Fragmentation de la table Client

Sous ENSIAS 1 :

Create table Client_1 as select *No*, Nom, Prénom, Adresse from client where Ville = ‘Casablanca’;

COPY FROM nassar/nassar2020@ensias1
TO nassar/nassar2020@ensias2
Create Client_2 USING select *No*, Nom, Prénom, Adresse from client where Ville = ‘Rabat’;

Agence à ne pas fragmenter.

Fragmentation de la table Compte

Sous ENSIAS 1 :

Create table Compte_1 as select * from Compte where **Client_No** IN (select No from Client Where Ville =‘Casablanca’);

COPY FROM nassar/nassar2020@ensias1
TO nassar/nassar2020@ensias2
Create Compte_2 USING select * from Compte where **Client_No** IN (select No from Client Where Ville =‘Rabat’);

Fragmentation de la table Operation

Sous ENSIAS 1 :

Create table Operation_1 as select * from Operation where **Compte_No** IN (select No from Compte Where Client_No IN (select No from Client where Ville ='Casablanca'));

COPY FROM nassar/nassar2020@ensias1
TO nassar/nassar2020@ensias2

Create Operation_2 USING select * from Operation where **Compte_No** IN (select No from Compte Where Client_No IN (select No from Client where Ville ='Rabat'));

Déplacement de la table Type_Compte

Sous ENSIAS 1 :

COPY FROM nassar/nassar2020@ensias1
TO nassar/nassar2020@ensias2

Create Type_Compte_2 USING select * from Type_Compte;

Déplacement de la table Type_Operation

Sous ENSIAS 1 :

COPY FROM nassar/nassar2020@ensias1
TO nassar/nassar2020@ensias2

Create Type_Operation_2 USING select * from Type_Operation;

3) Création des liens inter-bases (DATABASE LINK)

Sous ENSIAS 1 :
CREATE DATABASE LINK dl_ensias1
CONNECT TO **nassar** IDENTIFIED **nassar2020** USING **'ENSIAS2'**;

Test du lien : select * from Client_2@dl_ensias1;

Sous ENSIAS 2 : il faut se connecter à ENSIAS2
CREATE DATABASE LINK dl_ensias2
CONNECT TO **nassar** IDENTIFIED **nassar2020** USING **'ENSIAS1'**;

Test du lien : select * from Client_1@dl_ensias2;

3) Ajout des contraintes

Sous ENSIAS 1 :

ALTER TABLE CLient_1 ADD CONSTRAINT PK_Client_1 PRIMARY KEY (No);

Sous ENSIAS 2 :

ALTER TABLE CLient_2 ADD CONSTRAINT PK_Client_2 PRIMARY KEY (No);

ENSIAS 1 : Agence (*No*, Nom, Adresse, Ville) -- > Delete , Update OF 'No', ~~Inset~~

ENSIAS 2 : Compte_2 (*No*, Type_Compte_No, DateOuverture, Decouvert_autorise, Solde, *Client_No*, #*Agence_No*)

--> Insert, UPDATE OF 'Agence_No', ~~Delete~~

Sous ENSIAS 1 :

```
CREATE TRIGGER Trig_Agence_Compte_2
BEFORE Delete OR Update OF 'No' ON Agence
FOR EACH ROW
Declare
x number:=0;
Begin
Select COUNT(*) IN x  from Compte_2@dl_ensias1 where Agence_No=:OLD.No;
If x<> 0 then RAISE_APPLICATION_ERROR(-20750, 'Operation interdite : agence utilisée dans Compte_2');
End if ;
End;
/
```

ENSIAS 1 : **Agence** (*No*, Nom, Adresse, Ville) -- > Delete , Update OF 'No', ~~Inset~~

ENSIAS 2 : **Compte_2** (*No*, Type_Compte_No, DateOuverture, Decouvert_autorise, Solde, *Client_No*, #*Agence_No*)
--> Insert, UPDATE OF 'Agence_No', ~~Delete~~

Sous ENSIAS 2 :

```
CREATE TRIGGER Trig_Compte_2_Agence
BEFORE INSERT OR UPDATE OF 'Agence_No' ON Compte_2
FOR EACH ROW
Declare
x number:=0;
Begin
Select COUNT(*) IN x  from Agence@dbl_ensias1 where No=:NEW.Agence_No;
If x = 0 then RAISE_APPLICATION_ERROR(-207501, 'Operation interdite : agence inconnue);
End if ;
End;
/
```

5) Création des objets virtuels

Sous Base locale :

CREATE DATABASE LINK dl_locale_1
CONNECT TO **nassar** IDENTIFIED **nassar2020** USING **'ENSIAS1'**;

Test du lien : select * from Client_1@dl_locale_1;

CREATE DATABASE LINK dl_locale_2
CONNECT TO **nassar** IDENTIFIED **nassar2020** USING **'ENSIAS2'**;

Test du lien : select * from Client_2@dl_locale_2;

CREATE VIEW **Client** (*No*, Nom, Prénom, Adresse, Ville) AS
Select *No*, Nom, Prénom, Adresse, 'Casablanca' from Client_1@dl_locale_1
UNION
Select *No*, Nom, Prénom, Adresse, 'Rabat' from Client_2@dl_locale_2;

CREATE VIEW Agence AS Select * From Agence@dl_local_1; ou CREATE SYNONYM Agence FOR Agence@dl_locale_1;

CREATE SYNONYM Seq_Client FOR Seq_Client@dl_locale_1;

5) Création des objets virtuels

Sous Base locale :

Select * from client;
Select * from Agence;

Déclencheur pour Insert via la vue client :

```
CREATE TRIGGER  Trig_Insert_Client
INSTEAD OF INSERT ON CLIENT
FOR EACh ROW
BEGIN
IF :NEW.Ville = 'Casablanca' THEN Insert into Client_1@dl_locale_1 values(:NEW.No, :New.Nom, :New.Prenom,
:New.Adresse);
ELSIF :NEW.Ville = 'Rabat' THEN Insert into Client_2@dl_locale_2 values(:NEW.No, :New.Nom, :New.Prenom,
:New.Adresse);
ELSE RAISE_APPLICATION_ERROR(-20175, 'La ville Client ne peut être que Rabat ou Casablanca');
END IF;
END;
/
```

Test : Insert into Client values(Seq_Client.NEXTVAL, 'Filali','rachid', 'Rue 1','Casablanca');

Déclencheur pour Delete via la vue client :

```
CREATE TRIGGER  Trig_Insert_Client
INSTEAD OF Delete ON CLIENT
FOR EACh ROW
BEGIN
IF :OLD.Ville = ‘Casablanca’ THEN Delete From Client_1@dbl_locale_1 where No=:OLD.No;
ELSIF :NEW.Ville = ‘Rabat’ THEN Delete From Client_2@dbl_locale_2 where No=:OLD.No;
END IF;
END;
/
```

Test : Delete from Client where No=5;

TP2 : Dupllication Synchrone/Asynchrone

I- Mise en œuvre de la réplication synchrone

Sous ENSIAS 1

```
CREATE OR REPLACE TRIGGER MAJ_Images_APPAREIL
AFTER INSET OR DELETE OR UPDATE ON Appareil
FOR EACH ROW
BEGIN
IF ISERTING THEN
INSERT INTO Copie_Appareil@Dbl_BD_Locale
VALUES(( :NEW.No_Appareil, :NEW.Designation,
:NEW. Prix,
:NEW.caracteristiques_techniques) ;
ELSIF DELETING THEN
Delete Copie_Appareil@Dbl_BD_Locale  where No_Appareil= :OLD.No_Appareil ;
ELSIF UPDATING THEN
UPDATE Copie_Appareil@Dbl_BD_Locale
SET No_Appareil = :NEW.No_Appareil,
DEsignation = :NEW.Designation,
Prix= :NEW. Prix,
caracteristiques_techniques = :NEW.caracteristiques_techniques
Where No_Appareil= :OLD.No_Appareil;

END IF;
END;
/
```

II- Mise en œuvre de la réplication asynchrone

Sous ENSIAS 1 :
CREATE SNAPSHOT LOG ON Appareil;

Sous ENSIAS 2 :
CREATE SNAPSHOT Image_Appareil
REFRESH FAST
START WITH SYSDATE
NEXT SYSDATE+1/(24*60)
AS SELECT * FROM Appareil@Dbl_ENSIAS1;

Test :
SOUS ENSIAS1 :
INSERT INTO Appareil Values(100,'machine', 10000, 'NbTourParMinutes=10') ;
COMMIT ;
Patientez 1 Min et afficher Image_Appareil ;

Travail à réaliser :
Réplication asynchrone Commande Oracle : CREATE UPDATEABLE SNAPSHOT ….
Tester l'hétérogénéité enterme de SGB Différents : Oracle, SQL Server

Date limite pour l'envoi des comptes rendus (TP1 et TP2) : 26 Avril 2020 à 00h00.
Adresse email : mahmoud.nassar@um5.ac.ma