



## TP2: UML et modèle Objet-Relationnel

### Partie 1

#### (Cahier de charge)

La base de données étudiée concerne la gestion des comptes bancaires des clients d'une agence.

Un client d'une banque est caractérisé par (num, nom, adresse). Il faudra pouvoir stocker au plus trois numéros de téléphone (numTel) correspondant respectivement au numéro du domicile, du travail et du portable.

Un compte est rattaché à un client. Un client peut disposer de plusieurs comptes qu'ils soient compte épargne ou compte courant. Un compte est caractérisé par son numéro, son solde et sa date d'ouverture (nCompte, solde, dateOuv).

- Un compte épargne est rémunéré à un certain taux d'intérêt (txInt).
- Un compte courant se caractérise par :
  - o Un nombre d'opérations de carte bleu (nbOpCB) ;
  - o L'existence d'un ou de plusieurs signataires (qui sont aussi des clients de l'agence) qui peuvent disposer de différents droits (droit) sur le compte courant (D pour débiter le compte, X pour clôturer le compte, R pour retirer un chéquier au guichet).

On désire stocker les opérations (date et montant) réalisées par un client (qu'il soit signataire ou pas) sur un compte courant (dateOp, montant). Quand le montant est positif il s'agira d'un dépôt, quand il est négatif il s'agira d'un retrait.

### Partie 2

#### (Préparation de l'environnement de travail)

- Proposer un diagramme de classes correspondant au cahier de charge (Partie 1).
- Transformer le diagramme de classes obtenu en un schéma objet-relationnel correspondant.

### Partie 3

#### (Schéma Oracle Objet)

Ecrivez le script de création des types et tables objet. Je vous conseille de dessiner à la main la structure des tables pour mieux visualiser l'emplacement des collections et des références au sein du graphe d'héritage.

#### Types des colonnes:

num	NUMBER(5)	dateOuv	DATE
nom	VARCHAR2(30)	TxInt	NUMBER(2,1)
Adresse	VARCHAR2(30)	nbOpCB	NUMBER(5)
numTel	VARCHAR2(14)	Droit	CHAR(1)
nCompte	VARCHAR2(5)	dateOp	DATE
Solde	NUMBER(10,2)	montant	NUMBER(8,2)

#### Indications :

Association	Traduire l'association 1-n « Possède » à l'aide d'une collection <i>varray</i>
	Traduire l'association 1-n « Propriétaire » à l'aide d'une <i>référence</i>
	Traduire l'association 1-n « Signataires » à l'aide d'une collection <i>nested table</i>
	Traduire l'association n-aire « Opérations » à l'aide d'une table de liaison appelée « Mouvements » contenant deux références (une vers le type d'un client, une autre vers le type d'un compte courant).

Contraintes	En plus des clés primaires sur chaque table, programmez les contraintes suivantes :
	<ul style="list-style-type: none"> <li>▪ REFERENCES et NOT NULL sur chaque référence ;</li> <li>▪ Non nullité du couple (num, droit) pour un compte courant donné ;</li> <li>▪ La colonne nbOpCB est positive ;</li> <li>▪ La colonne droit vaut (D, X ou R) ;</li> <li>▪ La colonne dateOp vaut par défaut SYSDATE-2 (rappelez-vous les dates de valeur...).</li> <li>▪ La colonne txInt doit toujours être inférieure à 3,5%</li> </ul>

### Partie 4

#### (Manipulation des données)

Ecrivez le script SQL qui permettra d'insérer :

- Des clients ayant divers numéros de téléphone (un client sans téléphone, un client ayant seulement un numéro de téléphone portable, un autre client ayant à la fois un portable et un téléphone fixe, etc.).



- Quelques comptes courants et d'épargne (soldes et dates d'ouverture initialisés avec des valeurs) ;
- Des clients ayant plusieurs comptes courants ;
- Un client n'ayant qu'un seul compte épargne ;
- Un client n'ayant qu'un seul compte courant ;
- Un client sans compte ;
- Des comptes courants ayant un ou plusieurs signataires titulaires de différents droits ;
- Des comptes courants sans signataire ;
- Des clients ayant plusieurs comptes épargne avec différents taux d'intérêt ;
- Des clients ayant plusieurs comptes courants réalisés par des clients propriétaires de ces comptes et par les signataires.

Un jeu de données sera fourni par votre encadrant pour que toute la classe aura la même base de données. [[compte.sql](#)]

## Partie 5 (Manipulation de la base)

- a. Ajout le client 99, *Paturel, Port - Royal – Paris*, téléphone domicile 04-47-56-98-16, ( initialisez le **varray** à un seul élément).
- b. Créez un nouveau compte courant CC99, de solde 0 sans signataire et opération CB, à la date du jour (SYSDATE) associé au client.
- c. Associer ce nouveau compte courant au client 99. Vérifier.
- d. Effectuez un mouvement de 50€ sur ce compte par ce client, en ne renseignant pas la date d'opération. [Vérifier l'insertion et date du jour]

## Partie 6 (Requête SQL : Simple)

1. Liste des clients sans leur numéro de téléphone.
2. Liste des comptes courant (numéro, solde, date d'ouverture, nombre d'opération CB).
3. Liste des comptes épargne (numéro, solde, date d'ouverture, taux d'intérêt).

## Partie 7 (Requête SQL : Références)

4. Liste des comptes courant du client de numéro 4 (jointure implicite dans le WHERE ).

5. Même requête pour tous ses comptes (courant et épargne).
6. Liste des clients et numéro des comptes courants de solde inférieur à 400€ donné (jointures implicites dans le SELECT ).
7. Numéro, nom et adresse des clients titulaires d'au moins un compte épargne.
8. Numéro, nom et adresse des clients titulaires d'un seul compte épargne.
9. Pour chaque client (numéro, nom), afficher le nombre de compte épargne qu'il possède.
10. Même requête pour afficher aussi les clients n'ayant pas de compte épargne.
11. Numéro, nom et nombre de compte épargne du titulaire ayant le plus de comptes épargne.
12. Liste des clients (numéro et nom) qui ont fait des opérations (date et montant) sur le compte courant de numéro CC7.
13. Liste des opérations sur le compte courant CC7 (numéro de compte, date et montant) qui ont été faites par un client propriétaire du compte modifié (jointures dans le SELECT et dans le WHERE ).
14. Même requête pour les opérations qui ont été faites par les clients non propriétaire du compte modifié.

## Partie 8 (Requête SQL : Collections)

15. Nombre de téléphones d'un client donné (première écriture de l'opérateur TABLE).
16. Même requête pour chaque client (deuxième écriture de l'opérateur TABLE).
17. Nombre de signataire d'un compte courant donné (première écriture de l'opérateur TABLE).
18. Même requête pour chaque compte courant (deuxième écriture de l'opérateur TABLE).
19. Numéros de téléphone d'un client donné (première écriture de l'opérateur TABLE).
20. Même requête pour chaque client (deuxième écriture de l'opérateur TABLE).
21. Numéro de client et droits correspondants aux signataires d'un compte courant donné (première écriture de l'opérateur TABLE ).
22. Même requête pour tous les comptes courant (deuxième écriture de l'opérateur TABLE).
23. Même requête avec un *nested cursor*.
24. Numéro de client et adresse correspondant aux signataires d'un compte courant donné (première écriture de l'opérateur TABLE ).

Données

num	nom	adresse	Telephone_vry
1	Albaric	Pont Vieux - Vielle Toulouse	05-61-75-68-39
			Null
2	Bidal	Port Royal - Paris	Null
			Null
			06-76-85-14-89
3	Miranda	Antipolis - Nice	Null
			04-35-60-77-89
			06-81-94-44-31
4	Payrissat	Salas - Ramonville St Agne	05-61-75-98-44
			Null
			06-46-45-72-30
5	Vielle	INRA - Auzeville Tolosane	05-61-73-12-74
			05-62-74-75-63
			06-65-41-83-35

ncpt	solde	dateOuv	Ref_client	nbOpB	Signataire_nt	
					num	droit
CC1	4030	01-02-2001	@1	509		
CC2	3000	15-02-2002	@1	0	2	D
					2	R
					3	R
CC3	460	13-05-2000	@4	678	1	D
					5	D
CC4	730	17-09-2002	@4	0		
CC5	15	10-12-1998	@4	1390		
CC6	55	16-01-1965	@2	2400	3	D
CC7	6700	04-03-1976	@3	5600	2	D
					2	R
					2	X
					1	D
					5	D

ncpt	solde	dateOuv	Ref_client	txInt
CE1	600	05-02-1965	@2	2.7
CE2	4500	04-12-1998	@2	2.9
CE3	500	05-03-2000	@4	2.9
CE4	500	05-02-2011	@4	2.4
CE5	500	13-05-1995	@4	3.4
CE6	3300	23-08-1997	@3	3.3

Ref-client	Ref_cptcour	Date_op	montant
@1	@CC1	SYSDATE-7	+100
@1	@CC1	SYSDATE-5	-60
@1	@CC1	SYSDATE-5	+40
@2	@CC1	SYSDATE-3	-80
@2	@CC1	SYSDATE-3	-50
@2	@CC6	SYSDATE-7	+30
@2	@CC6	SYSDATE-7	-15
@3	@CC6	SYSDATE-5	-20
@3	@CC7	SYSDATE-7	+300
@3	@CC7	SYSDATE-6	-105
@1	@CC7	SYSDATE-5	-20
@1	@CC7	SYSDATE-4	-10
@5	@CC7	SYSDATE-5	-60
@4	@CC4	SYSDATE-2	-10
@4	@CC4	SYSDATE-2	-70
@4	@CC5	SYSDATE-5	+300
@4	@CC3	SYSDATE-1	-50
@1	@CC3	SYSDATE-2	-70

**Partie 9 (PL\_SQL)**

- 1- Numéro de téléphone du travail du client 3.
- 2- Utiliser un curseur pour afficher le numéro du téléphone du travail de tous les clients titulaires d'un compte courant.

**Partie 10 (Méthodes)**

- 1- Méthode *nbCepargne* qui renvoie le nombre de compte épargne d'un client donné.
  - a. Modifier le type *client\_type* en ajoutant la méthode *nbCepargne*  
**Utiliser :** *ALTER TYPE client\_type*  
*ADD MEMEBRE FUNCTION nbCepargne RETURN NUMBER*  
*CASCADE;*
  - b. Implémenter la méthode *nbCepargne*  
*CREATE OR REPLACE TYPE BODY client\_type*  
*AS MEMBER FUNCTION nbCepargne RETURN NUMBER IS*  
*....*  
*....*  
*..*  
*END;*  
*/*
  - c. Tester cette méthode dans une requête et dans un bloc PL/SQL pour le client numéro 4.

```

DECLARE
un_Client client_type;
BEGIN
SELECT VALUE(cli) INTO un_Client
FROM Client cli WHERE cli.num = 4;
DBMS_OUTPUT.PUT_LINE('Le client numéro 4 possède '|| un_Client.nbCepargne()
|| ' compte(s) épargne(s)');
END;
/

```

- 2- Méthode *nbSignataire* qui renvoie le nombre de signataire d'un compte courant donné.
  - a. Même syntaxe et mêmes démarches
  - b. Tester cette méthode dans une requête et dans un bloc PL/SQL pour le compte CC7.

- 3- Méthode *nbSignataire*(droitparam IN CHAR) qui ne tient compte que les signataires de droit passé en paramètre.
  - a. Tester cette méthode dans une requête pour chaque droit de chaque compte.
- 4- Méthode *calculeInteret* qui renvoie la somme augmentée des intérêts d'un compte épargne donné.
  - a. Tester cette méthode pour le compte épargne CE5. Avant d'exécuter cette méthode, consulter la table.
- 5- Méthode booléenne *estTitulaire*(cli IN NUMBER) qui renvoie TRUE si le client de numéro passé en paramètre est titulaire du compte courant donné.
  - a. Tester cette méthode dans un bloc en cherchant le titulaire du compte CC4.
- 6- Méthode booléenne *estSignataire*(cli IN NUMBER, droitparam IN CHAR) qui renvoie TRUE si le client de numéro passé en paramètre est signataire du droit passé en paramètre pour un compte courant donné.
  - a. Tester cette méthode dans un bloc en cherchant des signataires au droit D du compte CC7.