

Rendu phase 3

Groupe 1

Jérémie Santoro, Théo Coutaudier
& Oscar Baume

Pour: René Rentsch



| | |
|-------------------------------------|----------|
| 1. Introduction | 3 |
| 2. Modélisation conceptuelle | 3 |
| 2.1 Schéma relationnel | 3 |
| 2.2 Script SQL | 4 |
| 2.3 Liste des CI | 6 |
| 3. Conclusion | 7 |

1.Introduction

Ce rapport explique nos étapes et nos choix pour passer de notre modèle EA à notre modèle relationnel. Notre modèle a été effectué par un script SQL dont nous expliquons également les choix.

2.Modélisation conceptuelle

2.1 Schéma relationnel

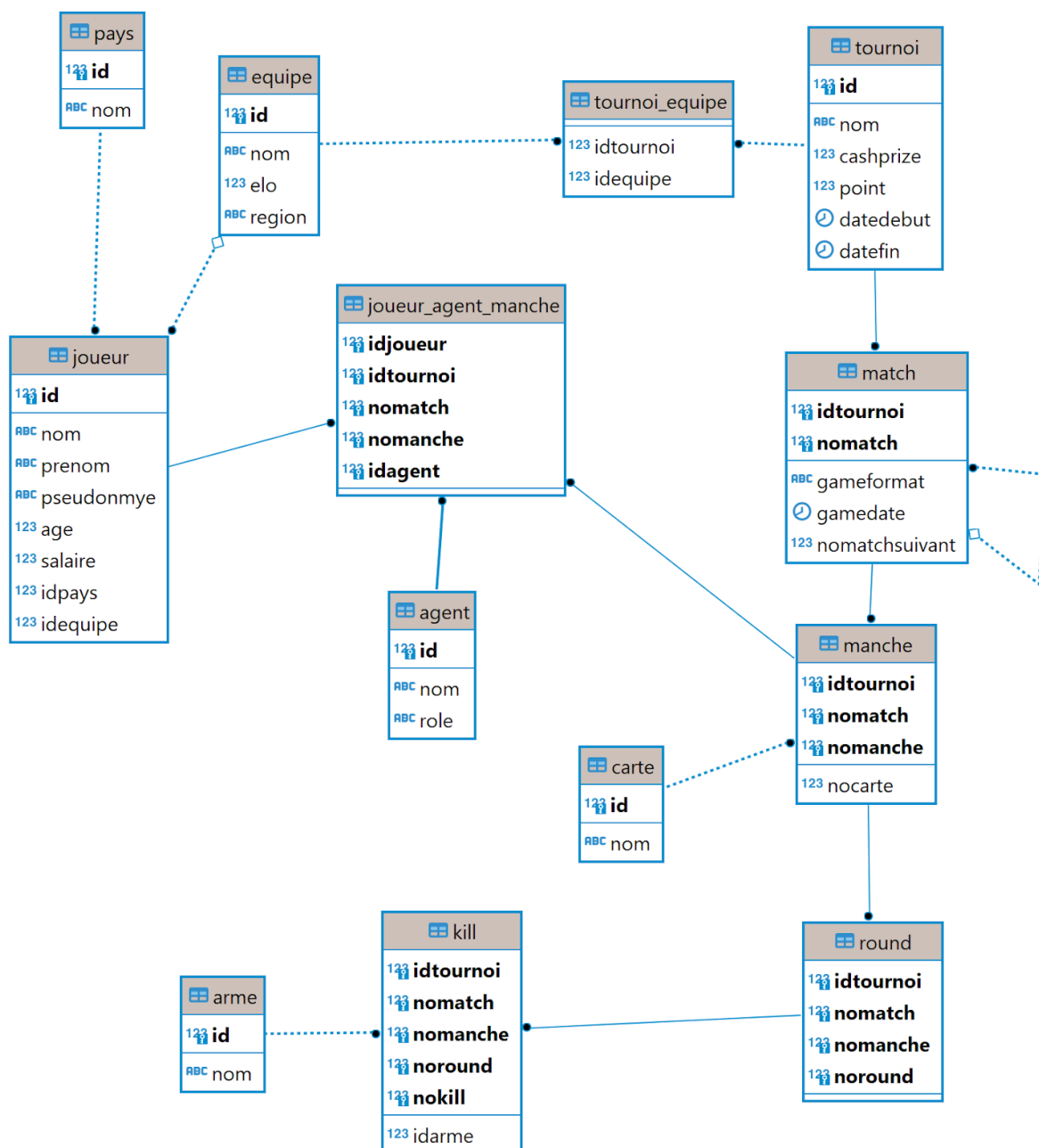


Schéma relationnel provenant de Dbeaver

2.2 Script SQL

```
DROP TYPE IF EXISTS AGENT_TYPE CASCADE;  
CREATE TYPE AGENT_TYPE AS ENUM('Controlleur','Duelliste','Initiateur','Sentinelle');
```

```
DROP TYPE IF EXISTS REGION CASCADE;  
CREATE TYPE REGION AS ENUM('EMEA','AMERICAS','PACIFIC');
```

```
DROP TYPE IF EXISTS GAME_FORMAT CASCADE;  
CREATE TYPE GAME_FORMAT AS ENUM ('bo1','bo3','bo5');
```

```
DROP DOMAIN IF EXISTS NAME_FORMAT CASCADE;  
CREATE DOMAIN NAME_FORMAT VARCHAR(20) NOT NULL;
```

```
DROP TABLE IF EXISTS Pays CASCADE;  
CREATE TABLE Pays(  
    id SMALLSERIAL,  
    nom NAME_FORMAT,  
    CONSTRAINT PK_Pays PRIMARY KEY (id)  
);
```

```
DROP TABLE IF EXISTS Agent CASCADE;  
CREATE TABLE Agent(  
    id SMALLSERIAL,  
    nom NAME_FORMAT,  
    role AGENT_TYPE not null,  
    CONSTRAINT PK_Agent PRIMARY KEY (id)  
);
```

```
DROP TABLE IF EXISTS Carte CASCADE;  
CREATE TABLE Carte(  
    id SMALLSERIAL,  
    nom NAME_FORMAT,  
    CONSTRAINT PK_Carte PRIMARY KEY (id)  
);
```

```
DROP TABLE IF EXISTS Equipe CASCADE;  
CREATE TABLE Equipe(  
    id SMALLSERIAL,  
    nom NAME_FORMAT,  
    elo INTEGER,  
    region REGION,  
    CONSTRAINT PK_Equipe PRIMARY KEY (id)  
);
```

```
DROP TABLE IF EXISTS Joueur CASCADE;  
CREATE TABLE Joueur(  
    id SMALLSERIAL,  
    nom NAME_FORMAT,  
    prenom NAME_FORMAT,  
    pseudonyme NAME_FORMAT,  
    age INTEGER not null,  
    salaire DECIMAL not NULL,  
    idPays INTEGER not null,  
    idEquipe INTEGER,  
    CONSTRAINT PK_Joueur PRIMARY KEY (id),  
    CONSTRAINT FK_Joueur_idPays FOREIGN KEY (idPays) REFERENCES Pays(id),  
    CONSTRAINT FK_Joueur_idEquipe FOREIGN KEY (idEquipe) REFERENCES Equipe(id)
```

```
);
```

```
DROP TABLE IF EXISTS Tournoi CASCADE;
```

```
CREATE TABLE Tournoi(  
    id SMALLSERIAL,  
    nom NAME_FORMAT,  
    cashprize DECIMAL,  
    point INTEGER,  
    dateDebut DATE,  
    dateFin DATE,  
    CONSTRAINT PK_Tournoi PRIMARY KEY (id)  
);
```

```
DROP TABLE IF EXISTS Tournoi_Equipe CASCADE;
```

```
CREATE TABLE Tournoi_Equipe(  
    idTournoi INTEGER NOT NULL,  
    idEquipe INTEGER NOT NULL,  
    CONSTRAINT FK_Tournoi_Equipe_idTournoi FOREIGN KEY (idTournoi) REFERENCES Tournoi(id),  
    CONSTRAINT FK_Tournoi_Equipe_idEquipe FOREIGN KEY (idEquipe) REFERENCES Equipe(id)  
);
```

```
DROP TABLE IF EXISTS Match CASCADE;
```

```
CREATE TABLE Match(  
    idTournoi INTEGER NOT NULL,  
    noMatch SMALLSERIAL,  
    gameFormat GAME_FORMAT,  
    gameDate DATE,  
    noMatchSuivant INTEGER,  
    CONSTRAINT FK_Match_idTournoi FOREIGN KEY (idTournoi) REFERENCES Tournoi(id),  
    CONSTRAINT PK_Match PRIMARY KEY (idTournoi,noMatch),  
    CONSTRAINT FK_Match_noMatchSuivant FOREIGN KEY (idTournoi,noMatchSuivant) REFERENCES  
Match(idTournoi,noMatch)  
);
```

```
DROP TABLE IF EXISTS Manche CASCADE;
```

```
CREATE TABLE Manche(  
    idTournoi INTEGER NOT NULL,  
    noMatch INTEGER NOT NULL,  
    noManche SMALLSERIAL,  
    noCarte INTEGER NOT NULL,  
    CONSTRAINT PK_Manche PRIMARY KEY(idTournoi,noMatch,noManche),  
    CONSTRAINT FK_Manche_idTournoi_noMatch FOREIGN KEY (idTournoi,noMatch) REFERENCES  
Match(idTournoi,noMatch),  
    CONSTRAINT FK_Manche_idCarte FOREIGN KEY (noCarte) REFERENCES Carte(id)  
);
```

```
DROP TABLE IF EXISTS Round CASCADE;
```

```
CREATE TABLE Round(  
    idTournoi INTEGER NOT NULL,  
    noMatch INTEGER NOT NULL,  
    noManche INTEGER NOT NULL,  
    noRound SMALLSERIAL,  
    CONSTRAINT PK_Round PRIMARY KEY (idTournoi,noMatch,noManche,noRound),  
    CONSTRAINT FK_Round_idTournoi_noMatch_noManche FOREIGN KEY  
(idTournoi,noMatch,noManche) REFERENCES Manche(idTournoi,noMatch,noManche)  
);
```

```
DROP TABLE IF EXISTS Arme CASCADE;
```

```

CREATE TABLE Arme(
    id SMALLSERIAL,
    nom NAME_FORMAT,
    CONSTRAINT PK_Arme PRIMARY KEY (id)
);

DROP TABLE IF EXISTS Kill CASCADE;
CREATE TABLE Kill(
    idTournoi INTEGER NOT NULL,
    noMatch INTEGER NOT NULL,
    noManche INTEGER NOT NULL,
    noRound INTEGER NOT NULL,
    noKill SMALLSERIAL,
    idArme INTEGER NOT NULL,
    CONSTRAINT PK_Kill PRIMARY KEY (idTournoi,noMatch,noManche,noRound,noKill),
    CONSTRAINT FK_Kill_idTournoi_noMatch_noManche_noRound FOREIGN KEY
(idTournoi,noMatch,noManche,noRound) REFERENCES Round(idTournoi,noMatch,noManche,noRound),
    CONSTRAINT FK_Kill_idArme FOREIGN KEY (idArme) REFERENCES Arme(id)
);

DROP TABLE IF EXISTS Joueur_Agent_Manche CASCADE;
CREATE TABLE Joueur_Agent_Manche(
    idJoueur INTEGER NOT NULL,
    idTournoi INTEGER NOT NULL,
    noMatch INTEGER NOT NULL,
    noManche INTEGER NOT NULL,
    idAgent INTEGER NOT NULL,
    CONSTRAINT FK_Joueur_Agent_Manche_idJoueur FOREIGN KEY (idJoueur) REFERENCES
Joueur(id),
    CONSTRAINT FK_Joueur_Agent_Manche_idAgent FOREIGN KEY (idAgent) REFERENCES Agent(id),
    CONSTRAINT FK_Joueur_Agent_Manche_noManche FOREIGN KEY (idTournoi,noMatch,noManche)
REFERENCES Manche(idTournoi,noMatch,noManche),
    CONSTRAINT PK_Joueur_Agent_Manche PRIMARY KEY
(idJoueur,idTournoi,noMatch,noManche,idAgent)
);

```

2.3 Liste des CI

1. Un **match** *BOX*, se joue en un maximum de X **manches**. X étant défini via l'attribut *format* de **match**.
2. Un **match** fait partie d'un arbre de tournoi, donc pour les vainqueurs d'un **match** un autre **match** les attends. Pour modéliser cela on a une relation **match** à **match**. Cette relation dit qu'un **match** a un **match** suivant et 2 **matches** précédents.
3. Une **manche** sera jouée entre 13-25 **rounds**.
4. Chaque **round** comportera entre 5 et 9 **kills**.
5. Les **matches** d'un **tournoi** doivent être après sa *date de début* et avant sa *date de fin*.
6. Un **tournoi** ne peut pas avoir une *date de fin* qui se trouve avant celle de *début*.
7. Dans **Kill** un **joueur** ne peut tuer qu' un **joueur** de l'**équipe** adverse.
8. Un **joueur** ne peut pas être tué 2x dans le même **round**.
9. 2 **agents** ne peuvent pas être pris simultanément dans la même **équipe** lors de la même **manche**

10. 2 **manches** du même **match** ne peuvent pas avoir la même carte
11. Si une **équipe** a gagné $(X / 2 + 1)$ **manche** d'un **match**, alors le **match** est terminé et les autres **manches** ne sont pas jouées. (ex : si le format est BO5 et qu'une équipe a déjà gagné 3 **manches** $(5 / 2 + 1)$ alors les **manches** suivantes ne se jouent pas et le match est terminé).
12. Si une **équipe** gagne 13 **rounds** dans une **manche**, alors la **manche** est terminée et les **rounds** restants ne sont pas joués.
13. Un **joueur** ne peut pas changer **d'équipe** si ceux-ci sont dans un **tournoi** en cours
14. Une **équipe** ne peut pas participer à 2 **tournois** se chevauchant.
15. Lors d'un **round**, si les 5 **joueurs** d'une **équipe** ont été "**KILL**", alors le round est terminé, et il ne peut plus y avoir de "**KILL**".
16. Un **match** ne peut avoir que 0 ou 2 **matches** précédents. Si il n'a pas de match précédent cela signifie qu'il s'agit d'un match du premier tour.
17. Un **match** a toujours un match suivant sauf si il s'agit de la finale.

3. Conclusion

Notre script permet actuellement de créer la "base" d'une BD mais ne gère pas énormément de CI n'ayant pas accès aux *triggers* et aux *checks* par exemple.