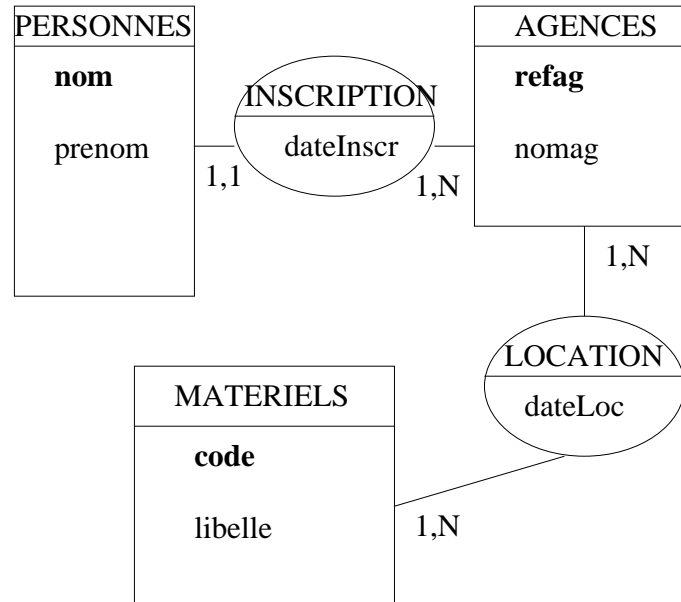


# LS5 Info - Examen de Bases de Données 2de session

Durée 2h, aucun document autorisé, appareils électroniques interdits

## Question 1 :

Donnez le *schéma relationnel* correspondant au *schéma conceptuel* (MCD) de la figure suivante :



## Question 2 :

On considère maintenant la base commerciale définie par le schéma relationnel suivant :

```

vendeurs(numVendeur, nomVendeur, salaire)
ventes(numVendeur, numProduit, dateVente, quantiteVendue)
produits(numProduit, libelleProduit)
  
```

Pour chacune des requêtes erronées suivantes, donnez très précisément la nature des erreurs et précisez comment en assurer la correction. Beaucoup d'entre-elles comportent une, voire plusieurs erreurs syntaxiques. D'autre(s) peuvent n'avoir que des erreurs de conception. Vous devez énoncer toutes les erreurs que vous pouvez distinguer. A titre d'exemple, voici une requête erronée et l'ensemble des réponses à proposer :

Donnez la liste des vendeurs dont le salaire n'est pas compris entre 2000 et 2500 euros et dont le nom ne contient pas deux fois la lettre 'L' (sans tenir compte de la casse).

```
select nom, prenom from vendeurs where salaires > 2000 and nom not like 'L%l' ;
```

Erreurs :

- l'attribut **salaire** est mal orthographié
- La condition portant sur le salaire est fausse
- La condition portant sur le nom ne retient que ceux dont le nom commence par un L majuscule et se termine par un l minuscule.

Correction :

```
select nom, prenom from vendeurs where salaire not between 2000 and 2500 and lower(nom)
not like '%l%l%' ;
```

Voici maintenant les requêtes à commenter et corriger :

1. *Quels sont les vendeurs ayant vendu plus de 10 produits*  
`select numVendeur, max(quantiteVendue) from ventes where max(quantiteVendue) > 10;`
2. *Donnez, par vendeur, le total des ventes effectuées*  
`select numVendeur, sum(quantiteVendue) from ventes order by numVendeur;`
3. *Donnez la liste ordonnée des produits vendus par chaque vendeur*  
`select numVendeur, libelleProduit from ventes, produits order by numVendeur, libelleProduit;`
4. *Donnez la liste des vendeurs dont les ventes ont porté sur plus de 10 exemplaires d'un produit*  
`select distinct numvendeur from vendeurs where quantiteVendue>10;`
5. *Quels sont les vendeurs ayant vendu plus de 10 produits différents le même jour*  
`select numVendeur, sum(quantiteVendue) from vente group by numVendeur order by numProduit, dateVente;`

### Question 3 :

Une société commerciale est composée de plusieurs agences décrites par la table relationnelle :  
agences(numAgence, nomAgence)

Au sein de cette société opèrent des vendeurs, et on fera les hypothèses suivantes : un vendeur ne travaille que pour le compte d'une seule agence et il n'y a pas plus d'une vente par jour d'un produit donné par un vendeur donné. Les tables **Vendeurs** et **Produits** sont identiques à celles définies dans la question deux.

Voici maintenant plusieurs versions possibles de la table **ventes**. Pour chacune d'elle, on vous demande de préciser, en le justifiant, la forme normale. Chaque fois que ce sera nécessaire, vous donnerez une décomposition conservative correspondant au moins à la troisième forme normale.

1. Ventes(numVendeur, numProduit, dateVente, quantiteVendue)
2. Ventes(numVendeur, numProduit, dateVente, quantiteVendue, libelleProduit, montantVente)
3. Ventes(numVendeur, numProduit, dateVente, quantiteVendue, prixUnitaireProduit, montantVente)
4. Ventes(numVendeur, numProduit, dateVente, quantitevendue, numClient, nomClient)
5. Ventes(numAgence, numProduit, dateVente, quantiteVendue, numVendeur)

### Question 4 :

Une société de vente de voitures gère la base suivante :

Clients(numClient, nom, prenom)

Vehicules(codeVehicule, marque, type, puissance, couleur, prixBase)

Options(codeOption, libelle, prixOption)

Compatibilite(codeOption, codeVehicule)

Configuration(immatriculation, codeOption)

Ventes(numClient, codeVehicule, dateVente, immatriculation, montantReprise, montantOptions, prixVente)

La configuration d'un véhicule représente le profil exact du véhicule acheté.

Les véhicules mis en vente ont un prix de base. Un client peut demander de rajouter des options. Certaines sont possibles, d'autres pas, selon le modèle de véhicule qu'il souhaite acquérir. Toutes ces options ont naturellement un prix, et le montant de toutes les options commandées devra être rajouté au prix de base du véhicule. Il arrive que le vendeur propose une offre de reprise pour l'ancien véhicule du client. La somme correspondante est dans ce cas déduite du prix de base. Le prix de vente final se déduit de ces ajouts et déductions.

1. Ecrire la procédure PL/SQL `montantVente(numCli number, codVec char, dV char)` qui calcule et insère dans la table des ventes le prix de vente du véhicule que le client s'apprête à acheter. On fera l'hypothèse qu'un client n'achète qu'un seul véhicule d'un type donné à une date donnée.  
Les paramètres de cette procédure représentent, respectivement, le numéro du client, le code du véhicule et la date de la vente.
2. La société propose des crédits à la vente. Les clients acceptant la proposition de crédit devront effectuer leur premier versement le 5 du premier mois qui suit la date de vente du véhicule acheté. La durée du crédit est de 36 mois.  
Ecrire la fonction PL/SQL `debutCredit(dateVente char) return char` qui retourne la date du premier versement.  
Ecrire de même la fonction `finCredit(dateVente char) return char` qui retourne la date de fin du dernier versement.  
Rappel : le type SQL `date` possède une arithmétique portant sur le nombre de jours. Les fonctions de conversion sont, sous Oracle :  
`to_date(uneDate char, format char) return date` et  
`to_char(uneDate date, format char) return char`.  
Le format est une chaîne indiquant la présentation de la date à la saisie ou l'affichage. On pourra adopter le format standard européen : 'DD/MM/YYYY' pour les années à 4 chiffres.
3. Ecrire en PL/SQL le trigger permettant de forcer, lors d'une insertion comme d'une mise à jour, les noms et prénoms en *minuscule* avec *majuscule commençante* du client, quelque soit le mode de saisie des données. Rappel : la fonction SQL (oracle) permettant de transformer une chaîne selon ce critère est `initcap(chaîne)`.