**BUT Informatique  
Année 2022-2023, Semestre 3**

**Étudiants (TDII)**

* **CATTAROSSI DARTIGUELONGUE Thomas TP3**
* **DUJARDIN Esteban TP4**
* **GAIGNE Lilian TP4**
* **LANUSSE Damien TP3**

**Enseignante tutrice : BRUYERE Marie**

**S3.A.D - Gestion des données**

**Schéma UML de la base de données et script**

Projet n°4 :

**Jdsel**

-

Application web de jeux de société

**TABLES DES MATIERES**

[**1. Diagramme de classe UML de la Base de Données**](#_l2zim3pcgvv4) **3**

[**2.Schéma Relationnel**](#_fsh71ya3sdhg) **4**

[**3. Création de la base de données**](#_u56t4nforvqn) **4**

[**4. Requêtes**](#_f8wj3nwg2w7g) **12**

[4.1. Requêtes d’insertions](#_tww4p6b16kda) 12

[4.1.1. Création de compte d’un Joueur](#_xo57263ib9tg) 12

[4.1.2. Création d’une partie dans la base](#_5pchqqx48umn) 12

[4.2. Requêtes de sélection](#_mx2z6hulqacn) 12

[4.2.1. Sélection des 50 meilleurs joueurs d’un jeu](#_8evi0hx913t3) 12

[4.2.2. Sélection des messages entre deux joueurs dans l’ordre d’envoi](#_devhxztdgki7) 12

[4.3. Requêtes de modification](#_r2ezebng6q09) 12

[4.3.1. Changer le thème sélectionné par le Joueur](#_mx24ueifavku) 12

[4.3.2. Changer le mot de passe](#_99hy94sffui6) 12

[4.4. Requêtes de cohérence](#_gkjimzb3fj47) 13

[4.4.1. Vérification du nombre de partie en cours d’un Joueur](#_446ive2flzr3) 13

[4.4.2. Vérification de validité du nom d’un nouveau thème](#_usmtrc1jrbzr) 13

[**5. Test de certaines requêtes**](#_eb5w52xsmcpt) **13**

## 

## Diagramme de classe UML de la Base de Données

## 2.Schéma Relationnel

Moderateur(identifiant, mdp)

Theme(identifiant, lienImage, prix, nom, #idModerateur)

Jeu(identifiant, nom, lienMinuature, description, lienJeu, #idModerateur)

Joueur(identifiant, pseudonyme, anneeNaiss, email, mdp, #idTheme)

Partie(identifiant, date, pointJoueur1, pointJoueur2, pointJoueur3, pointJoueur4, estFini, #idJeu, #idJoueur1, #idJoueur2, #idJoueur3, #idJoueur4)

Amis(#idJoueur1,#idJoueur2)

Message(identifiant, #idJEnvoi, #idJRetour, contenu, date, estAccessible, #idPartie)

## 3. Création de la base de données

CREATE TABLE Moderateur(

identifiant VARCHAR(50) PRIMARY KEY,

mdp VARCHAR(50) );

| Nom de l’élément | Elément | Définition |
| --- | --- | --- |
| identifiant | Clé primaire du modérateur (Chaîne de caractère) | Identifiant unique du modérateur |
| mdp | Chaîne de caractère | Mot de passe du modérateur pour se connecter à la base |
|  |  |  |

CREATE TABLE Theme(

identifiant VARCHAR(50) PRIMARY KEY,

lienImage VARCHAR(150),

prix INT(10),

nom VARCHAR(25),

idModerateur VARCHAR(50),

CONSTRAINT fkTheMod FOREIGN KEY(idModerateur) REFERENCES Moderateur(identifiant));

| Nom de l’élément | Elément | Définition |
| --- | --- | --- |
| identifiant | Clé primaire du thème (Chaîne de caractère) | Identifiant unique du thème |
| lienImage | Clé primaire de l’image du thème (Chaîne de caractère) | Lien de l’image associé au thème |
| prix | Entier | Prix du thème dans la boutique |
| nom | Chaîne de caractère | Nom du thème |
| idModerateur | Clé étrangère du modérateur | Identifiant faisant référence à l’identifiant présent dans la table modérateur |
|  |  |  |

CREATE TABLE Jeu(

identifiant VARCHAR(50) PRIMARY KEY,

nom VARCHAR(50),

lienMiniature VARCHAR(150),

description VARCHAR(50),

lienJeu VARCHAR(50),

idModerateur VARCHAR(50),

CONSTRAINT fkJeuMod FOREIGN KEY(idModerateur) REFERENCES Moderateur(identifiant));

| Nom de l’élément | Elément | Définition |
| --- | --- | --- |
| identifiant | Clé primaire du jeu (Chaîne de caractère) | Identifiant unique du jeu |
| nom | Chaîne de caractère | Nom du jeu |
| lienMiniature | Chaîne de caractère | Lien de l’image associé au jeu pour l’affichage à l’écran |
| description | Chaîne de caractère | Description du jeu |
| lienJeu | Chaîne de caractère | Url du jeu |
| idModerateur | Clé étrangère du modérateur | Identifiant faisant référence à l’identifiant présent dans la table modérateur |
|  |  |  |

CREATE TABLE Joueur(

identifiant VARCHAR(50) PRIMARY KEY,

pseudonyme VARCHAR(50),

anneeNaiss INT(4),

email VARCHAR(50),

mdp VARCHAR(50),

idTheme VARCHAR(50),

CONSTRAINT fkJouThe FOREIGN KEY(idTheme) REFERENCES Theme(identifiant));

| Nom de l’élément | Elément | Définition |
| --- | --- | --- |
| identifiant | Clé primaire du joueur (Chaîne de caractère) | Identifiant unique du joueur |
| pseudonyme | Chaîne de caractère | Pseudonyme que le joueur a renseigné lors de son inscription |
| anneeNaiss | Entier | Année de naissance que le joueur a renseigné lors de son inscription |
| email | Chaîne de caractère | Email que le joueur a renseigné lors de son inscription |
| mdp | Chaîne de caractère | Mot de passe qui sert au joueur à se connecter à son compte |
| idTheme | Clé étrangère du thème | Identifiant faisant référence à l’identifiant présent dans la table thème |
|  |  |  |

CREATE TABLE Partie(

identifiant VARCHAR(50) PRIMARY KEY,

datePartie DATETIME,

pointJoueur1 INT(3),

pointJoueur2 INT(3),

pointJoueur3 INT(3),

pointJoueur4 INT(3),

estFini BOOLEAN,

idJeu VARCHAR(50) NOT NULL,

idJoueur1 VARCHAR(50),

idJoueur2 VARCHAR(50),

idJoueur3 VARCHAR(50),

idJoueur4 VARCHAR(50),

CONSTRAINT fkParJeu FOREIGN KEY(idJeu) REFERENCES Jeu(identifiant),

CONSTRAINT fkParJou1 FOREIGN KEY(idJoueur1) REFERENCES Joueur(identifiant),

CONSTRAINT fkParJou2 FOREIGN KEY(idJoueur2) REFERENCES Joueur(identifiant),

CONSTRAINT fkParJou3 FOREIGN KEY(idJoueur3) REFERENCES Joueur(identifiant),

CONSTRAINT fkParJou4 FOREIGN KEY(idJoueur4) REFERENCES Joueur(identifiant));

| Nom de l’élément | Elément | Définition |
| --- | --- | --- |
| identifiant | Clé primaire de la partie (Chaîne de caractère) | Identifiant unique de la partie |
| date | Date | Date à laquelle le jeu à été ajouté |
| pointJoueur1 | Entier | Point que le joueur 1 a récupéré durant la partie |
| pointJoueur2 | Entier | Point que le joueur 2 a récupéré durant la partie |
| pointJoueur3 | Entier | Point que le joueur 3 a récupéré durant la partie |
| pointJoueur4 | Entier | Point que le joueur 4 a récupéré durant la partie |
| estFini | Boléen | Booléen de fin de partie |
| idJeu | Clé étrangère du jeu  (Chaîne de caractère) | Identifiant faisant référence à l'identifiant de la table Jeu |
| idJoueur1 | Clé étrangère du joueur 1  (Chaîne de caractère) | Identifiant faisant référence à l’identifiant présent dans la table Joueur |
| idJoueur2 | Clé étrangère du joueur 2  (Chaîne de caractère) | Identifiant faisant référence à l’identifiant présent dans la table Joueur |
| idJoueur3 | Clé étrangère du joueur 3  (Chaîne de caractère) | Identifiant faisant référence à l’identifiant présent dans la table Joueur |
| idJoueur4 | Clé étrangère du joueur 4  (Chaîne de caractère) | Identifiant faisant référence à l’identifiant présent dans la table Joueur |
|  |  |  |

CREATE TABLE Amis(

idJoueur1 VARCHAR(50) NOT NULL,

idJoueur2 VARCHAR(50) NOT NULL,

CONSTRAINT fkAmiJou1 FOREIGN KEY(idJoueur1) REFERENCES Joueur(identifiant),

CONSTRAINT fkAmiJou2 FOREIGN KEY(idJoueur2) REFERENCES Joueur(identifiant),

CONSTRAINT Pk PRIMARY KEY(idJoueur1, idJoueur2));

| Nom de l’élément | Elément | Définition |
| --- | --- | --- |
| idJoueur1 | Clé étrangère du joueur 1  (Chaîne de caractère) | Identifiant faisant référence à l’identifiant présent dans la table Joueur |
| idJoueur2 | Clé primaire du joueur 2 (Chaîne de caractère) | Identifiant faisant référence à l’identifiant présent dans la table Joueur |
|  |  |  |

CREATE TABLE Message(

identifiant VARCHAR(10) PRIMARY KEY,

idJEnvoi VARCHAR(50),

idJRetour VARCHAR(50),

contenu VARCHAR(10000),

date DATETIME,

estAccessible BOOLEAN,

idPartie VARCHAR(50),

CONSTRAINT fkMesPar FOREIGN KEY(idPartie) REFERENCES Partie(identifiant),

CONSTRAINT fkMesJou1 FOREIGN KEY(idJEnvoi) REFERENCES Joueur(identifiant),

CONSTRAINT fkMesJou2 FOREIGN KEY(idJRetour) REFERENCES Joueur(identifiant));

| Nom de l’élément | Elément | Définition |
| --- | --- | --- |
| identifiant | Clé primaire du message (Chaîne de caractère) | Identifiant unique du message |
| idJEnvoi | Clé étrangère de l’envoi du message du joueur | Identifiant faisant référence à l’identifiant présent dans la table Joueur |
| idJRetour | Clé étrangère du retour du message du joueur | Identifiant faisant référence à l’identifiant présent dans la table Joueur |
| contenu | Chaîne de caractère | Contenu du message envoyé par les joueurs |
| date | Date | Date d’envoi du message par le joueur |
| estAccessible | Booléen | Booléen de vérification d’accessibilité des messages |
| idPartie | Clé étrangère de l’identification de la partie | Identifiant faisant référence à l’identifiant présent dans la table Partie |
|  |  |  |

## 4. Requêtes

### 4.1. Requêtes d’insertions

#### 4.1.1. Création de compte d’un Joueur

INSERT INTO Joueur VALUES(identifiant, pseudonyme, anneeNaiss, email, mdp, themeParDefaut);

#### 4.1.2. Création d’une partie dans la base

INSERT INTO Partie VALUES(identifiant, idJoueur1, idJoueur2, idJoueur3, idJoueur4, 0, 0,0, 0, idJeu, False);

### 4.2. Requêtes de sélection

#### 4.2.1. Sélection des 50 meilleurs joueurs d’un jeu

SELECT pseudonyme, SUM(

(SELECT COALESCE(SUM(pointJoueur1),0) FROM Partie

WHERE idJeu = 'JE0' AND idJoueur1 = Joueur.identifiant)+

(SELECT COALESCE(SUM(pointJoueur2),0) FROM Partie

WHERE idJeu = 'JE0' AND idJoueur2 = Joueur.identifiant)+

(SELECT COALESCE(SUM(pointJoueur3),0) FROM Partie

WHERE idJeu = 'JE0' AND idJoueur3 = Joueur.identifiant)+

(SELECT COALESCE(SUM(pointJoueur4),0) FROM Partie

WHERE idJeu = 'JE0' AND idJoueur4 = Joueur.identifiant)

) as nbPoints FROM Joueur GROUP BY pseudonyme ORDER BY nbPoints DESC LIMIT 50;

#### 4.2.2. Sélection des messages entre deux joueurs dans l’ordre d’envoi

SELECT \* FROM Message WHERE (idJEnvoi = [idJoueur1] AND idJRetour = [idJoueur2])

OR (idJEnvoi = [idJoueur2] AND idJRetour = [idJoueur1]) ORDER BY date;

### 4.3. Requêtes de modification

#### 4.3.1. Changer le thème sélectionné par le Joueur

UPDATE JOUEUR SET idTheme = [idThemeChoisi] WHERE identifiant = [identifiant du Joueur];

#### 4.3.2. Changer le mot de passe

UPDATE JOUEUR SET mdp = [nouveauMdpChoisi] WHERE identifiant = [identifiant du Joueur];

### 4.4. Requêtes de cohérence

#### 4.4.1. Vérification du nombre de partie en cours d’un Joueur

SELECT COUNT(Partie.identifiant) FROM Partie

WHERE (Partie.idJoueur1 = [idJoueur] OR Partie.idJoueur2 = [idJoueur]

OR Partie.idJoueur3 = [idJoueur] OR Partie.idJoueur4 = [idJoueur])

AND Partie.estFini LIKE (FALSE);

#### 4.4.2. Vérification de validité du nom d’un nouveau thème

SELECT COUNT(nom) FROM Theme WHERE nom= [nomNouveauTheme];

## 5. Test de certaines requêtes

INSERT INTO Moderateur VALUES('M0', 'M0MDPINCONNU');

INSERT INTO Theme VALUES('T0', 'random.jpg', 20000, 'Cher', 'M0');

INSERT INTO Jeu VALUES('JE0', 'Memory', 'Mem.jpg', 'Le Jeu du Memory', 'Mem.exe', 'M0');

INSERT INTO Joueur VALUES('JO0', 'Bot', 2022, NULL, 'UYAbBERQBqUwsrmW&dPVQ29JmRKKkF59Ye95IWHo', 'T0');

INSERT INTO Joueur VALUES('JO1', 'Jean Pierre Pierregens', 1969, 'pierregens.jp@hotmail.fr', '123456789’, 'T0');

INSERT INTO Partie VALUES('P0', '2022-11-16', 5, 3, 0, 0, True, 'JE0', 'JO0', 'JO1', NULL, NULL);

INSERT INTO Partie VALUES('P1', '2022-11-15', 1, 5, 0, 0, True, 'JE0', 'JO0', 'JO1', NULL, NULL);

INSERT INTO Partie VALUES('P2', '2022-11-14', 1, 5, 0, 0, True, 'JE0', 'JO0', 'JO1', NULL, NULL);

INSERT INTO Partie VALUES('P3', '2022-11-13', 1, 5, 0, 0, True, 'JE0', 'JO1', 'JO0', NULL, NULL);

INSERT INTO Amis VALUES('JO0', 'JO1');

INSERT INTO Message VALUES('ME0','JO0', 'JO1', 'La Base de Données c\'est un gros bazard', '2022-11-15', True, NULL);

INSERT INTO Message VALUES('ME1','JO1', 'JO0', 'La Base de Données c\'est encore plus un gros bazard', '2022-11-16', True, NULL);