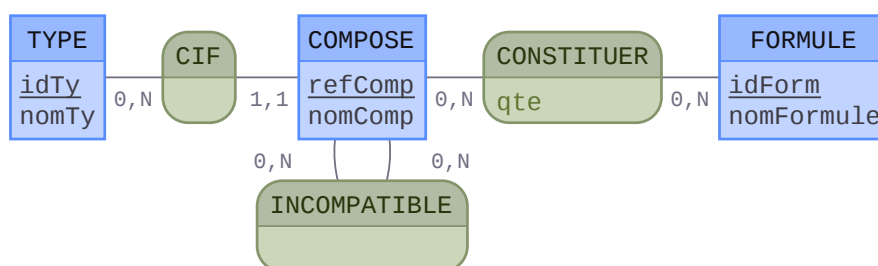


M2106: BASES DE DONNÉES AVANCÉES

Feuille de TP n°A Rendre

Les procédures et les fonctions stockées – Trigger
A rendre sur Celene sous la forme d'un script SQL

Une entreprise pharmaceutique possède la base de données suivante lui permettant de stocker les différentes formules qu'elle fabrique. Une formule est constituée de composés. Certains composés sont incompatibles et il est légalement interdit de les mettre ensemble dans une formule.



Le MLD correspondant est le suivant

Compose (**refComp**, nomComp)

Constituer (**#refComp**, **#idForm**, qte)

Formule (**idForm**, nomForm)

Incompatible (**#refComp.1**, **#refComp.2**)

Et le script de création des tables est le suivant

```
CREATE TABLE COMPOSE (  
    refcomp varchar(4),  
    nomcomp varchar(20),  
    PRIMARY KEY (refcomp)  
);  
  
CREATE TABLE CONSTITUER (  
    idform varchar(5),  
    refcomp varchar(4),  
    qte decimal(5,2),  
    PRIMARY KEY (refcomp, idform)  
);  
  
CREATE TABLE FORMULE (  
    idform varchar(5),  
    nomform varchar(15),  
    PRIMARY KEY (idform)  
);  
  
CREATE TABLE INCOMPATIBLE (  
    refcomp_1 varchar(4),  
    refcomp_2 varchar(4),  
    PRIMARY KEY (refcomp_1, refcomp_2)  
);  
  
ALTER TABLE CONSTITUER ADD FOREIGN KEY (idform) REFERENCES FORMULE (idform);  
ALTER TABLE CONSTITUER ADD FOREIGN KEY (refcomp) REFERENCES COMPOSE (refcomp);  
ALTER TABLE INCOMPATIBLE ADD FOREIGN KEY (refcomp_1) REFERENCES COMPOSE (refcomp);  
ALTER TABLE INCOMPATIBLE ADD FOREIGN KEY (refcomp_2) REFERENCES COMPOSE (refcomp);
```

Exercice 1 *Les curseurs : Incompatibilités*

La table INCOMPATIBLE pour être cohérente doit être symétrique c'est-à-dire que si la ligne ('xxxx', 'yyyy') est dans la table, il faut que la ligne ('yyyy', 'xxxx') doit y être aussi. On vous demande d'écrire une procédure `completeIncompatible()` qui va rajouter toutes les lignes manquantes dans la table INCOMPATIBLE.

Exercice 2 *Curseurs : Commande de produits*

Les quantités indiquées dans la table CONSTITUER sont pour fabriquer 1 litre de la formule. La base de données intègre la nouvelle table suivante

```
create table PRODUCTION(idform varchar(5) primary key, qteAProduire int,
                        foreign key (idform) references FORMULE(idform) );
insert into PRODUCTION values (Co001,5),(Po001,10);
```

Cette table indique la production prévue pour chaque formule ; la colonne `qte` est indiquée en litres.

On vous demande de créer une fonction stockée `effectuerCommande()` `returns varchar(500)` permettant d'éditer automatiquement la commande de composés à faire à partir des informations contenues dans la table PRODUCTION (ainsi que les informations des autres tables de la base de données). Votre fonction ne doit effectuer qu'une seule requête d'interrogation. Celle-ci doit retourner toutes les informations dont vous avez besoin triées dans le bon ordre pour effectuer le traitement.

Pour l'instance de la base de données fournie avec le TP, et les quantités de production données ci-dessus, vous devez obtenir (Attention ! L'affichage sous MySQL est contient une entête plein de signes '-' non reproduits ici) :

```
select effectuerCommande();
```

Al01 Alcool	
Pommade	12.00
quantite totale	12.00
Di01 Dioxyde de soufre	
Pommade	5.00
Comprimés	0.05
quantite totale	5.05
Di02 Dioxyde d'oxygène	
Comprimés	6.00
quantite totale	6.00
Su01 Sucre	
Comprimés	26.50
quantite totale	26.50

Pour obtenir un bon alignement des chiffres vous pouvez utiliser les fonctions `rpad` et `lpad` qui permettent de forcer une chaîne de caractères à une certaine longueur en remplissant les caractères manquants par un caractère de remplacement (`pad`). Par exemple `rpad('toto',6,'-')` va produire la chaîne `'toto--'` car `'toto'` ne contient que 4 caractères et donc `rpad` va l'étendre à 6 caractères en ajoutant deux `'-'` à droite. De même, `lpad(6.52,6,'0')` va retourner `'006.52'`. Noter que le premier argument peut être un nombre.