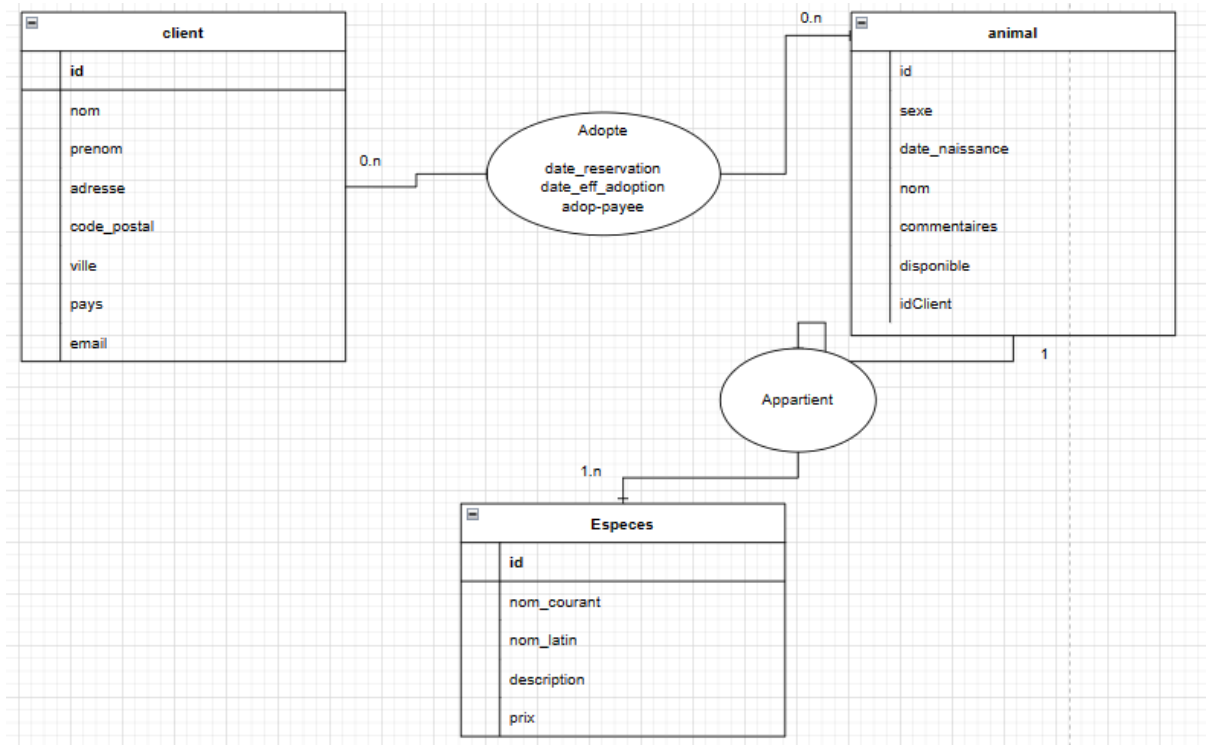


# Exercice 2 : Modélisation et Base de Données

## Modèle entité-association :



## Entités et Associations

### Client

Attributs : id (clé primaire), nom, prénom, adresse, code\_postal, ville, pays, email

### Animal

Attributs : id (clé primaire), sexe, date\_naissance, nom, commentaires, disponible, Espece\_id (clé étrangère), idClient (clé étrangère nullable)

### Espèce

Attributs : Espece\_id (clé primaire), nom\_courant, nom\_latin, description, prix

### Adoption

Attributs : AdoptionID (clé primaire), date\_reservation, date\_adoption, adop\_payee, animal\_id (clé étrangère), idClient (clé étrangère)

## Relations

Un Client peut adopter plusieurs Animaux (relation 1-n entre Client et Adoption).

Un Animal est adopté par un seul Client à la fois (relation 1-1 entre Animal et Adoption avec la possibilité que l'Animal ne soit pas encore adopté).

Un Animal appartient à une seule Espèce mais il peut y avoir plusieurs Animaux d'une même espèce (relation 1-n entre Espèce et Animal).

## Hypothèses

Un animal peut être réservé (indiqué par la date de réservation) mais pas encore officiellement adopté (indiqué par la date d'adoption).

La disponibilité d'un animal est indiquée par un booléen. Si l'animal est adopté, cette valeur est fausse.

L'adoption est considérée comme payée ou non payée au moment de l'adoption effective.

## Dictionnaire de données :

### Client

- id : INT, identifiant unique du client.
- nom : VARCHAR, nom du client.
- prénom : VARCHAR, prénom du client.
- adresse : VARCHAR, adresse du client.
- code\_postal : INT, code postal du client.
- ville : VARCHAR, ville du client.
- pays : VARCHAR, pays du client.
- Email : VARCHAR, email du client.

### Animal

- id : INT, identifiant unique de l'animal.
- sexe : VARCHAR, sexe de l'animal.
- nom : VARCHAR, nom de l'animal.
- date\_naissance : DATE, date de naissance de l'animal.
- commentaires : TEXT, commentaires sur l'animal.
- disponible : BOOLEAN, si l'animal est disponible pour adoption.
- espece\_id : INT, clé étrangère vers l'espèce de l'animal.
- idClient : INT, clé étrangère vers le client adoptant (nullable si pas adopté).

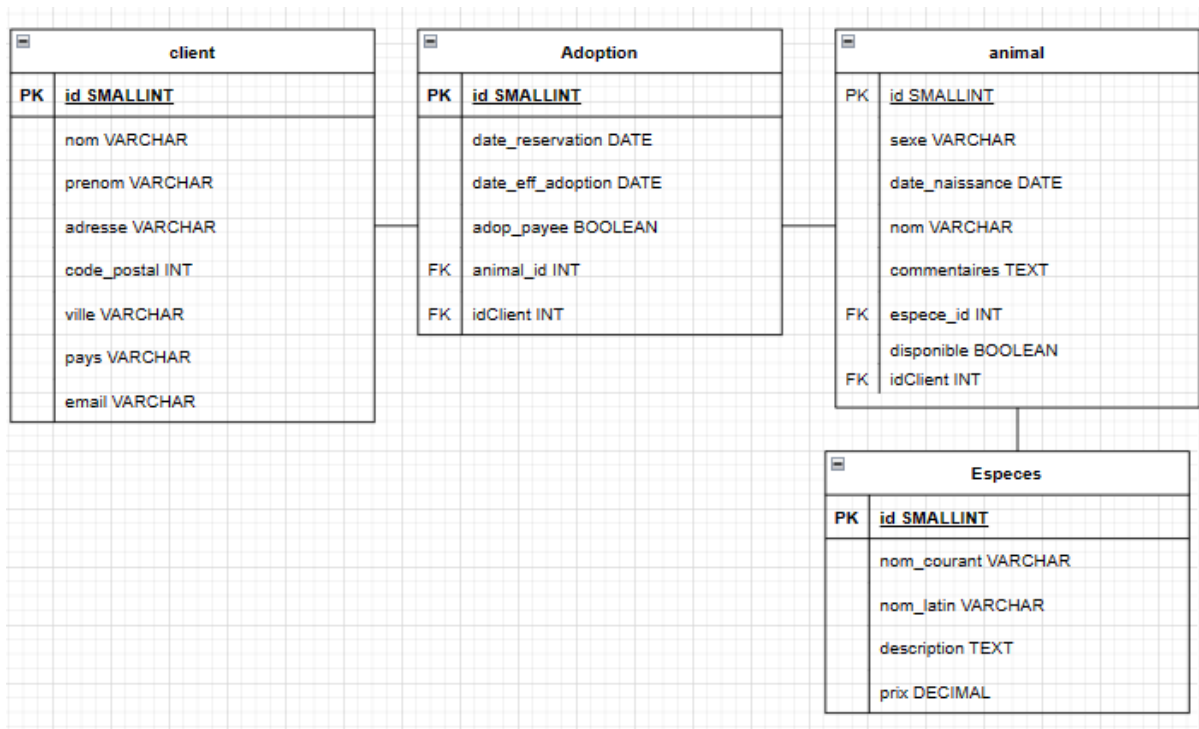
### Espèce

- espece\_id : INT, identifiant unique de l'espèce.
- nom\_courant : VARCHAR, nom courant de l'espèce.
- nom\_latin : VARCHAR, nom latin de l'espèce.
- description : TEXT, description de l'espèce.
- prix : DECIMAL, prix d'adoption de l'espèce.

## Adoption

- id: INT, identifiant unique de l'adoption.
- date\_reservation : DATE, date de réservation pour l'adoption.
- date\_adoption : DATE, date effective de l'adoption.
- adop\_payee : BOOLEAN, si l'adoption a été payée.
- animal\_id : INT, clé étrangère vers l'animal adopté.
- idClient : INT, clé étrangère vers le client adoptant.

## Modèle Relationnel



## Requêtes SQL :

Sélectionner les noms, commentaires et dates de naissance des animaux disponibles à l'adoption (selectDispo.sql) :

- `SELECT nom,commentaires,date_naissance FROM animal WHERE disponible = 1`

Supprimez l'adoption de l'animal dont l'id = 5. (deleteAdoption.sql)

- `DELETE FROM `adoption` WHERE id = 5`

Mettre à jour le statut disponible de l'animal dont l'id = 5. (updateAnimal.sql)

- `Update animal SET disponible = 0 WHERE id= 5`

Sélectionner les animaux de l'espèce "chat" : On suppose que le nom courant de l'espèce "chat" est stocké dans la table Espèce :

- `SELECT * FROM animal WHERE espece_id = (SELECT id FROM espece WHERE nom_courant = 'Chat');` (selectChat.sql).

Compter le nombre d'animaux par espèces :

- `SELECT e.nom_courant AS nom_espece, COUNT(*) AS nombre_animaux FROM animal a JOIN espece e ON a.espece_id = e.id GROUP BY e.nom_courant;` (nbEspece.sql).

## Exercice 4 : Récursivité :

Fonction CompterNonValides(defaults: tableau de réels, index: entier) : entier

Si index < 0 alors Retourner 0

FinSi

Si defaults[index] > 2 alors

Retourner 1 + CompterNonValides(defaults, index - 1)

Sinon

Retourner CompterNonValides(defaults, index - 1)

FinSi

FinFonction

## Exercice 5 : Programmation Orientée Objet :

### Classes

- Character : Classe de base pour les personnages du jeu.
- Elfe & Dragon : Classes dérivées de Character, représentant respectivement l'elfe et le dragon.
- Game : Contient la logique principale du jeu.
- Program : Va permettre de sélectionner le nom de l'elfe c'est tentative, le nom du dragon et lance la partie.