

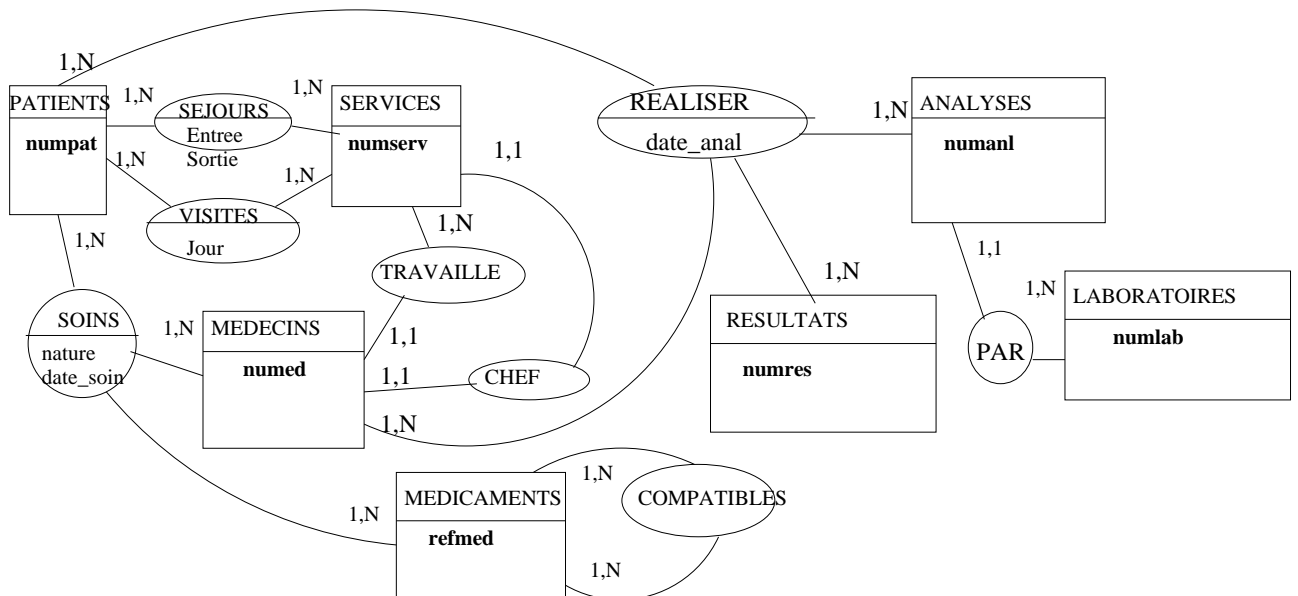
## TP de BASES DE DONNEES PROGRAMMATION PL/SQL

Ce TP a pour but de vous faire écrire des fonctions et procédures en PL/SQL. Dans le TP suivant, vous travaillerez les triggers.

On vous donne ci-dessous le MCD correspondant à la *gestion d'un hôpital*, et en fin de ce sujet, le schéma relationnel de la base. Ce TP est la propriété d'un pseudo-utilisateur nommé **HOSTO**, qui vous a donné des droits de lecture sur ses tables. Vous ne pouvez cependant écrire pour autant des traitements destinés à manipuler ces dernières. Vous devrez donc, dans un premier temps, recopier chez vous celles qui vous seront nécessaires (et seulement celles-là afin d'économiser l'espace commun au groupe LICENCE).

Vous trouverez à l'adresse suivante le script sql permettant de créer la base avec ses données :  
[http ://www.lifl.fr/~kuttler/bdd2010/tps/hosto.sql](http://www.lifl.fr/~kuttler/bdd2010/tps/hosto.sql)

Vous n'êtes pas obligé de tout importer, mais vous devrez impérativement être propriétaire des tables sur lesquelles vous mettrez en place les triggers qu'il vous sera demandé d'écrire dans le TP suivant.



### Fonctions et procédures

- la fonction `duree_sejour(patient number, de char)` **return number** qui calcule la durée du séjour d'un patient de numéro donné, entré à l'hôpital à la date d'entrée `de`.
- la fonction `cout_sejour(patient number, de char)` **return number** qui calcule le coût total du séjour d'un patient donné. Ce coût est composé :
  - du prix de chaque journée passée à l'hôpital
  - de celui des visites de médecins
  - de celui des analyses et des médicaments prescrits lors des soins prodigués.

REM : pour ce faire, vous devrez réaliser des comparaisons entre les dates de séjour. Or, celles-ci sont représentées sous forme de chaînes de caractères. De ce fait, une date d'entrée correspondant par exemple au 28 novembre sera considérée comme postérieure à la date de

sortie correspondant, elle, au 2 décembre. Une éventuelle interrogation pour un soin compris entre ces deux dates ne ramènera que la valeur NULL.

Pour contourner cette difficulté, vous devrez concevoir, écrire et tester une fonction auxiliaire permettant de réaliser une comparaison valide entre deux dates, quelles qu'elles soient.

3. la fonction `cout_moyen_journee()` **return number** qui retourne le résultat en invoquant la fonction précédente pour l'ensemble de tous les séjours effectués par l'ensemble des patients ayant séjourné dans l'hôpital.
4. la fonction `cout_total_patient(patient number)` **return number** qui évalue ce qu'a pu coûter un patient donné lors de l'ensemble de ses séjours à l'hôpital.
5. la procédure `historique_medicaments(patient number)` qui affiche, classés par date de soin, la désignation de tous les médicaments prescrits à un patient donné.
6. la procédure `historique_soins(numpat number)` qui, pour un patient donné :
  - affiche les dates d'entrée et de sortie des différents séjours.
  - pour un séjour donné, indique le nom des médecins ayant assurés des soins
  - pour chaque médecin, la liste des médicaments prescrits, ainsi que celle des analyses avec leur résultat.
7. la procédure `qui_a_soigne(patient number)` qui affiche la liste des médecins ayant assuré des soins à un patient donné, en précisant la date d'intervention.

## Schéma relationnel

```
create table medicaments(refmed number primary key, designation varchar2(30),
    laboratoire varchar2(30), prix float);

create table patients(numpat number primary key, nom varchar2(30), prenom varchar2(30),
    ville varchar2(50), sexe char(1));

create table medecins(numed number primary key, nom varchar2(30), prenom varchar2(30),
    specialite varchar2(50), service varchar2(50));

create table laboratoires(numlab number primary key, nom varchar2(50));

create table services(numserv number primary key, nomserv varchar2(30), chef number);

create table analyses(numanl number primary key, designation varchar2(30),
    laboratoire number, prix float);

create table sejours(numsejour number primary key, numpat number, numserv number,
    entree char(10), sortie char(10));

create table soins(numsoin number primary key, numpat number, numed number,
    refmed number, date_soin char(10));

create table realiser(numrealise number primary key, numpat number, numed number,
    numanl number, numres number, date_analyse char(10));

create table resultats(numres number primary key, libelle varchar2(100));

create table compatibles(refmed1 number, refmed2 number, primary key(refmed1,refmed2));

create table tarifs(prix_journee number(10,2), prix_visite number(10,2));
```