

Bases de données réparties: TP2 (exam0506)

Première partie:

Plusieurs solutions envisageables:

- Créer de nouvelles tables et faire des vues des anciennes tables :

F/G = français / anglais

TypeAG(codeTAG,codeTAF,codeTAA,LibelleTAF,LibelleTAA) => TypeAction / KOA

Lieu(codeLieu,libelleLieu) => Lieu

ActionG(codeAG,CodeActF,codeActA,code lieu,codeTAG,LibelleA)

ActFundG(codeAF,codeAG,date,montant)

TypeAction :

```
SELECT codeTAF AS codeTA, libelleTAF AS libelleTA
```

```
FROM TypeAG
```

```
WHERE codeTAF NOT NULL
```

trigger :

```
INSTEAD OF INSERT ON TypeAction INSERT
```

```
INTO TypeAG(codeTAG,codeTAF,SEQTAG.NEXTVAL,:NEW.CodeTA,:NEW.LibelleTA);
```

codeTAG pourrait être alphanumérique de la forme : <lieu_provenance><numéro> pour résoudre des problèmes en cas d'ajout de nouveaux sites.

Partie 2:

2) Réplication multimaitre synchrone : pas utile ici car les données ne sont pas critiques, pas de risques de conflits

multimaitre / monomaitre : dépend de la volumétrie des données de chaque site (ex : multimaitre pas utile pour 20ko de MAJ / jour)

vue matérialisée

Partie 3:

1) Comme les données ne sont modifiables qu'à Paris, il suffit de faire des vues matérialisées sur les sites de Bruxelles et Madrid

Ligne(numligne, villedép, villearr, duree, capacite)

Convoi(numligne, date, heuredép,nbplacesprises)

```
CREATE DATABASE LINK Paris CONNECT TO user IDENTIFIED BY pass USING 'Paris'
```

```
CREATE MATERIALIZED VIEW Ligne
```

```
REFRESH FORCE
```

```
START WITH sysdate
```

```
NEXT sysdate+1
```

```
WITH PRIMARY KEY
```

```
AS
```

```
SELECT * FROM Ligne@Paris;
```

```
CREATE MATERIALIZED VIEW Convoi  
REFRESH FORCE  
START WITH sysdate  
NEXT sysdate+1/24  
WITH PRIMARY KEY  
AS  
SELECT * FROM Convoi@Paris;
```

2) Création d'un vue globale Transporteur
TransporteurP(numtrans, nomtrans,adressesetans) (fragment à Paris)

```
CREATE DATABASE LINK Madrid CONNECT TO user IDENTIFIED BY pass USING 'Madrid'  
CREATE DATABASE LINK Bruxelles CONNECT TO user IDENTIFIED BY pass USING  
'Bruxelles'
```

```
CREATE VIEW Transporteur  
AS SELECT * FROM TransporteurP  
UNION  
SELECT * FROM TransporteurB@Bruxelle  
UNION  
SELECT * FROM TransporteurM@Madrid;
```

3) Requete pour connaitre le transporteur ayant le plus gros montant de factures en cours
FactureP(numtrans,numconvoi,montant,etat) (fragment à Paris)

```
CREATE VIEW Facture  
AS SELECT * FROM FactureP  
UNION  
SELECT * FROM FactureB@Bruxelle  
UNION  
SELECT * FROM FactureM@Madrid;
```

```
CREATE VIEW FactEnc  
AS SELECT numtrans, SUM(montant) AS Enc  
FROM Facture  
WHERE etat='en cours'  
GROUP BY numtrans
```

```
SELECT nomtrans, Enc  
FROM Transporteur T, FactEnc F  
WHERE T.numtrans = F. numtrans  
AND F.Enc = (SELECT MAX(Enc) FROM FactEnc);
```

4) Solutions:
a/ MM synchrone (multi maitre)
b/ VM + trigger
c/ MM asynchrone
d/ Accès direct table distante

Solutions les plus propres: a ou d.

5) On peut s'assurer de ça en plaçant des lock.