudiants FOREIGN KEY (ID_etudiant) REFERENCES Etudiants(ID_etudiant) ON DELETE CASCADE;
Table altered.
SQL>
SQL> alter table Etudiants_Universites add CONSTRAINT FK_EtudUniv_Universite FOREIGN KEY (ID_universite) REFERENCES Universites(ID_universite) ON DELETE CASCADE;
Table altered.
```

Table de Bourses de site 1 :

```
alter table Bourses drop constraint FK_Brs_Etud;
```

```
alter table Bourses add CONSTRAINT FK_Brs_Etud FOREIGN KEY (ID_etudiant)
REFERENCES Etudiants(ID_etudiant) ON DELETE CASCADE;
```

```
SQL> alter table Bourses drop constraint FK_Brs_Etud;
Table altered.
SQL>
SQL> alter table Bourses add CONSTRAINT FK_Brs_Etud FOREIGN KEY (ID_etudiant) REFERENCES Etudiants(ID_etudiant) ON DELETE CASCADE;
Table altered.
```

Table de Etud Univ Maroc de site 2 :

```
SELECT constraint_name, delete_rule FROM all_constraints WHERE table_name
= 'ETUD_UNIV_MAROC' AND constraint_type = 'R';
```

```
alter table Etud_Univ_Maroc drop constraint SYS_C007827;
```



alter table Etud\_Univ\_Maroc drop constraint SYS\_C007798;

alter table Etud\_Univ\_Maroc add constraint fk\_etud\_univ\_maroc\_etud foreign key (ID\_etudiant) references Etudiants\_Maroc(ID\_etudiant) **on delete cascade;**

alter table Etud\_Univ\_Maroc add constraint fk\_etud\_univ\_maroc\_univ foreign key (ID\_universite) references Universites\_Maroc(ID\_universite) **on delete cascade;**

```
SQL> SELECT constraint_name, delete_rule FROM all_constraints WHERE table_name = 'ETUD_UNIV_MAROC' AND constraint_type = 'R';
CONSTRAINT_NAME          DELETE_RULE
-----
SYS_C007827              NO ACTION
SYS_C007798              NO ACTION

SQL> alter table Etud_Univ_Maroc drop constraint SYS_C007827;
Table altered.

SQL> alter table Etud_Univ_Maroc drop constraint SYS_C007798;
Table altered.

SQL> alter table Etud_Univ_Maroc add constraint fk_etud_univ_maroc_etud foreign key (ID_etudiant) references Etudiants_Maroc(ID_etudiant) on delete cascade;
Table altered.

SQL> alter table Etud_Univ_Maroc add constraint fk_etud_univ_maroc_univ foreign key (ID_universite) references Universites_Maroc(ID_universite) on delete cascade;
Table altered.

SQL>
```

Table de Bourses\_Maroc de site 2 :

SELECT constraint\_name, delete\_rule FROM all\_constraints WHERE table\_name = 'BOURSES\_MAROC' AND constraint\_type = 'R';

alter table Bourses\_Maroc drop constraint SYS\_C007826;

alter table Bourses\_Maroc add constraint fk\_bourses\_maroc\_etud foreign key (ID\_etudiant) references Etudiants\_Maroc(ID\_etudiant) **on delete cascade;**

```
SQL> SELECT constraint_name, delete_rule FROM all_constraints WHERE table_name = 'BOURSES_MAROC' AND constraint_type = 'R';
CONSTRAINT_NAME          DELETE_RULE
-----
SYS_C007826              NO ACTION

SQL> alter table Bourses_Maroc drop constraint SYS_C007826;
Table altered.

SQL> alter table Bourses_Maroc add constraint fk_bourses_maroc_etud foreign key (ID_etudiant) references Etudiants_Maroc(ID_etudiant) on delete cascade;
Table altered.

SQL>
```

Table de Etud\_Univ\_Etrangers de site 3 :

SELECT constraint\_name, delete\_rule FROM all\_constraints WHERE table\_name = 'ETUD\_UNIV\_ETRANGERS' AND constraint\_type = 'R';

alter table Etud\_Univ\_Etrangers drop constraint SYS\_C007804;

alter table Etud\_Univ\_Etrangers drop constraint SYS\_C007805;

alter table Etud\_Univ\_Etrangers add constraint fk\_etud\_univ\_etr\_etud foreign key (ID\_etudiant) references Etudiants\_Etrangers(ID\_etudiant) **on delete cascade;**

alter table Etud\_Univ\_Etrangers add constraint fk\_etud\_univ\_etr\_univ foreign key (ID\_universite) references Universites\_Etrangers(ID\_universite) **on delete cascade;**

```
SQL> SELECT constraint_name, delete_rule FROM all_constraints WHERE table_name = 'ETUD_UNIV_ETRANGERS' AND constraint_type = 'R';
CONSTRAINT_NAME          DELETE_RULE
-----
SYS_C007804              NO ACTION
SYS_C007805              NO ACTION

SQL>
SQL> alter table Etud_Univ_Etrangers drop constraint SYS_C007804;
Table altered.
SQL> alter table Etud_Univ_Etrangers drop constraint SYS_C007805;
Table altered.
SQL> alter table Etud_Univ_Etrangers add constraint fk_etud_univ_etr_etud foreign key (ID_etudiant) references Etudiants_Etrangers(ID_etudiant) on delete cascade;
Table altered.
SQL> alter table Etud_Univ_Etrangers add constraint fk_etud_univ_etr_univ foreign key (ID_universite) references Universites_Etrangers(ID_universite) on delete cascade;
Table altered.
```

Table de Bourses Etrangers de site 3 :

```
SELECT constraint_name, delete_rule FROM all_constraints WHERE table_name = 'BOURSES_ETRANGERS' AND constraint_type = 'R';
```

alter table Bourses\_Etrangers drop constraint SYS\_C007803;

alter table Bourses\_Etrangers add constraint fk\_bourses\_etr\_etud foreign key (ID\_etudiant) references Etudiants\_Etrangers(ID\_etudiant) **on delete cascade;**

```
SQL> SELECT constraint_name, delete_rule FROM all_constraints WHERE table_name = 'BOURSES_ETRANGERS' AND constraint_type = 'R';
CONSTRAINT_NAME          DELETE_RULE
-----
SYS_C007803              NO ACTION

SQL> alter table Bourses_Etrangers drop constraint SYS_C007803;
Table altered.
```

```
SQL> alter table Bourses_Etrangers add constraint fk_bourses_etr_etud foreign key (ID_etudiant) references Etudiants_Etrangers(ID_etudiant) on delete cascade;
Table altered.
```

### Création du trigger TRIGGER DELETE etudiant:

CREATE OR REPLACE **TRIGGER TRIGGER\_DELETE\_etudiant**

BEFORE DELETE ON Etudiants

FOR EACH ROW

BEGIN

IF :OLD.Etranger = 0 THEN

-- supprimer la ligne dans la table Etudiants\_Maroc

DELETE FROM Etudiants\_Maroc WHERE ID\_etudiant = :OLD.ID\_etudiant;

ELSE

-- supprimer la ligne dans la table Etudiants\_Etrangers

DELETE FROM Etudiants\_Etrangers WHERE ID\_etudiant = :OLD.ID\_etudiant;

END IF;

END;

/

```

12 rows selected.
SQL> set lines 150
SQL> CREATE OR REPLACE TRIGGER TRIGGER_DELETE_etudiant
2 BEFORE DELETE ON Etudiants
3 FOR EACH ROW
4 BEGIN
5     IF :OLD.Etranger = 0 THEN
6         -- supprimer la ligne dans la table Etudiants_Maroc
7         DELETE FROM Etudiants_Maroc WHERE ID_etudiant = :OLD.ID_etudiant;
8     ELSE
9         -- supprimer la ligne dans la table Etudiants_Etrangers
10        DELETE FROM Etudiants_Etrangers WHERE ID_etudiant = :OLD.ID_etudiant;
11    END IF;
12 END;
13 /
Trigger created.

```

select \* from ETUDIANTS;

```

SQL> select * from ETUDIANTS;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
21	EN-NAHEL	Med	Marocain	0
22	Steve	Jobs	American	1
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karin	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Americain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0
12	Bill	Gates	Allemand	1

```

12 rows selected.

```

```

SQL> select * from Etudiants_Maroc;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karin	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0
21	EN-NAHEL	Med	Marocain	0

```

7 rows selected.

```

delete from ETUDIANTS where ID\_ETUDIANT = 21;

commit;

```
SQL>
SQL> delete from ETUDIANTS where ID_ETUDIANT = 21;
1 row deleted.
SQL> select * from ETUDIANTS;
ID_ETUDIANT NOM PRENOM NATIONALITE ETRANGER
-----
22 Steve Jobs American 1
1 EN-NAHEL Aissam Marocain 0
2 Amina Ben-Ali Marocaine 0
3 Karim Ben-Moussa Marocain 0
4 Fatima Ben-Mohammed Marocaine 0
5 Elon Musk Américain 1
6 John Smith Britannique 1
7 Mark Zuckerberg Chinois 1
9 EN-NAHEL Hamza Marocain 0
11 EN-NAHEL Hassan Marocain 0
12 Bill Gates Allemand 1
11 rows selected.
SQL> commit;
Commit complete.
SQL>
```

```
SQL> select * from Etudiants_Maroc;
ID_ETUDIANT NOM PRENOM NATIONALITE ETRANGER
-----
1 EN-NAHEL Aissam Marocain 0
2 Amina Ben-Ali Marocaine 0
3 Karim Ben-Moussa Marocain 0
4 Fatima Ben-Mohammed Marocaine 0
9 EN-NAHEL Hamza Marocain 0
11 EN-NAHEL Hassan Marocain 0
6 rows selected.
SQL>
```

### Création du trigger TRIGGER DELETE universite:

```
CREATE OR REPLACE TRIGGER TRIGGER_DELETE_universite
AFTER DELETE ON Universites
FOR EACH ROW
BEGIN
    -- supprimer la ligne dans la table Universites_Maroc
    DELETE FROM Universites_Maroc WHERE ID_universite = :OLD.ID_universite;
    -- supprimer la ligne dans la table Universites_Etrangers
    DELETE FROM Universites_Etrangers WHERE ID_universite =
:OLD.ID_universite;
END;
/
```

```
select * from Universites;
```

```
DELETE FROM Universites WHERE ID_UNIVERSITE = 6;
```

```
COMMIT;
```

```
select * from Universites_Maroc; --(Dans le site 2)
```

```
select * from Universites_Etrangers; --(Dans le site 3)
```

```

SQL> CREATE OR REPLACE TRIGGER TRIGGER_DELETE_universite
2 AFTER DELETE ON Universities
3 FOR EACH ROW
4 BEGIN
5     -- supprimer la ligne dans la table Universities_Maroc
6     DELETE FROM Universities_Maroc WHERE ID_universite = :OLD.ID_universite;
7     -- supprimer la ligne dans la table Universities_Etrangers
8     DELETE FROM Universities_Etrangers WHERE ID_universite = :OLD.ID_universite;
9 END;
10 /
Trigger created.
SQL> select * from Universities;

```

ID_UNIVERSITE	NOM	LOCALISATION
22	Université Hassan 3	Fes
23	Université Hassan 4	Meknes
1	Université Ibn Zohr	Agadir
2	Université Cadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed V	Rabat
5	Université Moulay Ismail	Meknès
6	Université Al Quaraouiyine	Fès

```

8 rows selected.
SQL> DELETE FROM Universities WHERE ID_UNIVERSITE = 6;
1 row deleted.
SQL> COMMIT;
Commit complete.

```

```

SQL> select * from Universities_Maroc;

```

ID_UNIVERSITE	NOM	LOCALISATION
1	Université Ibn Zohr	Agadir
2	Université Cadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed V	Rabat
5	Université Moulay Ismail	Meknès
6	Université Al Quaraouiyine	Fès
22	Université Hassan 3	Fes
23	Université Hassan 4	Meknes

```

8 rows selected.

```

```

SQL> select * from Universities_Etrangers;

```

ID_UNIVERSITE	NOM	LOCALISATION
1	Université Ibn Zohr	Agadir
2	Université Cadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed V	Rabat
5	Université Moulay Ismail	Meknès
6	Université Al Quaraouiyine	Fès
22	Université Hassan 3	Fes
23	Université Hassan 4	Meknes

```

8 rows selected.

```

select \* from Universities; --(Dans le site 1)

select \* from Universities\_Maroc; --(Dans le site 2)

select \* from Universities\_Etrangers; --(Dans le site 3)

```

SQL> select * from Universities;

```

ID_UNIVERSITE	NOM	LOCALISATION
22	Université Hassan 3	Fes
23	Université Hassan 4	Meknes
1	Université Ibn Zohr	Agadir
2	Université Cadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed V	Rabat
5	Université Moulay Ismail	Meknès

```

7 rows selected.
SQL>

```

```

SQL> select * from Universities_Maroc;

```

ID_UNIVERSITE	NOM	LOCALISATION
1	Université Ibn Zohr	Agadir
2	Université Cadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed V	Rabat
5	Université Moulay Ismail	Meknès
22	Université Hassan 3	Fes
23	Université Hassan 4	Meknes

```

7 rows selected.
SQL>

```

```
SQL> select * from Universites_Etrangers;
```

ID_UNIVERSITE	NOM	LOCALISATION
1	Université Ibn Zohr	Agadir
2	Université Gadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed VI	Rabat
5	Université Moulay Ismail	Meknès
22	Université Hassan III	Fes
23	Université Hassan IV	Meknes

```
7 rows selected.
SQL>
```

### Création du trigger TRIGGER DELETE\_bourse:

```
select * from Bourses;
```

```
CREATE OR REPLACE TRIGGER TRIGGER_DELETE_bourse
AFTER DELETE ON Bourses
FOR EACH ROW
BEGIN
    -- supprimer la ligne dans la table Bourses_Maroc
    DELETE FROM Bourses_Maroc WHERE ID_Bourse = :OLD.ID_Bourse;
    -- supprimer la ligne dans la table Bourses_Etrangers
    DELETE FROM Bourses_Etrangers WHERE ID_Bourse = :OLD.ID_Bourse;
END;
/
```

```
DELETE FROM Bourses WHERE ID_Bourse = 32;
```

```
commit;
```

```
select * from Bourses_Etrangers; --(Dans le site 3)
```

```
SQL>
SQL> select * from Bourses;
```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
30	9	Bourses Minhaty	5000
34	22	Bourseses FME	3500
32	12	Bourses Minhaty	5000
1	1	Bourses Minhaty	1500
2	2	Bourse Istihqaq	3000
3	3	Bourses FME	2000
4	4	Bourse AMCI	4000
5	5	Bourses Minhaty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhaty	1500

```
10 rows selected.
```

```
SQL>
SQL> CREATE OR REPLACE TRIGGER TRIGGER_DELETE_bourse
2 AFTER DELETE ON Bourses
3 FOR EACH ROW
4 BEGIN
5     -- supprimer la ligne dans la table Bourses_Maroc
6     DELETE FROM Bourses_Maroc WHERE ID_Bourse = :OLD.ID_Bourse;
7     -- supprimer la ligne dans la table Bourses_Etrangers
8     DELETE FROM Bourses_Etrangers WHERE ID_Bourse = :OLD.ID_Bourse;
9 END;
10 /

Trigger created.
SQL>
```

```
SQL> DELETE FROM Bourses WHERE ID_Bourse = 32;

1 row deleted.

SQL> commit;

Commit complete.

SQL>
```

```
SQL> select * from Bourses_Etrangers;
```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
5	5	Bourses Minhaty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhaty	1500
32	12	Bourses Minhaty	5000
34	22	Bourseses FME	3500

select \* from Bourses;

select \* from Bourses\_Etrangers; --(Dans le site 3)

```
SQL> select * from Bourses;
```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
30	9	Bourses Minhaty	5000
34	22	Bourseses FME	3500
1	1	Bourses Minhaty	1500
2	2	Bourse Istihgaq	3000
3	3	Bourses FME	2000
4	4	Bourse AMCI	4000
5	5	Bourses Minhaty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhaty	1500

9 rows selected.

```
SQL> select * from Bourses_Etrangers;
```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
5	5	Bourses Minhaty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhaty	1500
34	22	Bourseses FME	3500

```
SQL>
```

### Création du trigger TRIGGER DELETE etud univ:

```
CREATE OR REPLACE TRIGGER TRIGGER_DELETE_etud_univ
AFTER DELETE ON Etudiants_Universites
FOR EACH ROW
BEGIN
    -- supprimer la ligne dans la table Etud_Univ_Maroc
    DELETE FROM Etud_Univ_Maroc WHERE ID_etudiant = :OLD.ID_etudiant;
    -- supprimer la ligne dans la table Etud_Univ_Etrangers
    DELETE FROM Etud_Univ_Etrangers WHERE ID_etudiant = :OLD.ID_etudiant;
END;
/
```

select \* from Etudiants\_Universites;

DELETE FROM Etudiants\_Universites WHERE ID\_etudiant = 11;

commit;

select \* from Etud\_Univ\_Maroc; --(Dans le site 2)

```

SQL>
SQL> CREATE OR REPLACE TRIGGER TRIGGER_DELETE_etud_univ
2  AFTER DELETE ON Etudiants_Universites
3  FOR EACH ROW
4  BEGIN
5  -- supprimer la ligne dans la table Etud_Univ_Maroc
6  DELETE FROM Etud_Univ_Maroc WHERE ID_etudiant = :OLD.ID_etudiant;
7  -- supprimer la ligne dans la table Etud_Univ_Etrangers
8  DELETE FROM Etud_Univ_Etrangers WHERE ID_etudiant = :OLD.ID_etudiant;
9  END;
10 /

Trigger created.
SQL>

```

```

SQL>
SQL> select * from Etudiants_Universites;

ID_ETUDIANT  ID_UNIVERSITE
-----
1            1
2            2
3            3
4            2
5            4
6            5
9            22
11           3
12           22
22           23

10 rows selected.

SQL> DELETE FROM Etudiants_Universites WHERE ID_etudiant = 11;

1 row deleted.

SQL> commit;

Commit complete.

SQL>

```

```

SQL>
SQL> select * from Etud_Univ_Maroc;

ID_ETUDIANT  ID_UNIVERSITE
-----
1            1
2            2
3            3
4            2
9            22
11           3

6 rows selected.

```

select \* from Etudiants\_Universites;

select \* from Etud\_Univ\_Maroc; --(Dans le site 2)

```

SQL>
SQL> select * from Etudiants_Universites;

ID_ETUDIANT  ID_UNIVERSITE
-----
1            1
2            2
3            3
4            2
5            4
6            5
9            22
12           22
22           23

9 rows selected.

```



```
SQL> select * from Etud_Univ_Maroc;
ID_ETUDIANT ID_UNIVERSITE
-----
1           1
2           2
3           3
4           2
9           22
SQL>
```

## - Création des procédures

### Création de la procédure INSERT\_etudiant:

```
CREATE OR REPLACE PROCEDURE INSERT_etudiant(p_Nom
Etudiants.Nom%TYPE,
        p_Prenom Etudiants.Prenom%TYPE,
        p_Nationalite Etudiants.Nationalite%TYPE,
        p_Etranger Etudiants.Etranger%TYPE ) AS
n NUMBER := 0;
etudiant_duplique EXCEPTION;

BEGIN
    SELECT COUNT(*) INTO n FROM Etudiants WHERE Nom = p_Nom AND
Prenom = p_Prenom;
    IF n = 0 THEN
        -- verifier si l'etudiant n'existe pas
        -- Insérer la ligne dans la table Etudiants
        INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, p_Nom, p_Prenom,
p_Nationalite, p_Etranger);
        dbms_output.put_line('L"etudiant "'||p_Nom||' "'||p_Prenom||'" a ete ajoute');
    ELSE
        -- si l'etudiant existe deja
        RAISE etudiant_duplique;
    END IF;
EXCEPTION
    WHEN etudiant_duplique THEN
        dbms_output.put_line('!!! L"etudiant "'||p_Nom||' "'||p_Prenom||'" deja existe !!!');
END;
/
```

```

SQL>
SQL> CREATE OR REPLACE PROCEDURE INSERT_etudiant(p_Nom Etudiants.Nom%TYPE,
2          p_Prenom Etudiants.Prenom%TYPE,
3          p_Nationalite Etudiants.Nationalite%TYPE,
4          p_Etranger Etudiants.Etranger%TYPE) AS
5          n NUMBER := 0;
6          etudiant_duplique EXCEPTION;
7
8      BEGIN
9          SELECT COUNT(*) INTO n FROM Etudiants WHERE Nom = p_Nom AND Prenom = p_Prenom;
10         IF n = 0 THEN
11             -- verifier si l'etudiant n'existe pas
12             -- Insérer la ligne dans la table Etudiants
13             INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, p_Nom, p_Prenom, p_Nationalite, p_Etranger);
14             dbms_output.put_line('L''etudiant '''||p_Nom||' '||p_Prenom||''' a ete ajoute');
15         ELSE
16             -- si l'etudiant existe deja
17             RAISE etudiant_duplique;
18         END IF;
19     EXCEPTION
20     WHEN etudiant_duplique THEN
21         dbms_output.put_line('!!! L''etudiant '''||p_Nom||' '||p_Prenom||''' deja existe !!!');
22     END;
23 /

```

Procedure created.

```

SQL>
SQL>
SQL>
SQL> SELECT * FROM Etudiants;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0
12	Bill	Gates	Allemand	1

10 rows selected.

SELECT \* FROM Etudiants;

```

SQL>
SQL> SELECT * FROM Etudiants;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0
12	Bill	Gates	Allemand	1

10 rows selected.

SELECT \* FROM Etudiants\_Maroc;

SELECT \* FROM Etudiants\_Etrangers;

```

C:\> Select Administrator: Command Prompt - sqlplus admin_bourse/aissam

SQL> SELECT * FROM Etudiants_Maroc;

ID_ETUDIANT  NOM                PRENOM              NATIONALITE          ETRANGER
-----
1 EN-NAHEL      Aissam              Marocain             0
2 Amina         Ben-Ali             Marocaine            0
3 Karim         Ben-Moussa          Marocain             0
4 Fatima        Ben-Mohammed        Marocaine            0
9 EN-NAHEL      Hamza               Marocain             0
11 EN-NAHEL     Hassan              Marocain             0

6 rows selected.

SQL> SELECT * FROM Etudiants_Etrangers;

ID_ETUDIANT  NOM                PRENOM              NATIONALITE          ETRANGER
-----
5 Elon        Musk               Américain           1
6 John        Smith              Britannique          1
7 Mark        Zuckerberg          Chinois             1
12 Bill       Gates              Allemand            1

SQL>

```

```

BEGIN
    INSERT_etudiant('EN-NAHEL', 'Med', 'Marocain', 0);
    INSERT_etudiant('Steve', 'Jobs', 'American', 1);
END;
/

```

```

SELECT * FROM Etudiants;
SELECT * FROM Etudiants_Maroc;
SELECT * FROM Etudiants_Etrangers;

```

```

SQL>
SQL> BEGIN
2   INSERT_etudiant('EN-NAHEL', 'Med', 'Marocain', 0);
3   INSERT_etudiant('Steve', 'Jobs', 'American', 1);
4   END;
5   /
PL/SQL procedure successfully completed.

SQL>
SQL> SELECT * FROM Etudiants;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
21	EN-NAHEL	Med	Marocain	0
22	Steve	Jobs	American	1
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karin	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0

```

ID_ETUDIANT NOM          PRENOM          NATIONALITE      ETRANGER
-----
12 Bill      Gates          Allemand        1
12 rows selected.

SQL> SELECT * FROM Etudiants_Maroc;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karin	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0
21	EN-NAHEL	Med	Marocain	0

```

7 rows selected.

SQL> SELECT * FROM Etudiants_Etrangers;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1
12	Bill	Gates	Allemand	1
22	Steve	Jobs	American	1

```

SQL>

```

### Création de la procédure INSERT universite:

```

CREATE OR REPLACE PROCEDURE INSERT_universite(p_Nom
Universites.Nom%TYPE,
          p_Localisation Universites.Localisation%TYPE ) AS

n NUMBER := 0;
universite_duplique EXCEPTION;

BEGIN
  SELECT COUNT(*) INTO n FROM Universites WHERE Nom = p_Nom;
  IF n = 0 THEN
    -- verifier si l'universite n'existe pas
    -- Insérer la ligne dans la table Universites
    INSERT INTO Universites VALUES(sq_Id_Univetsite.NEXTVAL, p_Nom,
p_Localisation);
    dbms_output.put_line('L"Universite "'||p_Nom||'" a ete ajoute');
  ELSE

```

```

        -- si l'universite existe deja
        RAISE universite_duplique;
    END IF;
EXCEPTION
    WHEN universite_duplique THEN
        dbms_output.put_line('!!! L''Universite ""||p_Nom||"" deja existe !!!');
    END ;
/

SELECT * FROM Universites;
SELECT * FROM Universites_Maroc;
SELECT * FROM Universites_Etrangers;

```

```

C:\Administrator: Command Prompt - sqlplus admin_bourse/aissam

SQL> CREATE OR REPLACE PROCEDURE INSERT_universite(p_Nom Universites.Nom%TYPE,
2         p_Localisation Universites.Localisation%TYPE ) AS
3     n NUMBER := 0;
4     universite_duplique EXCEPTION;
5
6     BEGIN
7         SELECT COUNT(*) INTO n FROM Universites WHERE Nom = p_Nom;
8         IF n = 0 THEN
9             -- verifier si l'universite n'existe pas
10            -- Insérer la ligne dans la table Universites
11            INSERT INTO Universites VALUES(sq_Id_Univetsite.NEXTVAL, p_Nom, p_Localisation);
12            dbms_output.put_line('L''Universite ""||p_Nom||"" a été ajouté');
13        ELSE
14            -- si l'universite existe deja
15            RAISE universite_duplique;
16        END IF;
17    EXCEPTION
18    WHEN universite_duplique THEN
19        dbms_output.put_line('!!! L''Universite ""||p_Nom||"" deja existe !!!');
20    END ;
21 /

Procedure created.

SQL>
SQL> SELECT * FROM Universites;

```

ID_UNIVERSITE	NOM	LOCALISATION
22	Université Hassan 3	Fes
1	Université Ibn Zohr	Agadir
2	Université Cadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed V	Rabat
5	Université Moulay Ismail	Meknès
6	Université Al Quaraouiyine	Fès

```

7 rows selected.

SQL> SELECT * FROM Universites_Maroc;

```

ID_UNIVERSITE	NOM	LOCALISATION
1	Université Ibn Zohr	Agadir
2	Université Cadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed V	Rabat
5	Université Moulay Ismail	Meknès
6	Université Al Quaraouiyine	Fès
22	Université Hassan 3	Fes

```

7 rows selected.

```

```

SQL> SELECT * FROM Universites_Etrangers;

```

ID_UNIVERSITE	NOM	LOCALISATION
1	Université Ibn Zohr	Agadir
2	Université Cadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed V	Rabat
5	Université Moulay Ismail	Meknès
6	Université Al Quaraouiyine	Fès
22	Université Hassan 3	Fes

```

7 rows selected.

```

```

BEGIN
    INSERT_universite('Universite Hassan 4', 'Meknes');
END;
/

SELECT * FROM Universites;
SELECT * FROM Universites_Maroc;
SELECT * FROM Universites_Etrangers;

```

```

SQL>
SQL> BEGIN
  2   INSERT_universite('Universite Hassan 4', 'Meknes');
  3 END;
  4 /

PL/SQL procedure successfully completed.

SQL>
SQL> SELECT * FROM Universites;

ID_UNIVERSITE NOM                                LOCALISATION
-----
22 Université Hassan 3                          Fes
23 Université Hassan 4                          Meknes
1 Université Ibn Zohr                           Agadir
2 Université Cadi Ayyad                         Marrakech
3 Université Hassan II                          Casablanca
4 Université Mohammed V                         Rabat
5 Université Moulay Ismail                      Meknès
6 Université Al Quaraouiyine                    Fès

8 rows selected.

SQL> SELECT * FROM Universites_Maroc;

ID_UNIVERSITE NOM                                LOCALISATION
-----
1 Université Ibn Zohr                           Agadir
2 Université Cadi Ayyad                         Marrakech
3 Université Hassan II                          Casablanca
4 Université Mohammed V                         Rabat
5 Université Moulay Ismail                      Meknès
6 Université Al Quaraouiyine                    Fès
22 Université Hassan 3                          Fes
23 Université Hassan 4                          Meknes

8 rows selected.

SQL> SELECT * FROM Universites_Etrangers;

ID_UNIVERSITE NOM                                LOCALISATION
-----
1 Université Ibn Zohr                           Agadir
2 Université Cadi Ayyad                         Marrakech
3 Université Hassan II                          Casablanca
4 Université Mohammed V                         Rabat
5 Université Moulay Ismail                      Meknès
6 Université Al Quaraouiyine                    Fès
22 Université Hassan 3                          Fes
23 Université Hassan 4                          Meknes

8 rows selected.

```

### Création de la procédure INSERT\_etudiant\_universite:

```

CREATE OR REPLACE PROCEDURE INSERT_etudiant_universite
(p_ID_etudiant Etudiants_Universites.ID_etudiant%TYPE,
  p_ID_universite Etudiants_Universites.ID_universite%TYPE )
AS
  n NUMBER := 0;
  est_duplique EXCEPTION;

  BEGIN
    SELECT COUNT(*) INTO n FROM Etudiants_Universites WHERE ID_etudiant =
p_ID_etudiant AND ID_universite = p_ID_universite;
    IF n = 0 THEN
      -- verifier si l'etudiant n'est pas enregistre dans une universite
      -- Insérer la ligne dans la table Etudiants_Universites
      INSERT INTO Etudiants_Universites VALUES(p_ID_etudiant,
p_ID_universite);
      dbms_output.put_line('L''etudiant N°'||p_ID_etudiant||' est enregistre dans
l''universite N°'||p_ID_universite);
    ELSE
      ---- si l'etudiant est enregistre deja dans une universite
      RAISE est_duplique;
    END IF;
  EXCEPTION
    WHEN est_duplique THEN
      dbms_output.put_line('!!! L''etudiant N°'||p_ID_etudiant||' est deja
enregistre dans l''universite N°'||p_ID_universite||' !!!');

```

```

END ;
/
SELECT * FROM Etudiants;
SELECT * FROM Etudiants_Universites;

```

```

Administrator: Command Prompt - sqlplus admin_bourse/aissam

SQL> CREATE OR REPLACE PROCEDURE INSERT_etudiant_universite(p_ID_etudiant Etudiants_Universites.ID_etudiant%TYPE,
2      p_ID_universite Etudiants_Universites.ID_universite%TYPE > AS
3      n NUMBER := 0;
4      est_duplique EXCEPTION;
5
6      BEGIN
7          SELECT COUNT(*) INTO n FROM Etudiants_Universites WHERE ID_etudiant = p_ID_etudiant AND ID_universite = p_ID_universite;
8          IF n = 0 THEN
9              -- verifier si l'etudiant n'est pas enregistre dans une universite
10             -- Insérer la ligne dans la table Etudiants_Universites
11             INSERT INTO Etudiants_Universites VALUES(p_ID_etudiant, p_ID_universite);
12             dbms_output.put_line('L''etudiant N°'||p_ID_etudiant||' est enregistre dans l''universite N°'||p_ID_universite);
13         ELSE
14             -- si l'etudiant est enregistre deja dans une universite
15             RAISE est_duplique;
16         END IF;
17     EXCEPTION
18     WHEN est_duplique THEN
19         dbms_output.put_line('!!! L''etudiant N°'||p_ID_etudiant||' est deja enregistre dans l''universite N°'||p_ID_universite||' !!!');
20     END ;
21 /

Procedure created.

SQL>
SQL> SELECT * FROM Etudiants;

ID_ETUDIANT  NOM                PRENOM                NATIONALITE            ETRANGER
-----
21 EN-NAHEL        Med                   Marocain               0
22 Steve          Jobs                 American              1
1  EN-NAHEL        Aissam               Marocain               0
2 Amina           Ben-Ali              Marocaine              0
3 Karim           Ben-Moussa           Marocain               0
4 Fatima          Ben-Mohammed         Marocaine              0
5 Elon            Musk                 Américain             1
6 John            Smith                Britannique            1
7 Mark            Zuckerberg            Chinois                1
9 EN-NAHEL        Hamza                Marocain               0
11 EN-NAHEL       Hassan               Marocain               0

ID_ETUDIANT  NOM                PRENOM                NATIONALITE            ETRANGER
-----
12 Bill       Gates              Allenand              1

12 rows selected.

```

```

Select Administrator: Command Prompt - sqlplus admin_bourse/aissam

12 rows selected.

SQL> SELECT * FROM Etudiants_Universites;

ID_ETUDIANT  ID_UNIVERSITE
-----
1             1
2             2
3             3
4             2
5             4
6             5
7             6

7 rows selected.

```

```

SELECT * FROM Etud_Univ_Maroc;
SELECT * FROM Etud_Univ_Etrangers;

```

```
SQL> SELECT * FROM Etud_Univ_Maroc;
ID_ETUDIANT ID_UNIVERSITE
-----
1 1
2 2
3 3
4 2

SQL> SELECT * FROM Etud_Univ_Etrangers;
ID_ETUDIANT ID_UNIVERSITE
-----
5 4
6 5
7 6

SQL>
```

SELECT \* FROM Universities;

```
SQL>
SQL> SELECT * FROM Universities;
ID_UNIVERSITE NOM LOCALISATION
-----
22 Université Hassan 3 Fes
23 Université Hassan 4 Meknes
1 Université Ibn Zohr Agadir
2 Université Cadi Ayyad Marrakech
3 Université Hassan II Casablanca
4 Université Mohammed V Rabat
5 Université Moulay Ismail Meknes
6 Université Al Quaraouiyine Fes

8 rows selected.
SQL>
```

```
BEGIN
    INSERT_etudiant_universite(21, 23);
    INSERT_etudiant_universite(22, 23);
END;
/
```

```
SELECT * FROM Etudiants_Universities;
SELECT * FROM Etud_Univ_Maroc;
SELECT * FROM Etud_Univ_Etrangers;
```



```

SQL>
SQL> BEGIN
2   INSERT _etudiant_universite(21, 23);
3   INSERT _etudiant_universite(22, 23);
4   END;
5   /
PL/SQL procedure successfully completed.
SQL>
SQL> SELECT * FROM Etudiants_Universites;
ID_ETUDIANT ID_UNIVERSITE
-----
1           1
2           2
3           3
4           2
5           4
6           5
7           6
21          23
22          23
9 rows selected.
SQL> SELECT * FROM Etud_Univ_Maroc;
ID_ETUDIANT ID_UNIVERSITE
-----
1           1
2           2
3           3
4           2
21          23
SQL> SELECT * FROM Etud_Univ_Etrangers;
ID_ETUDIANT ID_UNIVERSITE
-----
5           4
6           5
7           6
22          23
SQL>

```

### Création de la procédure INSERT bourse:

```

CREATE OR REPLACE PROCEDURE INSERT_bourse(p_ID_etudiant
Bourses.ID_etudiant%TYPE,
        p_Nom Bourses.Nom%TYPE,
        p_Montant Bourses.Montant%TYPE ) AS
n NUMBER := 0;
est_duplique EXCEPTION;

BEGIN
    SELECT COUNT(*) INTO n FROM Bourses WHERE Nom = p_Nom AND
ID_etudiant = p_ID_etudiant;
    IF n = 0 THEN
        -- verifier si l'etudiant n'est pas enregistre dans une bourse
        -- Insérer la ligne dans la table Bourses
        INSERT INTO Bourses VALUES(sq_Id_Bourse.NEXTVAL, p_ID_etudiant,
p_Nom, p_Montant);
        dbms_output.put_line('L'etudiant N°'||p_ID_etudiant||' est enregistre dans
la bourse "'||p_Nom||'");
    ELSE
        ---- si l'etudiant est enregistre deja dans une bourse
        RAISE est_duplique;
    END IF;
EXCEPTION
    WHEN est_duplique THEN

```

```

        dbms_output.put_line('!!! L''etudiant N°'||p_ID_etudiant||' est deja
enregistre dans la bourse "'||p_Nom||'" !!!');
    END ;
/

```

```

Administrator: Command Prompt - sqlplus admin_bourse/aissam

SQL> CREATE OR REPLACE PROCEDURE INSERT_bourse(p_ID_etudiant Bourses.ID_etudiant%TYPE,
2      p_Nom Bourses.Nom%TYPE,
3      p_Montant Bourses.Montant%TYPE ) AS
4      n NUMBER := 0;
5      est_dupee EXCEPTION;
6
7      BEGIN
8          SELECT COUNT(*) INTO n FROM Bourses WHERE Nom = p_Nom AND ID_etudiant = p_ID_etudiant;
9          IF n = 0 THEN
10             -- verifier si l'etudiant n'est pas enregistre dans une bourse
11             -- Insérer la ligne dans la table Bourses
12             INSERT INTO Bourses VALUES(sq_Id_Bourse.NEXTVAL, p_ID_etudiant, p_Nom, p_Montant);
13             dbms_output.put_line('L''etudiant N°'||p_ID_etudiant||' est enregistre dans la bourse "'||p_Nom||'"');
14          ELSE
15             -- si l'etudiant est enregistre deja dans une bourse
16             RAISE est_dupee;
17          END IF;
18      EXCEPTION
19      WHEN est_dupee THEN
20          dbms_output.put_line('!!! L''etudiant N°'||p_ID_etudiant||' est deja enregistre dans la bourse "'||p_Nom||'" !!!');
21      END ;
22
Procedure created.

SQL>

```

```

SELECT * FROM Etudiants;
SELECT * FROM Bourses;
SELECT * FROM Bourses_Maroc;
SELECT * FROM Bourses_Etrangers;

```

```

SQL>
SQL>
SQL> SELECT * FROM Etudiants;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
21	EN-NAHEL	Med	Marocain	0
22	Steve	Jobs	American	1
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1
9	EN-NAHEL	Hamza	Marocain	0
11	EN-NAHEL	Hassan	Marocain	0
ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
12	Bill	Gates	Allemand	1

12 rows selected.

```

SQL> SELECT * FROM Bourses;

```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
30	9	Bourses Minhatty	5000
31	11	Bourses Minhatty	5000
32	12	Bourses Minhatty	5000
1	1	Bourses Minhatty	1500
2	2	Bourse Istihqaq	3000
3	3	Bourses FME	2000
4	4	Bourse AMCI	4000
5	5	Bourses Minhatty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhatty	1500

```

10 rows selected.
SQL> SELECT * FROM Bourses_Maroc;

```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
1	1	Bourses Minhatty	1500
2	2	Bourse Istihqaq	3000
3	3	Bourses FME	2000
4	4	Bourse AMCI	4000
30	9	Bourses Minhatty	5000
31	11	Bourses Minhatty	5000

```

6 rows selected.
SQL> SELECT * FROM Bourses_Etrangers;

```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
5	5	Bourses Minhatty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhatty	1500
32	12	Bourses Minhatty	5000

```

SQL>
SQL> BEGIN
2   INSERT_bourse(21, 'Bourseses FME', 3500);
3   INSERT_bourse(22, 'Bourseses FME', 3500);
4 END;
5 /
PL/SQL procedure successfully completed.
SQL>

```

BEGIN

**INSERT\_bourse**(21, 'Bourseses FME', 3500);

**INSERT\_bourse**(22, 'Bourseses FME', 3500);

END;

/

SELECT \* FROM Bourses;

SELECT \* FROM Bourses\_Maroc;

SELECT \* FROM Bourses\_Etrangers;

```

SQL>
SQL> BEGIN
2   INSERT_bourse(21, 'Bourseses FME', 3500);
3   INSERT_bourse(22, 'Bourseses FME', 3500);
4 END;
5 /
PL/SQL procedure successfully completed.
SQL>
SQL> SELECT * FROM Bourses;

```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
30	9	Bourses Minhatty	5000
34	22	Bourseses FME	3500
31	11	Bourses Minhatty	5000
32	12	Bourses Minhatty	5000
33	21	Bourseses FME	3500
1	1	Bourses Minhatty	1500
2	2	Bourse Istihqaq	3000
3	3	Bourses FME	2000
4	4	Bourse AMCI	4000
5	5	Bourses Minhatty	1500
6	6	Bourse AMCI	4000

```

ID_BOURSE ID_ETUDIANT NOM
-----
7         7 Bourses Minhatty
12 rows selected.
SQL> SELECT * FROM Bourses_Maroc;

```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
1	1	Bourses Minhatty	1500
2	2	Bourse Istihqaq	3000
3	3	Bourses FME	2000
4	4	Bourse AMCI	4000
30	9	Bourses Minhatty	5000
31	11	Bourses Minhatty	5000
33	21	Bourseses FME	3500

```

7 rows selected.

```

```
SQL> SELECT * FROM Bourses_Etrangers;
```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
5	5	Bourses Minhaty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhaty	1500
32	12	Bourses Minhaty	5000
34	22	Bourseses FME	3500

```
SQL>
```

## 2. Site du Modérateur des bourses des étudiants marocains

Dans le deuxième site, celui du **Modérateur des bourses des étudiants marocains**, je vais procéder à la copie des données des **étudiants marocains** depuis l'**Administrateur des bourses** vers ce site, ensuite, je vais créer une database link qui établira une connexion depuis ce site vers le site 1 (Administrateur des bourses), puis je vais créer les synonymes,

Enfin, j'ajouterai les contraintes de clé primaire, de clé étrangère, pour garantir l'intégrité des données dans la base du Modérateur des bourses des étudiants marocains.

Cela permettra d'avoir une base de données locale avec les informations des étudiants marocains.

### - Copie des données

```
sqlplus modérateur_bourse_marocaine/aissam
```

```
select * from cat;
```

```
Copy from admin_bourse/aissam@xe To
modérateur_bourse_marocaine/aissam@xe Replace Etudiants_Maroc using
select ID_etudiant, Nom, Prenom, Nationalite, Etranger from Etudiants where
Etranger = 0;
```

```
desc Etudiants_Maroc
set lines 150
select * from Etudiants_Maroc;
```

```

Administrator: Command Prompt - sqlplus_moderateur_bourse_marocaine/aissam
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\TP414>sqlplus_moderateur_bourse_marocaine/aissam
SQL*Plus: Release 11.2.0.2.0 Production on Mar. 30 01:44:02 2023
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
SQL>
SQL> select * from cat;
no rows selected

SQL> Copy from admin_bourse/aissam@xe To moderateur_bourse_marocaine/aissam@xe Replace Etudiants_Maroc using select ID_etudiant, Nom, Prenom, Nationalite, Etranger from Etudiants where Etranger = 0;
Array fetch/bind size is 15. (arraysize is 15)
Will commit when done. (copycommit is 0)
Maximum long size is 80. (long is 80)
Table ETUDIANTS_MAROC created.
  4 rows selected from admin_bourse@xe.
  4 rows inserted into ETUDIANTS_MAROC.
  4 rows committed into ETUDIANTS_MAROC at moderateur_bourse_marocaine@xe.

SQL>
SQL> desc Etudiants_Maroc
Name                                     Null?    Type
-----
ID_ETUDIANT                             NOT NULL NUMBER(30)
NOM                                       VARCHAR2(20)
PRENOM                                  VARCHAR2(20)
NATIONALITE                             VARCHAR2(20)
ETRANGER                                NUMBER(1)

SQL> set lines 150
SQL> select * from Etudiants_Maroc;
ID_ETUDIANT  NOM          PRENOM          NATIONALITE          ETRANGER
-----
1  EN-NAHEL    Aissam          Marocain             0
2  Amina      Ben-Ali         Marocaine            0
3  Karim      Ben-Moussa      Marocain             0
4  Fatima     Ben-Mohammed    Marocaine            0

```

**Copy** from admin\_bourse/aissam@xe To moderateur\_bourse\_marocaine/aissam@xe Replace **Universites\_Maroc** using select ID\_universite, Nom, Localisation from Universites;

desc **Universites\_Maroc**  
select \* from **Universites\_Maroc**;

```

SQL>
SQL> Copy from admin_bourse/aissam@xe To moderateur_bourse_marocaine/aissam@xe Replace Universites_Maroc using select ID_universite, Nom, Localisation from Universites;
Array fetch/bind size is 15. (arraysize is 15)
Will commit when done. (copycommit is 0)
Maximum long size is 80. (long is 80)
Table UNIVERSITES_MAROC created.
  6 rows selected from admin_bourse@xe.
  6 rows inserted into UNIVERSITES_MAROC.
  6 rows committed into UNIVERSITES_MAROC at moderateur_bourse_marocaine@xe.

SQL>
SQL> desc Universites_Maroc
Name                                     Null?    Type
-----
ID_UNIVERSITE                             NOT NULL NUMBER(30)
NOM                                       VARCHAR2(100)
LOCALISATION                             VARCHAR2(30)

SQL> select * from Universites_Maroc;
ID_UNIVERSITE  NOM          LOCALISATION
-----
1  Université Ibn Zohr          Agadir
2  Université Cadi Ayyad       Marrakech
3  Université Hassan II        Casablanca
4  Université Mohammed V       Rabat
5  Université Moulay Ismail     Meknès
6  Université Al Quaraouiyine   Fès

6 rows selected.

SQL>

```

**Copy** from admin\_bourse/aissam@xe To moderateur\_bourse\_marocaine/aissam@xe Replace **Etud\_Univ\_Maroc** using select ID\_etudiant, ID\_universite from Etudiants\_Universites where ID\_etudiant in (select ID\_etudiant from Etudiants where Etranger = 0);

desc **Etud\_Univ\_Maroc**  
select \* from **Etud\_Univ\_Maroc**;

```

SQL>
SQL> Copy from admin_bourse/aissam@xe To modérateur_bourse_marocaine/aissam@xe Replace Etud_Univ_Maroc using select ID_etudiant, ID_universite from Etudiants_Universites where ID_etudiant in (select ID_etudiant from Etudiants where Etranger = 0);
Array fetch/bind size is 15. (arraysize is 15)
Will commit when done. (copycommit is 0)
Maximum long size is 80. (long is 80)
Table ETUD_UNIV_MAROC created.
  4 rows selected from admin_bourse@xe.
  4 rows inserted into ETUD_UNIV_MAROC.
  4 rows committed into ETUD_UNIV_MAROC at modérateur_bourse_marocaine@xe.

SQL>
SQL> desc Etud_Univ_Maroc
Name                                                    Null?    Type
-----
ID_ETUDIANT                                             NOT NULL NUMBER(38)
ID_UNIVERSITE                                           NOT NULL NUMBER(38)

SQL> select * from Etud_Univ_Maroc;

ID_ETUDIANT ID_UNIVERSITE
-----
1           1
2           2
3           2
4           2

```

**Copy** from admin\_bourse/aissam@xe To modérateur\_bourse\_marocaine/aissam@xe Replace **Bourses\_Maroc** using select ID\_bourse, ID\_etudiant, Nom, Montant from Bourses where ID\_etudiant in (select ID\_etudiant from Etudiants where Etranger = 0);

desc **Bourses\_Maroc**

select \* from **Bourses\_Maroc**;

commit;

```

SQL>
SQL> Copy from admin_bourse/aissam@xe To modérateur_bourse_marocaine/aissam@xe Replace Bourses_Maroc using select ID_bourse, ID_etudiant, Nom, Montant from Bourses where ID_etudiant in (select ID_etudiant from Etudiants where Etranger = 0);
Array fetch/bind size is 15. (arraysize is 15)
Will commit when done. (copycommit is 0)
Maximum long size is 80. (long is 80)
Table BOURSES_MAROC created.
  4 rows selected from admin_bourse@xe.
  4 rows inserted into BOURSES_MAROC.
  4 rows committed into BOURSES_MAROC at modérateur_bourse_marocaine@xe.

SQL>

```

```

SQL>
SQL> desc Bourses_Maroc
Name                                                    Null?    Type
-----
ID_BOURSE                                             NOT NULL NUMBER(38)
ID_ETUDIANT                                           NOT NULL NUMBER(38)
NOM                                                    VARCHAR2(100)
MONTANT                                                NUMBER(38)

SQL> select * from Bourses_Maroc;

ID_BOURSE ID_ETUDIANT NOM                                MONTANT
-----
1         1 Bourses Minhaty                                1500
2         2 Bourses Istihqag                        3000
3         3 Bourses FME                             2000
4         4 Bourse AMCI                             4000

SQL>
SQL> commit;
Commit complete.

SQL>

```

## - Création de database link

select \* from Etudiants;

create **database link dbl\_marocaine\_to\_admin** connect to admin\_bourse identified by aissam using 'xe';

col db\_link for A30

col host for A20

select \* from USER\_DB\_LINKS;

select \* from Etudiants@**dbl\_marocaine\_to\_admin**;

```

Administrator: Command Prompt - sqlplus moderateur_bourse_marocaine/aissam

SQL> select * from Etudiants;
select * from Etudiants
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> create database link dbl_marocaine_to_admin connect to admin_bourse identified by aissam using 'xe';
Database link created.

SQL>
SQL> col db_link for A30
SQL> col host for A20
SQL> select * from USER_DB_LINKS;

```

DB_LINK	USERNAME	PASSWORD	HOST	CREATED
DBL_MAROCAINE_TO_ADMIN	ADMIN_BOURSE		xe	30/05/23

```

SQL>
SQL> select * from Etudiants@dbl_marocaine_to_admin;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1

```

7 rows selected.
SQL>

```

## - Création des Synonymes

create or REPLACE **synonym Etudiants** FOR  
admin\_bourse.Etudiants@dbl\_marocaine\_to\_admin;

select \* from **Etudiants**;

```

SQL>
SQL> create or REPLACE synonym Etudiants FOR admin_bourse.Etudiants@dbl_marocaine_to_admin;
Synonym created.

SQL> select * from Etudiants;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1

```

7 rows selected.
SQL>

```

create or REPLACE **synonym Universites** FOR  
admin\_bourse.Universites@dbl\_marocaine\_to\_admin;

select \* from **Universites**;

```

SQL>
SQL> create or REPLACE synonym Etudiants_Universites FOR admin_bourse.Etudiants_Universites@dbl_marocaine_to_admin;
Synonym created.

SQL> select * from Etudiants_Universites;

```

ID_ETUDIANT	ID_UNIVERSITE
1	1
2	2
3	3
4	2
5	4
6	5
7	6

```

7 rows selected.
SQL>

```

create or REPLACE **synonym Etudiants\_Universites** FOR  
admin\_bourse.Etudiants\_Universites@dbl\_marocaine\_to\_admin;

select \* from Etudiants\_**Universites**;

create or REPLACE **synonym Bourses** FOR  
admin\_bourse.Bourses@db1\_marocaine\_to\_admin;

select \* from **Bourses**;

```
SQL> create or REPLACE synonym Bourses FOR admin_bourse.Bourses@db1_marocaine_to_admin;
Synonym created.
SQL> select * from Bourses;
```

ID_BOURSE	ID_ETUDIANT	NOM	MONTANT
1	1	Bourses Minhaty	1500
2	2	Bourse Istihqaq	3000
3	3	Bourses FME	2000
4	4	Bourse AMCI	4000
5	5	Bourses Minhaty	1500
6	6	Bourse AMCI	4000
7	7	Bourses Minhaty	1500

```
7 rows selected.
SQL>
```

create or REPLACE **synonym sq\_Id\_Etudiant** FOR  
admin\_bourse.sq\_Id\_Etudiant@db1\_marocaine\_to\_admin;

create or REPLACE **synonym sq\_Id\_Univetsite** FOR  
admin\_bourse.sq\_Id\_Univetsite@db1\_marocaine\_to\_admin;

create or REPLACE **synonym sq\_Id\_Bourse** FOR  
admin\_bourse.sq\_Id\_Bourse@db1\_marocaine\_to\_admin;

```
SQL> create or REPLACE synonym sq_Id_Etudiant FOR admin_bourse.sq_Id_Etudiant@db1_marocaine_to_admin;
Synonym created.
SQL> create or REPLACE synonym sq_Id_Univetsite FOR admin_bourse.sq_Id_Univetsite@db1_marocaine_to_admin;
Synonym created.
SQL>
SQL> create or REPLACE synonym sq_Id_Bourse FOR admin_bourse.sq_Id_Bourse@db1_marocaine_to_admin;
Synonym created.
SQL>
```

### - Ajoute des contraintes de base

alter table Bourses\_Maroc add **primary key** (ID\_bourse);

alter table Etudiants\_Maroc add **primary key** (ID\_etudiant);

alter table Etud\_Univ\_Maroc add **primary key** (ID\_etudiant, ID\_universite);

alter table Universites\_Maroc add **primary key** (ID\_universite);

alter table Bourses\_Maroc add **foreign key**(ID\_etudiant) references  
Etudiants\_Maroc(ID\_etudiant);

alter table Etud\_Univ\_Maroc add **foreign key** (ID\_etudiant) references  
Etudiants\_Maroc(ID\_etudiant);



```

SQL> alter table Bourses_Maroc add primary key <ID_bourse>;
Table altered.
SQL> alter table Etudiants_Maroc add primary key <ID_etudiant>;
Table altered.
SQL> alter table Etud_Univ_Maroc add primary key <ID_etudiant, ID_universite>;
Table altered.
SQL> alter table Universites_Maroc add primary key <ID_universite>;
Table altered.
SQL> alter table Bourses_Maroc add foreign key<ID_etudiant> references Etudiants_Maroc<ID_etudiant>;
Table altered.
SQL> alter table Etud_Univ_Maroc add foreign key <ID_etudiant> references Etudiants_Maroc<ID_etudiant>;
Table altered.

```

alter table Etud\_Univ\_Maroc add **foreign key** (ID\_universite) references Universites\_Maroc(ID\_universite);

```

SQL> alter table Etud_Univ_Maroc add foreign key <ID_universite> references Universites_Maroc<ID_universite>;
Table altered.
SQL>

```

### 3. Site du Modérateur des bourses des étudiants étrangers

Dans le deuxième site, celui du **Modérateur des bourses des étudiants étrangers**, je vais procéder à la copie des données des étudiants étrangers depuis l'**Administrateur des bourses** vers ce site, ensuite, je vais créer une database link qui établira une connexion depuis ce site vers le site 1 (Administrateur des bourses), puis je vais créer les synonymes,

Enfin, j'ajouterai les contraintes de clé primaire, de clé étrangère, pour garantir l'intégrité des données dans la base du Modérateur des bourses des étudiants marocains.

Cela permettra d'avoir une base de données locale avec les informations des **étudiants étrangers**.

#### - Copie des données

```

sqlplus moderateur_bourse_etrananger/aissam
select * from cat;

```

```

Copy from admin_bourse/aissam@xe To
moderateur_bourse_etrananger/aissam@xe Replace Etudiants_Etrangers using
select ID_etudiant, Nom, Prenom, Nationalite, Etranger from Etudiants where
Etranger = 1;

```

```

desc Etudiants_Etrangers
set lines 150
select * from Etudiants_Etrangers;

```

```

Administrator: Command Prompt - sqlplus moderateur_bourse_etrange/aissam
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\TP414>sqlplus moderateur_bourse_etrange/aissam

SQL*Plus: Release 11.2.0.2.0 Production on Mar. 30 01:48:19 2023
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL>
SQL> select * from cat;

no rows selected

SQL> Copy from admin_bourse/aissam@xe To moderateur_bourse_etrange/aissam@xe Replace Etudiants_Etrangers using select ID_etudiant, Nom, Prenom, Nationalite, Etranger from Etudiants where Etranger = 1;

Array fetch/bind size is 15. (arraysize is 15)
Will commit when done. (copycommit is 0)
Maximum long size is 80. (long is 80)
Table ETUDIANTS_ETRANGERS created.

3 rows selected from admin_bourse@xe.
3 rows inserted into ETUDIANTS_ETRANGERS.
3 rows committed into ETUDIANTS_ETRANGERS at moderateur_bourse_etrange@xe.

SQL>
SQL> desc Etudiants_Etrangers
Name Null? Type
-----
ID_ETUDIANT NOT NULL NUMBER(38)
NOM VARCHAR2(20)
PRENOM VARCHAR2(20)
NATIONALITE VARCHAR2(20)
ETRANGER NUMBER(1)

SQL> set lines 150
SQL> select * from Etudiants_Etrangers;

ID_ETUDIANT NOM PRENOM NATIONALITE ETRANGER
-----
5 Elon Musk Américain 1
6 John Smith Britannique 1
7 Mark Zuckerberg Chinois 1

SQL>

```

**Copy** from admin\_bourse/aissam@xe To moderateur\_bourse\_etrange/aissam@xe Replace **Universites\_Etrangers** using select ID\_universite, Nom, Localisation from Universites;

desc **Universites\_Etrangers**  
select \* from **Universites\_Etrangers**;

**Copy** from admin\_bourse/aissam@xe To moderateur\_bourse\_etrange/aissam@xe Replace **Etud\_Univ\_Etrangers** using select ID\_etudiant, ID\_universite from Etudiants\_Universites where ID\_etudiant in (select ID\_etudiant from Etudiants where Etranger = 1);

```

Select Administrator: Command Prompt - sqlplus moderateur_bourse_etrange/aissam

SQL>
SQL>
SQL> Copy from admin_bourse/aissam@xe To moderateur_bourse_etrange/aissam@xe Replace Universites_Etrangers using select ID_universite, Nom, Localisation from Universites;

Array fetch/bind size is 15. (arraysize is 15)
Will commit when done. (copycommit is 0)
Maximum long size is 80. (long is 80)
Table UNIVERSITES_ETRANGERS created.

6 rows selected from admin_bourse@xe.
6 rows inserted into UNIVERSITES_ETRANGERS.
6 rows committed into UNIVERSITES_ETRANGERS at moderateur_bourse_etrange@xe.

SQL>
SQL> desc Universites_Etrangers
Name Null? Type
-----
ID_UNIVERSITE NOT NULL NUMBER(38)
NOM VARCHAR2(100)
LOCALISATION VARCHAR2(30)

SQL> select * from Universites_Etrangers;

ID_UNIVERSITE NOM LOCALISATION
-----
1 Université Ibn Zohr Agadir
2 Université Cadi Ayyad Marrakech
3 Université Hassan II Casablanca
4 Université Mohammed V Rabat
5 Université Moulay Ismail Meknes
6 Université Al Quaraouiyine Fes

6 rows selected.

SQL>
SQL> Copy from admin_bourse/aissam@xe To moderateur_bourse_etrange/aissam@xe Replace Etud_Univ_Etrangers using select ID_etudiant, ID_universite from Etudiants_Universites where ID_etudiant in (select ID_etudiant from Etudiants where Etranger = 1);

Array fetch/bind size is 15. (arraysize is 15)
Will commit when done. (copycommit is 0)
Maximum long size is 80. (long is 80)
Table ETUD_UNIV_ETRANGERS created.

3 rows selected from admin_bourse@xe.
3 rows inserted into ETUD_UNIV_ETRANGERS.
3 rows committed into ETUD_UNIV_ETRANGERS at moderateur_bourse_etrange@xe.

```

desc **Etud\_Univ\_Etrangers**

```
select * from Etud_Univ_Etrangers;
```

**Copy** from admin\_bourse/aissam@xe To  
moderateur\_bourse\_etranger/aissam@xe Replace **Bourses\_Etrangers** using  
select ID\_bourse, ID\_etudiant, Nom, Montant from Bourses where ID\_etudiant in  
(select ID\_etudiant from Etudiants where Etranger = 1);

```
desc Bourses_Etrangers
```

```
select * from Bourses_Etrangers;
```

```
commit;
```

```
SQL>
SQL> desc Etud_Univ_Etrangers
Name                                         Null?     Type
-----
ID_ETUDIANT                                NOT NULL  NUMBER(38)
ID_UNIVERSITE                             NOT NULL  NUMBER(38)

SQL> select * from Etud_Univ_Etrangers;
ID_ETUDIANT ID_UNIVERSITE
-----
5           4
6           5
7           6

SQL>
SQL> Copy from admin_bourse/aissam@xe To moderateur_bourse_etranger/aissam@xe Replace Bourses_Etrangers using select ID_bourse, ID_etudiant, Nom, Montant from Bourses where ID_etudiant in (select ID_etudiant from Etudiants where Etranger = 1);
Array fetch/bind size is 15. (arraysize is 15)
Will commit when done. (copycommit is 0)
Maximum long size is 80. (long is 80)
Table BOURSES_ETRANGERS created.
  3 rows selected from admin_bourse@xe.
  3 rows inserted into BOURSES_ETRANGERS.
  3 rows committed into BOURSES_ETRANGERS at moderateur_bourse_etranger@xe.

SQL>
SQL> desc Bourses_Etrangers
Name                                         Null?     Type
-----
ID_BOURSE                                NOT NULL  NUMBER(38)
ID_ETUDIANT                             NOT NULL  NUMBER(38)
NOM                                       VARCHAR2(100)
MONTANT                                  NUMBER(38)

SQL> select * from Bourses_Etrangers;
ID_BOURSE ID_ETUDIANT NOM                                MONTANT
-----
5         5 Bourses Minhaty          1500
6         6 Bourse AMCI          4000
7         7 Bourses Minhaty          1500

SQL>
```

## - Création de database link

```
select * from Etudiants;
```

create **database link dbl\_etranger\_to\_admin** connect to admin\_bourse  
identified by aissam using 'xe';

```
col db_link for A30
```

```
col host for A20
```

```
select * from USER_DB_LINKS;
```

```
select * from Etudiants@dbl_etranger_to_admin;
```

```

Administrator: Command Prompt - sqlplus moderateur_bourse_etrange/aissam

SQL> select * from Etudiants;
select * from Etudiants
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> create database link dbl_etrange_to_admin connect to admin_bourse identified by aissam using 'xe';
Database link created.

SQL>
SQL> col db_link for A30
SQL> col host for A20
SQL> select * from USER_DB_LINKS;

```

DB_LINK	USERNAME	PASSWORD	HOST	CREATED
DBL_ETRANGER_TO_ADMIN	ADMIN_BOURSE		xe	30/05/23

```

SQL>
SQL> select * from Etudiants@dbl_etrange_to_admin;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1

```

7 rows selected.

```

## - Création des Synonymes

create or REPLACE **synonym Etudiants** FOR Etudiants@dbl\_etrange\_to\_admin;

select \* from **Etudiants**;

create or REPLACE **synonym Universites** FOR  
Universites@dbl\_etrange\_to\_admin;

select \* from **Universites**;

```

SQL> create or REPLACE synonym Etudiants FOR Etudiants@dbl_etrange_to_admin;
Synonym created.
SQL> select * from Etudiants;

```

ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER
1	EN-NAHEL	Aissam	Marocain	0
2	Amina	Ben-Ali	Marocaine	0
3	Karim	Ben-Moussa	Marocain	0
4	Fatima	Ben-Mohammed	Marocaine	0
5	Elon	Musk	Américain	1
6	John	Smith	Britannique	1
7	Mark	Zuckerberg	Chinois	1

```

7 rows selected.

SQL>
SQL> create or REPLACE synonym Universites FOR Universites@dbl_etrange_to_admin;
Synonym created.
SQL> select * from Universites;

```

ID_UNIVERSITE	NOM	LOCALISATION
1	Université Ibn Zohr	Agadir
2	Université Cadi Ayyad	Marrakech
3	Université Hassan II	Casablanca
4	Université Mohammed VI	Rabat
5	Université Moulay Ismail	Meknès
6	Université Al Quaraouiyine	Fès

```

6 rows selected.

```

create or REPLACE **synonym Etudiants\_Universites** FOR  
Etudiants\_Universites@dbl\_etrange\_to\_admin;

select \* from **Etudiants\_Universites**;

create or REPLACE **synonym Bourses** FOR Bourses@dbl\_etrange\_to\_admin;

select \* from **Bourses**;

```
SQL> create or REPLACE synonym Etudiants_Universites FOR Etudiants_Universites@db1_etranger_to_admin;
Synonym created.
SQL> select * from Etudiants_Universites;
ID_ETUDIANT ID_UNIVERSITE
-----
1           1
2           2
3           3
4           2
5           4
6           5
7           6
7 rows selected.

SQL>
SQL> create or REPLACE synonym Bourses FOR Bourses@db1_etranger_to_admin;
Synonym created.
SQL> select * from Bourses;
ID_BOURSE ID_ETUDIANT NOM                                MONTANT
-----
1         1 Bourses Minhaty                                1500
2         2 Bourse Istihqag                               3000
3         3 Bourses FME                                    2000
4         4 Bourse AMCI                                    4000
5         5 Bourses Minhaty                                1500
6         6 Bourse AMCI                                    4000
7         7 Bourses Minhaty                                1500
7 rows selected.
```

create or REPLACE **synonym sq\_Id\_Etudiant** FOR  
sq\_Id\_Etudiant@db1\_etranger\_to\_admin;

```
SQL>
SQL> create or REPLACE synonym sq_Id_Etudiant FOR sq_Id_Etudiant@db1_etranger_to_admin;
Synonym created.
```

create or REPLACE **synonym sq\_Id\_Univetsite** FOR  
sq\_Id\_Univetsite@db1\_etranger\_to\_admin;

create or REPLACE **synonym sq\_Id\_Bourse** FOR  
sq\_Id\_Bourse@db1\_etranger\_to\_admin;

```
SQL>
SQL> create or REPLACE synonym sq_Id_Univetsite FOR sq_Id_Univetsite@db1_etranger_to_admin;
Synonym created.
SQL>
SQL> create or REPLACE synonym sq_Id_Bourse FOR sq_Id_Bourse@db1_etranger_to_admin;
Synonym created.
```

### - Ajoute des contraintes de base

alter table Bourses\_Etrangers add **primary key** (ID\_bourse);

alter table Etudiants\_Etrangers add **primary key** (ID\_etudiant);

alter table Etud\_Univ\_Etrangers add **primary key** (ID\_etudiant, ID\_universite);

alter table Universites\_Etrangers add **primary key** (ID\_universite);

alter table Bourses\_Etrangers add **foreign key**(ID\_etudiant) references  
Etudiants\_Etrangers(ID\_etudiant);

alter table Etud\_Univ\_Etrangers add **foreign key** (ID\_etudiant) references  
Etudiants\_Etrangers(ID\_etudiant);

alter table Etud\_Univ\_Etrangers add **foreign key** (ID\_universite) references  
Universites\_Etrangers(ID\_universite);

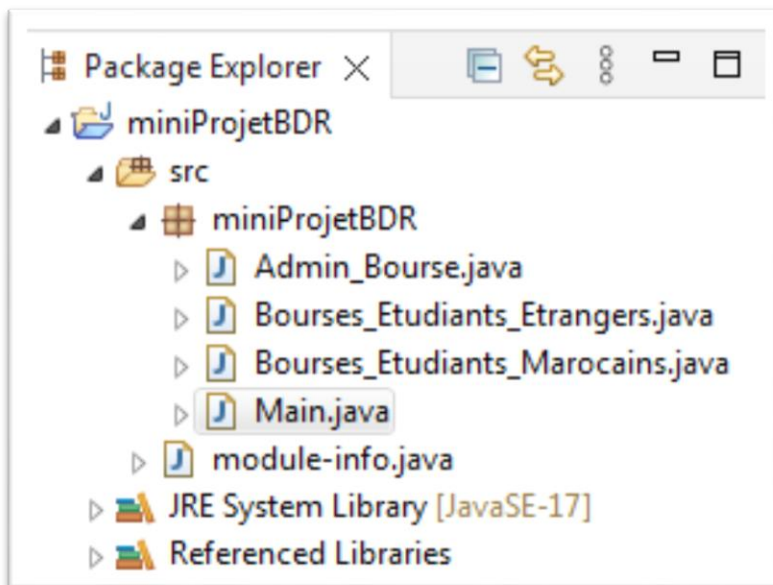
```
SQL> alter table Bourses_Etrangers add primary key (ID_bourse);
Table altered.
SQL> alter table Etudiants_Etrangers add primary key (ID_etudiant);
Table altered.
SQL> alter table Etud_Univ_Etrangers add primary key (ID_etudiant, ID_universite);
Table altered.
SQL> alter table Universites_Etrangers add primary key (ID_universite);
Table altered.
SQL>
SQL> alter table Bourses_Etrangers add foreign key (ID_etudiant) references Etudiants_Etrangers(ID_etudiant);
Table altered.
SQL> alter table Etud_Univ_Etrangers add foreign key (ID_etudiant) references Etudiants_Etrangers(ID_etudiant);
Table altered.
SQL> alter table Etud_Univ_Etrangers add foreign key (ID_universite) references Universites_Etrangers(ID_universite);
Table altered.
SQL>
```

## Chapitre 4 : Intégration avec JDBC

### L'architecture de projet JAVA JDBC

Dans cette partie, je vais vous présenter l'architecture de projet Java "**miniProjetBDR**" dans Eclipse, pour intégrer la base de données réparties. Voici les fichiers du projet :

- **Main.java** : Cette classe représente l'entrée du programme, elle contiendra la méthode principale (main) qui démarrera l'application.
- **Admin\_Bourse.java** : Cette classe est responsable de la gestion des fonctionnalités liées à l'Administrateur des bourses, elle implémentera les méthodes nécessaires pour effectuer des opérations telles que l'ajout, la modification ou la suppression des données.
- **Bourses\_Etudiants\_Marocains.java** : Cette classe est responsable de la gestion des fonctionnalités liées aux étudiants marocains, elle implémentera des méthodes permettant d'accéder les données liées aux étudiants marocains.
- **Bourses\_Etudiants\_Etranger.java** : Cette classe est responsable de la gestion des fonctionnalités liées aux étudiants étrangers, elle implémentera des méthodes permettant d'accéder les données liées aux étudiants étrangers.



## Interface console

Le menu de l'Administrateur de bourse :

```
--- JDBC ---
Choisir une option :
1. Connectez en tant que *Admin Bourse*
2. Bourses Etudiants Marocains
3. Bourses Etudiants Etrangers
4. Quitter

choix: 1
--- Admin Bourse ---
(admin_bourse/aissam)
User: admin_bourse
Password: aissam
Connection ...
Connection --> OK
Connection avec Admin Bourse --> OK
-----
```

*** ADMINISTRATEUR DE LA BOURSE AU MAROC ***	
Etudiants	Universites
1. Afficher Les Etudiants	8. Afficher Les Universites
2. *Afficher Tous Les Infos D'etudiant*	9. Afficher Les Bourses
3. Ajouter un Etudiant	10. Ajouter une Universite
4. Supprimer un Etudiant	11. Supprimer une Universite
5. Modifier un Etudiant	12. Modifier une Universite
6. Ajouter un Etudiant a Universite	
7. Ajouter un Etudiant a Bourse	
0. Quitter	

Choisir une option ==> 1

Le menu de le Modérateur bourses des étudiants marocains :



```

--- JDBC ---
Choisir une option :
1. Connectez en tant que *Admin Bourse*
2. Bourses Etudiants Marocains
3. Bourses Etudiants Etrangers
4. Quitter

choix: 2
--- Bourses Etudiants Marocains ---
(modérateur_bourse_marocaine/aissam)
User: modérateur_bourse_marocaine
Password: aissam
Connection ...
Connection --> OK
Connection avec Bourses Etudiants Marocains --> OK
-----

```

*** LA BOURSE DES ETUDIANTS -MAROCAINES- ***	
Etudiants	Universites
1. Afficher Les Etudiants Marocaines	4. Afficher Les Universites
2. Afficher Etudiants Marocaines / Universites	5. Afficher Les Bourses
3. *Afficher Tous Les Infos D'etudiant*	
0. Quitter	

Choisir une option ==> 1

-----  
LES ETUDIANTS \*MAROCAINES\*

Le menu de le Modérateur bourses des étudiants étrangers :

```

--- JDBC ---
Choisir une option :
1. Connectez en tant que *Admin Bourse*
2. Bourses Etudiants Marocains
3. Bourses Etudiants Etrangers
4. Quitter

choix: 3
--- Bourses Etudiants Etrangers ---
(moderateur_bourse_etrange/aissam)
User: modérateur_bourse_etrange
Password: aissam
Connection ...
Connection --> OK
Connection avec Bourses Etudiants Etrangers --> OK
-----

```

*** LA BOURSE DES ETUDIANTS -ETRANGES- ***	
Etudiants	Universites
1. Afficher Les Etudiants Etranges	4. Afficher Les Universites
2. Afficher Etudiants Etranges / Universites	5. Afficher Les Bourses
3. *Afficher Tous Les Infos D'etudiant*	
0. Quitter	

Choisir une option ==> 1

LES ETUDIANTS *ETRANGERS*				
ID_ETUDIANT	NOM	PRENOM	NATIONALITE	ETRANGER

## Le code complète

### Main.java

```

package miniProjetBDR;

import java.sql.*;
import java.util.Scanner;

public class Main {

    private static Scanner sc = new Scanner(System.in);
    private static String url = "jdbc:oracle:thin:@localhost:1521:xe";
    private static String user = "";
    private static String passwd = "";
    private static Connection conn = null;

    // ----- Connection Avec BD
    public static Connection connexion(String p_url, String p_user, String p_passwd,
    Connection p_conn) {
        try {
            // chargement des pilotes Oracle :
            DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
            // établissement de la connexion :

```

```

        p_conn = DriverManager.getConnection(p_url, p_user, p_passwd);

        System.out.println("Connection ... ");
        System.out.println("Connection --> OK");
    }
    // gestion des exceptions
    catch (SQLException e) {
        System.out.println("ERROR de Connection");
        e.printStackTrace();
    }
    return p_conn;
}

// ----- fonction de loop infinie pour choisir
public static void menu() throws SQLException {
    while (true) {
        System.out.println("\n\n   --- JDBC --- ");
        System.out.println(" Choisir une option : ");
        System.out.println(" 1. Connectez en tant que *Admin Bourse* ");
        System.out.println(" 2. Bourses Etudiants Marocains ");
        System.out.println(" 3. Bourses Etudiants Etrangers ");
        System.out.println(" 4. Quitter ");

        System.out.print("\nchoix: ");
        int choice = sc.nextInt();

        switch(choice) {
        case 1:
            System.out.println(" --- Admin Bourse --- ");
            System.out.println("      (admin_bourse/aissam) ");

            System.out.print(" User: ");
            user = sc.next();

            System.out.print(" Password: ");
            passwd = sc.next();

            conn = connexion(url, user, passwd, conn);
            System.out.println(" Connection avec Admin Bourse --> OK");
            System.out.println(" -----");

            Admin_Bourse ad = new Admin_Bourse(); ad.admin_bourse(conn);

            conn.close();

            break;
        case 2:
            System.out.println(" --- Bourses Etudiants Marocains --- ");
            System.out.println("      (moderateur_bourse_marocaine/aissam) ");

            System.out.print(" User: ");
            user = sc.next();

            System.out.print(" Password: ");
            passwd = sc.next();

            conn = connexion(url, user, passwd, conn);
            System.out.println(" Connection avec Bourses Etudiants Marocains -->
OK");

```

```

        System.out.println(" -----");
    );

    Bourses_Etudiants_Marocains bem = new Bourses_Etudiants_Marocains();
    bem.bourses_etudiants_marocains(conn);

    conn.close();

    break;

case 3:
    System.out.println(" --- Bourses Etudiants Etrangers --- ");
    System.out.println("      (moderateur_bourse_etranger/aissam) ");

    System.out.print("User: ");
    user = sc.next();

    System.out.print("Password: ");
    passwd = sc.next();

    conn = connexion(url, user, passwd, conn);
    System.out.println("  Connection avec Bourses Etudiants Etrangers -->
OK");
    System.out.println(" -----");

    Bourses_Etudiants_Etrangers bee = new Bourses_Etudiants_Etrangers();
    bee.bourses_etudiants_etrangers(conn);

    conn.close();

    break;

case 4:

    conn.close();

    System.out.println("Quitter ... OK");
    sc.close();
    System.exit(0);

default:
    System.out.println("Choix invalide. Sélectionner une option valide (1, 2,
3 ou 4).");
}
}
}

public static void main(String[] args) throws SQLException {
    menu();
}
}

```

## Admin\_Bourse.java

```

package miniProjetBDR;
//sqlplus admin_bourse/aissam

```

```

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class Admin_Bourse {

    private static Scanner sc = new Scanner(System.in);
    private static Statement stmt = null;
    private static ResultSet result = null;
    private static String sqlString = "";
    private static PreparedStatement pstmt = null;

    // ----- pour Admin Bourse
    public void admin_bourse(Connection conn) throws SQLException {

        boolean continuer = true;
        while (continuer) {
            System.out.printf("%-90s\n",
, "
-----");
            System.out.printf("|%-45s %-45s|\n", " ", " ");
        );
            System.out.printf("|%-90s|\n", " ", " ");
DE LA BOURSE AU MAROC ***
            System.out.printf("|%-45s|%-45s|\n", " ", " ");
            System.out.printf("-----");
            System.out.printf("|%-45s|%-45s|\n", " ", " ");
            System.out.printf("-----");
            System.out.printf("1. Afficher Les Etudiants "
, " 8. Afficher Les Universites ");
            System.out.printf("2. *Afficher Tous Les Infos
D'etudiant* " , " 9. Afficher Les Bourses ");
            System.out.printf("-----");
            System.out.printf("3. Ajouter un Etudiant "
, " 10. Ajouter une Universite ");
            System.out.printf("4. Supprimer un Etudiant "
, " 11. Supprimer une Universite ");
            System.out.printf("5. Modifier un Etudiant "
, " 12. Modifier une Universite ");
            System.out.printf("-----");
            System.out.printf("6. Ajouter un Etudiant a Universite "
, " ");
            System.out.printf("7. Ajouter un Etudiant a Bourse "
, " ");
            System.out.printf("-----");
            System.out.printf("0. Quitter"
, " ");
            System.out.printf("45s|\n", " ", " ");
        );
    }
}

```

```

System.out.print("\n Choisir une option ==> ");
int choix = sc.nextInt();

switch (choix) {
case 1:
    afficher_etudiant(conn);
    break;
case 2:
    System.out.print("\n Enter ID de l'etudiant: ");
    int id_etud = sc.nextInt();
    etudiant_tout_infos(conn, id_etud);
    break;
case 3:
    ajouter_etudiant_(conn);
    break;
case 4:
    supprimer_etudiant(conn);
    break;
case 5:
    modifier_etudiant(conn);
    break;
case 6:
    ajouter_etudiant_universite_(conn);
    break;
case 7:
    ajouter_etudiant_bourse_(conn);
    break;
case 8:
    afficher_universite(conn);
    break;
case 9:
    afficher_bourse(conn);
    break;
case 10:
    ajouter_universite_(conn);
    break;
case 11:
    supprimer_universite(conn);
    break;
case 12:
    modifier_universite(conn);
    break;
case 0:
    continuer = false;
    break;
default:
    System.out.println("\n          ****!* Choix invalide *!****!");
}
}
}

```

```

public static void afficher_etudiant(Connection conn) throws SQLException {
    stmt = conn.createStatement();
    result = stmt.executeQuery("SELECT ID_ETUDIANT, NOM, PRENOM, NATIONALITE,
ETRANGER FROM Etudiants");
}

```

```

System.out.println("\n-----");
System.out.println("                                LES ETUDIANTS MAROCAINES ET ETRANGERS");
System.out.println("-----");
System.out.printf("|%-14s|%-20s|%-20s|%-20s|%-14s|\n", " ID_ETUDIANT", "
NOM", " PRENOM", " NATIONALITE", " ETRANGER");
System.out.println("-----");
while (result.next()) {
    System.out.printf("| %12d| %18s| %18s| %18s| %12d|\n",
result.getInt(1), result.getString(2), result.getString(3), result.getString(4),
result.getInt(5));
}
System.out.printf("|%-14s|%-20s|%-20s|%-20s|%-14s|\n\n\n",
"_____", "_____", "_____",
"_____", "_____"");
};

public static void afficher_universite(Connection conn) throws SQLException {
    stmt = conn.createStatement();
    result = stmt.executeQuery("SELECT ID_UNIVERSITE, NOM, LOCALISATION FROM
UNIVERSITES");

    System.out.println("\n-----");
    System.out.println("                                LES UNIVERSITES ");
    System.out.println("-----");
    System.out.printf("|%-16s|%-35s|%-20s|\n", " ID_UNIVERSITE", " NOM", "
LOCALISATION");
    System.out.println("-----");
    while (result.next()) {
        System.out.printf("| %14d| %33s| %18s|\n", result.getInt(1),
result.getString(2), result.getString(3));
    }
    System.out.printf("|%-16s|%-35s|%-20s|\n\n\n", "_____",
"_____"");
};

public static void afficher_etudiant_universite(Connection conn) throws
SQLException {
    stmt = conn.createStatement();
    result = stmt.executeQuery("SELECT ID_Etudiant, ID_UNIVERSITE FROM
ETUDIANTS_UNIVERSITES");

    System.out.println("\n-----");
    System.out.println(" LES ETUDIANTS_UNIVERSITES ");
    System.out.println("-----");
    System.out.printf("|%-16s|%-16s|\n", " ID_ETUDIANT", " ID_UNIVERSITE");
    System.out.println("-----");
    while (result.next()) {
        System.out.printf("| %14d| %14d|\n", result.getInt(1),
result.getInt(2));
    }
    System.out.printf("|%-16s|%-16s|\n\n\n", "_____",
"_____"");
};

```

```

};

public static void afficher_bourse(Connection conn) throws SQLException {
    stmt = conn.createStatement();
    result = stmt.executeQuery("SELECT ID_Bourse, ID_Etudiant, Nom, Montant FROM BOURSES");

    System.out.println("\n-----");
    System.out.println("          LES BOURSES ");
    System.out.println("-----");
    System.out.printf("|%-16s|%-16s|%-20s|%-15s|\n", " ID_BOURSE", " ID_ETUDIANT", " NOM", " MONTANT");
    System.out.println("-----");
    while (result.next()) {
        System.out.printf("| %14d| %14d| %18s| %13d|\n", result.getInt(1), result.getInt(2), result.getString(3), result.getInt(4));
    }
    System.out.printf("|%-16s|%-16s|%-20s|%-15s|\n\n\n", " _____", " _____", " _____");
};

public static void etudiant_tout_infos(Connection conn, int p_ID_Etudiant) throws SQLException {
    sqlString = "SELECT ID_Etudiant, Nom, Prenom, Nationalite, Etranger FROM Etudiants WHERE ID_Etudiant = ?";
    pstmt = conn.prepareStatement(sqlString);
    pstmt.setInt(1, p_ID_Etudiant);
    result = pstmt.executeQuery();
    if (!result.isBeforeFirst()) {
        System.out.println(" ***** L'etudiant de ID N°"+p_ID_Etudiant+" n'existe pas *****\n\n\n");
    } else {
        while (result.next()) {
            System.out.println("\n-----");
            System.out.println("          LES INFORMATION DE L'ETUDIANT:");
            System.out.println("-----");
            System.out.println("          + Personal Information: ");
            System.out.println("          - ID_ETUDIANT: "+result.getInt(1));
            System.out.println("          - NOM: "+result.getString(2));
            System.out.println("          - PRENOM: "+result.getString(3));
            System.out.println("          - NATIONALITE: "+result.getString(4));
            System.out.print("          ETRANGER: ");
            if (result.getInt(5)==1) {
                System.out.println("Oui");
            } else {
                System.out.println("Non");
            }
        }
    }

    sqlString = "SELECT ID_Universite, Nom, Localisation FROM Universites WHERE ID_Universite IN ( SELECT ID_Universite FROM Etudiants_Universites WHERE ID_Etudiant = ? )";
}

```



```

        pstmt = conn.prepareStatement(sqlString);
        pstmt.setInt(1, p_ID_Etudiant);
        result = pstmt.executeQuery();
        while (result.next()) {
            System.out.println("                + Universite: ");
            System.out.println("                - ID_UNIVERSITE: "+result.getInt(1));
            System.out.println("                - NOM : "+result.getString(2));
            System.out.println("                - LOCALISATION : "+result.getString(3));
        }

        sqlString = "SELECT ID_Bourse, Nom, Montant FROM Bourses WHERE ID_Etudiant =
?";
        pstmt = conn.prepareStatement(sqlString);
        pstmt.setInt(1, p_ID_Etudiant);
        result = pstmt.executeQuery();
        while (result.next()) {
            System.out.println("                + Bourse: ");
            System.out.println("                - ID_BOURSE: "+result.getInt(1));
            System.out.println("                - NOM : "+result.getString(2));
            System.out.println("                - MONTANT : "+result.getInt(3));
        }

        System.out.println("_____
\n\n\n");
    }
};

//methode 1 ; sans procedure
public static void ajouter_etudiant(Connection conn) throws SQLException {
    String sqlString = "INSERT INTO ETUDIANTS values(?,?,?,?,?)";
    //INSERT INTO Etudiants VALUES(sq_Id_Etudiant.NEXTVAL, 'EN-NAHEL', 'Aissam',
'Marocain', 0);
    pstmt = conn.prepareStatement(sqlString);

    Statement stmt_sq_Id_Etudiant = conn.createStatement();
    ResultSet result_sq_Id_Etudiant = stmt_sq_Id_Etudiant.executeQuery("SELECT
sq_Id_Etudiant.NEXTVAL FROM dual");

    int sq_Id_Etudiant_nextVal = 0;
    String nom_etud = "";
    String prenom_etud = "";
    String nationalite_etud = "";
    int etranger_etud = 0;

    if (result_sq_Id_Etudiant.next()) {
        sq_Id_Etudiant_nextVal = result_sq_Id_Etudiant.getInt(1);

        System.out.println("\n\n        - Ajouter un Etudiant - ");
        System.out.println("\n        Entrez les information de l'etudiant : ");
        System.out.print("\n        -Nom: ");
        nom_etud = sc.next();
        sc.nextLine();

        System.out.print("\n        -Prenom: ");
        prenom_etud = sc.next();
        sc.nextLine();

        System.out.print("\n        -Nationalite: ");

```

```

        nationalite_etud = sc.next();
        sc.nextLine();

        System.out.print("\n -Est-ce un étudiant marocain? (oui / non): ");
        String str_maroc = sc.next();
        sc.nextLine();

        if(str_maroc.toLowerCase().equals("non")){
            etranger_etud = 1;
        }
    }

    pstmt.setInt(1, sq_Id_Etudiant_nextVal);
    pstmt.setString(2, nom_etud);
    pstmt.setString(3, prenom_etud);
    pstmt.setString(4, nationalite_etud);
    pstmt.setInt(5, etranger_etud);

    pstmt.executeUpdate();

    System.out.println("\n                ==> Ajouter Etudiant '"+nom_etud+"
    "+prenom_etud+"' --> OK");
}

//methode 2 : avec la procedure INSERT_etudiant(p_Nom, p_Prenom, p_Nationalite,
p_Etranger)
public static void ajouter_etudiant(Connection conn) throws SQLException {
    String nom_etud = "";
    String prenom_etud = "";
    String nationalite_etud = "";
    int etranger_etud = 0;

    sc.nextLine();

    System.out.println("\n\n - Ajouter un Etudiant - ");
    System.out.println("\n Entrez les information de l'etudiant : ");
    System.out.print("\n -Nom: ");
    nom_etud = sc.nextLine();
    sc.nextLine();

    System.out.print("\n -Prenom: ");
    prenom_etud = sc.next();
    sc.nextLine();

    System.out.print("\n -Nationalite: ");
    nationalite_etud = sc.next();
    sc.nextLine();

    System.out.print("\n -Est-ce un étudiant marocain? (oui / non): ");
    String str_maroc = sc.next();
    sc.nextLine();

    if(str_maroc.toLowerCase().equals("non")){
        etranger_etud = 1;
    }

    //INSERT_etudiant(p_Nom, p_Prenom, p_Nationalite, p_Etranger)

```

```

CallableStatement etatAppelable = conn.prepareStatement("{call
INSERT_etudiant(?,?,?,?)}");

etatAppelable.setString(1,nom_etud);
etatAppelable.setString(2,prenom_etud);
etatAppelable.setString(3,nationalite_etud);
etatAppelable.setInt(4,etranger_etud);

etatAppelable.execute();

System.out.println("\n                ==> Ajouter Etudiant '" +nom_etud+"
"+prenom_etud+" ' --> OK");
}

//methode ; sans procedure
public static void supprimer_etudiant(Connection conn) throws SQLException {

    afficher_etudiant(conn);

    System.out.println("\n\n        - Supprimer un Etudiant - ");
    System.out.print("\n        Entrez l'ID de l'Etudiant: ");
    int id_etud = sc.nextInt();

    pstmt = conn.prepareStatement("DELETE FROM ETUDIANTS where ID_ETUDIANT = ?");
    // Lier les variables
    pstmt.setInt(1, id_etud);
    pstmt.executeUpdate();

    System.out.println("\n                ==> Supprimer Etudiant ID='"+id_etud+"' -->
OK");
}

//methode ; sans procedure
public static void modifier_etudiant(Connection conn) throws SQLException {

    afficher_etudiant(conn);

    System.out.println("\n\n        - Modifier un Etudiant - ");
    System.out.print("\n        Entrez ID de l'etudiant qui sera modifié: ");
    int id_etud = sc.nextInt();
    sc.nextLine();

    System.out.print("        -Nom de l'etudiant: ");
    String nom_etud = sc.nextLine();
    sc.nextLine();

    System.out.print("        -Prenom de l'etudiant: ");
    String prenom_etud = sc.nextLine();
    sc.nextLine();

    System.out.print("        -Nationalite de l'etudiant: ");
    String nationalite_etud = sc.nextLine();
    sc.nextLine();

    int etranger_etud = 0;

```

```

System.out.print("    -Est-ce un étudiant marocain? (oui / non): ");
String str_maroc = sc.next();
sc.nextLine();
if(str_maroc.toLowerCase().equals("non")){
    etranger_etud = 1;
}

pstmt = conn.prepareStatement("UPDATE ETUDIANTS SET NOM = ? , PRENOM = ? ,
NATIONALITE = ? , ETRANGER = ? where ID_ETUDIANT = ?");

// Lier les variables
pstmt.setString(1, nom_etud);
pstmt.setString(2, prenom_etud);
pstmt.setString(3, nationalite_etud);
pstmt.setInt(4, etranger_etud);
pstmt.setInt(5, id_etud);
pstmt.executeUpdate();

System.out.println("\n                ==> Modifier Etudiant ID='"+id_etud+"' -->
OK");
}

//methode : avec la procedure INSERT_etudiant_universite(p_ID_etudiant,
p_ID_universite)
public static void ajouter_etudiant_universite(Connection conn) throws
SQLException {

    afficher_etudiant(conn);
    afficher_universite(conn);

    System.out.println("\n\n    - Ajouter un Etudiant a Universite - ");

    System.out.print("\n    -Entrez le ID de l'etudiant: ");
    int id_etud = sc.nextInt();
    sc.nextLine();

    System.out.print("\n    -Entrez le ID de l'universite: ");
    int id_univ = sc.nextInt();
    sc.nextLine();

    //INSERT_etudiant_universite(p_ID_etudiant, p_ID_universite)
    CallableStatement etatAppelable = conn.prepareCall("{call
INSERT_etudiant_universite(?,?)");

    etatAppelable.setInt(1,id_etud);
    etatAppelable.setInt(2,id_univ);

    etatAppelable.execute();

    System.out.println("\n                ==> Ajouter Etudiant de ID = '"+id_etud+" a
l'Universite ID = '"+id_univ+"' --> OK");
}

//methode : avec la procedure INSERT_bourse(p_ID_etudiant, p_Nom, p_Montant)

```

```

    public static void ajouter_etudiant_bourse_(Connection conn) throws SQLException
    {

        afficher_etudiant(conn);
        afficher_bourse(conn);

        System.out.println("\n\n    - Ajouter un Etudiant a Une Bourse - ");
        System.out.println("\n    Entrez les information: ");

        System.out.print("\n    -ID de l'etudiant: ");
        int id_etud = sc.nextInt();
        sc.nextLine();

        System.out.print("\n    -Nom de la Bourse: ");
        String nom_brs = sc.nextLine();
        sc.nextLine();

        System.out.print("\n    -Montant de la Bourse: ");
        int montant_brs = sc.nextInt();
        sc.nextLine();

        //INSERT_bourse(p_ID_etudiant, p_Nom, p_Montant)
        CallableStatement etatAppelable = conn.prepareCall("{call
INSERT_bourse(?,?,?)}");

        etatAppelable.setInt(1,id_etud);
        etatAppelable.setString(2,nom_brs);
        etatAppelable.setInt(3,montant_brs);

        etatAppelable.execute();

        System.out.println("\n                ==> Ajouter Etudiant de ID='"+id_etud+"' a la
Bourse '"+nom_brs+"' --> OK");
    }

    //methode : avec la procedure INSERT_universite(p_Nom, p_Localisation)
    public static void ajouter_universite_(Connection conn) throws SQLException {

        System.out.println("\n\n    - Ajouter une Universite - ");
        System.out.println("\n    Entrez les information de l'Universite : ");

        sc.nextLine();
        System.out.print("\n    -Nom: ");
        String nom_univ = sc.nextLine();
        sc.nextLine();

        System.out.print("\n    -Localisation: ");
        String loc_univ = sc.nextLine();
        sc.nextLine();

        //INSERT_universite(p_Nom, p_Localisation)
        CallableStatement etatAppelable = conn.prepareCall("{call
INSERT_universite(?,?)}");

        etatAppelable.setString(1,nom_univ);
        etatAppelable.setString(2,loc_univ);
    }

```

```

etatAppelable.execute();

    System.out.println("\n                ==> Ajouter Universite "+nom_univ+" --> OK");
}

//methode ; sans procedure
public static void supprimer_universite(Connection conn) throws SQLException {

    afficher_universite(conn);

    System.out.println("\n\n        - Supprimer une Universite - ");
    System.out.print("\n        Entrez l'ID de l'Universite: ");
    int id_univ = sc.nextInt();

    pstmt = conn.prepareStatement("DELETE FROM UNIVERSITES where ID_UNIVERSITE =
?");
    // Lier les variables
    pstmt.setInt(1, id_univ);
    pstmt.executeUpdate();

    System.out.println("\n                ==> Supprimer Universite ID='"+id_univ+"' -->
OK");
}

//methode ; sans procedure
public static void modifier_universite(Connection conn) throws SQLException {

    afficher_universite(conn);

    System.out.println("\n\n        - Modifier une Universite - ");
    System.out.print("\n        Entrez ID de l'universite qui sera modifié: ");
    int id_univ = sc.nextInt();
    sc.nextLine();

    System.out.print("        -Nom de l'Universite: ");
    String nom_univ = sc.nextLine();
    sc.nextLine();

    System.out.print("        -Localisation de l'Universite: ");
    String loc_univ = sc.nextLine();
    sc.nextLine();

    pstmt = conn.prepareStatement("UPDATE UNIVERSITES SET NOM = ? , LOCALISATION =
? where ID_UNIVERSITE = ?");

    // Lier les variables
    pstmt.setString(1, nom_univ);
    pstmt.setString(2, loc_univ);
    pstmt.setInt(3, id_univ);
    pstmt.executeUpdate();

    System.out.println("\n                ==> Modifier Universite ID='"+id_univ+"' -->
OK");
}

```







```

        System.out.println("\n-----");
        System.out.println("                LES UNIVERSITES *MAROCAINES*");
        System.out.println("-----");
        System.out.printf("|%-16s|%-35s|%-20s|\n", " ID_UNIVERSITE", " NOM", "
LOCALISATION");
        System.out.println("-----");
        while (result.next()) {
            System.out.printf("| %14d| %33s| %18s|\n", result.getInt(1),
result.getString(2), result.getString(3));
        }
        System.out.printf("|%-16s|%-35s|%-20s|\n\n\n", " _____",
" _____", " _____");
    };

    public static void afficher_etudiant_universite(Connection conn) throws
SQLException {
        stmt = conn.createStatement();
        result = stmt.executeQuery("SELECT ID_Etudiant, ID_UNIVERSITE FROM
ETUD_UNIV_MAROC");

        System.out.println("\n-----");
        System.out.println(" LES ETUDIANTS_UNIVERSITES *MAROCAINES*");
        System.out.println("-----");
        System.out.printf("|%-18s|%-18s|\n", " ID_ETUDIANT", " ID_UNIVERSITE");
        System.out.println("-----");
        while (result.next()) {
            System.out.printf("| %16d| %16d|\n", result.getInt(1),
result.getInt(2));
        }
        System.out.printf("|%-18s|%-18s|\n\n\n", " _____",
" _____");
    };

    public static void afficher_bourse(Connection conn) throws SQLException {
        stmt = conn.createStatement();
        result = stmt.executeQuery("SELECT ID_Bourse, ID_Etudiant, Nom, Montant FROM
BOURSES_MAROC");

        System.out.println("\n-----");
        System.out.println("                LES BOURSES DES ETUDIANTS *MAROCAINES* ");
        System.out.println("-----");
        System.out.printf("|%-16s|%-16s|%-20s|%-15s|\n", " ID_BOURSE", "
ID_ETUDIANT", " NOM", " MONTANT");
        System.out.println("-----");
        while (result.next()) {
            System.out.printf("| %14d| %14d| %18s| %13d|\n", result.getInt(1),
result.getInt(2), result.getString(3), result.getInt(4));
        }
        System.out.printf("|%-16s|%-16s|%-20s|%-15s|\n\n\n", " _____",
" _____", " _____", " _____");
    };

```

```

    public static void etudiant_tout_infos(Connection conn, int p_ID_Etudiant)
    throws SQLException {

        sqlString = "SELECT ID_Etudiant, Nom, Prenom, Nationalite, Etranger FROM
Etudiants_Maroc WHERE ID_Etudiant = ?";
        pstmt = conn.prepareStatement(sqlString);
        pstmt.setInt(1, p_ID_Etudiant);
        result = pstmt.executeQuery();
        if (!result.isBeforeFirst()) {
            System.out.println(" **** L'etudiant marocain de ID N°"+p_ID_Etudiant+"
n'existe pas ****\n\n");
        }else {
            while (result.next()) {
                System.out.println("\n-----
-----");
                System.out.println("      MAROC : LES INFORMATION DE L'ETUDIANT:
"+result.getString(3)+" "+result.getString(2));
                System.out.println("-----
-----");
                System.out.println("      + Personal Information: ");
                System.out.println("      - ID_ETUDIANT: "+result.getInt(1));
                System.out.println("      - NOM: "+result.getString(2));
                System.out.println("      - PRENOM: "+result.getString(3));
                System.out.println("      - NATIONALITE: "+result.getString(4));
                System.out.print("      ETRANGER: ");
                if(result.getInt(5)==1) {
                    System.out.println("Oui");
                }else {
                    System.out.println("Non");
                }
            }

            sqlString = "SELECT ID_Universite, Nom, Localisation FROM Universites_Maroc
WHERE ID_Universite IN ( SELECT ID_Universite FROM Etud_Univ_Maroc WHERE
ID_Etudiant = ? )";
            pstmt = conn.prepareStatement(sqlString);
            pstmt.setInt(1, p_ID_Etudiant);
            result = pstmt.executeQuery();
            while (result.next()) {
                System.out.println("      + Universite: ");
                System.out.println("      - ID_UNIVERSITE: "+result.getInt(1));
                System.out.println("      - NOM : "+result.getString(2));
                System.out.println("      - LOCALISATION : "+result.getString(3));
            }

            sqlString = "SELECT ID_Bourse, Nom, Montant FROM Bourses_Maroc WHERE
ID_Etudiant = ?";
            pstmt = conn.prepareStatement(sqlString);
            pstmt.setInt(1, p_ID_Etudiant);
            result = pstmt.executeQuery();
            while (result.next()) {
                System.out.println("      + Bourse: ");
                System.out.println("      - ID_BOURSE: "+result.getInt(1));
                System.out.println("      - NOM : "+result.getString(2));
                System.out.println("      - MONTANT : "+result.getInt(3));
            }

```

```

System.out.println("
\n\n\n");
    }
};
}

```

## Bourses\_Etudiants\_Etrangers.java

```

package miniProjetBDR;
//sqlplus moderateur_bourse_etranger/aissam

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class Bourses_Etudiants_Etrangers {

    private static Scanner sc = new Scanner(System.in);
    private static Statement stmt = null;
    private static ResultSet result = null;
    private static String sqlString = "";
    private static PreparedStatement pstmt = null;

    // ----- pour Bourses Etudiants Etrangers
    public void bourses_etudiants_etrangers(Connection conn) throws SQLException {

        boolean continuer = true;
        while (continuer) {
            System.out.printf("%-100s\n"
, "
");
            System.out.printf("|%-50s %-50s|\n", "
", "
");
            System.out.printf("|%-100s|\n", "
", "
*** LA BOURSE
DES ETUDIANTS -ETRANGES- ***
");
            System.out.printf("|%-50s|%-50s|\n", "
", "
-----
");
            System.out.printf("|%-50s|%-50s|\n", "
", "
Etudiants
", "
Universites
");
            System.out.printf("|%-50s|%-50s|\n", "
", "
-----
");
            System.out.printf("|%-50s|%-50s|\n", "
", "
1. Afficher Les Etudiants Etranges"
, "
4. Afficher Les Universites ");
            System.out.printf("|%-50s|%-50s|\n", "
", "
2. Afficher Etudiants Etranges /
Universites "
, "
5. Afficher Les Bourses ");
            System.out.printf("|%-50s|%-50s|\n", "
", "
-----
");
            System.out.printf("|%-50s|%-50s|\n", "
", "
3. *Afficher Tous Les Infos
D'etudiant* "
, "
");
            System.out.printf("|%-50s|%-50s|\n", "
", "
-----
");
            System.out.printf("|%-50s|%-50s|\n", "
", "
0. Quitter"
, "
");
        }
    }
}

```

```

        System.out.printf("|%-50s|%-50s|\n", "_____", "_____"
    );

    System.out.print("\n Choisir une option ==> ");
    int choix = sc.nextInt();

    switch (choix) {
        case 1:
            afficher_etudiant(conn);
            break;
        case 2:
            afficher_etudiant_universite(conn);
            break;
        case 3:
            System.out.print("\n Enter ID de l'etudiant etrange: ");
            int id_etud = sc.nextInt();
            etudiant_tout_infos(conn, id_etud);
            break;
        case 4:
            afficher_universite(conn);
            break;
        case 5:
            afficher_bourse(conn);
            break;
        case 0:
            continuer = false;
            break;
        default:
            System.out.println("\n          !***!!* Choix invalide !***!!*");
    }
}
}

```

```

public static void afficher_etudiant(Connection conn) throws SQLException {
    stmt = conn.createStatement();
    result = stmt.executeQuery("SELECT ID_ETUDIANT, NOM, PRENOM, NATIONALITE,
    ETRANGER FROM Etudiants_Etrangers");

    System.out.println("\n-----");
    System.out.println("          LES ETUDIANTS *ETRANGERS* ");
    System.out.println("-----");
    System.out.printf("|%-14s|%-20s|%-20s|%-20s|%-14s|\n", " ID_ETUDIANT", "
    NOM", " PRENOM", " NATIONALITE", " ETRANGER");
    System.out.println("-----");
    while (result.next()) {
        System.out.printf("| %12d| %-18s| %-18s| %-18s| %12d|\n",
        result.getInt(1), result.getString(2), result.getString(3), result.getString(4),
        result.getInt(5));
    }
    System.out.printf("|%-14s|%-20s|%-20s|%-20s|%-14s|\n\n\n",
    "_____", "_____", "_____", "_____", "_____"
    );
}

```

```

    public static void afficher_universite(Connection conn) throws SQLException {
        stmt = conn.createStatement();
        result = stmt.executeQuery("SELECT ID_UNIVERSITE, NOM, LOCALISATION FROM
UNIVERSITES_ETRANGERS");

        System.out.println("\n-----");
        System.out.println("                LES UNIVERSITES *ETRANGERS* ");
        System.out.println("-----");
        System.out.printf("|%-16s|%-35s|%-20s|\n", " ID_UNIVERSITE", " NOM", "
LOCALISATION");
        System.out.println("-----");
        while (result.next()) {
            System.out.printf("| %14d| %33s| %18s|\n", result.getInt(1),
result.getString(2), result.getString(3));
        }
        System.out.printf("|%-16s|%-35s|%-20s|\n\n\n", " _____",
" _____");
    };

    public static void afficher_etudiant_universite(Connection conn) throws
SQLException {
        stmt = conn.createStatement();
        result = stmt.executeQuery("SELECT ID_Etudiant, ID_UNIVERSITE FROM
ETUD_UNIV_ETRANGERS");

        System.out.println("\n-----");
        System.out.println(" LES ETUDIANTS_UNIVERSITES *ETRANGERS*");
        System.out.println("-----");
        System.out.printf("|%-18s|%-18s|\n", " ID_ETUDIANT", " ID_UNIVERSITE");
        System.out.println("-----");
        while (result.next()) {
            System.out.printf("| %16d| %16d|\n", result.getInt(1),
result.getInt(2));
        }
        System.out.printf("|%-18s|%-18s|\n\n\n", " _____",
" _____");
    };

    public static void afficher_bourse(Connection conn) throws SQLException {
        stmt = conn.createStatement();
        result = stmt.executeQuery("SELECT ID_Bourse, ID_Etudiant, Nom, Montant FROM
BOURSES_ETRANGERS");

        System.out.println("\n-----");
        System.out.println("                LES BOURSES DES ETUDIANTS *ETRANGERS* ");
        System.out.println("-----");
        System.out.printf("|%-16s|%-16s|%-20s|%-15s|\n", " ID_BOURSE", "
ID_ETUDIANT", " NOM", " MONTANT");
        System.out.println("-----");
        while (result.next()) {
            System.out.printf("| %14d| %14d| %18s| %13d|\n", result.getInt(1),
result.getInt(2), result.getString(3), result.getInt(4));
        }
    }

```

```

    }
    System.out.printf("|%-16s|%-16s|%-20s|%-15s|\n\n\n", "_____",
"_____", "_____", "_____"");
};

    public static void etudiant_tout_infos(Connection conn, int p_ID_Etudiant)
    throws SQLException {

        sqlString = "SELECT ID_Etudiant, Nom, Prenom, Nationalite, Etranger FROM
Etudiants_Etrangers WHERE ID_Etudiant = ?";
        pstmt = conn.prepareStatement(sqlString);
        pstmt.setInt(1, p_ID_Etudiant);
        result = pstmt.executeQuery();
        if (!result.isBeforeFirst()) {
            System.out.println(" *!*!*!* L'etudiant etranger de ID N°"+p_ID_Etudiant+"
n'existe pas *!*!*!*");
        }else {
            while (result.next()) {
                System.out.println("\n-----
-----");
                System.out.println("      ETRANGERS : LES INFORMATION DE L'ETUDIANT:
"+result.getString(3)+" "+result.getString(2));
                System.out.println("-----
-----");
                System.out.println("          + Personal Information: ");
                System.out.println("          - ID_ETUDIANT: "+result.getInt(1));
                System.out.println("          - NOM: "+result.getString(2));
                System.out.println("          - PRENOM: "+result.getString(3));
                System.out.println("          - NATIONALITE: "+result.getString(4));
                System.out.print("          ETRANGER: ");
                if(result.getInt(5)==1) {
                    System.out.println("Oui");
                }else {
                    System.out.println("Non");
                }
            }

            sqlString = "SELECT ID_Universite, Nom, Localisation FROM
Universites_Etrangers WHERE ID_Universite IN ( SELECT ID_Universite FROM
Etud_Univ_Etrangers WHERE ID_Etudiant = ? )";
            pstmt = conn.prepareStatement(sqlString);
            pstmt.setInt(1, p_ID_Etudiant);
            result = pstmt.executeQuery();
            while (result.next()) {
                System.out.println("          + Universite: ");
                System.out.println("          - ID_UNIVERSITE: "+result.getInt(1));
                System.out.println("          - NOM : "+result.getString(2));
                System.out.println("          - LOCALISATION : "+result.getString(3));
            }

            sqlString = "SELECT ID_Bourse, Nom, Montant FROM Bourses_Etrangers WHERE
ID_Etudiant = ?";
            pstmt = conn.prepareStatement(sqlString);
            pstmt.setInt(1, p_ID_Etudiant);
            result = pstmt.executeQuery();
            while (result.next()) {
                System.out.println("          + Bourse: ");

```

```

        System.out.println("          - ID_BOURSE: "+result.getInt(1));
        System.out.println("          - NOM : "+result.getString(2));
        System.out.println("          - MONTANT : "+result.getInt(3));
    }

    System.out.println("_____ \n\n");
}
};
}

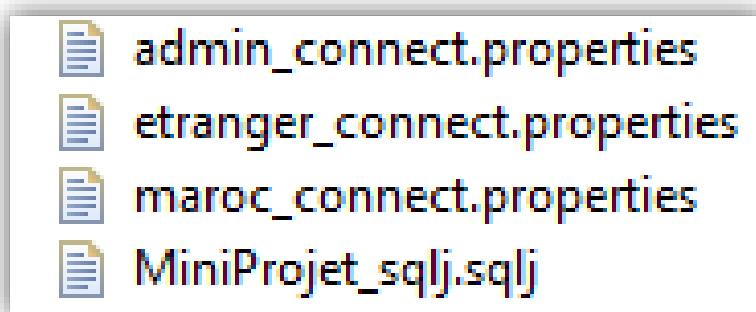
```

## Chapitre 5 : Utilisation de SQLJ

### Architecture SQLJ

Pour intégrer SQLJ avec la base de données réparties, nous avons créé quatre fichiers importants. Voici une description de chacun fichier :

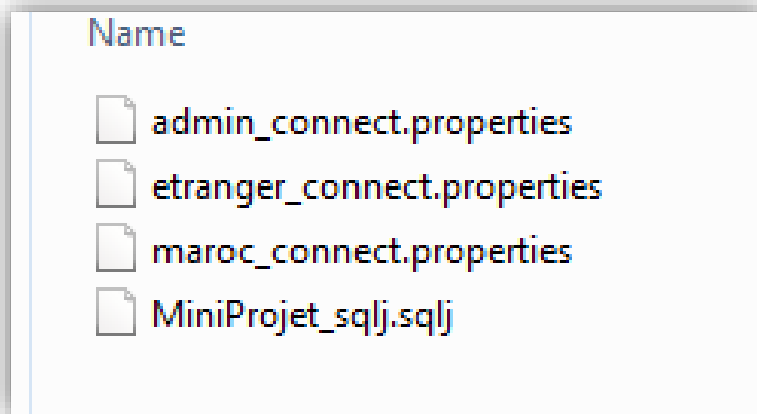
- **MiniProjet\_sqlj.sqlj** : Ce fichier contient le code SQLJ qui sera utilisé pour interagir avec la base de données, il comprendra des instructions SQLJ pour effectuer l'insertion, la mise à jour et la suppression, et aussi la récupération de données à partir de la base de données.
- **admin\_connect.properties** : Ce fichier contient les informations de connexion de l'utilisateur Administrateur des bourses, il comprendra l'adresse et le nom d'utilisateur et le mot de passe, pour permet l'application de se connecter à la base de données en tant qu'**Administrateur des bourses**.
- **maroc\_connect.properties** : Ce fichier contient les informations de connexion de l'utilisateur Administrateur des bourses, il comprendra l'adresse et le nom d'utilisateur et le mot de passe, pour permet l'application de se connecter à la base de données en tant que **Modérateur des bourses des étudiants marocains**.
- **etranger\_connect.properties** : Ce fichier contient les informations de connexion de l'utilisateur Administrateur des bourses, il comprendra l'adresse et le nom d'utilisateur et le mot de passe, pour permet l'application de se connecter à la base de données en tant que **Modérateur des bourses des étudiants étrangers**.



### Interface console

Les fichiers avant la compilation du fichier **MiniProjet\_sqlj.sqlj** :







































sqlj MiniProjet\_sqlj.sqlj

```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\BDRuser\eclipse-workspace\miniProjetBDR_sqlj\src\miniProjetBDR_sqlj>sqlj MiniProjet_sqlj.sqlj
C:\Users\BDRuser\eclipse-workspace\miniProjetBDR_sqlj\src\miniProjetBDR_sqlj>
```

Les fichiers g n r s apr s la compilation du fichier **MiniProjet\_sqlj.sqlj** :

 admin_connect.properties	 MiniProjet_sqlj\$3iter_info_person.class
 etranger_connect.properties	 MiniProjet_sqlj\$3iter_info_univ.class
 maroc_connect.properties	 MiniProjet_sqlj.class
 MiniProjet_sqlj\$1iter_a_aff_brs.class	 MiniProjet_sqlj
 MiniProjet_sqlj\$1iter_a_aff_etud.class	 MiniProjet_sqlj.sqlj
 MiniProjet_sqlj\$1iter_a_aff_univ.class	
 MiniProjet_sqlj\$1iter_a_aj_t_etud_brs_1.class	
 MiniProjet_sqlj\$1iter_a_aj_t_etud_brs_2.class	
 MiniProjet_sqlj\$1iter_a_aj_t_etud_univ_1.class	
 MiniProjet_sqlj\$1iter_a_aj_t_etud_univ_2.class	
 MiniProjet_sqlj\$1iter_a_mod_etude.class	
 MiniProjet_sqlj\$1iter_a_mod_univ.class	
 MiniProjet_sqlj\$1iter_a_sup_etud.class	
 MiniProjet_sqlj\$1iter_a_sup_univ.class	
 MiniProjet_sqlj\$1iter_e_aff_brs.class	
 MiniProjet_sqlj\$1iter_e_aff_etud.class	
 MiniProjet_sqlj\$1iter_e_aff_etud_univ.class	
 MiniProjet_sqlj\$1iter_e_aff_univ.class	
 MiniProjet_sqlj\$1iter_info_bourse.class	
 MiniProjet_sqlj\$1iter_info_person.class	
 MiniProjet_sqlj\$1iter_info_univ.class	
 MiniProjet_sqlj\$1iter_m_aff_brs.class	
 MiniProjet_sqlj\$1iter_m_aff_etud.class	
 MiniProjet_sqlj\$1iter_m_aff_etud_univ.class	
 MiniProjet_sqlj\$1iter_m_aff_univ.class	
 MiniProjet_sqlj\$2iter_info_bourse.class	
 MiniProjet_sqlj\$2iter_info_person.class	
 MiniProjet_sqlj\$2iter_info_univ.class	
 MiniProjet_sqlj\$3iter_info_bourse.class	

Le fichier MiniProjet\_sqlj.java a 3046 lignes :

```

3016     } finally { __sJT_ec.oracleCloseQuery(); }
3017 }
3018
3019
3020 // *****
3021
3022 /*@lineinfo:user-code/*@lineinfo:818^99*/
3023     System.out.println("");
3024     System.out.println("-----");
3025     System.out.println("|                LES BOURSES DES
3026     System.out.println("-----");
3027     while (bourse_e_aff_brs.next()) {
3028         System.out.println("| - ID_BOURSE: "+bourse
3029         System.out.println("|      ID_ETUDIANT: " +l
3030         System.out.println("|      NOM: " +bourse_e_
3031         System.out.println("|      MONTANT : "+bour
3032         System.out.println("| *****");
3033     }
3034     bourse_e_aff_brs.close();
3035     break;
3036
3037     case 0:
3038         continuer = false;
3039         break;
3040
3041     default:
3042         System.out.println("\n          *!*!*!*! Choix ir
3043     } //fin de switch
3044     } //fin de while
3045     } //fin de etranger_bourse_menu()
3046 }/*@lineinfo:generated-code*/

```

Le menu de l'Administrateur de bourse :

```

C:\Users\BDRuser\eclipse-workspace\miniProjetBDR_sqlj\src\miniProjetBDR_sqlj>java MiniProjet_sqlj

--- SQLJ ---
Choisir une option :
1. Connectez en tant que *Admin Bourse*
2. Bourses Etudiants Marocains
3. Bourses Etudiants Etrangers
4. Quitter
choix: 1

--- SQLJ ---
Connection avec Admin Bourse --> OK

*** ADMINISTRATEUR DE LA BOURSE AU MAROC ***

+-----+-----+
| Etudiants | Universites |
+-----+-----+
| 1. Afficher Les Etudiants | 8. Afficher Les Universites |
| 2. *Afficher Tous Les Infos D'etudiant* | 9. Afficher Les Bourses |
+-----+-----+
| 3. Ajouter un Etudiant | 10. Ajouter une Universite |
| 4. Supprimer un Etudiant | 11. Supprimer une Universite |
| 5. Modifier un Etudiant | 12. Modifier une Universite |
+-----+-----+
| 6. Ajouter un Etudiant a Universite | |
| 7. Ajouter un Etudiant a Bourse | |
+-----+-----+
| 0. Quitter | |
+-----+-----+

Choisir une option ==> 0

```

## Le menu de le Modérateur bourses des étudiants marocains :

```
Choisir une option ==> 0

--- SQLJ ---
Choisir une option :
1. Connectez en tant que *Admin Bourse*
2. Bourses Etudiants Marocains
3. Bourses Etudiants Etrangers
4. Quitter
choix: 2

--- SQLJ ---
Connection avec Bourses Etudiants Marocains --> OK

*** LA BOURSE DES ETUDIANTS -MAROCAINES- ***
-----
Etudiants                                Universites
-----
1. Afficher Les Etudiants Marocaines      4. Afficher Les Universites
2. Afficher Etudiants Marocaines / Universites  5. Afficher Les Bourses
3. *Afficher Tous Les Infos D'etudiant*
0. Quitter
-----
Choisir une option ==> 0
```

## Le menu de le Modérateur bourses des étudiants étrangers :

```
Choisir une option ==> 0

--- SQLJ ---
Choisir une option :
1. Connectez en tant que *Admin Bourse*
2. Bourses Etudiants Marocains
3. Bourses Etudiants Etrangers
4. Quitter
choix: 3

--- SQLJ ---
Connection avec Bourses Etudiants Etrangers --> OK

*** LA BOURSE DES ETUDIANTS -ETRANGES- ***
-----
Etudiants                                Universites
-----
1. Afficher Les Etudiants Etranges        4. Afficher Les Universites
2. Afficher Etudiants Etranges / Universites  5. Afficher Les Bourses
3. *Afficher Tous Les Infos D'etudiant*
0. Quitter
-----
Choisir une option ==> 0
```

## Le code complète

**admin\_connect.properties**

```
# URL
sqlj.url=jdbc:oracle:thin:@localhost:1521:xe

# USER
sqlj.user=admin_bourse

# PASSWORD
sqlj.password=aissam
```

### maroc\_connect.properties

```
# URL
sqlj.url=jdbc:oracle:thin:@localhost:1521:xe

# USER
sqlj.user=moderateur_bourse_etranger

# PASSWORD
sqlj.password=aissam
```

### etranger\_connect.properties

```
# URL
sqlj.url=jdbc:oracle:thin:@localhost:1521:xe

# USER
sqlj.user=moderateur_bourse_marocaine

# PASSWORD
sqlj.password=aissam
```

## MiniProjet\_sqlj.sqlj

```
import java.sql.SQLException ;
import oracle.sqlj.runtime.Oracle;
import java.io.*;

class MiniProjet_sqlj {

    public static void main(String[] args) throws IOException, SQLException {
```

```

int n=0, m=0, nbr_ligne;
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
int item_menu = 1;

while (true) {
    System.out.println("\n\n   --- SQLJ --- ");
    System.out.println(" Choisir une option : ");
    System.out.println(" 1. Connectez en tant que *Admin Bourse* ");
    System.out.println(" 2. Bourses Etudiants Marocains ");
    System.out.println(" 3. Bourses Etudiants Etrangers ");
    System.out.println(" 4. Quitter ");

    System.out.print("\nchoix: ");
    item_menu = Integer.parseInt(reader.readLine());

    System.out.println("");

    switch (item_menu) {
        case 1 :
            System.out.println("\n   --- SQLJ --- ");
            Oracle.connect(MiniProjet_sqlj.class, "admin_connect.properties");
            System.out.println(" Connection avec Admin Bourse --> OK");
            System.out.println(" -----");
            admin_bourse_menu();
            break;
        case 2 :
            System.out.println("\n   --- SQLJ --- ");
            Oracle.connect(MiniProjet_sqlj.class, "maroc_connect.properties");
            System.out.println(" Connection avec Bourses Etudiants Marocains -->
OK");
            System.out.println(" -----");
            maroc_bourse_menu();

            break;
        case 3 :
            System.out.println("\n   --- SQLJ --- ");
            Oracle.connect(MiniProjet_sqlj.class, "etranger_connect.properties");
            System.out.println(" Connection avec Bourses Etudiants Etrangers -->
OK");
            System.out.println(" -----");
            etranger_bourse_menu();

            break;
        case 4:
            System.out.println("Quitter SQLJ... OK");
            System.exit(0);

        default:
            System.out.println("Choix invalide. Sélectionner une option valide (1, 2,
3 ou 4).");
    }
}

//----- admin_bourse_menu ()
=====
=====

```

```

public static void admin_bourse_menu() throws IOException, SQLException {
    int n=0, m=0, nbr_ligne;

    BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

    int item_menu = 1;
    //while(1<=item_menu && item_menu<=5) {
    boolean continuer = true;
    while (continuer) {

System.out.println("_____");
        System.out.println("|                                     |");
        System.out.println("|                                     |");
        *** System.out.println("|                                     |");
        System.out.println("-----|");
        System.out.println("| Etudiants |");
Universites System.out.println("|");
        System.out.println("-----|");
        System.out.println("| 1. Afficher Les Etudiants | 8. Afficher");
Les Universites System.out.println("|");
        System.out.println("| 2. *Afficher Tous Les Infos D'etudiant* | 9.");
Afficher Les Bourses System.out.println("|");
        System.out.println("-----|");
        System.out.println("| 3. Ajouter un Etudiant | 10. Ajouter une");
Universite System.out.println("|");
        System.out.println("| 4. Supprimer un Etudiant | 11. Supprimer");
une Universite System.out.println("|");
        System.out.println("| 5. Modifier un Etudiant | 12. Modifier");
une Universite System.out.println("-----|");
        System.out.println("-----|");
        System.out.println("| 6. Ajouter un Etudiant a Universite |");
|");
        System.out.println("| 7. Ajouter un Etudiant a Bourse |");
|");
        System.out.println("-----|");
        System.out.println("-----|");
        System.out.println("| 0. Quitter |");
|");

System.out.println("_____");
        System.out.println("_____");

        System.out.print("\n Choisir une option ==> ");
        item_menu = Integer.parseInt(reader.readLine());

        System.out.println("");

        switch (item_menu) {
            case 1:
                System.out.println("\n -- 1. Afficher Les Etudiants --\n");
                #sql iterator iter_a_aff_etud( int ID_ETUDIANT, String NOM, String PRENOM,
String NATIONALITE, int ETRANGER );
                iter_a_aff_etud etudiant_a_aff_etud;

```

```

        #sql etudiant_a_aff_etud = { SELECT ID_ETUDIANT, NOM, PRENOM, NATIONALITE,
ETRANGER FROM Etudiants };
        System.out.println("");
        System.out.println("-----");
        System.out.println("|          LES ETUDIANTS MAROCAINES ET ETRANGERS
|");
        System.out.println("-----");
        while (etudiant_a_aff_etud.next()) {
            System.out.println("| - ID_ETUDIANT: "+etudiant_a_aff_etud.ID_ETUDIANT()
);
            System.out.println("|      NOM: " +etudiant_a_aff_etud.NOM() );
            System.out.println("|      PRENOM : "+etudiant_a_aff_etud.PRENOM() );
            System.out.println("|      NATIONALITE :
"+etudiant_a_aff_etud.NATIONALITE() );
            System.out.println("|      ETRANGER: "+etudiant_a_aff_etud.ETRANGER() );
            System.out.println("*****");
        }
        etudiant_a_aff_etud.close();
        break;

    case 2:
        System.out.println("\n      -- 2. *Afficher Tous Les Infos D'un Etudiant* -
-\n");
        System.out.print("\n Saisir ID_ETUDIANT: ");
        int id_etud = Integer.parseInt(reader.readLine());
        //les infos Personel
        #sql iterator iter_info_person( int ID_Etudiant, String Nom, String
Prenom, String Nationalite, int Etranger );
        iter_info_person etudiant_info_person;
        #sql etudiant_info_person = { SELECT ID_Etudiant, Nom, Prenom,
Nationalite, Etranger FROM Etudiants WHERE ID_Etudiant = :id_etud };
        System.out.println("");
        System.out.println("-----");
        System.out.println("|          LES INFORMATION DE L'ETUDIANT: ");
        //System.out.print("|          LES INFORMATION DE L'ETUDIANT: "); #sql
:Nom; System.out.print(""); #sql :Prenom; System.out.println("");
        System.out.println("-----");
        while (etudiant_info_person.next()) {
            System.out.println("|      + Personal Information: ");
            System.out.println("|      - ID_ETUDIANT:
"+etudiant_info_person.ID_Etudiant() );
            System.out.println("|      - NOM: " +etudiant_info_person.Nom() );
            System.out.println("|      - PRENOM : "+etudiant_info_person.Prenom()
);
            System.out.println("|      - NATIONALITE :
"+etudiant_info_person.Nationalite() );
            System.out.println("|      - ETRANGER:
"+etudiant_info_person.Etranger() );
        }
        etudiant_info_person.close();

        //les infos de l universite
        #sql iterator iter_info_univ( int ID_Universite, String Nom, String
Localisation );

```



```

        iter_info_univ universite;
        #sql universite = { SELECT ID_Universite, Nom, Localisation FROM
Universites WHERE ID_Universite IN ( SELECT ID_Universite FROM
Etudiants_Universites WHERE ID_Etudiant = :id_etud ) };
        while (universite.next()) {
            System.out.println("          + Universite: ");
            System.out.println("          - ID_UNIVERSITE:
"+universite.ID_Universite() );
            System.out.println("          - NOM: " +universite.Nom() );
            System.out.println("          - LOCALISATION :
"+universite.Localisation() );
        }
        universite.close();

        // les infos de labourse
        #sql iterator iter_info_bourse( int ID_Bourse, String Nom, int Montant );
        iter_info_bourse bourse;
        #sql bourse = { SELECT ID_Bourse, Nom, Montant FROM Bourses WHERE
ID_Etudiant = :id_etud };
        while (bourse.next()) {
            System.out.println("          + Bourse: ");
            System.out.println("          - ID_BOURSE: "+bourse.ID_Bourse() );
            System.out.println("          - NOM: " +bourse.Nom() );
            System.out.println("          - MONTANT : "+bourse.Montant() );
            System.out.println("*****");
        }
        bourse.close();
        break;

    case 3:
        System.out.println("\n          -- 3. Ajouter un Etudiant --\n");

        String nom_etud = "";
        String prenom_etud = "";
        String nationalite_etud = "";
        int etranger_etud = 0;

        System.out.println("\n\n          - Ajouter un Etudiant - ");
        System.out.println("\n          Entrez les information de l'etudiant : ");
        System.out.print("\n          -Nom: ");
        nom_etud = reader.readLine();

        System.out.print("\n          -Prenom: ");
        prenom_etud = reader.readLine();

        System.out.print("\n          -Nationalite: ");
        nationalite_etud = reader.readLine();

        System.out.print("\n          -Est-ce un étudiant marocain? (oui / non): ");
        String str_maroc = reader.readLine();

        if(str_maroc.toLowerCase().equals("non")){
            etranger_etud = 1;
        }

        #sql{ call INSERT_etudiant(:nom_etud, :prenom_etud,
:nationalite_etud,:etranger_etud) };

```

```

        System.out.println("\n                ==> Ajouter Etudiant '"+nom_etud+"
"+prenom_etud+"' --> OK");

        break;
    case 4:
        System.out.println("\n                -- 4. Supprimer un Etudiant --\n");

        //Afficher la liste des etudiants
        #sql iterator iter_a_sup_etud( int ID_ETUDIANT, String NOM, String PRENOM,
String NATIONALITE, int ETRANGER );
        iter_a_sup_etud  etudiant_a_sup_etud;
        #sql etudiant_a_sup_etud = { SELECT ID_ETUDIANT, NOM, PRENOM, NATIONALITE,
ETRANGER FROM Etudiants };
        System.out.println("");
        System.out.println("-----");
        -----");
        System.out.println("|                LES ETUDIANTS MAROCAINES ET ETRANGERS
|");
        System.out.println("-----");
        -----");
        while (etudiant_a_sup_etud.next()) {
            System.out.println("| - ID_ETUDIANT: "+etudiant_a_sup_etud.ID_ETUDIANT() );
            System.out.println("|          NOM: " +etudiant_a_sup_etud.NOM() );
            System.out.println("|          PRENOM : "+etudiant_a_sup_etud.PRENOM() );
            System.out.println("|          NATIONALITE : "+etudiant_a_sup_etud.NATIONALITE()
);
            System.out.println("|          ETRANGER: "+etudiant_a_sup_etud.ETRANGER() );
            System.out.println("|
*****");
        }
        etudiant_a_sup_etud.close();

        //supprimer un etudiant

        System.out.print("\n\n        Entrez l'ID de l'Etudiant: ");
        id_etud = Integer.parseInt(reader.readLine());

        #sql{ DELETE FROM ETUDIANTS where ID_ETUDIANT = :id_etud };

        System.out.println("\n                ==> Supprimer Etudiant ID='"+id_etud+"' -->
OK");

        break;

    case 5:
        System.out.println("\n                -- 5. Modifier un Etudiant --\n");

        //Afficher la liste des etudiants
        #sql iterator iter_a_mod_etude( int ID_ETUDIANT, String NOM, String
PRENOM, String NATIONALITE, int ETRANGER );
        iter_a_mod_etude  etudiant_a_mod_etude;
        #sql etudiant_a_mod_etude = { SELECT ID_ETUDIANT, NOM, PRENOM,
NATIONALITE, ETRANGER FROM Etudiants };
        System.out.println("");
        System.out.println("-----");
        -----");
        System.out.println("|                LES ETUDIANTS MAROCAINES ET ETRANGERS
|");

```

```

        System.out.println("-----");
    -----");
        while (etudiant_a_mod_etude.next()) {
            System.out.println("| - ID_ETUDIANT: "+etudiant_a_mod_etude.ID_ETUDIANT() );
            System.out.println("|      NOM: " +etudiant_a_mod_etude.NOM() );
            System.out.println("|      PRENOM : "+etudiant_a_mod_etude.PRENOM() );
            System.out.println("|      NATIONALITE : "+etudiant_a_mod_etude.NATIONALITE()
        );
            System.out.println("|      ETRANGER: "+etudiant_a_mod_etude.ETRANGER() );
            System.out.println("*****");
        }
        etudiant_a_mod_etude.close();

        //modifier un etudiant
        System.out.print("\n      Entrez ID de l'etudiant qui sera modifié: ");
        int id_etud_mod = Integer.parseInt(reader.readLine());

        System.out.print("      -Nom de l'etudiant: ");
        String nom_etud_mod = reader.readLine();

        System.out.print("      -Prenom de l'etudiant: ");
        String prenom_etud_mod = reader.readLine();

        System.out.print("      -Nationalite de l'etudiant: ");
        String nationalite_etud_mod = reader.readLine();

        int etranger_etud_mod = 0;
        System.out.print("      -Est-ce un étudiant marocain? (oui / non): ");
        String str_maroc_mod = reader.readLine();
        if(str_maroc_mod.toLowerCase().equals("non")){
            etranger_etud_mod = 1;
        }
        #sql{ UPDATE ETUDIANTS SET NOM = :nom_etud_mod , PRENOM = :prenom_etud_mod
, NATIONALITE = :nationalite_etud_mod , ETRANGER = :etranger_etud_mod where
ID_ETUDIANT = :id_etud_mod };
        System.out.println("\n          ==> Modifier Etudiant ID='"+id_etud_mod+"' --
> OK");
        break;

    case 6:
        System.out.println("\n      -- 6. Ajouter un Etudiant a Universite --\n");

        //Afficher la liste des etudiants
        #sql iterator iter_a_ajt_etud_univ_1( int ID_ETUDIANT, String NOM, String
PRENOM, String NATIONALITE, int ETRANGER );
        iter_a_ajt_etud_univ_1 etudiant_a_ajt_etud_univ_1;
        #sql etudiant_a_ajt_etud_univ_1 = { SELECT ID_ETUDIANT, NOM, PRENOM,
NATIONALITE, ETRANGER FROM Etudiants };
        System.out.println("");
        System.out.println("-----");
    -----");
        System.out.println("|      LES ETUDIANTS MAROCAINES ET ETRANGERS
|");
        System.out.println("-----");
    -----");
        while (etudiant_a_ajt_etud_univ_1.next()) {
            System.out.println("| - ID_ETUDIANT:
"+etudiant_a_ajt_etud_univ_1.ID_ETUDIANT() );

```

```

        System.out.println("|      NOM: " +etudiant_a_ajt_etud_univ_1.NOM() );
        System.out.println("|      PRENOM : "+etudiant_a_ajt_etud_univ_1.PRENOM() );
        System.out.println("|      NATIONALITE :
"+etudiant_a_ajt_etud_univ_1.NATIONALITE() );
        System.out.println("|      ETRANGER: "+etudiant_a_ajt_etud_univ_1.ETRANGER()
);
        System.out.println("|
*****");
    }
    etudiant_a_ajt_etud_univ_1.close();

    //Afficher la liste des universities
    #sql iterator iter_a_ajt_etud_univ_2( int ID_UNIVERSITE, String NOM,
String LOCALISATION);
    iter_a_ajt_etud_univ_2  univ_a_ajt_etud_univ_2;
    #sql univ_a_ajt_etud_univ_2 = { SELECT ID_UNIVERSITE, NOM, LOCALISATION
FROM Universities };
    System.out.println("");
    System.out.println("-----");
    -----");
    System.out.println("|      LES UNIVERSITES      |");
    System.out.println("-----");
    -----");
    while (univ_a_ajt_etud_univ_2.next()) {
        System.out.println("| - ID_UNIVERSITE:
"+univ_a_ajt_etud_univ_2.ID_UNIVERSITE() );
        System.out.println("|      NOM: " +univ_a_ajt_etud_univ_2.NOM() );
        System.out.println("|      LOCALISATION :
"+univ_a_ajt_etud_univ_2.LOCALISATION() );
        System.out.println("|
*****");
    }
    univ_a_ajt_etud_univ_2.close();

    //ajouter etudiant a universite
    System.out.println("\n\n      - Ajouter un Etudiant a Universite - ");

    System.out.print("\n      -Entrez le ID de l'etudiant: ");
    int id_etud_aj_univ = Integer.parseInt(reader.readLine());

    System.out.print("\n      -Entrez le ID de l'universite: ");
    int id_univ_aj_univ = Integer.parseInt(reader.readLine());

    #sql{ call INSERT_etudiant_universite(:id_etud_aj_univ, :id_univ_aj_univ)
};

    System.out.println("\n      ==> Ajouter Etudiant de ID =
"+id_etud_aj_univ+" a l'Universite de ID = "+id_univ_aj_univ+" --> OK");
    break;

    case 7:
        System.out.println("\n      -- 7. Ajouter un Etudiant a Bourse --\n");

        //Afficher la liste des etudiants
        #sql iterator iter_a_ajt_etud_brs_1( int ID_ETUDIANT, String NOM, String
PRENOM, String NATIONALITE, int ETRANGER );
        iter_a_ajt_etud_brs_1 etudiant_a_ajt_etud_brs_1;
        #sql etudiant_a_ajt_etud_brs_1 = { SELECT ID_ETUDIANT, NOM, PRENOM,
NATIONALITE, ETRANGER FROM Etudiants };

```

```

        System.out.println("");
        System.out.println("-----");
    -----");
        System.out.println("|          LES ETUDIANTS MAROCAINES ET ETRANGERS
|");
        System.out.println("-----");
    -----");
        while (etudiant_a_aj_t_etud_brs_1.next()) {
            System.out.println("| - ID_ETUDIANT:
"+etudiant_a_aj_t_etud_brs_1.ID_ETUDIANT() );
            System.out.println("|      NOM: " +etudiant_a_aj_t_etud_brs_1.NOM() );
            System.out.println("|      PRENOM : "+etudiant_a_aj_t_etud_brs_1.PRENOM() );
            System.out.println("|      NATIONALITE :
"+etudiant_a_aj_t_etud_brs_1.NATIONALITE() );
            System.out.println("|      ETRANGER: "+etudiant_a_aj_t_etud_brs_1.ETRANGER()
);
            System.out.println("|
*****");
        }
        etudiant_a_aj_t_etud_brs_1.close();

        //Afficher la liste des bourses
        #sql iterator iter_a_aj_t_etud_brs_2( int ID_Bourse, int ID_Etudiant,
String Nom, int Montant);
        iter_a_aj_t_etud_brs_2 bourse_a_aj_t_etud_brs_2;
        #sql bourse_a_aj_t_etud_brs_2 = { SELECT ID_Bourse, ID_Etudiant, Nom,
Montant FROM BOURSES };
        System.out.println("");
        System.out.println("-----");
    -----");
        System.out.println("|          LES BOURSES          |");
        System.out.println("-----");
    -----");
        while (bourse_a_aj_t_etud_brs_2.next()) {
            System.out.println("| - ID_BOURSE: "+bourse_a_aj_t_etud_brs_2.ID_Bourse() );
            System.out.println("|      ID_ETUDIANT: "
+bourse_a_aj_t_etud_brs_2.ID_Etudiant() );
            System.out.println("|      NOM: " +bourse_a_aj_t_etud_brs_2.Nom() );
            System.out.println("|      MONTANT : "+bourse_a_aj_t_etud_brs_2.Montant() );
            System.out.println("|
*****");
        }
        bourse_a_aj_t_etud_brs_2.close();

        //ajouter etudiant a bourse
        System.out.println("\n\n    - Ajouter un Etudiant a Bourse - ");

        System.out.println("\n    Entrez les information: ");

        System.out.print("\n    -ID de l'etudiant: ");
        int id_etud_aj_brs = Integer.parseInt(reader.readLine());

        System.out.print("\n    -Nom de la Bourse: ");
        String nom_brs_aj_brs = reader.readLine();

        System.out.print("\n    -Montant de la Bourse: ");
        int montant_brs_aj_brs = Integer.parseInt(reader.readLine());

```

```

        #sql{ call INSERT_bourse(:id_etud_aj_brs, :nom_brs_aj_brs,
:montant_brs_aj_brs) };

        System.out.println("\n          ==> Ajouter Etudiant de
ID='"+id_etud_aj_brs+"' a la Bourse '"+nom_brs_aj_brs+"' --> OK");
        break;

    case 8:
        System.out.println("\n          -- 8. Afficher Les Universites --\n");
        #sql iterator iter_a_aff_univ( int ID_UNIVERSITE, String NOM, String
LOCALISATION);
        iter_a_aff_univ universite_a_aff_univ;
        #sql universite_a_aff_univ = { SELECT ID_UNIVERSITE, NOM, LOCALISATION
FROM Universites };
        System.out.println("");
        System.out.println("-----");
        System.out.println("|          LES UNIVERSITES          |");
        System.out.println("-----");
        while (universite_a_aff_univ.next()) {
            System.out.println("| - ID_UNIVERSITE:
"+universite_a_aff_univ.ID_UNIVERSITE() );
            System.out.println("|          NOM: " +universite_a_aff_univ.NOM() );
            System.out.println("|          LOCALISATION :
"+universite_a_aff_univ.LOCALISATION() );
            System.out.println("|
*****");
        }
        universite_a_aff_univ.close();
        break;

    case 9:
        System.out.println("\n          -- 9. Afficher Les Bourses --\n");
        #sql iterator iter_a_aff_brs( int ID_Bourse, int ID_Etudiant, String Nom,
int Montant);
        iter_a_aff_brs bourse_a_aff_brs;
        #sql bourse_a_aff_brs = { SELECT ID_Bourse, ID_Etudiant, Nom, Montant FROM
BOURSES };
        System.out.println("");
        System.out.println("-----");
        System.out.println("|          LES BOURSES          |");
        System.out.println("-----");
        while (bourse_a_aff_brs.next()) {
            System.out.println("| - ID_BOURSE: "+bourse_a_aff_brs.ID_Bourse() );
            System.out.println("|          ID_ETUDIANT: " +bourse_a_aff_brs.ID_Etudiant() );
            System.out.println("|          NOM: " +bourse_a_aff_brs.Nom() );
            System.out.println("|          MONTANT : "+bourse_a_aff_brs.Montant() );
            System.out.println("|
*****");
        }
        bourse_a_aff_brs.close();
        break;

    case 10:
        System.out.println("\n          -- 10. Ajouter une Universite --\n");

```

```

System.out.println("\n    Entrez les information de l'Universite : ");

System.out.print("\n    -Nom: ");
String nom_univ = reader.readLine();

System.out.print("\n    -Localisation: ");
String loc_univ = reader.readLine();

#sql{ call INSERT_universite(:nom_univ, :loc_univ) };

System.out.println("\n                ==> Ajouter Universite "+nom_univ+" --> OK");
break;

case 11:
    System.out.println("\n                -- 11. Supprimer une Universite --\n");

    //Afficher la liste des Universites
    #sql iterator iter_a_sup_univ( int ID_UNIVERSITE, String NOM, String
LOCALISATION);
    iter_a_sup_univ universite_a_sup_univ;
    #sql universite_a_sup_univ = { SELECT ID_UNIVERSITE, NOM, LOCALISATION
FROM Universites };
    System.out.println("");
    System.out.println("-----");
    System.out.println("
                                LES UNIVERSITES
                                ");
    System.out.println("-----");
    while (universite_a_sup_univ.next()) {
        System.out.println("
- ID_UNIVERSITE:
"+universite_a_sup_univ.ID_UNIVERSITE() );
        System.out.println("
    NOM: " +universite_a_sup_univ.NOM() );
        System.out.println("
    LOCALISATION :
"+universite_a_sup_univ.LOCALISATION() );
        System.out.println("
*****");
    }
    universite_a_sup_univ.close();

    //supprimer une universite

    System.out.print("\n    Entrez l'ID de l'Universite: ");
    int id_univ = Integer.parseInt(reader.readLine());

    #sql{ DELETE FROM UNIVERSITES where ID_UNIVERSITE = :id_univ };

    System.out.println("\n                ==> Supprimer Universite ID='"+id_univ+"' -->
OK");
    break;

case 12:
    System.out.println("\n                -- 12. Modifier une Universite --\n");

    //Afficher la liste des Universites
    #sql iterator iter_a_mod_univ( int ID_UNIVERSITE, String NOM, String
LOCALISATION);
    iter_a_mod_univ universite_a_mod_univ;

```

```

        #sql universite_a_mod_univ = { SELECT ID_UNIVERSITE, NOM, LOCALISATION
FROM Universites };
        System.out.println("");
        System.out.println("-----");
    -----");
        System.out.println("|          LES UNIVERSITES          |");
        System.out.println("-----");
    -----");
        while (universite_a_mod_univ.next()) {
            System.out.println("| - ID_UNIVERSITE:
"+universite_a_mod_univ.ID_UNIVERSITE() );
            System.out.println("|      NOM: " +universite_a_mod_univ.NOM() );
            System.out.println("|      LOCALISATION :
"+universite_a_mod_univ.LOCALISATION() );
            System.out.println("|
*****");
        }
        universite_a_mod_univ.close();

        //modifier une universite
        System.out.print("\n    Entrez ID de l'universite qui sera modifié: ");
        int id_univ_mod_univ = Integer.parseInt(reader.readLine());

        System.out.print("    -Nom de l'Universite: ");
        String nom_univ_mod_univ = reader.readLine();

        System.out.print("    -Localisation de l'Universite: ");
        String loc_univ_mod_univ = reader.readLine();

        #sql{ UPDATE UNIVERSITES SET NOM = :nom_univ_mod_univ , LOCALISATION =
:loc_univ_mod_univ where ID_UNIVERSITE = :id_univ_mod_univ };

        System.out.println("\n          ==> Modifier Universite
ID='"+id_univ_mod_univ+"' --> OK");
        break;

    case 0:
        continuer = false;
        break;

    default:
        System.out.println("\n          *!*!*!* Choix invalide *!*!*!*");
    } //fin de switch
} //fin de while
} //fin de admin_bourse_menu()

//----- maroc_bourse_menu ()
=====
=====
public static void maroc_bourse_menu() throws IOException, SQLException {
    int n=0, m=0, nbr_ligne;

    BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

    int item_menu = 1;
    //while(1<=item_menu && item_menu<=5) {
    boolean continuer = true;

```



```

        while (continuer) {

System.out.println("_____");
        System.out.println("|
|");
        System.out.println("|          *** LA BOURSE DES ETUDIANTS -MAROCAINES-
***          |");
        System.out.println("|-----|-----|-----|");
        System.out.println("|          Etudiants          |
Universites          |");
        System.out.println("|-----|-----|-----|");
        System.out.println("|  1. Afficher Les Etudiants Marocaines      |  4.
Afficher Les Universites          |");
        System.out.println("|  2. Afficher Etudiants Marocaines / Universites |  5.
Afficher Les Bourses          |");
        System.out.println("|-----|-----|-----|");
        System.out.println("|  3. *Afficher Tous Les Infos D'etudiant*      |
|");
        System.out.println("|-----|-----|-----|");
        System.out.println("|  0. Quitter          |
|");

System.out.println("_____|_____");

        System.out.print("\n Choisir une option ==> ");
        item_menu = Integer.parseInt(reader.readLine());

        System.out.println("");

        switch (item_menu) {
        case 1:
            System.out.println("\n      -- 1. Afficher Les Etudiants Marocaines --\n");

            #sql iterator iter_m_aff_etud( int ID_ETUDIANT, String NOM, String PRENOM,
String NATIONALITE, int ETRANGER );
            iter_m_aff_etud  etudiant_m_aff_etud;
            #sql etudiant_m_aff_etud = { SELECT ID_ETUDIANT, NOM, PRENOM, NATIONALITE,
ETRANGER FROM Etudiants_Maroc };
            System.out.println("");
            System.out.println("-----");
            System.out.println("|          LES ETUDIANTS *MAROCAINES*          |");
            System.out.println("-----");
            while (etudiant_m_aff_etud.next()) {
                System.out.println("|  - ID_ETUDIANT: "+etudiant_m_aff_etud.ID_ETUDIANT() );
                System.out.println("|          NOM: " +etudiant_m_aff_etud.NOM() );
                System.out.println("|          PRENOM : "+etudiant_m_aff_etud.PRENOM() );
                System.out.println("|          NATIONALITE : "+etudiant_m_aff_etud.NATIONALITE()
);
                System.out.println("|          ETRANGER: "+etudiant_m_aff_etud.ETRANGER() );
                System.out.println("*****");

```

```

    }
    etudiant_m_aff_etud.close();
    break;

    case 2:
        System.out.println("\n      -- 2. Afficher Etudiants Marocaines /
Universites --\n");

        #sql iterator iter_m_aff_etud_univ( int ID_ETUDIANT, int ID_UNIVERSITE );
        iter_m_aff_etud_univ etu_univ_m_aff_etud_univ;
        #sql etu_univ_m_aff_etud_univ = { SELECT ID_ETUDIANT, ID_UNIVERSITE FROM
ETUD_UNIV_MAROC };

        System.out.println("");
        System.out.println("-----");
        System.out.println("|          LES ETUDIANTS_UNIVERSITES *MAROCAINES*
|");
        System.out.println("-----");
        while (etu_univ_m_aff_etud_univ.next()) {
            System.out.println("| - ID_ETUDIANT:
"+etu_univ_m_aff_etud_univ.ID_ETUDIANT() );
            System.out.println("| - ID_UNIVERSITE:
"+etu_univ_m_aff_etud_univ.ID_UNIVERSITE() );
            System.out.println("|
*****");
        }
        etu_univ_m_aff_etud_univ.close();
        break;

    case 3:
        System.out.println("\n      -- 3. *Afficher Tous Les Infos D'un Etudiant* -
-\n");
        System.out.print("\n Saisir ID_ETUDIANT: ");
        int id_etud = Integer.parseInt(reader.readLine());
        //les infos Personel
        #sql iterator iter_info_person( int ID_Etudiant, String Nom, String
Prenom, String Nationalite, int Etranger );
        iter_info_person etudiant_info_person;
        #sql etudiant_info_person = { SELECT ID_Etudiant, Nom, Prenom,
Nationalite, Etranger FROM ETUDIANTS_MAROC WHERE ID_Etudiant = :id_etud };
        System.out.println("");
        System.out.println("-----");
        System.out.println("|          MAROC : LES INFORMATION DE L'ETUDIANT: ");
        //System.out.print("|          MAROC : LES INFORMATION DE L'ETUDIANT: ");
        #sql :Nom; System.out.print(""); #sql :Prenom; System.out.println("");
        System.out.println("-----");
        while (etudiant_info_person.next()) {
            System.out.println("|          + Personal Information: ");
            System.out.println("|          - ID_ETUDIANT:
"+etudiant_info_person.ID_Etudiant() );
            System.out.println("|          - NOM: " +etudiant_info_person.Nom() );
            System.out.println("|          - PRENOM : " +etudiant_info_person.Prenom() );
            System.out.println("|          - NATIONALITE :
"+etudiant_info_person.Nationalite() );

```

```

        System.out.println("|          - ETRANGER: "+etudiant_info_person.Etranger()
    );
    }
    etudiant_info_person.close();

    //les infos de l universite
    #sql iterator iter_info_univ( int ID_Universite, String Nom, String
Localisation );
    iter_info_univ universite_info_univ;
    #sql universite_info_univ = { SELECT ID_Universite, Nom, Localisation FROM
UNIVERSITES_MAROC WHERE ID_Universite IN ( SELECT ID_Universite FROM
ETUD_UNIV_MAROC WHERE ID_Etudiant = :id_etud ) };
    while (universite_info_univ.next()) {
        System.out.println("|          + Universite: ");
        System.out.println("|          - ID_UNIVERSITE:
"+universite_info_univ.ID_Universite() );
        System.out.println("|          - NOM: " +universite_info_univ.Nom() );
        System.out.println("|          - LOCALISATION :
"+universite_info_univ.Localisation() );
    }
    universite_info_univ.close();

    // les infos de labourse
    #sql iterator iter_info_bourse( int ID_Bourse, String Nom, int Montant );
    iter_info_bourse bourse_info_bourse;
    #sql bourse_info_bourse = { SELECT ID_Bourse, Nom, Montant FROM
BOURSES_MAROC WHERE ID_Etudiant = :id_etud };
    while (bourse_info_bourse.next()) {
        System.out.println("|          + Bourse: ");
        System.out.println("|          - ID_BOURSE: "+bourse_info_bourse.ID_Bourse()
    );
        System.out.println("|          - NOM: " +bourse_info_bourse.Nom() );
        System.out.println("|          - MONTANT : "+bourse_info_bourse.Montant() );
        System.out.println("|
*****");
    }
    bourse_info_bourse.close();
    break;

    case 4:
        System.out.println("\n      -- 4. Afficher Les Universites --\n");
        #sql iterator iter_m_aff_univ( int ID_UNIVERSITE, String NOM, String
LOCALISATION);
        iter_m_aff_univ univ_m_aff_univ;
        #sql univ_m_aff_univ = { SELECT ID_UNIVERSITE, NOM, LOCALISATION FROM
UNIVERSITES_MAROC };
        System.out.println("");
        System.out.println("-----");
        System.out.println("-----");
        System.out.println("|          LES UNIVERSITES *MAROCAINES*          |");
        System.out.println("-----");
        System.out.println("-----");
        while (univ_m_aff_univ.next()) {
            System.out.println("| - ID_UNIVERSITE: "+univ_m_aff_univ.ID_UNIVERSITE() );
            System.out.println("|          NOM: " +univ_m_aff_univ.NOM() );
            System.out.println("|          LOCALISATION : "+univ_m_aff_univ.LOCALISATION() );
            System.out.println("|
*****");
        }
    }
}

```

```

    }
    univ_m_aff_univ.close();
    break;

    case 5:
        System.out.println("\n      -- 5. Afficher Les Bourses --\n");
        #sql iterator iter_m_aff_brs( int ID_Bourse, int ID_Etudiant, String Nom,
int Montant);
        iter_m_aff_brs bourse;
        #sql bourse = { SELECT ID_Bourse, ID_Etudiant, Nom, Montant FROM
BOURSES_MAROC };
        System.out.println("");
        System.out.println("-----");
        System.out.println("|          LES BOURSES DES ETUDIANTS *MAROCAINES*
|");
        System.out.println("-----");
        while (bourse.next()) {
            System.out.println("| - ID_BOURSE: "+bourse.ID_Bourse() );
            System.out.println("|      ID_ETUDIANT: " +bourse.ID_Etudiant() );
            System.out.println("|      NOM: " +bourse.Nom() );
            System.out.println("|      MONTANT : "+bourse.Montant() );
            System.out.println("*****");
        }
        bourse.close();
        break;

    case 0:
        continuer = false;
        break;

    default:
        System.out.println("\n      *!*!*!* Choix invalide *!*!*!*");
    } //fin de switch
} //fin de while
} //fin de maroc_bourse_menu()

//----- etranger_bourse_menu ()
=====
public static void etranger_bourse_menu() throws IOException, SQLException {
    int n=0, m=0, nbr_ligne;

    BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

    int item_menu = 1;
    //while(1<=item_menu && item_menu<=5) {
    boolean continuer = true;
    while (continuer) {

System.out.println("_____");
        System.out.println("|
|");
        System.out.println("|          *** LA BOURSE DES ETUDIANTS -
ETRANGES- ***      |");

```

```

        System.out.println("|-----|-----|");
        System.out.println("|          Etudiants          |");
Universites      |");
        System.out.println("|-----|-----|");
        System.out.println("| 1. Afficher Les Etudiants Etranges      | 4.");
Afficher Les Universites |");
        System.out.println("| 2. Afficher Etudiants Etranges / Universites | 5.");
Afficher Les Bourses      |");
        System.out.println("|-----|-----|");
        System.out.println("| 3. *Afficher Tous Les Infos D'etudiant*      |");
|");
        System.out.println("|-----|-----|");
        System.out.println("| 0. Quitter          |");
|");

System.out.println("|_____|_____");
|_____");

System.out.print("\n Choisir une option ==> ");
item_menu = Integer.parseInt(reader.readLine());

System.out.println("");

switch (item_menu) {
case 1:
    System.out.println("\n      -- 1. Afficher Les Etudiants Etranges --\n");

    #sql iterator iter_e_aff_etud( int ID_ETUDIANT, String NOM, String PRENOM,
String NATIONALITE, int ETRANGER );
    iter_e_aff_etud  etudiant_e_aff_etud;
    #sql etudiant_e_aff_etud = { SELECT ID_ETUDIANT, NOM, PRENOM, NATIONALITE,
ETRANGER FROM ETUDIANTS_ETRANGERS };
    System.out.println("");
    System.out.println("-----");
    System.out.println("|          LES ETUDIANTS *ETRANGERS*          |");
    System.out.println("-----");
    while (etudiant_e_aff_etud.next()) {
        System.out.println("| - ID_ETUDIANT: "+etudiant_e_aff_etud.ID_ETUDIANT() );
        System.out.println("|      NOM: " +etudiant_e_aff_etud.NOM() );
        System.out.println("|      PRENOM : "+etudiant_e_aff_etud.PRENOM() );
        System.out.println("|      NATIONALITE : "+etudiant_e_aff_etud.NATIONALITE()
);
        System.out.println("|      ETRANGER: "+etudiant_e_aff_etud.ETRANGER() );
        System.out.println("*****");
    }
    etudiant_e_aff_etud.close();
    break;

case 2:
    System.out.println("\n      -- 2. Afficher Etudiants Etranges / Universites
--\n");

```

```

        #sql iterator iter_e_aff_etud_univ( int ID_ETUDIANT, int ID_UNIVERSITE );
        iter_e_aff_etud_univ etu_univ_e_aff_etud_univ;
        #sql etu_univ_e_aff_etud_univ = { SELECT ID_ETUDIANT, ID_UNIVERSITE FROM
ETUD_UNIV_ETRANGERS };

        System.out.println("");
        System.out.println("-----");
    -----");
        System.out.println("|          LES ETUDIANTS_UNIVERSITES *ETRANGERS*
|");
        System.out.println("-----");
    -----");
        while (etu_univ_e_aff_etud_univ.next()) {
            System.out.println("| - ID_ETUDIANT:
"+etu_univ_e_aff_etud_univ.ID_ETUDIANT() );
            System.out.println("| - ID_UNIVERSITE:
"+etu_univ_e_aff_etud_univ.ID_UNIVERSITE() );
            System.out.println("|
*****");
        }
        etu_univ_e_aff_etud_univ.close();
        break;

    case 3:
        System.out.println("\n      -- 3. *Afficher Tous Les Infos D'un Etudiant* -
-\n");
        System.out.print("\n Saisir ID_ETUDIANT: ");
        int id_etud = Integer.parseInt(reader.readLine());
        //les infos Personel
        #sql iterator iter_info_person( int ID_Etudiant, String Nom, String
Prenom, String Nationalite, int Etranger );
        iter_info_person etudiant_info_person;
        #sql etudiant_info_person = { SELECT ID_Etudiant, Nom, Prenom,
Nationalite, Etranger FROM ETUDIANTS_ETRANGERS WHERE ID_Etudiant = :id_etud };
        System.out.println("");
        System.out.println("-----");
    -----");
        System.out.println("|          ETRANGER : LES INFORMATION DE L'ETUDIANT:
");
        //System.out.print("|          ETRANGER : LES INFORMATION DE L'ETUDIANT:
"); #sql :Nom; System.out.print(""); #sql :Prenom; System.out.println("");
        System.out.println("-----");
    -----");
        while (etudiant_info_person.next()) {
            System.out.println("|          + Personal Information: ");
            System.out.println("|          - ID_ETUDIANT:
"+etudiant_info_person.ID_Etudiant() );
            System.out.println("|          - NOM: " +etudiant_info_person.Nom() );
            System.out.println("|          - PRENOM : " +etudiant_info_person.Prenom() );
            System.out.println("|          - NATIONALITE :
"+etudiant_info_person.Nationalite() );
            System.out.println("|          - ETRANGER: " +etudiant_info_person.Etranger()
);
        }
        etudiant_info_person.close();

        //les infos de l universite
        #sql iterator iter_info_univ( int ID_Universite, String Nom, String
Localisation );

```

```

        iter_info_univ universite_info_univ;
        #sql universite_info_univ = { SELECT ID_Universite, Nom, Localisation FROM
UNIVERSITES_ETRANGERS WHERE ID_Universite IN ( SELECT ID_Universite FROM
ETUD_UNIV_ETRANGERS WHERE ID_Etudiant = :id_etud ) };
        while (universite_info_univ.next()) {
            System.out.println("|          + Universite: ");
            System.out.println("|          - ID_UNIVERSITE:
"+universite_info_univ.ID_Universite() );
            System.out.println("|          - NOM: " +universite_info_univ.Nom() );
            System.out.println("|          - LOCALISATION :
"+universite_info_univ.Localisation() );
        }
        universite_info_univ.close();

        // les infos de labourse
        #sql iterator iter_info_bourse( int ID_Bourse, String Nom, int Montant );
        iter_info_bourse bourse_info_bourse;
        #sql bourse_info_bourse = { SELECT ID_Bourse, Nom, Montant FROM
BOURSES_ETRANGERS WHERE ID_Etudiant = :id_etud };
        while (bourse_info_bourse.next()) {
            System.out.println("|          + Bourse: ");
            System.out.println("|          - ID_BOURSE: "+bourse_info_bourse.ID_Bourse()
);
            System.out.println("|          - NOM: " +bourse_info_bourse.Nom() );
            System.out.println("|          - MONTANT : "+bourse_info_bourse.Montant() );
            System.out.println("*****");
        }
        bourse_info_bourse.close();
        break;

    case 4:
        System.out.println("\n      -- 4. Afficher Les Universites --\n");
        #sql iterator iter_e_aff_univ( int ID_UNIVERSITE, String NOM, String
LOCALISATION);
        iter_e_aff_univ universite_e_aff_univ;
        #sql universite_e_aff_univ = { SELECT ID_UNIVERSITE, NOM, LOCALISATION
FROM UNIVERSITES_ETRANGERS };
        System.out.println("");
        System.out.println("-----");
        System.out.println("|          LES UNIVERSITES *ETRANGERS*          |");
        System.out.println("-----");
        while (universite_e_aff_univ.next()) {
            System.out.println("| - ID_UNIVERSITE:
"+universite_e_aff_univ.ID_UNIVERSITE() );
            System.out.println("|          NOM: " +universite_e_aff_univ.NOM() );
            System.out.println("|          LOCALISATION :
"+universite_e_aff_univ.LOCALISATION() );
            System.out.println("*****");
        }
        universite_e_aff_univ.close();
        break;

    case 5:
        System.out.println("\n      -- 5. Afficher Les Bourses --\n");

```

```

        #sql iterator iter_e_aff_brs( int ID_Bourse, int ID_Etudiant, String Nom,
int Montant);
        iter_e_aff_brs bourse_e_aff_brs;
        #sql bourse_e_aff_brs = { SELECT ID_Bourse, ID_Etudiant, Nom, Montant FROM
BOURSES_ETRANGERS };
        System.out.println("");
        System.out.println("-----");
    -----");
        System.out.println("|          LES BOURSES DES ETUDIANTS *ETRANGERS*
|");
        System.out.println("-----");
    -----");
        while (bourse_e_aff_brs.next()) {
            System.out.println("| - ID_BOURSE: "+bourse_e_aff_brs.ID_Bourse() );
            System.out.println("|      ID_ETUDIANT: " +bourse_e_aff_brs.ID_Etudiant() );
            System.out.println("|      NOM: " +bourse_e_aff_brs.Nom() );
            System.out.println("|      MONTANT : "+bourse_e_aff_brs.Montant() );
            System.out.println("|
*****");
        }
        bourse_e_aff_brs.close();
        break;

    case 0:
        continuer = false;
        break;

    default:
        System.out.println("\n          *!*!*!* Choix invalide *!*!*!*");
    } //fin de switch
} //fin de while
} //fin de etranger_bourse_menu()
}

```