INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

2017

Compte-rendu de TP : Bases de données distribuées

SGBD 2017 – TP 3

B3412 LUCAS POISSE - ZIGGY VERGNE
B3407 HUBERT HAMELIN - CYRIL POTTIEZ
B3414 JULIEN CHARLES-NICOLAS - HORIA BURCA

2017 – GROUPE 4 – B3412/B3407/B3414

COMPTE-RENDU DE TP : BASES DE DONNEES DISTRIBUEES

Table des matières

B3412 Lucas POISSE – Ziggy VERGNE
B3407 Hubert HAMELIN – Cyril POTTIEZ
B3414 Julien CHARLES-NICOLAS – Horia BURCA

I. Rappel rapide du contexte et des objectifs du TP

Ryori est une entreprise implantée dans plusieurs grandes régions distantes : les Amériques, l'Europe du Nord et du Sud. L'entreprise décide de fragmenter sa base de données en plusieurs morceaux afin de les répartir sur chacun de ses sites. L'objectif est de reproduire le comportement habituel de la base centralisée du point de vue de l'utilisateur et des applications, mais avec les données de la base réparties sur les trois sites.

Ainsi, notre objectif est d'effectuer cette répartition tout en maintenant la cohérence des données et les propriétés de la base centralisée. Ainsi, nous devons maintenir les contraintes de type « clé primaire » et « clé étrangère » sur l'ensemble des bases même si les informations dont nous avons besoin pour vérifier ces contraintes ne sont pas sur la même base de données que la nôtre. Nous devons donc effectuer des traitements spéciaux à l'insertion, la mise-à-jour et la suppression des tuples sur la base. De plus, nous devons veiller à attribuer chaque tuple à un seul et unique site et s'assurer que tous les tuples ont un site attribué. De manière générale, la réunion des bases doit former la base originale.

II. Présentation du groupe de travail et des rôles de chacun

Notre groupe est constitué des binômes suivants :

- B3412, composé de Lucas Poisse et de Ziggy Vergne
- B3407, composé de Hubert Hamelin et de Cyril Pottiez
- B3414, composé de Julien Charles-Nicolas et Horia Burca

Après discussion, nous avons décidé de nous répartir les sites à gérer de la maniére suivante :

- Lucas Poisse et de Ziggy Vergne ont reçu l'Europe du Sud
- Hubert Hamelin et de Cyril Pottiez ont reçu l'Europe du Nord
- Julien Charles-Nicolas et Horia Burca ont reçu les Amériques

Naturellement, nous avons spécifié des personnes responsables des éléments clés du TP. Ainsi, la coordination générale fût confiée à Lucas Poisse, et la responsabilité de la documentation et du rapport fût confiée à Ziggy Vergne.

III. Fragmentation (Analyse globale et commune à tous les sites)

A. Détermination des fragments

Avant toute chose, nous allons déterminer les usages qui sont faits sur la base de données. Les applications utilisant la base sont MakeIt, DesignIt, SellIt, RH. Après consultation de l'énoncé, nous avons extrait les informations suivantes sur l'utilisation de la base par les applications :

MakeIT :

- N'est utilisé qu'en Europe du Nord
- Lit, écrit et supprime des tuples dans la table Stock quand ils portent sur des objets localisés en Allemagne (et donc en Europe du Nord)
- Lit, écrit et met à jour des tuples dans la tables Fournisseurs

DesignIT :

- N'est utilisé qu'en Europe du Sud
- Lit, écrit et met à jour des tuples dans la tables Categories
- Lit, écrit et met à jour des tuples dans la table Produit
- C'est la seule application pouvant faire modifier Catégories et Produits

SellIT :

- Est déployée sur tous les sites
- Lit, écrit et met à jour des tuples dans les tables Commandes/DetailsCommandes pour les commandes portant sur les clients locaux
- Lit, écrit et met à jour des tuples dans Clients uniquement quand ce sont des clients locaux
- Lit, écrit et met à jour des tuples dans Stock concernent le stock local
- Lit (mais rarement) les tuples dans Stock concernant le stock des autres sites
- Lit les tuples de la table Produit/Categories/Fournisseurs/Employes

RH :

- Est localisée uniquement aux Amériques
- Lit, écrit et met à jour des tuples dans Employes

Nous allons nous préoccuper de l'application SellIt. Elle est présente sur les trois sites, il existe donc trois versions de l'application :

SellIt-US : La version de l'application SellIt pour le site des Amériques

SellIt-EN : La version de l'application SellIt pour le site d'Europe du Nord

SellIt-ES : La version de l'application SellIt pour le site d'Europe du Sud

Nous pouvons donc remarquer la présence de prédicats discriminants, liés à la localisation géographique du pays donc le nom est contenu dans l'attribut « Pays » des tuples de la table « Client » et « Stock ». Il nous est donné les informations suivantes vis-à-vis de la constitution des continents :

- Europe du Nord : Norvège, Suède, Danemark, Islande, Finlande, Royaume-Uni, Irlande, Belgique, Luxembourg, Pays-Bas, Allemagne, Pologne
- Europe du Sud : Espagne, Portugal, Andorre, France, Gibraltar, Italie, Saint Marin, Vatican, Malte, Albanie, Bosnie-Herzégovine, Croatie, Grèce, Macédoine, Monténégro, Serbie, Slovénie, Bulgarie
- Amériques : Antigua-et-Barbuda, Argentine, Bahamas, Barbade, Belize, Bolivie, Brésil, Canada, Chili, Colombie, Costa Rica, Cuba, République dominicaine, Dominique, Équateur, États-Unis, Grenade, Guatemala, Guyana, Haïti, Honduras, Jamaïque, Mexique, Nicaragua, Panama, Paraguay, Pérou, Saint-Christophe-et-Niévès, Sainte Lucie, Saint-Vincent-et-les Grenadines, Salvador, Suriname, Trinité-et-Tobago, Uruguay, Venezuela.

Après consultation de la base de données, nous remarquons qu'il existe des tuples dans la table Clients ayant pour Pays « Autriche » et « Suisse » alors que ces pays n'apparaissent dans aucun des ensembles de pays précédemment décrits. Après concertation, nous avons décidé les inclure dans "Europe du Sud".

Voici donc une conjonction de toutes les combinaisons possibles vis-à-vis de la valeur que peut prendre l'attribut Pays.

Possible ?	Pays IN EuropeSud	Pays IN EuropeNord	Pays IN Ameriques
Oui	Non	Non	Non
Oui	Non	Non	Oui
Oui	Non	Oui	Non
Non	Non	Oui	Oui
Oui	Oui	Non	Non
Non	Oui	Non	Oui
Non	Oui	Oui	Non
Non	Oui	Oui	Oui
Non	null	null	null

Nous remarquons que les intersections des 3 ensembles (Europe du Sud, Europe du Nord, Amériques) sont nulles entre elles. Il ne peut donc pas exister de pays référencés dans deux groupes. De même, la contrainte "NOT NULL" est appliquée sur les attributs Pays dans la base de données, le pays est donc forcément connu. Par

contre, nous pouvons voir qu'un pays peut faire partie d'aucun des continents, son continent est donc considéré comme inconnu.

Nous pouvons donc voir que séparer les tuples par pays nous donne quatre fragments. Un pour les pays appartenant à l'Europe du Nord, un pour les pays appartenant à l'Europe du Sud, un pour les pays appartenant aux Amériques et un dernier pour les pays n'ayant pas de continents attribués.

N'ayant pas identifié d'autre prédicats discriminants et n'ayant pas besoin d'effectuer une fragmentation verticale des tables, nous allons donc procéder à la fragmentation des tables de la base.

Tout d'abord, nous remarquons que les seuls tuples faisant mention de « Pays » sont ceux dans la table « Clients » et « Stock ». Si nous fragmentons ces tables, il convient de fragmenter également les tables ayant des clefs étrangères référençant les tuples de ces deux tables ainsi que les tuples ayant des clefs étrangères sur ces dernières. Ainsi, en plus de fragmenter les tables « Stock » et « Client » nous allons également fragmenter les tables «Commandes », « DetailCommandes ».

Voici la base après fragmentation :

- Clients-AM si Clients.Pays est dans l'ensemble "Amériques"
- Clients-ES si Clients.Pays est dans l'ensemble "Europe du Sud"
- Clients-EN si Clients.Pays est dans l'ensemble "Europe du Nord"
- Clients-OI si le continent d'origine du client est inconnu, c'est-à-dire si le pays du Client n'est pas contenu dans "Amériques", "Europe du Nord" ou "Europe du Sud".
- Stock-AM si Stock.Pays est dans l'ensemble "Amériques"
- Stock-ES si Stock.Pays est dans l'ensemble "Europe du Sud"
- Stock-EN si Stock.Pays est dans l'ensemble "Europe du Nord"
- Stock-OI si le continent de Stock.Pays est inconnu, c'est-à-dire s'il n'est pas contenu dans "Amériques", "Europe du Nord" ou "Europe du Sud".
- Commandes-AM si la commande a été passé par un Client contenu dans Clients-AM
- Commandes-ES si la commande a été passé par un Client contenu dans Clients-ES
- Commandes-EN si la commande a été passé par un Client contenu dans Clients-EN
- Commandes-OI si la commande a été passé par un Client contenu dans Clients-OI
- DétailsCommandes-AM si la commande associée est dans Commandes-US
- DétailsCommandes-ES si la commande associée est dans Commandes-ES
- DétailsCommandes-EN si la commande associée est dans Commandes-EN
- DétailsCommandes-OI si la commande associée est dans Commandes-OI
- Produits
- Catégories
- Fournisseurs
- Employes

Nous pouvons constater que la fragmentation s'effectue sans perte d'information et qu'elle permet la reconstruction de la base d'origine par une simple union. La fragmentation ayant été effectuée, nous allons répartir les fragments sur les différents sites.

B. Placement des fragments sur les sites (sans réplication)

Nous allons tout d'abord voir la répartition des applications sur les différents sites. Nous avons choisi de considérer que l'application SellIt-OI allait être utilisée sur le site en Europe du Nord.

Application/Site	Europe du Nord	Europe du Sud	Amériques
MakeIt	X		
DesignIt		X	
SellIt-AM			X
SellIt-ES		X	
SellIt-EN	X		
SellIt-OI	X		
RH			X

A partir des informations de l'énoncé sur l'utilisation des données, nous établissons un tableau d'utilisation des données de la base par les applications. Ce tableau nous permettra de répartir les fragments en fonction de leur utilisation. R signifie une utilisation en lecture et W une utilisation en écriture, la majuscule différencie les petites et les grandes utilisations.

Fragment/Site	MakeIt	DesignIt	RH	SellIt-AM	SellIt-ES	SellIt-EN	SellIt-OI
Commandes-US				R/W			
Commandes-ES					R/W		
Commandes-EN						R/W	
Commandes-OI							R/W
DétailsCommandes-US				R/W			
DétailsCommandes-ES					R/W		
DétailsCommandes-EN						R/W	
DétailsCommandes-OI							R/W
Clients-AM				R/W			

Clients-ES					R/W		
Clients-EN						R/W	
Clients-OI							R/W
Stock-AM				R/W	r	r	r
Stock-ES				r	R/W	r	r
Stock-EN	RW			r	r	R/W	r
Stock-OI				r	r	r	R/W
Produits		RW					
Catégories		RW					
Fournisseurs	RW			R	R	R	R
Employés			R/W	R	R	R	R

A partir des deux tableaux, nous avons déduit la répartition des différents fragments manière à ne pas dupliquer l'information et de rapprocher les données des lieux les lisant le plus. Si plusieurs lieux lisent les mêmes données, nous plaçons le fragment sur le lieu pouvant éditer les données. Voici notre choix de répartition :

Fragment/Site	Europe du Nord	Europe du Sud	Amériques
Commandes-AM			X
Commandes-ES		X	
Commandes-EN	X		
Commandes-OI	X		
DétailsCommandes-AM			X
DétailsCommandes-ES		X	
DétailsCommandes-EN	X		
DétailsCommandes-OI	X		
Clients-AM			X
Clients-ES		X	
Clients-EN	X		
Clients-OI	X		
Stock-AM			X

Stock-ES		X	
Stock-EN	X		
Stock-OI	X		
Produits		X	
Catégories		X	
Fournisseurs	X		
Employés			X

Maintenant que chacun des fragments a trouvé un site, nous allons pouvoir passer à la mise en place de la base sans réplication.

C Mise en œuvre de la base sans réplication

1 Europe du Nord

Nous allons maintenant mettre en œuvre la base sans réplication pour le site de l'Europe du Nord.

a) Binôme responsable

Le binôme responsable de l'Europe du Nord est le binôme B3407, composé de Hubert Hamelin et de Cyril Pottiez.

b) Création des liens entre les bases

Nous créons les liens vers les bases de données des autres binômes. Veuillez noter que nous utilisons la machine de départ pour héberger le site de l'Europe du Nord, DB1, car la machine initialement prévue à cet effet, DB2, est non-fonctionnelle.

Nous appellons LinkToDBES le lien vers la base de données d'Europe du Sud et LinkToDBUS le lien vers la base des Amériques.

```
CREATE DATABASE LINK LinkToDBES CONNECT TO hhamelin IDENTIFIED BY mdporacle
USING 'DB3';
CREATE DATABASE LINK LinkToDBUS CONNECT TO hhamelin IDENTIFIED BY mdporacle
USING 'DB4';
```

d) Création des tables & peuplement des tables

```
-- création de la table pour les clients de l'europe du nord
CREATE TABLE clientsEN AS
  SELECT *
  FROM ryori.clients
  WHERE
    pays IN ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne');

-- création de la table pour les clients d'un nouveau site
CREATE TABLE clientsOI AS
```

```
  SELECT *
  FROM ryori.clients
  WHERE pays = 'babla';

-- création table des commandes pour les clients de l'europe du nord
CREATE TABLE commandesEN AS
  SELECT r.*
  FROM clientsEN c, ryori.commandes r
  WHERE c.code_client = r.code_client;

-- création table des commandes pour les clients d'un nouveau site
CREATE TABLE commandesOI AS
  SELECT r.*
  FROM clientsOI c, ryori.commandes r
  WHERE c.code_client = r.code_client;

-- création table de détails des commande pour les clients de l'europe du
nord
CREATE TABLE details_commandesEN AS
  SELECT r.*
  FROM commandesEN c, ryori.details_commandes r
  WHERE c.no_commande = r.no_commande;

-- création table de détails des commande pour les clients d'un nouveau
site
CREATE TABLE details_commandesOI AS
  SELECT r.*
  FROM commandesOI c, ryori.details_commandes r
  WHERE c.no_commande = r.no_commande;

--création table de stock des sites de l'europe du nord
CREATE TABLE stockEN AS
  SELECT DISTINCT r.*
  FROM clientsEN c, ryori.stock r
  WHERE c.pays = r.pays;

--création table de stock des nouveaux sites
CREATE TABLE stockOI AS
  SELECT DISTINCT r.*
  FROM clientsOI c, ryori.stock r
  WHERE c.pays = r.pays;

-- table fournisseurs
CREATE TABLE fournisseurs AS
  SELECT *
  FROM ryori.fournisseurs;
```

c) Contraintes d'intégrité

Tout d'abord, nous ajoutons les contraintes pouvant être gérées en local, c'est-à-dire les contraintes de type clef primaire et certaines contraintes de type clef étrangères.

```
-- contraintes de clé primaire
ALTER TABLE ClientsEN
  ADD CONSTRAINT ClientsENPk PRIMARY KEY (code_client);
```

```
ALTER TABLE ClientsOI
  ADD CONSTRAINT ClientsOIPk PRIMARY KEY (code_client);

ALTER TABLE commandesEN
  ADD CONSTRAINT CommandesENPk PRIMARY KEY (no_commande);

ALTER TABLE commandesOI
  ADD CONSTRAINT CommandesOIPk PRIMARY KEY (no_commande);

ALTER TABLE details_commandesEN
  ADD CONSTRAINT Details_commandesENPk PRIMARY KEY (no_commande,
ref_produit);

ALTER TABLE details_commandesOI
  ADD CONSTRAINT Details_commandesOIPk PRIMARY KEY (no_commande,
ref_produit);

ALTER TABLE fournisseurs
  ADD CONSTRAINT FournisseursPk PRIMARY KEY (no_fournisseur);

ALTER TABLE stockEN
  ADD CONSTRAINT StockENPk PRIMARY KEY (ref_produit, pays);

ALTER TABLE stockOI
  ADD CONSTRAINT StockOIPk PRIMARY KEY (ref_produit, pays);

--contraintes de clé étrangère

ALTER TABLE commandesEN
  ADD CONSTRAINT Fk_ClientsEN FOREIGN KEY (code_client) REFERENCES
clientsEN (code_client);

ALTER TABLE commandesOI
  ADD CONSTRAINT Fk_ClientsOI FOREIGN KEY (CODE_CLIENT) REFERENCES
CLIENTSOI (CODE_CLIENT);


ALTER TABLE DETAILS_COMMANDESEN
  ADD CONSTRAINT Fk_CommandesEN FOREIGN KEY (NO_COMMANDE) REFERENCES
COMMANDESEN (NO_COMMANDE);

ALTER TABLE DETAILS_COMMANDESOI
  ADD CONSTRAINT Fk_CommandesOI FOREIGN KEY (NO_COMMANDE) REFERENCES
COMMANDESOI (NO_COMMANDE);
```

De plus, nous devons ajouter des contraintes supplémentaires dans les tables CommandesEN, CommandesOI, DetailsCommandesEN, DetailsCommandesOI, StockEN, StockOI et Fournisseurs car ces tables ont besoin de vérifier l'états de tables distantes pour maintenir leurs contraintes de clef étrangére.

```
-- trigger pour vérifier la présence de l'employé à l'insertion ou maj dans
CommandesEN
CREATE OR REPLACE TRIGGER trg_check_employesEN
BEFORE INSERT OR UPDATE ON COMMANDESEN
FOR EACH ROW
  DECLARE
    Employee NUMBER(6);
```

```
  BEGIN
    SELECT e.no_employe
    INTO employee
    FROM hcburca.Employes@LinkToDBUS e
    WHERE e.no_employe = :new.no_employe;

    EXCEPTION
    WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20001,'Invalid no_employe : There is no
Employe with this no_employe in the company !');

  END;

-- trigger pour vérifier la présence de l'employé à l'insertion ou maj dans
CommandesOI
CREATE OR REPLACE TRIGGER trg_check_employesOI
BEFORE INSERT OR UPDATE ON COMMANDESOI
FOR EACH ROW
  DECLARE
    Employee NUMBER(6);
  BEGIN
    SELECT e.no_employe
    INTO employee
    FROM hcburca.Employes@LinkToDBUS e
    WHERE e.no_employe = :new.no_employe;

    EXCEPTION
    WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20001,'Invalid no_employe : There is no
Employe with this no_employe in the company !');

  END;

--  trigger pour vérifier la présence du produit à insérer dans
DETAILS_COMMANDESEN
CREATE OR REPLACE TRIGGER trg_check_CEN_produits
BEFORE INSERT OR UPDATE ON DETAILS_COMMANDESEN
FOR EACH ROW
  DECLARE
    produit NUMBER(6);
  BEGIN
    SELECT p.ref_produit
    INTO produit
    FROM lpoisse.produits@LinkToDBES p
    WHERE p.ref_produit = :new.REF_PRODUIT;

    EXCEPTION
    WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20001, 'Invalid ref_produit : There is no
product with this ref_produit in the company !');
  END;

--  trigger pour vérifier la présence du produit à insérer dans
DETAILS_COMMANDESOI
CREATE OR REPLACE TRIGGER trg_check_COI_produits
BEFORE INSERT OR UPDATE ON DETAILS_COMMANDESOI
FOR EACH ROW
  DECLARE
```

```
    produit NUMBER(6);
  BEGIN
    SELECT p.ref_produit
    INTO produit
    FROM lpoisse.produits@LinkToDBES p
    WHERE p.ref_produit = :new.REF_PRODUIT;

    EXCEPTION
    WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20001,'Invalid ref_produit : There is no
ref_produit with this ref_produit in the company !');
  END;

--  trigger pour vérifier si le produit inséré est bien référencé dans
stockEN
CREATE OR REPLACE TRIGGER trg_check_stockEN_produits
BEFORE INSERT OR UPDATE ON STOCKEN
FOR EACH ROW
  DECLARE
    produit NUMBER(6);
  BEGIN
    SELECT p.ref_produit
    INTO produit
    FROM lpoisse.produits@LinkToDBES p
    WHERE p.ref_produit = :new.REF_PRODUIT;

    EXCEPTION
    WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20001, 'Invalid ref_produit : There is no
ref_produit with this ref_produit in the company !');
  END;

--  trigger pour vérifier si le produit inséré est bien référencé dans
STOCKOI
CREATE OR REPLACE TRIGGER trg_check_stockoi_produits
BEFORE INSERT OR UPDATE ON STOCKOI
FOR EACH ROW
  DECLARE
    produit NUMBER(6);
  BEGIN
    SELECT p.ref_produit
    INTO produit
    FROM lpoisse.produits@LinkToDBES p
    WHERE p.ref_produit = :new.REF_PRODUIT;

    EXCEPTION
    WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20001, 'Invalid ref_produit : There is no
ref_produit with this ref_produit in the company !');
  END;

-- vérifier si le fournisseurs à supprimer/mettre à jour n'est déjà
référencé dans la table produits
CREATE OR REPLACE TRIGGER chk_Fournisseurs
BEFORE DELETE OR UPDATE ON FOURNISSEURS
FOR EACH ROW
  DECLARE
```

```
    number_of_rows NUMBER; -- nombre de tuples trouvés dans table produit
pour le fournisseurs qui doit être supprimé

  BEGIN

    SELECT count(*)
    INTO number_of_rows
    FROM lpoisse.produits@LinkToDBES
    WHERE NO_FOURNISSEUR = :old.NO_FOURNISSEUR;

    IF (DELETING)
    THEN
      IF number_of_rows <> 0
      THEN
        raise_application_error(-20002,'Erreur : le fournisseur à supprimer
est déjà référencé dans la table produits en europe du sud');
      END IF;
    END IF;

    IF (UPDATING)
    THEN
      IF number_of_rows <> 0
      THEN
        raise_application_error(-20002, 'Erreur : le fournisseur à mettre à
jour est déjà référencé dans la table produits en europe du sud');
      END IF;
    END IF;

  END;
```

d) Droits d'accès

Nous donnons les droits d'accès en lecture, écriture et suppressions sur les tables de notre base aux autres membres du groupe.

```
GRANT SELECT, update, insert, delete ON stockEN TO lpoisse;
GRANT SELECT, update, insert, delete ON stockEN TO zvergne;
GRANT SELECT, update, insert, delete ON stockEN TO hcburca;
GRANT SELECT, update, insert, delete ON stockEN TO jcharlesni;
GRANT SELECT, update, insert, delete ON stockEN TO hhamelin;

GRANT SELECT, update, insert, delete ON stockOI TO lpoisse;
GRANT SELECT, update, insert, delete ON stockOI TO zvergne;
GRANT SELECT, update, insert, delete ON stockOI TO hcburca;
GRANT SELECT, update, insert, delete ON stockOI TO jcharlesni;
GRANT SELECT, update, insert, delete ON stockOI TO hhamelin;

GRANT SELECT, update, insert, delete ON fournisseurs TO lpoisse;
GRANT SELECT, update, insert, delete ON fournisseurs TO zvergne;
GRANT SELECT, update, insert, delete ON fournisseurs TO hcburca;
GRANT SELECT, update, insert, delete ON fournisseurs TO jcharlesni;
GRANT SELECT, update, insert, delete ON fournisseurs TO hhamelin;

GRANT SELECT, update, insert, delete ON COMMANDESEN TO lpoisse;
GRANT SELECT, update, insert, delete ON COMMANDESEN TO zvergne;
GRANT SELECT, update, insert, delete ON COMMANDESEN TO hcburca;
```

```
GRANT SELECT, update, insert, delete ON COMMANDESEN TO jcharlesni;
GRANT SELECT, update, insert, delete ON COMMANDESEN TO hhamelin;

GRANT SELECT, update, insert, delete ON commandesoi TO lpoisse;
GRANT SELECT, update, insert, delete ON commandesoi TO zvergne;
GRANT SELECT, update, insert, delete ON commandesoi TO hcburca;
GRANT SELECT, update, insert, delete ON commandesoi TO jcharlesni;
GRANT SELECT, update, insert, delete ON commandesoi TO hhamelin;

GRANT SELECT, update, insert, delete ON details_commandesoi TO lpoisse;
GRANT SELECT, update, insert, delete ON details_commandesoi TO zvergne;
GRANT SELECT, update, insert, delete ON details_commandesoi TO hcburca;
GRANT SELECT, update, insert, delete ON details_commandesoi TO jcharlesni;
GRANT SELECT, update, insert, delete ON details_commandesoi TO hhamelin;

GRANT SELECT, update, insert, delete ON details_commandesEn TO lpoisse;
GRANT SELECT, update, insert, delete ON details_commandesEn TO zvergne;
GRANT SELECT, update, insert, delete ON details_commandesEn TO hcburca;
GRANT SELECT, update, insert, delete ON details_commandesEn TO jcharlesni;
GRANT SELECT, update, insert, delete ON details_commandesEn TO hhamelin;

GRANT SELECT, update, insert, delete ON clientsen TO lpoisse;
GRANT SELECT, update, insert, delete ON clientsen TO zvergne;
GRANT SELECT, update, insert, delete ON clientsen TO hcburca;
GRANT SELECT, update, insert, delete ON clientsen TO jcharlesni;
GRANT SELECT, update, insert, delete ON clientsen TO hhamelin;

GRANT SELECT, update, insert, delete ON clientsoi TO lpoisse;
GRANT SELECT, update, insert, delete ON clientsoi TO zvergne;
GRANT SELECT, update, insert, delete ON clientsoi TO hcburca;
GRANT SELECT, update, insert, delete ON clientsoi TO jcharlesni;
GRANT SELECT, update, insert, delete ON clientsoi TO hhamelin;
```

f) Définition de synonymes et de vues pour interrogation de la base comme si elle était en centralisé.

Nous rédigeons la définition des vues afin d'avoir le même comportement qu'avec la base en centralisée. Nous définissons donc une vue pour la table Stock, Commandes, Clients, Commandes, Détails_Commandes. De plus, nous créons des vues sur les tables Employes, Categories et Produits pour avoir un accès facile en lecture à ces dernières.

```
-- Vue "Stock", création avec WHERE pour optimiser le plan d'exécution
CREATE OR REPLACE VIEW Stock
AS
  (SELECT *
   FROM lpoisse.stockes@LinkToDBES
   WHERE StockES.PAYS IN
         ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican','Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie',
          'Autriche', 'Suisse')
   UNION ALL
```

```
   SELECT *
   FROM stockEN
   WHERE pays IN
         ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas')
   UNION ALL
   SELECT *
   FROM stockOI
   WHERE pays NOT IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil','Canada', 'Chili', 'Colombie',
'Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique','Equateur',
'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras',
'Jamaique','Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-
Christophe-et-Nieves', 'Sainte-Lucie','Saint-Vincent-et-les Grenadines',
'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay',
                     'Venezuela') AND pays NOT IN
                                    ('Suede', 'Norvege', 'Danemark',
'Finlande', 'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne',
'Islande', 'Luxembourg', 'Pays-Bas')
         AND pays NOT IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican',
                                  'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie',
                          'Autriche', 'Suisse')
   UNION ALL
   SELECT *
   FROM hcburca.stock_am@LinkToDBUS
   WHERE pays IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade',
'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica',
'Cuba', 'Republique dominicaine', 'Dominique', 'Equateur', 'Etats-Unis',
'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaique',
'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie','Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay','Venezuela')
  );

-- Vue 'Produits'
CREATE OR REPLACE VIEW PRODUITS
AS
  (SELECT *
   FROM lpoisse.produits@LinkToDBES
  );

-- Vue 'Categories'
CREATE OR REPLACE VIEW CATEGORIES
AS
  (SELECT *
   FROM lpoisse.categories@LinkToDBES
  );

-- Vue 'Employes'
CREATE OR REPLACE VIEW EMPLOYES
AS
  (SELECT *
   FROM hcburca.employes@LinkToDBUS
  );
```

```
SELECT *
FROM Stock;

--Vue 'Clients', création avec WHERE pour optimiser le plan d'exécution
CREATE OR REPLACE VIEW Clients
AS
  (SELECT *
   FROM lpoisse.ClientsES@LinkToDBES
   WHERE PAYS IN
         ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican','Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie','Autriche', 'Suisse')
   UNION ALL
   SELECT *
   FROM clientsEN
   WHERE pays IN
         ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique','Irlande',
'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg', 'Pays-Bas')
   UNION ALL
   SELECT *
   FROM clientsOI
   WHERE pays NOT IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil','Canada', 'Chili', 'Colombie',
'Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique', 'Equateur',
'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras',
'Jamaique','Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-
Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines',
'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay','Venezuela') AND
pays NOT IN ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas') AND pays NOT IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican','Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie','Autriche', 'Suisse')
   UNION ALL
   SELECT *
   FROM hcburca.clients_am@LinkToDBUS
   WHERE pays IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade',
'Belize', 'Bolivie', 'Bresil','Canada', 'Chili', 'Colombie', 'Costa Rica',
'Cuba', 'Republique dominicaine', 'Dominique','Equateur', 'Etats-Unis',
'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras',
'Jamaique','Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-
Christophe-et-Nieves', 'Sainte-Lucie','Saint-Vincent-et-les Grenadines',
'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay','Venezuela')
  );

--Vue 'Commandes'
CREATE OR REPLACE VIEW Commandes
AS
  (SELECT *
   FROM COMMANDESOI
   UNION ALL
   SELECT *
   FROM COMMANDESEN
   UNION ALL
```

```
   SELECT *
   FROM lpoisse.Commandeses@LinkToDBES
   UNION ALL
   SELECT *
   FROM hcburca.Commandes_AM@LinkToDBUS
  );

-- Vue 'Details_Commandes'
CREATE OR REPLACE VIEW details_commandes
AS
  (SELECT *
   FROM DETAILS_COMMANDESOI
   UNION ALL
   SELECT *
   FROM DETAILS_COMMANDESEN
   UNION ALL
   SELECT *
   FROM lpoisse.details_commandeses@LinkToDBES
   UNION ALL
   SELECT *
   FROM hcburca.Details_Commandes_AM@LinkToDBUS
        );
```

Naturellement, nous ajoutons la possibilité d'effectuer les opérations classiques de modification, insertion, suppression en plus de la selection dans ces vues nouvellement crées. Pour cela, nous redirigeons les requêtes effectuées sur les vues vers les tables correspondantes, mêmes si celles-ci sont localisés sur des sites éloignés.

```
/*
Trigger appelé lors d'une action LMD sur la vue 'Stock'
*/
CREATE OR REPLACE TRIGGER modify_Stocks
INSTEAD OF
UPDATE OR
INSERT OR
DELETE ON Stock
FOR EACH ROW
  BEGIN IF INSERTING
  THEN --Insertion à contrôler
    IF (:NEW.pays IN
        ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique', 'Irlande',
'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg', 'Pays-
Bas'))
    THEN
      INSERT
      INTO STOCKEN VALUES
        (
          :new.ref_produit,
          :new.pays,
          :new.unites_stock,
          :new.unites_commandees,
          :new.indisponible
        );
    ELSIF (:NEW.pays IN
           ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie', 'Autriche', 'Suisse'))
```

```
      THEN
        INSERT
        INTO lpoisse.stockes@LinkToDBES VALUES
          (
            :new.ref_produit,
            :new.pays,
            :new.unites_stock,
            :new.unites_commandees,
            :new.indisponible
          );
    ELSIF (:NEW.pays IN
           ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade',
'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica',
'Cuba', 'Republique dominicaine', 'Dominique', 'Equateur', 'Etats-Unis',
'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaique',
'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela'))
      THEN
        INSERT
        INTO hcburca.stock_am@LinkToDBUS VALUES
          (
            :new.ref_produit,
            :new.pays,
            :new.unites_stock,
            :new.unites_commandees,
            :new.indisponible
          );
    ELSE
      INSERT
      INTO STOCKOI VALUES
        (
          :new.ref_produit,
          :new.pays,
          :new.unites_stock,
          :new.unites_commandees,
          :new.indisponible
        );
    END IF;
  END IF;
    IF UPDATING
    THEN
      IF (:New.pays <> :OLD.pays)
      THEN
           Delete from stock WHERE ref_produit = :old.ref_produit AND pays
= :old.pays;
       INSERT
     INTO stock VALUES
    (
      :new.ref_produit,
      :new.pays,
      :new.unites_stock,
      :new.unites_commandees,
      :new.indisponible
      );


      ELSIF (:NEW.pays IN
```

```
            ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas'))
        THEN
          UPDATE STOCKEN
          SET ref_produit     = :new.ref_produit,
            PAYS              = :NEW.PAYS,
            UNITES_STOCK      = :new.UNITES_STOCK,
            UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
            INDISPONIBLE      = :new.INDISPONIBLE
          WHERE ref_produit = :old.ref_produit
                AND pays = :old.PAYS;
      ELSIF (:NEW.pays IN
            ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie', 'Autriche', 'Suisse'))
        THEN
          UPDATE lpoisse.stockes@LinkToDBES
          SET ref_produit     = :new.ref_produit,
            PAYS              = :NEW.PAYS,
            UNITES_STOCK      = :new.UNITES_STOCK,
            UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
            INDISPONIBLE      = :new.INDISPONIBLE
          WHERE ref_produit = :old.ref_produit
                AND pays = :old.PAYS;
      ELSIF (:NEW.pays IN
            ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade',
'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica',
'Cuba', 'Republique dominicaine', 'Dominique', 'Equateur', 'Etats-Unis',
'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaique',
'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela'))
        THEN
          UPDATE hcburca.stock_am@LinkToDBUS
          SET ref_produit     = :new.ref_produit,
            PAYS              = :NEW.PAYS,
            UNITES_STOCK      = :new.UNITES_STOCK,
            UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
            INDISPONIBLE      = :new.INDISPONIBLE
          WHERE ref_produit = :old.ref_produit
                AND pays = :old.PAYS;
      ELSE
        UPDATE STOCKOI
        SET ref_produit     = :new.ref_produit,
          PAYS              = :NEW.PAYS,
          UNITES_STOCK      = :new.UNITES_STOCK,
          UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
          INDISPONIBLE      = :new.INDISPONIBLE
        WHERE ref_produit = :old.ref_produit
              AND pays = :old.PAYS;
      END IF;
    END IF;
    IF DELETING
    THEN --Suppression à vérifier ; on ne peut pas supprimer un stock ne
faisant pas partie de la région du site (gestion du stock LOCAL seulement)
      IF (:old.pays IN
```

```
          ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas'))
      THEN
        DELETE FROM STOCKEN
        WHERE ref_produit = :old.ref_produit AND pays = :old.pays;
      ELSIF (:old.pays IN
             ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie', 'Autriche', 'Suisse'))
        THEN
          DELETE
          FROM lpoisse.stockes@LinkToDBES
          WHERE ref_produit = :old.ref_produit
                AND pays = :old.pays;
      ELSIF (:old.pays IN
             ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade',
'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica',
'Cuba', 'Republique dominicaine', 'Dominique', 'Equateur', 'Etats-Unis',
'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaique',
'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela'))
        THEN
          DELETE
          FROM hcburca.stock_am@LinkToDBUS
          WHERE ref_produit = :old.ref_produit
                AND pays = :old.pays;
      ELSE
        DELETE FROM STOCKOI
        WHERE ref_produit = :old.ref_produit AND pays = :old.pays;
      END IF;
    END IF;
  END;



/*
  Trigger de modification d'un client
*/

CREATE OR REPLACE TRIGGER modify_Clients
INSTEAD OF UPDATE OR INSERT OR DELETE ON Clients
FOR EACH ROW
  BEGIN
    IF INSERTING
    THEN

      IF (:NEW.pays IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican',
                                  'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie',
                        'Autriche', 'Suisse'))
      THEN

        INSERT INTO lpoisse.ClientsES@LinkToDBES
```

```
        VALUES (:new.code_client, :new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone,
                :new.fax);

      ELSIF (:new.pays IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil',
                                                'Canada', 'Chili',
'Colombie', 'Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',

'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana', 'Haiti',
'Honduras', 'Jamaique',

'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',

'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
                            'Venezuela'))
        THEN
          INSERT INTO hcburca.clients_am@LinkToDBUS
          VALUES (:new.code_client, :new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone,
                  :new.fax);

      ELSIF (:new.pays IN
             ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas'))
        THEN
          INSERT INTO CLIENTSEN
          VALUES (:new.code_client, :new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone,
                  :new.fax);

      ELSE
        INSERT INTO CLIENTSOI
        VALUES (:new.code_client, :new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone,
                :new.fax);
      END IF;
    END IF;

    IF UPDATING
    THEN

      IF (:New.pays <> :OLD.pays)
      THEN
        RAISE_APPLICATION_ERROR(-20010, 'Attention, vous n''êtes pas
autorisé à changer le pays du client');
      ELSIF (:NEW.pays IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse'))
        THEN
          UPDATE lpoisse.ClientsES@LinkToDBES
          SET
            code_client = :new.code_client,
            societe     = :NEW.societe,
```

```
            adresse     = :new.adresse,
            ville       = :new.ville,
            code_postal = :new.code_postal,
            pays        = :new.pays,
            telephone   = :new.telephone,
            fax         = :new.fax
          WHERE code_client = :old.code_client;
      ELSIF (:new.pays IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie',
'Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique','Equateur',
'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras',
'Jamaique','Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-
Christophe-et-Nieves', 'Sainte-Lucie','Saint-Vincent-et-les Grenadines',
'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay','Venezuela'))
        THEN
          UPDATE hcburca.clients_am@LinkToDBUS
          SET
            code_client = :new.code_client,
            societe     = :NEW.societe,
            adresse     = :new.adresse,
            ville       = :new.ville,
            code_postal = :new.code_postal,
            pays        = :new.pays,
            telephone   = :new.telephone,
            fax         = :new.fax
          WHERE code_client = :old.code_client;
      ELSIF (:new.pays IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande',
                                    'Pologne', 'Royaume-Uni', 'Allemagne',
'Islande', 'Luxembourg', 'Pays-Bas'))
        THEN
          UPDATE CLIENTSEN
          SET
            code_client = :new.code_client,
            societe     = :NEW.societe,
            adresse     = :new.adresse,
            ville       = :new.ville,
            code_postal = :new.code_postal,
            pays        = :new.pays,
            telephone   = :new.telephone,
            fax         = :new.fax
          WHERE code_client = :old.code_client;
      ELSE
        UPDATE CLIENTSOI
        SET
          code_client = :new.code_client,
          societe     = :NEW.societe,
          adresse     = :new.adresse,
          ville       = :new.ville,
          code_postal = :new.code_postal,
          pays        = :new.pays,
          telephone   = :new.telephone,
          fax         = :new.fax
        WHERE code_client = :old.code_client;
      END IF;
    END IF;

    IF DELETING
```

```
    THEN

      IF (:old.pays IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican',
                                    'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie',
                        'Autriche', 'Suisse'))
      THEN
        DELETE FROM lpoisse.ClientsES@LinkToDBES
        WHERE code_client = :old.code_client;

      ELSIF (:old.pays IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie',
'Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique','Equateur',
'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras',
'Jamaique','Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-
Christophe-et-Nieves', 'Sainte-Lucie','Saint-Vincent-et-les Grenadines',
'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela'))
        THEN
          DELETE FROM hcburca.clients_am@LinkToDBUS
          WHERE code_client = :old.code_client;
      ELSIF (:old.pays IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas'))
        THEN
          DELETE FROM CLIENTSEN
          WHERE code_client = :old.code_client;
      ELSE
        DELETE FROM CLIENTSOI
        WHERE code_client = :old.code_client;
      END IF;
    END IF;
  END;

/*
      Trigger pour insert, update, delete on Commandes
*/
CREATE OR REPLACE TRIGGER MODIFY_COMMANDES
INSTEAD OF DELETE OR INSERT OR UPDATE ON COMMANDES
FOR EACH ROW
  DECLARE
    paysclient$ VARCHAR2(15);
  BEGIN
    IF (INSERTING OR UPDATING)
    THEN
      SELECT DISTINCT Pays
      INTO paysclient$
      FROM CLIENTS
      WHERE code_client = :new.code_client;
    ELSIF (DELETING)
      THEN
        SELECT DISTINCT Pays
        INTO paysclient$
        FROM CLIENTS
        WHERE code_client = :old.code_client;
    END IF;
```

```
    IF (INSERTING)
    THEN
      IF (paysclient$ IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie',
'Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique','Equateur',
'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras',
'Jamaique', 'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-
Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines',
'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay','Venezuela'))
      THEN
        INSERT INTO hcburca.Commandes_AM@LinkToDBUS
        VALUES (:new.no_commande, :new.code_client, :new.no_employe,
:new.date_commande, :new.date_envoi, :new.port);
      ELSIF (paysclient$ IN
             ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie','Autriche', 'Suisse'))
        THEN
          INSERT INTO lpoisse.Commandeses@LinkToDBES
          VALUES (:new.no_commande, :new.code_client, :new.no_employe,
:new.date_commande, :new.date_envoi, :new.port);
      ELSIF (paysclient$ IN
             ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas'))
        THEN
          INSERT INTO COMMANDESEN
          VALUES (:new.no_commande, :new.code_client, :new.no_employe,
:new.date_commande, :new.date_envoi, :new.port);
      ELSE
        INSERT INTO COMMANDESOI
        VALUES (:new.no_commande, :new.code_client, :new.no_employe,
:new.date_commande, :new.date_envoi, :new.port);
      END IF;
    ELSIF (UPDATING)
      THEN
        IF (:new.code_client <> :old.code_client)
        THEN
          RAISE_APPLICATION_ERROR(-20010, 'Attention, vous n''êtes pas
autorisé à changer le client de cet commande');
        ELSE
          IF (paysclient$ IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie',
'Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana', 'Haiti',
'Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama', 'Paraguay',
'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie','Saint-Vincent-et-les
Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay',
'Venezuela'))
          THEN
            UPDATE hcburca.Commandes_AM@LinkToDBUS
            SET no_commande = :new.no_commande, code_client =
:new.code_client, no_employe = :new.no_employe,
                date_commande = :new.date_commande, date_envoi =
:new.date_envoi, port = :new.port
            WHERE no_commande = :old.no_commande;
          ELSIF (paysclient$ IN
```

```
                ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie','Autriche', 'Suisse'))
            THEN
              UPDATE lpoisse.Commandeses@LinkToDBES
              SET no_commande = :new.no_commande, code_client =
:new.code_client, no_employe = :new.no_employe,
                date_commande = :new.date_commande, date_envoi =
:new.date_envoi, port = :new.port
              WHERE no_commande = :old.no_commande;
          ELSIF (paysclient$ IN
                ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas'))
            THEN
              UPDATE commandesEN
              SET no_commande = :new.no_commande, code_client =
:new.code_client, no_employe = :new.no_employe,
                date_commande = :new.date_commande, date_envoi =
:new.date_envoi, port = :new.port
              WHERE no_commande = :old.no_commande;
          ELSE
            UPDATE commandesOI
            SET no_commande = :new.no_commande, code_client =
:new.code_client, no_employe = :new.no_employe,
              date_commande = :new.date_commande, date_envoi =
:new.date_envoi, port = :new.port
            WHERE no_commande = :old.no_commande;
          END IF;
        END IF;
    ELSIF (DELETING)
      THEN
        IF (paysclient$ IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie',
'Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique','Equateur',
'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras',
'Jamaique','Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-
Christophe-et-Nieves', 'Sainte-Lucie','Saint-Vincent-et-les Grenadines',
'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay','Venezuela'))
        THEN
          DELETE FROM hcburca.Commandes_AM@LinkToDBUS
          WHERE no_commande = :old.no_commande;
        ELSIF (paysclient$ IN
                ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie','Autriche', 'Suisse'))
          THEN
            DELETE FROM lpoisse.Commandeses@LinkToDBES
            WHERE no_commande = :old.no_commande;
        ELSIF (paysclient$ IN
                ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
                        'Luxembourg', 'Pays-Bas'))
          THEN
            DELETE FROM commandesEN
            WHERE no_commande = :old.no_commande;
```

```
        ELSE
          DELETE FROM commandesOI
          WHERE no_commande = :old.no_commande;
        END IF;
    END IF;

    EXCEPTION
    WHEN no_data_found THEN
    raise_application_error(-20011, 'Erreur : le client indiqué est
inconnu.');

  END;


/*
Trigger appelé lors d'une action LMD sur la vue 'DETAILS_COMMANDES'
*/
CREATE OR REPLACE TRIGGER modify_DetailsCommandes
INSTEAD OF
UPDATE OR
INSERT OR
DELETE ON DETAILS_COMMANDES
FOR EACH ROW
  DECLARE
    paysTest VARCHAR2(24);
  BEGIN
    IF (UPDATING OR INSERTING)
    THEN
      SELECT pays
      INTO paystest
      FROM CLIENTS
        NATURAL JOIN COMMANDES
      WHERE no_commande = :new.no_commande;
    END IF;

    IF INSERTING
    THEN --Insertion à contrôler

      IF (paysTest IN
          ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas'))
      THEN
        INSERT
        INTO DETAILS_COMMANDESEN VALUES
          (
            :new.no_commande,
            :new.ref_produit,
            :new.prix_unitaire,
            :new.quantite,
            :new.remise
          );
      ELSIF (paysTest IN
           ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie', 'Autriche', 'Suisse'))
        THEN
```

```
          INSERT
          INTO lpoisse.details_commandeses@LinkToDBES VALUES
            (
              :new.no_commande,
              :new.ref_produit,
              :new.prix_unitaire,
              :new.quantite,
              :new.remise
            );
      ELSIF (paysTest IN
             ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade',
'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica',
'Cuba', 'Republique dominicaine', 'Dominique', 'Equateur', 'Etats-Unis',
'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaique',
'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela'))
        THEN
          INSERT
          INTO hcburca.Details_Commandes_AM@LinkToDBUS VALUES
            (
              :new.no_commande,
              :new.ref_produit,
              :new.prix_unitaire,
              :new.quantite,
              :new.remise
            );
      ELSE
        INSERT
        INTO details_commandesOI VALUES
          (
            :new.no_commande,
            :new.ref_produit,
            :new.prix_unitaire,
            :new.quantite,
            :new.remise
          );
      END IF;
    END IF;
    IF UPDATING
    THEN -- Modification à contrôler : on ne peut pas modifier les données
d'un stock non local !

      IF (:old.ref_produit = :new.ref_produit AND :old.NO_COMMANDE =
:new.no_commande)
      THEN
        IF (paysTest IN
            ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas'))
        THEN
          UPDATE DETAILS_COMMANDESEN
          SET ref_produit = :new.ref_produit,
            NO_COMMANDE   = :new.no_commande,
            PRIX_UNITAIRE = :new.PRIX_UNITAIRE,
            QUANTITE      = :new.QUANTITE,
            REMISE        = :new.REMISE
          WHERE ref_produit = :old.ref_produit
```

```
                AND no_commande = :old.no_commande;
        ELSIF (paysTest IN
                ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie', 'Autriche', 'Suisse'))
          THEN
            UPDATE lpoisse.details_commandeses@LinkToDBES
            SET ref_produit = :new.ref_produit,
              NO_COMMANDE   = :new.no_commande,
              PRIX_UNITAIRE = :new.PRIX_UNITAIRE,
              QUANTITE      = :new.QUANTITE,
              REMISE        = :new.REMISE
            WHERE ref_produit = :old.ref_produit
                  AND no_commande = :old.no_commande;
        ELSIF (paysTest IN
                ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade',
'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica',
'Cuba', 'Republique dominicaine', 'Dominique', 'Equateur', 'Etats-Unis',
'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaique',
'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela'))
          THEN
            UPDATE hcburca.Details_Commandes_AM@LinkToDBUS
            SET ref_produit = :new.ref_produit,
              NO_COMMANDE   = :new.no_commande,
              PRIX_UNITAIRE = :new.PRIX_UNITAIRE,
              QUANTITE      = :new.QUANTITE,
              REMISE        = :new.REMISE
            WHERE ref_produit = :old.ref_produit
                  AND no_commande = :old.no_commande;
        ELSE
          UPDATE details_commandesOI
          SET ref_produit = :new.ref_produit,
            NO_COMMANDE   = :new.no_commande,
            PRIX_UNITAIRE = :new.PRIX_UNITAIRE,
            QUANTITE      = :new.QUANTITE,
            REMISE        = :new.REMISE
          WHERE ref_produit = :old.ref_produit
                AND no_commande = :old.no_commande;
        END IF;
        ELSE
          raise_application_error(-20012, 'Vous ne pouvez pas changez le
numero de commande');
      END IF;
    END IF;
    IF DELETING
    THEN

      --   DBMS_OUTPUT.PUT_LINE(:old.no_commande||' - ' ||
:old.ref_produit);
      SELECT pays
      INTO paysTest
      FROM CLIENTS
        NATURAL JOIN COMMANDES
      WHERE no_commande = :old.no_commande;
```

```
      IF (paysTest IN
          ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas'))
      THEN
        DELETE
        FROM DETAILS_COMMANDESEN
        WHERE ref_produit = :old.ref_produit
              AND no_commande = :old.no_commande;

      ELSIF (paysTest IN
             ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie', 'Autriche', 'Suisse'))
        THEN

          DELETE
          FROM lpoisse.details_commandeses@LinkToDBES
          WHERE ref_produit = :old.ref_produit
                AND no_commande = :old.no_commande;
      ELSIF (paysTest IN
             ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade',
'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica',
'Cuba', 'Republique dominicaine', 'Dominique', 'Equateur', 'Etats-Unis',
'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaique',
'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela'))
        THEN
          DELETE
          FROM hcburca.Details_Commandes_AM@LinkToDBUS
          WHERE ref_produit = :old.ref_produit
                AND no_commande = :old.no_commande;
      ELSE
        DELETE
        FROM details_commandesOI
        WHERE ref_produit = :old.ref_produit
              AND no_commande = :old.no_commande;
      END IF;
    END IF;

    EXCEPTION
    WHEN no_data_found THEN
    raise_application_error(-20012, 'La commande indiquée est inconnue');
END;
```

g) Nettoyages éventuels

Les nettoyages des tables ont été effectués. Ils consistaient à supprimer les tuples ajoutés pour les tests.

h) Tests de vérification du bon fonctionnement

Pour des raisons de lisibilité, nous avons placé l'ensemble de nos tests en Annexe, à la fin de ce document. En effet, ces derniers font cinq pages.

2 Europe du Sud

a) Binôme responsable

Le binôme responsable de l'Europe du Sud est le binôme B3412, constitué de Lucas Poisse et de Ziggy Vergne.

b) Création des liens entre les bases

Nous créons les liens vers les bases de données des autres binômes. Veuillez noter que nous utilisons la machine de départ pour héberger le site de l'Europe du Nord, DB1, car la machine initialement prévue à cet effet, DB2, est non-fonctionnelle.

Nous appellons dblinkMain le lien vers la base de données d'Europe du Nord et DBlinkUS le lien vers la base des Amériques.

```
--Liens de BDD
create database link dblinkMain CONNECT TO lpoisse IDENTIFIED BY mdporacle
USING 'DB1';
CREATE DATABASE LINK dbLinkUS CONNECT TO lpoisse IDENTIFIED BY mdporacle
USING 'DB4';
```

c) Création des tables & Peuplement des tables

Nous procédons à la création des différentes tables sur notre base et nous les remplissons aussitôt grâce à la commande CREATE TABLE AS. Le remplissage se fait en sélectionnant les données en fonction des prédicats définis précédemment.

```
--Création de la table clients de l'Europe du Sud AVEC Autriche/Suisse
--Nous nommons cette table « clientsES »
CREATE TABLE clientsES as
(SELECT * FROM ryori.clients@dblinkMain
WHERE pays IN ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie', 'Autriche', 'Suisse'));

--Création de la table commandes de l'europe du Sud (CommandesES)
--Pour que une commande soit sélectionnée il faut qu'elle soit relative à
--un client du continent Europe du Sud

CREATE TABLE commandesES as
(
SELECT * from ryori.commandes@dblinkMain WHERE NO_COMMANDE IN (
SELECT NO_COMMANDE FROM (
SELECT * FROM ryori.commandes@dblinkMain com NATURAL JOIN
ryori.clients@dblinkMain cli
WHERE cli.pays IN ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie', 'Autriche', 'Suisse'))
));
```

```
-- Création de la table détails commande de l'europe du Sud
-- (details_CommandesES)
-- Pour que une commande soit sélectionnée il faut qu'elle soit relative à
-- un client du continent Europe du Sud
CREATE TABLE details_commandesES as
(
SELECT * from ryori.details_commandes@dblinkMain WHERE NO_COMMANDE IN (
SELECT NO_COMMANDE FROM (
SELECT * FROM ryori.commandes@dblinkMain com NATURAL JOIN
ryori.clients@dblinkMain cli
WHERE cli.pays IN ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie', 'Autriche', 'Suisse'))
));



 -- Création de la table stock de l'europe du Sud (stockES)
CREATE TABLE stockES as
(
SELECT * from ryori.stock@dblinkMain
WHERE pays IN ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-
Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie',
'Slovenie', 'Bulgarie', 'Autriche', 'Suisse'));


 --Création de la table Produits à partir de l'originale
CREATE TABLE produits as
(
SELECT * from ryori.produits@dblinkMain);



 --Création de la table Categories à partir de l'originale
CREATE TABLE CATEGORIES as
(
SELECT * from ryori.CATEGORIES@dblinkMain);
```

d) Contraintes d'intégrité

Nous devons désormais appliquer des contraintes d'intégrités à nos tables. Ces contraintes doivent coller aux contraintes présentes sur la base originale. Les contraintes de la base originale sont de type « clef primaire », « clef etrangere », « not null ». L'écriture des contraintes peut donc se séparer en deux parties : la partie « locale » qui ne concerne que les tuples présents en local et la partie déportée qui elle fait intervenir des Triggers qui sont nécessaires pour regarder et éditer le contenu des autres bases de données pour maintenir la cohérence des données en cas d'insertion, de modification et suppression.

Voici la partie « locale » :

Clefs primaires :

```
      ALTER TABLE clientsES ADD CONSTRAINT pk_clientsES PRIMARY KEY
(CODE_CLIENT);
      ALTER TABLE commandesES ADD CONSTRAINT pk_commandesES PRIMARY KEY
(NO_COMMANDE);
      ALTER TABLE details_commandesES ADD CONSTRAINT pk_detailsCommandesES
PRIMARY KEY (NO_COMMANDE, REF_PRODUIT);
      ALTER TABLE stockES ADD CONSTRAINT pk_StockES PRIMARY KEY
(REF_PRODUIT, PAYS);
      ALTER TABLE produits add constraint pk_produits PRIMARY KEY
(REF_PRODUIT);
      ALTER TABLE categories ADD CONSTRAINT pk_Categories PRIMARY KEY
(CODE_CATEGORIE);
```

Not null :

```
      ALTER TABLE CommandesES ADD CONSTRAINT chk_ccnotnull CHECK
(CODE_CLIENT IS NOT NULL);
      ALTER TABLE CommandesES ADD CONSTRAINT chk_noempnotnull CHECK
(NO_EMPLOYE IS NOT NULL);
      ALTER TABLE CommandesES ADD CONSTRAINT chk_datecnotnull CHECK
(DATE_COMMANDE IS NOT NULL);

      ALTER TABLE CLIENTSES ADD CONSTRAINT chk_socnotnull CHECK (societe IS
NOT NULL);
      ALTER TABLE CLIENTSES ADD CONSTRAINT chk_adrnull CHECK (adresse IS
NOT NULL);
      ALTER TABLE CLIENTSES ADD CONSTRAINT chk_villenotnull CHECK (ville IS
NOT NULL);
      ALTER TABLE CLIENTSES ADD CONSTRAINT chk_cpnotnull CHECK (code_postal
IS NOT NULL);
      ALTER TABLE CLIENTSES ADD CONSTRAINT chk_paysClientnotnull CHECK
(pays IS NOT NULL);
      ALTER TABLE CLIENTSES ADD CONSTRAINT chk_telnotnull CHECK (telephone
IS NOT NULL);

      ALTER TABLE DETAILS_Commandeses ADD CONSTRAINT chk_nocom CHECK
(no_commande IS NOT NULL);
      ALTER TABLE DETAILS_Commandeses ADD CONSTRAINT chk_refpdtnotnull
CHECK (REF_PRODUIT IS NOT NULL);
      ALTER TABLE DETAILS_Commandeses ADD CONSTRAINT chk_unprixnotnull
CHECK (PRIX_UNITAIRE IS NOT NULL);
      ALTER TABLE DETAILS_Commandeses ADD CONSTRAINT chk_quantnotnull CHECK
(quantite IS NOT NULL);
      ALTER TABLE DETAILS_Commandeses ADD CONSTRAINT chk_remisenotnull
CHECK (remise IS NOT NULL);

      ALTER TABLE stockes ADD CONSTRAINT chk_stockrefpdtnotnull CHECK
(ref_produit IS NOT NULL);
      ALTER TABLE stockes ADD CONSTRAINT chk_stockespays CHECK (pays IS NOT
NULL);
```

Clefs étrangères :

```
--FK possibles pour assurer les clés étrangères locales

      alter table details_commandeses add constraint
fk_detailscmdesproduits foreign key (REF_PRODUIT) REFERENCES Produits;
```

```
      alter table details_commandeses add constraint fk_detailsCmdeCmde
foreign key (no_commande) references commandeses;
      alter table stockES add constraint fk_stockESproduits foreign key
(REF_PRODUIT) REFERENCES Produits;
      alter table Produits add constraint fk_ProduitsCategories foreign key
(code_categorie) references categories;
      alter table Commandeses add constraint FK_CommandesesClientses
foreign key (code_client) references clientses;
```

Partie déportée (Triggers)

Pour la table Produits:

Notre trigger est chargé de vérifier si le fournisseur du produit existe bien dans la table fournisseur située sur le site en Europe du Nord en cas d'ajout et de modification de la table Produits. De plus, en cas de suppression, il est chargé de vérifier si il n'existe pas de tuples dans les autres bases référençant le produit qu'on tente de supprimer. Si c'est le cas, la suppression est refusée.

```
create or replace TRIGGER chk_Produits BEFORE INSERT OR UPDATE OR DELETE ON
Produits
FOR EACH ROW
DECLARE
idFourn number; --Id du fournisseur renseigné à vérifier
any_rows_found NUMBER;  --variable indiquant si un produit à supprimer
existe dans une table secondaire

BEGIN
      IF INSERTING OR UPDATING THEN
        SELECT NO_FOURNISSEUR INTO idFourn
        from cpottiez.Fournisseurs@dblinkMain rel
        where rel.NO_FOURNISSEUR = :NEW.NO_FOURNISSEUR;
      END IF;

  IF DELETING THEN

    SELECT count(*) into any_rows_found
    from cpottiez.details_commandesEN@dblinkMain
    where ref_produit = :NEW.REF_PRODUIT;

    IF any_rows_found <> 0 THEN
      raise_application_error(-20002, 'Erreur : le produit à supprimer est
déjà référencé dans la table DétailsCommandes en Europe du Nord');
    end if;

    SELECT count(*) INTO any_rows_found
    FROM cpottiez.details_commandesOI@dblinkMain
    WHERE ref_produit = :NEW.REF_PRODUIT;

    IF any_rows_found <> 0 THEN
      raise_application_error(-20003, 'Erreur : le produit à supprimer est
déjà référencé dans la table DétailsCommandes pour un pays inconnu');
    end if;

    SELECT count(*) INTO any_rows_found
    FROM hcburca.Details_Commandes_AM@dbLinkUS
    WHERE ref_produit = :NEW.REF_PRODUIT;
```

```
    IF any_rows_found <> 0 THEN
      raise_application_error(-20004, 'Erreur : le produit à supprimer est
déjà référencé dans la table DétailsCommandes en Amérique');
    end if;

    SELECT count(*) INTO any_rows_found
    FROM cpottiez.stockEN@dblinkMain
    WHERE ref_produit = :NEW.REF_PRODUIT;

    IF any_rows_found <> 0 THEN
      raise_application_error(-20005, 'Erreur : le produit à supprimer est
déjà référencé dans la table Stocks en Europe du Nord');
    end if;

    SELECT count(*) INTO any_rows_found
    FROM cpottiez.stockOI@dblinkMain
    WHERE ref_produit = :NEW.REF_PRODUIT;

    IF any_rows_found <> 0 THEN
      raise_application_error(-20006, 'Erreur : le produit à supprimer est
déjà référencé dans la table Stocks pour un pays inconnu');
    end if;


    SELECT count(*) INTO any_rows_found
    FROM hcburca.Stock_AM@dbLinkUS
    WHERE ref_produit = :NEW.REF_PRODUIT;

    IF any_rows_found <> 0 THEN
      raise_application_error(-20007, 'Erreur : le produit à supprimer est
déjà référencé dans la table Stocks en Amérique');
    end if;
  END IF;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20002, 'Erreur : tout fournisseur référencé
doit exister dans la table des fournisseurs');
END;
```

Pour la table CommandesES :

Notre trigger sur Commandes ES est chargé de vérifer que la commande que l'on cherche à modifier ou insérer fait bien référence à un employé existant. Pour cela, le trigger regarde si un empoyé comportant cet id existe bien dans la base Employe située aux Amériques

```
create or replace TRIGGER chkInsert_Commandes BEFORE INSERT OR UPDATE ON
CommandesES
FOR EACH ROW
DECLARE
idEmp number;

BEGIN
  SELECT No_employe INTO idEmp
  from hcburca.Employes@dbLinkUS rel
```

```
  where rel.no_employe = :NEW.no_employe;

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20001, 'Erreur : tout employé référencé doit
exister dans la table des employés');
END;
```

e) Droits d'accès

Nous donnons les droits d'accès à notre base aux autres membres du groupe. Nous rappelons que, comme en centralisé, nous donnons les droits de modification, suppression, insertion et consultation à tous les autres sites.

```
/*
  Permissions accordées : STOCKES (lecture/mise à
jour/insertion/suppression depuis les applications externes)
*/
GRANT SELECT, update, insert, delete ON stockES to cpottiez;
GRANT SELECT, update, insert, delete ON stockES to hhamelin;
GRANT SELECT, update, insert, delete ON stockES to zvergne;
GRANT SELECT, update, insert, delete ON stockES to jcharlesni;
GRANT SELECT, update, insert, delete ON stockES to hcburca;

/*
  Permissions accordées : PRODUITS (lecture/mise à
jour/insertion/suppression depuis les applications externes)
*/
GRANT SELECT, update, insert, delete ON produits to cpottiez;
GRANT SELECT, update, insert, delete ON produits to hhamelin;
GRANT SELECT, update, insert, delete ON produits to zvergne;
GRANT SELECT, update, insert, delete ON produits to jcharlesni;
GRANT SELECT, update, insert, delete ON produits to hcburca;

/*
  Permissions accordées : CATEGORIES (lecture/mise à
jour/insertion/suppression depuis les applications externes)
*/
GRANT SELECT, update, insert, delete ON categories to cpottiez;
GRANT SELECT, update, insert, delete ON categories to hhamelin;
GRANT SELECT, update, insert, delete ON categories to zvergne;
GRANT SELECT, update, insert, delete ON categories to jcharlesni;
GRANT SELECT, update, insert, delete ON categories to hcburca;

/*
  Permissions accordées : DETAILS_COMMANDESES (lecture/mise à
jour/insertion/suppression depuis les applications externes)
*/
GRANT SELECT, update, insert, delete ON details_commandeses to cpottiez;
GRANT SELECT, update, insert, delete ON details_commandeses to hhamelin;
GRANT SELECT, update, insert, delete ON details_commandeses to zvergne;
GRANT SELECT, update, insert, delete ON details_commandeses to jcharlesni;
GRANT SELECT, update, insert, delete ON details_commandeses to hcburca;

/*
  Permissions accordées : COMMANDESES (lecture/mise à
jour/insertion/suppression depuis les applications externes)
```

```
*/
GRANT SELECT, update, insert, delete on commandeses TO cpottiez;
GRANT SELECT, update, insert, delete on commandeses TO hhamelin;
GRANT SELECT, update, insert, delete on commandeses TO zvergne;
GRANT SELECT, update, insert, delete on commandeses TO jcharlesni;
GRANT SELECT, update, insert, delete on commandeses TO hcburca;

/*
  Permissions accordées : CLIENTSES (lecture/mise à
jour/insertion/suppression depuis les applications externes)
*/
GRANT SELECT, update, insert, delete on clientses TO cpottiez;
GRANT SELECT, update, insert, delete on clientses TO hhamelin;
GRANT SELECT, update, insert, delete on clientses TO zvergne;
GRANT SELECT, update, insert, delete on clientses TO jcharlesni;
GRANT SELECT, update, insert, delete on clientses TO hcburca;
```

f) Définition de synonymes et de vues pour interrogation de la base comme si elle était en centralisé.

Afin de rendre les changements transparents pour les applications, nous allons désormais ajouter des vues à notre base de données. Les vues correspondront à l'union des fragments dispersés sur les trois sites.

Notez l'utilisation du mot « WHERE » pour indiquer au SGBD le contenu des vues et ainsi lui permettre d'optimiser son plan d'exécution.

```
-- Vue "Stock", création avec WHERE pour optimiser le plan d'exécution
CREATE OR REPLACE VIEW Stock
AS
(SELECT * FROM StockES where StockES.PAYS in
('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-
Marin', 'Vatican',
    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece',
'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
    'Autriche', 'Suisse')
UNION ALL
SELECT * FROM cpottiez.stockEN@dblinkMain where pays in ('Suede',
'Norvege', 'Danemark', 'Finlande', 'Belgique', 'Irlande', 'Pologne',
'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg', 'Pays-Bas')
UNION ALL
SELECT * FROM cpottiez.stockOI@dblinkMain where pays not in ('Antigua-et-
Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela') and pays not in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')
  and pays not in ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican',
    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece',
'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
    'Autriche', 'Suisse')
```

```
UNION ALL
SELECT * FROM hcburca.stock_am@dbLinkUS where pays in ('Antigua-et-
Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela')
);

--Vue 'Clients', création avec WHERE pour optimiser le plan d'exécution

CREATE OR REPLACE VIEW Clients
AS
(SELECT * FROM ClientsES where PAYS in
('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-
Marin', 'Vatican',
    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece',
'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
    'Autriche', 'Suisse')
UNION ALL
SELECT * FROM cpottiez.clientsEN@dblinkMain where pays in ('Suede',
'Norvege', 'Danemark', 'Finlande', 'Belgique', 'Irlande', 'Pologne',
'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg', 'Pays-Bas')
UNION ALL
SELECT * FROM cpottiez.clientsOI@dblinkMain where pays not in ('Antigua-et-
Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela') and pays not in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')
  and pays not in ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican',
    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece',
'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
    'Autriche', 'Suisse')
UNION ALL
SELECT * FROM hcburca.clients_am@dbLinkUS where pays in ('Antigua-et-
Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique','Equateur', 'Etats-Unis', 'Grenade', 'Guatemala',
'Guyana', 'Haiti','Honduras', 'Jamaique','Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie','Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay','Venezuela')
);
```

```
--Vue 'Commandes'
CREATE OR REPLACE VIEW Commandes
AS
(SELECT * FROM Commandeses
UNION ALL
SELECT * FROM cpottiez.commandesEN@dblinkMain
UNION ALL
SELECT * FROM cpottiez.commandesOI@dblinkMain
UNION ALL
SELECT * FROM hcburca.Commandes_AM@dbLinkUS
);

-- Vue 'Details_Commandes'
CREATE OR REPLACE VIEW details_commandes
AS
(SELECT * FROM details_commandeses
UNION ALL
SELECT * FROM cpottiez.details_commandesEN@dblinkMain
UNION ALL
SELECT * FROM cpottiez.details_commandesOI@dblinkMain
UNION ALL
SELECT * FROM hcburca.Details_Commandes_AM@dbLinkUS
);

--Vue fournisseurs
CREATE OR REPLACE VIEW Fournisseurs
AS
(SELECT * FROM cpottiez.fournisseurs@dblinkmain);

-- Vue employés
CREATE OR REPLACE VIEW employes
AS
(SELECT * FROM hcburca.Employes@dbLinkUS
);
```

Nous devons rediriger les requêtes de mise à jour, d'insertion et de suppression en fonction de la localisation des tuples que nous souhaitons modifier. Veuillez noter que nous acceptons qu'une application située sur un autre site puisse modifier des tuples présents sur un site tiers, ce qui signifie que l'application SellIT-ES peut modifier des données du stock d'Amériques alors que ces derniers sont utilisés par l'application SellIT-US. Ceci a été fait pour calquer le comportement de la base centralisée sur la base distribuée. L'ajout de l'interdiction des sites de modifier les tuples des autres sites pourra être ajouté si besoin par une simple modification des triggers suivants.

Pour la vue stock :

Ce trigger est chargé de rediriger les modifications, insertions ou suppressions de tuples vers la base de données contenant ce tuples. Veuillez noter que le cas de modification du pays de l'élément du stock, le trigger déplace le stock vers la base correspondant à l'appartenance du pays nouvellement associé si besoin.

```
CREATE OR REPLACE TRIGGER modify_Stocks INSTEAD OF
  UPDATE OR
  INSERT OR
```

```
  DELETE ON Stock
FOR EACH ROW
BEGIN
IF INSERTING THEN --Insertion à contrôler
      IF (:NEW.pays IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) THEN
  INSERT
  INTO cpottiez.STOCKEN@dblinkMain VALUES
      (
      :new.ref_produit,
      :new.pays,
      :new.unites_stock,
      :new.unites_commandees,
      :new.indisponible
      );
ELSIF (:NEW.pays IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
  INSERT
  INTO stockes VALUES
      (
      :new.ref_produit,
      :new.pays,
      :new.unites_stock,
      :new.unites_commandees,
      :new.indisponible
      );
ELSIF (:NEW.pays IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
  INSERT
  INTO hcburca.stock_am@dbLinkUS VALUES
      (
      :new.ref_produit,
      :new.pays,
      :new.unites_stock,
      :new.unites_commandees,
      :new.indisponible
      );
ELSE
  INSERT
  INTO cpottiez.STOCKOI@dblinkMain VALUES
      (
      :new.ref_produit,
      :new.pays,
      :new.unites_stock,
      :new.unites_commandees,
      :new.indisponible
      );
END IF;
END IF;
```

```
IF UPDATING THEN
  if(:New.pays <> :OLD.pays) then
     --RAISE_APPLICATION_ERROR(-20013,'Attention, vous n''êtes pas autorisé
à changer le pays du stock');
     Delete from stock WHERE ref_produit = :old.ref_produit AND pays =
:old.pays;
       INSERT
      INTO stock VALUES
     (
       :new.ref_produit,
       :new.pays,
       :new.unites_stock,
       :new.unites_commandees,
       :new.indisponible
       );


  elsIF (:NEW.pays IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) THEN
       UPDATE cpottiez.STOCKEN@dblinkMain
       SET ref_produit   = :new.ref_produit,
       PAYS              = :NEW.PAYS,
       UNITES_STOCK      = :new.UNITES_STOCK,
       UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
       INDISPONIBLE      = :new.INDISPONIBLE
       WHERE ref_produit   = :old.ref_produit
       AND pays          = :old.PAYS;
  ELSIF (:NEW.pays IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
       UPDATE stockes
       SET ref_produit   = :new.ref_produit,
       PAYS              = :NEW.PAYS,
       UNITES_STOCK      = :new.UNITES_STOCK,
       UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
       INDISPONIBLE      = :new.INDISPONIBLE
       WHERE ref_produit   = :old.ref_produit
       AND pays          = :old.PAYS;
  ELSIF (:NEW.pays IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
       UPDATE hcburca.stock_am@dbLinkUS
       SET ref_produit   = :new.ref_produit,
       PAYS              = :NEW.PAYS,
       UNITES_STOCK      = :new.UNITES_STOCK,
       UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
       INDISPONIBLE      = :new.INDISPONIBLE
       WHERE ref_produit   = :old.ref_produit
       AND pays          = :old.PAYS;
  ELSE
       UPDATE  cpottiez.STOCKOI@dblinkMain
```

```
      SET ref_produit   = :new.ref_produit,
      PAYS              = :NEW.PAYS,
      UNITES_STOCK      = :new.UNITES_STOCK,
      UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
      INDISPONIBLE      = :new.INDISPONIBLE
      WHERE ref_produit   = :old.ref_produit
      AND pays          = :old.PAYS;
  END IF;
END IF;

IF DELETING THEN
  IF (:old.pays IN ('Suede', 'Norvege', 'Danemark', 'Finlande', 'Belgique',
'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas')) THEN
      DELETE FROM cpottiez.STOCKEN@dblinkMain WHERE ref_produit =
:old.ref_produit AND pays = :old.pays;
  ELSIF (:old.pays IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
      DELETE
      FROM stockes
      WHERE ref_produit = :old.ref_produit AND pays= :old.pays;
  ELSIF (:old.pays IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
      DELETE
      FROM hcburca.stock_am@dbLinkUS
      WHERE ref_produit = :old.ref_produit AND pays= :old.pays;
  ELSE
      DELETE FROM  cpottiez.STOCKOI@dblinkMain WHERE ref_produit =
:old.ref_produit AND pays = :old.pays;
  END IF;
END IF;
END;
```

Pour la vue Commandes :

Ce trigger est chargé de rediriger les suppressions, modifications et insertions de tuples depuis la vue Commandes vers la base de données correspondante en fonction de l'appartenance à un continent du pays du client ayant passé la commande.

```
Create Or Replace TRIGGER MODIFY_COMMANDES
INSTEAD OF DELETE OR INSERT OR UPDATE ON COMMANDES
FOR EACH ROW
DECLARE
  paysclient$ varchar2(15);
BEGIN
  if (inserting or updating) then
    SELECT DISTINCT Pays INTO paysclient$
```

```
    FROM CLIENTS
    WHERE code_client = :new.code_client;
  elsif (deleting) then
    SELECT DISTINCT Pays INTO paysclient$
    FROM CLIENTS
    WHERE code_client = :old.code_client;
  end if;

  if (inserting) then
           if (paysclient$ in ('Antigua-et-Barbuda', 'Argentine',
      'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada',
      'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine',
      'Dominique','Equateur', 'Etats-Unis', 'Grenade', 'Guatemala',
      'Guyana', 'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua',
      'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-
      Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname',
      'Trinite-et-Tobago', 'Uruguay', 'Venezuela')) then

            INSERT INTO hcburca.Commandes_AM@dblinkus VALUES
(:new.no_commande, :new.code_client, :new.no_employe, :new.date_commande,
:new.date_envoi, :new.port);

    elsif (paysclient$ in ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) then

      INSERT INTO commandesES VALUES (:new.no_commande, :new.code_client,
:new.no_employe, :new.date_commande, :new.date_envoi, :new.port);

    elsif (paysclient$ in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) then
      INSERT INTO cpottiez.commandesEN@dblinkmain VALUES (:new.no_commande,
:new.code_client, :new.no_employe, :new.date_commande, :new.date_envoi,
:new.port);
    else
      INSERT INTO cpottiez.commandesOI@dblinkmain VALUES (:new.no_commande,
:new.code_client, :new.no_employe, :new.date_commande, :new.date_envoi,
:new.port);
    end if;

ELSIF (updating) then
    if (paysclient$ in ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine',
'Dominique','Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique','Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie','Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay','Venezuela')) then

      UPDATE hcburca.Commandes_AM@dblinkus SET
      no_commande = :new.no_commande,
      code_client = :new.code_client,
      no_employe = :new.no_employe,
      date_commande = :new.date_commande,
      date_envoi = :new.date_envoi,
```

```
      port = :new.port
      WHERE no_commande = :old.no_commande;

      ELSIF (paysclient$ in ('Espagne', 'Portugal', 'Andorre', 'France',
      'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
      'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
      'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) then

      UPDATE commandesES SET
      no_commande = :new.no_commande,
      code_client = :new.code_client,
      no_employe = :new.no_employe,
      date_commande = :new.date_commande,
      date_envoi = :new.date_envoi,
      port = :new.port
      WHERE no_commande = :old.no_commande;

       ELSIF (paysclient$ in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne',
'Islande','Luxembourg', 'Pays-Bas')) then

      UPDATE cpottiez.commandesEN@dblinkmain SET
      no_commande = :new.no_commande,
      code_client = :new.code_client,
      no_employe = :new.no_employe,
      date_commande = :new.date_commande,
      date_envoi = :new.date_envoi,
      port = :new.port
    WHERE no_commande = :old.no_commande;
    ELSE

      UPDATE cpottiez.commandesOI@dblinkmain SET
no_commande = :new.no_commande,
code_client = :new.code_client,
no_employe = :new.no_employe,
date_commande = :new.date_commande,
date_envoi = :new.date_envoi,
port = :new.port
      WHERE no_commande = :old.no_commande;
    end if;

ELSIF (deleting) then
    if (paysclient$ in ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine',
'Dominique','Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) then

      DELETE FROM hcburca.Commandes_AM@dblinkus WHERE no_commande =
:old.no_commande;

    elsif (paysclient$ in ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican','Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie','Autriche', 'Suisse')) then
```

```
      DELETE FROM commandesES WHERE no_commande = :old.no_commande;

    elsif (paysclient$ in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
                           'Luxembourg', 'Pays-Bas')) then
      DELETE FROM cpottiez.commandesEN@dblinkmain WHERE no_commande =
:old.no_commande;

    else
      DELETE FROM cpottiez.commandesOI@dblinkmain WHERE no_commande =
:old.no_commande;
    end if;
  end if;

EXCEPTION
  WHEN no_data_found then
  raise_application_error(-20011, 'Erreur : le client indiqué est
inconnu.');
END;
```

Pour la vue Details_Commandes :

Ce trigger est chargé de rediriger les suppressions, modifications et insertions de tuples depuis la vue Details_Commandes vers la base de données correspondante en fonction de l'appartenance à un continent du pays du client ayant passé la commande référencée dans le tuple.

```
create or replace TRIGGER modify_DetailsCommandes INSTEAD OF
  UPDATE OR
  INSERT OR
  DELETE ON DETAILS_COMMANDES FOR EACH ROW
  DECLARE
  paysTest VARCHAR2(24);
  BEGIN
  if(updating or inserting) then
    SELECT pays into paystest
    FROM CLIENTS natural join COMMANDES
    where no_commande = :new.no_commande;
    end if;

  IF INSERTING THEN         --Insertion à contrôler

    IF (paysTest   IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) THEN
      INSERT
      INTO cpottiez.DETAILS_COMMANDESEN@dblinkmain VALUES
        (
          :new.no_commande,
          :new.ref_produit,
          :new.prix_unitaire,
          :new.quantite,
          :new.remise
        );
    ELSIF (paysTest IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
```

```
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
      INSERT
      INTO details_commandesES VALUES
        (
          :new.no_commande,
          :new.ref_produit,
          :new.prix_unitaire,
          :new.quantite,
          :new.remise
        );
    ELSIF (paysTest IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
      INSERT
      INTO hcburca.Details_Commandes_AM@dbLinkUS VALUES
        (
          :new.no_commande,
          :new.ref_produit,
          :new.prix_unitaire,
          :new.quantite,
          :new.remise
        );
    ELSE
      INSERT
      INTO cpottiez.details_commandesOI@dblinkmain VALUES
        (
          :new.no_commande,
          :new.ref_produit,
          :new.prix_unitaire,
          :new.quantite,
          :new.remise
        );
    END IF;
  END IF;
  IF UPDATING THEN

    IF(:old.ref_produit= :new.ref_produit AND :old.NO_COMMANDE=
:new.no_commande) then
      IF (paysTest IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) THEN
        UPDATE cpottiez.DETAILS_COMMANDESEN@dblinkmain
        SET ref_produit   = :new.ref_produit,
          NO_COMMANDE     = :new.no_commande,
          PRIX_UNITAIRE   = :new.PRIX_UNITAIRE,
          QUANTITE        = :new.QUANTITE,
          REMISE          = :new.REMISE
        WHERE ref_produit = :old.ref_produit
        AND no_commande   = :old.no_commande;
      ELSIF (paysTest IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
```

```
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
        UPDATE details_commandesES
        SET ref_produit   = :new.ref_produit,
          NO_COMMANDE     = :new.no_commande,
          PRIX_UNITAIRE   = :new.PRIX_UNITAIRE,
          QUANTITE        = :new.QUANTITE,
          REMISE          = :new.REMISE
        WHERE ref_produit = :old.ref_produit
        AND no_commande   = :old.no_commande;
      ELSIF (paysTest IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
        UPDATE hcburca.Details_Commandes_AM@dbLinkUS
        SET ref_produit   = :new.ref_produit,
          NO_COMMANDE     = :new.no_commande,
          PRIX_UNITAIRE   = :new.PRIX_UNITAIRE,
          QUANTITE        = :new.QUANTITE,
          REMISE          = :new.REMISE
        WHERE ref_produit = :old.ref_produit
        AND no_commande   = :old.no_commande;
      ELSE
        UPDATE cpottiez.details_commandesOI@dblinkmain
        SET ref_produit   = :new.ref_produit,
          NO_COMMANDE     = :new.no_commande,
          PRIX_UNITAIRE   = :new.PRIX_UNITAIRE,
          QUANTITE        = :new.QUANTITE,
          REMISE          = :new.REMISE
        WHERE ref_produit = :old.ref_produit
        AND no_commande   = :old.no_commande;
      END IF;
    END IF;
  END IF;
  IF DELETING THEN

--    DBMS_OUTPUT.PUT_LINE(:old.no_commande||' - ' || :old.ref_produit);
    SELECT pays into paysTest
    FROM CLIENTS natural join COMMANDES
    where no_commande = :old.no_commande;

   IF (paysTest  IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) THEN
      DELETE
      FROM cpottiez.DETAILS_COMMANDESEN@dblinkmain
      WHERE ref_produit = :old.ref_produit
      AND no_commande   = :old.no_commande;

      ELSIF (paysTest IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
```

```
      DELETE
      FROM details_commandesES
      WHERE ref_produit = :old.ref_produit
      AND no_commande   = :old.no_commande;
    ELSIF (paysTest IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
      DELETE
      FROM hcburca.Details_Commandes_AM@dbLinkUS
      WHERE ref_produit = :old.ref_produit
      AND no_commande   = :old.no_commande;
    ELSE
      DELETE
      FROM cpottiez.details_commandesOI@dblinkmain
      WHERE ref_produit = :old.ref_produit
      AND no_commande   = :old.no_commande;
    END IF;
  END IF;

  exception
    when no_data_found then
      raise_application_error(-20012, 'La commande indiquée est inconnue');
END;
```

Pour la vue Clients :

Ce trigger est chargé de rediriger les suppressions, modifications et insertions de tuples depuis la vue Clients vers la base de données correspondante en fonction de l'appartenance à un continent du pays du client. Notez que on interdit le changement de pays du Clients, l'opération de transfert des données associées à ce client et situées dans d'autres tables étant jugé trop complexe.

```
create or replace TRIGGER modify_Clients INSTEAD OF UPDATE OR INSERT or
delete ON Clients
FOR EACH ROW
BEGIN
IF INSERTING then

IF (:NEW.pays in ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican',
    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece',
'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
    'Autriche', 'Suisse')) THEN

      INSERT INTO Clientses
      VALUES (:new.code_client,:new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone, :new.fax);

   ELSIF (:new.pays in  ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil',
```

```
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela')) then
          INSERT INTO hcburca.Clients_am@dbLinkUS
         VALUES (:new.code_client,:new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone, :new.fax);

      ELSIF  (:new.pays in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas'))
        then
         INSERT INTO cpottiez.ClientsEN@dblinkMain
         VALUES (:new.code_client,:new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone, :new.fax);

      else
         INSERT INTO cpottiez.ClientsOI@dblinkMain
          VALUES (:new.code_client,:new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone, :new.fax);
  end if;
      END IF;

  if UPDATING THEN

  if(:New.pays <> :OLD.pays) then
     RAISE_APPLICATION_ERROR(-20010,'Attention, vous n''êtes pas autorisé à
changer le pays de l''entreprise');
    end if;
     IF (:NEW.pays in ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican',
    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece',
'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
    'Autriche', 'Suisse')) THEN
    update clientses
        set
        code_client = :new.code_client,
        societe = :NEW.societe,
        adresse = :new.adresse,
        ville = :new.ville,
        code_postal = :new.code_postal,
        pays = :new.pays,
        telephone = :new.telephone,
        fax = :new.fax
      WHERE code_client = :old.code_client;
      ELSIF  (:new.pays in ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
```

```
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela')) then
        update hcburca.clients_am@dblinkUS
        set
        code_client = :new.code_client,
        societe = :NEW.societe,
        adresse = :new.adresse,
        ville = :new.ville,
        code_postal = :new.code_postal,
        pays = :new.pays,
        telephone = :new.telephone,
        fax = :new.fax
      WHERE code_client = :old.code_client;
      ELSIF  (:new.pays in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande',
      'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande', 'Luxembourg',
'Pays-Bas')) then
       update cpottiez.clientsen@dbLinkMain
        set
        code_client = :new.code_client,
        societe = :NEW.societe,
        adresse = :new.adresse,
        ville = :new.ville,
        code_postal = :new.code_postal,
        pays = :new.pays,
        telephone = :new.telephone,
        fax = :new.fax
      WHERE code_client = :old.code_client;
      else
        update cpottiez.clientsoi@dbLinkMain
        set
        code_client = :new.code_client,
        societe = :NEW.societe,
        adresse = :new.adresse,
        ville = :new.ville,
        code_postal = :new.code_postal,
        pays = :new.pays,
        telephone = :new.telephone,
        fax = :new.fax
      WHERE code_client = :old.code_client;
      end if;
  end if;

  IF DELETING THEN

    IF  (:old.pays in ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican',
    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece',
'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
    'Autriche', 'Suisse')) THEN
    delete from clientses where code_client = :old.code_client;

    ELSIF (:old.pays in  ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
```

```
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela')) then
       delete from hcburca.clients_AM@dbLinkUS where code_client =
:old.code_client;
   ELSIF (:old.pays in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique',
   'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas'))then
        delete from cpottiez.clientsEN@dbLinkMain where code_client =
:old.code_client;
    else
        delete from cpottiez.clientsOI@dbLinkMain where code_client=
:old.code_client;
    end if;
  END IF;
     end;
```

Ainsi, ces triggers sur les vues Commandes, DétailsCommandes, Stock et Clients garantissent une intégrité lors de l'insertion, modification et suppression des données lorsqu'on effectue des opérations dessus.

h) Nettoyages éventuels

Nous supprimons les différents tuples ajoutés dans la table afin de tester son bon fonctionnement.

i) Tests de vérification du bon fonctionnement

Pour des raisons de lisibilité, nous avons placé l'ensemble de nos tests en Annexe, à la fin de ce document. En effet, ces derniers font cinq pages.

3 Amériques

a) Binôme responsable

Le binôme responsable du site d'Amérique est le binôme B3414, constitué de Horia Burca et de Julien Charles-Nicolas.

b) Création des liens entre les bases

Dans un premier temps, nous créons les vers les bases de données des autres sites. Toujours en utlisant la machine de départ pour héberger le site de l'Europe du Nord, DB1.

Nous appelons tplink le lien vers la base de données d'Europe du Nord et linkES le lien vers la base de données de l'Europe du Sud.

```
CREATE DATABASE LINK tplink CONNECT TO hcburca IDENTIFIED BY mdporacle
USING 'DB1';
CREATE DATABASE LINK linkES CONNECT TO hcburca IDENTIFIED BY mdporacle USING
'DB3';
```

c) Création des tables & Peuplement des tables

Nous procédons maintenant à la création des différentes tables sur notre base et nous les remplissons aussitôt grâce à la commande CREATE TABLE AS.
Le remplissage se fait donc en sélectionnant les données en fonction des prédicats définis précédemment.

```
CREATE table Clients_AM as (
  Select * from Ryori.Clients@tplink
  where Pays in ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade',
'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela')
);


CREATE TABLE Employes as (
  Select * from Ryori.Employes@tplink
);


CREATE table Commandes_AM as(
  Select com.* from Ryori.Commandes@tplink com, Ryori.Clients@tplink cli
  where com.code_client = cli.code_client and cli.pays in ('Antigua-et-
Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela')
);

CREATE table Details_Commandes_AM as(
  Select detcom.* from Ryori.Details_Commandes@tplink detcom, Commandes_AM
com
  where detcom.no_commande = com.no_commande
);
```

```
CREATE table Stock_AM as(
  Select * from Ryori.Stock@tplink
  where Pays in ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade',
'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela')
      );
```

d) Contraintes d'intégrité

Nous devons désormais appliquer des contraintes d'intégrités à nos tables. Ces contraintes doivent coller aux contraintes présentes sur la base originale. Les contraintes de la base originale sont de type « clé primaire », « clé étrangère » L'écriture des contraintes peut donc se séparer en deux parties : la partie « locale » qui ne concerne que les tuples présents sur le site local et la partie déportée qui, elle, fait intervenir des triggers nécessaires pour regarder et éditer le contenu des autres bases de données pour maintenir la cohérence des données en cas d'insertion, de modification et suppression.

Voici la partie locale :

```
-- Contraintes primary key

ALTER TABLE clients_am ADD CONSTRAINT pkClient PRIMARY KEY(code_client);
ALTER TABLE commandes_am ADD CONSTRAINT pkCommande PRIMARY KEY
(no_commande);
ALTER TABLE employes ADD CONSTRAINT pkEmploye PRIMARY KEY (no_employe);
ALTER TABLE details_commandes_am ADD CONSTRAINT pkDetails PRIMARY KEY
(no_commande, ref_produit);
ALTER TABLE stock_am ADD CONSTRAINT pkStock PRIMARY KEY (ref_produit,
pays);

-- Contraintes foreign key

ALTER TABLE employes ADD CONSTRAINT fkEmployeEmploye FOREIGN KEY
(rend_Compte) REFERENCES Employes(no_employe);
ALTER TABLE commandes_am ADD CONSTRAINT fkCommandesClients FOREIGN KEY
(code_client) REFERENCES Clients_am(code_client);
ALTER TABLE commandes_am ADD CONSTRAINT fkCommandesEmployes FOREIGN KEY
(no_employe) REFERENCES Employes(no_employe);
      ALTER TABLE details_commandes_am ADD CONSTRAINT
      fkDetailsCommandesCommandes FOREIGN KEY (no_commande) REFERENCES
      Commandes_am(no_commande) ON DELETE CASCADE;
```

Partie déportée (Triggers) :

```
-- Trigger contraintes foreign key pour les references non-locales

-- details_commandes_am -> produits
CREATE OR REPLACE TRIGGER FKDETAILSCOMMANDESPRODUITS
BEFORE INSERT OR UPDATE ON details_commandes_am
FOR EACH ROW
DECLARE

  refprod$ details_commandes_am.ref_produit%TYPE;

BEGIN
  Select ref_produit into refprod$
  From lpoisse.produits@linkES
  Where ref_produit = :New.ref_produit;

  EXCEPTION
      WHEN NO_DATA_FOUND THEN
        Raise_application_error(-20001, 'Le produit n''existe pas');
END;
/

-- stock_am -> produits
CREATE OR REPLACE TRIGGER FKSTOCKPRODUITS
BEFORE INSERT OR UPDATE ON stock_am
FOR EACH ROW
DECLARE

  refprod$ stock_am.ref_produit%TYPE;

BEGIN
  Select ref_produit into refprod$
  From lpoisse.produits@linkES
  Where ref_produit = :New.ref_produit;

  EXCEPTION
      WHEN NO_DATA_FOUND THEN
        Raise_application_error(-20001, 'Le produit n''existe pas');
END;
      /
```

Ces deux triggers vont en fait vérifier si le produit existe bien dans la table produits du site Europe du Sud.

e) Droits d'accès

Nous donnons les droits d'accès à notre base aux autres membres du groupe. Nous rappelons que, comme en centralisé, nous donnons les droits de modification, suppression, insertion et consultation à tous les autres sites.

```
-- toutes les droits pour autres sites pour la table stock
```

```
GRANT select, update, insert, delete on stock_am to lpoisse;
GRANT select, update, insert, delete on stock_am to zvergne;
Grant select, update, insert, delete on stock_am to hhamelin;
Grant select, update, insert, delete on stock_am to cpottiez;

-- toutes les droits autres sites pour la table employes

GRANT select, update, insert, delete on employes to lpoisse;
GRANT select, update, insert, delete on employes to zvergne;
Grant select, update, insert, delete on employes to hhamelin;
Grant select, update, insert on employes to cpottiez;

-- toutes les droits autres sites pour la table clients_am

GRANT select, update, insert, delete on clients_am to lpoisse;
GRANT select, update, insert, delete on clients_am to zvergne;
Grant select, update, insert, delete on clients_am to hhamelin;
Grant select, update, insert, delete on clients_am to cpottiez;

-- toutes les droits autres sites pour la table commandes_am

GRANT select, update, insert, delete on commandes_am to lpoisse;
GRANT select, update, insert, delete on commandes_am to zvergne;
Grant select, update, insert, delete on commandes_am to hhamelin;
Grant select, update, insert, delete on commandes_am to cpottiez;

-- toutes les droits autres sites pour la table details_commandes_am

GRANT select, update, insert, delete on details_commandes_am to lpoisse;
GRANT select, update, insert, delete on details_commandes_am to zvergne;
Grant select, update, insert, delete on details_commandes_am to hhamelin;
Grant select, update, insert, delete on details_commandes_am to cpottiez;
```

f) Définition de synonymes et de vues pour interrogation de la base comme si elle était en centralisé

Afin de rendre les changements transparents pour les applications, nous allons désormais ajouter des vues à notre base de données. Les vues correspondront à l'union des fragments dispersés sur les trois sites. Notez l'utilisation du mot « WHERE » pour indiquer au SGBD le contenu des vues et ainsi lui permettre d'optimiser son plan d'exécution.

```
-- Vue 'CLIENTS'
CREATE OR REPLACE VIEW CLIENTS AS
  SELECT *
  FROM CLIENTS_AM where pays in ('Antigua-et-Barbuda', 'Argentine',
'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
```

```
  'Venezuela')
  UNION ALL
  SELECT *
  FROM lpoisse.clientsES@linkES where PAYS in
('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-
Marin', 'Vatican',
    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece',
'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
    'Autriche', 'Suisse')
  UNION ALL
  SELECT *
  FROM cpottiez.clientsEN@tplink where pays in ('Suede', 'Norvege',
'Danemark', 'Finlande', 'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni',
'Allemagne', 'Islande', 'Luxembourg', 'Pays-Bas')
  UNION ALL
  SELECT *
  FROM cpottiez.clientsOI@tplink where pays not in ('Antigua-et-Barbuda',
'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela') and pays not in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')
  and pays not in ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican',
    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece',
'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
    'Autriche', 'Suisse');

-- Vue 'COMMANDES'
CREATE OR REPLACE VIEW COMMANDES AS
  SELECT *
  FROM COMMANDES_AM
  UNION ALL
  SELECT *
  FROM lpoisse.COMMANDESES@linkES
  UNION ALL
  SELECT *
  FROM cpottiez.COMMANDESEN@tplink
  UNION ALL
  SELECT *
  FROM cpottiez.COMMANDESOI@tplink ;

-- Vue 'DETAILS_COMMANDES'
CREATE OR REPLACE VIEW DETAILS_COMMANDES AS
  SELECT *
  FROM Details_COMMANDES_AM
  UNION ALL
  SELECT *
  FROM lpoisse.Details_COMMANDESES@linkES
  UNION ALL
  SELECT *
```

```
  FROM cpottiez.Details_COMMANDESEN@tplink
  UNION ALL
  SELECT *
  FROM cpottiez.Details_COMMANDESOI@tplink;

-- Vue 'Stock'
CREATE OR REPLACE VIEW STOCK AS
   SELECT *
   FROM stock_AM where pays in ('Antigua-et-Barbuda', 'Argentine',
'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela')
   UNION ALL
   SELECT *
   FROM lpoisse.stockES@linkES where PAYS in
('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-
Marin', 'Vatican',
    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece',
'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
    'Autriche', 'Suisse')
   UNION ALL
   SELECT *
   FROM cpottiez.stockEN@tplink where pays in ('Suede', 'Norvege',
'Danemark', 'Finlande', 'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni',
'Allemagne', 'Islande', 'Luxembourg', 'Pays-Bas')
   UNION ALL
   SELECT *
   FROM cpottiez.stockOI@tplink where pays not in ('Antigua-et-Barbuda',
'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil',
  'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba', 'Republique
dominicaine', 'Dominique',
  'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique',
  'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-
et-Nieves', 'Sainte-Lucie',
  'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-
Tobago', 'Uruguay',
  'Venezuela') and pays not in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')
  and pays not in ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar',
'Italie', 'Saint-Marin', 'Vatican',
    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece',
'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
    'Autriche', 'Suisse');

-- Vue 'Produits'
CREATE OR REPLACE VIEW PRODUITS
AS
  (SELECT *
   FROM lpoisse.produits@LinkES
```

```
  );

-- Vue 'Fournisseurs'
CREATE OR REPLACE VIEW FOURNISSEURS
AS
   (SELECT *
    FROM cpottiez.fournisseurs@tplink
          );
```

Nous devons rediriger les requêtes de mise à jour, d'insertion et de suppression en fonction de la localisation des tuples que nous souhaitons modifier. Veuillez noter que nous acceptons qu'une application située sur un autre site puisse modifier des tuples présents sur un site tiers, ce qui signifie que l'application SelllT-ES peut modifier des données du stock d'Amériques alors que ces derniers sont utilisés par l'application SelllT-US. Ceci a été fait pour calquer le comportement de la base centralisée sur la base distribuée. L'ajout de l'interdiction des sites de modifier les tuples des autres sites pourra être ajouté si besoin par une simple modification des triggers suivants.

Pour la vue stock :

Ce trigger est chargé de rediriger les modifications, insertions ou suppressions de tuples vers la base de données contenant ce tuples. Veuillez noter que le cas de modification du pays de l'élément du stock, le trigger déplace le stock vers la base correspondant à l'appartenance du pays nouvellement associé si besoin.

```
create or replace TRIGGER MODIFY_STOCK
INSTEAD OF DELETE OR INSERT OR UPDATE ON STOCK
BEGIN

  IF INSERTING THEN --Insertion à contrôler
    IF (:NEW.pays IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) THEN
      INSERT
      INTO cpottiez.stocken@tplink VALUES
      (
        :new.ref_produit,
        :new.pays,
        :new.unites_stock,
        :new.unites_commandees,
        :new.indisponible
      );
    ELSIF (:NEW.pays IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
      INSERT
      INTO lpoisse.stockes@linkES VALUES
      (
        :new.ref_produit,
        :new.pays,
        :new.unites_stock,
        :new.unites_commandees,
        :new.indisponible
      );
```

```
    ELSIF (:NEW.pays IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
      INSERT
      INTO stock_am VALUES
      (
        :new.ref_produit,
        :new.pays,
        :new.unites_stock,
        :new.unites_commandees,
        :new.indisponible
      );
    ELSE
      INSERT
      INTO cpottiez.stockOI@tplink VALUES
      (
        :new.ref_produit,
        :new.pays,
        :new.unites_stock,
        :new.unites_commandees,
        :new.indisponible
      );
    END IF;
  END IF;

  IF UPDATING THEN -- Modification à contrôler : on ne peut pas modifier
les données d'un stock non local !
    IF (:NEW.pays IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) THEN
      UPDATE cpottiez.stocken@tplink
      SET ref_produit   = :new.ref_produit,
        PAYS            = :NEW.PAYS,
        UNITES_STOCK    = :new.UNITES_STOCK,
        UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
        INDISPONIBLE    = :new.INDISPONIBLE
      WHERE ref_produit   = :old.ref_produit
      AND pays          = :old.PAYS;
    ELSIF (:NEW.pays IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
      UPDATE lpoisse.stockes@linkES
      SET ref_produit   = :new.ref_produit,
        PAYS            = :NEW.PAYS,
        UNITES_STOCK    = :new.UNITES_STOCK,
        UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
        INDISPONIBLE    = :new.INDISPONIBLE
      WHERE ref_produit   = :old.ref_produit
      AND pays          = :old.PAYS;
    ELSIF (:NEW.pays IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
```

```
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
      UPDATE stock_am
      SET ref_produit   = :new.ref_produit,
        PAYS            = :NEW.PAYS,
        UNITES_STOCK    = :new.UNITES_STOCK,
        UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
        INDISPONIBLE    = :new.INDISPONIBLE
      WHERE ref_produit   = :old.ref_produit
      AND pays          = :old.PAYS;
    ELSE
      UPDATE cpottiez.stockOI@tplink
      SET ref_produit   = :new.ref_produit,
        PAYS            = :NEW.PAYS,
        UNITES_STOCK    = :new.UNITES_STOCK,
        UNITES_COMMANDEES = :new.UNITES_COMMANDEES,
        INDISPONIBLE    = :new.INDISPONIBLE
      WHERE ref_produit   = :old.ref_produit
      AND pays          = :old.PAYS;
    END IF;
  END IF;

  IF DELETING THEN --Suppression à vérifier ; on ne peut pas supprimer un
stock ne faisant pas partie de la région du site (gestion du stock LOCAL
seulement)
    IF (:old.pays IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) THEN
        DELETE FROM cpottiez.stocken@tplink WHERE ref_produit =
:old.ref_produit AND pays = :old.pays;
    ELSIF (:old.pays IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
        DELETE FROM lpoisse.stockes@linkES WHERE ref_produit =
:old.ref_produit AND pays = :old.pays;
    ELSIF (:old.pays IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
      DELETE FROM stock_am WHERE ref_produit = :old.ref_produit AND pays =
:old.pays;
    ELSE
      DELETE FROM cpottiez.stockOI@tplink WHERE ref_produit =
:old.ref_produit AND pays = :old.pays;
    END IF;
  END IF;
      END;
```

Pour la vue Commandes :

Ce trigger est chargé de rediriger les suppressions, modifications et insertions de tuples depuis la vue Commandes vers la base de données correspondante en fonction de l'appartenance à un continent du pays du client ayant passé la commande.

```
-- trigger pour modifier la vue commandes
CREATE OR REPLACE TRIGGER MODIFY_COMMANDES
INSTEAD OF DELETE OR INSERT OR UPDATE ON COMMANDES
FOR EACH ROW
DECLARE
  paysclient$ clients_am.pays%TYPE;
BEGIN
  if (inserting or updating) then
    SELECT DISTINCT Pays INTO paysclient$
    FROM CLIENTS
    WHERE code_client = :new.code_client;
  elsif (deleting) then
    SELECT DISTINCT Pays INTO paysclient$
    FROM CLIENTS
    WHERE code_client = :old.code_client;
  end if;

  if (inserting) then
    if (paysclient$ in ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil',
                        'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba',
'Republique dominicaine', 'Dominique',
                        'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala',
'Guyana', 'Haiti','Honduras', 'Jamaique',
                        'Mexique', 'Nicaragua', 'Panama', 'Paraguay',
'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie',
                        'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay',
                        'Venezuela')) then
      INSERT INTO Commandes_AM VALUES (:new.no_commande, :new.code_client,
:new.no_employe, :new.date_commande, :new.date_envoi, :new.port);
    elsif (paysclient$ in ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican',
                           'Malte', 'Albanie', 'Bosnie-Herzegovine',
'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie',
'Bulgarie',
                           'Autriche', 'Suisse')) then
      INSERT INTO lpoisse.commandesES@linkES VALUES (:new.no_commande,
:new.code_client, :new.no_employe, :new.date_commande, :new.date_envoi,
:new.port);
    elsif (paysclient$ in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
                           'Luxembourg', 'Pays-Bas')) then
      INSERT INTO cpottiez.commandesEN@tplink VALUES (:new.no_commande,
:new.code_client, :new.no_employe, :new.date_commande, :new.date_envoi,
:new.port);
    else
      INSERT INTO cpottiez.commandesOI@tplink VALUES (:new.no_commande,
:new.code_client, :new.no_employe, :new.date_commande, :new.date_envoi,
:new.port);
    end if;
  elsif (updating) then
```

```
    if (paysclient$ in ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil',
                        'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba',
'Republique dominicaine', 'Dominique',
                        'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala',
'Guyana', 'Haiti','Honduras', 'Jamaique',
                        'Mexique', 'Nicaragua', 'Panama', 'Paraguay',
'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie',
                        'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay',
                        'Venezuela')) then
      UPDATE Commandes_AM SET no_commande = :new.no_commande, code_client =
:new.code_client, no_employe = :new.no_employe,
                               date_commande = :new.date_commande,
date_envoi = :new.date_envoi, port = :new.port
                            WHERE no_commande = :old.no_commande;
    elsif (paysclient$ in ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican',
                               'Malte', 'Albanie', 'Bosnie-Herzegovine',
'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie',
'Bulgarie',
                               'Autriche', 'Suisse')) then
      UPDATE lpoisse.commandesES@linkES SET no_commande = :new.no_commande,
code_client = :new.code_client, no_employe = :new.no_employe,
                               date_commande = :new.date_commande,
date_envoi = :new.date_envoi, port = :new.port
                            WHERE no_commande = :old.no_commande;
    elsif (paysclient$ in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
                               'Luxembourg', 'Pays-Bas')) then
      UPDATE cpottiez.commandesEN@tplink SET no_commande =
:new.no_commande, code_client = :new.code_client, no_employe =
:new.no_employe,
                               date_commande = :new.date_commande,
date_envoi = :new.date_envoi, port = :new.port
                            WHERE no_commande = :old.no_commande;
    else
      UPDATE cpottiez.commandesOI@tplink SET no_commande =
:new.no_commande, code_client = :new.code_client, no_employe =
:new.no_employe,
                               date_commande = :new.date_commande,
date_envoi = :new.date_envoi, port = :new.port
                            WHERE no_commande = :old.no_commande;
    end if;
  elsif (deleting) then
    if (paysclient$ in ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil',
                        'Canada', 'Chili', 'Colombie','Costa Rica', 'Cuba',
'Republique dominicaine', 'Dominique',
                        'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala',
'Guyana', 'Haiti','Honduras', 'Jamaique',
                        'Mexique', 'Nicaragua', 'Panama', 'Paraguay',
'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie',
                        'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay',
                        'Venezuela')) then
      DELETE FROM commandes_AM WHERE no_commande = :old.no_commande;
```

```
    elsif (paysclient$ in ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican',
                           'Malte', 'Albanie', 'Bosnie-Herzegovine',
'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie',
'Bulgarie',
                           'Autriche', 'Suisse')) then
      DELETE FROM lpoisse.commandesES@linkES WHERE no_commande =
:old.no_commande;
    elsif (paysclient$ in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
                           'Luxembourg', 'Pays-Bas')) then
      DELETE FROM cpottiez.commandesEN@tplink WHERE no_commande =
:old.no_commande;
    else
      DELETE FROM cpottiez.commandesOI@tplink WHERE no_commande =
:old.no_commande;
    end if;
  end if;

END;
/
```

Pour la vue Details_Commandes :

Ce trigger est chargé de rediriger les suppressions, modifications et insertions de tuples depuis la vue Details_Commandes vers la base de données correspondante en fonction de l'appartenance à un continent du pays du client ayant passé la commande référencée dans le tuple.

```
create or replace TRIGGER MODIFY_DETAILSCOMMANDES
INSTEAD OF DELETE OR INSERT OR UPDATE ON DETAILS_COMMANDES
DECLARE

  paysTest VARCHAR2(24);

BEGIN
  if(updating or inserting) then
    SELECT pays into paystest
    FROM CLIENTS natural join COMMANDES
    where no_commande = :new.no_commande;
  elsif (deleting) then
    SELECT pays into paysTest
    FROM CLIENTS natural join COMMANDES
    where no_commande = :old.no_commande;
  end if;

  IF INSERTING THEN          --Insertion à contrôler
    dbms_output.put_line('inserting ' ||' - ' || :new.ref_produit);
    IF (paysTest   IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) THEN
      INSERT
      INTO cpottiez.DETAILS_COMMANDESEN@tplink VALUES
        (
          :new.no_commande,
          :new.ref_produit,
          :new.prix_unitaire,
          :new.quantite,
```

```
          :new.remise
        );
    ELSIF (paysTest IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
      INSERT
      INTO lpoisse.details_commandesES@linkES VALUES
        (
          :new.no_commande,
          :new.ref_produit,
          :new.prix_unitaire,
          :new.quantite,
          :new.remise
        );
    ELSIF (paysTest IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
      INSERT
      INTO Details_Commandes_AM VALUES
        (
          :new.no_commande,
          :new.ref_produit,
          :new.prix_unitaire,
          :new.quantite,
          :new.remise
        );
    ELSE
      INSERT
      INTO cpottiez.details_commandesOI@tplink VALUES
        (
          :new.no_commande,
          :new.ref_produit,
          :new.prix_unitaire,
          :new.quantite,
          :new.remise
        );
    END IF;
  END IF;
  IF UPDATING THEN -- Modification à contrôler : on ne peut pas modifier
les données d'un stock non local !
  dbms_output.put_line('updating ' || :old.ref_produit||' - ' ||
:new.ref_produit);
    IF(:old.ref_produit= :new.ref_produit AND :old.NO_COMMANDE=
:new.no_commande) then
      IF (paysTest IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) THEN
        UPDATE cpottiez.DETAILS_COMMANDESEN@tplink
        SET ref_produit   = :new.ref_produit,
          NO_COMMANDE     = :new.no_commande,
          PRIX_UNITAIRE   = :new.PRIX_UNITAIRE,
          QUANTITE        = :new.QUANTITE,
```

```
          REMISE          = :new.REMISE
        WHERE ref_produit = :old.ref_produit
        AND no_commande   = :old.no_commande;
      ELSIF (paysTest IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
        UPDATE lpoisse.details_commandesES@linkES
        SET ref_produit   = :new.ref_produit,
          NO_COMMANDE     = :new.no_commande,
          PRIX_UNITAIRE   = :new.PRIX_UNITAIRE,
          QUANTITE        = :new.QUANTITE,
          REMISE          = :new.REMISE
        WHERE ref_produit = :old.ref_produit
        AND no_commande   = :old.no_commande;
      ELSIF (paysTest IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
        UPDATE Details_Commandes_AM
        SET ref_produit   = :new.ref_produit,
          NO_COMMANDE     = :new.no_commande,
          PRIX_UNITAIRE   = :new.PRIX_UNITAIRE,
          QUANTITE        = :new.QUANTITE,
          REMISE          = :new.REMISE
        WHERE ref_produit = :old.ref_produit
        AND no_commande   = :old.no_commande;
      ELSE
        UPDATE cpottiez.details_commandesOI@tplink
        SET ref_produit   = :new.ref_produit,
          NO_COMMANDE     = :new.no_commande,
          PRIX_UNITAIRE   = :new.PRIX_UNITAIRE,
          QUANTITE        = :new.QUANTITE,
          REMISE          = :new.REMISE
        WHERE ref_produit = :old.ref_produit
        AND no_commande   = :old.no_commande;
      END IF;
    END IF;
  END IF;
  IF DELETING THEN
  dbms_output.put_line('deleting ' || :old.ref_produit);
   IF (paysTest  IN ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) THEN
      DELETE
      FROM cpottiez.DETAILS_COMMANDESEN@tplink
      WHERE ref_produit = :old.ref_produit
      AND no_commande   = :old.no_commande;

      ELSIF (paysTest IN ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie',
'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie', 'Autriche', 'Suisse')) THEN
```

```
      DELETE
      FROM lpoisse.details_commandesES@linkES
      WHERE ref_produit = :old.ref_produit
      AND no_commande   = :old.no_commande;
    ELSIF (paysTest IN('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili',
'Colombie','Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique',
'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana',
'Haiti','Honduras', 'Jamaique', 'Mexique', 'Nicaragua', 'Panama',
'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-
Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Venezuela')) THEN
      DELETE
      FROM Details_Commandes_AM
      WHERE ref_produit = :old.ref_produit
      AND no_commande   = :old.no_commande;
    ELSE
      DELETE
      FROM cpottiez.details_commandesOI@tplink
      WHERE ref_produit = :old.ref_produit
      AND no_commande   = :old.no_commande;
    END IF;
  END IF;
      END;
```

Pour la vue Clients :

Ce trigger est chargé de rediriger les suppressions, modifications et insertions de tuples depuis la vue Clients vers la base de données correspondante en fonction de l'appartenance à un continent du pays du client. Notez que on interdit le changement de pays du Clients, l'opération de transfert des données associées à ce client et situées dans d'autres tables étant jugé trop complexe.

```
create or replace TRIGGER MODIFY_CLIENTS
INSTEAD OF DELETE OR INSERT OR UPDATE ON CLIENTS
BEGIN
IF INSERTING then
  IF (:NEW.pays in ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican',
                    'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie',
'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
                    'Autriche', 'Suisse')) THEN

      INSERT INTO lpoisse.Clientses@linkES
      VALUES (:new.code_client,:new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone, :new.fax);

   ELSIF (:new.pays in  ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil',
                         'Canada', 'Chili', 'Colombie','Costa Rica',
'Cuba', 'Republique dominicaine', 'Dominique',
                         'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala',
'Guyana', 'Haiti','Honduras', 'Jamaique',
                         'Mexique', 'Nicaragua', 'Panama', 'Paraguay',
'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie',
                         'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay',
                         'Venezuela')) then
```

```
         INSERT INTO Clients_am
         VALUES (:new.code_client,:new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone, :new.fax);

      ELSIF  (:new.pays in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas'))
        then

         INSERT INTO cpottiez.ClientsEN@tplink
         VALUES (:new.code_client,:new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone, :new.fax);

      else
         INSERT INTO cpottiez.ClientsOI@tplink
          VALUES (:new.code_client,:new.societe, :new.adresse, :new.ville,
:new.code_postal, :new.pays, :new.telephone, :new.fax);
    end if;
      END IF;

  if UPDATING THEN
    if(:New.pays <> :OLD.pays) then
      RAISE_APPLICATION_ERROR(-20010,'Attention, vous n''êtes pas autorisé
à changer le pays de l''entreprise');
    end if;
    IF (:NEW.pays in ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican',
                      'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie',
'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
                      'Autriche', 'Suisse')) THEN
      update lpoisse.clientses@linkES
      set
        code_client = :new.code_client,
        societe = :NEW.societe,
        adresse = :new.adresse,
        ville = :new.ville,
        code_postal = :new.code_postal,
        pays = :new.pays,
        telephone = :new.telephone,
        fax = :new.fax
      WHERE code_client = :old.code_client;
    ELSIF  (:new.pays in ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil',
                          'Canada', 'Chili', 'Colombie','Costa Rica',
'Cuba', 'Republique dominicaine', 'Dominique',
                          'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala',
'Guyana', 'Haiti','Honduras', 'Jamaique',
                          'Mexique', 'Nicaragua', 'Panama', 'Paraguay',
'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie',
                          'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay',
                          'Venezuela')) then
      update clients_am
      set
        code_client = :new.code_client,
        societe = :NEW.societe,
        adresse = :new.adresse,
```

```
        ville = :new.ville,
        code_postal = :new.code_postal,
        pays = :new.pays,
        telephone = :new.telephone,
        fax = :new.fax
      WHERE code_client = :old.code_client;
    ELSIF  (:new.pays in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique', 'Irlande',
                          'Pologne', 'Royaume-Uni', 'Allemagne', 'Islande',
'Luxembourg', 'Pays-Bas')) then
      update cpottiez.clientsen@tplink
      set
        code_client = :new.code_client,
        societe = :NEW.societe,
        adresse = :new.adresse,
        ville = :new.ville,
        code_postal = :new.code_postal,
        pays = :new.pays,
        telephone = :new.telephone,
        fax = :new.fax
      WHERE code_client = :old.code_client;
    else
      update cpottiez.clientsoi@tplink
      set
        code_client = :new.code_client,
        societe = :NEW.societe,
        adresse = :new.adresse,
        ville = :new.ville,
        code_postal = :new.code_postal,
        pays = :new.pays,
        telephone = :new.telephone,
        fax = :new.fax
      WHERE code_client = :old.code_client;
    end if;
  end if;

  IF DELETING THEN
    IF  (:old.pays in ('Espagne', 'Portugal', 'Andorre', 'France',
'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican',
                       'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie',
'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie',
                       'Autriche', 'Suisse')) THEN
      delete from lpoisse.clientses@linkES where code_client =
:old.code_client;

    ELSIF (:old.pays in  ('Antigua-et-Barbuda', 'Argentine', 'Bahamas',
'Barbade', 'Belize', 'Bolivie', 'Bresil',
                          'Canada', 'Chili', 'Colombie','Costa Rica',
'Cuba', 'Republique dominicaine', 'Dominique',
                          'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala',
'Guyana', 'Haiti','Honduras', 'Jamaique',
                          'Mexique', 'Nicaragua', 'Panama', 'Paraguay',
'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie',
                          'Saint-Vincent-et-les Grenadines', 'Salvador',
'Suriname', 'Trinite-et-Tobago', 'Uruguay',
                          'Venezuela')) then
        delete from clients_AM where code_client = :old.code_client;
```

```
   ELSIF (:old.pays in ('Suede', 'Norvege', 'Danemark', 'Finlande',
'Belgique',
                        'Irlande', 'Pologne', 'Royaume-Uni', 'Allemagne',
'Islande', 'Luxembourg', 'Pays-Bas'))then
         delete from cpottiez.clientsEN@tplink where code_client =
:old.code_client;
    else
         delete from cpottiez.clientsOI@tplink where code_client=
:old.code_client;
    end if;
  END IF;
      end;
```

Ainsi, ces triggers sur les vues Commandes, DétailsCommandes, Stock et Clients garantissent une intégrité lors de l'insertion, modification et suppression des données lorsqu'on effectue des opérations dessus.

g) Nettoyages éventuels

Nous supprimons les différnts tuples ajoutés dans la table afin de tester son bon fonctionnement.

h) Tests de vérification du bon fonctionnement

```
-- test insert on commande
INSERT INTO COMMANDES VALUES (7000, '12345', 3, CURRENT_DATE, null, null);
-- ES
INSERT INTO COMMANDES VALUES (5000, 'FOLKO', 3, CURRENT_DATE, null, null);
-- EN
INSERT INTO COMMANDES VALUES (6666, 'PICCO', 3, CURRENT_DATE, null, null);
-- OI
COMMIT;

-- test update on commandes
UPDATE COmmandes SET date_envoi = CURRENT_DATE WHERE code_client = '12345';
-- ES
UPDATE COmmandes SET date_envoi = CURRENT_DATE WHERE code_client = 'FOLKO';
-- EN
UPDATE COmmandes SET date_envoi = CURRENT_DATE WHERE code_client = 'PICCO';
-- OI
COMMIT;

-- test delete on commandes
DELETE FROM Commandes WHERE code_client = '12345';  -- ES
DELETE FROM Commandes WHERE code_client = 'FOLKO';  -- EN
DELETE FROM Commandes WHERE code_client = 'PICCO';  -- OI
COMMIT;

-- test insert on clients
INSERT INTO CLIENTS VALUES ('haha', 'DARK side', 'death star', 'far far
away', 666, 'Espagne', 'here''s my number', 'call me maybe'); -- ES
INSERT INTO CLIENTS VALUES ('hihi', 'cold', 'st. cold, 31', 'Coldington',
'-300', 'Suede', '000', '111'); -- EN
INSERT INTO CLIENTS VALUES ('un', 'bla', 'blabla', 'Blablatown', '666',
'Autriche', '000', '111');  -- OI
COMMIT;
```

```
-- test update on clients
UPDATE CLIENTS SET code_postal = 'new666' WHERE code_client = 'haha'; -- ES
UPDATE CLIENTS SET code_postal = 'new-300' WHERE code_client = 'hihi';  --
EN
UPDATE CLIENTS SET code_postal = 'new' WHERE code_client = 'un';  -- OI
COMMIT;

-- test delete on clients
DELETE FROM CLIENTS WHERE code_client = 'haha'; -- ES
DELETE FROM CLIENTS WHERE code_client = 'hihi'; -- EN
DELETE FROM CLIENTS WHERE code_client = 'un'; -- OI
      COMMIT;
```

IV. Tests de requête distribuées et optimisations

Maintenant, nous faire des textes de requêtes distribuées sur nos bases de données et étudier les optimisions effectuées par le SGBC

A. Site Europe du Nord

Nous effectuons une sélection sur la vue stock.

```
select * from stock;
```

Résultat: 237 tuples en 0.02 secondes. Le plan d'exécution montre que la vue est faite d'une union entre toutes les tables avec les prédicats de filtre par défaut définis dans la vue. Sur les tables remote, il n'y a pas de sélection. Le coût est de 12.

On fait des sélections sur les pays.

```
SELECT * from STOCK where PAYS='Suede';
```

Résultats : 0 tuples en 0.013 secondes

Le plan d'exécution ne sélectionne que la table 'STOCKEN' (avec une sélection pays=Suède), puisque la combinaison des prédicats de filtre et du prédicat de la sélection donne des filtres de type 'null is not null' sur les autres tables. La Suède est dans la liste des pays d'Europe du nord mais pas dans les autres. Il est donc inutile de s'occuper des tables distantes et de 'STOCKOI'. De plus, le plan d'exécution montre que l'index est utilisé pour la sélection puisque le pays fait partie de la clé primaire et que l'index permet d'accélérer la sélection. Le coût est ici de 4.

On prend un pays de chaque zone

```
SELECT * from STOCK where PAYS in('Suede','Mexique','France');
```

Résultats : 78 tuples en 0.017 secondes

Le plan d'exécution montre que la table 'STOCKEN' est filtré avec un prédicat pays=Suede et une utilisation de l'index. La table 'STOCKOI' est elle aussi prise en compte avec la combinaison des prédicats déjà définis et les prédicats de sélection.

Elle doit être scannée entièrement. Les tables remotes sont elle aussi utilisées sans filtre préalable. Le coût est de 9.

```
select * from lpoisse.stockes@LinkToDBES;
```

Résultats : 77 tuples en 0.008 secondes. On sélectionne la table dans son intégralité

```
select * from lpoisse.stockes@LinkToDBES where pays='France';
```

Résultats : 76 tuples en 0.009 secondes. On utilise l'index sur la clé primaire pour la sélection.

```
select * from CLIENTS NATURAL JOIN  COMMANDES;
```

Résultats : 833 tuples en 0.609 secondes. Aucune sélection.

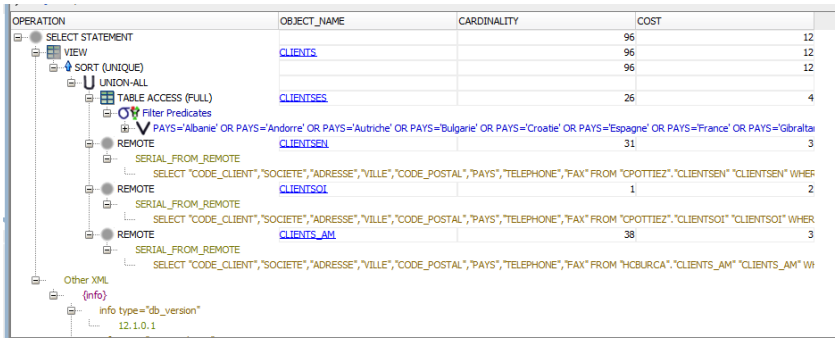
```
select count(*) from CLIENTS NATURAL JOIN  COMMANDES where pays='Espagne';
```

Résultats : 25 tuples en 0.024 secondes. On tire profit de l'utilisation des prédicats sur la vue clients.

Globalement, les prédicats définis au niveau de la vue servent à éliminer des accès aux tables. Si la combinaison des prédicats définis et des prédicats de sélection pour une table donne 'null is not null', on n'a donc nullement besoin d'accéder à la table en question. Sinon, il y a deux cas. Si la table est en local, l'expression est simplifié et normalisée pour la sélection. Par contre, si la table est distante, tous les tubles sont récupérés. A noter par ailleurs que les prédicats sur la table STOCKOI ne semblent pas se comporter comme ceux sur les autres tables. En effet, dès qu'il y a plus d'un prédicat simple sur cette table, l'optimiseur ne fait plus la combinaison logique des prédicats alors qu'il pourrait donner des 'null is not null'.

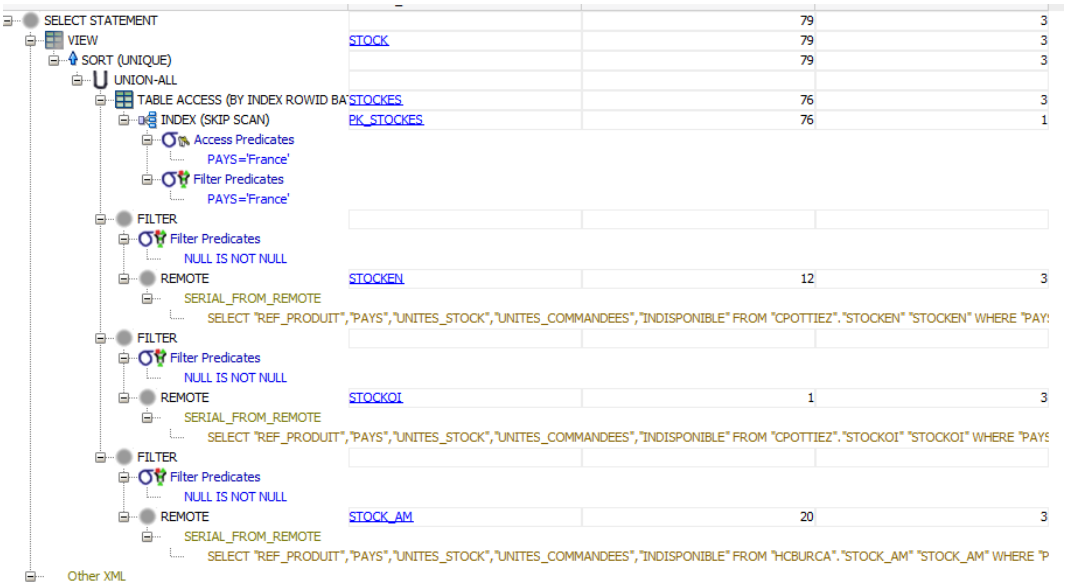
B . Site Europe du Sud

select * from stock;



On fait un full scan de toutes les vues, ce qui est logique car on doit extraires toutes les donnes relatives au stock dans toutes les bases.

SELECT * from STOCK where PAYS='Suede';



On remarque la présence de simplifications de type null is not null. Cette simplification a été faite par le SGBD car nous avons donné une description de la vue Stock lui permettant la simplification. Ainsi, le SGBD ne cherche pas les données sur les sites autres que le site de l'Europe du Sud.

```
select * from cpottiez.stockEN@dblinkMain;
select * from cpottiez.stockOI@dblinkMain;
select * from hcburca.stock_am@LinkToDBUS;
```

Dans ces trois cas, on se contente de faire un accés complets aux tables remotes.

```
select * from stockes;
```

Dans ce cas, on se contente de selectionner toutes les données en local.

```
select * from CLIENTS NATURAL JOIN COMMANDES where pays= « France »;
```

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT			23
HASH JOIN		513	23
Access Predicates			
CLIENTS.CODE_CLIENT=COMMANDES.CODE_CLIENT			
VIEW	CLIENTS	16	10
SORT (UNIQUE)			
UNION-ALL			
TABLE ACCESS (FULL)	CLIENTSES	11	4
Filter Predicates			
PAYS='France'			
FILTER			
Filter Predicates			
NULL IS NOT NULL			
REMOTE	CLIENTSEN	3	3
SERIAL_FROM_REMOTE			
SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "CPOTTIEZ"."CLIENTSEN" "CLIENTSEI			
FILTER			
Filter Predicates			
NULL IS NOT NULL			
REMOTE	CLIENTSOI	1	2
SERIAL_FROM_REMOTE			
SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "CPOTTIEZ"."CLIENTSOI" "CLIENTSO			
FILTER			
Filter Predicates			
NULL IS NOT NULL			
REMOTE	CLIENTS_AM	5	3
SERIAL_FROM_REMOTE			
SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "HCBURCA"."CLIENTS_AM" "CLIENTS			
VIEW	COMMANDES	834	13
SORT (UNIQUE)		834	13
UNION-ALL			
TABLE ACCESS (FULL)	COMMANDESES	201	4
REMOTE	COMMANDESEN	307	3
SERIAL_FROM_REMOTE			
SELECT "NO_COMMANDE","CODE_CLIENT","NO_EMPLOYE","DATE_COMMANDE","DATE_ENVOI","PORT" FROM "CPOTTIEZ"."COMMANDESEN" "COMM			
REMOTE	COMMANDESOI	1	3
SERIAL_FROM_REMOTE			
SELECT "NO_COMMANDE","CODE_CLIENT","NO_EMPLOYE","DATE_COMMANDE","DATE_ENVOI","PORT" FROM "CPOTTIEZ"."COMMANDESOI" "COMM			
REMOTE	COMMANDES_AM	325	3
SERIAL_FROM_REMOTE			
SELECT "NO_COMMANDE","CODE_CLIENT","NO_EMPLOYE","DATE_COMMANDE","DATE_ENVOI","PORT" FROM "HCBURCA"."COMMANDES_AM" "COMI			

Nous remarquons que le SGBD utilise les prédicats sur les tables clients mais pas sur commandes. En effet, cela s'explique par le fait que nous n'avons pas mis de conditions sur la vue Commandes.

C. Site Amériques

```
-- requête sur stock

-- on prend tout sans sélection
select * from stock;
```

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		237	13
VIEW	STOCK	237	13
UNION-ALL			
TABLE ACCESS (FULL)	STOCK_AM	81	4
Filter Predicates			
OR			
PAYS='Argentine'			
PAYS='Bahamas'			
PAYS='Barbade'			
PAYS='Belize'			
PAYS='Bolivie'			
PAYS='Bresil'			
PAYS='Canada'			
PAYS='Chili'			
PAYS='Colombie'			
PAYS='Costa Rica'			

```
/*
  résultat: 231 tuples en 0.0143 secondes
  le plan d'exécution montre que la vue est faite d'une union entre toutes
les tables avec les prédicats de filtre par défaut
  définis dans la vue. Sur les tables remote, il n'y a pas de sélection.
  On fait la sélection après la mise en place de la vue.
  Le coût est de 13.
 */

-- on fait des sélections sur les pays
SELECT * from STOCK where PAYS='Mexique';
/*
```

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		59	10
VIEW	STOCK	59	10
UNION-ALL			
TABLE ACCESS (BY INDEX ROWID BATC	STOCK_AM	2	2
INDEX (SKIP SCAN)	PKSTOCK	2	1
Access Predicates			
PAYS='Mexique'			
Filter Predicates			
PAYS='Mexique'			
FILTER			
Filter Predicates			
NULL IS NOT NULL			
REMOTE	STOCKES	39	3
FILTER			
Filter Predicates			
NULL IS NOT NULL			

```
  O tuples en 0.016 secondes
  le plan d'exécution ne sélectionne que la table 'STOCK_AM',
   puisque la combinaison des prédicats de filtre et du prédicat
  de la sélection donne des filtres de type 'null is not null' sur les
autres tables. Le Mexique est dans la liste des
  pays d'Amérique mais pas dans les autres. Il est donc inutile de
s'occuper des tables distantes.
  De plus, le plan d'exécution montre que l'index est utilisé pour la
selection puisque le pays fait partie de la
   clé primaire et que l'index permet d'accélèrer la sélection.
   Le coût est ici de 10.
 */

-- on prend un pays de chaque zone
SELECT * from STOCK where PAYS in('Suede','Mexique','France');
/*
```

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		164	10
VIEW	STOCK	164	10
UNION-ALL			
TABLE ACCESS (BY INDEX ROWID BATC	STOCK_AM	2	2
INDEX (SKIP SCAN)	PKSTOCK	2	1
Access Predicates			
PAYS='Mexique'			
Filter Predicates			
PAYS='Mexique'			
REMOTE	STOCKES	39	3
REMOTE	STOCKEN	1	2
REMOTE	STOCKOI	1	3

```
  76 tuples en 0.033 secondes
  Le plan d'exécution montre que la table 'STOCK_AM' est filtré avec  un
prédicat pays=Mexique et une utilisation de l'index.
  Les tables remotes sont elle aussi utilisées sans filtre préalable.
  Le coût est de 10.
 */
```

```
select * from lpoisse.stockes@linkES;
/*
```

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT (REMOTE)		77	4
TABLE ACCESS (FULL)	STOCKES	77	4

```
  76 tuples en 0.013 secondes
  on sélectionne la table dans son intégralité
 */

select * from lpoisse.stockes@linkES where pays='France';
/*
```

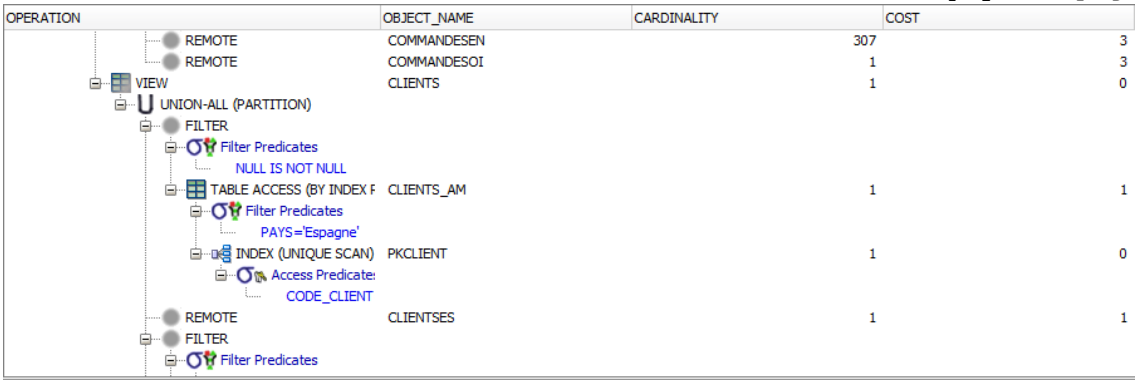
OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT (REMOTE)		76	3
TABLE ACCESS (BY INDEX ROWID BATCHED)	STOCKES	76	3
INDEX (SKIP SCAN)	PK_STOCKES	76	1
Access Predicates			
A1.PAYS='France'			
Filter Predicates			
A1.PAYS='France'			

```
  76 tuples en 0.012 secondes
  on utilise l'index sur la clé primaire pour la sélection
 */

/*
  Si on souhaite utiliser des prédicats prédifinies. Il faudrait définir
des vues avec des prédicats pour chaque
  fragment.
*/

select * from CLIENTS NATURAL JOIN  COMMANDES;
/*
```

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		2083	18
HASH JOIN		2083	18
Access Predicates			
CLIENTS.CODE_CLIENT=COMMANDES.			
VIEW	CLIENTS	95	9
UNION-ALL			
TABLE ACCESS (FULL)	CLIENTS_AM	38	4
Filter Predicates			
OR			
PAYS='Argentine'			
PAYS='Bahamas'			
PAYS='Barbade'			
PAYS='Belize'			
PAYS='Bolivie'			
PAYS='Bresil'			
PAYS='Canada'			

```
  833 tuples en 0.115 secondes
  aucune sélection
 */
```

```
select count(*) from CLIENTS NATURAL JOIN  COMMANDES where pays='Espagne';
```



OPERATION	OBJECT_NAME	CARDINALITY	COST
REMOTE	COMMANDESEN	307	3
REMOTE	COMMANDESOI	1	3
VIEW	CLIENTS	1	0
UNION-ALL (PARTITION)			
FILTER			
Filter Predicates			
NULL IS NOT NULL			
TABLE ACCESS (BY INDEX F	CLIENTS_AM	1	1
Filter Predicates			
PAYS='Espagne'			
INDEX (UNIQUE SCAN)	PKCLIENT	1	0
Access Predicate			
CODE_CLIENT			
REMOTE	CLIENTSES	1	1
FILTER			
Filter Predicates			

```
/*
  25 tuples en  0.461 secondes
  on tire profit de l'utilisation des prédicats sur la vue clients.
 */

/*
  Globalement, les prédicats définis au niveau de la vue servent à éliminer
des accès aux tables.
  Si la combinaison des prédicats définis et des prédicats de sélection
pour une table donne 'null is not null', on n'a donc
  nullement besoin d'accéder à la table en question.
  Sinon, il y a deux cas. Si la table est en local, l'expression est
simplifié et normalisé pour la sélection. Par contre,
  si la table est distante, tous les tubles sont récupérés.
  A noter par ailleurs que les prédicats sur la table STOCKOI ne semblent
pas se comporter comme ceux sur les autres tables.
  En effet, dès qu'il y a plus d'un prédicat simple sur cette table,
l'optimiseur ne fait plus la combinaison logique des
  prédicats alors qu'il pourrait donner des 'null is not null'.
 */
```

V. Exemples de Réplications

Europe du Nord

```
--création materialized view pour employes toutes les semaines
CREATE MATERIALIZED VIEW MW_Employes
  REFRESH FAST
  NEXT sysdate + 7 -- tous les 7 jours
  as
  SELECT *
  FROM hcburca.employes@LinkToDBUS;

-- consulter le salaire d'un employé d'europe du nord
SELECT *
 FROM MW_Employes where PRENOM='Nancy';-- ok

-- connaître les informations des employés du site EN
SELECT * from COMMANDESEN c NATURAL JOIN MW_Employes e;
```

```
--création materialized view pour produits tous les jours
CREATE MATERIALIZED VIEW MW_PRODUITS
  REFRESH COMPLETE
  NEXT sysdate + 1
  AS
  SELECT * FROM lpoisse.produits@LinkToDBES;


-- le produit le plus commandé en europe du nord
SELECT d.REF_PRODUIT,count(*)
FROM MW_PRODUITS m, DETAILS_COMMANDESEN d
WHERE m.REF_PRODUIT = d.REF_PRODUIT
GROUP BY d.REF_PRODUIT
HAVING count(*) =
       (SELECT max(count(*))
        FROM MW_PRODUITS m, DETAILS_COMMANDESEN d
        WHERE m.REF_PRODUIT = d.REF_PRODUIT
        GROUP BY d.REF_PRODUIT);

--création materialized view pour categories toutes les semaines
CREATE MATERIALIZED VIEW MW_CATEGORIES
  REFRESH FAST
  NEXT SYSDATE + 7
  AS
  SELECT *
    from lpoisse.categories@LinkToDBES;

-- quelle est la categorie de produits la plus commandé en EN
SELECT c.CODE_CATEGORIE, count(*)
FROM MW_PRODUITS m, DETAILS_COMMANDESEN d, MW_CATEGORIES c
WHERE m.REF_PRODUIT = d.REF_PRODUIT and m.CODE_CATEGORIE=c.CODE_CATEGORIE
GROUP BY c.CODE_CATEGORIE
HAVING count(*) =
       (SELECT max(count(*))
        FROM MW_PRODUITS m, DETAILS_COMMANDESEN d, MW_CATEGORIES c
        WHERE m.REF_PRODUIT = d.REF_PRODUIT and
m.CODE_CATEGORIE=c.CODE_CATEGORIE
        GROUP BY c.CODE_CATEGORIE);


-- log for founisseurs and grant access to log in order to enable other
sites to make a refresh fast
CREATE MATERIALIZED VIEW LOG ON FOURNISSEURS;

GRANT SELECT ON MLOG$_FOURNISSEURS TO lpoisse;
GRANT SELECT ON MLOG$_FOURNISSEURS TO zvergne;
GRANT SELECT ON MLOG$_FOURNISSEURS TO hcburca;
GRANT SELECT ON MLOG$_FOURNISSEURS TO jcharlesni;
GRANT SELECT ON MLOG$_FOURNISSEURS TO hhamelin;

-- quelle est la categorie de produits la plus commandée en EN avec la vue
matérialisée
SELECT c.CODE_CATEGORIE, count(*)
FROM MW_PRODUITS m, DETAILS_COMMANDESEN d, MW_CATEGORIES c
WHERE m.REF_PRODUIT = d.REF_PRODUIT and m.CODE_CATEGORIE=c.CODE_CATEGORIE
GROUP BY c.CODE_CATEGORIE
```

```
HAVING count(*) =
       (SELECT max(count(*))
        FROM MW_PRODUITS m, DETAILS_COMMANDESEN d, MW_CATEGORIES c
        WHERE m.REF_PRODUIT = d.REF_PRODUIT and
m.CODE_CATEGORIE=c.CODE_CATEGORIE
        GROUP BY c.CODE_CATEGORIE);
/*
 un tuple en 0.007 seconde
 */

-- quelle est la categorie de produits la plus commandée en EN sans la vue
matérialisée
SELECT c.CODE_CATEGORIE, count(*)
FROM PRODUITS m, DETAILS_COMMANDESEN d, CATEGORIES c
WHERE m.REF_PRODUIT = d.REF_PRODUIT and m.CODE_CATEGORIE=c.CODE_CATEGORIE
GROUP BY c.CODE_CATEGORIE
HAVING count(*) =
       (SELECT max(count(*))
        FROM PRODUITS m, DETAILS_COMMANDESEN d, CATEGORIES c
        WHERE m.REF_PRODUIT = d.REF_PRODUIT and
m.CODE_CATEGORIE=c.CODE_CATEGORIE
        GROUP BY c.CODE_CATEGORIE);

/*
un tuple en 0.055, soit environ 8 fois plus de temps qu'avec la
materialized view
 */



-- connaître le nombre d'employés du site EN avec la vue matérialisée
SELECT count(*) from COMMANDESEN c NATURAL JOIN MW_Employes e;
/*
        1 tuple en 0.008 seconde
 */

-- connaître le nombre d'employés du site EN sans la vue matérialisée
SELECT count(*) from COMMANDESEN c NATURAL JOIN Employes e;
/*
        1 tuple en 0.185 secondes soit environ 23 fois plus de temps
qu'avec la materialized view
 */

/*
  On constate un net gain de temps pour l'obtention du même résultat en
utilisant
  les vues matérialisées.
 */
```

2) Europe du Sud

Nous créons 3 materialized view. Une en refresh complete et une en refresh fast, les deux étant en read only.

```
  CREATE MATERIALIZED VIEW FournisseursMat
```

```
  Refresh complete
  next sysdate +(1/24/60)
  as select *
  from cpottiez.fournisseurs@dblinkMain;
select * from FournisseursMat;
--Ne fonctionne pas, c'est une materialized view
insert into fournisseursMat values(1337,'ZCorp','1337 rue du swag','JMP
Town',69100,'France','String','UnString');

Drop Materialized view EmployesMat;


CREATE MATERIALIZED VIEW EmployesMAT
  Refresh fast
  next sysdate +(1/24/60)
  as select *
  from hcburca.Employes@dblinkUS;
--Ne fonctionne pas, c'est materialized view
insert into employesMat
values(1337,10,'r','r','r','r',sysdate,sysdate,10,10);


Drop Materialized view EmployesMAT;
```

Naturellement, nous n'oublions pas de supprimer les vues matérialisées à la fin.

3) Amériques

Nous créons trois materialized view. Une en refresh complete et deux en refresh fast, toutes étant en read only et nous réalisons quelques tests:

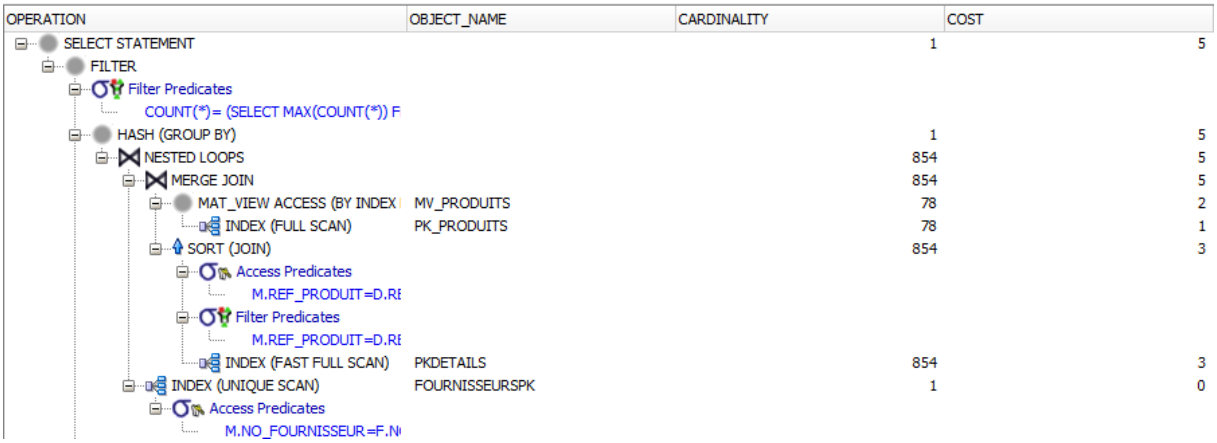
```
-- Création materialized view pour Produits tout les jours
CREATE MATERIALIZED VIEW MV_PRODUITS
  REFRESH COMPLETE
  NEXT SYSDATE +1
  AS
    SELECT *
    FROM LPOISSE.PRODUITS@LINKES;

-- Le produit le plus commandé en Amérique
SELECT d.REF_PRODUIT,count(*)
FROM MV_PRODUITS p, DETAILS_COMMANDES_AM d
WHERE p.REF_PRODUIT = d.REF_PRODUIT
GROUP BY d.REF_PRODUIT
HAVING count(*) =
       (SELECT max(count(*))
        FROM MV_PRODUITS p, DETAILS_COMMANDES_AM d
        WHERE p.REF_PRODUIT = d.REF_PRODUIT
        GROUP BY d.REF_PRODUIT);

-- consulter les informations d'un produit d'Amerique
SELECT *
 FROM MV_PRODUITS where REF_PRODUIT=2;-- ok

-- Création materialized view pour categories toutes les semaines
CREATE MATERIALIZED VIEW MV_CATEGORIES
  REFRESH FAST
```

```
  NEXT SYSDATE + 7
  AS
  SELECT *
    from lpoisse.categories@LinkES;

-- Création materialized view pour Fournisseurs toutes les semaines
CREATE MATERIALIZED VIEW MV_FOURNISSEURS
  REFRESH FAST
  NEXT SYSDATE + 7
  AS
  SELECT *
    from cpottiez.FOURNISSEURS@tplink;

DROP MATERIALIZED VIEW MV_FOURNISSEURS;
DROP MATERIALIZED VIEW MV_PRODUITS;

-- Connaître les informations des fournisseurs du site AM
SELECT * from COMMANDES_AM c NATURAL JOIN MV_FOURNISSEURS f;

-- log for employes and grant access to log in order to enable other sites
to make a refresh fast
CREATE MATERIALIZED VIEW LOG ON EMPLOYES;

GRANT SELECT ON MLOG$_EMPLOYES TO lpoisse;
GRANT SELECT ON MLOG$_EMPLOYES TO zvergne;
GRANT SELECT ON MLOG$_EMPLOYES TO jcharlesni;
GRANT SELECT ON MLOG$_EMPLOYES TO cpottiez;
GRANT SELECT ON MLOG$_EMPLOYES TO hhamelin;
```

Puis nous étudions la différence des plans d'exécutions avec des requêtes similaires mais appelant d'abord les vues matérialisées puis les vues de base :

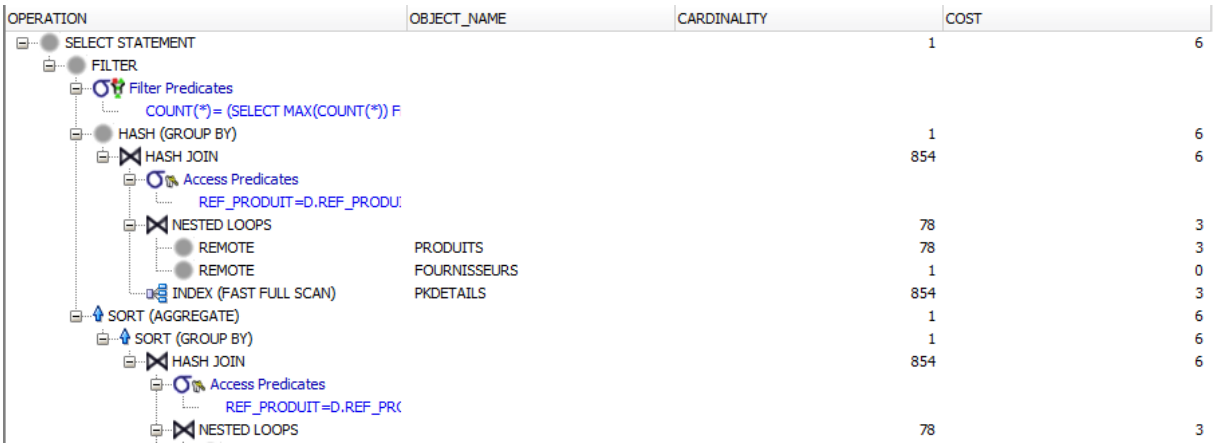
```
-- quel est le fournisseur de produits le plus employé en Amérique avec la
vue matérialisée
SELECT f.NO_FOURNISSEUR , count(*)
FROM MV_PRODUITS m, DETAILS_COMMANDES_AM d, MV_FOURNISSEURS f
WHERE m.REF_PRODUIT = d.REF_PRODUIT and m.NO_FOURNISSEUR=f.NO_FOURNISSEUR
GROUP BY f.NO_FOURNISSEUR
HAVING count(*) =
       (SELECT max(count(*))
        FROM MV_PRODUITS m, DETAILS_COMMANDES_AM d, MV_FOURNISSEURS f
        WHERE m.REF_PRODUIT = d.REF_PRODUIT and
m.NO_FOURNISSEUR=f.NO_FOURNISSEUR
        GROUP BY f.NO_FOURNISSEUR);
```

```
/*
```



OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		1	5
FILTER			
Filter Predicates			
COUNT(*)= (SELECT MAX(COUNT(*)) F			
HASH (GROUP BY)		1	5
NESTED LOOPS		854	5
MERGE JOIN		854	5
MAT_VIEW ACCESS (BY INDEX	MV_PRODUITS	78	2
INDEX (FULL SCAN)	PK_PRODUITS	78	1
SORT (JOIN)		854	3
Access Predicates			
M.REF_PRODUIT=D.RI			
Filter Predicates			
M.REF_PRODUIT=D.RI			
INDEX (FAST FULL SCAN)	PKDETAILS	854	3
INDEX (UNIQUE SCAN)	FOURNISSEURSPK	1	0
Access Predicates			
M.NO_FOURNISSEUR=F.N			

```
 un tuple en 0.012 secondes
 On remarque qu'il consulte la vue matérialisée qui subit un full scan sur
la primary key et la jointure qui subit sélection avec des access
predicates et des filter predicates toujours avec un full scan sur la
primary key.
 Le coup est de 5.
 */

-- quel est le fournisseur de produits le plus employé en Amériqye sans la
vue matérialisée
SELECT f.NO_FOURNISSEUR, count(*)
FROM PRODUITS m, DETAILS_COMMANDES_AM d, FOURNISSEURS f
WHERE m.REF_PRODUIT = d.REF_PRODUIT and m.NO_FOURNISSEUR=f.NO_FOURNISSEUR
GROUP BY f.NO_FOURNISSEUR
HAVING count(*) =
        FROM PRODUITS m, DETAILS_COMMANDES_AM d, FOURNISSEURS f
        WHERE m.REF_PRODUIT = d.REF_PRODUIT and
m.NO_FOURNISSEUR=f.NO_FOURNISSEUR
        GROUP BY f.NO_FOURNISSEUR);

/*
```



OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		1	6
FILTER			
Filter Predicates			
COUNT(*)= (SELECT MAX(COUNT(*)) F			
HASH (GROUP BY)		1	6
HASH JOIN		854	6
Access Predicates			
REF_PRODUIT=D.REF_PRODU			
NESTED LOOPS		78	3
REMOTE	PRODUITS	78	3
REMOTE	FOURNISSEURS	1	0
INDEX (FAST FULL SCAN)	PKDETAILS	854	3
SORT (AGGREGATE)		1	6
SORT (GROUP BY)		1	6
HASH JOIN		854	6
Access Predicates			
REF_PRODUIT=D.REF_PR			
NESTED LOOPS		78	3

```
un tuple en 0.149, soit environ 14 fois plus de temps qu'avec la
materialized view
On remarque qu'ici des nested loops apparaissent en bas des arbres et que
ce ne sont plus des merge join mais des hash join.
Le coût est de 6.
 */
```

```
-- connaître le nombre de produits du site Amérique avec la vue
matérialisée
SELECT count(*) from COMMANDES_AM c NATURAL JOIN MV_PRODUITS m;
/*
```

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		1	24
SORT (AGGREGATE)		1	
MERGE JOIN (CARTESIAN)		25350	24
INDEX (FULL SCAN)	PK_PRODUITS	78	1
BUFFER (SORT)		325	23
INDEX (FAST FULL SCAN)	PKCOMMANDE	325	0

```
        1 tuple en 0.003 secondes
        Encore une fois on effectue un merge join avec des full scan sur
les clés primaires
        Le coût est de 24.
 */

-- connaître le nombre de produits du site Amérique sans la vue
matérialisée
SELECT count(*) from COMMANDES_AM c NATURAL JOIN Produits p;
/*
```

OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		1	24
SORT (AGGREGATE)		1	
MERGE JOIN (CARTESIAN)		25350	24
REMOTE	PRODUITS	78	1
BUFFER (SORT)		325	23
INDEX (FAST FULL SCAN)	PKCOMMANDE	325	0

```
        1 tuple en 0.015 secondes soit environ 5 fois plus de temps qu'avec
la materialized view
        Ici ça va être très similaire à l'arbre de la requete précédente
sauf que l'on va consulter toute la table produits
        Le coût est de 24.
 */

/*
  On constate un net gain de temps pour l'obtention du même résultat en
utilisant
  les vues matérialisées.
 */
```

VI. Annexes

Voici les tests effectués par le binôme en charge d'Europe du Nord.

```
/*
  Test contraintes en local
 */
-- quelques tests pour vérifier les contraintes sur les bases locales
insert into CLIENTSEN values ('ALFKI',    'Alfreds Futterkiste','Obere Str.
57    ','Berlin'  ,'12209'    ,'Allemagne'      ,'030-0074321'    ,'030-
0076545');
-- déjà présent:  violation de contrainte unique

insert into COMMANDESOI values(1051,null,3 ,NULL ,null,null);
```

```
-- impossible d'insérer null

insert INTO COMMANDESEN values(1051,'clclo',3 ,sysdate ,sysdate,279);
--  pas de client référencé avec ce nom: violation de contrainte
d'intégrité (clé étrangère sur code_clients)

insert into DETAILS_COMMANDESEN values(850      ,54    ,37.25,      20     ,0);
-- pas de commande avec ce numéro: violation de contrainte de clé étrangère

insert into STOCKEN values ('62','Allemagne',null,'17','0');
-- clé primaire déjà utilisé:  violation de contrainte unique

UPDATE STOCKEN set REF_PRODUIT=58 where PAYS='Allemagne' and
REF_PRODUIT='100'; --maj correcte


/*
  Contraintes de clés étrangères sur les autres bases
 */
insert into COMMANDESEN values(105,'VAFFE',     15,sysdate,sysdate,555);
-- no_employe invalide: signalé via trigger
insert into DETAILS_COMMANDESEN values(101,     1555, 37.25,      20,    0);
-- no_produit invalide: signalé via trigger
insert into STOCKEN VALUES (1000,   'Allemagne',17,null,          0);
-- no_produit invalide: signalé via trigger
DELETE  from FOURNISSEURS where NO_FOURNISSEUR=11;
-- fournisseurs référencé dans produits, impossible à supprimer: signalé
via trigger

/*
  Gestion des IMD sur les vues
 */
--test sur la vue stock
SELECT REF_PRODUIT,count(*)
  FROM STOCK NATURAL JOIN PRODUITS
GROUP BY REF_PRODUIT;
insert into STOCK VALUES (2,'Croatie',17,40,0);
SELECT * FROM lpoisse.stockes@LinkToDBES where pays='Croatie';
update STOCK SET UNITES_COMMANDEES=50 where pays='Croatie';
DELETE FROM STOCK where pays='Croatie';

insert into STOCK VALUES (2,'Pays-Bas',17,40,0);
SELECT * FROM STOCKEN where pays='Pays-Bas';
update STOCK SET UNITES_COMMANDEES=50 where pays='Pays-Bas';
DELETE FROM STOCK where pays='Pays-Bas';


insert into STOCK VALUES (2,'Chili',17,40,0);
SELECT * FROM hcburca.stock_am@LinkToDBUS where pays='Chili';
update STOCK SET INDISPONIBLE=-1 where pays='Chili' and REF_PRODUIT=2; --
impossible de maj: étrange...
DELETE FROM STOCK where pays='Chili' and REF_PRODUIT=2; --impossible de
supprimer... manque de permission

insert into STOCK VALUES (2,'Russie',17,40,0);
SELECT * FROM STOCKOI where pays='Russie';
update STOCK SET UNITES_COMMANDEES=50 where pays='Russie';
DELETE FROM STOCK where pays='Russie';
```

```
update STOCK SET pays='Chine' WHERE PAYS='Russie'; -- pas autorisé à
changer le pays

--tests sur la vue 'Clients'
SELECT * from CLIENTS;
SELECT * from CLIENTS NATURAL join COMMANDES;

insert into CLIENTS values('CHO','Chop-suey', 'Chinese','Hauptstr. 29
      Bern','     3012','Bulgarie', '0452-076545      ',NULL );
SELECT * from  lpoisse.ClientsES@LinkToDBES where pays='Bulgarie';
update CLIENTS SET pays='France' where pays='Bulgarie'; -- pas autorisé à
changer le pays
update CLIENTS SET CODE_POSTAL=3568 where CODE_CLIENT='CHO'; -- update
réussi
SELECT * FROM CLIENTS where CODE_CLIENT='CHO'; --1 tuple

insert into CLIENTS values('CHO','Chop-suey', 'Chinese','Hauptstr. 29
      Bern','     3012','Norvege',  '0452-076545      ',NULL );
SELECT * from  CLIENTSEN where pays='Norvege';
update CLIENTS SET CODE_CLIENT='CHO' where pays='Norvege'; -- violation de
PK car 2 tuples selectionnés, clé primaire identiques mais sur 2 sites
différents
update CLIENTS SET CODE_POSTAL=3568 where CODE_CLIENT='CHO'; -- update
réussi
SELECT * FROM CLIENTS where CODE_CLIENT='CHO'; --2 tuples

insert into CLIENTS values('CHO','Chop-suey', 'Chinese','Hauptstr. 29
      Bern','     3012','Russie',   '0452-076545      ',NULL );
SELECT * from  CLIENTSOI where pays='Russie';
update CLIENTS SET CODE_POSTAL=3568 where CODE_CLIENT='CHO'; -- update
réussi
SELECT * FROM CLIENTS where CODE_CLIENT='CHO'; -- 3 tuples avec la même PK
dans 3 sites différents

DELETE  from CLIENTS where CODE_CLIENT='CHO';
COMMIT ;

insert into CLIENTS values('CHO1','Chop-suey', 'Chinese','Hauptstr. 29
      Bern','     3012','Chili',    '0452-076545      ',NULL );
SELECT * from  hcburca.clients_am@LinkToDBUS where pays='Chili';
update CLIENTS SET CODE_POSTAL=3568 where CODE_CLIENT='CHO1'; -- update
réussi
SELECT * FROM CLIENTS where CODE_CLIENT='CHO1'; -- 1 tuples

-- tests sur la vue 'Commandes'
select c.NO_EMPLOYE, count(*)
  FROM COMMANDES c,EMPLOYES e
    where c.NO_EMPLOYE=e.NO_EMPLOYE
  GROUP BY c.NO_EMPLOYE;

select cl.CODE_CLIENT, count(*)
  from CLIENTS cl, COMMANDES c
  where c.CODE_CLIENT=cl.CODE_CLIENT
  group by cl.CODE_CLIENT;

insert into CLIENTS values('CHO','Chop-suey', 'Chinese','Hauptstr. 29
      Bern','     3012','Russie',   '0452-076545      ',NULL ); --ok dans OI
```

```
insert into COMMANDES values(5415,'CHO',1,sysdate,sysdate,469); --ok dans
OI
SELECT * from COMMANDESOI where NO_COMMANDE=5415;
delete from COMMANDES where NO_COMMANDE=5415; --ok
delete from CLIENTS where CODE_CLIENT='CHO';--ok

insert INTO COMMANDES values(1051,'clclo',3 ,sysdate ,sysdate,279); --
client inconnu
insert INTO COMMANDES values(901,'ALFKI',3 ,sysdate ,sysdate,279);
select * from COMMANDESEN where NO_COMMANDE=901; --placé dans commandesEN,
en accord avec le pays du client
update COMMANDES set CODE_CLIENT='FURIB' where NO_COMMANDE=901; -- pas
autorisé à changer le code_client
delete FROM  COMMANDES where NO_COMMANDE='901';

-- tests sur la vue 'Details_Commandes'
select *
  from   DETAILS_COMMANDES NATURAL join COMMANDES ;

select NO_COMMANDE from COMMANDES MINUS SELECT DISTINCT NO_COMMANDE from
DETAILS_COMMANDES; --commande sans détails

insert into DETAILS_COMMANDES values(5000,70,75,14,0.1);
select * from DETAILS_COMMANDES where NO_COMMANDE=6000;
update DETAILS_COMMANDES set no_commande=6000 where NO_COMMANDE=5000 AND
REF_PRODUIT=70; -- ne peut pas changer le numéro de commande
update DETAILS_COMMANDES set no_commande=9000 where NO_COMMANDE=5000 AND
REF_PRODUIT=70; --numéro de commande inconnu

-- tests sur la vue Employes
insert into EMPLOYES values (15,5,'Suyama','Michael','ReprÈsentant(e)',
      'M.', sysdate,    sysdate,2534,     600); -- ok
UPDATE EMPLOYES SET  REND_COMPTE=55 WHERE NO_EMPLOYE=15; --  violation de
FK : 55 pas employé
DELETE FROM EMPLOYES WHERE NO_EMPLOYE=12;
SELECT * from EMPLOYES WHERE NO_EMPLOYE=12;

-- tests sur la vue Produits
insert into PRODUITS values (101,'Ikura',4,11,'12 pots (200 g)',155); --
violation de FK pas de catégorie
insert into PRODUITS values (100,'Ikura',4,8,'12 pots (200 g)',155); --
insertion ok
update PRODUITS set NO_FOURNISSEUR=35 where REF_PRODUIT=100; --fournisseur
inexistant
delete from PRODUITS where REF_PRODUIT=100;

SELECT f.NO_FOURNISSEUR, count(*)
  from FOURNISSEURS f, PRODUITS p
where f.NO_FOURNISSEUR=p.NO_FOURNISSEUR
GROUP BY f.NO_FOURNISSEUR;

SELECT c.CODE_CATEGORIE,count(*)
  from CATEGORIES c, PRODUITS p
where c.CODE_CATEGORIE=p.CODE_CATEGORIE
GROUP BY c.CODE_CATEGORIE;

-- test sur la vue 'Categorie'
```

```
insert into CATEGORIES values(9,'Poissons et fruits de mer','Poissons,
fruits de mer, escargots'); --insertion ok
SELECT * from CATEGORIES;
update CATEGORIES set NOM_CATEGORIE='Céréales' where CODE_CATEGORIE=9; --ok
DELETE from CATEGORIES where CODE_CATEGORIE=7; --violation contrainte
catégories référencée ailleurs
DELETE from CATEGORIES where CODE_CATEGORIE=9;
```

Voici les tests effectués par le binôme en charge d'Europe du Sud.

```
/*
  FICHIER DE TESTS : Europe du Sud
*/


/*
  Vue stock : sélection
*/
Select * from Stock;    --Requête standard
Select * from Stock natural join produits;    -- Jointure

Select count(*), Pays from Stock
GROUP BY PAYS;        --Test de regroupement

/*
  Vue Stock : LMD
*/
insert into stock values (1 , 'Mexique', 5 , 1 ,0);     -- Fonctionne : Le
tuple est inséré dans la table Stock_AM du site américain
commit;

delete from stock where ref_produit=1 and PAYS='Mexique';     -- Fonctionne
également : le tuple inséré est supprimé dans le fragment correspondant
commit;

insert into stock values (2,'Portugal',54,1,0);     --Fonctionne : le tuple
est inséré dans la table stockES du site d'Europe du Sud
insert into stock values (2, 'Portugal', 44, 12, 1);    --Erreur :
contrainte d'intégrité (PK) non respectée
insert into stock values (195, 'INEX', 4,4, 5); -- Erreur : contrainte
d'intégrité liée à Produits (FK distante assurée par trigger) non respectée
=> le tuple est destiné à la table stockOI
insert into stock values (195, 'Portugal', 4,4, 5);  -- Erreur : contrainte
d'intégrité (FK locale) non respectée

insert into Stock values (3, 'Suede', 12, 2, 7);
update Stock set unites_commandees = 24 where pays='Suede' and
ref_produit=3;   --Idem : le fragment concerné est ici StockEN
delete from stock where pays='Suede' and ref_produit=3;

update stock set unites_stock =null where pays='Portugal' and
ref_produit=2;  --  Fonctionne : la commande UPDATE concerne ici le
fragment stockES
UPDATE Stock set unites_stock=4 where pays='INEX' and ref_produit=null;
--Fonctionne, mais MAJ de 0 ligne
update stock set INDISPONIBLE=1 where pays='Allemagne' and ref_produit=1;
--Fonctionne : la commande UPDATE concerne ici le fragment stockEN
```

```
delete from stock where pays='Portugal' and ref_produit=2;      --
Fonctionne : la comande DELETE concerne le fragment stockES
delete from stock where pays= 'Honduras' and ref_produit=1;     --
Fonctionne, mais suppression de 0 ligne

rollback;          --  Rollback jusqu'avant l'insertion du produit 2 au
Portugal

/*
  Vue Clients : sélection
*/
Select * from Clients;    --Requête standard

Select * from Clients
where code_client NOT IN
(Select code_client from Clients natural join Commandes);     -- Jointure :
clients n'ayant fait aucune commande

Select * from Clients c1, Commandes c2, details_commandes dc, Produits p,
Stock s
WHERE c1.CODE_CLIENT = c2.code_client
and c2.no_commande = dc.no_commande
and dc.ref_produit = p.ref_produit and p.ref_produit = s.ref_produit
and c1.pays = s.pays;              -- Jointure : Clients ayant commandé un
produit en stock dans leur pays

/*
  Vue Clients : LMD
*/

insert into clients values ('54321', 'Testcomp', '1, Boulevard des
palmiers', 'Panama', '88855', 'Panama', '010203405', '0607080900');    --
Fonctionne : insère le tuple dans le fragment américain.
insert into clients values ('54322', 'Testcomp2', '1, rue de la Fayette',
'Dijon', '21000', 'France', '010203405', '0607080900');    --Fonctionne :
insère le tuple dans le fragment local européen du sud
insert into clients values (null,  'Testcomp', '1, Boulevard des palmiers',
'Panama', '88855', 'Panama', '010203405', '0607080900');  --Erreur:
violation de clé primaire
insert into commandes values ('66666', '54321', 3, SYSDATE, sysdate, 12);
---Fonctionne : la commande se trouve sur le fragment américain
delete from clients where code_client='54321';    -- Erreur : violation de
contrainte d'intégrité assurée par un trigger distant sur la base
américaine.
delete from commandes where no_commande='66666';  -- Fonctionne, commande
non référencée dans une table secondaire

delete from clients where code_client = '54321';      --Fonctionne :
supprime le client du fragment américain
update clients set code_client = null;        --Erreur : contrainte
d'intégrité de clé primaire
update clients set code_client = 'TESTX' where code_client = 'OTTIK';
--Ne fonctionne pas : le client allemand OTTIK est enregistré dans une
commande (FK du fragment ClientsEN)
delete from clients where code_client=null;   --Fonctionne mais ne supprime
aucun client (code_client est une PK)
```

```
rollback;       --Annulation

/*
  Vue Commandes : sélection
*/
select * from commandes;    --Sélection standard

select no_commande, code_client, (date_Envoi - date_Commande) as
dureeExpedition from commandes
where date_Envoi is not null and date_Commande is not null;     --Sélection
avec where (durée en jours du processus d'expédition des commandes par
commande)

select  as dureeExpedition from commandes
where date_Envoi is not null and date_Commande is not null;     --Sélection
avec where (durée du processus d'expédition des commandes par commande)

select sum(port) from commandes;        -- USage d'une fonction
d'agrégation sur un regroupement
ALTER TABLE commandesES ADD CONSTRAINT pk_commandesES PRIMARY KEY
(NO_COMMANDE);

/*
    Vue Commandes : LMD
*/


insert into commandes values (123457,'VINET',2,SYSDATE,SYSDATE,25); --
Fonctionne : commande destinée au fragment d'europe du Sud.
insert into commandes values (123457,'BLONP',2,SYSDATE,SYSDATE,25); -- Ne
fonctionne pas : violation de contrainte unique.

update commandes set code_client='OTTIK' where NO_COMMANDE='123457'; --
Fonctionne : OTTIK est allemand.
update commandes set code_client='ZIGGY' where NO_COMMANDE='123457'; -- Ne
fonctionne pas : ZIGGY est inconnu (peu importe le site)

delete from commandes where NO_COMMANDE='123457';   --OK : la commande est
liée à un client allemand (OTTIK)
delete from commandes where NO_COMMANDE='5555';         -- Comme en
centralisé, si aucun tuple n'est repéré, aucune erreur n'intervient.
insert into commandes values (159842, 'ANTON', 1, sysdate, sysdate, 2); ---
OK : le client Anton est Mexicain.
insert into details_commandes values(159842, 1, 2,3,4);   --OK : insertion
dans la table detailsCommandes

delete from commandes where NO_COMMANDE=159842;    --KO:  contrainte de clé
étrangère distante depuis DetailsCommande
delete from details_commandes where no_commande=159842;
delete from commandes where NO_COMMANDE=159842;  -- OK : La commande est
supprimée
rollback;

/*
  Vue DetailsCommande : Sélection
*/
select * from details_commandes;
select * from details_commandes where quantite >= 100;
```

```
/*
  Vue DetailsCommandes : LMD
*/
delete from details_commandes where NO_COMMANDE=10286 and ref_produit=3;
commit;
insert into details_Commandes values (10286, 3, 3, 4, 5);  --OK : insertion
dans le fragment details_commandesEN (Europe du Nord). LA commande provient
d'un client allemand
delete from commandes where no_commande=10286;          --KO : trigger de
violation de contrainte FK appelé à distance
delete from produits where ref_produit=3;       -- KO : trigger de
violation de contrainte local portant sur la table primaire Produits
insert into details_commandes values (444,2,54,1,1.2); --KO : le client de
la commande 444 est inconnu car la commande 444 n'existe pas
insert into details_Commandes values (10286, 888, 3, 4, 5);  --KO : le
produit spécifié est inexistant
update details_commandes set quantite=15 where NO_COMMANDE=10286 and
ref_produit=3;   --OK : La MAJ est appliquée sur le fragment local
update details_commandes set quantite=20 where NO_COMMANDE=10522 and
ref_produit=40;   --OK : La MAJ est appliquée sur un fragment distane
(Europe du Nord)
update details_commandes set no_commande = 122 where no_commande=123457; --
OK : la commande n'existe pas mais la mise à jour n'affecte donc aucune
ligne
delete from details_commandes where no_commande = 122;      -- OK mais ne
supprime rien car la commande 122 est absente de la table

rollback;



/*
  Vue Fournisseurs : sélection
*/

SELECT * FROM Fournisseurs;     --Sélection standard
SELECT * from Fournisseurs where societe LIKE '%Ltd%';    --Sélection sur
la table
SELECT avg(prix_unitaire) Moy, NO_Fournisseur from Fournisseurs natural
join Produits
group by no_Fournisseur order by moy; -- Jointure naturelle : prix unitaire
moyen de chacun des produits des fournisseurs

/*
  Vue Fournisseurs : LMD
*/
insert into Fournisseurs values (444, 'fourn', 'test', 'test', '4444',
'Australie', '3615', '000000');    --Insertion OK
insert into Fournisseurs values (445, 'fourn', null, 'test', '4444',
'Australie', '3615', '000000');    --Erreur : L'adresse ne doit pas être
inconnue
insert into Fournisseurs values (445, 'fourn', 'test', 'test', null,
'Australie', '3615', '000000');    --Erreur : Le code postal ne doit pas
être inconnue
insert into Produits values (4454, 'test', 444, 1, 5, 2);         --
Insertion OK
```

```
delete from Fournisseurs where NO_FOURNISSEUR = 444;          -- Erreur :
FK assurée par trigger distant (sur le site de l'europe du nord)
update Fournisseurs set no_fournisseur=6565 where no_fournisseur = 444; --
Erreur : clé référencée dans la table produits
update Produits set no_fournisseur=1 where ref_produit=4454;    --Update OK
update Fournisseurs set no_fournisseur=6565 where no_fournisseur = 444; --
Update OK
delete from Fournisseurs where NO_FOURNISSEUR = 6565;          -- Delete OK
rollback;


/*
  Vue Employés : Sélection
*/

select * from Employes;
select trunc((Date_Embauche - Date_naissance)/365.2425 ) ageEmploi,
No_employe, Nom, Prenom from Employes;
select * from Employes natural join commandes;

/*
  Vue Employés : LMD
*/
desc employes;
rollback;
insert into employes values (4545, null, 'test', 'test2', 'test3', 'uzaiy',
sysdate, sysdate, 4500, 210);   --Insertion OK
insert into employes values (4545, null, 'test2', 'test2', 'test3',
'uzaiy', sysdate, sysdate, 4500, 210);   --KO : non respect de la clé
primaire
insert into employes values (4546, null, null, 'test2', 'test3', 'uzaiy',
sysdate, sysdate, 4500, 210);   --KO : insertion nulle non permise sur NOM
insert into employes values (4546, null, 'test', 'test2', 'test3', null,
sysdate, sysdate, 4500, 210);   --KO : insertion nulle non permise sur
TITRE

insert into Commandes values (88775, 'OTTIK', 4545, sysdate, sysdate, 45);
--Insertion OK
delete from employes where no_employe=4545;   --Erreur : l'employé est déjà
référencé sur une commande
update employes set no_employe=4546 where no_employe=4545;   --Erreur :
l'employé est déjà référencé sur une commande
delete from commandes where no_commande=88775;
update employes set no_employe=4546 where no_employe=4545;   --Erreur :
l'employé est déjà référencé sur une commande
delete from employes where no_employe=4546;   --Erreur : l'employé est déjà
référencé sur une commande
rollback;

/*
  Table locale Produits : LMD
*/
insert into Produits values (555, null, 5, 2, null, 3); -- Erreur :
contrainte chk_not_null de l'attribut nom_produit
insert into Produits values (555, 'nomTest', 5, 2, null, 3); --Insertion OK
update Produits set CODE_CATEGORIE = 9999;      --Erreur : violation de
contrainte d'intégrité. La catégorie 9999 n'existe pas
rollback;
```

```
/*
  Table locale Catégorie : LMD
*/

insert into Categories values(4444, 'test', null);  --Erreur : la
description ne doit pas être nulle.
insert into Categories values(4444, null, 'test');  --Erreur : le nom ne
doit pas être null.
insert into Categories values(4444, 'test', 'desc');   --Insertion OK
insert into Produits values (8989, 'ttt', 1, 4444, 5, 8);   --Insertion OK
delete from Categories where code_Categorie=4444;     --KO : violation de
cintrainte d'intégrité
update Categories set description = 'aabbcc' where code_categorie=4444; ---
Update OK
update PRoduits set code_categorie = 1 where REF_PRODUIT=8989;      --
Update OK
delete from Categories where code_categorie=4444;     --  Suppression OK :
plus aucune référence à 4444 dans la table secondaire
rollback;
```