

Objet-Relationnel et JDBC

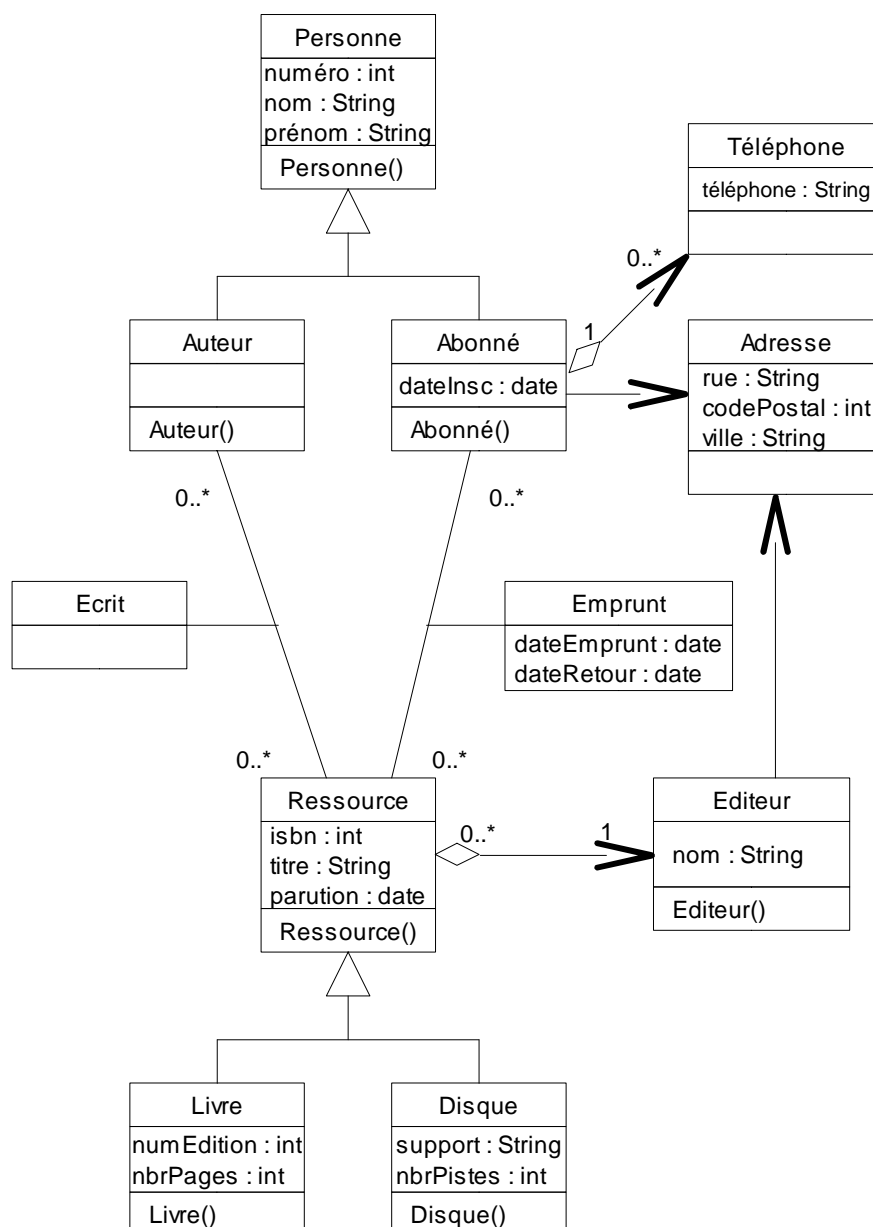
L'objet de cette section est la modélisation d'une base de données élémentaire de gestion d'emprunts de ressources par des abonnés. La description de la base de données vous est donnée dans la figure 1.

Figure 1 : Description de la base de données

Les ressources empruntées peuvent être des disques ou des livres; toutes sont identifiées par leur numéro ISBN et, possèdent un titre et une date de parution. Chaque ressource est éditée par un éditeur, dont on stocke le nom et l'adresse, et est écrite par un ou plusieurs auteurs. Les livres ont un numéro d'édition et un nombre de pages. Les disques peuvent être de support « vinyle », « CD » ou « DVD » et possèdent un nombre de pistes. On stocke pour chaque abonné et chaque auteur son numéro identifiant, son nom et son prénom. On mémorise également la date d'inscription des abonnés, ainsi que leur(s) numéro(s) de téléphone et leur adresse postale afin de pouvoir les contacter. Lorsqu'un abonné emprunte une ressource, on mémorise la date d'emprunt et la date de restitution. Les emprunts d'une même ressource par un même abonné sont distingués par la date d'emprunt.

Le schéma UML de la base de données est représenté dans la figure 2.

Figure 2 : Schéma UML



Exercice 1. On souhaite dans un premier temps stocker les données en utilisant un SGBD Objet-Relationnel. Ecrivez les requêtes SQL3 de création des tables de la base de données Objet-Relationnelle correspondant aux objets Abonné, Livre, Disque et Emprunt du schéma UML. Vous utiliserez le type d'objet SQL3 qui vous paraît le plus adapté pour la représentation de chaque élément. [7 pts]

```
CREATE TYPE t_telephones AS VARRAY(3) OF VARCHAR(10);

CREATE ROW TYPE t_adresse (rue VARCHAR(30),
                           cp NUMERIC(5,0),
                           ville VARCHAR(20));

CREATE TYPE t_personne AS OBJECT (numero INTEGER,
                                  nom VARCHAR(20),
                                  prenom VARCHAR(20)) NOT FINAL;

CREATE TYPE t_abonne UNDER t_personne (dateinsc DATE,
                                         adrabo t_adresse,
                                         telephones t_telephones,

                                         adresse REF t_adresse,
                                         telephones SET(t_telephones));

CREATE TYPE t_ressource AS OBJECT (isbn INTEGER,
                                   titre VARCHAR(50),
                                   parution DATE) NOT FINAL;

CREATE TYPE t_livre UNDER t_ressource (numedition INTEGER
                                       nbrpages INTEGER);

CREATE TYPE t_disque UNDER t_ressource (support VARCHAR(7),
                                       nbrpistes INTEGER);

CREATE TYPE t_emprunt AS OBJECT (abonne REF t_abonne,
                                 ressource REF t_ressource,
                                 dateemprunt DATE,
                                 dateretour DATE);

CREATE TABLE abonne OF t_abonne (PRIMARY KEY (numabo));

CREATE TABLE livre OF t_livre (PRIMARY KEY (isbn));

CREATE TABLE disque OF t_disque (PRIMARY KEY (isbn));

CREATE TABLE emprunt OF t_emprunt (
    PRIMARY KEY (abonne.numabo, ressource.isbn, dateemprunt));
```

Pour le fun, le reste :

```
CREATE TYPE t_auteur UNDER personne();

CREATE TYPE t_ecrit AS OBJECT (numaut INTEGER,
                              isbn INTEGER);

CREATE TYPE t_editeur AS OBJECT (nomedit VARCHAR(20),
                                adredit t_adresse);
```

Avec un type pour l'objet numéro de téléphone :

```
CREATE TYPE t_telephone AS OBJECT (numtel VARCHAR(10));

CREATE TYPE t_telephones AS VARRAY(3) OF t_telephone;
```

Exercice 2. Pour chacune des tables correspondant aux objets Abonné, Livre, Disque et Emprunt de votre

base de données, donnez un exemple de requête SQL3 d'insertion d'une instance. On suppose que les tuples correspondants aux auteurs et éditeurs du livre et du disque que vous insérez sont déjà enregistrés dans la base de données. [2 pts]

```
INSERT INTO abonne VALUES
(101, 'Dupond', 'Jean', '01-01-2006', 'Rue Minant', 06001, 'NICE',
'0493000001', '0493000002','0493000003');
INSERT INTO livre(isbn, titre, parution , numedition, nbrpages)
;
INSERT INTO disque(isbn, titre, parution , support, nbrpiste)
;
INSERT INTO emprunt (numAbo, isbn, dateemprunt, dateretour)
;
```

Exercice 3. On souhaite maintenant proposer une solution pour stocker cette base de données dans un SGBD Relationnel et non pas Objet-Relationnel. Proposez un schéma relationnel pour cette base de données. [4 pts]

Auteur (NumAut, NomAut, PrénomAut)
Abonné (NumAbo, NomAbo, PrénomAbo, DateInsc, RueAbo, CPAbo, VilleAbo)
Téléphone (NumAbonné, Téléphone)
Ressource (ISBN, Titre, Parution)
Livre (ISBN, NumEdition, NbrPages)
Disque (ISBN, Support, NbrPistes)
Editeur (NomEdit, RueEdit, CPEdit, VilleEdit)
Emprunt (ISBN, NumAbo, DateEmprunt, DateRetour, NomEdit)
Ecrit (ISBN, NumAut)

Exercice 4. En supposant que les données sont stockées dans une base de données Relationnelle, écrivez un programme JDBC permettant de rechercher les livres d'un auteur donné qui ont été empruntés par un abonné résidant dans une ville donnée. L'auteur et la ville de résidence des abonnés seront saisis par l'utilisateur lors de l'exécution. Le résultat doit être affiché à l'écran. Votre programme doit être aussi portable que possible, c'est à dire qu'il doit être le plus facilement possible adaptable à divers SGBD Relationnels. [7 pts]

```
import java.sql.*;
import unsa.*;
public class Test1
{
    public static void main(String[] args)
    throws SQLException, ClassNotFoundException, java.io.IOException
    {
        Connection conn = null;
        PreparedStatement pstmt1 = null;
        PreparedStatement pstmt2 = null;
        Statement stmt = null;
        String query = "SELECT ressource.*, livre.*" +
            "FROM livre, ecrit, emprunt, abonne, ressource " +
            "where ecrit.numaut = ? " +
            "and villeabo= ? " +
            "and ressource.isbn = livre.isbn " +
            "and ecrit.isbn = livre.isbn " +
            "and emprunt.isbn = livre.isbn " +
            "and emprunt.numabo= abonne.numabo";

        try
        {
            conn = ConfigConnection.getConnection("fichierProp");
            pstmt = conn.prepareStatement();
```

```

int numero = Console.in.readInt();
String ville = Console.in.readLine();
pstmt.setInt(1, numero);
pstmt.setInt(2, ville);
ResultSet rset = pstmt.executeQuery(query);
while (rset.next())
{
    System.out.println("ISBN : " + rset.getString(1) +
                        "Titre : " + rset.getString(2) +
                        "Parution : " + rset.getDate(3) +
                        "NumEdition : " + rset.getInt(5) +
                        "NbrPages : " + rset.getInt(6));
}
}

finally
{
    if (stmt != null)
        stmt.close();
    if (conn != null)
        conn.close();
}
}

```