



# TD Distribution – Replication

Donnée Repartie

Paul Orhon

LP – MiAR – Université de Nantes

2 février 2018

## Table des matières

<b>1</b>	<b>Notre bibliothèque s'agrandit</b>	<b>2</b>
1.1	Question 1 . . . . .	2
1.2	Question 2 . . . . .	2
1.3	Question 3 . . . . .	3
1.4	Question 4 . . . . .	3
1.5	Question 5 . . . . .	4
1.6	Question 6 . . . . .	4
<b>2</b>	<b>La centralisation des adhérents</b>	<b>5</b>
2.1	Question 7 . . . . .	5
2.2	Question 8 . . . . .	5
2.3	Question 9 . . . . .	5

## Listings

1	Trigger : lock Exemple	2
2	Trigger : ISBN exemplaire ref livre	2
3	Trigger : lock livre BD1	2
4	Trigger : ISBN livre ref exemplaire	3
5	Trigger : lock exemplaire	3
6	Trigger : mise à jour des valeur de livre	3
7	Trigger : Max exemplaire DB2	4
8	Un procédure de <i>Lecture</i> sur BD1	4
9	Log d'adhérent sur BD1	5
10	View d'adhérent sur BD2	5
11	Trigger max exemplaire emprunter version BD1	5

# 1 Notre bibliothèque s’agrandit

## 1.1 Question 1

Les tables requises sur DB2 sont :

- Emprunt
- Exempleaire
- ExempleaireEmprunte

Pour le détail voir le fichier CreationServeur2.sql

Pour éviter des problème de cohérence des donnée il faut verrouiller les table présente sur les deux bd avant un **insert** ou un **update**.

```
1 create or replace TRIGGER Lock_Exempleaire_Ins_Up
2 BEFORE INSERT OR UPDATE ON Exempleaire
3 BEGIN
4   LOCK TABLE Exempleaire IN SHARE MODE;
5   LOCK TABLE Exempleaire@vers_Angers IN SHARE MODE;
6 END;
```

Listing 1 – Trigger : lock Exempleaire

## 1.2 Question 2

Pour que le champ **ISBN** de la table **EXEMPLAIRE** de la BD2 référence au champ **ISBN** de la table **LIVRE** de la BD1 il faut ajouter un trigger.

```
1 CREATE OR replace TRIGGER Isbn_ex_ref_livre_to_BD1
2   AFTER INSERT OR UPDATE on EXEMPLAIRE
3   DECLARE
4     rec_ex EXEMPLAIRE%rowtype;
5   begin
6     SELECT * INTO rec_ex
7       FROM EXEMPLAIRE
8       WHERE ISBN not IN
9         (SELECT LIVRE.ISBN FROM LIVRE@vers_Nantes);
10    raise_application_error (-20202, 'viol de fk ISBN');
11  EXCEPTION
12    WHEN no_data_found then NULL;
13    WHEN too_many_rows then
14      raise_application_error (-20202, 'viol de fk ISBN');
15  END;
```

Listing 2 – Trigger : ISBN exempleaire ref livre

Ce trigger permet, aprêr un **insert** ou un **update** de la table **EXEMPLAIRE** de BD2, de vérifier la présence des **ISBN** de cette table dans la table **LIVRE** de BD1. Si le **selecte** trouve 0 enregistrement il n’y a pas de problème, mais si il trouve 1 ou plusieurs enregistrement il lève une exception.

Pour vérifier que le champ **numA** de la table **EMPRUNT** du serveur BD2, référence le champ **numA** de la table **ADHERENT** du serveur BD1 on met en place un trigger identique au précédant (1.2) mais en replacent **EXEMPLAIRE** par **EMPRUNT** et **LIVRE** par **ADHERENT** et en changeant le deuxième selecte par **ADHERENT.NUMA**.

Il faut aussi créer un trigger pour verrouiller les table lors de changement. Avant un **insert** ou un **update** sur **EXEMPLAIRE** du serveur BD2 il faut verrouiller la table **LIVRE** :

```
1 CREATE OR replace TRIGGER Lock_Livre_toBD1
2   BEFORE INSERT OR UPDATE ON EXEMPLAIRE
3   BEGIN
4     lock table LIVRE@vers_Nantes in share mode;
5   end;
```

Listing 3 – Trigger : lock livre BD1

Et pour un **insert** ou un **update** sur la table **EMPRUNT** du serveur BD2 il faut verrouiller la table **ADHERENT**.

### 1.3 Question 3

Pour que le champ **ISBN** de la table **LIVRE** de la BD1 référence au champ **ISBN** de la table **EXEMPLAIRE** de la BD2 il faut ajouter un trigger.

```

1 CREATE OR replace TRIGGER Isbn_livre_isRefBy_ex_to_BD2
2   AFTER DELETE ON LIVRE
3   DECLARE
4     rec_ex EXEMPLAIRE@vers_Angers%rowtype;
5   begin
6     SELECT * INTO rec_ex
7       FROM EXEMPLAIRE@vers_Angers
8       WHERE ISBN not IN
9         (SELECT LIVRE.ISBN FROM LIVRE);
10    raise_application_error (-20202,'viol de fk ISBN. Enfant dans BD2');
11  EXCEPTION
12    WHEN no_data_found then NULL;
13    WHEN too_many_rows then
14      raise_application_error (-20202,'viol de fk ISBN. Enfants dans BD2');
15  END;
```

Listing 4 – Trigger : ISBN livre ref exemplaire

Ce trigger permet, avant un **delete** sur la table **LIVRE** de BD1, de vérifier la présence des **ISBN** de cette table dans la table **EXEMPLAIRE** de BD2. Si le **selecte** trouve 0 enregistrement il n'y a pas de problème, mais si il trouve 1 ou plusieurs enregistrement il lève une exception.

Pour vérifier que le champ **numA** de la table **ADHERENT** du serveur BD1, référence le champ **numA** de la table **EMPRUNT** du serveur BD2 on met en place un trigger identique au précédant (1.3) mais en remplaçant **LIVRE** par **ADHERENT** et **EXEMPLAIRE** par **EMPRUNT** et en changeant le deuxième selecte par **ADHERENT.NUMA**.

Il faut aussi créer un trigger pour verrouiller les table lors de changement. Avant un **delete** sur **LIVRE** il faut verrouiller la table **EXEMPLAIRE** du serveur BD2 :

```

1 CREATE OR replace TRIGGER Lock_ex_toBD2
2   BEFORE DELETE ON LIVRE
3   BEGIN
4     lock table EXEMPLAIRE@vers_Angers in share mode;
5   end;
```

Listing 5 – Trigger : lock exemplaire

Et pour un **delete** sur la table **ADHERENT** il faut verrouiller la table **EMPRUNT** du serveur BD2.

### 1.4 Question 4

Pour mettre à jour le compteur, la date et le lieu d'emprunt dans la table **LIVRE** de BD1, on peut copier et modifier le trigger déjà présent dans BD1.

```

1 create or replace trigger up_val_livre_DB1
2   AFTER INSERT ON EXEMPLAIREEMPRUNTE for each row
3   BEGIN
4     UPDATE LIVRE@vers_Nantes
5       SET NBE = NBE+1,
6         DATEDE = :NEW.dateE,
7         LIEUDE = 'Angers'
8     WHERE ISBN = (
9       SELECT ISBN
10      FROM EXEMPLAIRE
11      where EXEMPLAIRE.NUMINV = :NEW.numInv
12    );
13  END;
```

Listing 6 – Trigger : mise à jour des valeur de livre

## 1.5 Question 5

Pour cette question il ne faut pas oublier les triggers qui lock les table **EXEMPLAIREEMPRUNTE** dans les deux serveur pour éviter les problèmes. Ils ont été réaliser à la question 1.

Ensuite on reprend le trigger déjà réaliser sur le serveur 1 qui permet de ne pas emprunter plus de 5 exemplaire que l'on modifie et que l'on place sur les deux serveur, en remodifiant les @vers\_.\*.

```

1 create or replace TRIGGER max_exem_emp
2   AFTER INSERT on EXEMPLAIREEMPRUNTE
3   declare
4     ligne_exem_emp ADHERENT@vers_Nantes%rowtype;
5 begin
6   SELECT * into ligne_exem_emp FROM ADHERENT@vers_Nantes
7     WHERE (
8       (SELECT COUNT(*) FROM EXEMPLAIREEMPRUNTE exN
9        WHERE ADHERENT.NUMA=exN.NUMA)
10      +
11      (SELECT COUNT(*) FROM EXEMPLAIREEMPRUNTE@vers_Nantes exA
12       WHERE ADHERENT.NUMA=exA.NUMA)
13     )>5;
14   raise_application_error (-20001, 'un adhérent a emprunté plus de 5 exemplaires');
15 exception
16   when no_data_found then null;
17   when too_many_rows then
18     raise_application_error (-20002, 'des adhérents ont emprunté plus de 5 exemplaires');
19 end;
```

Listing 7 – Trigger : Max exemplaire DB2

## 1.6 Question 6

Sur BD1 il faut seulement modifier le package *Lecture* pour prendre en compte BD2. Pour le package *Maj* il n'est pas utile de prendre en compte BD2 car les mise à jour des table doit s'effectuer seulement sur la base actuelle.

Sur BD2 il faut metre à jour le package *Lecture* pour prendre en compte BD1, et le package *Maj* pour réaliser les modification utile sur BD1 (la création d'adhérent).

```

1 PROCEDURE listEmpruntsForAdherent(idAdh NUMBER, liste out liste_Cursor) IS
2 BEGIN
3   SET TRANSACTION READ ONLY;
4   OPEN liste FOR select * from(
5     SELECT Exempleire.NUMINV, Livre.titre
6     FROM Livre, EXEMPLAIRE
7     WHERE Livre.ISBN = EXEMPLAIRE.ISBN
8     AND EXEMPLAIRE.numInv IN (
9       SELECT numInv
10      FROM ExempleireEmprunte
11      WHERE numA = idAdh
12    )
13
14    Union all
15
16    SELECT exA.NUMINV, Livre.titre
17    FROM Livre, EXEMPLAIRE@vers_Angers exA
18    WHERE Livre.ISBN = exA.ISBN
19    AND exA.numInv IN (
20      SELECT numInv
21      FROM ExempleireEmprunte@vers_Angers
22      WHERE numA = idAdh
23    )
24  );
25 COMMIT;
26 END;
```

Listing 8 – Un procédure de *Lecture* sur BD1

## 2 La centralisation des adhérents

### 2.1 Question 7

Les principaux problèmes posés par le partage d'une table en lecture/écriture est la cohérence des données et la mise à jour de ces données. Par exemple si un adhérent avec un **numA** à 5 est créer sur la table de BD1 et de BD2 on ne sait pas le quelle prendre en compte l'or de la mise en commun des données.

### 2.2 Question 8

Pour réaliser le partage en lecture seule de la table adhérent de BD1 vers BD2, il faut mètre en place une table de log d'adhérent.

```
1 CREATE MATERIALIZED view log on Adherent;
```

Listing 9 – Log d'adhérent sur BD1

Puis sur BD2 il faut créer la vue qui va, toute les minute, récupérer les mise à jour présente dans les log de la table adhérent.

```
1 CREATE MATERIALIZED view Adherent_view
2   refresh force
3   start with sysdate
4   next sysdate+1/1440
5   as select * from Adherent;
```

Listing 10 – View d'adhérent sur BD2

### 2.3 Question 9

Il suffi de prendre le trigger existant et d'ajouter un select vers la table **exemplaireEmprunte** de l'autre serveur.

```
1 create or replace TRIGGER max_exem_emp
2   AFTER INSERT on EXEMPLAIREEMPRUNTE
3   declare
4     ligne_exem_emp ADHERENT%rowtype;
5 begin
6   SELECT * into ligne_exem_emp FROM ADHERENT
7   WHERE (
8     (SELECT COUNT(*) FROM EXEMPLAIREEMPRUNTE exN
9      WHERE ADHERENT.NUMA=exN.NUMA)
10    +
11    (SELECT COUNT(*) FROM EXEMPLAIREEMPRUNTE@vers_Angers exA
12     WHERE ADHERENT.NUMA=exA.NUMA)
13   )>5;
14   raise_application_error (-20001, 'un adhérent a emprunté plus de 5 exemplaires');
15 exception
16   when no_data_found then null;
17   when too_many_rows then
18     raise_application_error (-20002, 'des adhérents ont emprunté plus de 5 exemplaires');
19 end;
```

Listing 11 – Trigger max exemplaire emprunter version BD1