

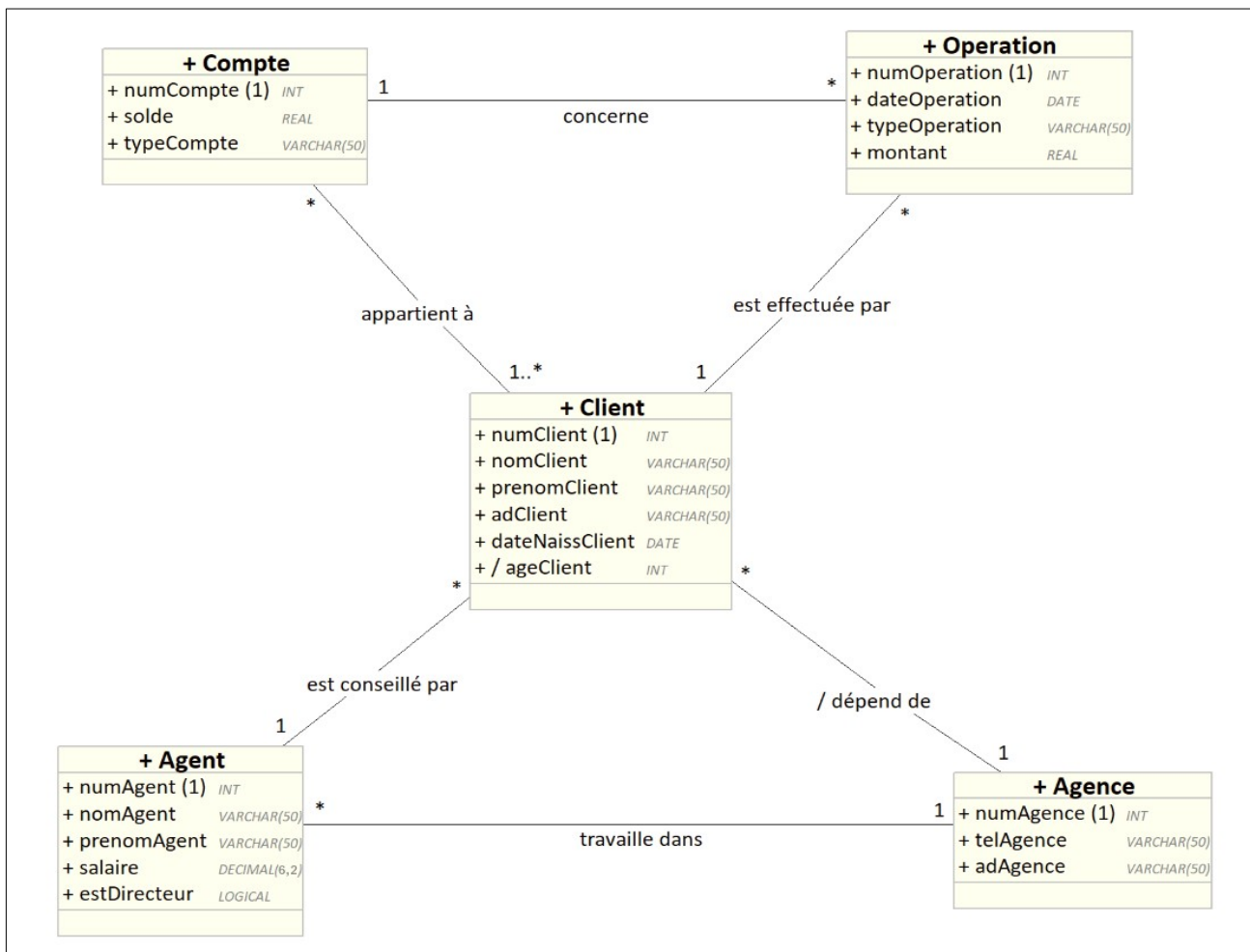
## R3.07 - SQL dans un langage de programmation 2023/2024

### TD/TP 1 : Banque

**Note:** L'objectif de ce TP est de vous offrir l'opportunité de réviser les requêtes SQL vues l'année dernière et de créer un premier bloc PL/SQL.

Assurez-vous de regrouper les scripts à la fin du TP sous forme d'un fichier compressé (.zip). Vous devez soumettre votre travail via l'espace de rendu dédié au TP sur Moodle, correspondant à votre groupe.

Considérons le diagramme de classes UML suivant complété par des contraintes textuelles :



### Contraintes textuelles :

- Le montant d'une opération est toujours positif
- Le type d'opération est soit **RETRAIT**, soit **DEPOT**
- Le type de compte considéré est soit **COURANT**, soit **EPARGNE**
- Par défaut, la date d'opération est la date du jour courant

- Aucun salaire ne doit être inférieur au Smic (1709.28 euros brut)
- Une augmentation de salaire ne doit pas dépasser 10% et une baisse 8%
- Un client ne peut pas être conseillé par un agent portant le même nom que lui
- Un client ne doit pas pouvoir effectuer un retrait dont le montant est supérieur au solde
- Un client ne doit pouvoir retirer de l'argent que sur un compte qui lui appartient
- Une agence a forcément un et un seul directeur
- Le directeur d'une agence est mieux payé que les agents de son agence

### Tables et vues:

1. Transformer le diagramme de classes UML en schéma relationnel
2. Écrire le script `tables.sql` permettant de créer les tables en gérant le maximum de contraintes. Justifier pourquoi on ne peut pas (pour l'instant) traiter certaines contraintes
3. Écrire le script `vues.sql` permettant de créer :
  - a) la vue fournissant tous les éléments dérivables (attribut et association)
  - b) la vue détectant les éventuels défauts de surjectivité

### Requêtes :

Écrire le script `requetes.sql` permettant de répondre aux questions suivantes :

4. Donner la liste des clients ayant effectué des retraits d'un montant supérieur à 1000 euros (Roux, Leclerc)
5. Qui sont les clients n'ayant pas de compte auprès de l'agence gérée par Monsieur Dupont ? (Blanc, Caron, Fernandez, Fournier, Garnier, Guillaume, Leclerc, Leconte, Lefevre, Legrand, Leroy, Marie, Muller, Noël, Roux, Roy)
6. Lister les agences avec le nombre total d'agents, même celles qui n'ont pas d'agents (1=>3, 2=>3, 3=>4, 4=>0)
7. Quel est le directeur le mieux payé ? (Dupont)
8. Pour chaque directeur, indiquer (ordre décroissant) le nombre d'agents (Morel => 3, Dupont => 2, Fontaine => 2, Dubois => 2)
9. Quels sont les agents qui gèrent le plus grand nombre de comptes épargnes ? (Lefevre => 4)
10. Lister les 5 derniers retraits du plus récent au plus ancien (13, 5, 6, 1, 8)
11. Pour chaque agence, indiquer le nombre de clients ne possédant pas encore de compte épargne (1 => 3, 2 => 4, 3 => 6, 4 => 0)
12. Quels sont les clients ayant effectué des opérations sur tous types de comptes ? (Roux, Roy)

Utiliser le script `remplissage.sql`, disponible sur Moodle, pour remplir vos tables et tester vos requêtes

## **PL/SQL:**

13. Dans un script `solde.sql`, créer un bloc **PL/SQL** interactif qui permet à l'utilisateur de saisir le numéro d'un client. Ce bloc doit extraire le nom complet du client correspondant et calculer le solde total de tous ses comptes sans utiliser explicitement une boucle **FOR**. À la fin, le bloc doit afficher le nom complet du client ainsi que le solde total de ses comptes