

Le Sujet : Système de Gestion d'une Clinique Vétérinaire "AnimalTech"

Le Contexte :

La clinique souhaite automatiser la gestion de ses consultations.

Un client (nom, tel) peut posséder plusieurs animaux (nom, espèce, date de naissance).

Une consultation est effectuée par un seul vétérinaire pour un seul animal à une date donnée.

Lors d'une consultation, le vétérinaire peut prescrire plusieurs médicaments avec une posologie précise.

Certains vétérinaires sont des spécialistes (ex: chirurgien, cardiologue) et possèdent un numéro d'accréditation spécifique.

L'analyse du projet est incomplète et ne comporte d'un script de création SQL et un script UML, correspondant au diagramme de classe UML.

Je vous demande :

- Sur base du script de création de tables pour un SGDB Mysql.
 - Créer le MCD correspondant à ce script. Utiliser l'outil Looping pour la réalisation.
 - Donner le MLD coorespondant au MCD.
- Sur base du script UML de créer le diagramme de classe avec l'outil StarUML
- Chaque document devra être parfaitement documenté.

1. Script SQL, SGBD Mysql

CREATE TABLE Personne(

*idPersonne INT AUTO_INCREMENT,
nomPersonne VARCHAR(50) ,
prenomPersonne VARCHAR(50) ,
PRIMARY KEY(idPersonne)*

);

CREATE TABLE Commune(

*idAdresse INT AUTO_INCREMENT,
rueAdresse VARCHAR(50) ,
codepostal VARCHAR(4) ,
commune VARCHAR(50) ,
PRIMARY KEY(idAdresse)*

);

CREATE TABLE Telephone(

*idTelephone INT AUTO_INCREMENT,
numeroTelephone VARCHAR(20) ,
typeTelephone VARCHAR(25) ,
idPersonne INT NOT NULL,
PRIMARY KEY(idTelephone),
FOREIGN KEY(idPersonne) REFERENCES*

Personne(idPersonne)

);

CREATE TABLE Medicament(

*idMedicament INT AUTO_INCREMENT,
PRIMARY KEY(idMedicament)*

);

CREATE TABLE Client(

*idPersonne INT,
numeroClient VARCHAR(50) ,
dateEnregistrement DATE,
PRIMARY KEY(idPersonne),
FOREIGN KEY(idPersonne) REFERENCES*

Personne(idPersonne)

);

CREATE TABLE Veterinaire(

*idPersonne INT,
titre VARCHAR(50) ,
numeroOrdre INT,
PRIMARY KEY(idPersonne),
FOREIGN KEY(idPersonne) REFERENCES*

Personne(idPersonne)

);

CREATE TABLE Specialiste(

*idPersonne INT,
domaine VARCHAR(50) ,*

*PRIMARY KEY(idPersonne),
FOREIGN KEY(idPersonne) REFERENCES
Veterinaire(idPersonne)*

);

CREATE TABLE Animal(

*idAnimal INT AUTO_INCREMENT,
nomAnimal VARCHAR(50) ,
espece VARCHAR(50) ,
dateNaissance DATE,
idPersonne INT NOT NULL,
PRIMARY KEY(idAnimal),
FOREIGN KEY(idPersonne) REFERENCES*

Client(idPersonne)

);

CREATE TABLE Consultation(

*idConsultation INT AUTO_INCREMENT,
date_heure DATETIME,
idAnimal INT NOT NULL,
idPersonne INT NOT NULL,
PRIMARY KEY(idConsultation),
FOREIGN KEY(idAnimal) REFERENCES*

Animal(idAnimal),

FOREIGN KEY(idPersonne) REFERENCES

Veterinaire(idPersonne)

);

CREATE TABLE Localiser(

*idPersonne INT,
idAdresse INT,
natureAdresse VARCHAR(15) ,
PRIMARY KEY(idPersonne, idAdresse),
FOREIGN KEY(idPersonne) REFERENCES*

Personne(idPersonne),

FOREIGN KEY(idAdresse) REFERENCES

Commune(idAdresse)

);

CREATE TABLE Prescrire(

*idConsultation INT,
idMedicament INT,
Traitement VARCHAR(50) ,
posologie VARCHAR(50) ,
idPrescription INT AUTO_INCREMENT,
PRIMARY KEY(idConsultation, idMedicament),
FOREIGN KEY(idConsultation) REFERENCES*

Consultation(idConsultation),

FOREIGN KEY(idMedicament) REFERENCES

Medicament(idMedicament)

);

2. UML, Diagramme de classe.

ClassDiagram

```
class Personne {  
    +int idPersonne  
    +string nomPersonne  
    +string prenomPersonne  
}
```

```
class Client {  
    +string numeroClient  
    +date dateEnregistrement  
}
```

```
class Veterinaire {  
    +string titre  
    +int numeroOrdre  
}
```

```
class Specialiste {  
    +string domaine  
}
```

```
class Commune {  
    +int idAdresse  
    +string rueAdresse  
    +string codepostal  
    +string commune  
}
```

```
class Telephone {  
    +int idTelephone  
    +string numeroTelephone  
    +string typeTelephone  
}
```

```
%% Héritage  
Personne <|-- Client  
Personne <|-- Veterinaire  
Veterinaire <|-- Specialiste
```

```
%% Relations simples  
Personne "1" *-- "1..*" Telephone : possède  
Client "1" -- "0..*" Animal : possède  
Animal "1" -- "0..*" Consultation : subit  
Veterinaire "1" -- "0..*" Consultation : effectue
```

```
%% Classes d'association  
(Personne, Commune) .. Localiser  
(Consultation, Medicament) .. Prescrire
```

```
class Animal {  
    +int idAnimal  
    +string nomAnimal  
    +string espece  
    +date dateNaissance  
}
```

```
class Consultation {  
    +int idConsultation  
    +datetime date_heure  
}
```

```
class Medicament {  
    +int idMedicament  
}
```

```
class Localiser {  
    <<association>>  
    +string natureAdresse  
}
```

```
class Prescrire {  
    <<association>>  
    +int idPrescription  
    +string Traitement  
    +string posologie  
}
```