

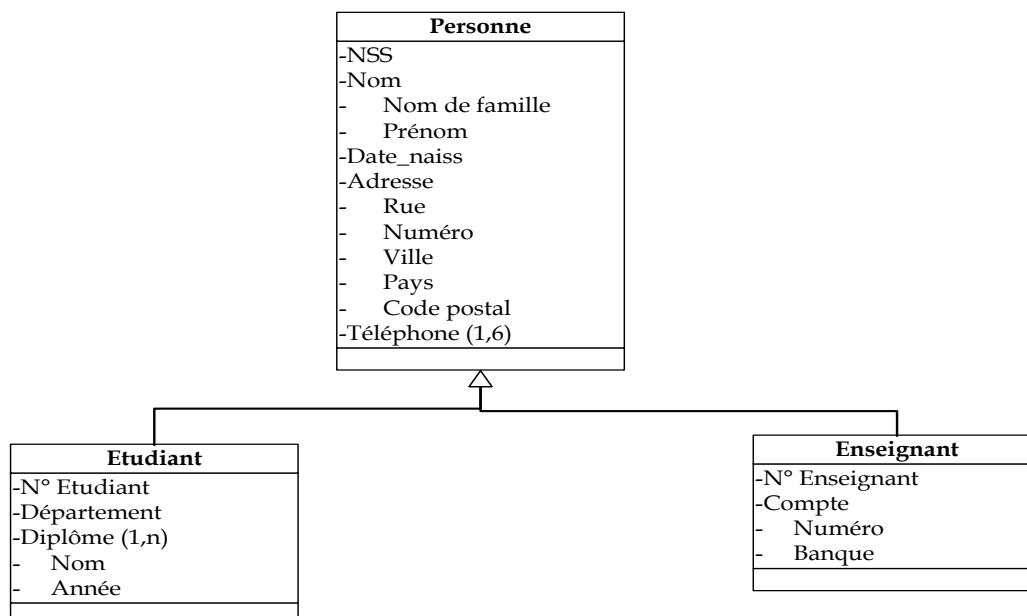
### TD N° 3 : SQL3

#### Exercice 1 :

1. Soit la classe personne suivante :

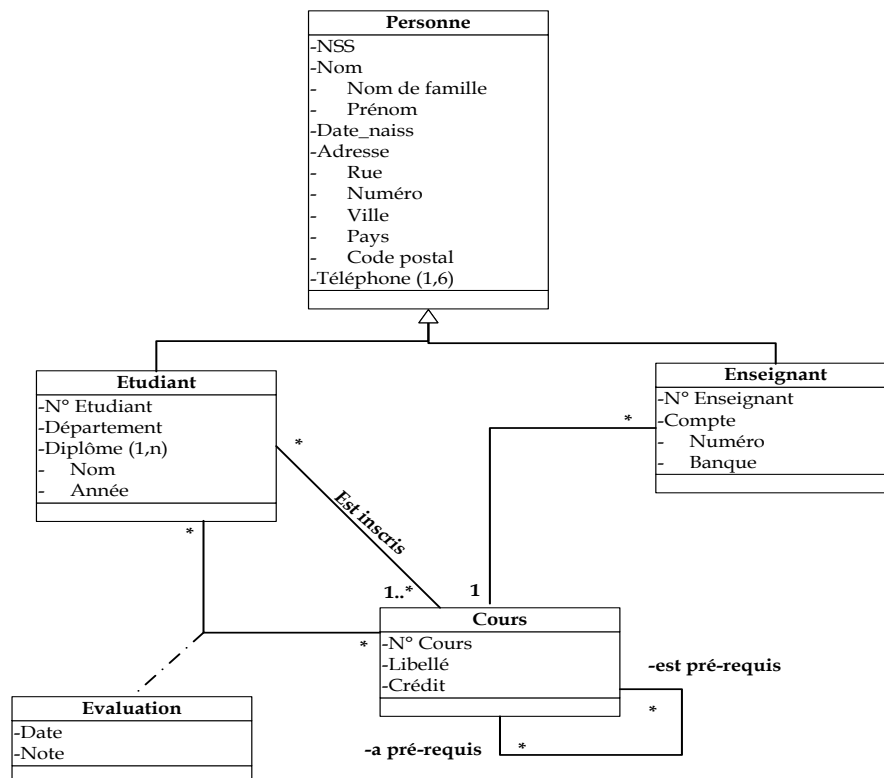
Personne
-NSS
-Nom
- Nom de famille
- Prénom
-Date_naiss
-Adresse
- Rue
- Numéro
- Ville
- Pays
- Code postal
-Téléphone (1,6)

- Créer tous les types nécessaires pour la création de type TPersonne.
  - Définir la méthode permettant de connaître l'âge de la personne.
  - Créer la table personne.
  - Ajouter une nouvelle personne avec NSS 123456789, le nom Mohamed ADIMI, né le 01/01/1980 et vivant au 22 rue de la Gare, 1600 Alger, Algérie, et ses téléphones sont 023112345 et 0554321921.
2. On hérite de la classe personne deux sous classes comme suit:



- Définir les deux types TEtudiant et TEnseignant.
- Ajouter un nouvel étudiant avec NSS 123123123, le nom Adam MERABETI, né le 01/05/1985, vivre à Didouche Mourad, Alger, Algérie, il n'a pas de téléphone, numéro d'étudiant 999, département d'informatique, sans diplômes.

- c. Ajouter un nouvel enseignant avec NSS 666999666, avec le nom Meriem LAMARI, née le 04/06/1975, demeurant à 99 Boulevard Colonel Amirouche, Alger, Algérie, il n'a pas de téléphone, numéro d'employé 777, compte bancaire 310 1234567 89 chez BDA.
  - d. Ajouter un nouvel enseignant avec NSS 556978566, avec le nom Ahmed SALEMI, né le 04/06/1965, demeurant à 99 Ismail Yafsah, Alger, Algérie, pas de téléphone, numéro d'employé 787, compte bancaire 330 1234897 56 chez BEA.
  - e. Ajouter les téléphones 022342222 et 066543333 à l'étudiant Adam MERABETI.
  - f. Ajouter les mêmes téléphones de l'étudiant Adam MERABETI à Meriem LAMARI.
  - g. Lister les étudiants (nom et numéro).
  - h. Trouver les couples de noms de personnes partageant au moins un numéro de téléphone.
3. Soit le schéma final suivant:



- a. Définir tous les types TCours et TEvaluation. Prendre en compte toutes les associations et les répercussions sur les types déjà créés.
- b. Définir les tables Cours et Evaluation.

## Exercice 2 :

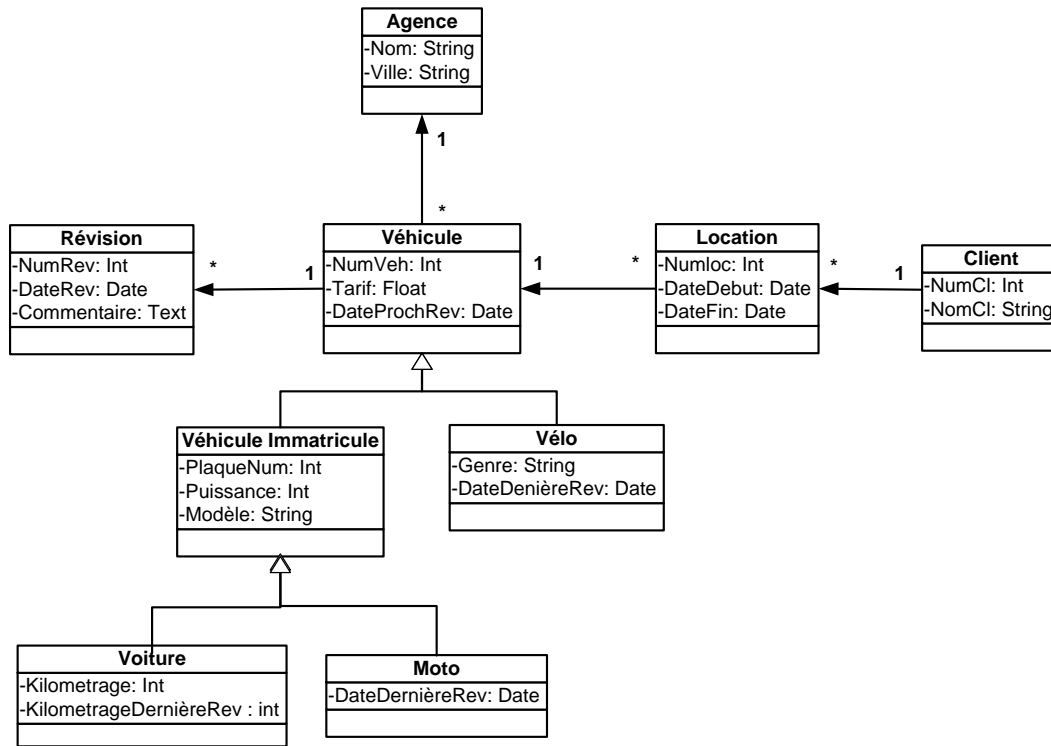
On considère le schéma SQL3 suivant décrivant des personnes qui suivent d'autres personnes et qui écrivent des messages. On connaît la date depuis laquelle une personne suit une autre personne. Pour un message, on connaît son texte, sa date de création et les tags (un ensemble de mots-clés) qui catégorisent le message.

<pre> create type T_Personne ; / create type T_SET_Tag as table of Varchar2(30); / create type T_Message as object ( texte varchar2(500), dateEcrit Date, tags T_SET_Tag ); / create type T_SET_Message as table of T_Message; / </pre>	<pre> create type T_Contact as object( p ref T_Personne, depuis Date ) / create type T_SET_Contact as table of T_Contact; / Create or Replace type T_Personne as object ( prenom varchar2(30), suit T_SET_Contact, ecrit T_SET_Message ); / </pre>
---	--

1. Donner le script de création de la table Personne définie sur le type T\_Personne.
2. La table Personne contient les personnes Amel et Badis ; Ecrire l'instruction SQL3 permettant d'insérer dans la table Personne la personne suivante : Manel suit Amel depuis le 01/01/2017 et a écrit le message 'Oran candidat aux jeux Méditerranéens 2022' concernant le tag 'JM2022' le 01/06/2018
3. Ecrire l'instruction SQL3 qui insère Badis dans les contacts de Manel le 01/06/2018.
4. Ecrire en SQL3 les requêtes suivantes :
  - a. Quels sont les prénoms des personnes que Amel suit depuis le 01/06/2018 ou une date antérieure et qui ont écrit un message contenant le tag 'JM2022' ? Le résultat ne contient pas de doublons. La solution ne doit pas contenir de sous-requête.
  - b. Quelles sont les personnes qui ont écrit un message le même jour qu'Amel. Afficher des couples formés d'une date et d'un objet personne. Trier le résultat par date croissante.
  - c. Pour chaque tag, combien de personnes ont écrit au moins un message concernant ce tag ? Afficher des couples formés d'un tag et d'un nombre de personnes. Utiliser la clause group by.
5. On complète le type T\_Personne avec la méthode mesTags retournant l'ensemble (sans doublons) des tags des messages d'une personne. La signature de la méthode est: **member function mesTags return T\_Set\_Tag;** Par exemple Badis a écrit un message ayant le tag 'Judo' et un autre message ayant les tags 'Karaté' et 'Judo'. Alors la méthode mesTags() invoquée sur Badis affiche \_SE\_Tag('Karaté', 'Judo') . Ecrire le corps de la méthode.

### Exercice 3 :

On considère le diagramme de classes suivant, décrivant une base de données modélisant un système de location de véhicules.



On veut modéliser ce schéma en SQL3. On décrit les associations dans le sens des flèches (ex : Véhicule → Agence, c-à-d qu'on ne sauvegarde dans la BD que le fait qu'un véhicule est géré par une agence).

1. Définir les types **T\_Agence**, **T\_Revision**, **T\_Vehicule**, **T\_VehiculeImmatricule** et **T\_Voiture**.
2. Définir les types **T\_Location** et **T\_Client**.
3. Définir les tables **Agences** stockant les objets de type T\_Agences, **Vehicules** stockant les objets de type T\_Vehicule, **Locations** stockant les objets de type T\_Location et **Clients** stockant les objets de type T\_Client.
4. La table Agences contient l'agence « Azimut Car à Bordj El Kiffan » ; Ecrire l'instruction SQL3 permettant d'insérer la voiture de numéro 111, tarif de location est 3500 DA, immatriculée 1245610816, de modèle PEUGEOT NEW 208, de puissance 136 chevaux et son kilométrage est à 20000 KM pour l'agence « Azimut Car ».
5. La table Clients contient le client « 102, Ahmed MESSAOUDI ». Ce client a loué la voiture N° 111 pour la période de 04/09/2019 à 09/09/2019. Ecrire l'instruction SQL3 permettant d'ajouter cette location de N° 1058 pour ce client.
6. Ecrire en SQL3 les requêtes suivantes :
  - a. Quelles sont les agences (nom) qui loue des voitures de modèle PEUGEOT NEW 208?
  - b. Quelles sont les clients qui ont loué au moins une fois le même véhicule. Afficher des couples formés des noms des clients. Quels sont les clients qui ont loué tous les véhicules de l'agence « Azimut Car » ?

7. On veut ajouter à ce schéma la méthode **Réviser** qui met à jour la date de la prochaine révision (attribut DateProchRev) à effectuer sur un véhicule. Les voitures doivent être révisées tous les 20000km, les motos tous les 6 mois et les vélos une fois par an. Définir la méthode **Réviser** (Signature et Corps) pour chaque type.

#### Exercice 4 : devoir à remettre

Soit la base de données relationnelle suivante, modélisant les livres (les clés des relations sont soulignées):

Relation	Description
LIVRE ( <u>NoISBN</u> , Titre, Année, Langue, NumEditeur)	NoISBN: numéro du livre Titre : titre du livre Année : année de publication du livre Langue : langue de publication du livre : français, anglais ou espagnol NumEditeur : numéro de la maison d'édition où a été publié ce livre
AUTEUR ( <u>NoAuteur</u> , Nom, Prénom)	NoAuteur: numéro de l'auteur Nom : nom de l'auteur Prénom : premier prénom de l'auteur
COLLABORATION ( <u>NoISBN</u> , <u>NoAuteur</u> , Rang)	NoISBN: numéro du livre NoAuteur: numéro de l'auteur Rang : rang de l'auteur. Cette relation décrit les collaborations des auteurs à la rédaction d'un livre ainsi que le rang de l'auteur dans la liste des auteurs du livre (1er auteur, 2ème auteur, etc...).
EDITEUR ( <u>NumEditeur</u> , NomEdit, Pays)	NumEditeur: numéro de l'éditeur NomEdit : nom de l'éditeur Pays : pays de l'éditeur

- Transformer ce schéma relationnel en un diagramme de classes.
- On souhaite stocker ce diagramme de classes dans un SGBD relationnel objet en utilisant le langage de définition de données SQL3 :
  - Définir tous les types nécessaires. Prendre en compte toutes associations qui existent
  - Définir les tables nécessaires à votre base de données
- La base de données contient les auteurs: David Fayon et Michaël Tartar et l'éditeur Pearson. Ecrire l'instruction SQL3 permettant d'insérer le livre de titre « Transformation digitale 2.0 », ISBN : 978-2-7440-6709-9 et édité par l'éditeur Pearson en juin 2019. Ce livre est écrit par David Fayon premier auteur et Michaël Tartar
- On souhaite ajouter dans la base de données une liste de numéros de téléphone pour chaque auteur. Chaque numéro de téléphone a un libellé indiquant la nature du numéro (mobile, bureau, maison). Ecrire les requêtes permettant de stocker ces informations dans la base de données
- On complète le type auteur avec la méthode **MesLivres** retournant l'ensemble (sans doublons) des livres où l'auteur est de rang égale à 1. Donner la signature et le corps de cette méthode
- Ecrire en SQL3 les requêtes suivantes :
  - Quels sont les auteurs qui ont participé dans l'écriture de même livre. Afficher des couples formés de numéro de livre et d'un objet auteur.
  - Quels sont les auteurs qui éditent chez tous les éditeurs.