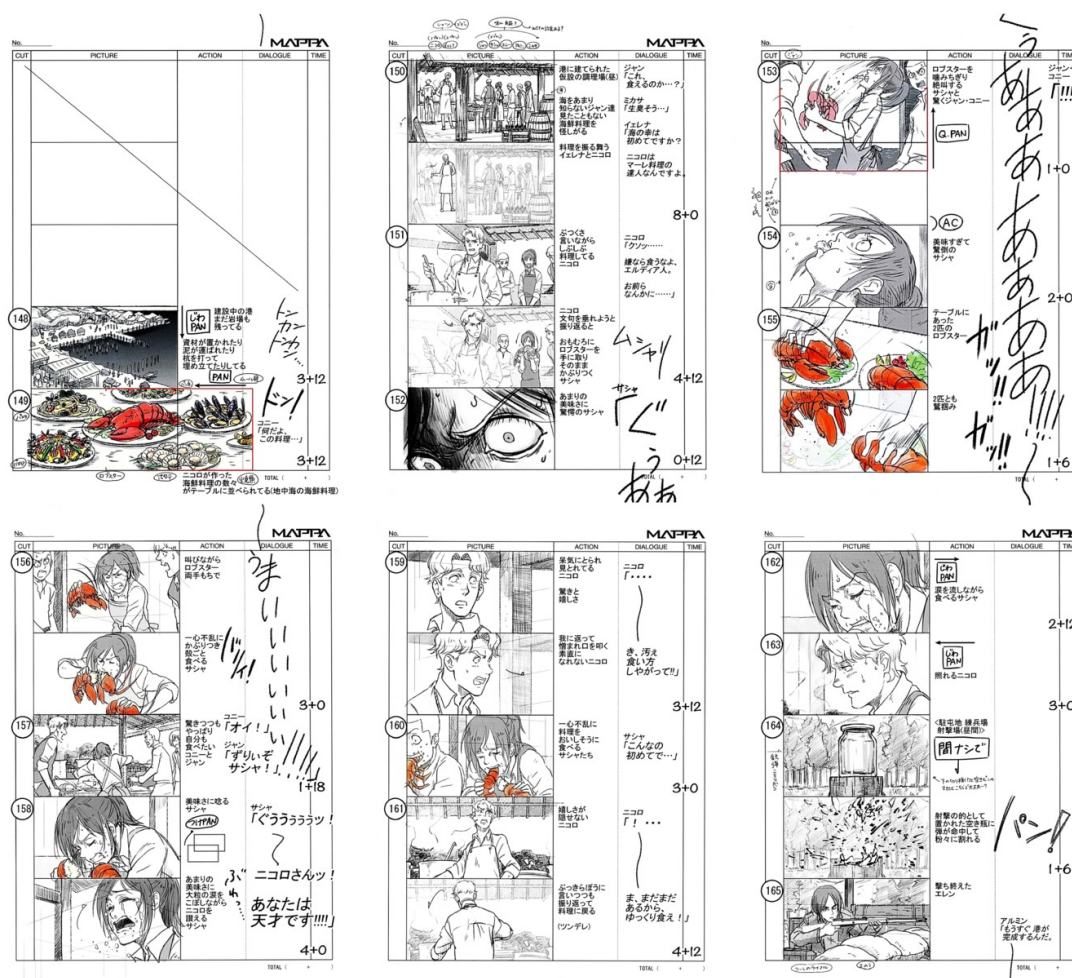


ENIB semestre S3 :
Informatique - Bases de données relationnelles

Studio Mappa : Optimiser le succès de leurs prochaines saisons.

Théo de Morais

Yann Weber Segarra



The storyboard consists of 15 panels, each with a numbered circle in the top left corner. The panels are organized into three columns. Each panel contains a visual sketch, an action description, and a dialogue box. Handwritten notes and time codes are present throughout the storyboard.

Panel 143: Visual of a kitchen with a large lobster. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 144: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 145: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 146: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 147: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 148: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 149: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 150: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 151: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 152: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 153: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 154: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 155: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 156: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 157: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 158: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 159: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 160: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 161: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 162: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 163: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 164: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Panel 165: Visual of a chef. Action: "建設中の港" (Port under construction). Dialogue: "ドン!" (Don!). Time: 3+12.

Table des matières

1	Cahier des charges initial	3
2	Identification des entités	3
3	Identification des associations	3
3.1	Entité-Association	3
3.2	Modélisation UML (version 1.1)	3
4	Identification des cardinalités	4
4.1	Association-Cardinalité	4
4.2	Modélisation UML (version 1.2)	4
5	Entité-Attributs	4
5.1	Identification des attributs	4
5.2	Modélisation UML (version 1.3)	5
6	Identification des cardinalités : Table associatives	5
6.1	identification des tables associatives	5
6.2	Modélisation UML (version 1.4)	6
7	de UML à SQL	6
7.1	Modélisation UML (version 1.4)	6
7.2	Structuration de la base de données	7
7.3	Insertion d'informations dans la base de données	8
8	Cas d'usage	9
8.1	Recherches des informations	9
8.2	Projection et restriction	9
8.3	Jointures	9

Table des figures

1	Modèle : Entité-Association	3
2	Modèle : Cardinalité d'associations	4
3	Modèle : Attributs	5
4	Modèle : Tables associatives	6
5	Modèle : Attributs clés	7

1 Cahier des charges initial

Le studio Mappa fait appel à nos services en Informatique pour mettre en place un Système d'Information pour étudier les paramètres qui font le succès d'une saison à partir du cahier des charges suivant :

Un studio possède un capital, des saisons qu'il a produite, Chaque animé a un titre, des saisons, un genre, un score moyen. Chaque saison comporte un titre, un score entre 0 et 10, un nombre de visionnages, des personnages, une date de diffusion, un personnel, un nombre d'épisodes.

Chaque membre du personnel a un nom, un prénom, un âge, un rôle. Chaque personnage a un rôle, et un doubleur dont on connaît le nom, prénom, âge, les animés dans lesquels il a doublé, ainsi que les personnages qu'il a doublé.

2 Identification des entités

A partir de ce cahier des charges on relève le nom des entités les plus importantes et les informations qui y sont liées.

- Un studio a un nom, dispose d'un capital et produit des saisons.
- Un animé a un titre, un genre, un score moyen, un nombre de saison.
- Une saison a un titre, un nombre de visionnage, une date de diffusion, un personnel, des personnages et un nombre d'épisode.
- Le personnel a des rôles, des personnes.
- Une personne a un nom, un prénom, un âge.
- Un personnage a un rôle, un nom.

3 Identification des associations

Les associations relient les entités entre elles. On relève les associations à partir de phrases (Sujet-Verbe-Complément) du cahier des charges. Le Verbe représente l'association entre deux entités (Sujet, Complément).

3.1 Entité-Association

partir du cahier des charges de l'entreprise BIEN on retient les phrases importantes reliant les entités par des verbes :

- un **animé** est composé de **saisons**.
- les **animés** ont des **personnages**.
- Une **saison** a un **personnel**.
- les **personnages** sont doublés par le **personnel**.
- le **personnel** est une **personne**.
- un **studio** produit des **saisons**.

3.2 Modélisation UML (version 1.1)

A partir de ces associations entre entités on peut proposer une première version de la structuration d'une base de données à l'aide du formalisme UML.

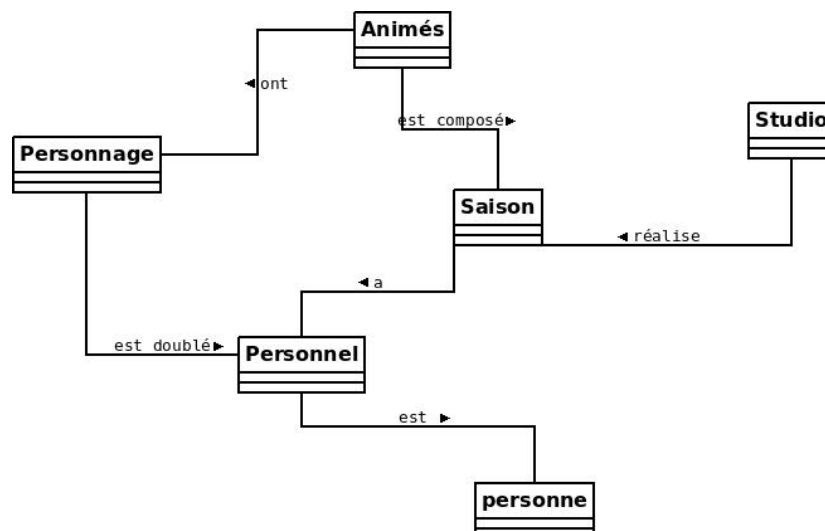


FIGURE 1 – Modèle : Entité-Association

4 Identification des cardinalités

A partir de ces phrases du cahier des charges on peut également identifier la cardinalité des associations (un-un, un-plusieurs, plusieurs-plusieurs, etc ...).

4.1 Association-Cardinalité

L'identification des cardinalités est très importante afin de pouvoir gérer les contraintes référentielles entre les entités. Ces cardinalités permettront de savoir sur quelle entité (table) il sera nécessaire de placer une référence (clé étrangère) sur l'autre entité (table) en association ou s'il est nécessaire de définir une nouvelle entité pour relier les deux entités, ce qui sera le cas pour les associations plusieurs-à-plusieurs (many-to-many).

- Un animé est composé de plusieurs saisons.
- Un animé ont plusieurs personnages.
- Un Studio réalise une ou plusieurs saison.
- Le personnel est une personne.
- Une saison a plusieurs personnels.
- Un personnage est doublé par un personnel.

4.2 Modélisation UML (version 1.2)

En UML les cardinalités sont représentées aux extrémités des associations (0..1,1,0..*,1..*, etc ...).

- un animé est composé d'au moins une saison (1..*)
- une saison se trouve nécessairement dans un animé (1)
- une saison comporte au moins un personnel (1..*)
- un personnel a participé à une ou plusieurs saison (1..*)
- une saison est réalisée par un ou plusieurs studios (1..*)
- un studio a réalisé une ou plusieurs saisons (1..*)
- le personnel a travaillé sur une saison ou plusieurs saisons (1..*)
- une saison est créée par un ou plusieurs personnels (1..*)
- un personnage est doublé par un seul personnel (1..1)
- un personnel peut doubler un personnage (0..1)
- une personne est un ou plusieurs personnels (1..*)
- un personnel est une et une seule personne (1..1)

A partir des cardinalités identifiées précédemment, on peut proposer une deuxième version du modèle UML.

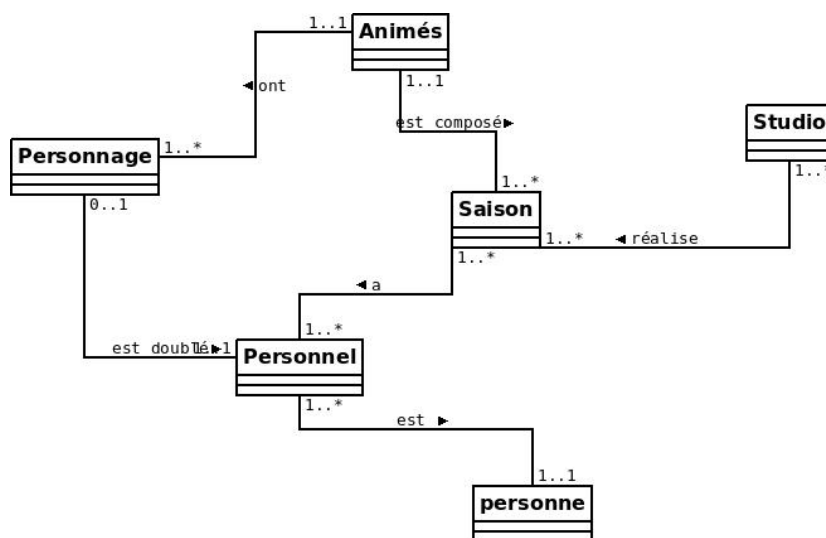


FIGURE 2 – Modèle : Cardinalité d'associations

5 Entité-Attributs

Après avoir identifié les entités importantes, à partir du cahier des charges on doit pouvoir relever les informations les caractérisant.

5.1 Identification des attributs

Dans notre cahier des charges on peut retenir les caractéristiques suivantes :

- Un studio a un nom, un capital

- Un animé a un titre, un nombre de saisons, un genre, un score moyen
- Une saison a un titre, un score, un nombre de visionnage, une date de diffusion, un nombre d'épisodes
- Le personnel a un rôle
- les personnes ont un nom, un prénom, une date de naissance.
- Les personnages ont un rôle et un nom.

5.2 Modélisation UML (version 1.3)

On peut représenter les attributs des entités dans une troisième version du modèle UML.

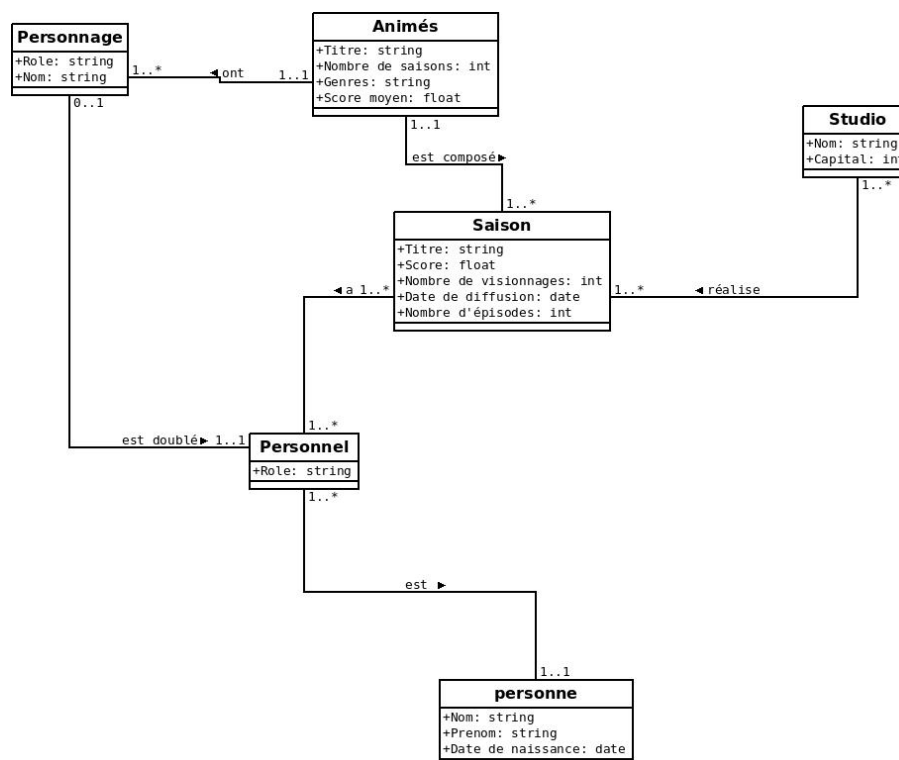


FIGURE 3 – Modèle : Attributs

6 Identification des cardinalités : Table associatives

6.1 identification des tables associatives

Dans le diagramme 1.2 nous pouvons voir que il y a des associations plusieurs a plusieurs. Nous allons créer des tables associatives pour les lever.

6.2 Modélisation UML (version 1.4)

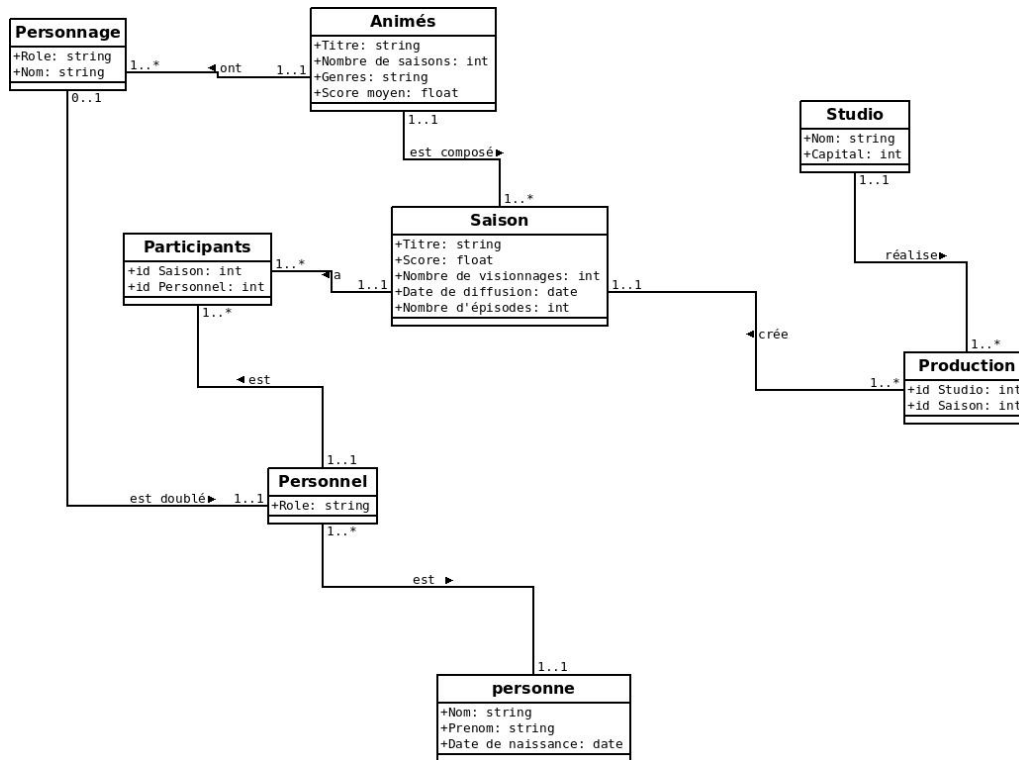


FIGURE 4 – Modèle : Tables associatives

Cette version peut-être considérée comme étant une première version que l'on peut proposer au client par rapport au cahier des charges initial.

7 de UML à SQL

Le passage d'une représentation à l'aide du formalisme UML à une structuration sous forme de base de données relationnelles nécessite de faire apparaître sur les entités des attributs qui représenteront les clés primaires de tables et les clés étrangères qui référenceront les clés primaires des entités (tables) en association.

7.1 Modélisation UML (version 1.4)

A partir des cardinalités des associations on peut représenter sur le modèle UML précédent les clés primaires et étrangères Personne.id, Saison.personnel_id ...).

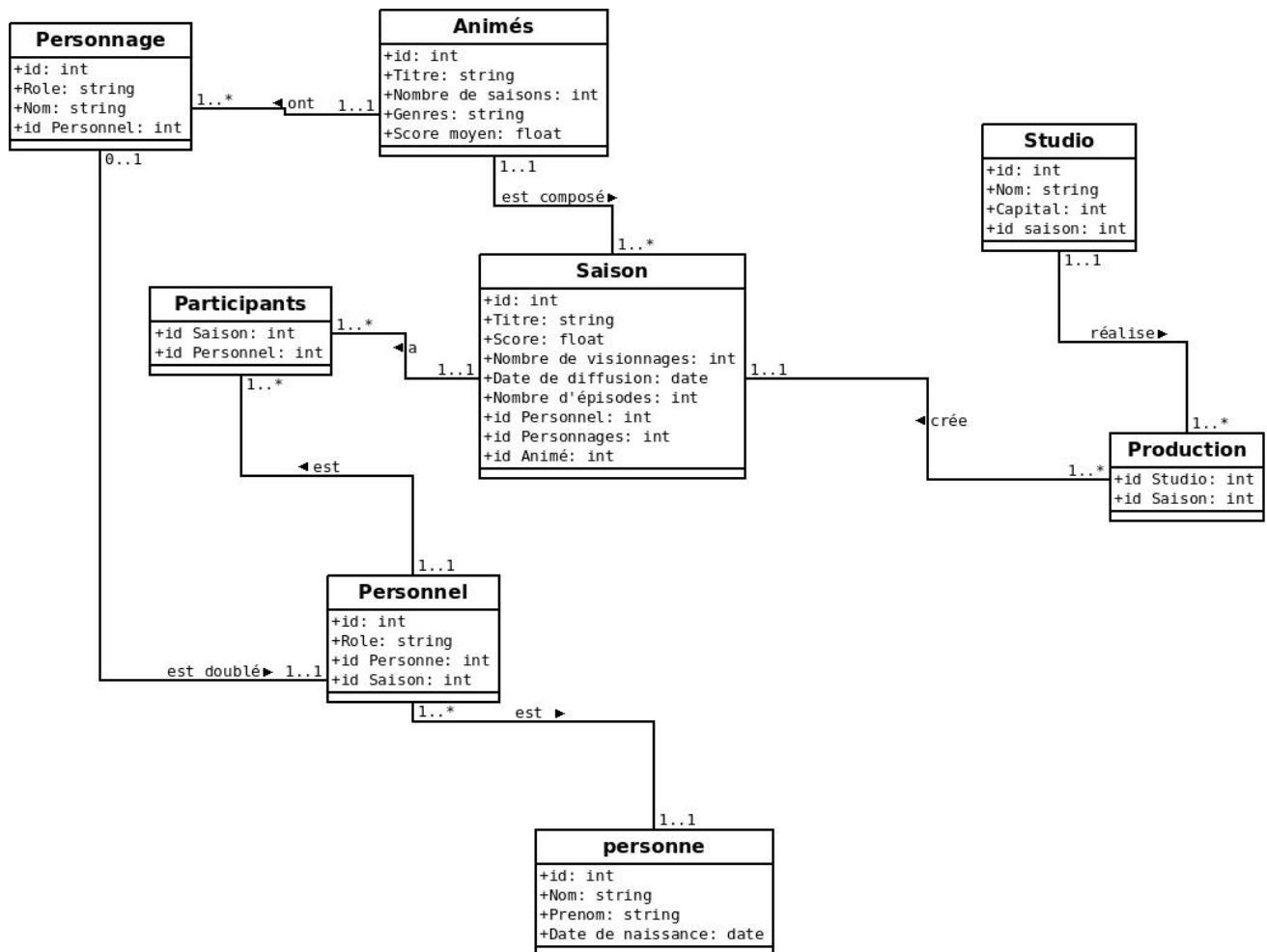


FIGURE 5 – Modèle : Attributs clés

7.2 Structuration de la base de données

A partir de ce schéma UML, on pourra mettre en œuvre la structuration de la base de données en SQL.

- Studio (id,nom,capital,#Saison_id);
- Saison (id,Titre,Score,Nombre de visionnages,Date de diffusion,Nombre d'épisodes,#Personnel_id,#Personnage_id,#Animé_id);
- Personnes (id,nom,prenom,Date de naissance);
- Personnel (id,Role,#Personne_id,#Saison_id);
- Personnage (Role,Nom,#Personnel_id,id);
- Participants (#Saison_id,#Personne_id);
- Production (#Studio_id,#Saison_id);

```

1 CREATE TABLE Animé
2 (
3     anime_id INTEGER NOT NULL PRIMARY KEY ,
4     titre varchar(20),
5     nombre_saisons INTEGER,
6     genre varchar(20),
7     score_moyen float DEFAULT 0,
8 );
9
10 CREATE TABLE Saison
11 (
12
13     saison_id INTEGER NOT NULL PRIMARY KEY,
14     titre varchar(20),
15     score float DEFAULT 0,
16     nombre_visionnages INTEGER,
17     date_diffusion date,
18     nombre_episodes INTEGER,
19     FOREIGN KEY ( personnel_id ) REFERENCES Personnel,
20     FOREIGN KEY ( personnage_id ) REFERENCES Personnage,
21     FOREIGN KEY ( anime_id ) REFERENCES Animé
22
23 );
24

```

```

25 CREATE TABLE Personnage
26 (
27
28     personnage_id INTEGER NOT NULL PRIMARY KEY,
29     role varchar(20),
30     nom varchar(20),
31     FOREIGN KEY ( personnage_id ) REFERENCES Personnel
32
33 );
34
35 CREATE TABLE Personnel
36 (
37
38     personnel_id INTEGER NOT NULL PRIMARY KEY,
39     role varchar(20),
40     FOREIGN KEY ( personne_id ) REFERENCES Personne,
41     FOREIGN KEY ( saison_id ) REFERENCES Saison
42
43 );
44
45 CREATE TABLE Personne
46 (
47
48     personne_id INTEGER NOT NULL PRIMARY KEY,
49     nom varchar(20),
50     prenom varchar(20),
51     date_naissance date
52
53 );
54
55 CREATE TABLE Studio
56 (
57
58     studio_id INTEGER NOT NULL PRIMARY KEY,
59     nom varchar(20),
60     capital float DEFAULT 0,
61     FOREIGN KEY ( saison_id ) REFERENCES Saison
62
63 );
64
65 CREATE TABLE Participants
66 (
67
68     FOREIGN KEY ( saison_id ) REFERENCES Saison,
69     FOREIGN KEY ( personnage_id ) REFERENCES Personnel
70
71 );
72
73 CREATE TABLE Production
74 (
75
76     FOREIGN KEY ( studio_id ) REFERENCES Studio,
77     FOREIGN KEY ( saison_id ) REFERENCES Saison
78
79 );

```

7.3 Insertion d'informations dans la base de données

A partir de cette structuration on peut alimenter la base de données en insérant des informations et faisant des mises à jour à l'aide de commandes SQL (INSERT INTO ... VALUES ...; UPDATE ... SET ... WHERE ...;)

```

1  -- insertion d un animé
2  INSERT INTO Animé ( titre,nombre_saisons,genre,score_moyen ) VALUES ( 'Tokyo Revengers',1,'Action'
    ,8.0);
3
4  -- insertion d'une saison
5  INSERT INTO Saison ( titre,score,nombre_visionnages,date_diffusion,nombre_episodes) VALUES ('Tokyo
    Revengers',8.41,200 000,5/03/2012,12);
6
7  -- insertion d un personnage
8  INSERT INTO Personnage (role,nom,prenom) VALUES ('Personnage principale','Takemichi','Hanagaki');
9
10 -- insertion d une personne
11 INSERT INTO Personne (nom, prenom,date_naissance) VALUES ('Yuuki','Shin');
12
13 -- insertion d un personnel
14 INSERT INTO Personnel(role) VALUES ('doubleur');
15
16 -- insertion d un studio
17 INSERT INTO Studio (nom, capital) VALUES ('LIDENFILMS',10000);

```


On trouvera en annexe (p.16) l'insertion d'informations dans l'ensemble des tables du modèle de données en SQL. Des exemples de mises à jour (p.17) sont également fournis.

8 Cas d'usage

A partir de ce modèle de données on peut représenter les cas d'usage qui permettront de tester les requêtes :

1. sur une seule table avec projections (Π) et critères de restriction (σ)
2. sur plusieurs tables par jointure (\bowtie) et critères de restriction (σ)
3. par des requêtes ensemblistes (\cup, \cap, \setminus)
4. avec une division relationnelle (\div)
5. en appliquant des fonctions d'agrégats ($\text{count}(), \text{sum}(), \text{max}(), \text{min}(), \text{avg}() \dots$)
6. en faisant des groupements (GROUP BY)
7. puis des groupements avec critères de restrictions (GROUP BY ... HAVING)

8.1 Recherches des informations

Sur ce modèle de comparaison des séries animées on pourrait mettre en oeuvre les cas d'usage suivants :

1. Π, σ : nom et capital pour les studios dont le capital est compris entre 0.00 € et 10000€
2. Π, \bowtie, σ : nom et prenom de tous les personnages de l'animé "Tokyo Revengers"
3. \cup, \cap : le nom des personnes qui ont participé à la saison 1 de "Tokyo Revengers" qui ne sont pas des doubleurs.
4. \div : le nom des personnes ayant eu tous les rôles.
5. $\text{count}(), \text{sum}(), \text{max}(), \text{min}(), \text{avg}() \dots$: les saisons ayant le minimum de participants
6. GROUP BY :

8.2 Projection et restriction

Nom et capital pour les studios dont le capital est compris entre 0.00 € et 10000.00 €

— calcul relationnel :

- $E = \{(e.\text{nom}, e.\text{capital}) \mid e \in \text{Studio} \wedge 0 \leq e.\text{capital} \leq 10000.00\}$ ou :
- $E = \{(e.\text{nom}, e.\text{capital}) \mid \text{Studio}(e) \wedge 0 \leq e.\text{capital} \leq 10000.00\}$

8.3 Jointures

Nom de tous les personnages de l'animé "Tokyo Revengers"

— calcul relationnel :

- $E1 = \{e.\text{anime_id} \mid \text{Animé}(e) \wedge \text{titre} = \text{'Tokyo Revengers'}\}$
- $E = \{p.\text{nom}, p.\text{prenom} \mid \text{Personnage}(p), E1(e1) \wedge p.\text{personnage_id} = e1.\text{anime_id}\}$