

Projet Sissa

Paul Andrieu

Antoine Lucas

Bruno Duarte Lopes

28 mars 2025

Table des matières

| | | |
|----------|--------------------------|----------|
| 1 | Introduction | 1 |
| 2 | MCD | 2 |
| 2.1 | MCD Complet | 2 |
| 2.1.1 | Utilisateur | 2 |
| 2.1.2 | Rang | 3 |
| 2.1.3 | a le rang | 3 |
| 2.1.4 | Partie | 3 |
| 2.1.5 | Joue | 4 |
| 2.1.6 | Coup | 4 |
| 2.1.7 | joue coup | 5 |
| 2.1.8 | Affronte | 5 |
| 2.1.9 | Robot | 5 |
| 2.1.10 | Article | 6 |
| 2.1.11 | achete | 6 |
| 2.1.12 | rembourse | 7 |
| 2.1.13 | Remboursement | 7 |
| 2.1.14 | Promotion | 8 |
| 2.1.15 | a la promotion | 8 |
| 3 | MLD | 9 |
| 3.1 | MLD Textuel | 9 |
| 3.2 | MLD Complet | 9 |

1 Introduction

Dans le cadre du projet nous avons besoin d'une base de donnée pour nos utilisateurs. La première fonctionnalité de notre projet est le jeu de Morpion contre un ordinateur. La seconde est une boutique permettant aux utilisateurs d'acheter des produits relatifs au jeux du Morpion.

Pour représenter un utilisateur, nous avons besoin des tables : **utilisateur** 2.1.1, **rang** 2.1.2, **a le rang** 2.1.3.

Pour la fonctionnalité *jouer une partie*, nous avons besoin des tables : **Partie** 2.1.4, **Joue** 2.1.5, **Coup** 2.1.6, **joue coup** 2.1.7, **affronte** 2.1.8, **Robot** 2.1.9.

Pour la partie *boutique* nous avons besoin des tables : **Article** 2.1.10, **achete** 2.1.11, **rembourse** 2.1.12, **Remboursement** 2.1.13 **Promotion** 2.1.14, **a la promotion** 2.1.15. La convention de type pour les prix est : décimal à 8 chiffres dont 2 après la virgule.

2 MCD

2.1 MCD Complet

Voici le MCD complet. La taille du MCD étant grande vous trouverez une image disponible avec ce PDF : *complete_mcd.png*.

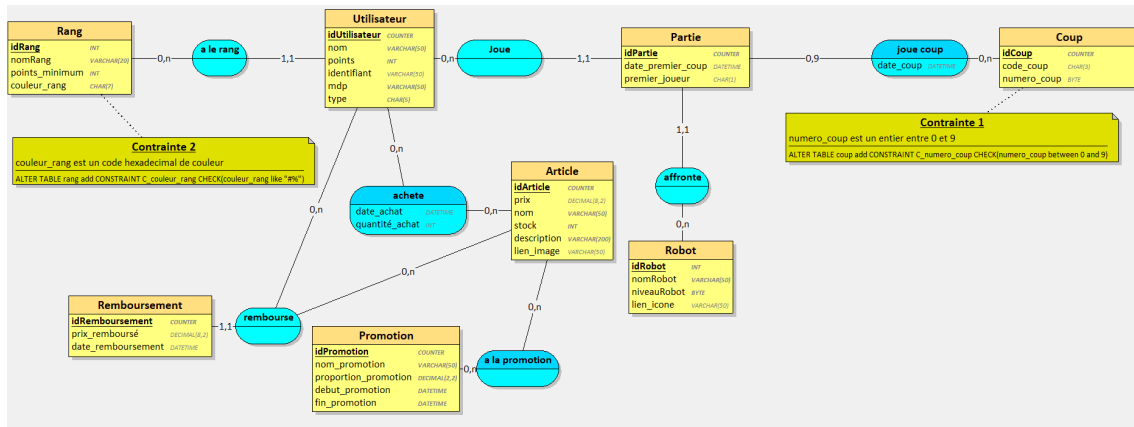


FIGURE 1 – MCD Complet

2.1.1 Utilisateur

La table *utilisateur* est la suivante :

| Utilisateur | |
|----------------------|-------------|
| idUtilisateur | COUNTER |
| nom | VARCHAR(50) |
| points | INT |
| identifiant | VARCHAR(50) |
| mdp | VARCHAR(50) |
| type | CHAR(5) |

FIGURE 2 – Table Utilisateur

Après les transformations en MLD, la création de la table *utilisateur* en SQL :

```
CREATE TABLE Utilisateur(
  idUtilisateur COUNTER,
  nom VARCHAR(50) NOT NULL,
  points INT NOT NULL,
  identifiant VARCHAR(50),
  mdp VARCHAR(50) NOT NULL,
  type CHAR(5) NOT NULL,
  idRang INT NOT NULL,
  PRIMARY KEY(idUtilisateur),
  FOREIGN KEY(idRang) REFERENCES Rang(idRang)
);
```

2.1.2 Rang

La table *rang* est la suivante :

| Rang | |
|----------------|-------------|
| <u>idRang</u> | INT |
| nomRang | VARCHAR(20) |
| points_minimum | INT |
| couleur_rang | CHAR(7) |

FIGURE 3 – Table Rang

Après les transformations en MLD, la création de la table *rang* en SQL :

```
CREATE TABLE Rang(  
  idRang INT,  
  nomRang VARCHAR(20) NOT NULL,  
  points_minimum INT NOT NULL,  
  couleur_rang VARCHAR(12) NOT NULL,  
  PRIMARY KEY(idRang)  
);
```

On notera l'ajout d'une contrainte pour le champ *couleur_rang*, car c'est un champ qui ne sera composé que de couleurs sous leur forme hexadécimale *#rrggbb* :

```
ALTER TABLE rang add CONSTRAINT C_couleur_rang CHECK(couleur_rang like "#%")
```

2.1.3 a le rang

La table associative *a le rang* est la suivante :



FIGURE 4 – Table a le rang

Après les transformations en MLD, la table associative *a le rang* disparaît et est remplacée par une clé étrangère dans la table *utilisateur* 2.1.1.

2.1.4 Partie

La table *Partie* est la suivante :

| Partie | |
|-------------------|----------|
| <u>idPartie</u> | COUNTER |
| date_premier_coup | DATETIME |
| premier_joueur | CHAR(1) |

FIGURE 5 – Table Partie

Le premier joueur est stocké par un code d'un unique caractère signifiant si le joueur ou le robot a commencé en premier.

Après les transformations en MLD, la création de la table *Partie* en SQL :

```
CREATE TABLE Partie(
    idPartie COUNTER,
    date_premier_coup DATETIME,
    premier_joueur CHAR(1),
    idRobot INT NOT NULL,
    idUtilisateur INT NOT NULL,
    PRIMARY KEY(idPartie),
    FOREIGN KEY(idRobot) REFERENCES Robot(idRobot),
    FOREIGN KEY(idUtilisateur) REFERENCES Utilisateur(idUtilisateur)
);
```

2.1.5 Joue

La table associative *Joue* est la suivante :

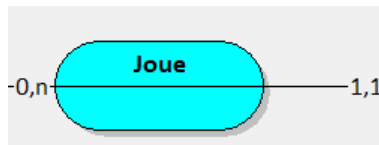


FIGURE 6 – Table Joue

Après les transformations en MLD, la table associative *Joue* disparaît et est remplacée par une clef étrangère dans *Partie* 2.1.4.

2.1.6 Coup

La table *Coup* est la suivante :

| Coup | |
|---------------|---------|
| <u>idCoup</u> | COUNTER |
| code_coup | CHAR(3) |
| numero_coup | BYTE |

FIGURE 7 – Table Coup

Après les transformations en MLD, la création de la table *Coup* en SQL :

```
CREATE TABLE Coup(
    idCoup COUNTER,
    code_coup CHAR(3),
    numero_coup BYTE,
    PRIMARY KEY(idCoup)
);
```

On notera l'ajout d'une contrainte pour le champ *numero_coup*, car c'est un champ qui encode le numéro du coup joué. Une partie de Morpion ne comportant qu'au maximum 9 coups le champ *numero_coup* ne comportera que 10 valeurs différentes (on garde une valeur pour identifier un coup illégal, erroné ... : code 0).

```
ALTER TABLE coup add CONSTRAINT C_numero_coup CHECK(numero_coup between 0 and 9)
```

2.1.7 joue coup

La table associative *joue coup* est la suivante :

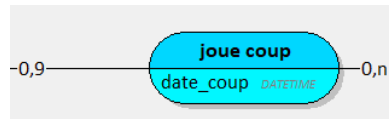


FIGURE 8 – Table joue coup

Après les transformations en MLD, la création de la table associative *joue coup* en SQL :

```
CREATE TABLE joue_coup(  
  idPartie INT,  
  idCoup INT,  
  date_coup DATETIME,  
  PRIMARY KEY(idPartie, idCoup),  
  FOREIGN KEY(idPartie) REFERENCES Partie(idPartie),  
  FOREIGN KEY(idCoup) REFERENCES Coup(idCoup)  
);
```

2.1.8 Affronte

La table associative *affronte* est la suivante :

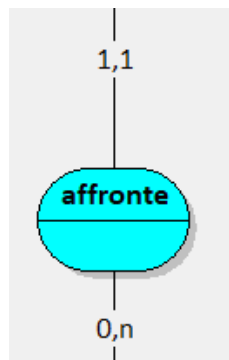


FIGURE 9 – Table affronte

Après les transformations en MLD, la table associative *affronte* disparaît et est transformée en clef étrangère dans la table *Partie* 2.1.4

2.1.9 Robot

La table *Robot* est la suivante :

| Robot | |
|----------------|-------------|
| <u>idRobot</u> | INT |
| nomRobot | VARCHAR(50) |
| niveauRobot | BYTE |
| lien_icone | VARCHAR(50) |

FIGURE 10 – Table Robot

Après les transformations en MLD, la création de la table *Robot* en SQL :

```
CREATE TABLE Robot(
  idRobot INT,
  nomRobot VARCHAR(50) NOT NULL,
  niveauRobot BYTE NOT NULL,
  lien_icone VARCHAR(50),
  PRIMARY KEY(idRobot)
);
```

2.1.10 Article

La table *Article* est la suivante :

| Article | |
|------------------|--------------|
| idArticle | COUNTER |
| prix | DECIMAL(8,2) |
| nom | VARCHAR(50) |
| stock | INT |
| description | VARCHAR(200) |
| lien_image | VARCHAR(50) |

FIGURE 11 – Table Article

Après les transformations en MLD, la création de la table *Article* en SQL :

```
CREATE TABLE Article(
  idArticle COUNTER,
  prix DECIMAL(8,2),
  nom VARCHAR(50) NOT NULL,
  stock INT NOT NULL,
  description VARCHAR(200) NOT NULL,
  lien_image VARCHAR(50),
  PRIMARY KEY(idArticle)
);
```

2.1.11 achete

La table associative *achete* est la suivante :

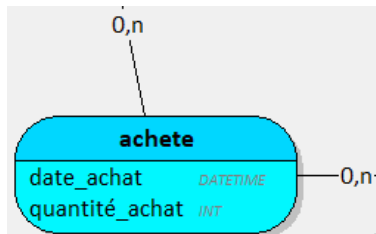


FIGURE 12 – Table achete

Après les transformations en MLD, la création de la table associative *achete* en SQL :

```
CREATE TABLE achete(
    idUtilisateur INT,
    idArticle INT,
    date_achat DATETIME,
    quantité_achat INT,
    PRIMARY KEY(idUtilisateur, idArticle),
    FOREIGN KEY(idUtilisateur) REFERENCES Utilisateur(idUtilisateur),
    FOREIGN KEY(idArticle) REFERENCES Article(idArticle)
);
```

2.1.12 rembourse

La table associative *rembourse* est la suivante :

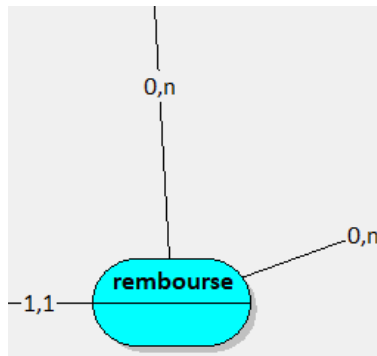


FIGURE 13 – Table rembourse

Après les transformations en MLD, la table associative *rembourse* disparaît et est remplacée par deux clefs étrangères dans la table *remboursement* 2.1.13.

2.1.13 Remboursement

La table *Remboursement* est la suivante :

| Remboursement | |
|------------------------|--------------|
| <u>idRemboursement</u> | COUNTER |
| prix_remboursé | DECIMAL(8,2) |
| date_remboursement | DATETIME |

FIGURE 14 – Table Remboursement

Après les transformations en MLD, la création de la table *Remboursement* en SQL :

```
CREATE TABLE Remboursement(
    idRemboursement COUNTER,
    prix_remboursé DECIMAL(8,2),
    date_remboursement DATETIME,
    idUtilisateur INT NOT NULL,
    idArticle INT NOT NULL,
    PRIMARY KEY(idRemboursement),
    FOREIGN KEY(idUtilisateur) REFERENCES Utilisateur(idUtilisateur),
    FOREIGN KEY(idArticle) REFERENCES Article(idArticle)
);
```

2.1.14 Promotion

La table *Promotion* est la suivante :

| Promotion | |
|----------------------|--------------|
| idPromotion | COUNTER |
| nom_promotion | VARCHAR(50) |
| proportion_promotion | DECIMAL(2,2) |
| debut_promotion | DATETIME |
| fin_promotion | DATETIME |

FIGURE 15 – Table Promotion

Après les transformations en MLD, la création de la table *Promotion* en SQL :

```
CREATE TABLE Promotion(  
  idPromotion COUNTER,  
  nom_promotion VARCHAR(50) NOT NULL,  
  proportion_promotion DECIMAL(2,2) NOT NULL,  
  debut_promotion DATETIME NOT NULL,  
  fin_promotion DATETIME NOT NULL,  
  PRIMARY KEY(idPromotion)  
);
```

2.1.15 a la promotion

La table associative *a la promotion* est la suivante :

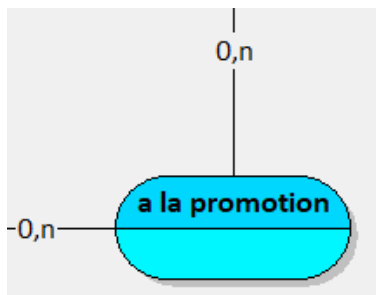


FIGURE 16 – Table a la promotion

Après les transformations en MLD, la création de la table associative *a la promotion* en SQL :

```
CREATE TABLE a_la_promotion(  
  idArticle INT,  
  idPromotion INT,  
  PRIMARY KEY(idArticle, idPromotion),  
  FOREIGN KEY(idArticle) REFERENCES Article(idArticle),  
  FOREIGN KEY(idPromotion) REFERENCES Promotion(idPromotion)  
);
```


3 MLD

3.1 MLD Textuel

```

Coup = (idCoup COUNTER, code_coup CHAR(3), numero_coup BYTE);
Article = (idArticle COUNTER, prix DECIMAL(8,2), nom VARCHAR(50), stock INT, description VARCHAR(200), lien_image VARCHAR(50));
Rang = (idRang INT, nomRang VARCHAR(20), points_minimum INT, couleur_rang CHAR(7));
Robot = (idRobot INT, nomRobot VARCHAR(50), niveauRobot BYTE, lien_icone VARCHAR(50));
Promotion = (idPromotion COUNTER, nom_promotion VARCHAR(50), proportion_promotion DECIMAL(2,2), debut_promotion DATETIME, fin_promotion DATETIME);
Utilisateur = (idUtilisateur COUNTER, nom VARCHAR(50), points INT, identifiant VARCHAR(50), mdp VARCHAR(50), type CHAR(5), #idRang);
Partie = (idPartie COUNTER, date_premier_coup DATETIME, premier_joueur CHAR(1), #idRobot, #idUtilisateur);
Remboursement = (idRemboursement COUNTER, prix_remboursé DECIMAL(8,2), date_remboursement DATETIME, #idUtilisateur, #idArticle);
achete = (#idUtilisateur, #idArticle, date_achat DATETIME, quantité_achat INT);
joue_coup = (#idPartie, #idCoup, date_coup DATETIME);
a_la_promotion = (#idArticle, #idPromotion);

```

FIGURE 17 – MLD Textuel

3.2 MLD Complet

Voici le MLD complet :

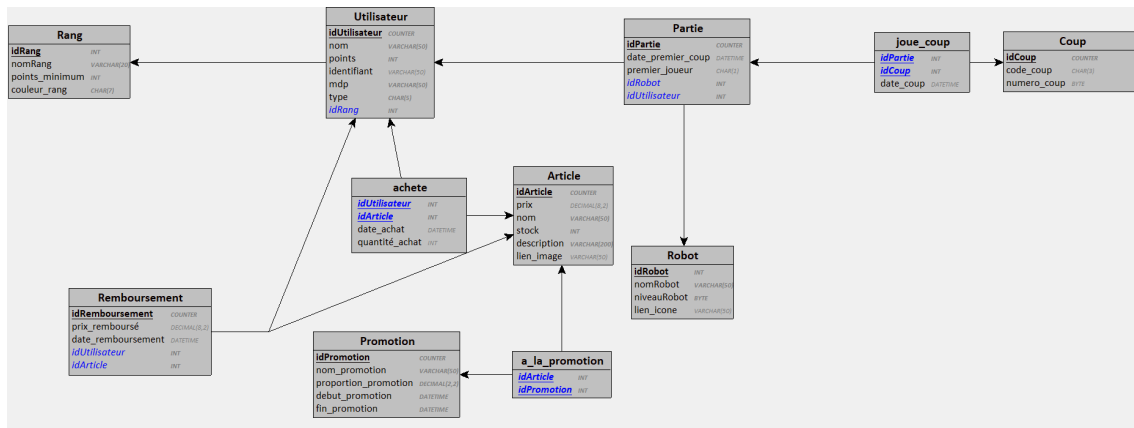


FIGURE 18 – MLD Complet