

Compte rendu TP2

Victor Lezaud

15 janvier 2018

Sommaire

I	Rappel rapide du contexte et des objectifs du TP	5
1	Présentation	5
2	Les applications	5
3	Les pays gérés par les différents sites	5
II	Présentation du groupe de travail et des rôles de chacun	6
4	Présentation du groupe de travail et des rôles de chacun	6
4.1	Membre du groupe	6
4.2	Répartition des sites	6
5	Identification des tâches à réaliser et répartition	6
III	Fragmentation	7
6	Détermination des fragments	7
6.1	Fragmentation horizontale	7
6.1.1	Fragmentation horizontale de STOCK	7
6.1.2	Fragmentation horizontale de CLIENTS	7
6.2	Fragmentation dérivée	8
6.2.1	Fragmentation dérivée de COMMANDES	8
6.2.2	Fragmentation dérivée de DETAILS_COMMANDES	8
6.3	Fragmentation verticale	8
6.4	Fragmentation mixte	8
7	Bilan de la Fragmentation	8

8	Placement des fragments sur les sites (sans réplication)	9
8.1	Analyse	9
8.2	Bilan	9
8.2.1	Par application	9
8.2.2	Par site	10
9	Mise en oeuvre de la base sans réplication	10
9.1	Site Europe du Nord	10
9.1.1	Binôme responsable	10
9.1.2	Création des liens entre les bases	10
9.1.3	Création des tables	10
9.1.4	Peuplement des tables	12
9.1.5	Contraintes d'intégrité	12
9.1.6	Droit d'accès	14
9.1.7	Transparence	15
9.1.8	Nettoyages éventuels	16
9.1.9	Tests de vérification du bon fonctionnement	16
9.2	Site Europe du Sud	16
9.2.1	Binôme responsable	16
9.2.2	Création des liens entre les bases	17
9.2.3	Création des tables	17
9.2.4	Peuplement des tables	17
9.2.5	Contraintes d'intégrité	17
9.2.6	Droit d'accès	17
9.2.7	Transparence	17
9.2.8	Nettoyages éventuels	17
9.2.9	Tests de vérification du bon fonctionnement	17
9.3	Site Amérique	17
9.3.1	Binôme responsable	17
9.3.2	Création des liens entre les bases	17
9.3.3	Création des tables	17
9.3.4	Peuplement des tables	17
9.3.5	Contraintes d'intégrité	17
9.3.6	Droit d'accès	17
9.3.7	Transparence	17
9.3.8	Nettoyages éventuels	17
9.3.9	Tests de vérification du bon fonctionnement	17
IV	Test de requête distribués et optimisations	18
10	Europe du Nord	18
10.1	Requete 1	18
10.1.1	Code de la requête	18
10.1.2	Résultat d'exécution	18
10.1.3	Analyse du plan d'exécution	18
10.1.4	Autre écriture possible	18

11 Europe du Sud	19
11.1 Requete 1	19
11.1.1 Résultat d'exécution	19
11.1.2 Analyse du plan d'exécution	19
11.1.3 Autre écriture possible	19
12 Amérique	19
12.1 Requete 1	19
12.1.1 Résultat d'exécution	19
12.1.2 Analyse du plan d'exécution	19
12.1.3 Autre écriture possible	19
V Répliques	20
13 Sur le site Europe du Nord	20
13.1 Rappel binôme responsable	20
13.2 Objectifs	20
13.3 Liste des répliques prévues	20
13.4 Analyse	20
13.4.1 EMPLOYES	20
13.5 Mise en oeuvre des répliques	20
13.5.1 Réplicat du fragment EMPLOYES	20
13.6 Demandes d'autres sites	21
13.6.1 Fragment FOURNISSEURS	21
14 Sur le site Europe du Sud	21
14.1 Rappel binôme responsable	21
14.2 Objectifs	21
14.3 Liste des répliques prévues	21
14.4 Analyse	21
14.5 Mise en oeuvre des répliques	21
14.5.1 Réplicat du fragment xxx	21
14.6 Demandes d'autres sites	22
14.6.1 Fragment xxx	22
15 Sur le site Amérique	22
15.1 Rappel binôme responsable	22
15.2 Objectifs	22
15.3 Liste des répliques prévues	22
15.4 Analyse	22
15.5 Mise en oeuvre des répliques	22
15.5.1 Réplicat du fragment xxx	22
15.6 Demandes d'autres sites	23
15.6.1 Fragment xxx	23
16 Bilan global des répliques	23
VI Requêtes distribuées : tests et optimisations	24

17 Site Europe du Nord	24
18 Site Europe du Sud	24
19 Site Amérique	24

Première partie

Rappel rapide du contexte et des objectifs du TP

1 Présentation

Dans ce TP, nous considérons une entreprise qui est implanté sur plusieurs continents nommée "Ryori". Cette entreprise possède une base de données centralisée. Dans le cadre de ce TP, nous allons proposer une solution distribuée sur les différents sites.

2 Les applications

L'entreprise utilise plusieurs applications :

- L'application MakeIt : gère la fabrication des produits 'maison'. Accès fréquent à la table FOURNISSEURS et au stock allemand. Déployée en Europe du Nord.
- L'application DesignIt : bureau d'étude. Accès fréquent et unique modificateur des tables PRODUITS et CATEGORIES. Déployée en Europe du Sud.
- l'application SellIt : gère la vente. Accès fréquent aux clients, stock et commandes locaux. Déployée sur tous les sites.
- L'application RH : gère les Ressources Humaines. Accès fréquent à la table EMPLOYES. Déployée en Amérique.

3 Les pays gérés par les différents sites

- 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
- Antigua-et-Barbuda, Argentine, Bahamas, Barbade, Belize, Bolivie, Brésil, Canada, Chili, Colombie, Costa Rica, Cuba, République dominicaine, Dominique, Equateur, 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

Deuxième partie

Présentation du groupe de travail et des rôles de chacun

4 Présentation du groupe de travail et des rôles de chacun

4.1 Membre du groupe

Le groupe est composé de :

- CHALLAL Mohamed
- FLORANT Clément
- GALY ADAM Marah
- LERAL Mathieu
- LEZAUD Victor
- RASOLDIER Aïna

Le chef de projet est Aïna Rasoldier.

4.2 Répartition des sites

Binôme	Site
Marah GALY ADAM, Aina RASOLDIER	Amérique
Clément FLORANT, Victor LEZAUD	Europe du Nord
Mohamed CHALLAL, Mathieu LE RAL	Europe du Sud

5 Identification des tâches à réaliser et réparti- tion

Troisième partie

Fragmentation

6 Détermination des fragments

6.1 Fragmentation horizontale

6.1.1 Fragmentation horizontale de STOCK

Prédicats discriminants L'application SellIt a besoin pour fonctionner du stock local. La définition de local est donné par la liste de pays donnée en introduction. Les accès fréquents sont définis par les prédicats suivants :

- $STOCK.PAYS \in PaysEN$
- $STOCK.PAYS \in PaysES$
- $STOCK.PAYS \in PaysA$

Prédicats complexe On peut former les prédicats complexes suivants :

- $STOCK.PAYS \in PaysEN \text{ and } STOCK.PAYS \in PaysEN \text{ and } STOCK.PAYS \in PaysEN$
- $STOCK.PAYS \notin PaysEN \text{ and } STOCK.PAYS \in PaysEN \text{ and } STOCK.PAYS \in PaysEN$
- $STOCK.PAYS \in PaysEN \text{ and } STOCK.PAYS \notin PaysEN \text{ and } STOCK.PAYS \in PaysEN$
- $STOCK.PAYS \in PaysEN \text{ and } STOCK.PAYS \in PaysEN \text{ and } STOCK.PAYS \notin PaysEN$
- $STOCK.PAYS \notin PaysEN \text{ and } STOCK.PAYS \notin PaysEN \text{ and } STOCK.PAYS \in PaysEN$
- $STOCK.PAYS \notin PaysEN \text{ and } STOCK.PAYS \in PaysEN \text{ and } STOCK.PAYS \notin PaysEN$
- $STOCK.PAYS \in PaysEN \text{ and } STOCK.PAYS \notin PaysEN \text{ and } STOCK.PAYS \notin PaysEN$
- $STOCK.PAYS \notin PaysEN \text{ and } STOCK.PAYS \notin PaysEN \text{ and } STOCK.PAYS \notin PaysEN$

Aucun pays n'appartient à plus d'une liste, ainsi l'appartenance à l'une d'elle implique l'absence des autres. On peut donc supprimer les prédicats impossibles et en simplifier d'autres. On obtient ainsi :

- $STOCK.PAYS \in PaysEN$
- $STOCK.PAYS \in PaysES$
- $STOCK.PAYS \in PaysA$
- $STOCK.PAYS \notin PaysEN \text{ and } STOCK.PAYS \notin PaysEN \text{ and } STOCK.PAYS \notin PaysEN$

On obtient donc quatre fragments pour la table STOCK

6.1.2 Fragmentation horizontale de CLIENTS

La fragmentation de CLIENTS est réalisé comme STOCK.

6.2 Fragmentation dérivée

6.2.1 Fragmentation dérivée de COMMANDES

Principe Il paraît évident que si un client est lié à une application ses commandes seront gérés par cette application. On réalise donc une fragmentation dérivée de COMMANDES à partir de la fragmentation de CLIENTS

Les fragments On obtient donc autant de fragment que pour CLIENTS et formés de la manière suivante pour chaque fragment i :

$$COMMANDES_i = \pi_{COMMANDES}(CLIENTS_i \bowtie COMMANDES)$$

6.2.2 Fragmentation dérivée de DETAILS_COMMANDES

La fragmentation dérivée de DETAILS_COMMANDES par rapport à COMMANDES est exactement comme la fragmentation de COMMANDES par rapport à CLIENTS.

6.3 Fragmentation verticale

Il n'y a aucune fragmentation verticale car rien dans les usages ne nous permet d'identifier des attributs particulièrement utilisés par une application.

6.4 Fragmentation mixte

En absence de fragmentation verticale il ne peut y avoir de fragmentation mixte.

7 Bilan de la Fragmentation

- FOURNISSEURS : 1 fragment
- PRODUITS : 1 fragment
- CATEGORIES : 1 fragment
- EMPLOYES : 1 fragment
- FOURNISSEURS : 1 fragment
- CLIENTS : 4 fragments
 - CLIENTS.PaysEN : $\sigma_{PAYS \in PaysEN}(CLIENTS)$
 - CLIENTS.PaysES : $\sigma_{PAYS \in PaysES}(CLIENTS)$
 - CLIENTS.PaysA : $\sigma_{PAYS \in PaysA}(CLIENTS)$
 - CLIENTS.PaysRM : $\sigma_{PAYS \notin PaysEN \cup PaysES \cup PaysA}(CLIENTS)$
- STOCK : 4 fragments
 - STOCK.PaysEN : $\sigma_{PAYS \in PaysEN}(STOCK)$
 - STOCK.PaysES : $\sigma_{PAYS \in PaysES}(STOCK)$
 - STOCK.PaysA : $\sigma_{PAYS \in PaysA}(STOCK)$
 - STOCK.PaysRM : $\sigma_{PAYS \notin PaysEN \cup PaysES \cup PaysA}(STOCK)$
- COMMANDES : 4 fragments
 - COMMANDES.PaysEN : $\pi_{COMMANDES}(CLIENTS.PaysEN \bowtie COMMANDES)$
 - COMMANDES.PaysES : $\pi_{COMMANDES}(CLIENTS.PaysES \bowtie COMMANDES)$

- $COMMANDES.PaysA : \pi_{COMMANDES}(CLIENTS.PaysA \bowtie COMMANDES)$
- $COMMANDES.PaysRM : \pi_{COMMANDES}(CLIENTS.PaysRM \bowtie COMMANDES)$
- $DETAILS_COMMANDES$: 4 fragments
 - $DETAILS_COMMANDES.PaysEN : \pi_{DETAILS_COMMANDES}(COMMANDES.PaysEN \bowtie DETAILS_COMMANDES)$
 - $DETAILS_COMMANDES.PaysES : \pi_{DETAILS_COMMANDES}(COMMANDES.PaysES \bowtie DETAILS_COMMANDES)$
 - $DETAILS_COMMANDES.PaysA : \pi_{DETAILS_COMMANDES}(COMMANDES.PaysA \bowtie DETAILS_COMMANDES)$
 - $DETAILS_COMMANDES.PaysRM : \pi_{DETAILS_COMMANDES}(COMMANDES.PaysRM \bowtie DETAILS_COMMANDES)$

8 Placement des fragments sur les sites (sans réplication)

8.1 Analyse

Les tables non fragmentées sont chacune particulièrement utilisées par une des applications. La table FOURNISSEURS est liée à MakeIt, EMPLOYES à RH et PRODUITS et CATEGORIES à DesignIt. L'application SellIt a des usages différents selon la localité. On définit donc quatre sous-application SellIt.EN, SellIt.ES, SellIt.A et SellIt.RM. Logiquement les fragments de CLIENTS et STOCK sont liés aux sous-applications de la même localité. Par fragmentation dérivée les fragments de COMMANDES (et donc DETAILS_COMMANDES) sont liés à la même sous-application que le fragment de CLIENTS auquel il est associé.

8.2 Bilan

8.2.1 Par application

- MakeIt : *FOURNISSEURS*
- DesignIt : *PRODUITS, CATEGORIES*
- RH : *EMPLOYES*
- SellIt :
 - SellIt.EN :
 - $CLIENTS.PaysEN$
 - $STOCK.PaysEN$
 - $COMMANDES.PaysEN$
 - $DETAILS_COMMANDES.PaysEN$
 - SellIt.ES :
 - $CLIENTS.PaysES$
 - $STOCK.PaysES$
 - $COMMANDES.PaysES$
 - $DETAILS_COMMANDES.PaysES$
 - SellIt.A :
 - $CLIENTS.PaysA$

- STOCK.PaysA
- COMMANDES.PaysA
- DETAILS_COMMANDES.PaysA
- SellIt.RM
- CLIENTS.PaysRM
- STOCK.PaysRM
- COMMANDES.PaysRM
- DETAILS_COMMANDES.PaysRM

8.2.2 Par site

- Europe du Nord : MakeIt, SellIt.EN et SellIt.RM
- Europe du Sud : DesignIt et SellIt.ES
- Amérique : RH et SellIt.A

9 Mise en oeuvre de la base sans réplication

9.1 Site Europe du Nord

9.1.1 Binôme responsable

Le binôme responsable du site Europe du Nord est composé de FLORANT Clément et LEZAUD Victor

9.1.2 Création des liens entre les bases

La préparation en amont a déjà été faite par le Data Base Administrator. Ainsi chacune des bases des bases de données possède un nom global unique et connaît celui des autres et chaque utilisateur a un compte pour accéder à chacune des tables avec le même mot de passe. On peut créer des liens vers les autres base de données de cette manière :

```
CREATE DATABASE LINK Centrale CONNECT TO CURRENT USER USING 'DB11';
CREATE DATABASE LINK EuropeSud CONNECT TO CURRENT USER USING 'DB13';
CREATE DATABASE LINK America CONNECT TO CURRENT USER USING 'DB14';
```

9.1.3 Création des tables

Préparation : dans un but de simplification des modifications des pays gérés par le site et des écritures ci-dessous nous avons réalisé la table et la vue suivante :

```
-- creation table
create table PaysEuropeNord (
  pays varchar2(15 BYTE) Primary KEY);
-- insert value
INSERT INTO PaysEuropeNord VALUES ('Norvege');
INSERT INTO PaysEuropeNord VALUES ('Suede');
INSERT INTO PaysEuropeNord VALUES ('Danemark');
INSERT INTO PaysEuropeNord VALUES ('Islande');
INSERT INTO PaysEuropeNord VALUES ('Finlande');
```

```

INSERT INTO PaysEuropeNord VALUES ('Royaume-Uni');
INSERT INTO PaysEuropeNord VALUES ('Irlande');
INSERT INTO PaysEuropeNord VALUES ('Belgique');
INSERT INTO PaysEuropeNord VALUES ('Luxembourg');
INSERT INTO PaysEuropeNord VALUES ('Allemagne');
INSERT INTO PaysEuropeNord VALUES ('Pologne');
INSERT INTO PaysEuropeNord VALUES ('Pays-Bas');
INSERT INTO PaysEuropeNord VALUES ('Autriche');
INSERT INTO PaysEuropeNord VALUES ('Suisse');

```

```

CREATE OR REPLACE VIEW PaysGere AS
Select pays as pays
from PaysEuropeNord
union all
select nom_pays as pays
from arasoldier.pays_a@america
union all
select nom AS pays
from mleral.pays_es@europeSud;

```

Table FOURNISSEURS

```

create table Fournisseurs as
Select * from ryori.fournisseurs@centrale;

```

Table STOCK_EN

```

create table STOCK_EN as
Select * from ryori.STOCK@centrale
where pays in (Select * from PaysEuropeNord);

```

Table STOCK_OTH

```

create table STOCK_EN as
Select * from ryori.STOCK@centrale
where pays not in (Select * from PAYSGERE);

```

Table CLIENTS_EN

```

create table Clients_EN as
Select * from ryori.clients@centrale
where pays in (Select * from PaysEuropeNord);

```

Table CLIENTS_OTH

```

create table Clients_Oth as
Select * from ryori.clients@centrale
where pays not in (Select * from PAYSGERE);

```

Table COMMANDES_EN

```
create table Commandes_EN as
Select * from ryori.Commandes@centrale
where RyoriCommandes.code_client
in (Select Code_Client from Clients_EN);
```

Table COMMANDES_OTH

```
create table Commandes_OTH as
Select * from ryori.Commandes@centrale
where RyoriCommandes.code_client
in (Select Code_Client from Clients_OTH);
```

Table DETAILS_COMMANDES_EN

```
create table Details_Commandes_EN as
Select * from ryori.Details_Commandes@centrale
where RyoriDetails_Commandes.NO_Commande
in (Select NO_Commande from Commandes_EN);
```

DETAILS_COMMANDES_OTH

```
create table Details_Commandes_OTH as
Select * from ryori.Details_Commandes@centrale
where RyoriDetails_Commandes.NO_Commande
in (Select NO_Commande from Commandes_OTH);
```

9.1.4 Peuplement des tables

Le peuplement a été réalisés en même temps que leur création.

9.1.5 Contraintes d'intégrité

Table FOURNISSEURS

```
ALTER TABLE Fournisseurs ADD CONSTRAINT pkFour
PRIMARY KEY (NO_FOURNISSEUR);
```

Table STOCK_EN

```
ALTER TABLE STOCK_EN ADD CONSTRAINT pkSTOCKEn
PRIMARY KEY (REF_PRODUIT, PAYS);
```

```
ALTER TABLE STOCK_EN ADD CONSTRAINT checkStockEN
CHECK (Pays in (
'Allemagne','Autriche','Belgique','Danemark',
'Finlande','Irlande','Islande','Luxembourg','Norvege',
'Pays-Bas','Pologne','Royaume-Uni','Suede','Suisse'));
```

Table STOCK_OTH

```
ALTER TABLE STOCK_OTH ADD CONSTRAINT pkSTOCKoth
PRIMARY KEY (REF_PRODUIT, PAYS);
```

```
ALTER TABLE STOCK_OTH ADD CONSTRAINT checkStockoth
CHECK (Pays not in (
'Albanie', 'Allemagne', 'Andorre', 'Antigua-et-Barbuda',
'Argentine', 'Autriche', 'Bahamas', 'Barbade', 'Belgique', 'Belize', 'Bolivie',
'Bosnie-Herzégovine', 'Bresil', 'Bulgarie', 'Canada', 'Chili', 'Colombie',
'Costa Rica', 'Croatie', 'Cuba', 'Danemark', 'Dominique', 'Equateur', 'Espagne',
'Etats-Unis', 'Finlande', 'France', 'Gibraltar', 'Grenade', 'Grèce', 'Guatemala',
'Guyana', 'Haïti', 'Honduras', 'Irlande', 'Islande', 'Italie', 'Jamaïque',
'Luxembourg', 'Macédoine', 'Malte', 'Mexique', 'Monténégro', 'Nicaragua',
'Norvege', 'Panama', 'Paraguay', 'Pays-Bas', 'Pologne', 'Portugal', 'Pérou',
'Royaume-Uni', 'République dominicaine', 'Saint-Christophe-et-Nieves',
'Saint-Marin', 'Saint-Vincent-et-les Grenadines', 'Sainte-Lucie', 'Salvador',
'Serbie', 'Slovénie', 'Suede', 'Suisse', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Vatican', 'Venezuela');
```

Table CLIENTS_EN

```
ALTER TABLE Clients_EN ADD CONSTRAINT pkClientsEn
PRIMARY KEY (CODE_CLIENT);
```

```
ALTER TABLE Clients_EN ADD CONSTRAINT
checkClientsEN CHECK (Pays in (
'Allemagne', 'Autriche', 'Belgique', 'Danemark',
'Finlande', 'Irlande', 'Islande', 'Luxembourg', 'Norvege',
'Pays-Bas', 'Pologne', 'Royaume-Uni', 'Suede', 'Suisse');
```

Table CLIENTS_OTH

```
ALTER TABLE Clients_Oth ADD CONSTRAINT
pkClientsOth PRIMARY KEY (CODE_CLIENT);
```

```
ALTER TABLE Clients_Oth ADD CONSTRAINT checkClientsOth
check (pays not in (
'Albanie', 'Allemagne', 'Andorre', 'Antigua-et-Barbuda',
'Argentine', 'Autriche', 'Bahamas', 'Barbade', 'Belgique', 'Belize', 'Bolivie',
'Bosnie-Herzégovine', 'Bresil', 'Bulgarie', 'Canada', 'Chili', 'Colombie',
'Costa Rica', 'Croatie', 'Cuba', 'Danemark', 'Dominique', 'Equateur', 'Espagne',
'Etats-Unis', 'Finlande', 'France', 'Gibraltar', 'Grenade', 'Grèce', 'Guatemala',
'Guyana', 'Haïti', 'Honduras', 'Irlande', 'Islande', 'Italie', 'Jamaïque',
'Luxembourg', 'Macédoine', 'Malte', 'Mexique', 'Monténégro', 'Nicaragua',
'Norvege', 'Panama', 'Paraguay', 'Pays-Bas', 'Pologne', 'Portugal', 'Pérou',
'Royaume-Uni', 'République dominicaine', 'Saint-Christophe-et-Nieves',
'Saint-Marin', 'Saint-Vincent-et-les Grenadines', 'Sainte-Lucie', 'Salvador',
'Serbie', 'Slovénie', 'Suede', 'Suisse', 'Suriname', 'Trinite-et-Tobago',
'Uruguay', 'Vatican', 'Venezuela');
```

Table COMMANDES_EN

```
-- primary key
ALTER TABLE Commandes_OTH ADD CONSTRAINT pkCommandesOTH
PRIMARY KEY (NO_Commande);
-- foreign key
ALTER TABLE Commandes_OTH ADD CONSTRAINT fkCommandesOTHCommandeClient
FOREIGN KEY (Code_Client) REFERENCES Clients_OTH(Code_Client);
```

Table COMMANDES_OTH

```
-- primary key
ALTER TABLE Details_Commandes_EN ADD CONSTRAINT pkDetails_CommandesEn
PRIMARY KEY (NO_Commande, Ref_Produit);
-- foreign key
ALTER TABLE Details_Commandes_EN ADD CONSTRAINT fkDetailsCommandes
FOREIGN KEY (NO_Commande) REFERENCES Commandes_EN(NO_Commande);
```

Table DETAILS_COMMANDES_EN

```
-- primary key
ALTER TABLE Details_Commandes_EN ADD CONSTRAINT pkDetails_CommandesEn
PRIMARY KEY (NO_Commande, Ref_Produit);
-- foreign key
ALTER TABLE Details_Commandes_EN ADD CONSTRAINT fkDetailsCommandes FOREIGN KEY (NO_Commande) REFERENCES Commandes_EN(NO_Commande);
```

DETAILS_COMMANDES_OTH

```
-- primary key
ALTER TABLE Details_Commandes_OTH ADD CONSTRAINT pkDetails_CommandesOTH
PRIMARY KEY (NO_Commande, Ref_Produit);
-- foreign key
ALTER TABLE Details_Commandes_OTH ADD CONSTRAINT fkDetailsCommandesOTH
FOREIGN KEY (NO_Commande) REFERENCES Commandes_OTH(NO_Commande);
```

9.1.6 Droit d'accès

Tous les sites doivent pouvoir lire les données comme si la base était resté centralisée. Les utilisateurs des sites distants doivent avoir les droits en lecture sur ces tables

```
GRANT SELECT on FOURNISSEURS to arasoldier;
GRANT SELECT on STOCK_EN to arasoldier;
GRANT SELECT on Details_Commandes_EN to arasoldier;
GRANT SELECT on Commandes_EN to arasoldier;
GRANT SELECT on Clients_EN to arasoldier;
GRANT SELECT on PaysEuropeNord to arasoldier;
GRANT SELECT on PaysGere to arasoldier;
GRANT SELECT on Clients_OTH to arasoldier;
GRANT SELECT on Commandes_OTH to arasoldier;
GRANT SELECT on Details_Commandes_OTH to arasoldier;
```

```

GRANT SELECT on FOURNISSEURS to mleral;
GRANT SELECT on STOCK_EN to mleral;
GRANT SELECT on Details_Commandes_EN to mleral;
GRANT SELECT on Commandes_EN to mleral;
GRANT SELECT on Clients_EN to mleral;
GRANT SELECT on PaysEuropeNord to mleral;
GRANT SELECT on PaysGere to mleral;
GRANT SELECT on Clients_OTH to mleral;
GRANT SELECT on Commandes_OTH to mleral;
GRANT SELECT on Details_Commandes_OTH to mleral;

```

9.1.7 Transparence

But : La distribution de la table doit être parfaitement transparente. Toutes les requêtes effectuées sur la base centralisée doivent avoir la réponse sur la requête distribué

Synonymes : Pour les tables entièrement stockés sur un site distant, on utilise un synonyme pour rendre transparent l'accès à la base distante. On le réalise de la manière suivante :

```

-- Employe
CREATE SYNONYM Employes FOR arasoldier.Employes@AMERICA;
DROP SYNONYM Employes;
-- Produits
CREATE SYNONYM Produits FOR mleral.Produits@EUROPESUD;
DROP SYNONYM Produits;
-- Categories
CREATE SYNONYM Categories FOR mleral.Categories@EUROPESUD;
DROP SYNONYM Categories;

```

Vues : Pour les tables fragmentés, il faut utiliser une vue pour reformer la table

```

-- CLIENTS
CREATE VIEW clients AS
SELECT * FROM clients_EN
UNION ALL
SELECT * FROM mleral.clients_ES@EUROPESUD
UNION ALL
SELECT * FROM arasoldier.clients_A@AMERICA
UNION ALL
SELECT * FROM clients_OTH
-- Commandes
CREATE VIEW Commandes AS
SELECT * FROM Commandes_EN
UNION ALL
SELECT * FROM mleral.Commandes_ES@EUROPESUD
UNION ALL

```

```

SELECT * FROM arasoldier.Commandes_A@AMERICA
UNION ALL
SELECT * FROM Commandes_OTH;
-- Details_Commandes
CREATE VIEW Details_Commandes AS
SELECT * FROM Details_Commandes_EN
UNION ALL
SELECT * FROM mleral.Details_Commandes_ES@EUROPESUD
UNION ALL
SELECT * FROM arasoldier.Details_Commandes_A@AMERICA
UNION ALL
SELECT * FROM Details_Commandes_OTH;
-- STOCK
CREATE VIEW STOCK AS
SELECT * FROM STOCK_EN
WHERE Pays IN (SELECT * from PaysEuropeNord)
UNION ALL
SELECT * FROM mleral.STOCK_ES@EUROPESUD
WHERE Pays IN (SELECT * from mleral.pays_es@europeSud)
UNION ALL
SELECT * FROM arasoldier.STOCK_A@AMERICA
WHERE Pays IN (SELECT * from arasoldier.pays_a@america);

```

9.1.8 Nettoyages éventuels

Une fois toutes ces opérations réalisés on peut supprimer :

— le lien vers la base de données centralisée

9.1.9 Tests de vérification du bon fonctionnement

9.2 Site Europe du Sud

9.2.1 Binôme responsable

Le binôme responsable du site Europe du Sud est composé de Mohamed CHALAL et Mathieu LE RAL.

9.2.2 Création des liens entre les bases

9.2.3 Création des tables

9.2.4 Peuplement des tables

9.2.5 Contraintes d'intégrité

9.2.6 Droit d'accès

9.2.7 Transparence

9.2.8 Nettoyages éventuels

9.2.9 Tests de vérification du bon fonctionnement

9.3 Site Amérique

9.3.1 Binôme responsable

Le binôme responsable du site Europe du Sud est composé de Marah GALY ADAM et Aina RASOLDIER.

9.3.2 Création des liens entre les bases

9.3.3 Création des tables

9.3.4 Peuplement des tables

9.3.5 Contraintes d'intégrité

9.3.6 Droit d'accès

9.3.7 Transparence

9.3.8 Nettoyages éventuels

9.3.9 Tests de vérification du bon fonctionnement

Quatrième partie

Test de requête distribués et optimisations

10 Europe du Nord

10.1 Requete 1

10.1.1 Code de la requête

```
SELECT *  
FROM clients  
WHERE pays = 'France';
```

10.1.2 Résultat d'exécution

10.1.3 Analyse du plan d'exécution

Description du plan d'exécution : L'exécution de cette requête entraine une opération de sélection sur tous les fragments de clients sur leurs machines respectives puis une union des résultats de toutes ces sélections

Problème d'optimisation : Cet exemple montre une limite de l'optimiseur ORACLE. En effet ne connaissant pas les critères de la fragmentation il réalise la sélection sur tous les fragments alors que nous savons par définition des fragments Clients_A, Clients_EN et Clients_RM qu'il ne possède aucun tuple vérifiant la requête ci-dessus.

10.1.4 Autre écriture possible

Pour permettre à l'optimiseur ORACLE de régler ce problème et d'optimiser correctement la requête il faut modifier notre définition de la vue CLIENTS. On redéfinie de la vue de la manière suivante

```
-- CLIENTS  
CREATE or replace VIEW clients AS  
SELECT * FROM clients_EN  
WHERE Pays in ('Allemagne','Autriche','Belgique','Danemark',  
'Finlande','Irlande','Islande','Luxembourg','Norvege',  
'Pays-Bas','Pologne','Royaume-Uni','Suede','Suisse')  
UNION ALL  
SELECT * FROM mleral.clients_ES@EUROPESUD  
WHERE Pays in ('Espagne','France','Portugal','Italie','Andorre',  
'Gibraltar','Saint-Marin','Vatican','Malte','Albanie',  
'Bosnie-Herzégovine','Croatie','Grèce','Macédoine','Monténégro',  
'Serbie','Slovénie','Bulgarie')  
UNION ALL  
SELECT * FROM arasoldier.clients_A@AMERICA  
WHERE Pays in ('Antigua-et-Barbuda','Argentine','Bahamas','Barbade',
```

```

'Belize','Bolivie','Bresil','Canada','Chili','Colombie','Costa Rica',
'Cuba','Dominique','Equateur','Etats-Unis','Grenade','Guatemala',
'Guyana','Haiti','Honduras','Jamaique','Mexique','Nicaragua','Panama',
'Paraguay','Pérou','République dominicaine','Saint-Christophe-et-Nieves',
'Saint-Vincent-et-les Grenadines','Sainte-Lucie','Salvador','Suriname',
'Trinite-et-Tobago','Uruguay','Venezuela')
UNION ALL
SELECT * FROM clients_OTH
WHERE Pays not in ('Albanie','Allemagne','Andorre','Antigua-et-Barbuda',
'Argentine','Autriche','Bahamas','Barbade','Belgique','Belize','Bolivie',
'Bosnie-Herzégovine','Bresil','Bulgarie','Canada','Chili','Colombie',
'Costa Rica','Croatie','Cuba','Danemark','Dominique','Equateur','Espagne',
'Etats-Unis','Finlande','France','Gibraltar','Grenade','Grèce','Guatemala',
'Guyana','Haiti','Honduras','Irlande','Islande','Italie','Jamaique',
'Norvege','Panama','Paraguay','Pays-Bas','Pologne','Portugal','Pérou',
'Royaume-Uni','République dominicaine','Saint-Christophe-et-Nieves',
'Saint-Marin','Saint-Vincent-et-les Grenadines','Sainte-Lucie','Salvador',
'Serbie','Slovénie','Suede','Suisse','Suriname','Trinite-et-Tobago',
'Uruguay','Vatican','Venezuela');

```

'Luxembou

De cette façon l'optimiseur va détecter l'antinomie entre la sélection et la définition des fragments et donc on évite trois sélections et une union inutile. La même optimisation peut-être réalisée sur la table STOCK

11 Europe du Sud

11.1 Requete 1

11.1.1 Résultat d'exécution

11.1.2 Analyse du plan d'exécution

11.1.3 Autre écriture possible

12 Amérique

12.1 Requete 1

12.1.1 Résultat d'exécution

12.1.2 Analyse du plan d'exécution

12.1.3 Autre écriture possible

Cinquième partie

Répliques

13 Sur le site Europe du Nord

13.1 Rappel binôme responsable

Le binôme responsable du site Europe du Nord est FLORANT Clément et LEZAUD Victor

13.2 Objectifs

L'application SellIt, qui est déployée sur tous les sites, ne modifie pas les tables PRODUITS et CATEGORIES mais y a souvent accès. De plus il serait pratique d'avoir un accès rapide aux données du personnel du site. Nous allons donc mettre en place des réplicats mono-maître de ces tables.

13.3 Liste des répliques prévues

- Table PRODUITS
- Table CATEGORIES
- Table EMPLOYES

13.4 Analyse

13.4.1 EMPLOYES

La table EMPLOYES comporte peu de tuples et on ne connaît pas sa fréquence de mise à jour. On réalise donc une réplique COMPLETE REFRESH.

13.5 Mise en oeuvre des répliques

13.5.1 Réplicat du fragment EMPLOYES

Opérations réalisées localement :

```
CREATE MATERIALIZED VIEW EMPLOYES
REFRESH COMPLETE
NEXT (SYSDATE+1/1440)
AS(
SELECT *
FROM arasoldier.Employes@AMERICA);
```

Message émis au site maître :

Cher collègues,
Nous vous informons que nous allons créer une vue-matérialisée read-only,
complete refresh de votre table Employes.
Cordialement,
Lezaud Victor
Ryori site Europe du Nord

Réponse du site maître

Test de vérification de bon fonctionnement de la réplication

Evolutions éventuelles des contraintes d'intégrité La vue étant réalisée en read-only, nous avons une réplication mono-maître. La table ne peut donc être modifiée que la par le maître comme précédemment il n'y a pas donc d'évolution des contraintes d'intégrité.

Evolutions éventuelles des vues et des synonymes

13.6 Demandes d'autres sites

13.6.1 Fragment FOURNISSEURS

Demande du site Europe du Sud

Description de la demande et suivi Le site d'Europe du Sud souhaite réaliser une réplication mono-maître de la table FOURNISSEURS en mode FAST REFRESH

Analyse de la demande Pour permettre cette réplication il nous faut créer un log sur la table FOURNISSEURS et leur donner les droit d'accès en lecture sur la table FOURNISSEURS et sur le log. Les droits d'accès en lecture ont déjà été donné sur FOURNISSEURS.

Opérations réalisées en local

```
CREATE MATERIALIZED VIEW LOG ON FOURNISSEURS;  
GRANT SELECT ON MLOG$_FOURNISSEURS TO mleral;
```

Tests de bon fonctionnement

14 Sur le site Europe du Sud

14.1 Rappel binôme responsable

Le binôme responsable du site Europe du Sud est composé de Mohamed CHALAL et Mathieu LE RAL.

14.2 Objectifs

14.3 Liste des répliquions prévues

14.4 Analyse

14.5 Mise en oeuvre des répliquions

14.5.1 Répliquat du fragment xxx

Opérations réalisées localement

Message émis au site maître

Réponse du site maître

Test de vérification de bon fonctionnement de la réplication

Evolutions éventuelles des contraintes d'intégrité

Evolutions éventuelles des vues et des synonymes

14.6 Demandes d'autres sites

14.6.1 Fragment xxx

Demande du site xxx

Description de la demande et suivi

Analyse de la demande

Opérations réalisées en local

Tests de bon fonctionnement

15 Sur le site Amérique

15.1 Rappel binôme responsable

Le binôme responsable du site Europe du Sud est composé de Marah GALY ADAM et Aina RASOLDIER.

15.2 Objectifs

15.3 Liste des répliques prévues

15.4 Analyse

15.5 Mise en oeuvre des répliques

15.5.1 Réplicat du fragment xxx

Opérations réalisées localement

Message émis au site maître

Réponse du site maître

Test de vérification de bon fonctionnement de la réplication

Evolutions éventuelles des contraintes d'intégrité

Evolutionnements éventuelles des vues et des synonymes

15.6 Demandes d'autres sites

15.6.1 Fragment xxx

Demande du site xxx

Description de la demande et suivi

Analyse de la demande

Opérations réalisées en local

Tests de bon fonctionnement

16 Bilan global des répliques

Sixième partie

Requêtes distribuées : tests et optimisations

17 Site Europe du Nord

18 Site Europe du Sud

19 Site Amérique