

Système de Gestion de Base de Données

TP n°3

Bases de Données Distribuées : Ryori

B3201 : Annie ABHAY et Sophanna NGOV

B3202 : Benoît ALGOURDIN et Grégoire De
LAMBERTYE

B3203 : Adam CHELLAOUI et Rim BENZEKRI

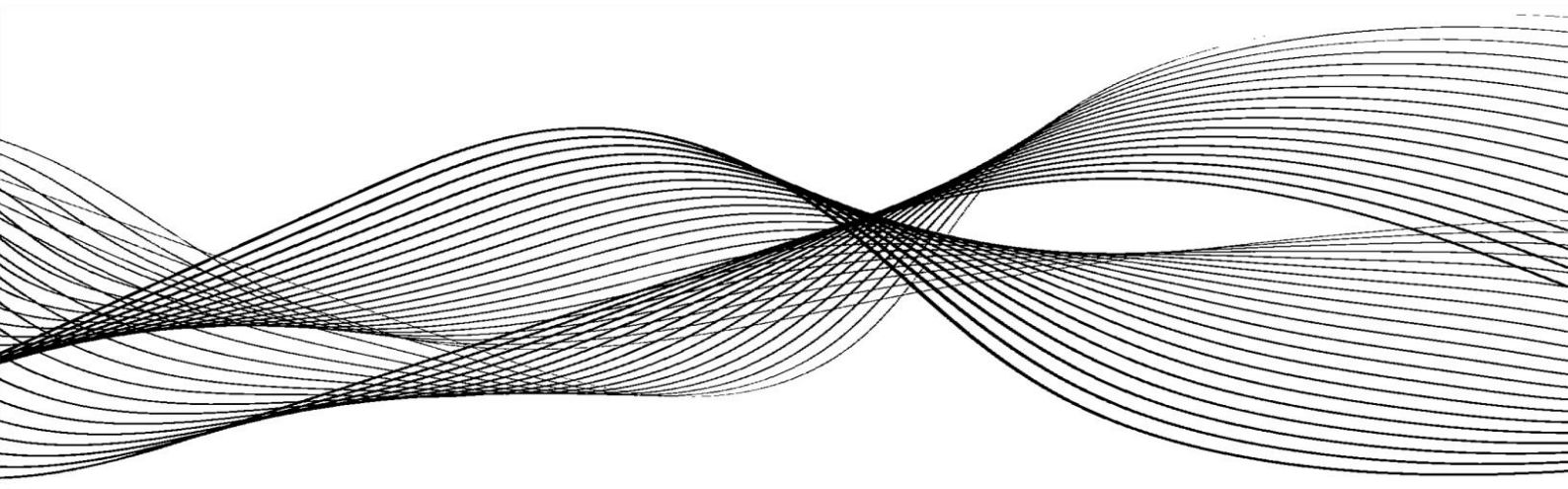


Table des matières

1. Objectifs du TP	4
2. Répartition des binômes sur les tâches	4
2.1 Répartition des rôles du groupe.....	4
2.2 Répartitions des sites pour la base de données distribuée.....	4
3. Distribution de la base de données.....	4
3.1 Fragmentation primaire	4
3.1.1 Table Clients	4
3.1.2 Table Stock	5
3.2 Fragmentation dérivée	6
3.2.1 Table Commandes	6
3.2.2 Table Détails_Commandes	6
3.3 Résumé de la fragmentation	6
3.4 Répartition des fragments.....	7
3.4.1 Utilisation des fragments par les sites.....	7
3.4.2 Répartition des fragments sur les sites	7
4. Conception des bases de données sur chaque site	8
4.1 Europe du Nord et reste du monde	8
4.2 Europe du Sud.....	17
4.3 Amérique.....	22
5. Tests de requête distribuées et optimisations	30
5.1 Europe du Nord et reste du monde	30
5.2 Europe du Sud.....	31
5.3 Amérique.....	32
6. Réplifications.....	34
6.1 Mise en œuvre des réplifications sur le site Europe du Nord.....	34
6.2 Mise en œuvre des réplifications sur le site Europe du Sud	37
Liste des réplifications prévues	37
6.3 Mise en œuvre des réplifications sur le site Amérique	40
Liste des réplifications prévues	40
6.4 Bilan global des réplifications mises en œuvre sur les différents sites. (Tableau de synthèse, etc.).....	43
7. Requetes distribuées : tests et optimisations	43
7.1 Site Europe du Nord.....	43

7.2	Site Europe du Sud	43
7.3	Site Amérique	44

1. Objectifs du TP

Une entreprise basée sur plusieurs continents gère une base de données centralisée. Notre objectif est de lui proposer une solution distribuée en répartissant les fragments selon les usages les plus fréquents des différents sites.

Cette solution distribuée devra respecter des contraintes de cohérence et d'intégrité des données pour ne pas perdre de données lors de la fragmentation et de la répartition. Nous devons aussi optimiser les performances de la base de données distribuée ainsi que laisser à l'utilisateur une transparence pour qu'il puisse réaliser les mêmes requêtes que lorsqu'il se trouvait sur la base de données centralisée et obtenir les mêmes résultats.

2. Répartition des binômes sur les tâches

2.1 Répartition des rôles du groupe

- Chef de projet : Sophanna NGOV
- Responsable de documentation : Grégoire De LAMBERTYE

2.2 Répartitions des sites pour la base de données distribuée

- 3201 : Annie ABHAY et Sophanna NGOV gèrent l'Europe du Nord
- 3202 : Benoît ALGOURDIN et Grégoire De LAMBERTYE gèrent l'Europe du Sud
- 3203 : Adam CHELLAOUI et Rim BENZEKRI gèrent l'Amérique

3. Distribution de la base de données

La distribution de la base de données a été faite de manière collégiale pour obtenir un résultat optimal et satisfaisant pour tous les membres du groupe. Cette partie est la partie la plus importante pour pouvoir répartir les tâches aux différentes personnes mais aussi comprendre comment sera fragmentée la base.

3.1 Fragmentation primaire

À la suite de notre réflexion nous avons pu déterminer des prédicats discriminants pour différentes tables.

3.1.1 Table Clients

Prédicats :

- Clients.Pays ∈ Europe du Sud (ES)
- Clients.Pays ∈ Europe du Nord (EN)
- Clients.Pays ∈ Amérique (A)

Combinaisons possibles :

Clients.Pays ∈ A	Clients.Pays ∈ ES	Clients.Pays ∈ EN
Clients.Pays ∈ A	Clients.Pays ∈ ES	Clients.Pays ∈ EN
\neg (Clients.Pays ∈ A)	Clients.Pays ∈ ES	Clients.Pays ∈ EN
Clients.Pays ∈ A	\neg (Clients.Pays ∈ ES)	Clients.Pays ∈ EN
\neg (Clients.Pays ∈ A)	\neg (Clients.Pays ∈ ES)	Clients.Pays ∈ EN
Clients.Pays ∈ A	Clients.Pays ∈ ES	\neg (Clients.Pays ∈ EN)
\neg (Clients.Pays ∈ A)	Clients.Pays ∈ ES	\neg (Clients.Pays ∈ EN)
Clients.Pays ∈ A	\neg (Clients.Pays ∈ ES)	\neg (Clients.Pays ∈ EN)
\neg (Clients.Pays ∈ A)	\neg (Clients.Pays ∈ ES)	\neg (Clients.Pays ∈ EN)

Nous obtenons donc 4 combinaisons possibles :

- $(\neg \text{Clients.Pays} \in A) \wedge (\neg \text{Clients.Pays} \in ES) \wedge \text{Clients.Pays} \in EN$
- $(\neg \text{Clients.Pays} \in A) \wedge (\text{Clients.Pays} \in ES) \wedge (\neg \text{Clients.Pays} \in EN)$
- $(\text{Clients.Pays} \in A) \wedge (\neg \text{Clients.Pays} \in ES) \wedge (\neg \text{Clients.Pays} \in EN)$
- $(\neg \text{Clients.Pays} \in A) \wedge (\neg \text{Clients.Pays} \in ES) \wedge (\neg \text{Clients.Pays} \in EN)$

Ces combinaisons nous donnent respectivement 4 fragments : Clients_EN, Clients_ES, Clients_A et Clients_RM

3.1.2 Table Stock

Prédicats :

- Stock.Pays ∈ Europe du Sud (ES)
- Stock.Pays ∈ Europe du Nord (EN)
- Stock.Pays ∈ Amérique (A)

Combinaisons possibles :

Stock.Pays ∈ A	Stock.Pays ∈ ES	Stock.Pays ∈ EN
Stock.Pays ∈ A	Stock.Pays ∈ ES	Stock.Pays ∈ EN
\neg (Stock.Pays ∈ A)	Stock.Pays ∈ ES	Stock.Pays ∈ EN
Stock.Pays ∈ A	\neg (Stock.Pays ∈ ES)	Stock.Pays ∈ EN
\neg (Stock.Pays ∈ A)	\neg (Stock.Pays ∈ ES)	Stock.Pays ∈ EN
Stock.Pays ∈ A	Stock.Pays ∈ ES	\neg (Stock.Pays ∈ EN)
\neg (Stock.Pays ∈ A)	Stock.Pays ∈ ES	\neg (Stock.Pays ∈ EN)
Stock.Pays ∈ A	\neg (Stock.Pays ∈ ES)	\neg (Stock.Pays ∈ EN)
\neg (Stock.Pays ∈ A)	\neg (Stock.Pays ∈ ES)	\neg (Stock.Pays ∈ EN)

Nous obtenons donc 4 combinaisons possibles :

- $(\neg \text{Stock.Pays} \in A) \wedge (\neg \text{Stock.Pays} \in ES) \wedge \text{Stock.Pays} \in EN$
- $(\neg \text{Stock.Pays} \in A) \wedge (\text{Stock.Pays} \in ES) \wedge (\neg \text{Stock.Pays} \in EN)$
- $(\text{Stock.Pays} \in A) \wedge (\neg \text{Stock.Pays} \in ES) \wedge (\neg \text{Stock.Pays} \in EN)$
- $(\neg \text{Stock.Pays} \in A) \wedge (\neg \text{Stock.Pays} \in ES) \wedge (\neg \text{Stock.Pays} \in EN)$

Ces combinaisons nous donnent respectivement 4 fragments : Stock_EN, Stock_ES, Stock_A et Stock_RM

3.2 Fragmentation dérivée

Les tables précédemment fragmentées sont directement liées à d'autres tables, il faut donc les fragmenter aussi.

3.2.1 Table Commandes

- Commandes_A= Commandes \bowtie Clients_A
- Commandes_ES= Commandes \bowtie Clients_ES
- Commandes_EN= Commandes \bowtie Clients_EN
- Commandes_RM= Commandes \bowtie Clients_RM

3.2.2 Table Détails_Commandes

- Détails_Commandes_A= Détails_Commandes \bowtie Commandes_A
- Détails_Commandes_ES= Détails_Commandes \bowtie Commandes_ES
- Détails_Commandes_EN= Détails_Commandes \bowtie Commandes_EN
- Détails_Commandes_RM= Détails_Commandes \bowtie Commandes_RM

3.3 Résumé de la fragmentation

La fragmentation que nous avons réalisée donne finalement les 20 fragments suivants :

- Clients_EN
- Clients_ES
- Clients_A
- Clients_RM
- Stock_EN
- Stock_ES
- Stock_A
- Stock_RM
- Commandes_A
- Commandes_ES
- Commandes_EN
- Commandes_RM
- Détails_Commandes_A
- Détails_Commandes_ES
- Détails_Commandes_EN
- Détails_Commandes_RM
- Produits
- Catégories
- Fournisseurs
- Employés

3.4 Répartition des fragments

3.4.1 Utilisation des fragments par les sites

La répartition des fragments se fait selon l'utilité que chaque site aura des fragments.

Fragments	Europe du Sud	Europe du Nord	Amérique
Clients_EN	0	++	0
Clients_ES	++	0	0
Clients_A	0	0	++
Clients_RM	0	++	0
Stock_EN	+	++	+
Stock_ES	++	+	+
Stock_A	+	+	++
Stock_RM	+	++	+
Commandes_A	0	0	++
Commandes_ES	++	0	0
Commandes_EN	0	++	0
Commandes_RM	0	++	0
Details_Commandes_A	0	0	++
Details_Commandes_ES	++	0	0
Details_Commandes_EN	0	++	0
Details_Commandes_RM	0	++	0
Produits	++	+	+
Catégories	++	+	+
Fournisseurs	+	++	+
Employés	+	+	++

Tableau 1: Probabilités d'accès aux fragments

3.4.2 Répartition des fragments sur les sites

Fragments	Europe du Sud	Europe du Nord	Amérique
Clients_EN		X	
Clients_ES	X		
Clients_A			X
Clients_RM		X	
Stock_EN		X	
Stock_ES	X		
Stock_A			X
Stock_RM		X	
Commandes_A			X
Commandes_ES	X		
Commandes_EN		X	
Commandes_RM		X	
Details_Commandes_A			X
Details_Commandes_ES	X		
Details_Commandes_EN		X	
Details_Commandes_RM		X	
Produits	X		
Catégories	X		
Fournisseurs		X	
Employés			X

Tableau 2 : Répartition des fragments sur les sites

4. Conception des bases de données sur chaque site

4.1 Europe du Nord et reste du monde

Le binôme responsable de la Base d'Europe du Nord est le B3201 composé d'Annie ABHAY et Sophanna NGOV.

Création des liens entre les bases	<pre>CREATE database link ConnexionRyori connect to sngov identified by XXXXXXXX USING 'DB11'; CREATE database link ConnexionEurS connect to sngov identified by XXXXXXXX USING 'DB13'; CREATE database link ConnexionAmer connect to sngov identified by XXXXXXXX USING 'DB14';</pre>
Création et peuplement des tables	<pre>-----Création et peuplement de Fournisseurs----- CREATE TABLE FOURNISSEURS AS SELECT * FROM RYORI.FOURNISSEURS @CONNEXIONDB11; -----Création et peuplement de Client_EN----- SELECT * FROM RYORI.CLIENTS @CONNEXIONDB11 WHERE PAYS IN ('Allemagne','Irlande','Norvege','Suede','Danemark', 'Islande','Belgique','Luxembourg','Pays-Bas','Pologne','Finlande', 'Royaume-Uni'); -----Création et peuplement de Client_RM----- CREATE TABLE Clients_RM AS --Reste du Monde (SELECT * FROM RYORI.CLIENTS @CONNEXIONDB11 WHERE PAYS NOT IN ('Norvege', 'Suede', 'Danemark', 'Islande', 'Finlande', 'Royaume-Uni', 'Irlande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Allemagne', 'Pologne', 'Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie- Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie', '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')); -----Création et peuplement de Stock_EN----- CREATE TABLE Stock_EN AS(SELECT * FROM RYORI.STOCK @CONNEXIONDB11 WHERE PAYS IN ('Allemagne','Irlande','Norvege','Suede','Danemark','Islande','Belgique','Luxembourg','Pays- Bas','Pologne','Finlande','Royaume-Uni')); -----Création et peuplement de Stock_RM----- CREATE TABLE Stock_RM --Reste du Monde</pre>

	<pre> AS (SELECT * FROM RYORI.STOCK @CONNEXIONDB11 WHERE PAYS NOT IN ('Norvege', 'Suede', 'Danemark', 'Islande', 'Finlande', 'Royaume-Uni', 'Irlande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Allemagne', 'Pologne', 'Espagne', 'Portugal', 'Andorre','France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie- Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie', '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')); -----Création et peuplement de Commandes_EN----- CREATE TABLE Commandes_EN AS (SELECT * FROM ryori.commandes@CONNEXIONDB11 WHERE CODE_CLIENT IN (SELECT CODE_CLIENT FROM Clients_EN)); -----Création et peuplement de Commandes_RM----- CREATE TABLE Commandes_RM AS (SELECT * FROM ryori. commandes@CONNEXIONDB11 WHERE CODE_CLIENT IN (SELECT CODE_CLIENT FROM Clients_RM)); -----Création et peuplement de Detail_Commandes_EN----- CREATE TABLE Detail_Commandes_EN AS (SELECT * FROM ryori. details_commandes@CONNEXIONDB11 WHERE NO_COMMANDE IN (SELECT NO_COMMANDE FROM Commandes_EN)); -----Création et peuplement de Detail_Commandes_RM----- CREATE TABLE Detail_Commandes_RM AS (SELECT * FROM ryori. details_commandes@CONNEXIONDB11 WHERE NO_COMMANDE IN (SELECT NO_COMMANDE FROM Commandes_RM)); </pre>
Contraintes d'intégrité	<pre> -----Création des Clés primaires----- ALTER TABLE FOURNISSEURS ADD CONSTRAINT pk_Fournisseurs PRIMARY KEY(NO_FOURNISSEUR); ALTER TABLE CLIENTS_EN ADD CONSTRAINT pk_ClientEN PRIMARY KEY (CODE_CLIENT); ALTER TABLE CLIENTS_RM ADD CONSTRAINT pk_ClientRM PRIMARY KEY (CODE_CLIENT); ALTER TABLE COMMANDES_EN ADD CONSTRAINT pk_CommandesEN PRIMARY KEY (NO_COMMANDE); ALTER TABLE COMMANDES_RM ADD CONSTRAINT pk_CommandesRM PRIMARY KEY (NO_COMMANDE); ALTER TABLE DETAIL_COMMANDES_EN ADD CONSTRAINT pk_DCommandesEN </pre>

	<pre> PRIMARY KEY (NO_COMMANDE, REF_PRODUIT); ALTER TABLE DETAIL_COMMANDES_RM ADD CONSTRAINT pk_DCommandesRM PRIMARY KEY (NO_COMMANDE, REF_PRODUIT); ALTER TABLE STOCK_EN ADD CONSTRAINT pk_StockEN PRIMARY KEY (REF_PRODUIT, PAYS); ALTER TABLE STOCK_RM ADD CONSTRAINT pk_StockRM PRIMARY KEY (REF_PRODUIT, PAYS); -----Création des Clés étrangères----- ALTER TABLE COMMANDES_EN ADD CONSTRAINT fk_CommCliEN FOREIGN KEY (CODE_CLIENT) REFERENCES CLIENTS_EN(CODE_CLIENT); ALTER TABLE COMMANDES_RM ADD CONSTRAINT fk_CommCliRM FOREIGN KEY (CODE_CLIENT) REFERENCES CLIENTS_RM(CODE_CLIENT); ALTER TABLE DETAIL_COMMANDES_EN ADD CONSTRAINT fk_DCommCommEN FOREIGN KEY (NO_COMMANDE) REFERENCES COMMANDES_EN(NO_COMMANDE); ALTER TABLE DETAIL_COMMANDES_RM ADD CONSTRAINT fk_DCommCommRM FOREIGN KEY (NO_COMMANDE) REFERENCES COMMANDES_RM(NO_COMMANDE); -----Création des clés étrangères sur des tables distantes (TRIGGER)----- -----Trigger remplaçant la clé étrangère de COMMANDES_EN sur le champ NO_EMPLOYE de la table EMPLOYES : CREATE OR REPLACE TRIGGER fk_CommEmployEN BEFORE INSERT ON COMMANDES_EN FOR EACH ROW DECLARE i NUMBER; BEGIN SELECT COUNT(*) INTO i FROM achellaoui.EMPLOYES@CONNEXIONAMER WHERE NO_EMPLOYE = :NEW.NO_EMPLOYE; IF i=0 THEN RAISE_APPLICATION_ERROR(-20001,'Employé inexistant'); END IF; END; -----Trigger remplaçant la clé étrangère de COMMANDES_RM sur le champ NO_EMPLOYE de la table EMPLOYES: CREATE OR REPLACE TRIGGER fk_CommEmployRM BEFORE INSERT ON COMMANDES_RM FOR EACH ROW DECLARE i NUMBER; </pre>
--	---

```
BEGIN
SELECT COUNT(*) INTO i
FROM achellaoui.EMPLOYES@CONNEXIONAMER
WHERE NO_EMPLOYE = :NEW.NO_EMPLOYE;
IF i=0
THEN RAISE_APPLICATION_ERROR(-20001,'Employé inexistant');
END IF;
END;

-----Trigger remplaçant la clé étrangère de DETAIL_COMMANDES_EN sur le champ
REF_PRODUIT de la table PRODUITS:

CREATE OR REPLACE TRIGGER fk_DCommProdEN
BEFORE INSERT ON DETAIL_COMMANDES_EN
FOR EACH ROW
DECLARE
i NUMBER;
BEGIN
SELECT COUNT(*) INTO i
FROM gdelambert.produits@CONNEXIONEURS
WHERE REF_PRODUIT = :NEW.REF_PRODUIT;
IF i=0
THEN RAISE_APPLICATION_ERROR(-20001,'Référence de produit inexistante');
END IF;
END;

-----Trigger remplaçant la clé étrangère de DETAIL_COMMANDES_RM sur le champ
REF_PRODUIT de la table PRODUITS:

CREATE OR REPLACE TRIGGER fk_DCommProdRM
BEFORE INSERT ON DETAIL_COMMANDES_RM
FOR EACH ROW
DECLARE
i NUMBER;
BEGIN
SELECT COUNT(*) INTO i
FROM gdelambert.produits@CONNEXIONEURS
WHERE REF_PRODUIT = :NEW.REF_PRODUIT;
IF i=0
THEN RAISE_APPLICATION_ERROR(-20001,'Référence de produit inexistante');
END IF;
END;

-----Trigger remplaçant la clé étrangère de STOCK_EN sur le champ REF_PRODUIT de la table
PRODUITS:

CREATE OR REPLACE TRIGGER fk_StockProdEN
BEFORE INSERT ON STOCK_EN
FOR EACH ROW
DECLARE
i NUMBER;
BEGIN
```

	<pre> SELECT COUNT(*) INTO i FROM gdelambert.produits@CONNEXIONEURS WHERE REF_PRODUIT = :NEW.REF_PRODUIT; IF i=0 THEN RAISE_APPLICATION_ERROR(-20001,'Référence de produit inexistante'); END IF; END; -----Trigger remplaçant la clé étrangère de STOCK_RM sur le champ REF_PRODUIT de la table PRODUITS: CREATE OR REPLACE TRIGGER fk_StockProdRM BEFORE INSERT ON STOCK_RM FOR EACH ROW DECLARE i NUMBER; BEGIN SELECT COUNT(*) INTO i FROM gdelambert.produits@CONNEXIONEURS WHERE REF_PRODUIT = :NEW.REF_PRODUIT; IF i=0 THEN RAISE_APPLICATION_ERROR(-20001,'Référence de produit inexistante'); END IF; END; -----Contraintes de type CHECK PAYS----- ALTER TABLE CLIENTS_EN ADD CONSTRAINT Pays_ClientEN CHECK(PAYS IN ('Allemagne','Irlande','Norvege','Suede','Danemark','Islande','Belgique','Luxembourg','Pays- Bas','Pologne','Finlande','Royaume-Uni')); ALTER TABLE CLIENTS_RM ADD CONSTRAINT Pays_ClientRM CHECK(PAYS NOT IN ('Norvege', 'Suede', 'Danemark', 'Islande', 'Finlande', 'Royaume-Uni', 'Irlande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Allemagne', 'Pologne', 'Espagne', 'Portugal', 'Andorre','France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie- Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie', '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','Jamaïque','Mexique','Nicaragua','Panama ','Paraguay','Perou','Saint-Christophe-et-Nieves', 'Sainte-Lucie','Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago','Uruguay','Venezuela')); ALTER TABLE STOCK_EN ADD CONSTRAINT Pays_StockEN CHECK(PAYS IN ('Allemagne','Irlande','Norvege','Suede','Danemark','Islande','Belgique','Luxembourg','Pays- Bas','Pologne','Finlande','Royaume-Uni')); </pre>
--	--

	<pre> ALTER TABLE STOCK_RM ADD CONSTRAINT Pays_StockRM CHECK(PAYS NOT IN ('Norvege', 'Suede', 'Danemark', 'Islande', 'Finlande', 'Royaume-Uni', 'Irlande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Allemagne', 'Pologne', 'Espagne', 'Portugal', 'Andorre','France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie- Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie', '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')); -----Contraintes de type CHECK IS NOT NULL----- ALTER TABLE FOURNISSEURS ADD CONSTRAINT NN_Fournisseurs CHECK(NO_FOURNISSEUR IS NOT NULL AND SOCIETE IS NOT NULL AND ADRESSE IS NOT NULL AND VILLE IS NOT NULL AND CODE_POSTAL IS NOT NULL AND PAYS IS NOT NULL AND TELEPHONE IS NOT NULL); ALTER TABLE CLIENTS_EN ADD CONSTRAINT NN_ClientEN CHECK(CODE_CLIENT IS NOT NULL AND SOCIETE IS NOT NULL AND ADRESSE IS NOT NULL AND VILLE IS NOT NULL AND CODE_POSTAL IS NOT NULL AND PAYS IS NOT NULL AND TELEPHONE IS NOT NULL); ALTER TABLE CLIENTS_RM ADD CONSTRAINT NN_ClientRM CHECK(CODE_CLIENT IS NOT NULL AND SOCIETE IS NOT NULL AND ADRESSE IS NOT NULL AND VILLE IS NOT NULL AND CODE_POSTAL IS NOT NULL AND PAYS IS NOT NULL AND TELEPHONE IS NOT NULL); ALTER TABLE COMMANDES_EN ADD CONSTRAINT NN_CommandesEN CHECK(NO_COMMANDE IS NOT NULL AND CODE_CLIENT IS NOT NULL AND NO_EMPLOYE IS NOT NULL AND DATE_COMMANDE IS NOT NULL AND PORT IS NOT NULL); ALTER TABLE COMMANDES_RM ADD CONSTRAINT NN_CommandesRM CHECK(NO_COMMANDE IS NOT NULL AND CODE_CLIENT IS NOT NULL AND NO_EMPLOYE IS NOT NULL AND DATE_COMMANDE IS NOT NULL AND PORT IS NOT NULL); ALTER TABLE DETAIL_COMMANDES_EN ADD CONSTRAINT NN_DCommandesEN CHECK(NO_COMMANDE IS NOT NULL AND REF_PRODUIT IS NOT NULL AND PRIX_UNITAIRE IS NOT NULL AND QUANTITE IS NOT NULL AND REMISE IS NOT NULL); ALTER TABLE DETAIL_COMMANDES_RM ADD CONSTRAINT NN_DCommandesRM CHECK(NO_COMMANDE IS NOT NULL AND REF_PRODUIT IS NOT NULL AND PRIX_UNITAIRE IS NOT NULL AND QUANTITE IS NOT NULL AND REMISE IS NOT NULL); ALTER TABLE STOCK_EN ADD CONSTRAINT NN_StockEN CHECK(REF_PRODUIT IS NOT NULL AND PAYS IS NOT NULL); ALTER TABLE STOCK_RM ADD CONSTRAINT NN_StockRM CHECK(REF_PRODUIT IS NOT NULL AND PAYS IS NOT NULL); </pre>
--	---

Droits d'accès	<p>GRANT select on CLIENTS_EN to gdelambert; GRANT select on CLIENTS_EN to achellaoui;</p> <p>GRANT select on CLIENTS_RM to gdelambert; GRANT select on CLIENTS_RM to achellaoui;</p> <p>GRANT select on COMMANDES_EN to gdelambert; GRANT select on COMMANDES_EN to achellaoui;</p> <p>GRANT select on COMMANDES_RM to gdelambert; GRANT select on COMMANDES_RM to achellaoui;</p> <p>GRANT select on Detail_Commandes_EN to gdelambert; GRANT select on Detail_Commandes_EN to achellaoui;</p> <p>GRANT select on Detail_Commandes_RM to gdelambert; GRANT select on Detail_Commandes_RM to achellaoui;</p> <p>GRANT select on STOCK_EN to gdelambert; GRANT select on STOCK_EN to achellaoui;</p> <p>GRANT select on STOCK_RM to gdelambert; GRANT select on STOCK_RM to achellaoui;</p> <p>GRANT select on FOURNISSEURS to gdelambert; GRANT select on FOURNISSEURS to achellaoui;</p>
Définition de synonymes et de vues pour interrogation de la base comme si elle était en centralisé	<p>----- CREATION DES VUES -----</p> <p>----- Vue CLIENTS -----</p> <p>CREATE OR REPLACE VIEW CLIENTS AS (SELECT* FROM clients_en WHERE PAYS = 'Allemagne' or pays = 'Irlande' or pays = 'Norvege' or pays = 'Suede' or pays = 'Danemark' or pays = 'Islande' or pays = 'Belgique' or pays = 'Luxembourg' or pays = 'Pays-Bas' or pays = 'Pologne' or pays = 'Finlande' or pays = 'Royaume-Uni') UNION ALL (SELECT* FROM gdelambert.clients_es@CONNEXIONEURS WHERE PAYS = 'Espagne' OR PAYS = 'Portugal' OR PAYS = 'Andorre' OR PAYS = 'France' OR PAYS = 'Gibraltar' OR PAYS = 'Italie' OR PAYS = 'Saint-Marin' OR PAYS = 'Vatican' OR PAYS = 'Malte' OR PAYS = 'Albanie' OR PAYS = 'Bosnie-Herzegovine' OR PAYS = 'Croatie' OR PAYS = 'Grece' OR PAYS = 'Macedoine' OR PAYS = 'Montenegro' OR PAYS = 'Serbie' OR PAYS = 'Slovenie' OR PAYS = 'Bulgarie') UNION ALL (SELECT* FROM achellaoui.clients_a@CONNEXIONAmer WHERE PAYS = 'Antigua-et-Barbuda' OR PAYS = 'Argentine' OR PAYS = 'Bahamas' OR PAYS = 'Barbade'</p>

	<p>OR PAYS = 'Belize' OR PAYS = 'Bolivie' OR PAYS = 'Bresil' OR PAYS = 'Canada' OR PAYS = 'Chili' OR PAYS = 'Colombie'</p> <p>OR PAYS = 'Costa Rica' OR PAYS = 'Cuba' OR PAYS = 'Republique dominicaine' OR PAYS = 'Dominique' OR PAYS = 'Equateur'</p> <p>OR PAYS = 'Etats-Unis' OR PAYS = 'Grenade' OR PAYS = 'Guatemala' OR PAYS = 'Guyana' OR PAYS = 'Haiti' OR PAYS = 'Honduras'</p> <p>OR PAYS = 'Jamaique' OR PAYS = 'Mexique' OR PAYS = 'Nicaragua' OR PAYS = 'Panama' OR PAYS = 'Paraguay' OR PAYS = 'Perou'</p> <p>OR PAYS = 'Saint-Christophe-et-Nieves' OR PAYS = 'Sainte-Lucie' OR PAYS = 'Saint-Vincent-et-les Grenadines' OR PAYS = 'Salvador' OR PAYS = 'Suriname' OR PAYS = 'Trinite-et-Tobago' OR PAYS = 'Uruguay' OR PAYS = 'Venezuela'</p> <p>)</p> <p>UNION ALL</p> <p>(SELECT* from clients_Rm</p> <p>WHERE PAYS NOT IN ('Norvege', 'Suede', 'Danemark', 'Islande', 'Finlande', 'Royaume-Uni', 'Irlande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Allemagne', 'Pologne', 'Espagne', 'Portugal', 'Andorre','France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie', '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')</p> <p>);</p> <p>----- Vue COMMANDES -----</p> <p>CREATE OR REPLACE VIEW COMMANDES AS</p> <p>SELECT * FROM COMMANDES_EN</p> <p>UNION ALL</p> <p>SELECT * FROM COMMANDES_RM</p> <p>UNION ALL</p> <p>SELECT * FROM gdelambert.commandes_es@CONNEXIONEURS</p> <p>UNION ALL</p> <p>SELECT * FROM achellaoui.commandes_a@CONNEXIONAMER;</p> <p>-----Vue DETAILS_COMMANDES-----</p> <p>CREATE OR REPLACE VIEW DETAILS_COMMANDES AS</p> <p>SELECT * FROM DETAIL_COMMANDES_EN</p> <p>UNION ALL</p> <p>SELECT * FROM DETAIL_COMMANDES_RM</p> <p>UNION ALL</p> <p>SELECT * FROM gdelambert.details_commandes_es@CONNEXIONEURS</p> <p>UNION ALL</p> <p>SELECT * FROM achellaoui.detail_commandes_a@CONNEXIONAMER;</p> <p>-----Vue STOCK-----</p> <p>CREATE OR REPLACE VIEW STOCK AS</p> <p>(SELECT* FROM stock_en</p>
--	---

	<p>WHERE PAYS = 'Allemagne' or pays = 'Irlande' or pays = 'Norvege' or pays = 'Suede' or pays = 'Danemark' or pays = 'Islande' or pays = 'Belgique' or pays = 'Luxembourg' or pays = 'Pays-Bas' or pays = 'Pologne' or pays = 'Finlande' or pays = 'Royaume-Uni'</p> <p>)</p> <p>UNION ALL</p> <p>(SELECT* FROM gdelambert.stock_es@CONNEXIONEURS</p> <p>WHERE PAYS = 'Espagne' OR PAYS = 'Portugal' OR PAYS = 'Andorre' OR PAYS = 'France' OR PAYS = 'Gibraltar' OR PAYS = 'Italie' OR PAYS = 'Saint-Marin' OR PAYS = 'Vatican' OR PAYS = 'Malte' OR PAYS = 'Albanie' OR PAYS = 'Bosnie-Herzegovine' OR PAYS = 'Croatie' OR PAYS = 'Grece' OR PAYS = 'Macedoine' OR PAYS = 'Montenegro' OR PAYS = 'Serbie' OR PAYS = 'Slovenie' OR PAYS = 'Bulgarie'</p> <p>)</p> <p>UNION ALL</p> <p>(SELECT* FROM achellaoui.stock_a@CONNEXIONAmer</p> <p>WHERE PAYS = 'Antigua-et-Barbuda' OR PAYS = 'Argentine' OR PAYS = 'Bahamas' OR PAYS = 'Barbade'</p> <p>OR PAYS = 'Belize' OR PAYS = 'Bolivie' OR PAYS = 'Bresil' OR PAYS = 'Canada' OR PAYS = 'Chili' OR PAYS = 'Colombie'</p> <p>OR PAYS = 'Costa Rica' OR PAYS = 'Cuba' OR PAYS = 'Republique dominicaine' OR PAYS = 'Dominique' OR PAYS = 'Equateur'</p> <p>OR PAYS = 'Etats-Unis' OR PAYS = 'Grenade' OR PAYS = 'Guatemala' OR PAYS = 'Guyana' OR PAYS = 'Haiti' OR PAYS = 'Honduras'</p> <p>OR PAYS = 'Jamaique' OR PAYS = 'Mexique' OR PAYS = 'Nicaragua' OR PAYS = 'Panama' OR PAYS = 'Paraguay' OR PAYS = 'Perou'</p> <p>OR PAYS = 'Saint-Christophe-et-Nieves' OR PAYS = 'Sainte-Lucie' OR PAYS = 'Saint-Vincent-et-les Grenadines' OR PAYS = 'Salvador' OR PAYS = 'Suriname' OR PAYS = 'Trinite-et-Tobago' OR PAYS = 'Uruguay' OR PAYS = 'Venezuela'</p> <p>)</p> <p>UNION ALL</p> <p>(SELECT* from stock_Rm</p> <p>WHERE PAYS NOT IN ('Norvege', 'Suede', 'Danemark', 'Islande', 'Finlande', 'Royaume-Uni', 'Irlande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Allemagne', 'Pologne', 'Espagne', 'Portugal', 'Andorre','France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie', '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')</p> <p>);</p> <p>-----CREATION DES SYNONYMES-----</p> <p>Comme nous n'avons pas encore à utiliser la réplcation il nous faut des synonymes pour Produits, Catégories et Employés pour la transparence:</p> <p>CREATE or REPLACE SYNONYM PRODUITS</p> <p>FOR gdelambert.produits@CONNEXIONEURS;</p> <p>CREATE or REPLACE SYNONYM CATEGORIES</p> <p>FOR gdelambert.categories@CONNEXIONEURS;</p>
--	---

	CREATE or REPLACE SYNONYM EMPLOYES FOR achellaoui.EMPLOYES@CONNEXIONAMER;
Nettoyages éventuels	DROP DATABASE LINK CONNEXIONDB11;
Tests de vérification du bon fonctionnement	SELECT* from FOURNISSEURS; --29/29 Tuples SELECT* from EMPLOYES; --9/9 Tuples SELECT* from PRODUITS; --77/77 Tuples SELECT* from CATEGORIES; --8/8 Tuples SELECT* from CLIENTS; --91/91 Tuples SELECT* from COMMANDES; --830/830 Tuples SELECT* from DETAILS_COMMANDES; --2155/2155 Tuples SELECT* from STOCK; --231/231 Tuples

4.2 Europe du Sud

Le binôme responsable de l'Europe du sud est le 3202 composé de benoît Algourdin et Grégoire de Lambertye.

Création des liens entre les bases	create database link lienVersDB11 connect to gdelambert identified by XXXXXXXX USING 'DB11'; create database link lienVersDB12 connect to gdelambert identified by XXXXXXXX USING 'DB12'; create database link lienVersDB14 connect to gdelambert identified by XXXXXXXX USING 'DB14';
Création et peuplement des tables	<p>-----Création et peuplement de Clients_ES-----</p> CREATE TABLE Clients_ES AS(SELECT * FROM Ryori.Clients@lienVersDB11 WHERE Pays IN ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzégovine', 'Croatie', 'Grèce', 'Macédoine', 'Monténégro', 'Serbie', 'Slovénie', 'Bulgarie'));

	<p>-----Création et peuplement de details_Commandes_ES-----</p> <pre>CREATE TABLE details_Commandes_ES AS(SELECT * FROM Ryori.Details_Commandes@lienVersDB11 WHERE NO_COMMANDE IN (SELECT NO_COMMANDE FROM commandes_ES));</pre> <p>-----Création et peuplement de Produits-----</p> <pre>CREATE TABLE Produits AS(SELECT * FROM Ryori.Produits@lienVersDB11);</pre> <p>-----Création et peuplement de Categories-----</p> <pre>CREATE TABLE Categories AS(SELECT * FROM Ryori.Categories@lienVersDB11);</pre>
Contraintes d'intégrité	<p>-----Création des Clés primaires-----</p> <pre>ALTER TABLE Clients_ES ADD CONSTRAINT PK_CLIENT_ES PRIMARY KEY (CODE_CLIENT);</pre> <pre>ALTER TABLE Stock_ES ADD CONSTRAINT PK_STOCK_ES PRIMARY KEY (REF_PRODUIT, PAYS);</pre> <pre>ALTER TABLE commandes_ES ADD CONSTRAINT PK_COMMANDES_ES PRIMARY KEY (NO_COMMANDE);</pre> <pre>ALTER TABLE details_Commandes_ES ADD CONSTRAINT PK_DETAILS_COMMANDES_ES PRIMARY KEY (NO_COMMANDE, REF_PRODUIT);</pre> <pre>ALTER TABLE Produits ADD CONSTRAINT PK_PRODUITS_ES PRIMARY KEY (REF_PRODUIT);</pre> <pre>ALTER TABLE Categories ADD CONSTRAINT PK_CATEGORIES_ES PRIMARY KEY (CODE_CATEGORIE);</pre> <p>-----Création des Clés étrangères-----</p> <pre>ALTER TABLE Produits ADD CONSTRAINT FK_PRODUIT_CODE_CATEGORIE FOREIGN KEY (CODE_CATEGORIE) REFERENCES CATEGORIES(CODE_CATEGORIE)NOT DEFERRABLE;</pre> <pre>ALTER TABLE commandes_ES ADD CONSTRAINT fk_commandes_es_code_client FOREIGN KEY(CODE_CLIENT) REFERENCES Clients_ES(CODE_CLIENT)NOT DEFERRABLE;</pre> <pre>ALTER TABLE details_Commandes_ES ADD CONSTRAINT fk_detailsCommandes_es_noCommande FOREIGN KEY(NO_COMMANDE) REFERENCES commandes_ES(NO_COMMANDE) NOT DEFERRABLE;</pre>

```
ALTER TABLE Stock_ES ADD CONSTRAINT fk_Stockes_es_refproduit
FOREIGN KEY(REF_PRODUIT) REFERENCES Produits(REF_PRODUIT) NOT DEFERRABLE;
```

```
ALTER TABLE details_Commandes_ES ADD CONSTRAINT
fk_detailsCommandes_es_refProduits
FOREIGN KEY(REF_PRODUIT) REFERENCES PRODUITS(REF_PRODUIT)NOT
DEFERRABLE;
```

-----CREATION DES CLES ETRANGERES SUR DES TABLES DISTANTES (Trigger)-----

-----Trigger remplaçant la clé étrangère de COMMANDES sur le champ NO_EMPLOYE de la table EMPLOYES :

```
CREATE OR REPLACE TRIGGER Commandes_es_fk
BEFORE INSERT OR UPDATE ON COMMANDES_ES
FOR EACH ROW
DECLARE
countEmployes NUMBER;
BEGIN
SELECT count(*) INTO countEmployes FROM achellaoui.EMPLOYES@lienVersDB14
WHERE NO_EMPLOYE= :NEW.NO_EMPLOYE;
```

```
if(countEmployes = 0)
THEN
raise_application_error(-20002, 'NO_EMPLOYES invalide');
END IF;
END;
```

-----Trigger remplaçant la clé étrangère de PRODUITS sur le champ NO_FOURNISSEUR de la table FOURNISSEURS :

```
CREATE OR REPLACE TRIGGER Produits_fk
BEFORE INSERT OR UPDATE ON Produits
FOR EACH ROW
DECLARE
countFournisseur NUMBER;
BEGIN
SELECT count(*) INTO countFournisseur FROM sngov.FOURNISSEURS@lienVersDB12
WHERE NO_FOURNISSEUR= :NEW.NO_FOURNISSEUR;
```

```
if(countFournisseur = 0)
THEN
raise_application_error(-20002, 'NO_FOURNISSEUR invalide');
END IF;
END;
```

-----Contraintes de type CHECK-----

```
ALTER TABLE CLIENTS_ES ADD CONSTRAINT Pays_ClientES
CHECK(PAYS IN ('Espagne', 'Portugal','Andorre','France','Gibraltar','Italie', 'Saint-Marin',
'Vatican','Malte','Albanie','Bosnie-Herzegovine','Croatie','Grece', 'Macedoine', 'Montenegro',
'Serbie', 'Slovenie', 'Bulgarie'));
```

	<pre>ALTER TABLE STOCK_ES ADD CONSTRAINT Pays_StockES CHECK(PAYS IN ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie'));</pre>
Droits d'accès	<pre>grant select on CLIENTS_ES to achellaoui; grant select on STOCK_ES to achellaoui; grant select on COMMANDES_ES to achellaoui; grant select on DETAILS_COMMANDES_ES to achellaoui; grant select on PRODUITS to achellaoui; grant select on CATEGORIES to achellaoui; grant select on CLIENTS_ES to sngov; grant select on STOCK_ES to sngov; grant select on COMMANDES_ES to sngov; grant select on DETAILS_COMMANDES_ES to sngov; grant select on PRODUITS to sngov; grant select on CATEGORIES to sngov;</pre>
Définition de synonymes et de vues pour interrogation de la base comme si elle était en centralisé	<pre>----- View clients ----- CREATE OR REPLACE VIEW Clients AS (SELECT * FROM achellaoui.Clients_A@lienVersDB14 WHERE Pays IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica', 'Cuba', 'République dominicaine', 'Dominique', 'Équateur', 'Etats-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')) UNION ALL (SELECT * FROM sngov.Clients_EN@lienVersDB12 WHERE Pays IN ('Norvege', 'Suede', 'Danemark', 'Islande', 'Finlande', 'Royaume-Uni', 'Irlande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Allemagne', 'Pologne')) UNION ALL (SELECT * FROM Clients_ES WHERE Pays IN ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie')) UNION ALL (SELECT * FROM sngov.Clients_RM@lienVersDB12 WHERE Pays NOT IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Brésil', 'Canada', 'Chili', 'Colombie', 'Costa Rica', 'Cuba', 'République dominicaine', 'Dominique', 'Équateur', 'Etats-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', 'Norvege', 'Suede', 'Danemark', 'Islande', 'Finlande', 'Royaume-Uni', 'Irlande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Allemagne', 'Pologne', 'Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican'));</pre>

	<p>'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie')));</p> <p>----- View stock -----</p> <p>CREATE OR REPLACE VIEW Stock</p> <p>AS (SELECT *</p> <p>FROM achellaoui.Stock_A@lienVersDB14</p> <p>WHERE Pays IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica', 'Cuba', 'République dominicaine', 'Dominique', 'Équateur', 'Etats-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'))</p> <p>UNION ALL</p> <p>(SELECT *</p> <p>FROM sngov.Stock_EN@lienVersDB12</p> <p>WHERE Pays IN ('Norvege', 'Suede', 'Danemark', 'Islande', 'Finlande', 'Royaume-Uni', 'Irlande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Allemagne', 'Pologne'))</p> <p>UNION ALL</p> <p>(SELECT *</p> <p>FROM Stock_ES</p> <p>WHERE Pays IN ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie'))</p> <p>UNION ALL</p> <p>(SELECT *</p> <p>FROM sngov.Stock_RM@lienVersDB12</p> <p>WHERE Pays NOT IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil', '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', 'Norvege', 'Suede', 'Danemark', 'Islande', 'Finlande', 'Royaume-Uni', 'Irlande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Allemagne', 'Pologne', 'Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie')));</p> <p>----- View commandes -----</p> <p>CREATE OR REPLACE VIEW Commandes</p> <p>AS (SELECT *</p> <p>FROM achellaoui.Commandes_A@lienVersDB14)</p> <p>UNION ALL</p> <p>(SELECT *</p> <p>FROM sngov.Commandes_EN@lienVersDB12)</p> <p>UNION ALL</p> <p>(SELECT *</p> <p>FROM Commandes_ES)</p> <p>UNION ALL</p> <p>(SELECT *</p> <p>FROM sngov.Commandes_RM@lienVersDB12);</p>
--	--

	<pre> ----- View Détails_commande ----- CREATE OR REPLACE VIEW Détails_Commandes AS (SELECT * FROM achellaoui.Detail_Commandes_A@lienVersDB14) UNION ALL (SELECT * FROM sngov.Detail_Commandes_EN@lienVersDB12) UNION ALL (SELECT * FROM Détails_Commandes_ES) UNION ALL (SELECT * FROM sngov.Detail_Commandes_RM@lienVersDB12); -----CREATION DES SYNONYMES----- CREATE OR REPLACE SYNONYM FOURNISSEURS FOR sngov.FOURNISSEURS@lienVersDB12; CREATE OR REPLACE SYNONYM EMPLOYES FOR achellaoui.EMPLOYES@lienVersDB14; </pre>
Nettoyages éventuels	DROP DATABASE LINK lienVersDB11 – Suppression du lien vers la base de données centralisée
Tests de vérification du bon fonctionnement	<pre> select * from clients; --91/91 Tuples select * from stock; --231/231 Tuples select * from fournisseurs; --29/29 Tuples select * from employes; --9/9 Tuples select * from commandes; --830/830 Tuples select * from details_commandes; --2155/2155 Tuples select * from produits; --77/77 Tuples select * from categories; --8/8 Tuples </pre>

4.3 Amérique

Le Le binôme responsable de la Base d'Amérique est le B3203 composé de Rim BENZEKRI et Adam CHELLAOUI

Création des liens entre les bases	<pre> CREATE DATABASE LINK linkToRyori CONNECT TO achellaoui IDENTIFIED BY MDPORACLE USING 'DB11'; CREATE DATABASE LINK linkToEDN CONNECT TO achellaoui IDENTIFIED BY MDPORACLE USING 'DB12'; CREATE DATABASE LINK linkToEDS CONNECT TO achellaoui IDENTIFIED BY MDPORACLE USING 'DB13'; </pre>
Création et peuplement des tables	<pre> -----Création et peuplement de Clients_A----- CREATE TABLE Clients_A AS (SELECT * FROM ryori.Clients@linkToRyori </pre>

	<p>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', 'Jamaïque', 'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Pérou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela'));</p> <p>-----Création et peuplement de Stock_A-----</p> <p>CREATE TABLE Stock_A AS (SELECT * FROM ryori.Stock@linkToRyori 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', 'Jamaïque', 'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Pérou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela'));</p> <p>-----Création et peuplement de commandes_A-----</p> <p>CREATE TABLE Commandes_A AS (SELECT com.* FROM ryori.Commandes@linkToRyori com, Clients_A cl WHERE com.Code_Client = cl.Code_Client);</p> <p>-----Création et peuplement de details_Commandes_A-----</p> <p>CREATE TABLE Detail_Commandes_A AS (SELECT dcom.* FROM ryori.Details_Commandes@linkToRyori dcom, Commandes_A com WHERE dcom.NO_COMMANDE = com.NO_COMMANDE);</p> <p>-----Création et peuplement de Employes-----</p> <p>- CREATE TABLE Employes AS (SELECT * FROM ryori.Employes@linkToRyori);</p>
<p>Contraintes d'intégrité</p>	<p>-----Création des Clés primaires-----</p> <p>ALTER TABLE Clients_ES ADD CONSTRAINT PK_CLIENT_ES ALTER TABLE Clients_A ADD CONSTRAINT PK_Clients_A PRIMARY KEY (CODE_CLIENT); ALTER TABLE Commandes_A ADD CONSTRAINT PK_Commandes_A PRIMARY KEY (NO_COMMANDE); ALTER TABLE Detail_Commandes_A ADD CONSTRAINT PK_Detail_Commandes PRIMARY KEY (NO_COMMANDE, REF_PRODUIT); ALTER TABLE Employes ADD CONSTRAINT PK_Employes PRIMARY KEY (NO_EMPLOYE); ALTER TABLE Stock_A ADD CONSTRAINT PK_Stock_A PRIMARY KEY (REF_PRODUIT, PAYS);</p> <p>-----Création des Clés étrangères-----</p> <p>ALTER TABLE Commandes_A ADD CONSTRAINT FK_Commandes_Clients FOREIGN KEY (CODE_CLIENT) REFERENCES Clients_A (CODE_CLIENT); ALTER TABLE Commandes_A ADD CONSTRAINT FK_Commande_Employes</p>

```

FOREIGN KEY (NO_EMPLOYE) REFERENCES Employes (NO_EMPLOYE) ;
ALTER TABLE Details_Commandes_A ADD CONSTRAINT
FK_Details_Commandes_Commandes
FOREIGN KEY (NO_COMMANDE) REFERENCES Commandes_A (REF_PRODUIT);
ALTER TABLE Employes ADD CONSTRAINT FK_Employes_Employes
FOREIGN KEY (REND_COMPTE) REFERENCES Employes (REND_COMPTE);

-----CREATION DE TRIGGER POUR VERIFIER LA CONFORMITE DES DONNEES-----
create or replace Trigger Delete_Employes_Commandes
before delete ON Employes
for each row
declare
nbKeys number;
Begin
  select count(*)
  into nbKeys
  from Commandes
  where No_Employe = :NEW.No_Employe;
  If nbKeys > 0
  then
    RAISE_APPLICATION_ERROR(-20009, 'Cannot Delete tuple in this table');
  END IF;
END;

-----CREATION DES CLES ETRANGERES SUR DES TABLES DISTANTES (Trigger)-----
-----Trigger remplaçant la clé étrangère de COMMANDES
CREATE or REPLACE Trigger FK_Details_Commandes_Produits
BEFORE INSERT or UPDATE ON Detail_Commandes_A
FOR EACH ROW
DECLARE
nbElements number;
BEGIN
  SELECT count(*)
  INTO nbElements
  FROM gdelambert.Produits@linkToEDS
  WHERE ref_produit = :NEW.ref_produit;
  if nbElements = 0
  then
    RAISE_APPLICATION_ERROR(-20009, 'Cannot Insert tuple in this table');
  END if;
END;

-----Trigger remplaçant la clé étrangère de Stock_A
CREATE or REPLACE Trigger FK_Stock_A_Produits
BEFORE INSERT or UPDATE ON Stock_A
FOR EACH ROW
DECLARE
nbElements number;
BEGIN
  SELECT count(*)
  INTO nbElements

```


	<pre> FROM gdelambert.produits@linkToEDS WHERE ref_produit = :NEW.ref_produit; if nbElements = 0 then RAISE_APPLICATION_ERROR(-20009, 'Cannot Insert tuple in this table'); END if; END;</pre> <p>-----Contraintes de type CHECK-----</p> <pre> ALTER TABLE Clients_A ADD CONSTRAINT soc_notnull CHECK (SOCIETE is not null); ALTER TABLE Clients_A ADD CONSTRAINT adr_notnull CHECK (ADRESSE is not null); ALTER TABLE Clients_A ADD CONSTRAINT vil_notnull CHECK (VILLE is not null); ALTER TABLE Clients_A ADD CONSTRAINT cdpstl_notnull CHECK (CODE_POSTAL is not null); ALTER TABLE Clients_A ADD CONSTRAINT pays_notnull CHECK (PAYS is not null); ALTER TABLE Clients_A ADD CONSTRAINT tel_notnull CHECK (TELEPHONE is not null); ALTER TABLE Commandes_A ADD CONSTRAINT codcl_notnull CHECK (CODE_CLIENT is not null); ALTER TABLE Commandes_A ADD CONSTRAINT noemp_notnull CHECK (NO_EMPLOYE is not null); ALTER TABLE Commandes_A ADD CONSTRAINT datecom_notnull CHECK (DATE_COMMANDE is not null); ALTER TABLE Detail_Commandes_A ADD CONSTRAINT nocom_notnull CHECK (NO_COMMANDE is not null); ALTER TABLE Detail_Commandes_A ADD CONSTRAINT ref_notnull CHECK (REF_PRODUIT is not null); ALTER TABLE Detail_Commandes_A ADD CONSTRAINT prix_notnull CHECK (PRIX_UNITAIRE is not null); ALTER TABLE Detail_Commandes_A ADD CONSTRAINT qte_notnull CHECK (QUANTITE is not null); ALTER TABLE Detail_Commandes_A ADD CONSTRAINT rem_notnull CHECK (REMISE is not null); ALTER TABLE Employes ADD CONSTRAINT nomemp_notnull CHECK (NOM is not null); ALTER TABLE Employes ADD CONSTRAINT pren_notnull CHECK (PRENOM is not null);</pre>
Droits d'accès	<pre> GRANT SELECT ON Clients_A TO sngov ,gdelambert; GRANT SELECT ON Commandes_A TO sngov ,gdelambert; GRANT SELECT ON Detail_Commandes_A TO sngov ,gdelambert; GRANT SELECT ON Employes TO sngov ,gdelambert; GRANT SELECT ON Stock_A TO sngov ,gdelambert;</pre>
	<p>----- View clients -----</p> <pre> CREATE OR REPLACE VIEW Clients AS (SELECT * FROM Clients_A WHERE Pays IN ('Antigua-et-Barbuda','Argentine','Bahamas', 'Barbade', 'Belize', 'Bolovie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica', 'Cuba', 'Republique dominicaine',</pre>

<p>Définition de synonymes et de vues pour interrogation de la base comme si elle était en centralisé</p>	<p>'Dominique', 'Equateur', 'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaïque', 'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Etats-Unis', 'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela')</p> <p>UNION ALL</p> <p>SELECT * FROM sngov.Clients_EN@linkToEDN WHERE Pays IN ('Allemagne', 'Irlande', 'Norvege', 'Suede', 'Danemark', 'Islande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Pologne', 'Finlande', 'Royaume-Uni')</p> <p>UNION ALL</p> <p>SELECT * FROM sngov.Clients_RM@linkToEDN WHERE Pays NOT IN ('Allemagne', 'Irlande', 'Norvege', 'Suede', 'Danemark', 'Islande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Pologne', 'Finlande', 'Royaume-Uni', 'Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie', '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', 'Jamaïque', 'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela')</p> <p>UNION ALL</p> <p>SELECT * FROM gdelambert.Clients_ES@linkToEDS WHERE pays IN ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie'));</p> <p>----- View stock -----</p> <p>CREATE OR REPLACE VIEW Stock AS (SELECT * FROM stock_A WHERE Pays IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombi', 'Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique', 'Equateur', 'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaïque', 'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Etats-Unis', 'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela')</p> <p>UNION ALL</p> <p>SELECT * FROM sngov.stock_EN@linkToEDN WHERE Pays IN ('Allemagne', 'Irlande', 'Norvege', 'Suede', 'Danemark', 'Islande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Pologne', 'Finlande', 'Royaume-Uni')</p> <p>UNION ALL</p>
--	---

	<pre> SELECT * FROM sngov.stock_RM@linkToEDN WHERE Pays NOT IN ('Allemagne', 'Irlande', 'Norvege', 'Suede', 'Danemark', 'Islande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Pologne', 'Finlande', 'Royaume-Uni', 'Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie', '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', 'Jamaïque', '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 gdelambert.stock_ES@linkToEDS WHERE Pays IN ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie')); ----- View commandes ----- CREATE OR REPLACE VIEW Commandes AS (SELECT ca.* FROM Commandes_A ca, Clients c WHERE c.Code_Client=ca.Code_Client AND c.Pays IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique', 'Equateur', 'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaïque', 'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint- Christophe-et-Nieves', 'Sainte-Lucie', 'Etats-Unis', 'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela') UNION ALL SELECT cen.* FROM sngov.Commandes_EN@linkToEDN cen, Clients c WHERE c.Code_Client=cen.Code_Client AND c.Pays IN ('Allemagne', 'Irlande', 'Norvege', 'Suede', 'Danemark', 'Islande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Pologne', 'Finlande', 'Royaume-Uni') UNION ALL SELECT crm.* FROM sngov.Commandes_RM@linkToEDN crm, Clients c WHERE c.Code_Client=crm.Code_Client AND c.Pays NOT IN ('Allemagne', 'Irlande', 'Norvege', 'Suede', 'Danemark', 'Islande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Pologne', 'Finlande', 'Royaume-Uni', 'Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie', 'Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica', 'Cuba', </pre>
--	--

	<p>'Republique dominicaine', 'Dominique', 'Equateur', 'Etats-Unis', 'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaïque', 'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela')</p> <p>UNION ALL</p> <p>SELECT ces.* FROM gdelambert.Commandes_ES@linkToEDS ces, Clients c WHERE c.Code_Client=ces.Code_Client AND c.Pays IN ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie'));</p> <p>----- View Détails_commande -----</p> <p>CREATE OR REPLACE VIEW Details_Commandes AS (SELECT dca.* FROM detail_Commandes_A dca, Commandes co, Clients c WHERE c.Code_Client=co.Code_Client AND co.no_Commande=dca.no_commande AND c.Pays IN ('Antigua-et-Barbuda', 'Argentine', 'Bahamas', 'Barbade', 'Belize', 'Bolivie', 'Bresil', 'Canada', 'Chili', 'Colombie', 'Costa Rica', 'Cuba', 'Republique dominicaine', 'Dominique', 'Equateur', 'Grenade', 'Guatemala', 'Guyana', 'Haiti', 'Honduras', 'Jamaïque', 'Mexique', 'Nicaragua', 'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Etats-Unis', 'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela')</p> <p>UNION ALL</p> <p>SELECT dcen.* FROM sngov.Detail_Commandes_EN@linkToEDN dcen, Clients c, Commandes co WHERE c.Code_Client=co.Code_Client AND co.no_commande = dcen.no_commande AND c.Pays IN ('Allemagne', 'Irlande', 'Norvege', 'Suede', 'Danemark', 'Islande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Pologne', 'Finlande', 'Royaume-Uni')</p> <p>UNION ALL</p> <p>SELECT dcrm.* FROM sngov.Detail_Commandes_RM@linkToEDN dcrm, Clients c, Commandes co WHERE c.Code_Client=co.Code_Client AND co.no_commande = dcrm.no_commande AND c.Pays NOT IN ('Allemagne', 'Irlande', 'Norvege', 'Suede', 'Danemark', 'Islande', 'Belgique', 'Luxembourg', 'Pays-Bas', 'Pologne', 'Finlande', 'Royaume-Uni', 'Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie', 'Saint-Marin', 'Vatican', 'Malte', 'Albanie', 'Bosnie-Herzegovine', 'Croatie', 'Grece', 'Macedoine', 'Montenegro', 'Serbie', 'Slovenie', 'Bulgarie', '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', 'Jamaïque', 'Mexique', 'Nicaragua',</p>
--	---

	<p>'Panama', 'Paraguay', 'Perou', 'Saint-Christophe-et-Nieves', 'Sainte-Lucie', 'Saint-Vincent-et-les Grenadines', 'Salvador', 'Suriname', 'Trinite-et-Tobago', 'Uruguay', 'Venezuela')</p> <p>UNION ALL</p> <p>SELECT dces.* FROM gdelambert.details_Commandes_ES@linkToEDS dces,Clients c, Commandes co WHERE c.Code_Client=co.Code_Client AND co.no_commande = dces.no_commande AND c.Pays IN ('Espagne', 'Portugal','Andorre','France','Gibraltar','Italie', 'Saint-Marin','Vatican', 'Malte','Albanie','Bosnie-Herzegovine','Croatie','Grece','Macedoine','Montenegro','Serbie', 'Slovenie', 'Bulgarie'));</p> <p>-----CREATION DES SYNONYMES-----</p> <p>CREATE OR REPLACE SYNONYM produits FOR gdelambert.produits@LinkToEDS; CREATE OR REPLACE SYNONYM categories FOR gdelambert.categories@LinkToEDS; CREATE OR REPLACE SYNONYM fournisseurs FOR sngov.fournisseurs@LinkToEDN;</p>																
Nettoyages éventuels	DROP DATABASE LINK lienVersDB11 – Suppression du lien vers la base de données centralisée																
Tests de vérification du bon fonctionnement	<table> <tr> <td>SELECT * FROM Commandes;</td> <td>830/830 Tuples</td> </tr> <tr> <td>SELECT * FROM Clients;</td> <td>91/91 Tuples</td> </tr> <tr> <td>SELECT * FROM Stock;</td> <td>231/231 Tuples</td> </tr> <tr> <td>SELECT * FROM Details_Commandes;</td> <td>2155/2155 Tuples</td> </tr> <tr> <td>SELECT * FROM Employes;</td> <td>9/9 Tuples</td> </tr> <tr> <td>SELECT * FROM Produits;</td> <td>77/77 Tuples</td> </tr> <tr> <td>SELECT * FROM Fournisseurs;</td> <td>29/29 Tuples</td> </tr> <tr> <td>SELECT * FROM Categories;</td> <td>8/8 Tuples</td> </tr> </table> <p>Nous retrouvons bien le bon nombre de tuples de départ, il n'y a donc pas de pertes d'information, ni de duplication de tuple.</p>	SELECT * FROM Commandes;	830/830 Tuples	SELECT * FROM Clients;	91/91 Tuples	SELECT * FROM Stock;	231/231 Tuples	SELECT * FROM Details_Commandes;	2155/2155 Tuples	SELECT * FROM Employes;	9/9 Tuples	SELECT * FROM Produits;	77/77 Tuples	SELECT * FROM Fournisseurs;	29/29 Tuples	SELECT * FROM Categories;	8/8 Tuples
SELECT * FROM Commandes;	830/830 Tuples																
SELECT * FROM Clients;	91/91 Tuples																
SELECT * FROM Stock;	231/231 Tuples																
SELECT * FROM Details_Commandes;	2155/2155 Tuples																
SELECT * FROM Employes;	9/9 Tuples																
SELECT * FROM Produits;	77/77 Tuples																
SELECT * FROM Fournisseurs;	29/29 Tuples																
SELECT * FROM Categories;	8/8 Tuples																

5. Tests de requête distribuées et optimisations


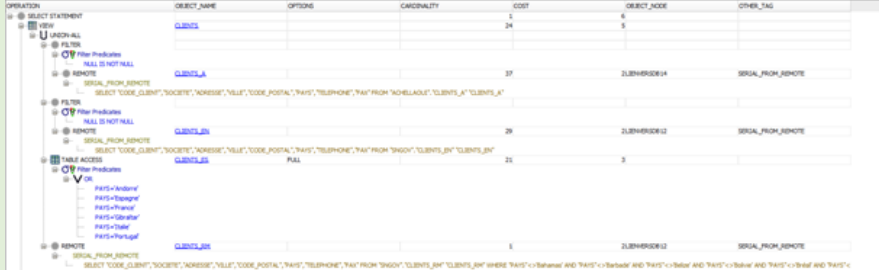
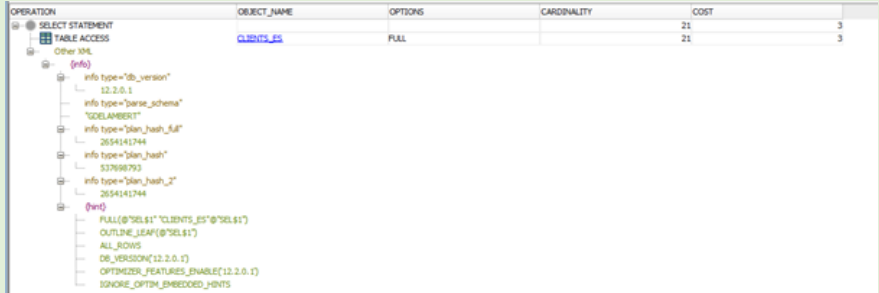
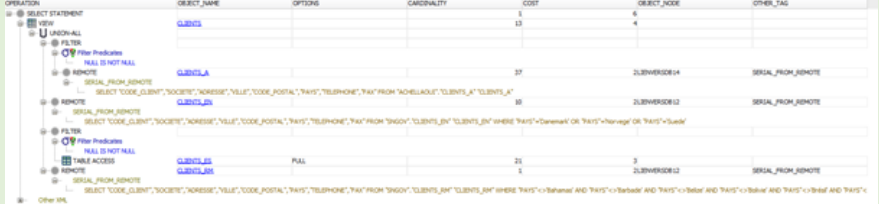
5.1 Europe du Nord et reste du monde

Le binôme responsable de l'Europe du Nord est le B3201: Annie Abhay et Sophanna Ngov.

SELECT* FROM CLIENTS;	
Plan d'exécution	
Analyse du plan d'exécution	Le UNION ALL est moins couteux que UNION, les fragments créés auparavant étants disjoints (contrainte de clé primaire sachant que tous les attributs sont sélectionnés pour créer la vue), on s'assure déjà de l'absence de doublons.
SELECT * FROM CLIENTS WHERE PAYS IN ('Royaume-Uni', 'Allemagne');	
Plan d'exécution	
Analyse du plan d'exécution	Avec la condition sur les pays (WHERE PAYS IN ()), l'optimiseur d'enlève les opérations d'union non nécessaires en cas de requête précisant le pays
select * from Clients_EN;	
Plan d'exécution	
Analyse du plan d'exécution	On trouve ici un simple accès à la table locale

5.2 Europe du Sud

Le binôme responsable de l'Europe du Sud est le B3202: Benoît Algourdin et Grégoire de Lambertye.

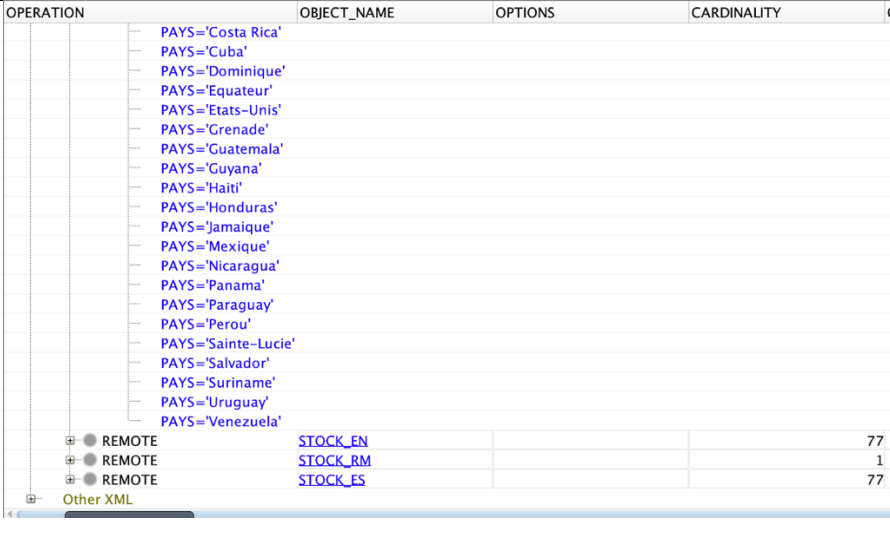
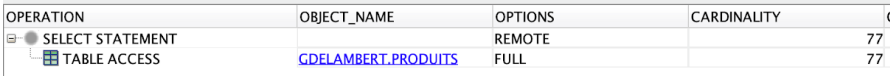
Select * from clients;	
Plan d'exécution	
Analyse du plan d'exécution	Grâce à UNION ALL, on ne cherche pas à éliminer les doublons. L'accès aux données se fait via les Base de Données distantes
select * from clients where pays in ('Espagne', 'Portugal', 'Andorre', 'France', 'Gibraltar', 'Italie');	
Plan d'exécution	
Analyse du plan d'exécution	Grâce aux conditions dans la création de la vue, on ne regarde pas dans les tables qui ne contiennent pas les pays recherchés.
select * from Clients_ES;	
Plan d'exécution	
Analyse du plan d'exécution	C'est un simple accès à une table hébergée sur le site.
select * from clients where pays in ('Norvege', 'Suede', 'Danemark');	
Plan d'exécution	
Analyse du plan d'exécution	On ne cherche bien qu'en Clients_EN, c'est une requête faite sur une table distante.

5.3 Amérique

Le binôme responsable de la Base d'Amérique est le B3203 composé de Rim BENZEKRI et Adam CHELLAOUI.

Select * from clients;		
Plan d'exécution	SQL 0,38 secondes	
	OPERATION	OBJECT_NAME OPTIONS
	SELECT STATEMENT	
	VIEW	CLIENTS
	UNION-ALL	
	TABLE ACCESS	CLIENTS_A FULL
	Filter Predicates	
	OR	
	PAYS='Argentine'	
	PAYS='Bahamas'	
PAYS='Barbade'		
PAYS='Belize'		
PAYS='Bolivie'		
PAYS='Bresil'		
PAYS='Canada'		
PAYS='Chili'		
PAYS='Colombi'		
PAYS='Costa Rica'		
PAYS='Cuba'		
PAYS='Dominique'		
PAYS='Equateur'		
PAYS='Etc...		
PAYS='Grenade'		
PAYS='Guatemala'		
PAYS='Guyana'		
PAYS='Haiti'		
PAYS='Honduras'		
PAYS='Jamaïque'		
PAYS='Mexique'		
PAYS='Nicaragua'		
PAYS='Panama'		
PAYS='Paraguay'		
PAYS='Perou'		
PAYS='Sainte-Lucie'		
PAYS='Salvador'		
PAYS='Suriname'		
PAYS='Uruguay'		
PAYS='Venezuela'		
REMOTE	CLIENTS_EN	
REMOTE	CLIENTS_RM	
REMOTE	CLIENTS_ES	
Analyse du plan d'exécution	Grâce à UNION ALL, on ne cherche pas à éliminer les doublons. La requête SQL répond en consultant tous les fragments. Il y a trois envois de requête à distance et un local.	
Select * From Clients WHERE PAYS IN ('Argentine', 'Brésil', 'Canada', 'Cuba', 'Panama', 'Mexique')		

Plan d'exécution	<table><tr><th>OPERATION</th><th>OBJECT_NAME</th><th>OPTIONS</th><th>CARDINALITY</th></tr><tr><td>SELECT STATEMENT</td><td></td><td></td><td>1</td></tr><tr><td>VIEW</td><td>CLIENTS</td><td></td><td>17</td></tr><tr><td>UNION-ALL</td><td></td><td></td><td></td></tr><tr><td>TABLE ACCESS</td><td>CLIENTS_A</td><td>FULL</td><td>14</td></tr><tr><td>Filter Predicates</td><td></td><td></td><td></td></tr><tr><td>PAYS='Argentine' OR PAYS='Canada' OR PAYS='Cuba' OR PAYS='Mexique' OR PAYS='Panama'</td><td></td><td></td><td></td></tr><tr><td>FILTER</td><td></td><td></td><td></td></tr><tr><td>Filter Predicates</td><td></td><td></td><td></td></tr><tr><td>NULL IS NOT NULL</td><td></td><td></td><td></td></tr><tr><td>REMOTE</td><td>CLIENTS_EN</td><td></td><td>29</td></tr><tr><td>SERIAL_FROM_REMOTE</td><td></td><td></td><td></td></tr><tr><td>SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "SNGOV"</td><td></td><td></td><td></td></tr><tr><td>REMOTE</td><td>CLIENTS_RM</td><td></td><td>1</td></tr><tr><td>SERIAL_FROM_REMOTE</td><td></td><td></td><td></td></tr><tr><td>SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "SNGOV"</td><td></td><td></td><td></td></tr><tr><td>FILTER</td><td></td><td></td><td></td></tr><tr><td>Filter Predicates</td><td></td><td></td><td></td></tr><tr><td>NULL IS NOT NULL</td><td></td><td></td><td></td></tr><tr><td>REMOTE</td><td>CLIENTS_ES</td><td></td><td>21</td></tr><tr><td>SERIAL_FROM_REMOTE</td><td></td><td></td><td></td></tr><tr><td>SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "GDELA"</td><td></td><td></td><td></td></tr><tr><td>Other XML</td><td></td><td></td><td></td></tr></table>	OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	SELECT STATEMENT			1	VIEW	CLIENTS		17	UNION-ALL				TABLE ACCESS	CLIENTS_A	FULL	14	Filter Predicates				PAYS='Argentine' OR PAYS='Canada' OR PAYS='Cuba' OR PAYS='Mexique' OR PAYS='Panama'				FILTER				Filter Predicates				NULL IS NOT NULL				REMOTE	CLIENTS_EN		29	SERIAL_FROM_REMOTE				SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "SNGOV"				REMOTE	CLIENTS_RM		1	SERIAL_FROM_REMOTE				SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "SNGOV"				FILTER				Filter Predicates				NULL IS NOT NULL				REMOTE	CLIENTS_ES		21	SERIAL_FROM_REMOTE				SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "GDELA"				Other XML															
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY																																																																																																						
SELECT STATEMENT			1																																																																																																						
VIEW	CLIENTS		17																																																																																																						
UNION-ALL																																																																																																									
TABLE ACCESS	CLIENTS_A	FULL	14																																																																																																						
Filter Predicates																																																																																																									
PAYS='Argentine' OR PAYS='Canada' OR PAYS='Cuba' OR PAYS='Mexique' OR PAYS='Panama'																																																																																																									
FILTER																																																																																																									
Filter Predicates																																																																																																									
NULL IS NOT NULL																																																																																																									
REMOTE	CLIENTS_EN		29																																																																																																						
SERIAL_FROM_REMOTE																																																																																																									
SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "SNGOV"																																																																																																									
REMOTE	CLIENTS_RM		1																																																																																																						
SERIAL_FROM_REMOTE																																																																																																									
SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "SNGOV"																																																																																																									
FILTER																																																																																																									
Filter Predicates																																																																																																									
NULL IS NOT NULL																																																																																																									
REMOTE	CLIENTS_ES		21																																																																																																						
SERIAL_FROM_REMOTE																																																																																																									
SELECT "CODE_CLIENT","SOCIETE","ADRESSE","VILLE","CODE_POSTAL","PAYS","TELEPHONE","FAX" FROM "GDELA"																																																																																																									
Other XML																																																																																																									
Analyse du plan d'exécution	<p>Les "Filter Predicates" NULL IS NOT NULL indiquent que l'optimiseur de requêtes Oracle constate qu'aucun des tuples recherchés ne seront dans les tables Clients_EN ou Clients_ES. Néanmoins, cela ne s'applique pas pour la table Clients_RM car cette vue est implémentée différemment, avec des conditions de non appartenance, pas des conditions d'appartenance. Le résultat est plus rapide grâce aux prédicats.</p>																																																																																																								
SELECT * FROM Clients_A																																																																																																									
Plan d'exécution	<table><tr><th>OPERATION</th><th>OBJECT_NAME</th><th>OPTIONS</th><th>CARDINALITY</th></tr><tr><td>SELECT STATEMENT</td><td></td><td></td><td></td></tr><tr><td>TABLE ACCESS</td><td>CLIENTS_A</td><td>FULL</td><td></td></tr><tr><td>Other XML</td><td></td><td></td><td></td></tr></table>	OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	SELECT STATEMENT				TABLE ACCESS	CLIENTS_A	FULL		Other XML																																																																																											
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY																																																																																																						
SELECT STATEMENT																																																																																																									
TABLE ACCESS	CLIENTS_A	FULL																																																																																																							
Other XML																																																																																																									
Analyse du plan d'exécution	<p>L'exécution de la requête est simple, on demande tous les tuples de notre table Clients_A qui est locale.</p>																																																																																																								
SELECT * FROM STOCK																																																																																																									
Plan d'exécution	<table><tr><th>OPERATION</th><th>OBJECT_NAME</th><th>OPTIONS</th><th>CARDINALITY</th></tr><tr><td>SELECT STATEMENT</td><td></td><td></td><td>232</td></tr><tr><td>VIEW</td><td>STOCK</td><td></td><td>232</td></tr><tr><td>UNION-ALL</td><td></td><td></td><td></td></tr><tr><td>TABLE ACCESS</td><td>STOCK_A</td><td>FULL</td><td>77</td></tr><tr><td>Filter Predicates</td><td></td><td></td><td></td></tr><tr><td>OR</td><td></td><td></td><td></td></tr><tr><td>PAYS='Argentine'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Bahamas'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Barbade'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Belize'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Bolivie'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Bresil'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Canada'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Chili'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Colombi'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Costa Rica'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Cuba'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Dominique'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Equateur'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Etats-Unis'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Grenade'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Guatemala'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Guyana'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Haiti'</td><td></td><td></td><td></td></tr><tr><td>PAYS='Honduras'</td><td></td><td></td><td></td></tr></table>	OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	SELECT STATEMENT			232	VIEW	STOCK		232	UNION-ALL				TABLE ACCESS	STOCK_A	FULL	77	Filter Predicates				OR				PAYS='Argentine'				PAYS='Bahamas'				PAYS='Barbade'				PAYS='Belize'				PAYS='Bolivie'				PAYS='Bresil'				PAYS='Canada'				PAYS='Chili'				PAYS='Colombi'				PAYS='Costa Rica'				PAYS='Cuba'				PAYS='Dominique'				PAYS='Equateur'				PAYS='Etats-Unis'				PAYS='Grenade'				PAYS='Guatemala'				PAYS='Guyana'				PAYS='Haiti'				PAYS='Honduras'			
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY																																																																																																						
SELECT STATEMENT			232																																																																																																						
VIEW	STOCK		232																																																																																																						
UNION-ALL																																																																																																									
TABLE ACCESS	STOCK_A	FULL	77																																																																																																						
Filter Predicates																																																																																																									
OR																																																																																																									
PAYS='Argentine'																																																																																																									
PAYS='Bahamas'																																																																																																									
PAYS='Barbade'																																																																																																									
PAYS='Belize'																																																																																																									
PAYS='Bolivie'																																																																																																									
PAYS='Bresil'																																																																																																									
PAYS='Canada'																																																																																																									
PAYS='Chili'																																																																																																									
PAYS='Colombi'																																																																																																									
PAYS='Costa Rica'																																																																																																									
PAYS='Cuba'																																																																																																									
PAYS='Dominique'																																																																																																									
PAYS='Equateur'																																																																																																									
PAYS='Etats-Unis'																																																																																																									
PAYS='Grenade'																																																																																																									
PAYS='Guatemala'																																																																																																									
PAYS='Guyana'																																																																																																									
PAYS='Haiti'																																																																																																									
PAYS='Honduras'																																																																																																									

	
Analyse du plan d'exécution	Un prédicat a été posé par Oracle automatiquement discriminant les pays de notre base de données locale. Ce prédicat permet de filtrer plus rapidement les requêtes qui concernent un pays de notre continent.
SELECT * FROM Produits	
Plan d'exécution	
Analyse du plan d'exécution	Nous obtenons 77 tuples, ce qui permet de vérifier qu'il n'y a pas de perte d'information. Le plan d'exécution de la requête est simple puisqu'il suffit d'accéder à toutes les informations de la Table Produits à distance à travers notre synonym.

6. Répliquions

6.1 Mise en œuvre des répliquions sur le site Europe du Nord

Le binôme responsable d'Europe du Nord est le binôme B3201: ABHAY Annie et NGOV Sophanna.

Objectifs	Un accès local des tables par la méthode de réplication serait plus efficace et moins coûteux que l'utilisation à distance des synonymes.
Des répliquions prévues	Réplication de la table EMPLOYES depuis l'Amérique Réplication de la table PRODUITS depuis l'Europe de sud Réplication de la table CATEGORIES depuis l'Europe de sud
	Il existe deux méthodes de répliquions : fast refresh et complete refresh

Analyse	<p>Le 'complete' fait une mise à jour pour toute la table à chaque modification (ne possède pas de Log), mais le 'fast' fait une mise à jour que pour des tuples qui ont subi une modification (mais il faudra créer un MATERIALIZED VIEW LOG).</p> <p>Le complete refresh est donc compatible avec les tables de petite taille (pas beaucoup de tuples) et le fast refresh est compatible avec les tables de grande taille (beaucoup de tuples):</p> <p>EMPLOYES: 9 tuples : méthode COMPLETE PRODUITS: 77 tuples : méthode FAST CATEGORIES: 8 tuples : méthode COMPLETE</p>
Mise en place des réplicas pour les besoins locaux	<p>Nous supposons qu'il est nécessaire de faire un rafraichissement quotidiennement à minuit</p> <p>-----Réplication de la table EMPLOYES-----</p> <pre>CREATE MATERIALIZED VIEW EMPLOYES REFRESH COMPLETE NEXT TRUNC(SYSDATE) +1 AS (SELECT * FROM achellaoui.EMPLOYES@CONNEXIONAMER)</pre> <p>-----Réplication de la table PRODUITS-----</p> <pre>CREATE MATERIALIZED VIEW PRODUITS REFRESH FAST NEXT TRUNC(SYSDATE) +1 AS (SELECT * FROM gdelambert.PRODUITS@CONNEXIONEURS)</pre> <p>-----Réplication de la table CATEGORIES-----</p> <pre>CREATE MATERIALIZED VIEW CATEGORIES REFRESH COMPLETE NEXT TRUNC(SYSDATE) +1 AS (SELECT * FROM gdelambert.CATEGORIES@CONNEXIONEURS)</pre>
Création du log de la vue matérialisée	<p>-----Création du LOG de la table FOURNISSEURS-----</p> <pre>CREATE MATERIALIZED VIEW LOG ON FOURNISSEURS;</pre>
Droit d'accès	<pre>GRANT SELECT ON MLOG\$_FOURNISSEURS TO achellaoui; GRANT SELECT ON MLOG\$_FOURNISSEURS TO gdelambert;</pre>
	<p>On retire les triggers sur les tables et on ajoute des contraintes de clés étrangères</p> <p>-----Création de clés étrangères-----</p> <pre>ALTER TABLE COMMANDES_EN ADD CONSTRAINT fk_CommEMPL FOREIGN KEY (NO_EMPLOYE) REFERENCES EMPLOYES(NO_EMPLOYE);</pre>

Les modifications effectuées	<pre> ALTER TABLE COMMANDES_RM ADD CONSTRAINT fk_CommRMEMPL FOREIGN KEY (NO_EMPLOYE) REFERENCES EMPLOYES(NO_EMPLOYE); ALTER TABLE STOCK_EN ADD CONSTRAINT fk_StockPROD FOREIGN KEY (REF_PRODUIT) REFERENCES PRODUITS(REF_PRODUIT); ALTER TABLE STOCK_RM ADD CONSTRAINT fk_StockRMPROD FOREIGN KEY (REF_PRODUIT) REFERENCES PRODUITS(REF_PRODUIT); ALTER TABLE DETAIL_COMMANDES_EN ADD CONSTRAINT fk_DCommProd FOREIGN KEY (REF_PRODUIT) REFERENCES PRODUITS(REF_PRODUIT); ALTER TABLE DETAIL_COMMANDES_RM ADD CONSTRAINT fk_DCommRMProd FOREIGN KEY (REF_PRODUIT) REFERENCES PRODUITS(REF_PRODUIT); ----- On supprimera les synonymes associés aux tables EMPLOYES, PRODUITS, CATEGORIES </pre>
Tests de vérification	<pre> SELECT* from FOURNISSEURS; --29/29 Tuples SELECT* from EMPLOYES; --9/9 Tuples SELECT* from PRODUITS; --77/77 Tuples SELECT* from CATEGORIES; --8/8 Tuples SELECT* from CLIENTS; --91/91 Tuples SELECT* from COMMANDES; --830/830 Tuples SELECT* from DETAILS_COMMANDES; --2155/2155 Tuples SELECT* from STOCK; --231/231 Tuples </pre>
Demandes des autres sites	<p>-----Demande du site Europe du Sud-----</p> <p>Demande de gdelambert d'interdire la suppression ou la modification du numéro d'un fournisseur qui serait enregistré pour un produit</p> <p>Cela correspond à l'implémentation d'une foreign key en PLSQL de la table PRODUITS sur le NO_FOURNISSEURS de la table FOURNISSEUR</p> <p>-----Code fournis à exécuter:</p> <pre> CREATE OR REPLACE TRIGGER Europe_Sud_NO_FOURNISSEUR_FK BEFORE DELETE OR UPDATE ON FOURNISSEURS FOR EACH ROW DECLARE countFournisseur NUMBER; BEGIN SELECT count(*) INTO countFournisseur FROM gdelambert.Produits@CONNEXIONEURS WHERE NO_FOURNISSEUR= :NEW.NO_FOURNISSEUR; if(countFournisseur != 0) THEN </pre>

	<pre> raise_application_error(-20002, 'Le fournisseur sert a fabriquer un produit, il ne peut pas etre supprimé ou modifié'); END IF; END; -----Test de bon fonctionnement Après l'exécution du code, on ne peut plus modifier ou supprimer un fournisseur référéncé par un produit. -----Demande du site Amérique----- -----Code fournis à exécuter: create or replace Trigger Delete_Employes_Commandes before delete ON Employes for each row declare nbKeys number; Begin select count(*) into nbKeys from Commandes where No_Employe = :NEW.No_Employe; If nbKeys > 0 then RAISE_APPLICATION_ERROR(-20009, 'Cannot Delete tuple in this table'); END IF; END; </pre>
--	---

6.2 Mise en œuvre des réplifications sur le site Europe du Sud

Le binôme responsable de l'Europe du sud est le B3202: Benoît Algourdin et Grégoire de Lambertye

Objectifs	L'objectif de la réplification est d'héberger en local des tables normalement distantes pour limiter les flux de données et gagner en performances
Liste des réplifications prévues	Pour le site d'Europe du sud nous avons décidé de ne répliquer que les table EMPLOYES et FOURNISSEURS
Analyse	<p>EMPLOYES contient peu de tuples, elle sera répliquée en mode REFRESH COMPLET une fois par jours.</p> <p>FOURNISSEURS contient plus de tuples, elle sera répliquée en mode REFRESH FAST toutes les minutes. (Il faut bien étaler ses connaissances)</p>

Mise en place des réplicas pour les besoins locaux	<p>-----Réplica de la table Employes-----</p> <pre>CREATE MATERIALIZED VIEW ViewEMPLOYES REFRESH COMPLETE NEXT sysdate + (1) AS select * from achellaoui.EMPLOYES@lienVersDB14;</pre> <pre>CREATE OR REPLACE SYNONYM Employes FOR ViewEMPLOYES;</pre> <p>----- Réplica de la table Fournisseurs-----</p> <pre>CREATE MATERIALIZED VIEW ViewFournisseur REFRESH FAST NEXT sysdate +(1/24/60) AS select * from sngov.FOURNISSEURS@lienVersDB12;</pre> <pre>CREATE OR REPLACE SYNONYM FOURNISSEURS FOR ViewFournisseur;</pre>
Création du log de la vue matérialisée	<pre>CREATE MATERIALIZED VIEW LOG ON PRODUITS;</pre>
Droit d'accès	<pre>GRANT SELECT ON MLOG\$_PRODUITS TO achellaoui ; GRANT SELECT ON MLOG\$_PRODUITS TO sngov;</pre>
Les modifications effectuées	<p>On pourrait éventuellement appliquer les foreign key qui se réfèrent à la table employes distante sur son réplica local.</p> <p>On pourrait éventuellement appliquer les foreign key qui se réfèrent à la table fournisseurs distante sur son réplica local.</p>
Tests de bon fonctionnement	<pre>Commande : Select * from EMPLOYES ; Résultat : 9/9 Tuples Dans le plan d'exécution, on voit que c'est le réplica local qui est utilisé pour la requête.</pre> <p>-----</p> <pre>Commande : Select * from FOURNISSEURS; Résultat : 29/29 Tuples Dans le plan d'exécution, on voit que c'est le réplica local qui est utilisé pour la requête.</pre>
	<p>----- Demande du site Europe du Nord-----</p> <p>Demande de sngov pour interdire la suppression de tuples dans produits lorsque le stock local en référence et que des commandes sont passées (clés étrangères)</p> <p>----- Code à exécuter-----</p> <pre>CREATE OR REPLACE TRIGGER EuropeNord_REF_PRODUITS_FK_DCOMM_EN BEFORE DELETE OR UPDATE ON PRODUITS FOR EACH ROW</pre>

<p>Demande des Autres Sites</p>	<pre> DECLARE countProd NUMBER; BEGIN SELECT count(*) INTO countProd FROM sngov.DETAIL_COMMANDES_EN@lienEURN --//CHANGER LE NOM DU LIEN (VERS EUROPE DU NORD) WHERE REF_PRODUITS= :NEW.REF_PRODUITS; if(countProd != 0) THEN raise_application_error(-20002, 'Les Produits deja commandés ne peuvent etre supprimé ou modifié'); END IF; END; CREATE OR REPLACE TRIGGER EuropeNord_REF_PRODUITS_FK_DCOMM_RM BEFORE DELETE OR UPDATE ON PRODUITS FOR EACH ROW DECLARE countProd NUMBER; BEGIN SELECT count(*) INTO countProd FROM sngov.DETAIL_COMMANDES_RM@lienEURN --//CHANGER LE NOM DU LIEN (VERS EUROPE DU NORD) WHERE REF_PRODUITS= :NEW.REF_PRODUITS; if(countProd != 0) THEN raise_application_error(-20002, 'Les Produits deja commandés ne peuvent etre supprimé ou modifié'); END IF; END; CREATE OR REPLACE TRIGGER EuropeNord_REF_PRODUITS_FK_STOCK_RM BEFORE DELETE OR UPDATE ON PRODUITS FOR EACH ROW DECLARE countProd NUMBER; BEGIN SELECT count(*) INTO countProd FROM sngov.STOCK_RM@lienEURN --//CHANGER LE NOM DU LIEN (VERS EUROPE DU NORD) WHERE REF_PRODUITS= :NEW.REF_PRODUITS; if(countProd != 0) THEN raise_application_error(-20002, 'Les Produits en stock ne peuvent etre supprimé ou modifié'); END IF; END; CREATE OR REPLACE TRIGGER EuropeNord_REF_PRODUITS_FK_STOCK_EN BEFORE DELETE OR UPDATE ON PRODUITS FOR EACH ROW DECLARE countProd NUMBER; </pre>
--	--

	<pre> BEGIN SELECT count(*) INTO countProd FROM sngov.STOCK_EN@lienEURN --//CHANGER LE NOM DU LIEN (VERS EUROPE DU NORD) WHERE REF_PRODUITS= :NEW.REF_PRODUITS; if(countProd != 0) THEN raise_application_error(-20002, 'Les Produits deja commandés ne peuvent etre supprimé ou modifié'); END IF; END; </pre>
--	--

6.3 Mise en œuvre des répliquions sur le site Amérique

Le binôme responsable du site Amérique est le B3203 composé de Rim BENZEKRI et Adam CHELLAOUI.

Objectifs	
Liste des répliquions prévues	L'objectif du processus de répliquion est d'éviter un appel à un site étranger le plus souvent possible. Cela permet de faciliter l'accès aux données, en en faisant des données locales.
Analyse	<p>La table Produits ainsi que la table Categories sont sur le site EDS. La table Fournisseurs est sur le site EDN. Nous allons donc répliquer les 3 tables précédentes, et ne nous répliquons évidemment pas la table Employes puisqu'elle est en local.</p> <p>Concernant les tables Fournisseurs et Categories, vu qu'elles ne connaissent pas un grand nombre de modifications et qu'elles seront pas très fréquentes nous les répliquons en mode read only - refresh complete. La fréquence de mise à jour est quotidienne.</p> <p>La Table Produits est assez souvent amenée à être modifiée et contient un grand nombre, elle sera donc répliquée en mode read only - refresh fast. La fréquence de mise à jour est de chaque heure.</p>
Mise en place des réplicas pour les besoins locaux (opérations réalisées localement)	<pre> -----Répliquat de la table Categories----- CREATE MATERIALIZED VIEW DMV_Categories REFRESH COMPLETE NEXT sysdate + (1) AS SELECT * FROM gdelambert.categories@LinkToEDS; CREATE OR REPLACE EDITIONABLE SYNONYM "CATEGORIES" FOR DMV_Categories; ----- Répliquat de la table Fournisseurs----- CREATE MATERIALIZED VIEW DMV_Fournisseurs REFRESH COMPLETE NEXT sysdate + (1) AS SELECT * FROM sngov.fournisseurs@LinkToEDN; </pre>

	<pre>CREATE OR REPLACE EDITIONABLE SYNONYM "FOURNISSEURS" FOR DM_Fournisseurs; ----- Réplicat de la table Produits----- CREATE MATERIALIZED VIEW MV_Produits REFRESH FAST NEXT sysdate + (0/1) AS SELECT * FROM gdelambert.produits@LinkToEDS;</pre>
Commentaires sur le réplikat Produits	<p>Nous mettons en place une réplication en mode refresh fast (et read only) puisque la table Produits contient un grand nombre de tuples. Cela nécessite néanmoins la mise en place d'un log pour pouvoir enregistrer les modifications qui ont eu lieu sur la table Produits basée sur le site EDS, et de mettre ainsi à jour nos données toutes les heures.</p>
Les modifications effectuées	<p>Il faut penser à modifier les triggers FK_Stock_A_Produits et FK_Details_Commandes_Produits, afin qu'ils utilisent les vues matérialisées.</p>
Tests de bon fonctionnement	<p>Commande : Select * from Categories ; Résultat : 8/8 Tuples Dans le plan d'exécution, on voit que c'est le réplica local qui est utilisé pour la requête.</p> <p>-----</p> <p>Commande : Select * from FOURNISSEURS; Résultat : 29/29 Tuples Dans le plan d'exécution, on voit que c'est le réplica local qui est utilisé pour la requête.</p> <p>-----</p> <p>Commande : Select * from Produits; Résultat : 77/77 Tuples Dans le plan d'exécution, on voit que c'est le réplica local qui est utilisé pour la requête.</p>
Demande des Autres Sites	<p>----- Demande du site Europe du Sud-----</p> <p>Demande de gdelambert d'interdire la suppression ou la modification du numéro d'un employé qui serait enregistré pour une commande.</p> <p>Cela correspond à l'implémentation d'une foreign key en PLSQL de la table COMMANDES sur le NO_EMPLOYES de la table EMPLOYES</p> <p>-----Code fournis à exécuter:</p> <pre>CREATE OR REPLACE TRIGGER Europe_Sud_NO_EMPLOYE_FK</pre>

	<pre> BEFORE DELETE OR UPDATE ON EMPLOYES FOR EACH ROW DECLARE countEmploye NUMBER BEGIN SELECT count(*) INTO countEmploye FROM gdelambert.COMMANDES_ES@lienVersDB13 --//CHANGER LE NOM DU LIEN (VERS EUROPE DU SUD) WHERE NO_EMPLOYE= :NEW.NO_EMPLOYE; if(countEmploye != 0) THEN raise_application_error(-20002, 'NO_EMPLOYE ne peut pas etre supprimé ou modifié'); END IF; END; -----Test de bon fonctionnement Après l'exécution du code, on ne peut plus modifier ou supprimer un employé référéncé par une commande. -----Demande du site Europe du Nord----- CREATE OR REPLACE TRIGGER Am_EMPLOYES_CommEN_FK BEFORE DELETE OR UPDATE ON EMPLOYES FOR EACH ROW DECLARE countEmployes NUMBER; BEGIN SELECT count(*) INTO countEmployes FROM snogv.Commandes_EN@LienVersDB12 WHERE NO_EMPLOYES= :NEW.NO_EMPLOYES; if(countEmployes != 0) THEN raise_application_error(-20002, 'Des Employes sont referencés dans des commandes, il ne peuvent être supprimés'); END IF; END; CREATE OR REPLACE TRIGGER Am_EMPLOYES_CommRM_FK BEFORE DELETE OR UPDATE ON EMPLOYES FOR EACH ROW DECLARE countEmployes NUMBER; BEGIN SELECT count(*) INTO countEmployes FROM snogv.Commandes_RM@LienVersDB12 WHERE NO_EMPLOYES= :NEW.NO_EMPLOYES; if(countEmployes != 0) THEN </pre>
--	---

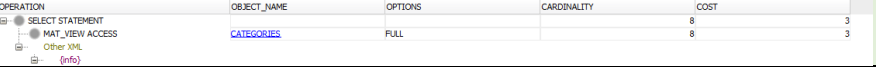
	<pre>raise_application_error(-20002, 'Des Employes sont referencés dans des commandes, il ne peuvent être supprimés'); END IF; END;</pre>
--	--

6.4 Bilan global des répliquations mises en œuvre sur les différents sites. (Tableau de synthèse, etc.)

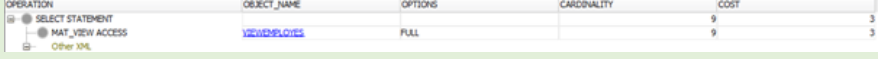
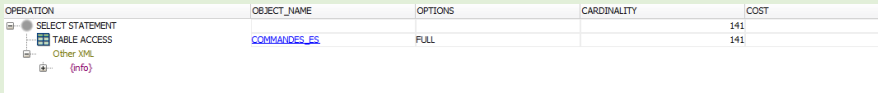
	Europe du Nord	Europe du Sud	Amériques
Employes	Replicat (complete)	Réplica (complete)	héberge
Fournisseurs	héberge	Réplica (fast)	Réplica
Produits	Réplicat (fast)	héberge	Réplica
Categories	Réplicat (complete)	héberge	Réplica

7. Requêtes distribuées : tests et optimisations

7.1 Site Europe du Nord

SELECT * FROM CATEGORIES;						
Plan d'exécution						
	La requête passe bien par la vue matérialisée.					
	Analyse du plan d'exécution					

7.2 Site Europe du Sud

Select * from EMPLOYES;					
Plan d'exécution					
Analyse du plan d'exécution	La requête s'effectue via le réplica local , on obtient le même résultat sur les autres tables distantes.				
SELECT * FROM Commandes_ES					
Plan d'exécution					

Analyse du plan d'exécution	La réplication n'a rien changé sur les tables locales.
SELECT * from FOURNISSEURS Natural JOIN PRODUITS WHERE PRIX_UNITAIRE < 40;	
Plan d'exécution	
Analyse du plan d'exécution	On observe que tout se passe sur les tables et réplicas locaux. L'optimiseur de requêtes débute par vérifier que les NO_FOURNISSEURS coïncident bien car c'est ce qu'il y a de plus discriminant.

7.3 Site Amérique

Nous effectuons une ultime vérification portant sur la réplication que nous avons effectuée. On cherche à récupérer les données de la table Produits :

Select * from Fournisseurs;	
Plan d'exécution	
Analyse du plan d'exécution	<p>Il y a bien 29 tuples, la réplication s'est donc bien déroulée. La requête s'effectue via le réplica local mais le plan d'exécution montre un cout inchangé (3).</p> <p>L'accès à la table FOURNISSEURS s'effectue en effet à travers la vue matérialisée que nous avons mis en place. Pour poursuivre les vérifications on effectue de même avec les tables CATEGORIES et PRODUITS. On obtient effectivement le bon nombre de tuples pour chacune des tables. Enfin il serait intéressant d'observer si les mises à jour s'effectuent également correctement, en demandant par exemple au binôme responsable du site d'EDS d'ajouter quelques tuples produits dans la table Produits, et de vérifier qu'au maximum une heure après, ils apparaissent également dans notre vue</p>
Select * From Produits NATURAL JOIN Fournisseurs Where PRIX_UNITAIRE>50;	

Plan d'exécution					
	OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
	SELECT STATEMENT			75	5
	MERGE JOIN			75	5
	MAT_VIEW ACCESS	DM_FOURNISSEURS	BY INDEX ROWID	29	2
	INDEX	SYS_C_SNAPS_1001PK_FO...	FULL SCAN	29	1
	SORT		JOIN	75	3
	Access Predicates	PRODUITS.NO_FOURNISSEUR=FOURNISSEURS.NO_FOURNISSEUR			
	Filter Predicates	PRODUITS.NO_FOURNISSEUR=FOURNISSEURS.NO_FOURNISSEUR			
	REMOTE	PRODUITS		75	2
SERIAL_FROM_REMOTE					
SELECT 'REF_PRODUIT','NOM_PRODUIT','NO_FOURNISSEUR','CODE_CATEGORIE','QUANTITE','PRIX_UNITAIRE' FROM 'GDELAMBERT'. 'PRODUIT'					
Analyse du plan d'exécution	Un prédicat sur la clé étrangère/primaire NO_Fournisseur a été mis en place par l'optimiseur de requêtes Oracle en premier lieu car il discrimine le plus de tuples. Ensuite nous remarquons un accès aux tuples par INDEX dans la vue matérialisée sur NO_Fournisseur pour un accès plus rapide.				
SELECT* FROM Categories ;					
Plan d'exécution					
	OPERATION	OBJECT_NAME		CARDINALITY	COST
	SELECT STATEMENT			8	
	MAT_VIEW ACCESS (FULL)	DMV_CATEGORIES		8	
Analyse du plan d'exécution	L'accès se fait bien par la vue matérialisée, et on retrouve nos 8 tuples. Néanmoins le cout est toujours de 3.				