

TD&TP6 : A propos d'un club de Voile



Un club nautique utilise une base de données relationnelles pour la gestion de son système d'Information. Ce club nautique dispose d'un terrain qui a été subdivisé en emplacements occupés par des bateaux que possèdent les adhérents du club. Plus précisément, nous nous intéressons ici aux notions ci-après :

Les emplacements : les emplacements sont identifiés par un numéro. Pour chaque emplacement, on connaît sa longueur qui fixe la longueur maximale des bateaux qui peuvent l'occuper. On place un seul bateau par emplacement. Un emplacement peut être de type *petit* (de 0m à 7m), *moyen* (entre 7m et 12m) ou *grand* (entre 12m et 20m). Cette caractéristique détermine le prix annuel d'occupation de l'emplacement.

Les bateaux : les bateaux stockés au club, sont identifiés par un numéro (entier positif). Chaque bateau est d'un certain modèle. Le modèle d'un bateau est identifié par son nom. Plusieurs bateaux peuvent être du même modèle, ils sont alors tous de la même longueur. Chaque bateau est affecté à un emplacement dont la longueur est compatible avec la longueur du bateau. Un bateau a une date de construction et il est la propriété d'un ou plusieurs adhérents.

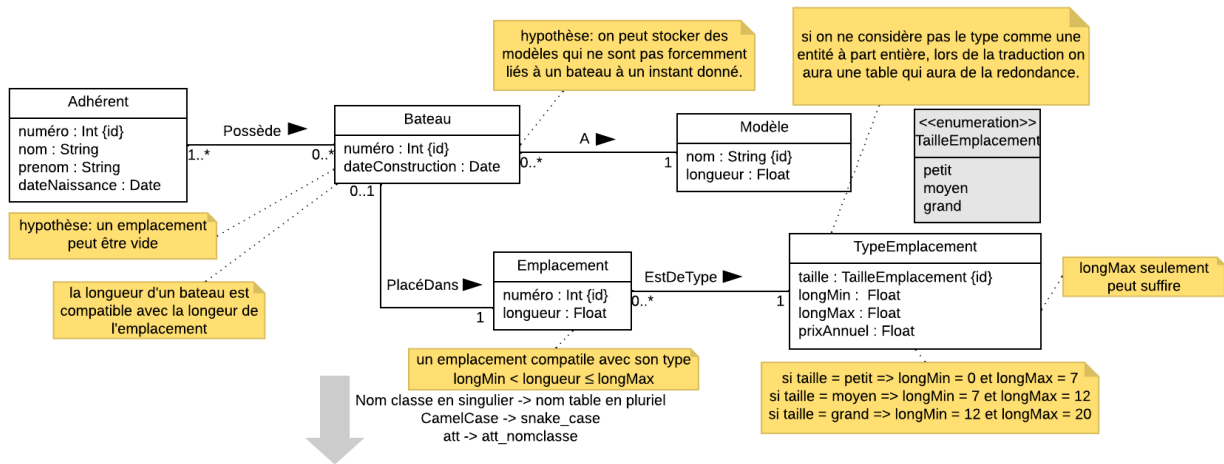
Les adhérents : les adhérents sont identifiés par leur numéro d'adhérent, on représente aussi leur nom, leur prénom et date de naissance. Un adhérent peut ne pas être propriétaire d'un bateau. Un adhérent peut être propriétaire de plusieurs bateaux.

1 Modélisation UML et passage au relationnel

Question 1 :

Proposer une modélisation UML de l'énoncé. Utiliser les commentaires UML pour identifier des possibles contraintes qui ne sont pas capturés par le modèle.

Solution UML. Notez qu'une bonne modélisation UML donne comme résultat un minimum de 3ème FN lors de la transformation car on isole les attributs qui sont liés en forme de classes.



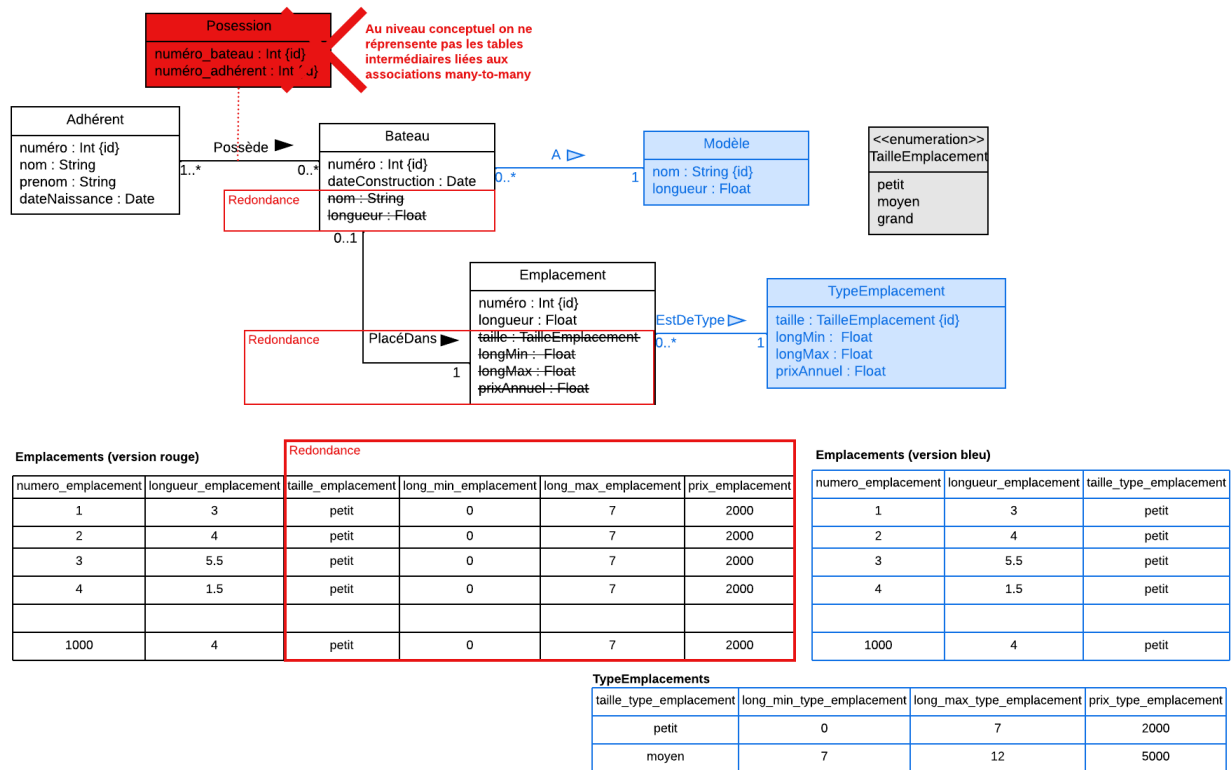
```

Adherents (numero_adherent, nom_adherent, prenom_adherent, date_naissance_adherent)
Modeles (nom_modele, longueur_modele)
Bateaux (numero_bateau, date_construction_bateau, nom_modele, numero_emplacement) /* nom_modele ne peut pas être absent (à cause du 1). C'est une clé aussi à cause de 0..1*/
Proprietaires (numero_adherent, numero_bateau) /* suite à l'association many-to-many */
Emplacements (numero_emplacement, longueur_emplacement, taille_type_emplacement) /* type_emplacement ne peut pas être absent (à cause du 1) */
TypesEmplacements (taille_type_emplacement, long_min_type_emplacement, long_max_type_emplacement, prix_annuel_type_emplacement)

Proprietaires[numero_adherent] ≤ Adherents[numero_adherent] /* inclut ou égal car cardinalité 0..* côté adhérent (un adhérent peut ne pas être propriétaire) */
Proprietaires[numero_bateau] = Bateaux[numero_bateau] /* égal car cardinalité 1 côté bateau (tout bateau a un propriétaire) */
Bateaux[nom_modele] ≤ Modeles[nom_modele] /* sous-ensemble car cardinalité 0..* côté bateau (on peut avoir des modèles qui ne sont liés à aucun bateau) */
Bateaux[numero_emplacement] ≤ Emplacements[numero_emplacement]
Emplacements[taille_type_emplacement] ≤ TypesEmplacements[taille_type_emplacement]
taille_type_emplacement ∈ {petit, moyen, grand}
  
```

Quelques points de vigilance :

- Il faut être précis et rigoureux avec la syntaxe UML (des associations avec des flèches, les attributs avec des minuscules avant, oubli des noms...).
- Pas de classe propriétaires en UML (cela est donné lors qu'on transforme une association Many-to-Many en Relationnel)
- Il faut nommer les associations (verbe en majuscule en général), définir le type des attributs et indiquer les identifiants avec id. Les rôles sont optionnels (ex. on pourrait avoir un rôle propriétaires à côté de la classe Adhérent pour l'association Possède).
- Les FK ne sont pas représentés en UML. On les indique lors qu'on fait la transformation en ajoutant une contrainte d'intégrité (voir corrigé)
- On cherche une cohérence entre les cardinalités et les contraintes d'intégrité :
 - Une cardinalité 1..* donnerait une égalité (ex. un bateau est toujours propriété de quelqu'un)
 - Une cardinalité 0..* donnerait un inclut ou égal (ex. si on n'a pas des bateaux de tous les modèles ou bien si on considère qu'un adhérent peut ne pas posséder des bateaux.).

**Question 2 :**

Réaliser le passage au modèle relationnel. Définissez les relations avec ces clés ainsi que toutes les contraintes.

Voir figure précédente

Question 3 :

Proposer le code SQL correspondant à l'implémentation du modèle relationnel.

```
-- ATTENTION : syntaxe Sqlite (IF EXISTS c'est propre à Sqlite)
DROP TABLE IF EXISTS Proprietaires;
DROP TABLE IF EXISTS Bateaux;
DROP TABLE IF EXISTS Emplacements;
DROP TABLE IF EXISTS Modeles;
DROP TABLE IF EXISTS TypesEmplacements;
DROP TABLE IF EXISTS Adherents;

-- Besoin du NOT NULL pour les PKs car Sqlite. Pas besoin dans d'autres SGBDs (par défaut NOT NULL)
CREATE TABLE Modeles (
  nom_modele TEXT PRIMARY KEY NOT NULL,
  longueur_modele DATE,
  CONSTRAINT CK_Bat_c0 CHECK (longueur_modele > 0)
);

CREATE TABLE TypesEmplacements (
  type_emplacement TEXT PRIMARY KEY NOT NULL,
  long_min_type_emplacement REAL NOT NULL,
  long_max_type_emplacement REAL NOT NULL,
  prix_type_emplacement REAL NOT NULL,
  CONSTRAINT CK_TEmp_c0 CHECK (longueur_max_type_emplacement > 0),
```

```

CONSTRAINT CK_TEmp_c1 CHECK (prix_type_emplacement > 0),
CONSTRAINT CK_TEmp_c3 CHECK (type_emplacement = 'petit' AND
long_min_type_emplacement=0 AND long_max_type_emplacement=7 OR
type_emplacement = 'moyen' AND
long_min_type_emplacement=7 AND long_max_type_emplacement=12 OR
type_emplacement = 'grand' AND
long_min_type_emplacement=12 AND long_max_type_emplacement=20)
);

```

```

CREATE TABLE Emplacements (
numero_emplacement INTEGER PRIMARY KEY NOT NULL,
longueur_emplacement REAL NOT NULL,
type_emplacement TEXT NOT NULL,
CONSTRAINT FK_Emp_c0 FOREIGN KEY (type_emplacement)
REFERENCES TypesEmplacements(type_emplacement),
CONSTRAINT CK_Emp_c1 CHECK (numero_emplacement > 0),
CONSTRAINT CK_Emp_c2 CHECK (longueur_emplacement > 0)
);

```

```

CREATE TABLE Bateaux (
numero_bateau INTEGER PRIMARY KEY NOT NULL,
date_construction_bateau DATE,
nom_modele TEXT NOT NULL,
numero_emplacement INTEGER UNIQUE NOT NULL,
CONSTRAINT FK_Bat_c0 FOREIGN KEY (nom_modele)
REFERENCES Modeles(nom_modele),
CONSTRAINT FK_Bat_c1 FOREIGN KEY (numero_emplacement)
REFERENCES Emplacements(numero_emplacement),
CONSTRAINT CK_Bat_c2 CHECK (numero_bateau > 0)
);

```

```

CREATE TABLE Adherents (
numero_adherent INTEGER PRIMARY KEY NOT NULL,
nom_adherent TEXT NOT NULL,
prenom_adherent TEXT NOT NULL,
date_naissance_adherent DATE,
CONSTRAINT CK_Adh_c0 CHECK (numero_adherent > 0)
);

```

```

CREATE TABLE Proprietaires (
numero_adherent INTEGER NOT NULL,
numero_bateau INTEGER NOT NULL,
CONSTRAINT PK_Prop_c0 PRIMARY KEY (numero_adherent, numero_bateau),
CONSTRAINT FK_Prop_c1 FOREIGN KEY (numero_adherent)
REFERENCES Adherents(numero_adherent),
CONSTRAINT FK_Prop_c2 FOREIGN KEY (numero_bateau)
REFERENCES Bateaux(numero_bateau)
);

```

-- Il aurait fallu des triggers pour vérifier la compatibilité de longueurs

Question 4 :

Proposer le code SQL correspondant à l'insertion d'au moins un bateau (et son emplacement) avec deux propriétaires, ainsi qu'un adhérent ayant au moins deux bateaux.

```

-- Jeux de données OK (ça doit marcher)
INSERT INTO Modeles VALUES ('Pirate', 12.5);

```

```
INSERT INTO Modeles VALUES ('Classique', 15);

INSERT INTO TypesEmplacements VALUES ('petit', 0, 7, 2000);
INSERT INTO TypesEmplacements VALUES ('moyen', 7, 12, 5000);
INSERT INTO TypesEmplacements VALUES ('grand', 12, 20, 10000);

INSERT INTO Emplacements VALUES (1001, 3, 'petit');
INSERT INTO Emplacements VALUES (1002, 4, 'petit');
INSERT INTO Emplacements VALUES (1003, 7.5, 'moyen');
INSERT INTO Emplacements VALUES (1004, 9, 'moyen');
INSERT INTO Emplacements VALUES (1005, 12, 'moyen');
INSERT INTO Emplacements VALUES (1006, 13, 'grand');
INSERT INTO Emplacements VALUES (1007, 20, 'grand');
INSERT INTO Emplacements VALUES (1008, 16, 'grand');

INSERT INTO Bateaux VALUES (1, '2013-04-13', 'Pirate', 1005);
INSERT INTO Bateaux VALUES (2, '2011-04-07', 'Pirate', 1006);
INSERT INTO Bateaux VALUES (3, '2017-04-07', 'Classique', 1007);

-- Date en format par défaut SQLite
INSERT INTO Adherents VALUES (10, 'Sparrow', 'Jack', '1983-04-10');
INSERT INTO Adherents VALUES (11, 'Barbossa', 'Hector', '1972-01-03');
INSERT INTO Adherents VALUES (12, 'Salazar', 'Armando', '1935-02-03');

INSERT INTO Proprietaires VALUES (10, 1);
INSERT INTO Proprietaires VALUES (10, 3);
INSERT INTO Proprietaires VALUES (11, 1);
```