Université du Havre
Faculté des Sciences & Techniques
Département Informatique

Master 2 SRO 2005/2006

BASE DE DONNÉES RÉPARTIE

M. Nakechbandi

TP PROJET
Conception et Implantation d'une base de données répartie - Application à l'Administration d'une Organisation Non Gouvernementale

Sébastien LOISEL
Jean-Philippe MOMMER
Soizic QUÉGUINER DE OLIVEIRA
Jérôme WERMESTER

Rendu le 1er Février 2006

Table des matières

1 Schémas conceptuels de la base de donnée **5**
1.1 Modèle Entités/Associations . . . 5
1.2 Modèle relationnel . . . 6
1.3 Modèle de répartition . . . 6

2 Site 1 : base centrale **8**
2.1 Alias de la base de données . . . 8
2.2 Entrées dans le fichier tnsnames.ora . . . 8
2.2.1 Description . . . 9
2.3 Lien de base de données base centrale . . . 9
2.3.1 Description . . . 9
2.3.2 Utilité . . . 9
2.4 Création d'un utilisateur . . . 9
2.4.1 Description . . . 10
2.5 Procédure insert_into_table . . . 10
2.5.1 Instruction SQL . . . 10
2.5.2 Description . . . 10
2.5.3 Utilité . . . 10
2.6 Procédure update_table . . . 10
2.6.1 Instruction SQL . . . 10
2.6.2 Description . . . 11
2.6.3 Utilité . . . 11
2.7 Table Type_action . . . 11
2.7.1 Instruction SQL . . . 11
2.7.2 Description . . . 11
2.7.3 Utilité . . . 11
2.8 SEQUENCE seq_type_action . . . 11
2.8.1 Description . . . 11
2.8.2 Utilité . . . 11
2.9 Déclencheur trig_cle_prim_type_action . . . 12
2.9.1 Description . . . 12
2.9.2 Utilité . . . 12
2.10 Table Materiel . . . 12
2.10.1 Instruction SQL . . . 12
2.10.2 Description . . . 12
2.10.3 Utilité . . . 12
2.11 SEQUENCE seq_materiel . . . 12
2.11.1 Description . . . 12
2.11.2 Utilité . . . 12
2.12 Déclencheur trig_cle_prim_materiel . . . 13
2.12.1 Description . . . 13
2.12.2 Utilité . . . 13
2.13 Table Pays . . . 13
2.13.1 Instruction SQL . . . 13
2.13.2 Description . . . 13
2.13.3 Utilité . . . 13
2.14 Table Annexe . . . 13
2.14.1 Instruction SQL . . . 13
2.14.2 Description . . . 13
2.14.3 Utilité . . . 14
2.15 SEQUENCE seq_annexe . . . 14
2.15.1 Description . . . 14

2.15.2 Utilité . . . 14
2.16 Déclencheur trig_cle_prim_annexe . . . 14
2.16.1 Description . . . 14
2.16.2 Utilité . . . 14
2.17 Table Personnel . . . 14
2.17.1 Instruction SQL . . . 14
2.17.2 Description . . . 15
2.17.3 Utilité . . . 15
2.18 SEQUENCE seq_personnel . . . 15
2.18.1 Description . . . 15
2.18.2 Utilité . . . 15
2.19 Vue personnel_totale . . . 15
2.19.1 Instruction SQL . . . 15
2.19.2 Description . . . 15
2.19.3 Utilité . . . 15
2.20 Déclencheur personnel_base . . . 15
2.20.1 Description . . . 16
2.20.2 Utilité . . . 16
2.21 Vue site_totale . . . 16
2.21.1 Description . . . 16
2.21.2 Utilité . . . 17
2.22 Vue action_totale . . . 17
2.22.1 Description . . . 17
2.22.2 Utilité . . . 17
2.23 Table Affectation . . . 17
2.23.1 Instruction SQL . . . 17
2.23.2 Description . . . 17
2.23.3 Utilité . . . 17
2.24 Vue affectation_totale . . . 17
2.24.1 Description . . . 17
2.24.2 Utilité . . . 18
2.25 Déclencheur trig_affectation . . . 18
2.25.1 Description . . . 19
2.25.2 Utilité . . . 20
2.26 SEQUENCE seq_action . . . 20
2.26.1 Description . . . 20
2.26.2 Utilité . . . 20
2.27 Déclencheur trig_action_centrale . . . 20
2.27.1 Description . . . 21
2.27.2 Utilité . . . 21
2.28 SEQUENCE seq_site . . . 21
2.28.1 Description . . . 21
2.28.2 Utilité . . . 21
2.29 Déclencheur trig_site_centrale . . . 21
2.29.1 Description . . . 22
2.29.2 Utilité . . . 22
2.30 Vue materiel_utilise_totale . . . 22
2.30.1 Description . . . 22
2.30.2 Utilité . . . 22
2.31 Déclencheur trig_materiel_utilise_centrale . . . 22
2.31.1 Description . . . 23
2.31.2 Utilité . . . 23

3 Site 2 : base locale 1 **24**

3.1 Entrées dans le fichier tnsnames.ora . . . 24
3.2 Création d'un utilisateur/administrateur 'annexe' pour la base locale . . . 24
3.2.1 Instruction SQL . . . 24
3.2.2 Description . . . 25
3.3 Création d'un utilisateur 'central' pour la base locale . . . 25
3.3.1 Instruction SQL . . . 25
3.3.2 Description . . . 25
3.4 Lien de base de données base centrale . . . 25
3.4.1 Instruction SQL . . . 25
3.4.2 Description et utilité . . . 25
3.5 Table Id_annexe . . . 25
3.5.1 Instruction SQL . . . 25
3.5.2 Description . . . 25
3.6 Table Personnel . . . 26
3.6.1 Instruction SQL . . . 26
3.6.2 Description . . . 26
3.7 Table Site . . . 26
3.7.1 Instruction SQL . . . 26
3.7.2 Description . . . 26
3.8 Table Action . . . 26
3.8.1 Instruction SQL . . . 26
3.8.2 Description . . . 27
3.9 Table Materiel_utilise . . . 27
3.9.1 Instruction SQL . . . 27
3.9.2 Description . . . 27
3.10 Table Affectation . . . 27
3.10.1 Instruction SQL . . . 27
3.10.2 Description . . . 28
3.11 Synonyme personnel_totale . . . 28
3.11.1 Instruction SQL . . . 28
3.11.2 Description . . . 28
3.11.3 Utilité . . . 28
3.12 Synonyme site_totale . . . 28
3.12.1 Instruction SQL . . . 28
3.12.2 Description . . . 28
3.12.3 Utilité . . . 28
3.13 Synonyme action_totale . . . 28
3.13.1 Instruction SQL . . . 28
3.13.2 Description . . . 28
3.13.3 Utilité . . . 29
3.14 Synonyme materiel_utilise_totale . . . 29
3.14.1 Instruction SQL . . . 29
3.14.2 Description . . . 29
3.14.3 Utilité . . . 29
3.15 Synonyme affectation_totale . . . 29
3.15.1 Instruction SQL . . . 29
3.15.2 Description . . . 29
3.15.3 Utilité . . . 29
3.16 Déclencheur personnel_local . . . 29
3.16.1 Instruction SQL . . . 29
3.16.2 Description . . . 30
3.16.3 Utilité . . . 30
3.17 Déclencheur site_local . . . 30
3.17.1 Instruction SQL . . . 30

3.17.2 Description . . . 31
3.17.3 Utilité . . . 31
3.18 Déclencheur trig_action_local . . . 31
3.18.1 Instruction SQL . . . 31
3.18.2 Description . . . 32
3.18.3 Utilité . . . 32
3.19 Déclencheur mat_util_local . . . 32
3.19.1 Instruction SQL . . . 32
3.19.2 Description . . . 34
3.19.3 Utilité . . . 34
3.20 Déclencheur trig_affectation . . . 34
3.20.1 Instruction SQL . . . 34
3.20.2 Description . . . 36
3.20.3 Utilité . . . 36
3.21 Vue matérialisée vue_mat_pays . . . 37
3.21.1 Instruction SQL . . . 37
3.21.2 Description . . . 37
3.21.3 Utilité . . . 37
3.22 Vue matérialisée vue_mat_type_action . . . 37
3.22.1 Instruction SQL . . . 37
3.22.2 Description . . . 37
3.22.3 Utilité . . . 37
3.23 Vue matérialisée vue_mat_materiel . . . 37
3.23.1 Instruction SQL . . . 37
3.23.2 Description . . . 38
3.23.3 Utilité . . . 38
3.24 Vue matérialisée vue_mat_annexe . . . 38
3.24.1 Instruction SQL . . . 38
3.24.2 Description . . . 38
3.24.3 Utilité . . . 38

4 Des requêtes distribuées **39**
4.1 Requête PL/SQL de la base centrale . . . 39
4.2 Requête PL/SQL de la base locale . . . 42

5 Annexes **47**
5.1 Script PL/SQL de la base centrale . . . 47
5.2 Script PL/SQL de la base locale . . . 59

Notre conception d'un système de gestion de bases de données réparti concerne l'organisation de bases de données et de tables contenant des informations utilisées par une Organisation Non Gouvernementale (ONG).

Une ONG est une organisation indépendante des gouvernements et dont le champs d'action est international. Dans notre cas, l'organisation possède une vocation humanitaire. De part sa politique d'action globale elle est amenée à intervenir en de nombreux lieux sur la planète, là où des désastres économiques, sociaux ou humains prennent place.

Nous considérons que notre organisation possède un siège central et de multiples annexes disséminées de part le monde. Le premier organe cité possède la majorité des prérogatives en ce qui concerne le fonctionnement général, cependant des tâches sont déléguées aux différentes administrations locales.

Plus particulièrement nous représentons dans notre modélisation la gestion géographique des actions menées, leur composition, l'affectation du personnel envers ces opérations et la logistique matérielle nécessaire à la réalisation des différents projets.

1 Schémas conceptuels de la base de donnée

1.1 Modèle Entités/Associations

Après réflexion, plusieurs entités et associations sont générées afin de satisfaire notre analyse. Il en resulte un schéma (cf. Fig. 1.1).

Ensuite il convient de définir une cardinalitée propre à chaque association de composants.

Le choix des indices numériques présentés ci dessus est justifié, dans notre modèle, par le fait que :

- un membre du personnel ne possède qu'une seule nationalité (1,1) et un pays existant dans la base peut être la patrie d'aucun ou d'un nombre fini indéterminé d'employés(0,N) ;

- un membre du personnel n' est pas affecté ou est affecté à un nombre indéterminé d'actions (0,N) , qui sont cependant distinctes temporellement, on utilise pour cela des marqueurs de début et de fin. En revanche une action implique obligatoirement une personne y prenant part voir plus si affinités (1,N) ;

- une action est caractérisée par une seule catégorie (1,1) mais un type d'action peut définir plusieurs actions(0,N) ;

- une annexe est situé intrinsèquement dans un seul pays (1,1) mais cet élément peut rassembler nulle annexe comme plusieurs (0,N) ;

- un pays contient aucun ou plusieurs sites d'actions (0,N) qui ne sont localisés que sur un seul pays (1,1) ;

- une annexe gère un nombre indéterminé de sites ou ne possède aucune responsabilité(0,N) . Un site est dépendant d'une seule et unique annexe (1,1) ;

- une action se déroule uniquement sur un site (1,1) alors qu'un site peut être le théâtre de plusieurs opérations (0,N) ;

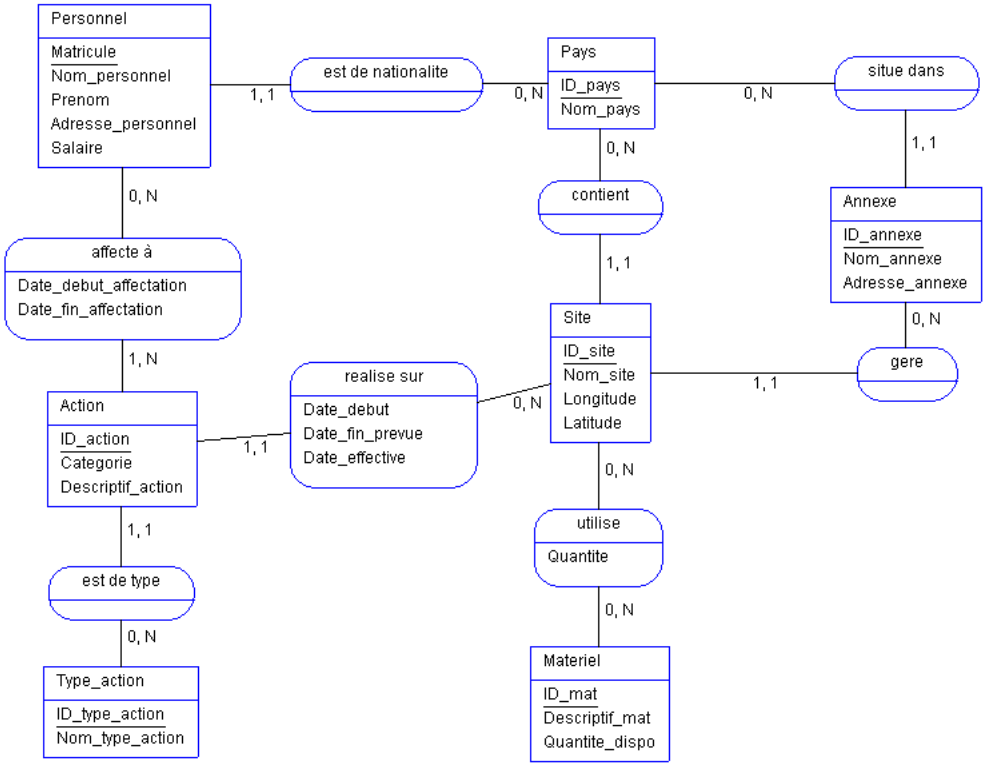

FIG. 1 – Modèle Entitées/Associations

– un site a en sa possession pas ou plusieurs type de matériel (0,N) , un type de matériel peut être utilisé par plusieurs sites (0,N).
.

1.2 Modèle relationnel

L'application au schéma précédent des règles de transformation du modèle E/A vers le modèle relationnel permet de construire l'agencement des tables suivant (cf Fig. 1.2).

1.3 Modèle de répartition

Afin d'implanter une base de données réparties, certaines spécificités sont définies quant à l'organisation de l' ONG.

Le siège de l'organisation assure la gestion des informations en rapport aux annexes et de fait aux pays dans lesquelles elles se trouvent. Il prend en charge également les données relatives aux membres du personnel salariés, leurs affectations, les différentes catégories d'actions et les stocks de matériel. Lieu de décision principal il possède la base de données "centrale".

La manipulation des informations concernant les sites d'intervention, les actions concrètes et le matériel utilisé incombe aux annexes, puisque ces éléments dépendent administrativement de celles-ci. De même le personnel bénévole est affilié indirectement à ces instances suppléantes par l'intermédiaire de l'opération qu'il intègre, ainsi les informations qui lui sont associées sont gérées par les annexes. Chacune de ces extensions de l'organisation administre une base "locale", nommée

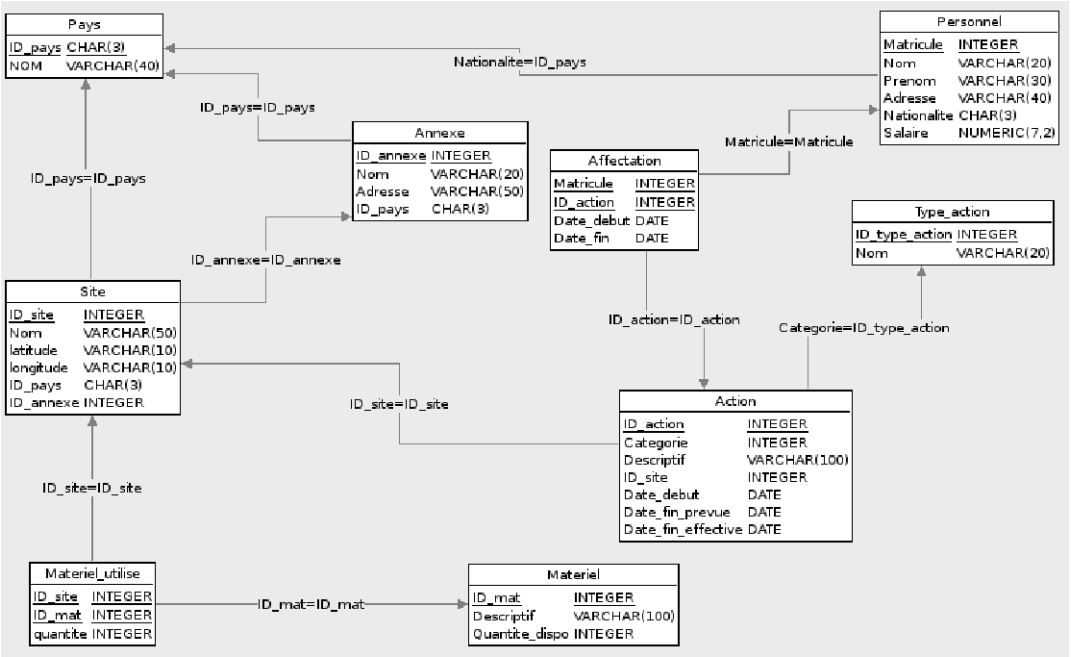

FIG. 2 – Modèle relationnel

ainsi au sens géographique du terme.

Il se dégage des informations précédentes et d'une analyse fonctionnelle de notre base de données réparties un schéma de réplication et de fragmentation (cf. Fig. 1.3).

Pour satisfaire ce schéma de répartition des tables il a fallu adapter voir octroyer certains élément à notre base de données.

Cela s'applique notamment aux contraintes d'intégrité référentielle qu'il a parfois fallu supprimer d'une table et remplacer par un déclencheur vérifiant l'existence du tuple associé dans une table distante.

De plus les bases locales possèdent une table non définie dans le modèle relationnel, celle-ci contient un unique tuple composé d'un seul élément représentant l'identifiant de l'annexe utilisée lors des insertions et ceux pour vérifier qu'elles sont exécutée sur les tables de l'annexe associée aux informations.

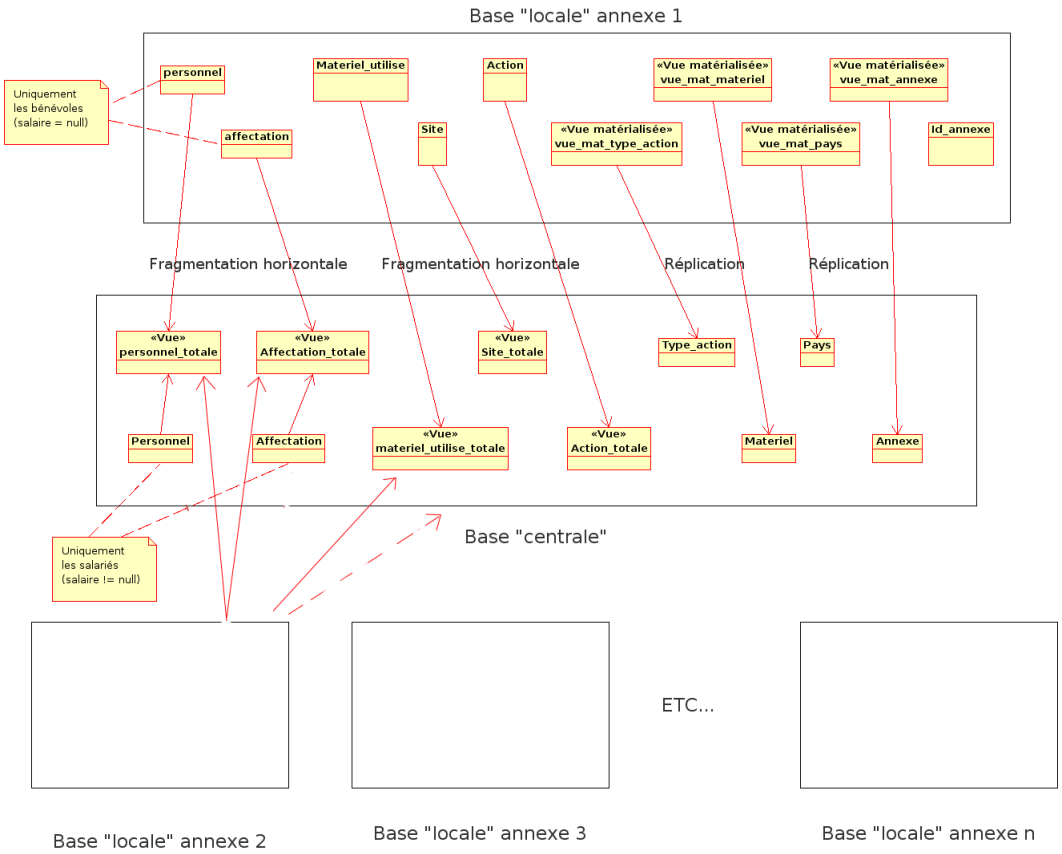

FIG. 3 – Modèle de répartition et de fragmentation

2 Site 1 : base centrale

2.1 Alias de la base de données

Alias de la base de données pour se connecter sur les différentes annexes : DB_JEROME

2.2 Entrées dans le fichier tnsnames.ora

```
DB_JEROME =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 172.16.20.109)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
    )
  )

ORCL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 172.16.21.11)(PORT = 1521))
    )
    (CONNECT_DATA =
```

```
      (SERVICE_NAME = orcl)
    )
  )

EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC))
    )
    (CONNECT_DATA =
      (SID = orcl)
      (PRESENTATION = RO)
    )
  )
```

2.2.1 Description

Cet alias permet de référencer et, dans un second temps, de se connecter à la base de données d'une annexe située sur le serveur ayant pour adresse IP 172.16.20.109 sur le port de connexion 1521 (port d'écoute par défaut d'Oracle).

2.3 Lien de base de données base centrale

```
CREATE PUBLIC DATABASE LINK annexe_1
   CONNECT TO central IDENTIFIED BY central\\
   using 'DB_JEROME';
```

2.3.1 Description

Ce lien permet la connexion à la base de données distante située sur une annexe. Les différents paramètres sont le nom d'utilisateur (ici, central) de la base de données distante ainsi que le mot de passe de celui-ci sur la base distante. Il faut aussi fournir l'alias de base de données défini précédemment dans le fichier tnsnames.ora et qui référence la base de données ORCL sur l'annexe dont l'adresse IP est 172.16.20.109.

2.3.2 Utilité

Ce lien permet par la suite de facilement référencer des objets situés sur la base distante.

2.4 Création d'un utilisateur

```
grant connect,resource to annexe identified by annexe;

grant select on central.materiel_utilise_totale to annexe;
grant select on central.personnel_totale to annexe;
grant select on central.action_totale to annexe;
grant select on central.site_totale to annexe;
grant select on central.affectation_totale to annexe;

grant select on central.pays to annexe;
grant select on central.type_action to annexe;
grant select on central.annexe to annexe;
grant select on central.materiel to annexe;

grant select on central.seq_personnel to annexe;
```

```
grant select on central.seq_site to annexe;
grant select on central.seq_action to annexe;
grant select on central.seq_annexe to annexe;
grant select on central.seq_materiel to annexe;
grant select on central.seq_type_action to annexe;
```

2.4.1 Description

L'utilisateur 'annexe' avec son mot de passe 'annexe' a le droit de connection, peut créer des tables, consulter toutes les tables de la base locale.

2.5 Procédure insert_into_table

2.5.1 Instruction SQL

```
CREATE OR REPLACE Procedure insert_into_table (
      dans  VARCHAR2,
      val    VARCHAR2)
      IS
   stmt_str    VARCHAR2(200);

BEGIN
   stmt_str := 'INSERT INTO ' || dans || ' values
      ( '|| val || ')';

   EXECUTE IMMEDIATE stmt_str ;
END;
/
```

2.5.2 Description

Cette procédure permet d'insérer simplement dans une table passée en paramètre sous forme de chaï£¡e de caractères des valeurs, elles aussi passées en paramètre.

2.5.3 Utilité

Cette procédure nous permet d'insérer des données dans n'importe quelle table de n'importe quelle base et ceci de façon dynamique. Dans notre situation, il nous faut parfois insérer des données dans une base donnée, base située en dehors du poste sur lequel est exécutée la requête, ce qui nous permet donc de créer dynamiquement le nom de la table et les valeur à insérer.

2.6 Procédure update_table

2.6.1 Instruction SQL

```
CREATE OR REPLACE Proc\'edure update_table (
      dans  VARCHAR2,
      col VARCHAR2,
      val    VARCHAR2,
      id_name VARCHAR2,
      id VARCHAR2
      )
      IS
   stmt_str    VARCHAR2(200);

BEGIN
```

```
   stmt_str := 'UPDATE ' || dans || '
      set ' || col || ' = ' ||
      val || ' WHERE ' ||
      id_name || ' = ' || id ;

   EXECUTE IMMEDIATE stmt_str ;
END;
/
```

2.6.2 Description

Cette procédure permet de mettre à jour simplement une table avec des conditions dynamiques.

2.6.3 Utilité

Dans notre situation, il nous mettre à jour dynamiquement des tables qui ne sont pas forcément situées en local sur notre machine. Par conséquent, nous pouvons construire dynamiquement le nom de la table à mettre à jour, ainsi que les différentes conditions de mise à jour.

2.7 Table Type_action

2.7.1 Instruction SQL

```
CREATE Table Type_action (
       ID_type_action INTEGER NOT NULL
     , Nom VARCHAR(20) NOT NULL
     , PRIMARY KEY (ID_type_action)
);
```

2.7.2 Description

La table Type_action permet de lister les différentes actions types qui sont disponibles.

2.7.3 Utilité

Cette table permet de simplifier la gestion des actions qui sont menées sur les différents sites. A chaque action est rattachée une catégorie d'action, listées dans cette table.

2.8 SEQUENCE seq_type_action

```
CREATE SEQUENCE seq_type_action
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;
```

2.8.1 Description

Cette séquence permet de gérer l'identifiant constituant la clé primaire de la table Type_action.

2.8.2 Utilité

seq_type_action permet une gestion facile et simple d'une clé primaire avec un contrôle de concurrence.

2.9 Déclencheur trig_cle_prim_type_action

```
CREATE OR REPLACE TRIGGER trig_cle_prim_type_action
BEFORE INSERT ON Type_action
FOR EACH ROW
BEGIN
SELECT seq_type_action.NEXTVAL INTO :NEW.ID_type_action from DUAL;
END;
/
```

2.9.1 Description

Ce déclencheur s'exécute lorsqu'une insertion est faite sur la table Type_action.

2.9.2 Utilité

La seule fonction de ce déclencheur est la génération de la clé primaire.

2.10 Table Materiel

2.10.1 Instruction SQL

```
CREATE Table Materiel (
       ID_mat INTEGER NOT NULL
     , Descriptif VARCHAR(100) NOT NULL
     , Quantite_dispo INTEGER NOT NULL
     , PRIMARY KEY (ID_mat)
);
```

2.10.2 Description

La table Materiel liste les différents matériels disponibles au sein de l'association.

2.10.3 Utilité

Cette table permet de gérer facilement les matériels disponibles avec leur quantité respective.

2.11 SEQUENCE seq_materiel

```
CREATE SEQUENCE seq_materiel
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;
```

2.11.1 Description

Cette séquence permet de gérer l'identifiant constituant la clé primaire de la table Materiel.

2.11.2 Utilité

seq_materiel permet une gestion facile et simple d'une clé primaire avec un contrôle de concurrence.

2.12 Déclencheur trig_cle_prim_materiel

```
CREATE OR REPLACE TRIGGER trig_cle_prim_materiel
BEFORE INSERT ON Materiel
FOR EACH ROW
BEGIN
SELECT seq_materiel.NEXTVAL INTO :NEW.ID_mat from DUAL;
END;
```

2.12.1 Description

Ce déclencheur s'exécute lorsqu'une insertion est faite sur la table Materiel.

2.12.2 Utilité

La seule fonction de ce déclencheur est la génération de la clé primaire.

2.13 Table Pays

2.13.1 Instruction SQL

```
CREATE Table Pays (
       ID_pays CHAR(3) NOT NULL
     , NOM VARCHAR(40) NOT NULL
     , PRIMARY KEY (ID_pays)
);
```

2.13.2 Description

La table Pays liste les différents pays.

2.13.3 Utilité

Cette table permet de gérer facilement les nationalités des personnels en entrant simplement l'identifiant du pays duquel est originaire la personne et permet aussi de permettre le référencement des annexes à un pays entré dans la base de données.

2.14 Table Annexe

2.14.1 Instruction SQL

```
CREATE Table Annexe (
       ID_annexe INTEGER NOT NULL
     , Nom VARCHAR(20) NOT NULL
     , Adresse VARCHAR(50) NOT NULL
     , ID_pays CHAR(3) NOT NULL
     , PRIMARY KEY (ID_annexe)
     , CONSTRAINT FK_Annexe_1 FOREIGN KEY (ID_pays)
                  REFERENCES Pays (ID_pays)
);
```

2.14.2 Description

La table Annexe liste les différents annexes de l'association.

2.14.3 Utilité

Cette table permet de gérer facilement les annexes de l'association avec des champs standards comme l'adresse et le pays de localisation.

2.15 SEQUENCE seq_annexe

```
CREATE SEQUENCE seq_annexe
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;
```

2.15.1 Description

Cette séquence permet de gérer l'identifiant constituant la clé primaire de la table Annexe.

2.15.2 Utilité

seq_annexe permet une gestion facile et simple d'une clé primaire avec un contrôle de concurrence.

2.16 Déclencheur trig_cle_prim_annexe

```
CREATE OR REPLACE TRIGGER trig_cle_prim_annexe
BEFORE INSERT ON Annexe
FOR EACH ROW
BEGIN
SELECT seq_annexe.NEXTVAL INTO :NEW.ID_annexe from DUAL;
END;
/
```

2.16.1 Description

Ce déclencheur s'exécute lorsqu'une insertion est faite sur la table Annexe.

2.16.2 Utilité

La seule fonction de ce déclencheur est la génération de la clé primaire.

2.17 Table Personnel

2.17.1 Instruction SQL

```
CREATE Table Personnel (
       Matricule INTEGER NOT NULL
     , Nom VARCHAR(20) NOT NULL
     , Prenom VARCHAR(30) NOT NULL
     , Adresse VARCHAR(40) NOT NULL
     , Nationalite CHAR(3) NOT NULL
     , Salaire NUMERIC(7,2)
     , PRIMARY KEY (Matricule)
     , CONSTRAINT FK_Personnels_1 FOREIGN KEY (Nationalite)
                  REFERENCES Pays (ID_pays)
);
```

2.17.2 Description

La table Personnel liste les personnels salariés de l'association.

2.17.3 Utilité

Cette table permet de gérer facilement les salariés de l'association avec des champs standards comme l'adresse et la nationalité ainsi que le salaire.

2.18 SEQUENCE seq_personnel

```
CREATE SEQUENCE seq_personnel
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;
```

2.18.1 Description

Cette séquence permet de gérer l'identifiant constituant la clé primaire de la table Personnel.

2.18.2 Utilité

seq_personnel permet une gestion facile et simple d'une clé primaire avec un contrôle de concurrence.

2.19 Vue personnel_totale

2.19.1 Instruction SQL

```
create view personnel_totale as select * from central.Personnel
   UNION select * from annexe.personnel@annexe_1;
```

2.19.2 Description

Cette vue permettra à n'importe qui sur n'importe quel site de consulter l'ensemble des personnels de l'association, salariés (sur la base centrale) ou bénévoles (sur les bases annexes).

2.19.3 Utilité

Cette vue est particulièrement intéressante pour les annexes pour consulter l'intégralité des salariés afin des les réquisitionner pour une tâche sur un de leur site.

2.20 Déclencheur personnel_base

```
create or replace trigger personnel_base
before insert on personnel
for each row
DECLARE
cursor c1 IS select * from annexe;
cursor c_pays IS select * from pays;
reponse varchar(2);
ck_pays INTEGER;

erreur_insertion EXCEPTION;
mauvais_pays EXCEPTION;
```

```
insertion_migree EXCEPTION;
BEGIN
if(:new.Salaire IS NULL) THEN
raise erreur_insertion;
ELSE
ck_pays := 0;
FOR cp IN c_pays LOOP
IF(:new.Nationalite = cp.ID_pays) THEN
ck_pays := 1;
END IF;
END LOOP;

if (ck_pays = 0) THEN raise mauvais_pays;
-- sinon, l'insertion se poursuit normalement
END IF;
-- A ce niveau, l'insertion est valide
-- on ajoute le num\'ero de s\'equence
select seq_personnel.nextval into :new.Matricule from Dual;
END IF;

EXCEPTION
when erreur_insertion THEN raise_application_error
   (-20001, 'Insertion impossible dans la base centrale d un benevole');
when mauvais_pays THEN raise_application_error
   (-20001, 'L identifiant du pays n est pas correct');
when insertion_migree THEN raise_application_error
   (-20001, 'L insertion a ete deplacee vers une base annexe');
END;
/
```

2.20.1 Description

Ce déclencheur s'exécute lorsqu'une insertion est faite sur la table Personnel.

2.20.2 Utilité

Le déclencheur personnel_base permet de tester la contrainte principale qui est le salaire qui doit être différent de NULL pour les personnels de la base centrale, ces personnels devant obligatoirement être des salariés. Si ceci n'est pas le cas, l'insertion est annulée. Ensuite, un test est effectué concernant la nationalité d'un personnel, si le test n'est pas concluant (un identifiant de pays qui ne correspond à aucun pays dans la base centrale), l'insertion n'est pas validée. Dans le cas contraire, l'insertion se poursuit normalement.

2.21 Vue site_totale

```
create view site_totale as select * from annexe.site@annexe_1;
```

2.21.1 Description

Cette vue permettra à n'importe qui sur n'importe quel site de consulter l'ensemble des sites de l'association, salariés (sur la base centrale) ou bénévoles (sur les bases annexes).

2.21.2 Utilité

Cette vue est particulièrement intéressante pour contrôler lors d'une insertion qu'elle concerne bien un site qui dépend de l'annexe dans laquelle l'insertion doit être faite (concernant la table Action par exemple).

2.22 Vue action_totale

```
create view action_totale as select * from annexe.Action@annexe_1;
```

2.22.1 Description

Cette vue permettra à n'importe qui sur n'importe quel site de consulter l'ensemble des actions de l'association qui sont réparties sur les différentes annexes.

2.22.2 Utilité

Cette vue est particulièrement intéressante pour la base centrale pour avoir une liste de l'intégralité des actions qui ont été menées au sein de l'association.

2.23 Table Affectation

2.23.1 Instruction SQL

```
CREATE Table Affectation (
       Matricule INTEGER NOT NULL
     , ID_action INTEGER NOT NULL
     , Date_debut DATE NOT NULL
     , Date_fin DATE NOT NULL
     , PRIMARY KEY (Matricule, ID_action)
     , CONSTRAINT FK_Affectation_1 FOREIGN KEY (Matricule)
                  REFERENCES Personnel (Matricule)
);
```

2.23.2 Description

La table Affectation liste les affectations des personnels salariés de l'association à des actions sur les tables annexes.

2.23.3 Utilité

Cette table permet de gérer facilement les affectations de salariés pour des actions précises avec une date de début et une date de fin pour l'affectation (dates ne correspondant pas forcément à celles de l'action en question).

2.24 Vue affectation_totale

```
CREATE VIEW affectation_totale AS SELECT * from affectation
   UNION SELECT * from annexe.affectation@annexe_1;
```

2.24.1 Description

Cette vue permet à toute personne située sur tout site de consulter l'intégralité des affectations de toutes les personnes de l'association.

2.24.2 Utilité

affectation_totale permet entre autre de savoir, pour une base annexe, si une personne n'est pas déjà affectée à une autre action sur un site qui n'est pas rattaché à l'annexe qui souhaite faire une nouvelle affectation.

2.25 Déclencheur trig_affectation

```
CREATE OR REPLACE TRIGGER trig_affectation
BEFORE INSERT ON affectation
FOR EACH ROW
DECLARE
date_erronee EXCEPTION;
deja_aff EXCEPTION;
insert_interdit EXCEPTION;
mauvaise_annexe EXCEPTION;
pb_reference EXCEPTION;
insertion_migree EXCEPTION;

dateDebAct action_totale.date_debut%TYPE;
dateFinAct action_totale.date_fin_prevue%TYPE;

CURSOR c1 IS SELECT date_debut, date_fin
   FROM affectation_totale WHERE matricule = :NEW.matricule;
dateDebAff action_totale.date_debut%TYPE;
dateFinAff action_totale.date_fin_prevue%TYPE;

sal personnel.salaire%TYPE;
mat personnel.Matricule%TYPE;
id_an annexe.id_annexe%TYPE;
BEGIN
SELECT date_debut INTO dateDebAct FROM action_totale
   WHERE ID_action=:NEW.ID_action;
SELECT Matricule INTO mat FROM personnel_totale
   WHERE matricule = :NEW.matricule;

IF(dateDebAct is null or mat is null) THEN
raise pb_reference;
END IF;

/* Verifier que la date de debut est chronologiquement
   inferireure a la date de fin */
IF (:NEW.date_debut > :NEW.date_fin
   OR :NEW.date_debut = :NEW.date_fin) THEN
RAISE date_erronee ;
END IF;

/* Si une affectation concernant une action d\'ebute avant celle-ci  */
IF (dateDebAct > :NEW.date_debut) THEN
RAISE date_erronee ;
END IF;

/* Si une affectation concernant une action se termine apr\‘es celle-ci
 * c'est possible (action est prolong\'ee)
```

```
 * mais demander confirmation de l'utilisateur
 */
IF (dateFinAct < :NEW.date_fin) THEN

dbms_output.put_line('ATTENTION: L affectation nouvellement entr\'ee
   se termine apr\'es la date de fin pr\'evue pour l action ');

END IF;

/* Verifier que l'utilisateur n'est pas deja affecte durant la periode
   de l'affectation candidate a l'insertion */
OPEN c1;
LOOP
FETCH c1 INTO dateDebAff, dateFinAff;
EXIT WHEN c1%NOTFOUND;

IF ((dateDebAff < :NEW.date_fin AND dateFinAff > :NEW.date_fin)
   OR (dateFinAff > :NEW.date_debut AND dateFinAff < :NEW.date_fin)) THEN
      RAISE deja_aff ;
END IF;
END LOOP;
CLOSE c1;

/* Determiner si l'insertion doit ï£¡re effectuee en base centrale
   (salaire != NULL) ou en base locale (salaire = NULL) */
SELECT salaire INTO sal FROM personnel_totale WHERE matricule = :NEW.matricule;

IF (sal = NULL) THEN
SELECT id_annexe INTO id_an from site_totale where id_site =
   (SELECT id_site FROM action_totale WHERE ID_action = :NEW.ID_action);

insert_into_table('annexe.affectation@annexe_'||id_an, :NEW.matricule||' ,
   '||:NEW.ID_action||' , '||:NEW.date_debut||' , '||:NEW.date_fin);
raise insertion_migree;
END IF;

EXCEPTION
WHEN date_erronee THEN raise_application_error
   (-20001, ' Les dates ne sont pas conforme aux dates de l action. ');
WHEN deja_aff THEN raise_application_error
   (-20001, ' Cette personne est d\'ej\'a affect\'ee durant cette p\'eriode. ');
WHEN pb_reference THEN raise_application_error
   (-20001, ' Un tuple reference n existe pas ');
WHEN insertion_migree THEN raise_application_error
   (-20001, ' L insertion a ete deplacee sur une base annexe ');
END;
/
```

2.25.1 Description

Ce déclencheur s'exécute lorsqu'une insertion est faite sur la table Affectation.

2.25.2 Utilité

Le déclencheur trig_affectation permet de tester les différentes contraintes d'intégrité. L'insertion doit, entre autre, faire correspondre l'affectation à une action existante dans une des différentes bases (centrale ou annexe). Il en est de même pour l'identifiant du personnel entré, celui-ci devant être valable avant d'autoriser l'insertion du tuple. Bien entendu, ce déclencheur test aussi si la nouvelle affectation n'affecte pas une personne alors qu'elle est déjà affectée à une autre action sur un site quelconque. Enfin, si le salaire n'est pas entré (= NULL), l'insertion est migrée sur la base annexe correspondante. Dans le cas contraire (salaire différent de NULL), l'insertion se poursuit sur la base centrale.

2.26 SEQUENCE seq_action

```
CREATE SEQUENCE seq_action
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;
```

2.26.1 Description

Cette séquence permet de gérer l'identifiant constituant la clé primaire de la table Action (sur les bases annexes).

2.26.2 Utilité

seq_action permet une gestion facile et simple d'une clé primaire avec un contrôle de concurrence.

2.27 Déclencheur trig_action_centrale

```
create or replace trigger trig_action_centrale
INSTEAD OF insert on action_totale
for each row
declare
pb_reference EXCEPTION;

categ Type_action.ID_type_action%TYPE;
num_site site_totale.ID_site%TYPE;
num_annexe_insert site_totale.ID_annexe%TYPE;
id_action INTEGER;
begin
select ID_type_action into categ from Type_action
   where ID_type_action = :new.Categorie;
select ID_site into num_site from site_totale
   where ID_site = :new.ID_site;

if ( categ is null or num_site is null ) then raise pb_reference;
end if;

select ID_annexe into num_annexe_insert from site_totale
   where ID_site = :new.ID_site;

if ( num_annexe_insert is not null ) then
select seq_action.nextval into id_action from dual;
```

```
insert_into_table('annexe.Action@annexe_'||num_annexe_insert,
   id_action ||' , '||:new.Categorie ||' , '||:new.Descriptif||' ,
   '||:new.ID_site||' , '||:new.Date_debut||' ,
   '||:new.Date_fin_prevue||' , '||:new.Date_fin_effective);
else raise pb_reference;
end if;

EXCEPTION
when pb_reference THEN raise_application_error
   ( -20001, ' Votre insertion r\'ef\'erence un tuple inexistant ' );
end;
/
```

2.27.1 Description

Ce déclencheur s'exécute lorsqu'une insertion est faite sur la vue action_totale.

2.27.2 Utilité

Le déclencheur trig_action_centrale est positionné sur la vue action_totale pour permettre à tout site d'insérer des données dans la table Action qui correspond. Cela permet aussi de faire un contrï£¡e concernant l'autorisation d'insérer une action (une base annexe ne peut modifier que ses propres tables). Evidemment, ce déclencheur teste si la catégorie d'action est valide et si le site entré l'est aussi. Enfin, trig_action_centrale insérera les données dans la base correspondante.

2.28 SEQUENCE seq_site

```
CREATE SEQUENCE seq_site
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;
```

2.28.1 Description

Cette séquence permet de gérer l'identifiant constituant la clé primaire de la table Site (sur les bases annexes).

2.28.2 Utilité

seq_site permet une gestion facile et simple d'une clé primaire avec un contrôle de concurrence.

2.29 Déclencheur trig_site_centrale

```
create or replace trigger trig_site_centrale
INSTEAD OF insert on site_totale
for each row
declare
pb_reference EXCEPTION;

num_pays Pays.ID_pays%TYPE;
num_annexe Annexe.ID_annexe%TYPE;
id_nv_site INTEGER;
begin
select ID_pays into num_pays from Pays where ID_pays = :new.ID_pays;
```

```
select ID_annexe into num_annexe from Annexe where ID_annexe = :new.ID_annexe;

if ( num_pays is null or num_annexe is null ) then raise pb_reference;
end if;

select seq_site.nextval into id_nv_site from dual;
insert_into_table( 'annexe.Site@annexe_'||num_annexe, id_nv_site||' ,
   '||:new.Nom||' , '||:new.latitude||' , '||:new.longitude||' ,
   '||:new.ID_pays||' , '||:new.ID_annexe);

EXCEPTION
when pb_reference THEN raise_application_error
   ( -20001, ' Votre insertion r\'ef\'erence un tuple inexistant ' );
end;
/
```

2.29.1 Description

Ce déclencheur s'exécute lorsqu'une insertion est faite sur la vue site_totale.

2.29.2 Utilité

Le déclencheur trig_site_centrale est positionné sur la vue site_totale pour permettre à toute annexe d'insérer des données dans la table Site qui correspond. Ce déclencheur teste si le pays du nouveau site existe et si le numéro d'annexe correspond bien à une annexe existante. Enfin, l'insertion est faite dans la base correspondante.

2.30 Vue materiel_utilise_totale

```
create view materiel_utilise_totale as select * from annexe.Materiel_utilise@annexe_1;
```

2.30.1 Description

Cette vue permet à toute annexe de consulter l'intégralité des entrées de toutes les tables Materiel_utilise au sein de toutes les bases distribuées.

2.30.2 Utilité

Cette vue est particulièrement intéressante pour consulter l'ensemble des matériels alloués sur les différents sites.

2.31 Déclencheur trig_materiel_utilise_centrale

```
create or replace trigger trig_materiel_utilise_centrale
INSTEAD OF insert on materiel_utilise_totale
for each row
declare
pb_reference EXCEPTION;

num_site site_totale.ID_site%TYPE;
num_annexe_insert site_totale.ID_annexe%TYPE;
begin
select ID_site into num_site from site_totale
   where ID_site = :new.ID_site;
```

```
if ( num_site is null ) then raise pb_reference;
end if;


select ID_annexe into num_annexe_insert from site_totale
   where ID_site = :new.ID_site;

if ( num_annexe_insert is not null ) then
insert_into_table('annexe.Materiel_utilise@annexe_'||num_annexe_insert,
   :new.ID_site||' , '||:new.ID_mat||' , '||:new.quantite);
else raise pb_reference;
end if;

EXCEPTION
when pb_reference THEN raise_application_error
   ( -20001, ' Votre insertion r\'ef\'erence un tuple inexistant ' );
end;
/
```

2.31.1 Description

Ce déclencheur s'exécute lorsqu'une insertion est faite sur la vue materiel_utilise_totale.

2.31.2 Utilité

Le déclencheur trig_materiel_utilise_centrale est positionné sur la vue materiel_utilise_totale pour permettre à toute annexe d'insérer des données dans la table Materiel_utilise qui correspond. Ce déclencheur teste si le site entré est un site existant. Enfin, l'insertion est faite dans la base correspondante.

3 Site 2 : base locale 1

3.1 Entrées dans le fichier tnsnames.ora

Alias de base de données DB_SEB

```
# tnsnames.ora Network Configuration File:
#/usr/local/oracle/product/10.1.0/db_1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

DB_SEB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 172.16.21.11)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
    )
  )

ORCL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 172.16.20.109)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
    )
  )


EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC))
    )
    (CONNECT_DATA =
      (SID = orcl)
      (PRESENTATION = RO)
    )
  )
```

Cet alias permet de référencer puis de se connecter, si besoin est, à la base 'orcl' sur le serveur d'adresse IP '172.16.21.11' et aussi à la base 'orcl' sur le serveur d'adresse IP '172.16.20.109'

3.2 Création d'un utilisateur/administrateur 'annexe' pour la base locale

3.2.1 Instruction SQL

```
create user annexe identified by annexe;
grant connect, resource, dba to annexe;
```

3.2.2 Description

Cet utilisateur 'annexe' avec son mot de passe 'annexe' a le droit de connection, de pouvoir créer des tables, de pouvoir créer des utilisateurs, de gérer leur droit,...

3.3 Création d'un utilisateur 'central' pour la base locale

3.3.1 Instruction SQL

```
create user central identified by central;
grant connect, resource to central;

grant select on ID_ANNEXE to central;
grant select on PERSONNEL to central;
grant select on SITE to central;
grant select on ACTION to central;
grant select on MATERIEL_UTILISE to central;
grant select on AFFECTATION to central;
```

3.3.2 Description

Cet utilisateur 'central' avec son mot de passe 'central' a le droit de connection et de pouvoir créer des tables. Il a également le droit de consulter toutes les tables de la base locale.

3.4 Lien de base de données base centrale

3.4.1 Instruction SQL

```
CREATE PUBLIC DATABASE LINK base_centrale CONNECT TO annexe
IDENTIFIED BY annexe using 'DB_SEB';
```

3.4.2 Description et utilité

Ce lien crée une sorte de "référence" sur la base distance située sur le site de la base centrale. Il utilise les paramètres de connexion 'annexe' (utilisateur) et 'annexe' (mot de passe). La création du lien s'appuie sur l'alias de base de données 'DB_SEB' déclaré sur le serveur local de la base locale, cet alias référençant la base 'DB_SEB' sur le serveur d'adresse IP '172.16.21.11' (base_centrale). Ce lien permet par la suite de référencer des objets se trouvant sur la base distante.

3.5 Table Id_annexe

3.5.1 Instruction SQL

```
CREATE TABLE Id_annexe (
       id INTEGER NOT NULL
     , PRIMARY KEY (id)
);
```

3.5.2 Description

id contient l'identifiant de l'annexe

3.6 Table Personnel

3.6.1 Instruction SQL

```
CREATE TABLE Personnel (
       Matricule INTEGER NOT NULL
     , Nom VARCHAR(20) NOT NULL
     , Prenom VARCHAR(30) NOT NULL
     , Adresse VARCHAR(40) NOT NULL
     , Nationalite CHAR(3) NOT NULL
     , Salaire NUMERIC(7,2)
     , PRIMARY KEY (Matricule)
);
```

3.6.2 Description

Matricule contient l'identifiant de l'employé
Nom contient le nom de l'employé
Prenom contient le prénom de l'employé
Adresse contient l'adresse de l'employé
Nationalité contient la nationalité de l'employé
Salaire contient le salaire mensuel de l'employé

3.7 Table Site

3.7.1 Instruction SQL

```
CREATE TABLE Site (
       ID_site INTEGER NOT NULL
     , Nom VARCHAR(50) NOT NULL
     , latitude VARCHAR(10)
     , longitude VARCHAR(10)
     , ID_pays CHAR(3) NOT NULL
     , ID_annexe INTEGER NOT NULL
     , PRIMARY KEY (ID_site)
);
```

3.7.2 Description

*ID_*Site contient l'identifiant du site
Nom contient le nom du site
latitude contient la latitude géographique du site
longitude contient la longitude géographique du site
ID_ pays contient l'identifiant du pays où est localisé le site
ID_ annexe contient l'identifiant de l'annexe qui gère le site.

3.8 Table Action

3.8.1 Instruction SQL

```
CREATE TABLE Action (
       ID_action INTEGER NOT NULL
     , Categorie INTEGER NOT NULL
     , Descriptif VARCHAR(100) NOT NULL
```

```
     , ID_site INTEGER NOT NULL
     , Date_debut DATE NOT NULL
     , Date_fin_prevue DATE NOT NULL
     , Date_fin_effective DATE
     , PRIMARY KEY (ID_action)
     , CONSTRAINT FK_Actions_2 FOREIGN KEY (ID_site)
                  REFERENCES Site (ID_site)
);
```

3.8.2 Description

ID_action contient l'identifiant de l'action
Categorie contient l'identifiant du type de l'action
Descriptif contient un bref descriptif de l'action
ID_site contient l'identifiant du site où l'action est menée
Date_debut contient la date de début de l'action
Date_fin_prevue contient la date de fin prévue de l'action
Date_fin_effective contient la date de fin réelle de l'action, celle-ci une fois terminée.

3.9 Table Materiel_utilise

3.9.1 Instruction SQL

```
CREATE TABLE Materiel_utilise (
       ID_site INTEGER NOT NULL
     , ID_mat INTEGER NOT NULL
     , quantite INTEGER NOT NULL
     , PRIMARY KEY (ID_site, ID_mat)
     , CONSTRAINT FK_Materiels_sur_site_1 FOREIGN KEY (ID_site)
                  REFERENCES Site (ID_site)
);
```

3.9.2 Description

ID_site contient l'identifiant du site où le matériel est utilisé
ID_mat contient l'identifiant du matériel
quantite contient le quantité de matériel utilisé

3.10 Table Affectation

3.10.1 Instruction SQL

```
CREATE TABLE Affectation (
       Matricule INTEGER NOT NULL
     , ID_action INTEGER NOT NULL
     , Date_debut DATE NOT NULL
     , Date_fin DATE NOT NULL
     , PRIMARY KEY (Matricule, ID_action)
     , CONSTRAINT FK_Affectation_1 FOREIGN KEY (Matricule)
                  REFERENCES Personnel (Matricule)
     , CONSTRAINT FK_Affectation_2 FOREIGN KEY (ID_action)
                  REFERENCES Action (ID_action)
);
```

3.10.2 Description

Matricule contient l'identifiant de l'employé affecté.
ID_action contient l'identifiant de l'action où l'employé est affecté
Date_debut contient la date de début d'affectation de l'employé
Date_fin contient la date de fin d'affectation de l'employé

3.11 Synonyme personnel_totale

3.11.1 Instruction SQL

```
CREATE SYNONYM personnel_totale
FOR central.personnel_totale@base_centrale;
```

3.11.2 Description

Ce synonyme permet d'accéder de manière transparente à la vue personnel_totale sur le serveur distant.

3.11.3 Utilité

L'utilité de ce synonyme est de pouvoir accéder à la liste de tous les employés travaillant pour l'ONG. Elle peut servir par exemple dans le cas de demande d'informations sur un employé que souhaite employé la base locale.

3.12 Synonyme site_totale

3.12.1 Instruction SQL

```
CREATE SYNONYM site_totale FOR
central.site_totale@base_centrale;
```

3.12.2 Description

Ce synonyme permet d'accéder de manière transparente à la vue site_totale sur le serveur distant.

3.12.3 Utilité

L'utilité de ce synonyme est de pouvoir accéder à la liste de tous les sites gérés par l'ONG.

3.13 Synonyme action_totale

3.13.1 Instruction SQL

```
CREATE SYNONYM action_totale FOR central.action_totale@base_centrale;
```

3.13.2 Description

Ce synonyme permet d'accéder de manière transparente à la vue action_totale sur le serveur distant.

3.13.3 Utilité

L'utilité de ce synonyme est de pouvoir accéder à la liste de toutes les actions réalisées par l'ONG.

3.14 Synonyme materiel_utilise_totale

3.14.1 Instruction SQL

```
CREATE SYNONYM materiel_utilise_totale FOR
central.materiel_utilise_totale@base_centrale;
```

3.14.2 Description

Ce synonyme permet d'accéder de manière transparente à la vue materiel_utilse_totale sur le serveur distant.

3.14.3 Utilité

L'utilité de ce synonyme est de pouvoir accéder à la liste de tous les matériels utilisés par l'ONG.

3.15 Synonyme affectation_totale

3.15.1 Instruction SQL

```
CREATE SYNONYM affectation_totale FOR
central.affectation_totale@base_centrale;
```

3.15.2 Description

Ce synonyme permet d'accéder de manière transparente à la vue affectation_totale sur le serveur distant.

3.15.3 Utilité

L'utilité de ce synonyme est de pouvoir accéder à la liste de toutes les affectations des employés l'ONG.

3.16 Déclencheur personnel_local

3.16.1 Instruction SQL

```
create or replace trigger personnel_local
before insert on personnel
for each row
DECLARE
        cursor c_pays IS select * from central.pays@base_centrale;
        ck_pays INTEGER;
        insert_interdit EXCEPTION;
        mauvais_pays EXCEPTION;
BEGIN
        if(:new.Salaire != NULL) THEN
```

```
                raise insert_interdit ;
ELSE
                select count(*) into ck_pays from vue_mat_pays
                where ID_pays = :new.Nationalite;

                if (ck_pays = 0 ) THEN raise mauvais_pays ;
                -- sinon, l'insertion se poursuit normalement
                END IF;
                -- A ce niveau, l'insertion est valide
                -- on ajoute le numÃľro de sÃľquence
                select central.seq_personnel.nextval@base_centrale
                into :new.Matricule from Dual;
        END IF;

EXCEPTION
        when insert_interdit THEN raise_application_error(-20001,
        'Insertion impossible dans une base locale d un
        salarie');
        when mauvais_pays THEN raise_application_error(-20001,
        'L identifiant du pays n est pas correct');
END;
/
```

3.16.2 Description

Ce trigger vérifie avant l'insertion d'un employé dans la base locale si cet employé est bénévole et si l'identifiant du pays est correct.

3.16.3 Utilité

Ce déclencheur permet de garantir la cohérence des données ainsi que la hiérarchie entre la base centrale et la base locale. L'insertion doit être autorisé uniquement pour un employé bénévole et l'identifiant de la table personnel est généré automatiquement par la base centrale.

3.17 Déclencheur site_local

3.17.1 Instruction SQL

```
create or replace trigger site_local
before insert on site
FOR EACH ROW
DECLARE
        ck_pays INTEGER;
        num_annexe ID_annexe.id%TYPE;
        mauvais_pays EXCEPTION;
        mauvaise_annexe EXCEPTION;
BEGIN
        select count(*) into ck_pays from vue_mat_pays
        where ID_pays = :new.ID_pays;

        IF (ck_pays = 0 ) THEN
                raise mauvais_pays ;
        ELSE
```

```
                -- sinon, l'insertion se poursuit
                select id into num_annexe from ID_annexe;
                IF(:new.ID_annexe != num_annexe) THEN
                        raise mauvaise_annexe;
                ELSE
                        -- sinon, les contraintes sont verifiees
                        select central.seq_site.nextval@base_centrale
                        into :new.ID_site from dual;
                END IF;
        END IF;
EXCEPTION
        WHEN mauvaise_annexe THEN raise_application_error(-20001,
        'Le site n est pas rattache a votre annexe');
        WHEN mauvais_pays THEN raise_application_error(-20001, 'Le
        site n est pas rattache a un pays valide');
END;
/
```

3.17.2 Description

Ce trigger vérifie avant l'insertion d'un site dans la base locale si ce site est géré par l'annexe qui a fait l'insertion et si l'identifiant du pays est correct.

3.17.3 Utilité

Ce déclencheur permet de garantir la cohérence des données ainsi que la hiérarchie entre les bases locales.
L'insertion doit être autorisé uniquement si l'annexe correspond. Ainsi, une base locale A ne peut pas attribuée un site à une base locale B. L'identifiant de la table site est généré automatiquement par la base centrale.

3.18 Déclencheur trig_action_local

3.18.1 Instruction SQL

```
CREATE OR REPLACE TRIGGER trig_action_local
BEFORE INSERT ON action
FOR EACH ROW
DECLARE
        pb_reference EXCEPTION;
        mauvaise_annexe EXCEPTION;

        id_cat vue_mat_type_action.id_type_action%TYPE;
        id_site site.id_site%TYPE;

        id_an vue_mat_annexe.id_annexe%TYPE;
        id_an_ici vue_mat_annexe.id_annexe%TYPE;

BEGIN
        /* Verifiant de l'existence des cles etrangeres */
        SELECT id_type_action INTO id_cat FROM vue_mat_type_action
        WHERE id_type_action = :NEW.categorie;
```

```
        SELECT id_site INTO id_site FROM site WHERE id_site = :NEW.id_site;

        IF (id_cat IS NULL OR id_site IS NULL) THEN
                raise pb_reference ;
        END IF;

        /* On verifie que l'insertion est bien effectuee dans la base
        de la bonne annexe */
        SELECT id_annexe INTO id_an from site_totale
        where id_site = :NEW.ID_site;

        SELECT id INTO id_an_ici from id_annexe;

        IF ( id_an = id_an_ici) THEN
                SELECT central.seq_action.nextval@base_centrale
                into :new.ID_action from dual;

        ELSE raise mauvaise_annexe ;

        END IF;

EXCEPTION
        WHEN pb_reference THEN raise_application_error(-20001,
        ' Votre insertion reference un tuple non existant. ');
        WHEN mauvaise_annexe THEN raise_application_error(-20001,
        'Le site n est pas rattache a votre annexe');
END;
/
```

3.18.2 Description

Ce trigger vérifie avant l'insertion d'une action dans la base locale si le site et la catégorie existe bien et également si ce site est géré par l'annexe qui a fait l'insertion

3.18.3 Utilité

Ce déclencheur permet de garantir la cohérence des données ainsi que la hiérarchie entre les bases locales.
L'insertion doit être autorisé uniquement si l'annexe correspond et si la catégorie de l'action existe. Ainsi, une base locale A ne peut pas attribuée une action à une base locale B L'identifiant de l'action est généré automatiquement par la base centrale

3.19 Déclencheur mat_util_local

3.19.1 Instruction SQL

```
create or replace trigger mat_util_local
before insert on materiel_utilise
for each row
DECLARE
        un_site site.ID_site%TYPE;
        un_materiel vue_mat_materiel.ID_mat%TYPE;
        num_annexe ID_annexe.id%TYPE;
```

```
        annexe_site vue_mat_annexe.ID_annexe%TYPE;
        qte_dispo vue_mat_materiel.Quantite_dispo%TYPE;
        i INTEGER;
        mauvais_site EXCEPTION;
        mauvais_materiel EXCEPTION;
        erreur_quantite EXCEPTION;
BEGIN
        select count(*) into i from site_totale
        where ID_site = :new.ID_site;

        IF(i = 1) THEN
                -- Reference bien un site existant
                select count(*) into i from materiel_utilise_totale
                where ID_mat = :new.ID_mat;

                IF(i = 1) THEN
                        -- Reference bien un materiel existant
                        select id into num_annexe from ID_annexe;
                        select a.ID_annexe into annexe_site
                        from vue_mat_annexe a, site s
                        where a.ID_annexe = s.ID_annexe
                        AND s.ID_site = :new.ID_site;
                        IF(annexe_site != num_annexe) THEN
                                -- Le numero de l'annexe local
                                -- ne corrsepond pas a
                                -- celui du site insere
                                raise mauvais_site;
                        ELSE
                                -- Inutile de faire un autre FETCH,
                                -- car le dernier doit etre le bon
                                -- (1 seul materiel avec la
                                -- meme cle primaire)
                                select quantite_dispo into qte_dispo
                                from central.materiel@base_centrale
                                where ID_mat = :new.ID_mat;
                                IF(:new.quantite > qte_dispo) THEN
                                        raise erreur_quantite;
                                -- sinon, insertion valable
                                ELSE
                                 update central.materiel@base_centrale
                                        set Quantite_dispo =
                                        (Quantite_dispo - :new.quantite)
                                        where ID_mat = :new.ID_mat;
                                        -- l'insertion se continue
                                        -- normalement
                                END IF;
                        END IF;
                ELSE
                        -- Erreur concernant un materiel
                        raise mauvais_materiel;
                END IF;
        ELSE
                -- Erreur concernant un site
                raise mauvais_site;
```

```
        END IF;

EXCEPTION
        when mauvais_site THEN raise_application_error(-20001,
        'Le site entre n est pas valable');
        when mauvais_materiel THEN raise_application_error(-20001,
        'Le materiel entre n est pas valable');
        when erreur_quantite THEN raise_application_error(-20001,
        'La quantite entree est trop importante');
END;
/
```

3.19.2 Description

Ce trigger vérifie avant l'insertion d'un matériel utilisé dans la base locale si le site et le matériel existe bien et également si ce site est géré par l'annexe qui a fait l'insertion. L'insertion est validée si la quantité demandée par la base locale est disponible au niveau de la base centrale. Si c'est le cas, ce trigger remet à jour le nombre de matériel dans la base centrale.

3.19.3 Utilité

Ce déclencheur permet de garantir la cohérence des données ainsi que la hiérarchie entre les bases locales.
L'insertion doit être autorisé uniquement si l'annexe correspond, si le matériel existe et si la base centrale dispose de la quantité demandé pour le matériel

3.20 Déclencheur trig_affectation

3.20.1 Instruction SQL

```
CREATE OR REPLACE TRIGGER trig_affectation
BEFORE INSERT ON affectation
FOR EACH ROW
DECLARE
        date_erronee EXCEPTION;
        deja_aff EXCEPTION;
        insert_interdit EXCEPTION;
        mauvaise_annexe EXCEPTION;
        pb_reference EXCEPTION;
        insertion_a_migrer EXCEPTION;

        dateDebAct action_totale.date_debut%TYPE;
        dateFinAct action_totale.date_fin_prevue%TYPE;

        CURSOR c1 IS SELECT date_debut, date_fin FROM affectation_totale
        WHERE matricule = :NEW.matricule;
        dateDebAff action_totale.date_debut%TYPE;
        dateFinAff action_totale.date_fin_prevue%TYPE;

        num_annexe Id_annexe.id%TYPE;
        annexe_affectation Id_annexe.id%TYPE;
        sal personnel.salaire%TYPE;
        mat personnel.Matricule%TYPE;
```

```
        id_an vue_mat_annexe.id_annexe%TYPE;
BEGIN
        SELECT date_debut INTO dateDebAct FROM action_totale
        WHERE ID_action=:NEW.ID_action;
        SELECT Matricule INTO mat FROM personnel_totale
        WHERE matricule = :NEW.matricule;

        IF(dateDebAct is null or mat is null) THEN
                raise pb_reference;
        END IF;

        /* Verifier que la date de debut est chronologiquement
        inferireure a la date de fin */
        IF (:NEW.date_debut > :NEW.date_fin OR :NEW.date_debut =
        :NEW.date_fin) THEN
                RAISE date_erronee ;
        END IF;

        /* Si une affectation concernant une action debute avant
        celle-ci  */
        IF (dateDebAct > :NEW.date_debut) THEN
                RAISE date_erronee ;
        END IF;

        /* Si une affectation concernant une action se termine
        apres celle-ci
         * c'est possible (action est prolongee) mais demander
        confirmation de l'utilisateur
         */
        IF (dateFinAct < :NEW.date_fin) THEN

                dbms_output.put_line('ATTENTION: L affectation
                nouvellement entree se termine apres la date
                de fin prevue pour l action ');

        END IF;

        /* Verifier que l'utilisateur n'est pas deja
        affecte durant la periode de l'affectation candidate
        Ãă l'insertion */
        OPEN c1;
                LOOP
                FETCH c1 INTO dateDebAff, dateFinAff;
                EXIT WHEN c1%NOTFOUND;

                        IF ((dateDebAff < :NEW.date_fin AND
                        dateFinAff > :NEW.date_fin) OR (dateFinAff
                        > :NEW.date_debut AND dateFinAff <
                        :NEW.date_fin)) THEN RAISE deja_aff ;
                        END IF;
                END LOOP;
        CLOSE c1;

        /* Determiner si l'insertion doit etre effectuee
```

```
        en base centrale (salaire != NULL) ou en base locale
        (salaire = NULL) */
        SELECT salaire INTO sal FROM personnel_totale
        WHERE matricule = :NEW.matricule;

        IF (sal != NULL) THEN
                raise insertion_a_migrer;
        ELSE
                select id into num_annexe from ID_annexe;
                select a.ID_annexe into annexe_affectation
                from vue_mat_annexe a, site s, action act
                where a.ID_annexe = s.ID_annexe
                AND s.ID_site = act.ID_site
                AND act.ID_action = :new.ID_action;
                IF(annexe_affectation != num_annexe) THEN
                        -- Le numero de l'annexe local ne
                        -- correspond pas a celui du site insere
                        raise mauvaise_annexe;
                END IF;
        END IF;

EXCEPTION
        WHEN date_erronee THEN raise_application_error(-20001,
        ' Les dates ne sont pas conforme aux dates de l action. ');
        WHEN deja_aff THEN raise_application_error(-20001,
        ' Cette personne est deja affectee durant cette
        periode. ');
        WHEN pb_reference THEN raise_application_error(-20001,
        ' Un tuple reference n existe pas ');
        WHEN insertion_a_migrer THEN
        raise_application_error(-20001, ' L insertion doit etre
        faite sur la base centrale ');
        WHEN mauvaise_annexe THEN raise_application_error(-20001,
        'Le site n est pas rattache a votre annexe');
END;
/
```

3.20.2 Description

Ce trigger vérifie avant l'insertion d'une affectation dans la base locale si l'action comporte bien une date de début, si l'employé existe bien et également si ce site est géré par l'annexe qui a fait l'insertion.
L'insertion est validée si la date de début de l'affectation est chronologiquement inférieure à la date de fin, si la date de début de l'action se situe chronologiquement avant la date de début d'affectation, si l'employé est libre pendant la période de l'affectation et bien sûr si cet employé est un bénévole. Un message à caractère informatif est affiché si l'affectation se termine après la date de fin prévue pour l'action.

3.20.3 Utilité

Ce déclencheur permet de garantir la cohérence des données ainsi que la hiérarchie entre les bases locales et la base centrale. L'insertion doit être autorisé uniquement si l'annexe correspond, si les dates sont cohérentes entres-elles et également par rapport aux dates d'action et aux dates

de disponibilité de l'employé, si l'employé existe et est bénévole.

3.21 Vue matérialisée vue_mat_pays

3.21.1 Instruction SQL

```
create materialized view vue_mat_pays
refresh complete
start with sysdate
next sysdate+1
as
select * from central.pays@base_centrale;
```

3.21.2 Description

Cette vue, réactualisée chaque jour permet à la base locale de connaÃőtre la liste des pays (avec leurs identifiants) présent dans la base de l'ONG.

3.21.3 Utilité

Cette vue est utile car elle permet à la base locale de connaÃőtre le code d'un pays avant une insertion afin de respecter les contraintes d'intégrité.

3.22 Vue matérialisée vue_mat_type_action

3.22.1 Instruction SQL

```
create materialized view vue_mat_type_action
refresh complete
start with sysdate
next sysdate+1
as
select * from central.type_action@base_centrale;
```

3.22.2 Description

Cette vue, réactualisée chaque jour permet à la base locale de connaÃőtre la liste des types d'action de l'ONG.

3.22.3 Utilité

Cette vue est utile car elle permet à la base locale de connaÃőtre le code d'un type d'action avant une insertion afin de respecter les contraintes d'intégrité.

3.23 Vue matérialisée vue_mat_materiel

3.23.1 Instruction SQL

```
create materialized view vue_mat_materiel
refresh complete
start with sysdate
next sysdate+1
```

```
as
select * from central.materiel@base_centrale;
```

3.23.2 Description

Cette vue, réactualisée chaque jour permet à la base locale de connaÃőtre la liste des matériels de l'ONG.

3.23.3 Utilité

Cette vue est utile car elle permet à la base locale de connaÃőtre le code d'un matériel avant une insertion afin de respecter les contraintes d'intégrité.

3.24 Vue matérialisée vue_mat_annexe

3.24.1 Instruction SQL

```
create materialized view vue_mat_annexe
refresh complete
start with sysdate
next sysdate+1
as
select * from central.annexe@base_centrale;
```

3.24.2 Description

Cette vue, réactualisée chaque jour permet à la base locale de connaÃőtre la liste des bases locales de l'ONG.

3.24.3 Utilité

Cette vue est utile car elle permet à la base locale de connaÃőtre le code d'une annexe avant une insertion afin de respecter les contraintes d'intégrité.

4 Des requêtes distribuées

4.1 Requête PL/SQL de la base centrale

```
INSERT INTO Personnel VALUES ('', 'DOE', 'Jenna', 'Canberra', 'AUS', NULL)
            *
ERROR at line 1:
ORA-20001: Insertion impossible dans la base centrale d un benevole
ORA-06512: at "CENTRAL.PERSONNEL_BASE", line 30
ORA-04088: error during execution of trigger 'CENTRAL.PERSONNEL_BASE'

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

insert into Personnel values ( '', 'Ramos', 'Maria', '51 rua da picante - Madrid', 'ESP', 325.15)
            *
ERROR at line 1:
ORA-20001: L identifiant du pays n est pas correct
ORA-06512: at "CENTRAL.PERSONNEL_BASE", line 31
ORA-04088: error during execution of trigger 'CENTRAL.PERSONNEL_BASE'

==> Pays non prÃľsent dans la table pays (cf. ci-dessous)

   SQL> select * from pays;

   ID_ NOM
   --- ----------------------------------------
   QAT Qatar
   AFS Afrique du Sud
   RUS Russie
   MAL Mali
   RDC Republique democratique du Congo
   KHM Cambodge
   SDN Soudan
   SEN Senegal
   KEN Kenya
   ETH Ethiopie
   IND Inde
   BFA Burkina-Faso
   LKA Sri-Lanka
   VUT Vanuatu
   SCG Serbie-et-Montenegro
   AUS Australie
   FRA France
   ANG Angola
   BUL Bulgarie
   DJI Djibouti
   KUW Kowe?t

   21 rows selected.

--------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------

insert into Annexe values ( '', 'Africa aide', 'Abidjan', 'CIV' )
*
ERROR at line 1:
ORA-02291: integrity constraint (CENTRAL.FK_ANNEXE_1) violated - parent key not
found

==> Pays non prÃľsent dans la table pays (cf. ci-dessus)

-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------

insert into Affectation values ( 5, 12, to_date('01/09/2005','dd/mm/yyyy'),
to_date('26/01/2006','dd/mm/yyyy') )
            *
ERROR at line 1:
ORA-20001:  Les dates ne sont pas conforme aux dates de l action.
ORA-06512: at "CENTRAL.TRIG_AFFECTATION", line 68
ORA-04088: error during execution of trigger 'CENTRAL.TRIG_AFFECTATION'

   ==> La date de dÃľbut de l action est 01/JAN/06 alors que l insertion
veut affecter un personnel Ãă partir du 01/SEP/05

   SQL> select * from action_totale;

   ID_ACTION  CATEGORIE
   ---------- ----------
   DESCRIPTIF                                          ID_SITE
   --------------------------------------------------- ----------
   DATE_DEBU  DATE_FIN_  DATE_FIN_
   ---------  ---------  ---------
   1          1
   Reconstruction hopital X suite guerilla           1
   12-DEC-04  08-SEP-05  08-SEP-05

   2          3
   Seule aire de jeux de la ville                    1
   13-NOV-05  06-MAR-06

   3          2
   Modernisation du reseau telephonique existant     1
   01-JUN-05  12-JUN-09

   4          5
   Creer dans cette zone un dispensaire provisoire   3
   10-JAN-05  11-JAN-07

   5          4
   Apporter a cette zone une premiere ecole          2
   16-DEC-02  01-JAN-05  23-JUL-05
```

```
6          8
Aider les habitants a cultiver                    3
22-NOV-03  22-JUN-04  01-JUN-04

7          9
Apprendre ? lire et ? ?crire aux enfants          1
09-SEP-04  12-JUL-05  30-JUN-05

8         11
Apporter de la nourriture aux habitants           3
20-MAR-98  22-APR-98  20-APR-98

9          6
Aider les habitants a retrouver un travail        2
22-SEP-00  12-DEC-00  12-DEC-00

10         15
Vaccination contre la rage                         1
25-JAN-06  25-MAR-06

11          4
Construction ?l?mentaire, 4 salles                  2
12-DEC-05  01-APR-06

12         16
Sauver le monde                                     2
01-JAN-06  31-DEC-24

13         15
Aide aux enfants malades                            3
15-FEB-06  12-FEB-10

14         13
Sauver la baleine bleue                             3
02-AUG-92  05-JAN-93  12-JAN-93

15         14
Construction d abris aux r?fugi?s de guerre         2
01-MAR-91  01-MAR-92  12-MAY-92

15 rows selected.
```

4.2 Requête PL/SQL de la base locale

```
liste des tables, des vues, des synonymes prÃľsents sur la base locale :

SQL> select * from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
SITE                           TABLE
VUE_MAT_PAYS                   TABLE
VUE_MAT_TYPE_ACTION            TABLE
VUE_MAT_MATERIEL               TABLE
VUE_MAT_ANNEXE                 TABLE
ID_ANNEXE                      TABLE
PERSONNEL                      TABLE
ACTION                         TABLE
MATERIEL_UTILISE               TABLE
AFFECTATION                    TABLE
MATERIEL_UTILISE_TOTALE        SYNONYM
ACTION_TOTALE                  SYNONYM
SITE_TOTALE                    SYNONYM
AFFECTATION_TOTALE             SYNONYM
PERSONNEL_TOTALE               SYNONYM

-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------

liste des pays de l ONG :

SQL> select * from vue_mat_pays;

ID_ NOM
--- ----------------------------------------
QAT Qatar
AFS Afrique du Sud
RUS Russie
MAL Mali
RDC Republique democratique du Congo
KHM Cambodge
SDN Soudan
SEN Senegal
KEN Kenya
ETH Ethiopie
IND Inde
BFA Burkina-Faso
LKA Sri-Lanka
VUT Vanuatu
SCG Serbie-et-Montenegro
AUS Australie
FRA France
ANG Angola
BUL Bulgarie
```

```
DJI Djibouti
KUW Kowe?t

21 rows selected.

-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------

liste des employÃľs de l ONG :

SQL> select * from personnel_totale;

 MATRICULE NOM                  PRENOM
---------- -------------------- ------------------------------
ADRESSE    NAT    SALAIRE
---------------------------------------- --- ----------
         1 Lemer                Robert
Nantes                                   FRA       1213
         2 Boubou               Mary
Diapomo                                  MAL       1020
         3 Travers              Babacar
18 Khartoum Street - Addis Abeba         ETH    1256.48
         4 Nathalie             Bazire
House n?51, Street n?15 - Nyala          SDN    1842.36
         5 Mamadou              Seck
95 rue Carno - Podort                    SEN     526.38
         6 Adama                Coulibally
130 Ushirika Road - Mombasa              KEN     478.52
         7 NONYME               Alex
Le Havre                                 FRA    1000.58
         8 NONYME               Yves
Rennes                                   FRA     785.64
         9 DOE                  John
Canberra                                 AUS    1540.25
        10 Dupond               Jacques
35 rue des verts pr?s - Paris            FRA        152
        11 Ivanov               Vladimir
Moscou                                   RUS
        12 Al Bijour            Mohamed
Doha                                     QAT
        13 Diahousa             Djibril
Picouli                                  RDC
        14 Petit                Fabrice
29 rue Blaise Diagne - Kaolack           SEN
        15 Salif                Keita
15 Harambee Avenue - Nairobi             KEN
        16 DOE                  Jenna
Canberra                                 AUS
        17 Boutros              Pierre
87 av de gaulle - Toulouse               FRA

17 rows selected.
```

```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

liste des employÃľs benevoles de la base locale :

SQL> select * from personnel;

 MATRICULE NOM                  PRENOM
---------- -------------------- ------------------------------
ADRESSE                                  NAT    SALAIRE
---------------------------------------- --- ----------
        11 Ivanov               Vladimir
Moscou                                   RUS
        12 Al Bijour            Mohamed
Doha                                     QAT
        13 Diahousa             Djibril
Picouli                                  RDC
        14 Petit                Fabrice
29 rue Blaise Diagne - Kaolack           SEN
        15 Salif                Keita
15 Harambee Avenue - Nairobi             KEN
        16 DOE                  Jenna
Canberra                                 AUS
        17 Boutros              Pierre
87 av de gaulle - Toulouse               FRA

7 rows selected.

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

Liste des affectations des bÃľnÃľvoles de la base locale

SQL> select * from affectation;

 MATRICULE  ID_ACTION DATE_DEBU DATE_FIN
---------- ---------- --------- ---------
        12          1 25-JAN-06 25-FEB-06
        11          2 20-DEC-05 25-FEB-06

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

Liste des affectations de l ONG

SQL> select * from affectation_totale;

 MATRICULE  ID_ACTION DATE_DEBU DATE_FIN
---------- ---------- --------- ---------
         1          6 05-MAY-06 05-JUL-06
         3          9 20-DEC-05 01-JAN-06
```

```
         9         10 15-FEB-06 01-JAN-07
        11          2 20-DEC-05 25-FEB-06
        12          1 25-JAN-06 25-FEB-06

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

insert into Personnel values ('', 'Lemer', 'Robert', 'Nantes', 'FRA', 1213)
            *
ERROR at line 1:
ORA-20001: Insertion impossible dans une base locale d un salari?
ORA-06512: at "ANNEXE.PERSONNEL_LOCAL", line 21
ORA-04088: error during execution of trigger 'ANNEXE.PERSONNEL_LOCAL'

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

insert into Personnel values ( '', 'Smith', 'Alexander',
'2 av Piccadilus - Angleterre', 'GBR', null )
            *
ERROR at line 1:
ORA-20001: L identifiant du pays n est pas correct
ORA-06512: at "ANNEXE.PERSONNEL_LOCAL", line 22
ORA-04088: error during execution of trigger 'ANNEXE.PERSONNEL_LOCAL'

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

insert into Materiel_utilise values (11, 10, 2)
            *
ERROR at line 1:
ORA-20001: Le site entre n est pas valable
ORA-06512: at "ANNEXE.MAT_UTIL_LOCAL", line 47
ORA-04088: error during execution of trigger 'ANNEXE.MAT_UTIL_LOCAL'

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

INSERT INTO Materiel_utilise VALUES (2, 3, 200)
            *
ERROR at line 1:
ORA-20001: La quantite entree est trop importante
ORA-06512: at "ANNEXE.MAT_UTIL_LOCAL", line 49
ORA-04088: error during execution of trigger 'ANNEXE.MAT_UTIL_LOCAL'

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

insert into Affectation values (10, 3, to_date ( '12/12/2004','dd/mm/yyyy'),
```

```
to_date ('12/01/2005','dd/mm/yyyy'))
            *
ERROR at line 1:
ORA-20001:  Les dates ne sont pas conforme aux dates de l action.
ORA-06512: at "ANNEXE.TRIG_AFFECTATION", line 74
ORA-04088: error during execution of trigger 'ANNEXE.TRIG_AFFECTATION'

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

INSERT INTO Affectation VALUES (6, 1, to_date ('26/03/2005','dd/mm/yyyy'),
to_date('23/12/2006','dd/mm/yyyy'))
            *
ERROR at line 1:
ORA-20001:  L insertion doit etre faite sur la base centrale
ORA-06512: at "ANNEXE.TRIG_AFFECTATION", line 77
ORA-04088: error during execution of trigger 'ANNEXE.TRIG_AFFECTATION'

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

INSERT INTO Affectation VALUES (12, 1, TO_DATE('25/02/2006','dd/mm/yyyy'),
TO_DATE('12/03/2006','dd/mm/yyyy'))
*
ERROR at line 1:
ORA-00001: unique constraint (ANNEXE.SYS_C007423) violated
```

5 Annexes

5.1 Script PL/SQL de la base centrale

```
-- grant connect,resource to annexe identified by annexe;

DROP PUBLIC DATABASE LINK annexe_1;
CREATE PUBLIC DATABASE LINK annexe_1 CONNECT TO central IDENTIFIED BY central using 'DB_JEROME';

CREATE OR REPLACE PROCEDURE insert_into_table (
      dans  VARCHAR2,
      val    VARCHAR2)
      IS
   stmt_str    VARCHAR2(200);

BEGIN
   stmt_str := 'INSERT INTO ' || dans || ' values
      ( '|| val || ')';

   EXECUTE IMMEDIATE stmt_str ;
END;
/

CREATE OR REPLACE PROCEDURE update_table (
      dans  VARCHAR2,
      col VARCHAR2,
      val    VARCHAR2,
      id_name VARCHAR2,
      id VARCHAR2
      )
      IS
   stmt_str    VARCHAR2(200);

BEGIN
   stmt_str := 'UPDATE ' || dans || ' set ' || col || ' = '
|| val || ' WHERE ' || id_name || ' = ' || id ;

   EXECUTE IMMEDIATE stmt_str ;
END;
/

DROP TABLE Affectation;
DROP TABLE Personnel;
DROP TABLE Annexe;
DROP TABLE Pays;
DROP TABLE Materiel;
DROP TABLE Type_action;

DROP SEQUENCE seq_type_action;
DROP SEQUENCE seq_materiel;
DROP SEQUENCE seq_personnel;
DROP SEQUENCE seq_annexe;
DROP SEQUENCE seq_action;
```

```
DROP SEQUENCE seq_site;

drop view personnel_totale;
drop view action_totale;
drop view site_totale;
drop view materiel_utilise_totale;
drop view affectation_totale;


--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

CREATE TABLE Type_action (
       ID_type_action INTEGER NOT NULL
     , Nom VARCHAR(50) NOT NULL
     , PRIMARY KEY (ID_type_action)
);

CREATE SEQUENCE seq_type_action
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;

CREATE OR REPLACE TRIGGER trig_cle_prim_type_action
BEFORE INSERT ON Type_action
FOR EACH ROW
BEGIN
SELECT seq_type_action.NEXTVAL INTO :NEW.ID_type_action from DUAL;
END;
/

--------------------------------------------------------------------------------

CREATE TABLE Materiel (
       ID_mat INTEGER NOT NULL
     , Descriptif VARCHAR(100) NOT NULL
     , Quantite_dispo INTEGER NOT NULL
     , PRIMARY KEY (ID_mat)
);

CREATE SEQUENCE seq_materiel
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;

CREATE OR REPLACE TRIGGER trig_cle_prim_materiel
BEFORE INSERT ON Materiel
FOR EACH ROW
BEGIN
SELECT seq_materiel.NEXTVAL INTO :NEW.ID_mat from DUAL;
END;
```

```
/

-------------------------------------------------------------------------------

CREATE TABLE Pays (
       ID_pays CHAR(3) NOT NULL
     , NOM VARCHAR(40) NOT NULL
     , PRIMARY KEY (ID_pays)
);

-------------------------------------------------------------------------------

CREATE TABLE Annexe (
       ID_annexe INTEGER NOT NULL
     , Nom VARCHAR(20) NOT NULL
     , Adresse VARCHAR(50) NOT NULL
     , ID_pays CHAR(3) NOT NULL
     , PRIMARY KEY (ID_annexe)
     , CONSTRAINT FK_Annexe_1 FOREIGN KEY (ID_pays)
                  REFERENCES Pays (ID_pays)
);

CREATE SEQUENCE seq_annexe
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;


CREATE OR REPLACE TRIGGER trig_cle_prim_annexe
BEFORE INSERT ON Annexe
FOR EACH ROW
BEGIN
SELECT seq_annexe.NEXTVAL INTO :NEW.ID_annexe from DUAL;
END;
/

-------------------------------------------------------------------------------


CREATE TABLE Personnel (
       Matricule INTEGER NOT NULL
     , Nom VARCHAR(20) NOT NULL
     , Prenom VARCHAR(30) NOT NULL
     , Adresse VARCHAR(40) NOT NULL
     , Nationalite CHAR(3) NOT NULL
     , Salaire NUMERIC(7,2)
     , PRIMARY KEY (Matricule)
     , CONSTRAINT FK_Personnels_1 FOREIGN KEY (Nationalite)
                  REFERENCES Pays (ID_pays)
);
```

```
CREATE SEQUENCE seq_personnel
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;

create view personnel_totale as select * from central.Personnel
UNION select * from annexe.personnel@annexe_1;

create or replace trigger personnel_base
before insert on personnel
for each row
DECLARE
cursor c1 IS select * from annexe;
cursor c_pays IS select * from pays;
reponse varchar(2);
ck_pays INTEGER;

erreur_insertion EXCEPTION;
mauvais_pays EXCEPTION;
insertion_migree EXCEPTION;
BEGIN
if(:new.Salaire is NULL) THEN
raise erreur_insertion;
ELSE
ck_pays := 0;
FOR cp IN c_pays LOOP
IF(:new.Nationalite = cp.ID_pays) THEN
ck_pays := 1;
END IF;
END LOOP;

if (ck_pays = 0) THEN raise mauvais_pays;
-- sinon, l'insertion se poursuit normalement
END IF;
-- A ce niveau, l'insertion est valide
-- on ajoute le numÃľro de sÃľquence
select seq_personnel.nextval into :new.Matricule from Dual;
END IF;

EXCEPTION
when erreur_insertion THEN raise_application_error
(-20001, 'Insertion impossible dans la base centrale d un benevole');
when mauvais_pays THEN raise_application_error
(-20001, 'L identifiant du pays n est pas correct');
when insertion_migree THEN raise_application_error
(-20001, 'L insertion a ete deplacee vers une base annexe');
END;
/

--------------------------------------------------------------------------------

create view site_totale as select * from annexe.site@annexe_1;
```

```
create view action_totale as select * from annexe.Action@annexe_1;

--------------------------------------------------------------------------------



CREATE TABLE Affectation (
       Matricule INTEGER NOT NULL
     , ID_action INTEGER NOT NULL
     , Date_debut DATE NOT NULL
     , Date_fin DATE NOT NULL
     , PRIMARY KEY (Matricule, ID_action)
     , CONSTRAINT FK_Affectation_1 FOREIGN KEY (Matricule)
                  REFERENCES Personnel (Matricule)
);

CREATE VIEW affectation_totale AS SELECT * from affectation UNION
SELECT * from annexe.affectation@annexe_1;

CREATE OR REPLACE TRIGGER trig_affectation
BEFORE INSERT ON affectation
FOR EACH ROW
DECLARE
date_erronee EXCEPTION;
deja_aff EXCEPTION;
insert_interdit EXCEPTION;
mauvaise_annexe EXCEPTION;
pb_reference EXCEPTION;
insertion_migree EXCEPTION;

dateDebAct action_totale.date_debut%TYPE;
dateFinAct action_totale.date_fin_prevue%TYPE;

CURSOR c1 IS SELECT date_debut, date_fin
FROM affectation_totale WHERE matricule = :NEW.matricule;
dateDebAff action_totale.date_debut%TYPE;
dateFinAff action_totale.date_fin_prevue%TYPE;

sal personnel.salaire%TYPE;
mat personnel.Matricule%TYPE;
id_an annexe.id_annexe%TYPE;
BEGIN
SELECT date_debut INTO dateDebAct FROM action_totale WHERE ID_action=:NEW.ID_action;
SELECT Matricule INTO mat FROM personnel_totale WHERE matricule = :NEW.matricule;

IF(dateDebAct is null or mat is null) THEN
raise pb_reference;
END IF;

/* VÃľrifier que la date de dÃľbut est chronologiquement infÃľrireure Ãă la date de fin */
IF (:NEW.date_debut > :NEW.date_fin OR :NEW.date_debut = :NEW.date_fin) THEN
RAISE date_erronee ;
END IF;
```

```
/* Si une affectation concernant une action dÃľbute avant celle-ci  */
IF (dateDebAct > :NEW.date_debut) THEN
RAISE date_erronee ;
END IF;

/* Si une affectation concernant une action se termine aprÃís celle-ci
 * c'est possible (action est prolongÃľe) mais demander confirmation de l'utilisateur
 */
IF (dateFinAct < :NEW.date_fin) THEN

dbms_output.put_line('ATTENTION: L affectation nouvellement
entrÃľe se termine aprÃís la date de fin prÃľvue pour l action ');

END IF;

/* VÃľrifier que l'utilisateur n'est pas dÃľja affÃľctÃľ
durant la pÃľriode de l'affectation candidate Ãă l'insertion */
OPEN c1;
LOOP
FETCH c1 INTO dateDebAff, dateFinAff;
EXIT WHEN c1%NOTFOUND;

IF ((dateDebAff < :NEW.date_fin AND dateFinAff > :NEW.date_fin) OR (dateFinAff >
:NEW.date_debut AND dateFinAff < :NEW.date_fin)) THEN RAISE deja_aff ;
END IF;
END LOOP;
CLOSE c1;

/* DÃľterminer si l'insertion doit Ãłtre effÃľctuÃľe en base centrale (salaire != NULL)
ou en base locale (salaire = NULL) */
SELECT salaire INTO sal FROM personnel_totale WHERE matricule = :NEW.matricule;

IF (sal is null) THEN
SELECT id_annexe INTO id_an from site_totale where id_site =
(SELECT id_site FROM action_totale WHERE ID_action = :NEW.ID_action);

insert_into_table('annexe.affectation@annexe_'||id_an, :NEW.matricule||' ,
'||:NEW.ID_action||', '||:NEW.date_debut||' , '||:NEW.date_fin);
raise insertion_migree;
END IF;

EXCEPTION
WHEN date_erronee THEN raise_application_error
(-20001, ' Les dates ne sont pas conforme aux dates de l action. ');
WHEN deja_aff THEN raise_application_error
(-20001, ' Cette personne est dÃľjÃă affectÃľe durant cette pÃľriode. ');
WHEN pb_reference THEN raise_application_error
(-20001, ' Un tuple reference n existe pas ');
WHEN insertion_migree THEN raise_application_error
(-20001, ' L insertion a ete deplacee sur une base annexe ');
END;
/

--------------------------------------------------------------------------------
```

```
CREATE SEQUENCE seq_action
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;

create or replace trigger trig_action_centrale
INSTEAD OF insert on action_totale
for each row
declare
pb_reference EXCEPTION;

categ Type_action.ID_type_action%TYPE;
num_site site_totale.ID_site%TYPE;
num_annexe_insert site_totale.ID_annexe%TYPE;
id_action INTEGER;
begin
select ID_type_action into categ from Type_action where ID_type_action = :new.Categorie;
select ID_site into num_site from site_totale where ID_site = :new.ID_site;

if ( categ is null or num_site is null ) then raise pb_reference;
end if;

select ID_annexe into num_annexe_insert from site_totale where ID_site = :new.ID_site;

if ( num_annexe_insert is not null ) then
select seq_action.nextval into id_action from dual;
insert_into_table('annexe.Action@annexe_'||num_annexe_insert, id_action ||' ,
 '||:new.Categorie ||' , '||:new.Descriptif||' , '||:new.ID_site||' , '||:new.Date_debut||', '||:
else raise pb_reference;
end if;

EXCEPTION
when pb_reference THEN raise_application_error
( -20001, ' Votre insertion rÃ'fÃ'rence un tuple inexistant ' );
end;
/


--------------------------------------------------------------------------------


CREATE SEQUENCE seq_site
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOCYCLE;

create or replace trigger trig_site_centrale
INSTEAD OF insert on site_totale
for each row
declare
pb_reference EXCEPTION;
```

```
num_pays Pays.ID_pays%TYPE;
num_annexe Annexe.ID_annexe%TYPE;
id_nv_site INTEGER;
begin
select ID_pays into num_pays from Pays where ID_pays = :new.ID_pays;
select ID_annexe into num_annexe from Annexe where ID_annexe = :new.ID_annexe;

if ( num_pays is null or num_annexe is null ) then raise pb_reference;
end if;

select seq_site.nextval into id_nv_site from dual;
insert_into_table( 'annexe.Site@annexe_'||num_annexe, id_nv_site||'
, '||:new.Nom||' , '||:new.latitude||' , '||:new.longitude||' , '||:new.ID_pays||' ,
'||:new.ID_annexe);

EXCEPTION
when pb_reference THEN raise_application_error
( -20001, ' Votre insertion rÃľfÃľrence un tuple inexistant ' );
end;
/

-------------------------------------------------------------------------------------

create view materiel_utilise_totale as select * from annexe.Materiel_utilise@annexe_1;

create or replace trigger trig_materiel_utilise_centrale
INSTEAD OF insert on materiel_utilise_totale
for each row
declare
pb_reference EXCEPTION;

num_site site_totale.ID_site%TYPE;
num_annexe_insert site_totale.ID_annexe%TYPE;
begin
select ID_site into num_site from site_totale where ID_site = :new.ID_site;

if ( num_site is null ) then raise pb_reference;
end if;


select ID_annexe into num_annexe_insert from site_totale where ID_site = :new.ID_site;

if ( num_annexe_insert is not null ) then
insert_into_table('annexe.Materiel_utilise@annexe_'||num_annexe_insert, :new.ID_site||' ,
 '||:new.ID_mat||' , '||:new.quantite);
else raise pb_reference;
end if;

EXCEPTION
when pb_reference THEN raise_application_error
( -20001, ' Votre insertion rÃľfÃľrence un tuple inexistant ' );
end;
/
```

```
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------

grant select on central.materiel_utilise_totale to annexe;
grant select on central.personnel_totale to annexe;
grant select on central.action_totale to annexe;
grant select on central.site_totale to annexe;
grant select on central.affectation_totale to annexe;

grant select on central.pays to annexe;
grant select on central.type_action to annexe;
grant select on central.annexe to annexe;
grant select on central.materiel to annexe;

grant select on central.seq_personnel to annexe;
grant select on central.seq_site to annexe;
grant select on central.seq_action to annexe;
grant select on central.seq_annexe to annexe;
grant select on central.seq_materiel to annexe;
grant select on central.seq_type_action to annexe;

grant update on central.materiel to annexe;

-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------

insert into Type_action values ('', 'Construire Hopital');
insert into Type_action values ('', 'Installation Telephone');
insert into Type_action values ('', 'Creation aire de jeux');
INSERT INTO Type_action VALUES ('', 'Construction d une ecole');
INSERT INTO Type_action VALUES ('', 'Construction d un dispensaire');
INSERT INTO Type_action VALUES ('', 'RÃ©insertion professionnelle');
INSERT INTO Type_action VALUES ('', 'Repas social');
INSERT INTO Type_action VALUES ('', 'Culture des sols');
INSERT INTO Type_action VALUES ('', 'Enseignement');
INSERT INTO Type_action VALUES ('', 'Campagne de vaccination');
INSERT INTO Type_action VALUES ('', 'Distribution alimentaire');
INSERT INTO Type_action VALUES ('', 'Construction collÃ©ge');
insert into Type_action values ( '', 'Sauvegarde des animaux' );
insert into Type_action values ( '', 'Construction de prefabriques' );
insert into Type_action values ( '', 'SantÃ© publique' );
insert into Type_action values ( '', 'ConfÃ©rence mondial' );


insert into Materiel values ('', 'Conteneur', 10);
insert into Materiel values ('', 'Stylo', 1000);
```

```
insert into Materiel values ('', 'Formulaire A23 vierge', 30);
INSERT INTO Materiel VALUES ('', 'Ordinateur', 2000);
INSERT INTO Materiel VALUES ('', 'Table', 1000);
INSERT INTO Materiel VALUES ('', 'Chaise', 4000);
INSERT INTO Materiel VALUES ('', 'Lit de bÃľbÃľ', 2500);
INSERT INTO Materiel VALUES ('', 'Lot de couverts', 800);
INSERT INTO Materiel VALUES ('', 'Trousse de secours', 1000);
INSERT INTO Materiel VALUES ('', 'Seringues intra-musculaire', 5000);
INSERT INTO Materiel VALUES ('', 'Livre mathÃľmatique classe ÃľlÃľmentaire', 200);
INSERT INTO Materiel VALUES ('', 'Paire de gants sanitaires', 10000);
INSERT INTO Materiel VALUES ('', 'Lit de camps', 356);
insert into Materiel values ( '', 'Tente', 50 );
insert into Materiel values ( '', 'Couverture', 150 );
insert into Materiel values ( '', 'Sac de riz', 25000 );
insert into Materiel values ( '', 'PrÃľservatif', 15200 );


insert into Pays values ('QAT', 'Qatar');
insert into Pays values ('AFS', 'Afrique du Sud');
insert into Pays values ('RUS', 'Russie');
insert into Pays values ('MAL', 'Mali');
insert into Pays values ('RDC', 'Republique democratique du Congo');
INSERT INTO Pays VALUES ('KHM', 'Cambodge');
INSERT INTO Pays VALUES ('SDN', 'Soudan');
INSERT INTO Pays VALUES ('SEN', 'Senegal');
INSERT INTO Pays VALUES ('KEN', 'Kenya');
INSERT INTO Pays VALUES ('ETH', 'Ethiopie');
INSERT INTO Pays VALUES ('IND', 'Inde');
INSERT INTO Pays VALUES ('BFA', 'Burkina-Faso');
INSERT INTO Pays VALUES ('LKA', 'Sri-Lanka');
INSERT INTO Pays VALUES ('VUT', 'Vanuatu');
INSERT INTO Pays VALUES ('SCG', 'Serbie-et-Montenegro');
INSERT INTO Pays VALUES ('AUS', 'Australie');
INSERT INTO Pays VALUES ('FRA', 'France');
insert into Pays values ( 'ANG', 'Angola' );
insert into Pays values ( 'BUL', 'Bulgarie' );
insert into Pays values ( 'DJI', 'Djibouti' );
insert into Pays values ( 'KUW', 'KoweÃŕt' );


insert into Annexe values ('', 'Ctr L. Boyer', 'Bamako', 'MAL');
insert into Annexe values ('', 'Le soleil', 'Kinshasa', 'RDC');
insert into Annexe values ('', 'Hamad bin Khalifa', 'Doha', 'QAT');
INSERT INTO Annexe VALUES ('', 'Soutien Cambodge',
80 Boulevard Preah Norodom - Phnom Penh', 'KHM');
INSERT INTO Annexe VALUES ('', 'Aide Cambodge',
'rue 184 - Siem Reap', 'KHM');
INSERT INTO Annexe VALUES ('', 'Senegal Assistance',
'Route de Khombole - Thies', 'SEN');
INSERT INTO Annexe VALUES ('', 'Ethiop aide',
'13 Taitu Street - Addis Abeba', 'ETH');
INSERT INTO Annexe VALUES ('', 'Senegal Solidarite',
'43 rue Jules Ferry - Dakar', 'SEN');
INSERT INTO Annexe VALUES ('', 'Collectif Inde',
```

```
'20 Chittaranjan Road - Madras', 'IND');
INSERT INTO Annexe VALUES ('', 'Annexe BF', 'Ouagadougou', 'BFA');
INSERT INTO Annexe VALUES ('', 'Annexe SL', 'Colombo', 'LKA');
INSERT INTO Annexe VALUES ('', 'Annexe SB', 'Belgrade', 'SCG');
INSERT INTO Annexe VALUES ('', 'Annexe AU', 'Sydney', 'AUS');
insert into Annexe values ( '', 'Africa aide', 'Abidjan', 'CIV' );
insert into Annexe values ( '', 'Latina soutien', 'amazonie', 'FRA' );
insert into Annexe values ( '', 'Annexe France', 'Rouen', 'FRA' );
insert into Annexe values ( '', 'Alaska perdu', 'route du grand nord', 'USA' );


insert into Personnel values ('', 'Lemer', 'Robert',
'Nantes', 'FRA', 1213);
insert into Personnel values ('', 'Ivanov', 'Vladimir',
'Moscou', 'RUS', null);
insert into Personnel values ('', 'Boubou', 'Mary', 'Diapomo',
'MAL', 1020);
insert into Personnel values ('', 'Al Bijour', 'Mohamed', 'Doha',
'QAT', null);
insert into Personnel values ('', 'Diahousa', 'Djibril', 'Picouli',
'RDC', null);
INSERT INTO Personnel VALUES ('', 'Petit','Fabrice',
'29 rue Blaise Diagne - Kaolack', 'SEN', null);
INSERT INTO Personnel VALUES ('', 'Travers', 'Babacar', '18 Khartoum Street - Addis Abeba',
'ETH', 1256.48);
INSERT INTO Personnel VALUES ('', 'Nathalie', 'Bazire', 'House nÂř51,
Street nÂř15 - Nyala', 'SDN', 1842.36);
INSERT INTO Personnel VALUES ('', 'Salif', 'Keita', '15 Harambee Avenue - Nairobi',
'KEN', null);
INSERT INTO Personnel VALUES ('', 'Mamadou', 'Seck', '95 rue Carno - Podort',
'SEN', 526.38);
INSERT INTO Personnel VALUES ('', 'Adama', 'Coulibally', '130 Ushirika Road - Mombasa',
'KEN', 478.52);
INSERT INTO Personnel VALUES ('', 'NONYME', 'Alex', 'Le Havre', 'FRA', 1000.58);
INSERT INTO Personnel VALUES ('', 'NONYME', 'Yves', 'Rennes', 'FRA', 785.64);
INSERT INTO Personnel VALUES ('', 'DOE', 'John', 'Canberra', 'AUS', 1540.25);
INSERT INTO Personnel VALUES ('', 'DOE', 'Jenna', 'Canberra', 'AUS', NULL);
insert into Personnel values ( '', 'Dupond', 'Jacques', '35 rue des verts prÃľs - Paris',
'FRA', 152 );
insert into Personnel values ( '', 'Ramos', 'Maria', '51 rua da picante - Madrid',
'ESP', 325.15);
insert into Personnel values ( '', 'Smith', 'Alexander', '2 av Piccadilus - Angleterre',
'GBR', null );
insert into Personnel values ( '', 'Boutros', 'Pierre', '87 av de gaulle - Toulouse',
'FRA', null );

insert into Affectation values ( 5, 12, to_date('01/09/2005','dd/mm/yyyy'),
to_date('26/01/2006','dd/mm/yyyy') );
insert into Affectation values ( 9, 10, to_date('15/02/2006','dd/mm/yyyy'),
to_date('01/01/2007','dd/mm/yyyy') );
insert into Affectation values ( 3, 9, to_date('20/12/2005','dd/mm/yyyy'),
to_date('01/01/2006','dd/mm/yyyy') );
insert into Affectation values ( 1, 6, to_date('05/05/2006','dd/mm/yyyy'),
to_date('05/07/2006','dd/mm/yyyy') );
```

```
commit;
```

5.2 Script PL/SQL de la base locale

```
DROP PUBLIC DATABASE LINK base_centrale;
CREATE PUBLIC DATABASE LINK base_centrale CONNECT TO annexe
IDENTIFIED BY annexe using 'DB_SEB';


DROP TABLE Affectation;
DROP TABLE Materiel_utilise;
DROP TABLE Action;
DROP TABLE Site;
DROP TABLE Personnel;
DROP TABLE Id_annexe;

DROP MATERIALIZED VIEW vue_mat_pays;
DROP MATERIALIZED VIEW vue_mat_type_action;
DROP MATERIALIZED VIEW vue_mat_materiel;
DROP MATERIALIZED VIEW vue_mat_annexe;

drop SYNONYM personnel_totale;
drop SYNONYM site_totale;
drop SYNONYM action_totale;
drop SYNONYM materiel_utilise_totale;
drop SYNONYM affectation_totale;

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------


create materialized view vue_mat_pays
refresh complete
start with sysdate
next sysdate+1
as
select * from central.pays@base_centrale;


create materialized view vue_mat_type_action
refresh complete
start with sysdate
next sysdate+1
as
select * from central.type_action@base_centrale;


create materialized view vue_mat_materiel
refresh complete
start with sysdate
next sysdate+1
as
select * from central.materiel@base_centrale;
```

```
create materialized view vue_mat_annexe
refresh complete
start with sysdate
next sysdate+1
as
select * from central.annexe@base_centrale;



--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------


CREATE TABLE Id_annexe (
       id INTEGER NOT NULL
     , PRIMARY KEY (id)
);

--------------------------------------------------------------------------------

-- CrÃľation d'un alias pour accÃľder Ãă la vue globale

CREATE SYNONYM personnel_totale FOR central.personnel_totale@base_centrale;

CREATE TABLE Personnel (
       Matricule INTEGER NOT NULL
     , Nom VARCHAR(20) NOT NULL
     , Prenom VARCHAR(30) NOT NULL
     , Adresse VARCHAR(40) NOT NULL
     , Nationalite CHAR(3) NOT NULL
     , Salaire NUMERIC(7,2)
     , PRIMARY KEY (Matricule)
);


create or replace trigger personnel_local
before insert on personnel
for each row
DECLARE
cursor c_pays IS select * from central.pays@base_centrale;
ck_pays INTEGER;
insert_interdit EXCEPTION;
mauvais_pays EXCEPTION;
BEGIN
if(:new.Salaire is not NULL) THEN
raise insert_interdit ;
ELSE
select count(*) into ck_pays from vue_mat_pays where
ID_pays = :new.Nationalite;

if (ck_pays = 0 ) THEN raise mauvais_pays ;
-- sinon, l'insertion se poursuit normalement
```

```
END IF;
-- A ce niveau, l'insertion est valide
-- on ajoute le numÃľro de sÃľquence
select central.seq_personnel.nextval@base_centrale
into :new.Matricule from Dual;
END IF;

EXCEPTION
when insert_interdit THEN raise_application_error
(-20001, 'Insertion impossible dans une base locale d un salariÃľ');
when mauvais_pays THEN raise_application_error
(-20001, 'L identifiant du pays n est pas correct');
END;
/


--------------------------------------------------------------------------------

CREATE SYNONYM site_totale FOR central.site_totale@base_centrale;

CREATE TABLE Site (
       ID_site INTEGER NOT NULL
     , Nom VARCHAR(50) NOT NULL
     , latitude VARCHAR(10)
     , longitude VARCHAR(10)
     , ID_pays CHAR(3) NOT NULL
     , ID_annexe INTEGER NOT NULL
     , PRIMARY KEY (ID_site)
);

create or replace trigger site_local
before insert on site
FOR EACH ROW
DECLARE
ck_pays INTEGER;
num_annexe ID_annexe.id%TYPE;
mauvais_pays EXCEPTION;
mauvaise_annexe EXCEPTION;
BEGIN
select count(*) into ck_pays from vue_mat_pays
where ID_pays = :new.ID_pays;

IF (ck_pays = 0 ) THEN
raise mauvais_pays ;
ELSE
-- sinon, l'insertion se poursuit
select id into num_annexe from ID_annexe;
IF(:new.ID_annexe != num_annexe) THEN
raise mauvaise_annexe;
ELSE
-- sinon, les contraintes sont vÃľrifiÃľes
select central.seq_site.nextval@base_centrale
into :new.ID_site from dual;
END IF;
```

```
END IF;
EXCEPTION
WHEN mauvaise_annexe THEN raise_application_error
(-20001, 'Le site n est pas rattache a votre annexe');
WHEN mauvais_pays THEN raise_application_error
(-20001, 'Le site n est pas rattache a un pays valide');
END;
/


--------------------------------------------------------------------------------

CREATE SYNONYM action_totale FOR central.action_totale@base_centrale;

CREATE TABLE Action (
       ID_action INTEGER NOT NULL
     , Categorie INTEGER NOT NULL
     , Descriptif VARCHAR(100) NOT NULL
     , ID_site INTEGER NOT NULL
     , Date_debut DATE NOT NULL
     , Date_fin_prevue DATE NOT NULL
     , Date_fin_effective DATE
     , PRIMARY KEY (ID_action)
     , CONSTRAINT FK_Actions_2 FOREIGN KEY (ID_site)
                  REFERENCES Site (ID_site)
);


CREATE OR REPLACE TRIGGER trig_action_local
BEFORE INSERT ON action
FOR EACH ROW
DECLARE
pb_reference EXCEPTION;
mauvaise_annexe EXCEPTION;

id_cat vue_mat_type_action.id_type_action%TYPE;
id_site site.id_site%TYPE;

id_an vue_mat_annexe.id_annexe%TYPE;
id_an_ici vue_mat_annexe.id_annexe%TYPE;

BEGIN
/* Verifiant de l'existence des clÃľes Ãľtrangãíres */
SELECT id_type_action INTO id_cat FROM vue_mat_type_action
 WHERE id_type_action = :NEW.categorie;

SELECT id_site INTO id_site FROM site WHERE id_site
= :NEW.ID_site;

IF (id_cat IS NULL OR id_site IS NULL) THEN
raise pb_reference ;
END IF;

/* On vÃľrifie que l'insertion est bien effectuÃľe
```

```
dans la base de la bonne annexe */
SELECT id_annexe INTO id_an from site_totale where id_site
= :NEW.ID_site;

SELECT id INTO id_an_ici from id_annexe;

IF ( id_an = id_an_ici) THEN
SELECT central.seq_action.nextval@base_centrale
into :new.ID_action from dual;

ELSE raise mauvaise_annexe ;

END IF;

EXCEPTION
WHEN pb_reference THEN raise_application_error
(-20001, ' Votre insertion rÃľfÃľrence un tuple non existant. ');
WHEN mauvaise_annexe THEN raise_application_error
(-20001, ' Veuillez insÃľrer dans la base locale de l annexe correspondant au bÃľnÃľvole ');
END;
/


--------------------------------------------------------------------------------

CREATE SYNONYM materiel_utilise_totale
FOR central.materiel_utilise_totale@base_centrale;

CREATE TABLE Materiel_utilise (
       ID_site INTEGER NOT NULL
     , ID_mat INTEGER NOT NULL
     , quantite INTEGER NOT NULL
     , PRIMARY KEY (ID_site, ID_mat)
     , CONSTRAINT FK_Materiels_sur_site_1 FOREIGN KEY (ID_site)
                 REFERENCES Site (ID_site)
);


create or replace trigger mat_util_local
before insert on materiel_utilise
for each row
DECLARE
un_site site.ID_site%TYPE;
un_materiel vue_mat_materiel.ID_mat%TYPE;
num_annexe ID_annexe.id%TYPE;
annexe_site vue_mat_annexe.ID_annexe%TYPE;
qte_dispo vue_mat_materiel.Quantite_dispo%TYPE;
i INTEGER;
mauvais_site EXCEPTION;
mauvais_materiel EXCEPTION;
erreur_quantite EXCEPTION;
BEGIN
select count(*) into i from site_totale
where ID_site = :new.ID_site;
```

```
IF(i = 1) THEN
-- RÃľfÃľrence bien un site existant
select count(*) into i from vue_mat_materiel
where ID_mat = :new.ID_mat;

IF(i = 1) THEN
-- RÃľfÃľrence bien un matÃľriel existant
select id into num_annexe from ID_annexe;
select a.ID_annexe into annexe_site from vue_mat_annexe a,
site s where a.ID_annexe = s.ID_annexe AND s.ID_site = :new.ID_site;
IF(annexe_site != num_annexe) THEN
-- Le numÃľro de l'annexe local ne
corrsepond pas Ãă celui du site insÃľrÃľ
raise mauvais_site;
ELSE
-- Inutile de faire un autre FETCH, car le dernier doit Ãłtre le bon
-- (1 seul matÃľriel avec la mÃłme clÃľ primaire)
select quantite_dispo into qte_dispo from central.materiel
@base_centrale where ID_mat = :new.ID_mat;
IF(:new.quantite > qte_dispo) THEN
raise erreur_quantite;
-- sinon, insertion valable
ELSE
update central.materiel@base_centrale set
Quantite_dispo = (Quantite_dispo - :new.quantite) where ID_mat = :new.ID_mat;
-- l'insertion se continue normalement
END IF;
END IF;
ELSE
-- Erreur concernant un matÃľriel
raise mauvais_materiel;
END IF;
ELSE
-- Erreur concernant un site
raise mauvais_site;
END IF;

EXCEPTION
when mauvais_site THEN raise_application_error
(-20001, 'Le site entre n est pas valable');
when mauvais_materiel THEN raise_application_error
(-20001, 'Le materiel entre n est pas valable');
when erreur_quantite THEN raise_application_error
(-20001, 'La quantite entree est trop importante');
END;
/


--------------------------------------------------------------------------------

CREATE SYNONYM affectation_totale FOR central.affectation_totale@base_centrale;

CREATE TABLE Affectation (
```

```
       Matricule INTEGER NOT NULL
     , ID_action INTEGER NOT NULL
     , Date_debut DATE NOT NULL
     , Date_fin DATE NOT NULL
     , PRIMARY KEY (Matricule, ID_action)
     , CONSTRAINT FK_Affectation_1 FOREIGN KEY (Matricule)
                  REFERENCES Personnel (Matricule)
     , CONSTRAINT FK_Affectation_2 FOREIGN KEY (ID_action)
                  REFERENCES Action (ID_action)
);


CREATE OR REPLACE TRIGGER trig_affectation
BEFORE INSERT ON affectation
FOR EACH ROW
DECLARE
date_erronee EXCEPTION;
deja_aff EXCEPTION;
insert_interdit EXCEPTION;
mauvaise_annexe EXCEPTION;
pb_reference EXCEPTION;
insertion_a_migrer EXCEPTION;

dateDebAct action_totale.date_debut%TYPE;
dateFinAct action_totale.date_fin_prevue%TYPE;

CURSOR c1 IS SELECT date_debut, date_fin
FROM affectation_totale WHERE matricule = :NEW.matricule;
dateDebAff action_totale.date_debut%TYPE;
dateFinAff action_totale.date_fin_prevue%TYPE;

num_annexe Id_annexe.id%TYPE;
annexe_affectation Id_annexe.id%TYPE;
sal personnel.salaire%TYPE;
mat personnel.Matricule%TYPE;
id_an vue_mat_annexe.id_annexe%TYPE;
BEGIN
SELECT date_debut INTO dateDebAct FROM action_totale
WHERE ID_action=:NEW.ID_action;
SELECT Matricule INTO mat FROM personnel_totale
WHERE matricule = :NEW.matricule;

IF(dateDebAct is null or mat is null) THEN
raise pb_reference;
END IF;

/* VÃľrifier que la date de dÃľbut est chronologiquement infÃľrireure Ãă la date de fin */
IF (:NEW.date_debut > :NEW.date_fin OR :NEW.date_debut = :NEW.date_fin) THEN
RAISE date_erronee ;
END IF;

/* Si une affectation concernant une action dÃľbute avant celle-ci  */
IF (dateDebAct > :NEW.date_debut) THEN
RAISE date_erronee ;
```

```
END IF;

/* Si une affectation concernant une action se termine
aprÃĺs celle-ci
 * c'est possible (action est prolongÃľe) mais demander
confirmation de l'utilisateur
 */
IF (dateFinAct < :NEW.date_fin) THEN

dbms_output.put_line('ATTENTION: L affectation
nouvellement entrÃľe se termine aprÃĺs la date de fin prÃľvue pour l action ');

END IF;

/* VÃľrifier que l'utilisateur n'est pas dÃľja affÃľctÃľ durant
la pÃľriode de l'affectation candidate Ãă l'insertion */
OPEN c1;
LOOP
FETCH c1 INTO dateDebAff, dateFinAff;
EXIT WHEN c1%NOTFOUND;

IF ((dateDebAff < :NEW.date_fin AND dateFinAff
> :NEW.date_fin) OR (dateFinAff > :NEW.date_debut
AND dateFinAff < :NEW.date_fin)) THEN RAISE deja_aff ;
END IF;
END LOOP;
CLOSE c1;

/* DÃľterminer si l'insertion doit Ãłtre effÃľctuÃľe en
base centrale (salaire != NULL) ou en base locale (salaire = NULL) */
SELECT salaire INTO sal FROM personnel_totale
WHERE matricule = :NEW.matricule;

IF (sal is not NULL) THEN
raise insertion_a_migrer;
ELSE
select id into num_annexe from ID_annexe;
select a.ID_annexe into annexe_affectation
from vue_mat_annexe a, site s, action act where a.ID_annexe = s.ID_annexe AND s.ID_site = act.ID_
IF(annexe_affectation != num_annexe) THEN
-- Le numÃľro de l'annexe local ne corrsepond pas Ãă celui du site insÃľrÃľ
raise mauvaise_annexe;
END IF;
END IF;

EXCEPTION
WHEN date_erronee THEN raise_application_error
(-20001, ' Les dates ne sont pas conforme aux dates de l action. ');
WHEN deja_aff THEN raise_application_error
(-20001, ' Cette personne est dÃľjÃă affectÃľe durant cette pÃľriode. ');
WHEN pb_reference THEN raise_application_error
(-20001, ' Un tuple reference n existe pas ');
WHEN insertion_a_migrer THEN raise_application_error
(-20001, ' L insertion doit etre faite sur la base centrale ');
```

```
WHEN mauvaise_annexe THEN raise_application_error
(-20001, 'Le site n est pas rattache a votre annexe');
END;
/

grant select on ID_ANNEXE to central;
grant select on PERSONNEL to central;
grant select on SITE to central;
grant select on ACTION to central;
grant select on MATERIEL_UTILISE to central;
grant select on AFFECTATION to central;

Commit;


-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------


insert into Id_annexe values (1);
commit;

insert into Personnel values ('', 'Lemer', 'Robert', 'Nantes', 'FRA', 1213);
insert into Personnel values ('', 'Ivanov', 'Vladimir', 'Moscou', 'RUS', null);
insert into Personnel values ('', 'Boubou', 'Mary', 'Diapomo', 'MAL', 1020);
insert into Personnel values ('', 'Al Bijour', 'Mohamed', 'Doha', 'QAT', null);
insert into Personnel values ('', 'Diahousa', 'Djibril', 'Picouli', 'RDC', null);
INSERT INTO Personnel VALUES ('', 'Petit','Fabrice',
'29 rue Blaise Diagne - Kaolack', 'SEN', null);
INSERT INTO Personnel VALUES ('', 'Travers', 'Babacar',
'18 Khartoum Street - Addis Abeba', 'ETH', 1256.48);
INSERT INTO Personnel VALUES ('', 'Nathalie', 'Bazire',
'House nÂř51, Street nÂř15 - Nyala', 'SDN', 1842.36);
INSERT INTO Personnel VALUES ('', 'Salif', 'Keita',
'15 Harambee Avenue - Nairobi', 'KEN', null);
INSERT INTO Personnel VALUES ('', 'Mamadou', 'Seck',
'95 rue Carno - Podort', 'SEN', 526.38);
INSERT INTO Personnel VALUES ('', 'Adama', 'Coulibally',
'130 Ushirika Road - Mombasa', 'KEN', 478.52);
INSERT INTO Personnel VALUES ('', 'NONYME', 'Alex',
'Le Havre', 'FRA', 1000.58);
INSERT INTO Personnel VALUES ('', 'NONYME', 'Yves',
'Rennes', 'FRA', 785.64);
INSERT INTO Personnel VALUES ('', 'DOE', 'John',
'Canberra', 'AUS', 1540.25);
INSERT INTO Personnel VALUES ('', 'DOE', 'Jenna',
'Canberra', 'AUS', NULL);
insert into Personnel values ( '', 'Dupond', 'Jacques',
'35 rue des verts prÃĽs - Paris', 'FRA', 152 );
insert into Personnel values ( '', 'Ramos', 'Maria',
'51 rua da picante - Madrid', 'ESP', 325.15);
insert into Personnel values ( '', 'Smith', 'Alexander',
'2 av Piccadilus - Angleterre', 'GBR', null );
```

```
insert into Personnel values ( '', 'Boutros', 'Pierre',
'87 av de gaulle - Toulouse', 'FRA', null );
commit;


INSERT INTO Site VALUES ('', 'Camp X10', '9,15 N', '103,27 E', 'LKA', 1);
insert into Site values ( '', 'Site du soleil levant', '156,52 N', '58,12 E', 'IND', 1 );
insert into Site values ( '', 'Site des montagnes', '145,56 N', '41,23 E', 'IND', 1 );
commit;


insert into Action values ('', 1, 'Reconstruction hopital X suite guerilla',
1, to_date('12/12/2004','dd/mm/yyyy'), to_date('08/09/2005','dd/mm/yyyy'), to_date('08/09/2005','

insert into Action values ('', 3, 'Seule aire de jeux de la ville',
1, to_date ('13/11/2005','dd/mm/yyyy'), to_date('06/03/2006','dd/mm/yyyy'), null);

insert into Action values ('', 2, 'Modernisation du reseau telephonique existant', 1, to_date('01

INSERT INTO Action VALUES ('', 5, 'Creer dans cette zone un dispensaire provisoire',
3, to_date ('10/01/2005','dd/mm/yyyy'), to_date('11/01/2007','dd/mm/yyyy'), null);

INSERT INTO Action VALUES ('', 4, 'Apporter a cette zone une premiere ecole',
2, to_date ('16/12/2002','dd/mm/yyyy'), to_date('01/01/2005','dd/mm/yyyy'),
to_date('23/07/2005','dd/mm/yyyy'));

INSERT INTO Action VALUES ('', 8, 'Aider les habitants a cultiver',
3, to_date ('22/11/2003','dd/mm/yyyy'), to_date('22/06/2004','dd/mm/yyyy'),
to_date('01/06/2004','dd/mm/yyyy'));

INSERT INTO Action VALUES ('', 9, 'Apprendre Ãă lire et Ãă Ãľcrire aux enfants',
1, to_date ('09/09/2004','dd/mm/yyyy'), to_date('12/07/2005','dd/mm/yyyy'),
to_date('30/06/2005','dd/mm/yyyy'));

INSERT INTO Action VALUES ('', 11, 'Apporter de la nourriture aux habitants',
3, to_date ('20/03/1998','dd/mm/yyyy'), to_date('22/04/1998','dd/mm/yyyy'),
to_date('20/04/1998','dd/mm/yyyy'));

INSERT INTO Action VALUES ('', 6, 'Aider les habitants a retrouver un travail',
2, to_date ('22/09/2000','dd/mm/yyyy'), to_date('12/12/2000','dd/mm/yyyy'),
to_date('12/12/2000','dd/mm/yyyy'));

INSERT INTO Action VALUES ('', 15, 'Vaccination contre la rage',
1, TO_DATE('25/01/2006','dd/mm/yyyy'), to_date('25/03/2006','dd/mm/yyyy'), null);

INSERT INTO Action VALUES ('', 4, 'Construction ÃľlÃľmentaire, 4 salles', 2,
TO_DATE('12/12/2005','dd/mm/yyyy'), to_date('01/04/2006','dd/mm/yyyy'), null);

insert into Action values ( '', 16, 'Sauver le monde', 2,
to_date('01/01/2006','dd/mm/yyyy'), to_date('31/12/2024','dd/mm/yyyy'), null );

insert into Action values ( '', 15, 'Aide aux enfants malades', 3,
to_date ('15/02/2006','dd/mm/yyyy'), to_date('12/02/2010','dd/mm/yyyy'), null);
```

```
insert into Action values ( '', 13, 'Sauver la baleine bleue', 3,
to_date('02/08/1992','dd/mm/yyyy'), to_date('05/01/1993','dd/mm/yyyy'),
to_date('12/01/1993','dd/mm/yyyy'));

insert into Action values ( '', 14, 'Construction d abris aux rÃľfugiÃľs de guerre', 2,
to_date ('01/03/1991','dd/mm/yyyy'), to_date('01/03/1992','dd/mm/yyyy'),
to_date('12/05/1992','dd/mm/yyyy'));

commit;


insert into Materiel_utilise values (11, 10, 2);
insert into Materiel_utilise values (12, 11, 50);
insert into Materiel_utilise values (12, 12, 3);
INSERT INTO Materiel_utilise VALUES (1, 4, 100);
INSERT INTO Materiel_utilise VALUES (1, 6, 200);
INSERT INTO Materiel_utilise VALUES (2, 2, 200);
INSERT INTO Materiel_utilise VALUES (2, 3, 200);
INSERT INTO Materiel_utilise VALUES (4, 5, 300);
INSERT INTO Materiel_utilise VALUES (2, 1, 50);
INSERT INTO Materiel_utilise VALUES (15, 1, 2500);
INSERT INTO Materiel_utilise VALUES (15, 3, 40);
INSERT INTO Materiel_utilise VALUES (6, 2, 50);
INSERT INTO Materiel_utilise VALUES (6, 4, 5);
insert into Materiel_utilise values ( 9, 4, 5 );
insert into Materiel_utilise values ( 2, 7, 64 );
insert into Materiel_utilise values ( 4, 9, 20 );
insert into Materiel_utilise values ( 8, 5, 3 );
commit;

insert into Affectation values (10, 3, to_date
( '12/12/2004','dd/mm/yyyy'), to_date ('12/01/2005','dd/mm/yyyy'));

insert into Affectation values (15, 2, to_date
('08/06/2005','dd/mm/yyyy'), to_date ('01/09/2005','dd/mm/yyyy'));

insert into Affectation values (12, 2, to_date
('01/06/2005','dd/mm/yyyy'), to_date ('23/08/2005','dd/mm/yyyy'));

INSERT INTO Affectation VALUES (1, 4,
to_date('10/09/2004','dd/mm/yyyy'), to_date ('16/01/2005','dd/mm/yyyy'));

INSERT INTO Affectation VALUES (2, 4,
to_date('19/12/2004','dd/mm/yyyy'), to_date ('01/03/2005','dd/mm/yyyy'));

INSERT INTO Affectation VALUES (3, 2, to_date ('05/01/2003','dd/mm/yyyy'),
to_date ('28/01/2003','dd/mm/yyyy'));

INSERT INTO Affectation VALUES (4, 6, to_date ('28/09/2000','dd/mm/yyyy'),
to_date ('24/11/2000','dd/mm/yyyy'));

INSERT INTO Affectation VALUES (5, 5, to_date('20/03/1998','dd/mm/yyyy'),
to_date('31/03/1998','dd/mm/yyyy'));
```

```
INSERT INTO Affectation VALUES (6, 1, to_date ('26/03/2005','dd/mm/yyyy'),
to_date('23/12/2006','dd/mm/yyyy'));

INSERT INTO Affectation VALUES (12, 1, TO_DATE('25/01/2006','dd/mm/yyyy'),
TO_DATE('25/02/2006','dd/mm/yyyy'));

INSERT INTO Affectation VALUES (12, 1, TO_DATE('25/02/2006','dd/mm/yyyy'),
TO_DATE('12/03/2006','dd/mm/yyyy'));

INSERT INTO Affectation VALUES (11, 2, TO_DATE('20/12/2005','dd/mm/yyyy'),
TO_DATE('25/02/2006','dd/mm/yyyy'));

INSERT INTO Affectation VALUES (8, 2, TO_DATE('25/01/2006','dd/mm/yyyy'),
TO_DATE('25/02/2006','dd/mm/yyyy'));

insert into Affectation values ( 5, 12, to_date('01/09/2005','dd/mm/yyyy'),
to_date('26/01/2006','dd/mm/yyyy') );

insert into Affectation values ( 9, 10, to_date('15/02/2006','dd/mm/yyyy'),
to_date('01/01/2007','dd/mm/yyyy') );

insert into Affectation values ( 3, 9, to_date('20/12/2005','dd/mm/yyyy'),
to_date('01/01/2006','dd/mm/yyyy') );

insert into Affectation values ( 1, 6, to_date('05/05/2006','dd/mm/yyyy'),
to_date('05/07/2006','dd/mm/yyyy') );
commit;
```