

Bases de Données Relationnelles

TP2 : Conception et création d'un schéma pour le traitement de marques déposées en SQL

SI2 et MAM2

1 Exposé du problème

On se propose de concevoir et réaliser une base de données de marques déposées et enregistrées. Les marques de ce système sont identifiées par un numéro à 5 chiffres démarrant à partir de 10000. Les attributs obligatoires des marques sont : nom, classe, pays (de dépôt et d'enregistrement). Dans un même pays et dans une même classe, on ne peut avoir deux marques de même nom. Autrement dit, une marque est parfaitement identifiée par le triplet (nom, classe, pays). Ainsi, Chanel dans la classe Textile en France n'est pas la même marque que Chanel dans les textiles en Espagne (même s'il existe des liens entre ces deux marques). La classe représente une catégorie normalisée (secteur industriel ou économique).

Plusieurs actes légaux peuvent être exercés sur une marque :

1. **Le dépôt** par un groupe de sociétés, sous son identité (nom, pays, classe). Plusieurs dépôts concurrents peuvent être effectués en parallèle par plusieurs groupes à propos de la même marque (i.e., pour le même nom de marque, dans la même classe, et dans le même pays). Tant que l'enregistrement d'un des dépôts n'a pas eu lieu, cette marque n'appartient à personne. Chaque dépôt est effectué conjointement par une ou plusieurs sociétés. Il s'effectue dans le pays de la marque. Un numéro à 7 chiffres de dépôt légal, global à l'ensemble des marques déposées dans le pays, est donné par le ministère de l'Industrie du pays concerné, ainsi qu'une date de dépôt. La numérotation est chronologique. Notons qu'une société localisée en France peut très bien déposer une marque en Espagne.
2. **L'enregistrement**. Un seul des dépôts effectués pour une marque est transformé en enregistrement (en général un an après le dépôt légal). Les déposants deviennent alors propriétaires légaux de la marque. Pour être enregistrée, une marque a donc dû être nécessairement déposée auparavant. L'enregistrement d'un dépôt d'une marque s'effectue bien sûr dans le même pays que le dépôt. Il comporte:
 - un numéro d'enregistrement chronologique global à l'ensemble des enregistrements effectués dans le pays,
 - une date d'enregistrement,
 - une référence au numéro de dépôt correspondant.

Seul un des groupes déposants voit son dépôt transformé en enregistrement. Les autres groupes reçoivent une notification de refus à leurs dépôts. Chaque refus comporte une référence au numéro de dépôt ainsi qu'une date de refus.

3. **La vente**. Une fois enregistrée, une marque peut être vendue conjointement par ses propriétaires à un autre groupe de sociétés. L'acte de vente est traduit légalement par un certificat constitué

d'un numéro d'acte de vente, d'une date, du nom et du pays de chacun des acheteurs, du numéro d'enregistrement de la marque, et éventuellement du numéro légal de la vente précédente. Ici encore le numéro de vente à 7 chiffres est global à l'ensemble d'un pays.

Lorsque tous les dépôts effectués à propos de la marque ont été refusés, le statut de cette marque devient libre vis-à-vis de tout futur dépôt. Grâce à cet état on peut savoir qu'une marque a déjà été déposée et refusée.

Les changements d'état d'une marque sont donc :

Nouvelle | Déposée | Libre \rightarrow DEPOT \rightarrow Déposée

Déposée \rightarrow REFUS \rightarrow Déposée | Libre

Déposée \rightarrow ENREGISTREMENT \rightarrow Enregistrée

Enregistrée \rightarrow VENTE \rightarrow Enregistrée

L'enchaînement des opérations est le suivant:

DEPOT \rightarrow REFUS | ENREGISTREMENT

ENREGISTREMENT \rightarrow VENTE

2 Travail demandé

2.1 Conception des objets SQL et des contraintes SQL

On demande de concevoir les objets SQL et les contraintes SQL permettant de représenter ces marques, dépôts, enregistrements, refus, ventes-achats, en respectant les objectifs suivants :

1. On souhaite stocker tous les actes légaux effectués.
2. On souhaite que le système puisse rapidement donner à tout instant :
 - le statut de la marque (déposée, enregistrée, libre)
 - le(s) propriétaire(s) actuel(s) d'une marque enregistrée (des ventes peuvent avoir lieu après l'enregistrement).

Dans la base de données *marquesdeposees* vous trouverez

- un schéma à votre nom dans lequel vous avez le droit de créer des tables
- un schéma *donnees* sur les tables duquel vous avez le droit d'opérer des SELECT et des REFERENCES. Ce schéma *donnees* contient quatre tables : *classe* qui répertorie les classes existantes, *societe* qui contient les sociétés propriétaires, déposantes ou simplement clientes, *pays* et *marque*

Les contraintes qui ne peuvent s'exprimer lors de la création seront décrites de manière textuelle .

Vous devez écrire les requêtes SQL pour vérifier les contraintes suivantes :

- Vérification que les marques, groupes, dépôts, enregistrements référencés dans les tables correspondantes existent bien
- Vérification que la numérotation des enregistrements est globale pour chaque pays
- Vérification que la numérotation des dépôts est chronologique pour chaque pays

Remarque : La seule difficulté de conception réside dans le fait qu'il peut y avoir, pour un même acte (dépôt, enregistrement., vente.) plusieurs sociétés impliquées. Notez bien la contrainte qui dit que les vendeurs d'une marque sont nécessairement tous les propriétaires de la marque au moment de la vente.

— TP2 : Création d'un schéma pour le traitement des marques déposées —

```
\c tpmarquedeposees;
```

— les tables vont être créées dans le schéma de l'utilisateur courant

— Suppression des tables existantes dans l'ordre inverse de leur création

```
DROP TABLE ventes ;
DROP TABLE enregistrements ;
DROP TABLE refus ;
DROP TABLE depots ;
DROP TABLE marque_new ;
DROP TABLE groupes_societes ;
DROP TABLE statuts ;
DROP TABLE groupes ;
```

— Création des tables pour le traitement des marques déposées

```
CREATE TABLE groupes (
    IdGroupe INT PRIMARY KEY);
```

```
CREATE TABLE groupes_societes (
    IdGroupe INT NOT NULL REFERENCES groupes,
    IdSociete INT NOT NULL REFERENCES societe,
    primary key (IdGroupe, IdSociete ));
```

```
CREATE TABLE statuts (
    Id char PRIMARY KEY,
    Nom varchar(20) NOT NULL);
insert into statuts values ('d', 'déposée');
insert into statuts values ('n', 'nouvelle');
insert into statuts values ('l', 'libre');
insert into statuts values ('e', 'enregistrée');
```

```
CREATE TABLE marque_new (
    IdMarque INT PRIMARY KEY CHECK (IdMarque >= 1000 and IdMarque < 100000),
    Nom varchar(20) NOT NULL,
    Classe INT NOT NULL,
```

```

Statut char DEFAULT 'n' NOT NULL references statuts ,
Pays char(2) NOT NULL REFERENCES pays ,
IdGroupe INT REFERENCES groupes ,
CONSTRAINT Cle_Sec1 UNIQUE (Nom, Classe , Pays));

```

```

CREATE TABLE depots (
    NumLegal INT NOT NULL,
    Pays char(2) NOT NULL REFERENCES pays ,
        PRIMARY KEY (NumLegal, Pays),
    DateDepot date NOT NULL,
    IdMarque INT NOT NULL REFERENCES marque_new,
    IdGroupe INT NOT NULL REFERENCES groupes
);

```

```

CREATE TABLE refus (
    NumDepot INT NOT NULL,
    PaysDepot char(2) NOT NULL REFERENCES pays ,
        FOREIGN KEY (NumDepot, PaysDepot) REFERENCES Depots ,
    DateRefus date NOT NULL)
;

```

— La table refus est nécessaire si on veut memoriser la date de refus

```

CREATE TABLE enregistrements (
    NumLegal INT NOT NULL,
    Pays char(2) NOT NULL REFERENCES pays ,
        PRIMARY KEY (NumLegal, Pays),
    DateEnreg date NOT NULL,
    NumDepot INT NOT NULL,
    PaysDepot char(2) NOT NULL REFERENCES pays ,
        FOREIGN KEY (NumDepot, PaysDepot) REFERENCES Depots
);

```

```

/*
Contrainte textuelle :
Une marque enregistrée doit forcément avoir été déposée: dans cette table ,
un enregistrement fait reference a un numéro de dépôt et par conséquent
la contrainte est toujours verifiée
*/

```

```

CREATE TABLE ventes (
    IdVente SERIAL PRIMARY KEY, — Ajouté pour facilité la référence à la vente précédente
    NumLegal INT NOT NULL,
    PaysVente char(2) NOT NULL REFERENCES pays ,
        UNIQUE (NumLegal, PaysVente),
    DateVente date NOT NULL,
    IdGroupe INT NOT NULL REFERENCES groupes ,
    IdEnreg INT NOT NULL,
    PaysEnreg char(2) NOT NULL REFERENCES pays ,
        FOREIGN KEY (IdEnreg, PaysEnreg) REFERENCES enregistrements ,
);

```

```

      VentePrec INT REFERENCES ventes) ;
/*
  Contrainte textuelle :
  Une marque ne peut etre vendue que si elle est enregistrée.
  Dans notre table, une vente fait référence a un numéro d'enregistrement et
  par conséquent la contrainte est toujours vérifiée
*/

```

— TP2 : Partie 2 : vérification de contraintes

— Vérification que les marques référencées dans les dépôts existent bien

```

SELECT D.IdMarque
FROM Depots D
WHERE NOT EXISTS (SELECT M.Id
                  FROM Marque M
                  WHERE M.Id = D.IdMarque);

```

— Vérification que les groupes référencés dans les dépôts existent bien

```

SELECT D.IdGroupe
FROM Depots D
WHERE NOT EXISTS (SELECT G.IdGroupe
                  FROM Groupes G
                  WHERE G.IdGroupe = D.IdGroupe);

```

— Vérification que les dépôts référencés dans les enregistrements existent bien

```

SELECT E. NumLegal,E.Pays
FROM Enregistrements E
WHERE NOT EXISTS (SELECT *
                  FROM Depots D
                  WHERE E. NumLegal = D. NumLegal AND E.Pays = D.Pays);

```

— Vérification que La numérotation des dépôts est chronologique pour chaque pays

```

SELECT  D1.NumLegal, D1.Pays, D2.NumLegal, D2.Pays
FROM    Depots D1, Depots D2
WHERE   D1.Pays = D2.Pays
      AND D2.DateDepot > D1.DateDepot
      AND D2.NumLegal < D1.NumLegal;

```

— Vérification que La numérotation des enregistrements est globale pour chaque pays (véri

```

SELECT  E1.NumLegal, E1.Pays, E2.NumLegal, E2.Pays
FROM    Enregistrements E1, Enregistrements E2
WHERE   E1.Pays = E2.Pays
        AND E1.NumLegal = E2.NumLegal
        AND E1.NumDepot != E2.NumDepot ;

```

2.2 Traitement d'un lot de Dépôts de Marques

On se propose de traiter un lot initial de dépôts de marques provenant du Ministère de l'Industrie et déjà saisi dans les fichiers *depots_init* et *deposants_init* (cf /net/home/r/rueher/BDR/Tables). Ces fichiers contiennent:

Table depots_init

Colonne	Type	Modificateurs
nom	character varying(30)	not null
classe	integer	not null
pays	character varying(50)	
num_dep	integer	not null
date_dep	date	not null

Table deposants_init

Colonne	Type	Modificateurs
idd	integer	not null default nextval('public.deposants_init1_idd_seq'::text)
num_dep	integer	not null
pays	character varying(50)	not null
deposant	integer	not null

Dans ce traitement vous devrez :

- Eliminer les dépôts entachés d'erreurs (i.e., violant certaines contraintes) et les stocker dans une table DEPOTS_INCOR
- Insérer les données correspondant aux dépôts corrects, dans les diverses tables constituant le schéma de votre base (marques, dépôts, groupes ...)

Vous procéderez en deux phases :

1. Copier les fichiers *depots_init* et *deposants_init* dans des tables de votre schéma. Pour chacun des dépôts intervenant dans la violation d'une des contraintes suivantes, copier ce dépôt dans votre table DEPOTS_INCOR, et l'éliminer de votre table DEPOT_INIT. Les contraintes à vérifier sont:
 - (a) Deux dépôts dans un même pays et avec le même numéro doivent concerner la même marque.
 - (b) Dans un même pays les numéros de dépôt respectent l'ordre chronologique
 - (c) La classe figurant dans un dépôt doit exister dans la table CLASSE du schéma "données" la base "tpmarquedéposées".
 - (d) Le pays figurant dans un dépôt doit exister dans la table PAYS du schéma "données" la base "tpmarquedéposées".
-

— TP2: Phase 3

```
\c tpmarquedeposees;
```

```
DROP VIEW cstc;  
DROP VIEW cstb;  
DROP VIEW csta;  
DROP TABLE depots_incor;  
DROP TABLE depots_init1 CASCADE;  
DROP TABLE depots_init;  
DROP TABLE depositants_init1;  
DROP TABLE depositants_init;
```

— Etape 1

— Copie des tables depots_init et depositants_init

```
CREATE TABLE depots_init (  
    nom VARCHAR(30) NOT NULL,  
    classe INT NOT NULL,  
    pays VARCHAR(50),  
    num_dep INT NOT NULL,  
    date_dep DATE NOT NULL  
);  
  
\copy depots_init from 'Tables/depots_init'
```

— Creation d'une copie de travail de la table des depots initiaux avec une clé

```
CREATE TABLE depots_init1 (  
    idd SERIAL,  
    nom VARCHAR(30) NOT NULL,  
    classe INT NOT NULL,  
    pays VARCHAR(50),  
    num_dep INT NOT NULL,  
    date_dep DATE NOT NULL  
);  
INSERT into depots_init1 (nom, classe, pays, num_dep, date_dep)  
SELECT nom, classe, pays, num_dep, date_dep from depots_init;
```

```
CREATE TABLE depositants_init (  
    num_dep INT NOT NULL,  
    pays VARCHAR(50) NOT NULL,  
    depositant INT NOT NULL  
);
```

```
\copy  depots_init from 'Tables/depots_init'
```

— Creation d'une copie de travail de la table des depots initiaux avec une clé

```
CREATE TABLE depots_init1 (
    idd SERIAL,
    num_dep INT NOT NULL,
    pays VARCHAR(50) NOT NULL,
    depot INT NOT NULL
);
INSERT into  depots_init1 (num_dep,pays,depot)
    select num_dep,pays,depot from depots_init;
```

— Etape 2 : vérification des contraintes a, b, c et d

— Les contraintes a et d peuvent être vérifiées à l'aide d'une
— référence si les tables concernées ont une clé primaire

— a) Deux depots dans un meme pays avec le meme numero doivent concerner
— la meme marque

```
select P1.idd, P1.nom, P2.idd, P2.nom from  depots_init1 P1,  depots_init1 P2
    where P1.pays=P2.pays and P1.num_dep=P2.num_dep and
    (P1.nom<P2.nom or P1.pays<P2.pays or P1.classe <P2.classe);
```

```
— idd |      nom      | idd |      nom
-----+-----+-----+-----
—  80 | COCA COLA |  59 | Coca Cola
```

— b) dans un meme pays, les numeros de depot respectent l'ordre
— chronologique

```
select P1.idd,  P1.date_dep, P2.idd, P2.date_dep
    from depots_init1 P1,  depots_init1 P2
    where P1.pays=P2.pays and P1.num_dep<P2.num_dep and P1.date_dep>P2.date_dep
```

```
— idd |  date_dep  | idd |  date_dep
—  35 | 1990-05-01 |  81 | 1990-04-01 (35 : Dune, 81 : Poison )
```

— c) la classe figurant dans un depot doit exister dans la table
— classe

```
select * from depots_init1 P where not exists
    (select * from classe C where P.classe=C.num);
```

```
— idd |  nom  | classe |  pays  | num_dep |  date_dep
—  82 | Passat |      37 | Germany |  715400 | 1991-05-01
```



```

— d) le pays figurant dans un depot doit exister dans la table
—     pays

```

```

select D.num_dep, D.pays from depots_init D where D.pays not in
  (select P.nom from pays P );
— num_dep | pays
— (0 rows)

```

```

—
— Etape 3 : pré-traitement des données
—

```

```

— creation de la table des depots incorrects

```

```

create table depots_incor
(
  noer      INTEGER, — numéro de la contrainte violée
  idd       INTEGER,
  nom VARCHAR(30) not null,
  classe   INTEGER not null,
  pays     VARCHAR(50) not null,
  num_dep  INTEGER not null,
  date_dep DATE not null,
  constraint cle_dinc primary key(idd,noer)
);

```

```

— insertion des tuples qui violent les constraints a, b et c dans la table des depots

```

```

— contrainte a: il s'agit du 59 et du 80

```

```

create VIEW  csta AS (select 1, D1.idd, D1.nom, D1.classe, D1.pays, D1.num_dep, D1.date_dep
from depots_init1 D1, depots_init1 D2
  where D1.pays=D2.pays and D1.num_dep=D2.num_dep and
        D1.nom<>D2.nom) ;

```

```

— contrainte b: il s'agit des tuples 35 et 81

```

```

create VIEW  cstb AS (select 2, D1.idd, D1.nom, D1.classe, D1.pays, D1.num_dep, D1.date_dep
from depots_init1 D1, depots_init1 D2
where D1.pays=D2.pays and D1.num_dep<D2.num_dep
and D1.date_dep>D2.date_dep
UNION
select 2, D2.idd, D2.nom, D2.classe, D2.pays, D2.num_dep, D2.date_dep
from depots_init1 D1, depots_init1 D2
where D1.pays=D2.pays and D1.num_dep<D2.num_dep
and D1.date_dep>D2.date_dep
);

```

```
-- contrainte c: il s'agit du tuples 82
```

```
create VIEW  cstc  AS (select 3,  
    D.idd, D.nom, D.classe, D.pays, D.num_dep, D.date_dep  
    from depots_init1 D where not exists  
        (select * from classe C where D.classe=C.num));
```

```
-- insertion dans la table des dépôts incorrects  
insert into depots_incor  (select * from  csta union select * from  cstb union select  
    cstc);
```

```
-- INSERT 0 5
```

```
-- retrait de la table depots_init des tuples invalides  
delete from depots_init1  
where idd in (select idd from depots_incor);  
-- DELETE 5
```

-
2. Pour chacune des dépôts corrects, insérer dans votre schéma les données correspondantes.
-

```
-- TP2:  Phase 4
```

```
DROP SEQUENCE seq_idm;  
DROP Table tmp_grp;  
DROP SEQUENCE seq_groupe;
```

```
-- Etape 4 : remplissage des tables
```

```
-- Suppression des anciennes valeurs  
delete from depots;  
delete from marque_new;  
delete from groupes_societes;  
delete from groupes;
```

```
-- Groupes:  
-- creation de la sequence des groupes
```

```
CREATE SEQUENCE seq_groupe START 1 INCREMENT 1;
```

```
-- creation d'une table temporaire tel que (num_dep,pays)<-> idg  
-- (il s'agit d'un majorant du nombre de groupes)
```

```
CREATE TEMPORARY TABLE tmp_grp
```

```

(      idg      INT NOT NULL,
      ndep      INT NOT NULL, — ndep et pays : utiles pour l'identification d'un dépôt
      Pays      varchar(50) NOT NULL
);

— INSERT INTO tmp_grp (SELECT nextval('seq_groupe'), T.numdep, T.pays
—      FROM (SELECT DISTINCT D.num_dep AS numdep, D.pays AS pays
—      FROM depots_init1 D) AS T);
—INSERT 0 77

INSERT INTO tmp_grp
      SELECT nextval('seq_groupe'), D.num_dep, D.pays
      FROM (SELECT num_dep, pays
      FROM depots_init1
      GROUP BY num_dep, pays) AS D;

INSERT INTO Groupes SELECT idg FROM tmp_grp;
—INSERT 0 77

— Societes (on suppose que les sociétés qui font des dépôts existent dans la table so

— Groupes_societes
INSERT INTO Groupes_societes (IdGroupe, IdSociete)
      SELECT G.idg, T.deposant
      FROM tmp_grp AS G, depositants_init AS T
      WHERE G.ndep = T.num_dep
      AND G.pays = T.pays;
—INSERT 0 79

— Marques
— on cree une sequence des id. de marque pour que ceux-ci verifient
— la contrainte sur le nombre de chiffres

create sequence seq_idm start 10000 increment 1 MAXVALUE 99999;

insert into marque_new
      select nextval('seq_idm'), NOM, CLASSE, 'n', PAYS from marque M;
—INSERT 0 92

—Depots
— on cree la sequence des id de depots qui est locale a ce schema

insert into depots
      select D.num_dep, P.code, D.Date_dep, M.idMarque, G.idg
      from depots_init1 D, marque_new M, tmp_grp G, pays P
      where M.nom = D.nom
      and M.classe = D.classe
      and M.pays = P.code
      and P.nom = D.pays
      and G.ndep = D.num_dep

```

```

                                and G.pays = D.pays
;
—INSERT 0 77

— Mise à jour du statut dans marque_new
— (les groupes ne sont pas mis à jour car une groupe ne devient propriétaire que lors

UPDATE  marque_new  SET statut = 'd'
      WHERE IdMarque in (select D.IdMarque from depots D );
— UPDATE 77

```
