Projet BDD Matrix

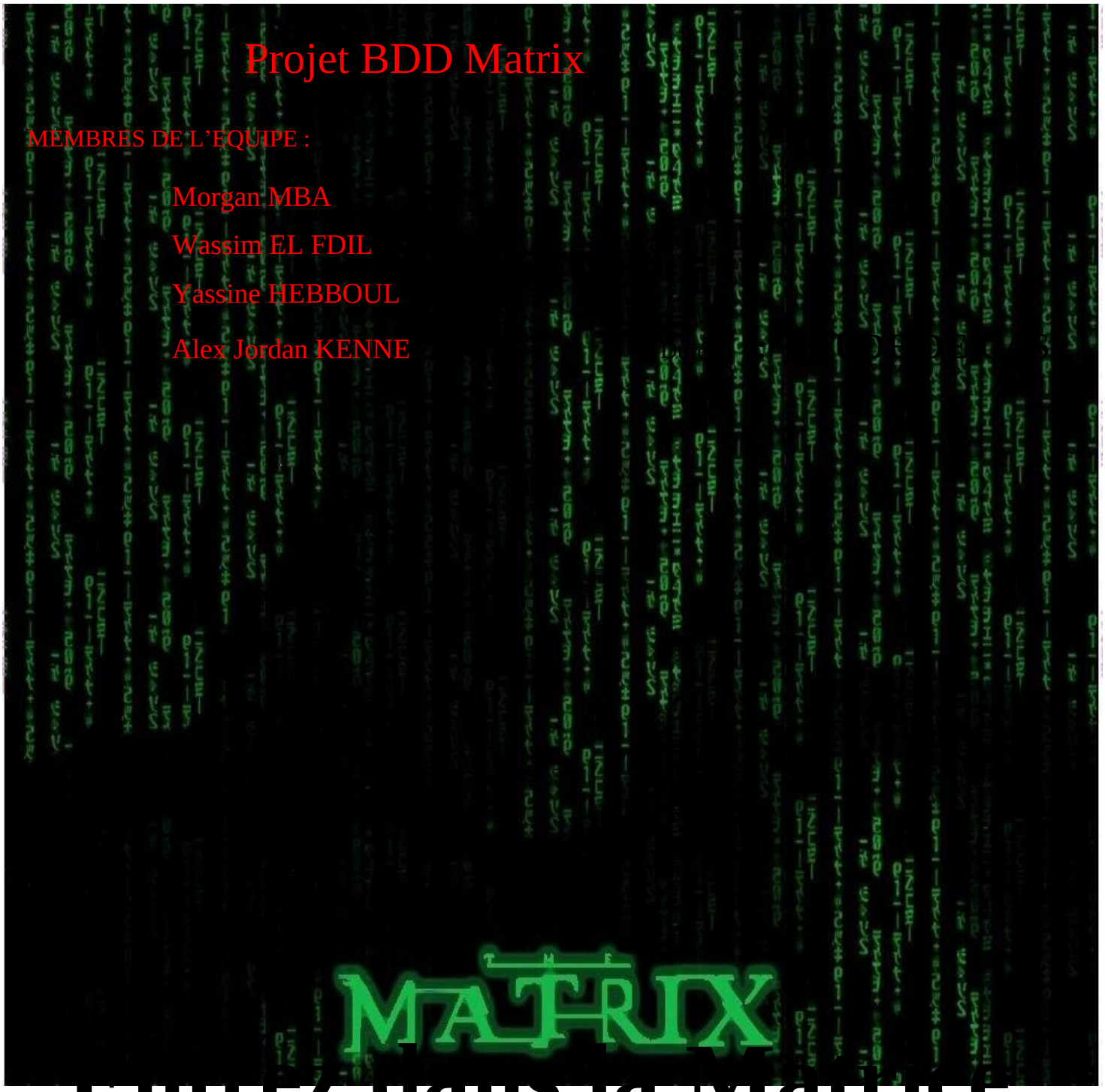
MEMBRES DE L'EQUIPE :

Morgan MBA

Wassim EL FDIL

Yassine HEBBOUL

Alex Jordan KENNE



Entrez dans la Matrice

Néo a enfin été trouvé par Morphéus. Ce dernier est persuadé qu'il est en présence de l'Elu, celui qui sauvera tous les hommes, prisonniers des machines.

En effet, nous sommes en 2199 et depuis de nombreuses années, les machines ont pris possession de la terre. Une épaisse couche nuageuse que le soleil ne peut plus percer recouvre toute la planète. En l'absence de soleil, les machines ont trouvé dans les hommes la source d'énergie nécessaire à leur fonctionnement. Pour cela, ils doivent les garder en vie. Les hommes sont ainsi maintenus dans un état inconscient et plongés dans un rêve permanent. Ils sont persuadés de vivre dans un monde qui n'est en réalité qu'un immense ensemble de programmes écrits pour créer l'illusion : la matrice.
Certains hommes ont pu se déconnecter et ainsi échapper à la matrice. Ils ont aussi trouvé le moyen d'entrer et de sortir de la matrice quand ils le désirent, mais, où qu'ils soient, ils sont poursuivis en permanence : par les sentinelles dans le monde réel, et par les agents (des programmes très puissants) dans la matrice. De plus en plus d'hommes sont libérés mais les machines représentent une menace permanente. Morphéus, un des hommes libérés, a foi en l'Oracle qui a prédit qu'un jour un homme viendra sauver tous les autres. Il sera plus fort, plus rapide, plus sage et trouvera le moyen de vaincre les machines.

"Bienvenue dans le monde réel." (Morphéus)

Morphéus a libéré Néo. Maintenant, il s'agit de l'entraîner, de lui expliquer les règles du monde réel et surtout celles de la matrice.
Au service de Morphéus, Link est chargé de développer les programmes qui permettent de créer des matrices d'entraînement. Ce n'est pas la vraie matrice mais elle doit fonctionner de la même façon pour permettre aux hommes de se préparer à affronter les agents (des programmes super puissants qui poursuivent les hommes libérés afin de les éliminer).

Link doit créer une nouvelle matrice d'entraînement avec les caractéristiques suivantes :

La matrice doit contenir un ensemble de lieux. Chaque lieu est localisable suivant 3 valeurs (X, Y et Z pour un repérage dans un espace à 3 dimensions) et comporte aussi un nom et une description. Les lieux sont de différents types (salle, plate forme, rue…), et certains lieux ont des rôles précis (entraînement au combat…).

Des personnages peuvent être créés et supprimés à chaque instant. Un personnage possède un nom et des caractéristiques d'apparence précises : un ensemble de caractéristiques (classées par catégorie, comme par exemple la catégorie "vêtement") est à la disposition du développeur. Ainsi, il crée à chaque fois un personnage en combinant plusieurs caractéristiques (par exemple une veste noire dans la catégorie vêtement).
Une fois un personnage créé, il est possible de l'utiliser plusieurs fois en le dupliquant. Il est nécessaire de connaître, pour chaque clone d'un personnage, son numéro (1er clone, 2ème clone…) et sa localisation actuelle (le lieu où il se trouve). Cette information sera mise à jour automatiquement : le système de mise à jour des localisations n'interviendra qu'au niveau programmation.

A tout moment et quelque soit le lieu choisi, il doit être possible aussi de faire apparaître des armes. Les armes ont un nom précis, une description détaillée et sont classées par catégorie. Pour les armes à feu, il est nécessaire de connaître les types de munitions nécessaires et la quantité maximum que l'arme peut stocker. Comme pour les personnages, les armes peuvent être dupliquées autant que nécessaire, mais il faudra aussi savoir, pour chaque arme dupliquée, sa localisation.

Les personnages sont séparés en 2 groupes : les personnages générés par le programme (donc fictifs) et ceux, bien réels, qui vont être insérés dans la matrice. Pour les personnages réels, il n'y a pas de caractéristiques à mémoriser mais juste le nom. Biensur il n'y a pas de clone généré à partir d'un personnage réel. Par contre, il est toujours nécessaire de savoir où il se trouve. Parmi les personnages fictifs, il existe 2 groupes : les personnages quelconques et les agents (simulation des ennemis).

Il est possible d'affecter un exemplaire d'arme à un clone précis. **Dans ce cas, l'exemplaire d'arme n'est plus rattaché à un lieu.**
Attention, chaque personnage fictif a le droit ou non de manipuler certaines armes suivant le lieu où il pourrait se trouver. **Donc, lorsqu'un exemplaire d'arme est affecté à un clone, cela ne doit être possible que si le personnage concerné à le droit de manipuler l'arme concernée à l'endroit où il se trouve.**

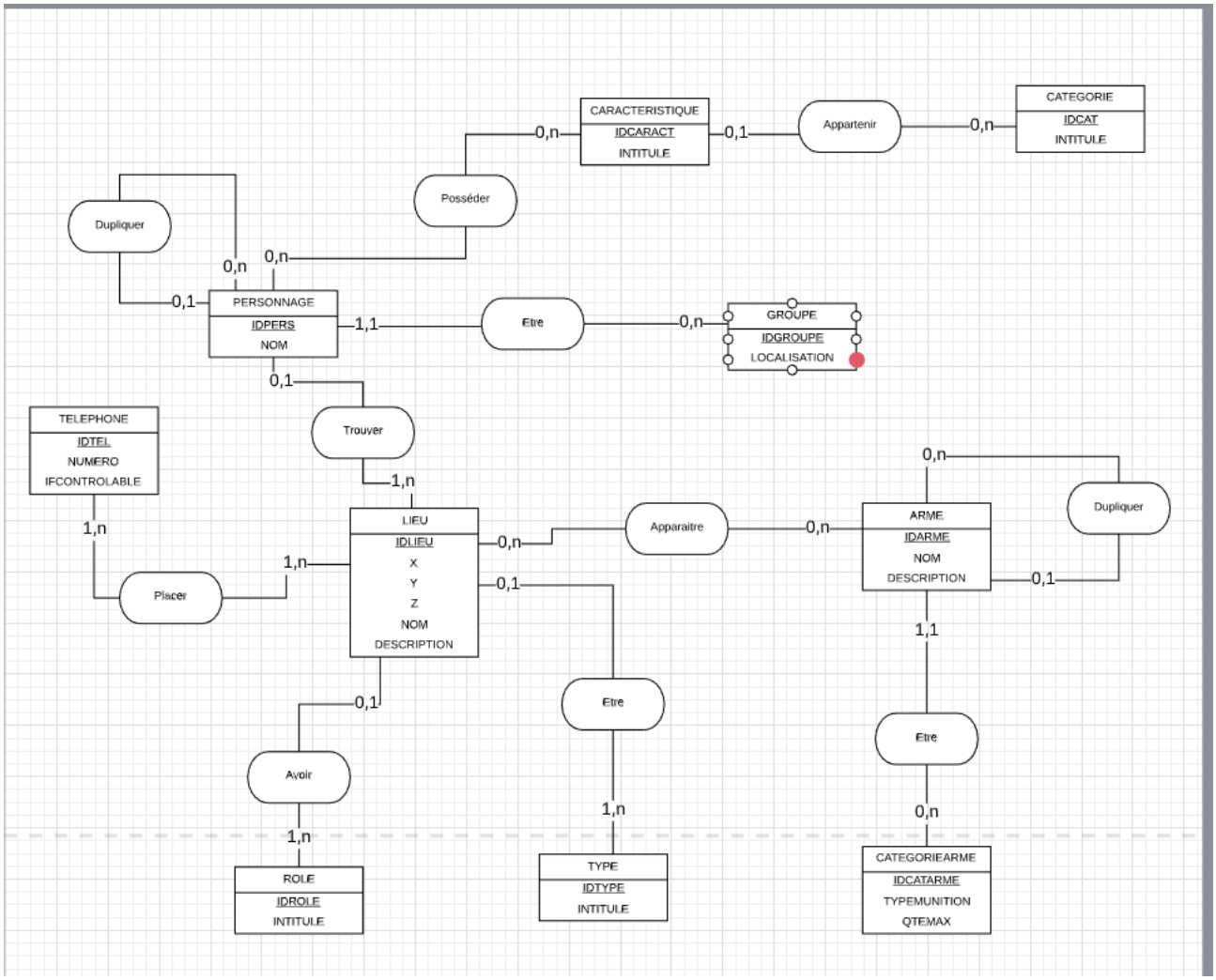
Les lignes téléphoniques qui ne sont pas sous le contrôle des machines, représentent le seul moyen, pour un homme libéré, de sortir de la vraie matrice. Dans un but d'exercice, la matrice d'entraînement doit contenir des téléphones, placés aléatoirement dans différents lieux (éventuellement plusieurs par lieu). Chaque téléphone est repérable par son numéro et il faut connaître le lieu où ils se trouve. Il faut aussi savoir si le téléphone est contrôlé par les machines. Certains types de lieux (escalier, couloir…) ne peuvent recevoir un téléphone.

Travail à réaliser :

1.1 Construire le MCD nécessaire à l'élaboration de la création de la matrice d'entraînement.
1.2 Traduire le MCD en MPD

Partie 1- – conception, création et remplissage de la BD initiale

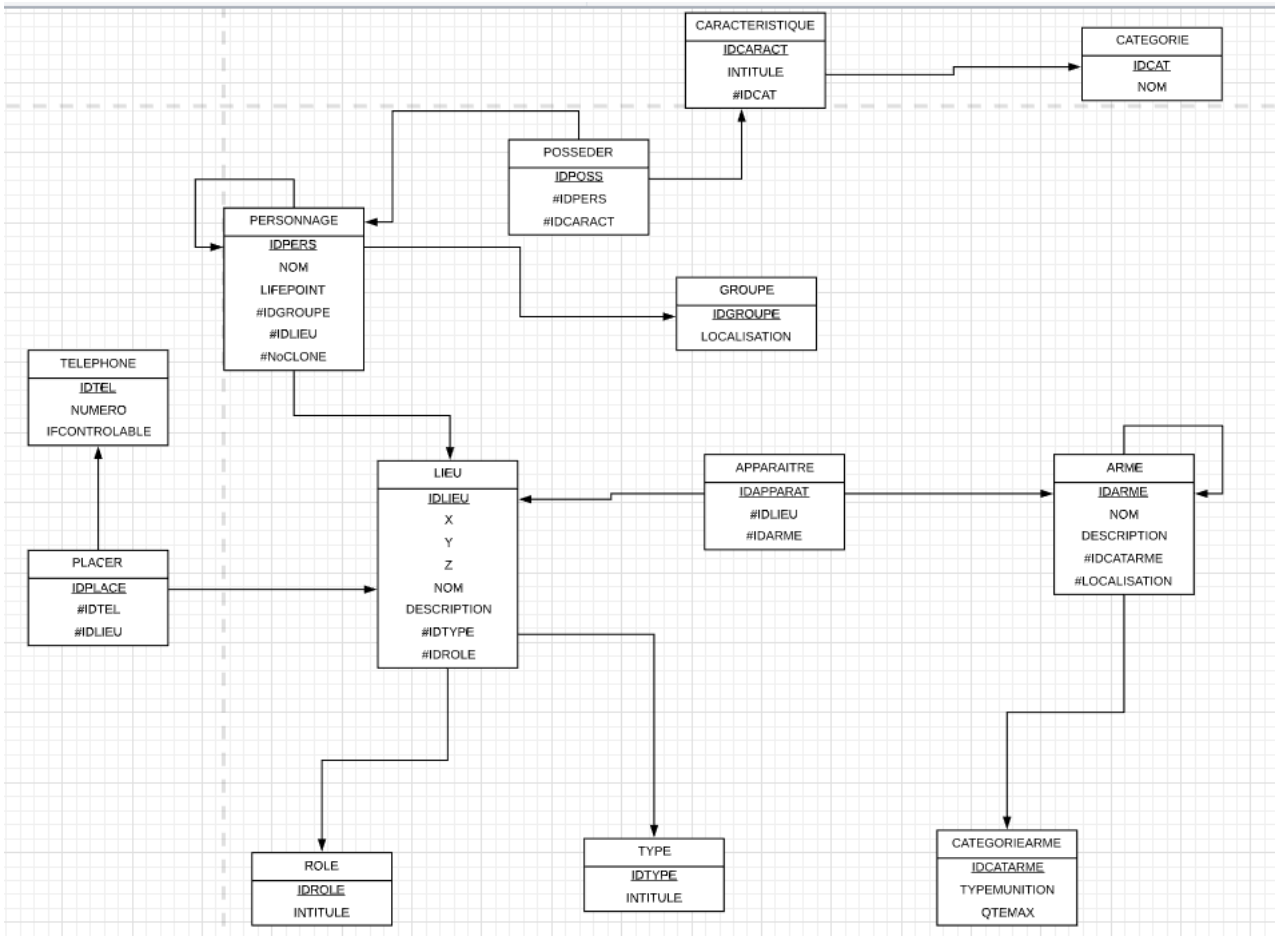
MCD version 1



Un champ « NOM » a été oublié, puis rajouté dans la table GROUPE

Un champ « LIFEPOINT » a été oublié, puis rajouté dans la table PERSONNAGE

MLD version 1



Requêtes de création de la bdd Matrix

Création de la bdd :

CREATE DATABASE Matrix;

Création des tables :

Table TYP fait référence à TYPE
Table ROL fait référence à ROLE

```
 CREATE TABLE dbo.LIEU
(
IDLIEU int identity(0, 1) not null,
X int not null,
Y int not null,
Z int not null,
NOM varchar(50) not null,
DESCRIP varchar(50),
constraint PK_LIEU primary key(IDLIEU)
);
```

```
CREATE TABLE dbo.ROL
(
IDROL int identity(0, 1) not null,
NOM varchar(50) not null,
constraint PK_ROL primary key(IDROL)
);
CREATE TABLE dbo.TYP
(
IDTYP int identity(0, 1) not null,
NOM varchar(50) not null,
constraint PK_TYP primary key(IDTYP)
);
CREATE TABLE dbo.CATEGORIE_ARME
(
IDCATARME int identity(0, 1) not null,
TYPE_MUNITION varchar(50) not null,
QTEMAX int not null,
constraint PK_CATEGORIE_ARME primary key(IDCATARME)
);
CREATE TABLE dbo.ARME
(
IDARME int identity(0, 1) not null,
NOM varchar(50) not null,
DESCRIP varchar(50),
constraint PK_ARME primary key(IDARME)
);
CREATE TABLE dbo.CARACTERISTIQUE
(
IDCARACT int identity(0, 1) not null,
INTITULE varchar(50) not null,
constraint PK_CARACTERISTIQUE primary key(IDCARACT)
);
CREATE TABLE dbo.CATEGORIE
(
IDCAT int identity(0, 1) not null,
NOM varchar(50) not null,
constraint PK_CATEGORIE primary key(IDCAT)
);
CREATE TABLE dbo.GROUPE
(
IDGROUPE int identity(0, 1) not null,
LOCALISATION varchar(50) not null,
constraint PK_GROUPE primary key(IDGROUPE)
);
CREATE TABLE dbo.TELEPHONE
(
IDTEL int identity(0, 1) not null,
CONTROLABLE bit not null,
constraint PK_TELEPHONE primary key(IDTEL)
```

```
);
CREATE TABLE dbo.PERSONNAGE
(
IDPERS int identity(0, 1) not null,
NOM varchar(50)
constraint PK_PERSONNAGE primary key(IDPERS)
);
CREATE TABLE dbo.POSSEDER
(
IDPOSS int identity(0, 1) not null,
constraint PK_POSSEDER primary key(IDPOSS)
);
CREATE TABLE dbo.PLACER
(
IDPLACE int identity(0, 1) not null,
constraint PK_PLACER primary key(IDPLACE)
);
CREATE TABLE dbo.APPARAITRE
(
IDAPPARAT int identity(0, 1) not null,
constraint PK_APPARAITRE primary key(IDAPPARAT)
);

USE Matrix;

ALTER TABLE dbo.POSSEDER
ADD IDPERS int;

ALTER TABLE dbo.POSSEDER
ADD CONSTRAINT FK_POSSEDER_PERSONNAGE
FOREIGN KEY (IDPERS) REFERENCES dbo.PERSONNAGE(IDPERS);

ALTER TABLE dbo.POSSEDER
ADD IDCARACT int;

ALTER TABLE POSSEDER
ADD CONSTRAINT FK_POSSEDER_CARACTERISTIQUE
FOREIGN KEY (IDCARACT) REFERENCES dbo.CARACTERISTIQUE(IDCARACT);

ALTER TABLE dbo.CARACTERISTIQUE
ADD IDCAT int;

ALTER TABLE CARACTERISTIQUE
ADD CONSTRAINT FK_CARACTERISTIQUE_CATEGORIE
FOREIGN KEY (IDCAT) REFERENCES CATEGORIE(IDCAT);

ALTER TABLE dbo.PERSONNAGE
ADD IDGROUPE int;

ALTER TABLE PERSONNAGE
ADD CONSTRAINT FK_PERSONNAGE_GROUPE
FOREIGN KEY (IDGROUPE) REFERENCES GROUPE(IDGROUPE);
```

```
ALTER TABLE dbo.PERSONNAGE
ADD IDLIEU int;

ALTER TABLE PERSONNAGE
ADD CONSTRAINT FK_LIEU
FOREIGN KEY (IDLIEU) REFERENCES LIEU(IDLIEU);

ALTER TABLE dbo.PLACER
ADD IDTEL int;

ALTER TABLE PLACER
ADD CONSTRAINT FK_TELEPHONE
FOREIGN KEY (IDTEL) REFERENCES TELEPHONE(IDTEL);

ALTER TABLE dbo.PLACER
ADD IDLIEU int;

ALTER TABLE PLACER
ADD CONSTRAINT FK_LIEU
FOREIGN KEY (IDLIEU) REFERENCES LIEU(IDLIEU);

ALTER TABLE dbo.LIEU
ADD IDTYP int;

ALTER TABLE LIEU
ADD CONSTRAINT FK_TYP
FOREIGN KEY (IDTYP) REFERENCES TYP(IDTYP);

ALTER TABLE dbo.LIEU
ADD IDROL int;

ALTER TABLE LIEU
ADD CONSTRAINT FK_ROL
FOREIGN KEY (IDROL) REFERENCES ROL(IDROL);

ALTER TABLE dbo.APPARAITRE
ADD IDLIEU int;

ALTER TABLE APPARAITRE
ADD CONSTRAINT FK_LIEU
FOREIGN KEY (IDLIEU) REFERENCES LIEU(IDLIEU);

ALTER TABLE dbo.APPARAITRE
ADD IDARME int;

ALTER TABLE APPARAITRE
ADD CONSTRAINT FK_ARME
FOREIGN KEY (IDARME) REFERENCES ARME(IDARME);

ALTER TABLE dbo.ARME
ADD IDCATARME int;

ALTER TABLE ARME
ADD CONSTRAINT FK_CATEGORIE_ARME
FOREIGN KEY (IDCATARME) REFERENCES CATEGORIE_ARME(IDCATARME);
```

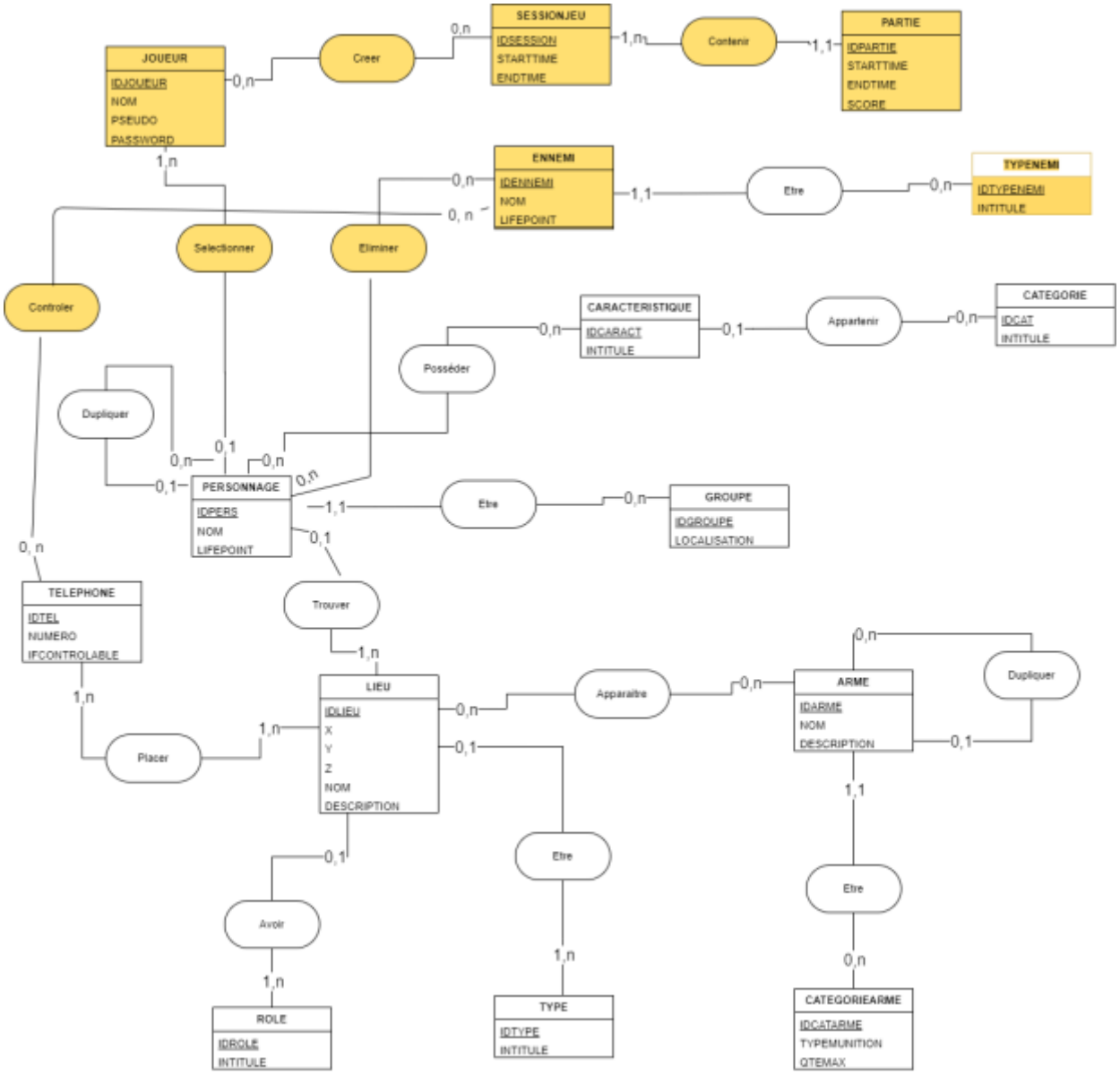
Insertion des données

```
INSERT INTO CATEGORIE VALUES ('Armurie');
INSERT INTO CATEGORIE VALUES ('Chapeau');


INSERT INTO CARACTERISTIQUE VALUES ('Armure du roi Arthur', 1);
INSERT INTO CARACTERISTIQUE VALUES ('Cape magique', 1);

INSERT INTO POSSEDER VALUES (1, 1);
INSERT INTO POSSEDER VALUES (1, 2);
```

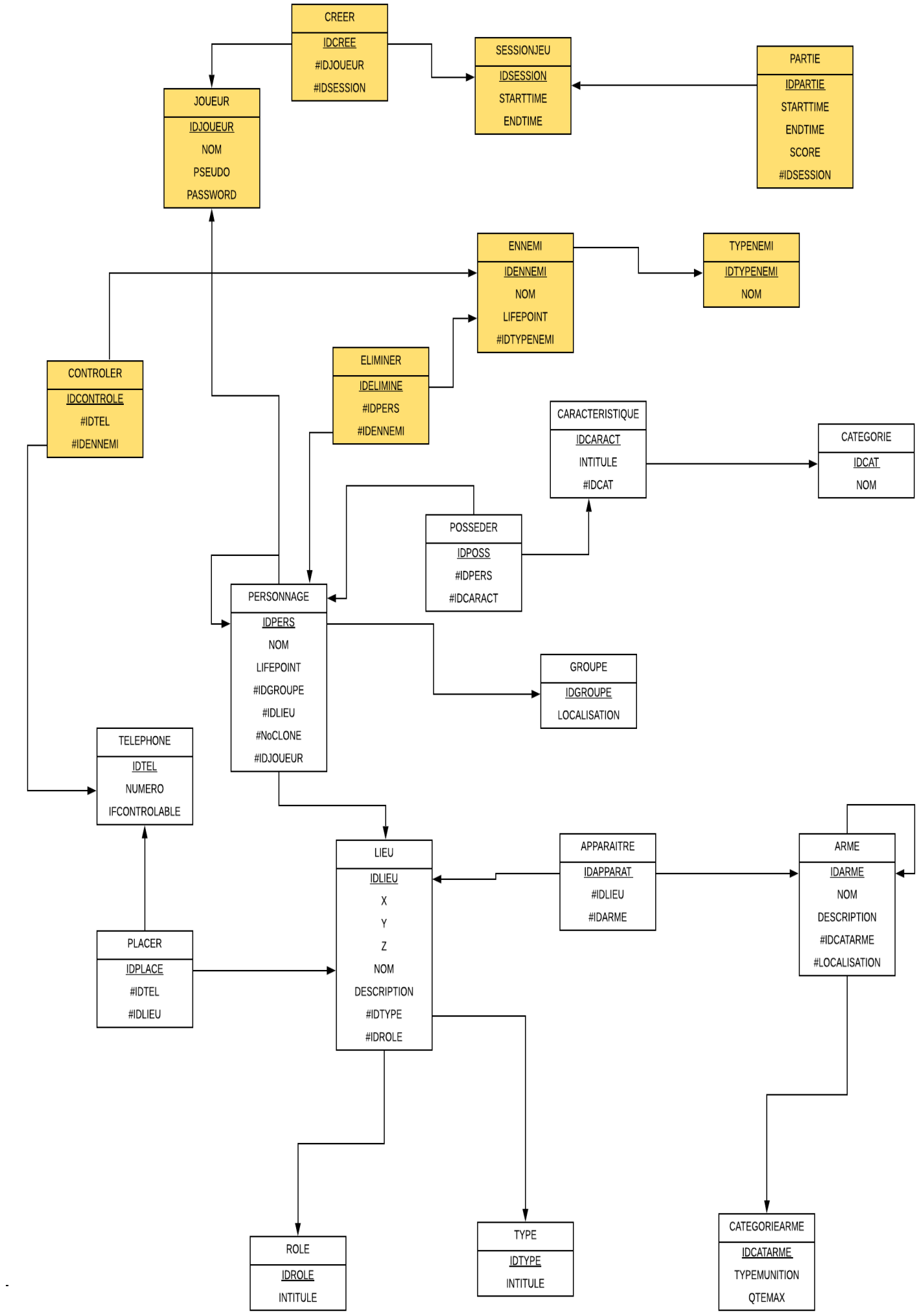
```
INSERT INTO TYP VALUES ('salle');
INSERT INTO TYP VALUES ('rue');
INSERT INTO TYP VALUES ('ville détruite');

INSERT INTO GROUPE VALUES ('', 'fictif');
INSERT INTO GROUPE VALUES ('', 'réel');


INSERT INTO ROL VALUES ('entrainement combat');
INSERT INTO ROL VALUES ('combat');
INSERT INTO ROL VALUES ('vendetta');

INSERT INTO CATEGORIE_ARME VALUES ('M10 rafale', 45);
INSERT INTO CATEGORIE_ARME VALUES ('Explosif', 5);

INSERT INTO ARME VALUES ('Ak47', 'mitraillette', 1);
INSERT INTO ARME VALUES ('Lance-grenade', 'portée de 100m', 2);

INSERT INTO LIEU VALUES (20, 10, 5, 'Cité perdue', 'grande bataille', 3, 3);
INSERT INTO LIEU VALUES (1, 1, 1, 'entraînement boot', 'salle entraînement', 1, 1);


INSERT INTO APPARAITRE VALUES (1, 1);
INSERT INTO APPARAITRE VALUES (2, 1);
INSERT INTO APPARAITRE VALUES (1, 2);

INSERT INTO TELEPHONE VALUES (0, '5945553');
INSERT INTO TELEPHONE VALUES (1, '5940663');

INSERT INTO PLACER VALUES (1, 1);
INSERT INTO PLACER VALUES (2, 1);

INSERT INTO APPARAITRE VALUES (1, 1);
```

PARTIE 2 - les modifications d'une BD existante et l'ajout de contraintes complémentaires

MCD version 2



MLD version 2

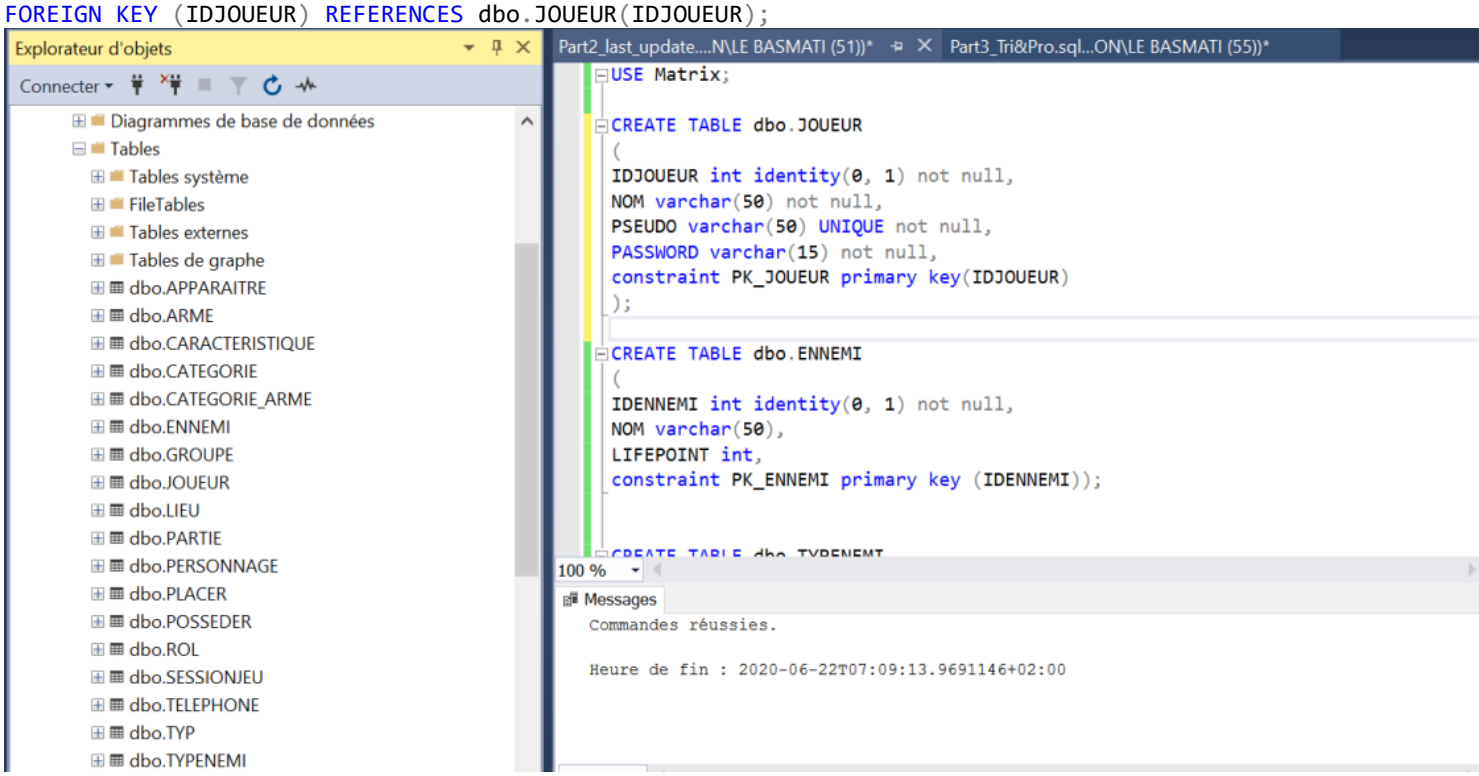

CREER
IDCREE
#IDJOUEUR
#IDSESSION
SESSIONJEU
IDSESSION
STARTTIME
ENDTIME
PARTIE
IDPARTIE
STARTTIME
ENDTIME
SCORE
#IDSESSION
JOUEUR
IDJOUEUR
NOM
PSEUDO
PASSWORD
ENNEMI
IDENNEMI
NOM
LIFEPOINT
#IDTYPENEMI
TYPENEMI
IDTYPENEMI
NOM
CONTROLER
IDCONTROLE
#IDTEL
#IDENNEMI
ELIMINER
IDELIMINE
#IDPERS
#IDENNEMI
CARACTERISTIQUE
IDCARACT
INTITULE
#IDCAT
CATEGORIE
IDCAT
NOM
POSSEDER
IDPOSS
#IDPERS
#IDCARACT
PERSONNAGE
IDPERS
NOM
LIFEPOINT
#IDGROUPE
#IDLIEU
#NoCLONE
#IDJOUEUR
GROUPE
IDGROUPE
LOCALISATION
TELEPHONE
IDTEL
NUMERO
IFCONTROLABLE
LIEU
IDLIEU
X
Y
Z
NOM
DESCRIPTION
#IDTYPE
#IDROLE
APPARAITRE
IDAPPARAT
#IDLIEU
#IDARME
ARME
IDARME
NOM
DESCRIPTION
#IDCATARME
#LOCALISATION
PLACER
IDPLACE
#IDTEL
#IDLIEU
ROLE
IDROLE
INTITULE
TYPE
IDTYPE
INTITULE
CATEGORIEARME
IDCATARME
TYPEMUNITION
QTEMAX

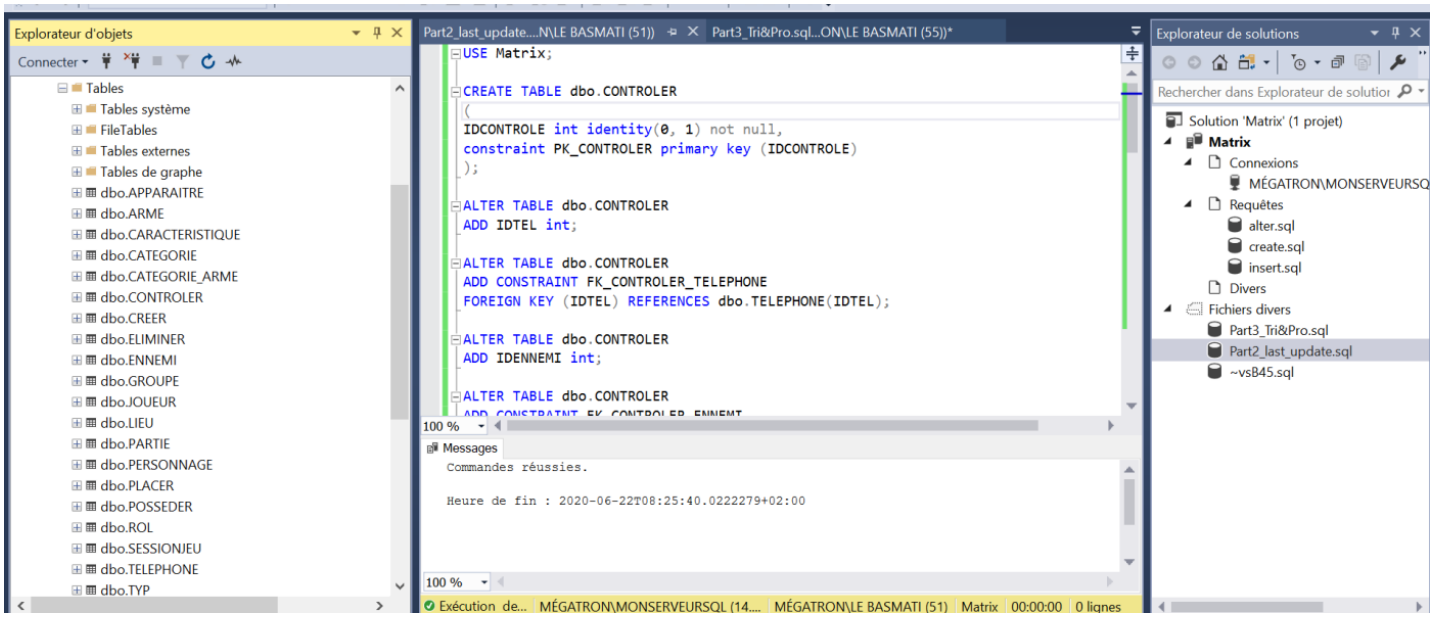
Requêtes de modification de la bdd Matrix

```
USE Matrix;

ALTER TABLE dbo.TELEPHONE
ADD NIUMERO VARCHAR(8)

ALTER TABLE dbo.GROUPE
ADD NOM VARCHAR(50);

ALTER TABLE dbo.LIEU
ADD IDTYP int;

ALTER TABLE dbo.LIEU
ADD CONSTRAINT FK_LIEU_TYPE
FOREIGN KEY (IDTYP) REFERENCES dbo.TYP(IDTYP);

ALTER TABLE dbo.LIEU
ADD IDROL int;

ALTER TABLE dbo.LIEU
ADD CONSTRAINT FK_LIEU_ROLE
FOREIGN KEY (IDROL) REFERENCES dbo.ROL(IDROL);


CREATE TABLE dbo.JOUEUR
(
IDJOUEUR int identity(0, 1) not null,
NOM varchar(50) not null,
PSEUDO varchar(50) UNIQUE not null,
PASSWORD varchar(15) not null,
constraint PK_JOUEUR primary key(IDJOUEUR)
);

CREATE TABLE dbo.ENNEMI
(
IDENNEMI int identity(0, 1) not null,
NOM varchar(50),
LIFEPOINT int,
constraint PK_ENNEMI primary key (IDENNEMI));


CREATE TABLE dbo.TYPENEMI
(
IDTYPENEMI int identity(0, 1) not null,
NOM varchar(50) UNIQUE not null,
constraint PK_TYPENEMI primary key (IDTYPENEMI));

CREATE TABLE dbo.SESSIONJEU
(
IDSESSION int identity(0, 1) not null,
STARTIME  DATETIME,
ENDTIME  DATETIME,
constraint PK_SESSIONJEU primary key (IDSESSION));

CREATE TABLE dbo.PARTIE
(
IDPARTIE int identity(0, 1) not null,
STARTIME  DATETIME,
ENDTIME  DATETIME,
SCORE int,
constraint PK_PARTIE primary key (IDPARTIE)
);

CREATE TABLE dbo.CONTROLER
(
```

```
IDCONTROLE int identity(0, 1) not null,
constraint PK_CONTROLER primary key (IDCONTROLE)
);

ALTER TABLE dbo.CONTROLER
ADD IDTEL int;

ALTER TABLE dbo.CONTROLER
ADD CONSTRAINT FK_CONTROLER_TELEPHONE
FOREIGN KEY (IDTEL) REFERENCES dbo.TELEPHONE(IDTEL);

ALTER TABLE dbo.CONTROLER
ADD IDENNEMI int;

ALTER TABLE dbo.CONTROLER
ADD CONSTRAINT FK_CONTROLER_ENNEMI
FOREIGN KEY (IDENNEMI) REFERENCES dbo.ENNEMI(IDENNEMI);


CREATE TABLE dbo.ELIMINER
(
IDELIMINE int identity(0, 1) not null,
constraint PK_ELIMINER primary key (IDELIMINE)
);

ALTER TABLE dbo.ELIMINER
ADD IDPERS int;

ALTER TABLE dbo.ELIMINER
ADD CONSTRAINT FK_ELIMINER_PERSONNAGE
FOREIGN KEY (IDPERS) REFERENCES dbo.PERSONNAGE(IDPERS);

ALTER TABLE dbo.ELIMINER
ADD IDENNEMI int;

ALTER TABLE dbo.ELIMINER
ADD CONSTRAINT FK_ELIMINER_ENNEMI
FOREIGN KEY (IDENNEMI) REFERENCES dbo.ENNEMI(IDENNEMI);


CREATE TABLE dbo.CREER
(
IDCREE int identity(0, 1) not null,
constraint PK_CREER primary key (IDCREE)
);

ALTER TABLE dbo.CREER
ADD IDJOUEUR int;

ALTER TABLE dbo.CREER
ADD CONSTRAINT FK_CREER_JOUEUR
FOREIGN KEY (IDJOUEUR) REFERENCES dbo.JOUEUR(IDJOUEUR);

ALTER TABLE dbo.CREER
ADD IDSESSION int;

ALTER TABLE dbo.CREER
ADD CONSTRAINT FK_CREER_SESSIONJEU
FOREIGN KEY (IDSESSION) REFERENCES dbo.SESSIONJEU(IDSESSION);

ALTER TABLE dbo.CREER
ADD IDJOUEUR int;

ALTER TABLE dbo.CREER
ADD CONSTRAINT FK_CREER_JOUEUR
FOREIGN KEY (IDJOUEUR) REFERENCES dbo.JOUEUR(IDJOUEUR);

ALTER TABLE dbo.PARTIE
```

```
ADD IDSESSION int;

ALTER TABLE dbo.PARTIE
ADD CONSTRAINT FK_PARTIE_SESSIONJEU
FOREIGN KEY (IDSESSION) REFERENCES dbo.SESSIONJEU(IDSESSION);

ALTER TABLE dbo.ENNEMI
ADD IDTYPENEMI int;

ALTER TABLE dbo.ENNEMI
ADD CONSTRAINT FK_ENNEMI_TYPENEMI
FOREIGN KEY (IDTYPENEMI) REFERENCES dbo.TYPENEMI(IDTYPENEMI);

ALTER TABLE dbo.PERSONNAGE
ADD IDJOUEUR int;

ALTER TABLE dbo.PERSONNAGE
ADD CONSTRAINT FK_PERSONNAGE_JOUEUR
FOREIGN KEY (IDJOUEUR) REFERENCES dbo.JOUEUR(IDJOUEUR);
```





Insertion des données dans la nouvelle table

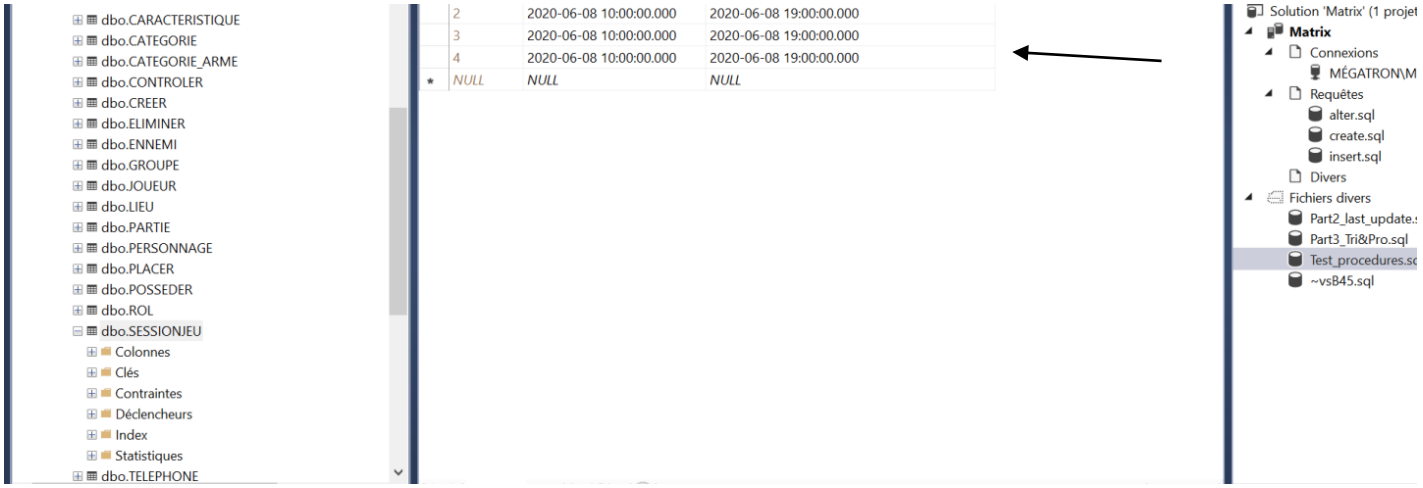
```
USE Matrix;

INSERT INTO PERSONNAGE VALUES ('Zeya', 50, 2, 1, 1);
INSERT INTO PERSONNAGE VALUES ('Optimus', 100, 1, 1, 2);

INSERT INTO TYPENEMI VALUES ('Boot');
INSERT INTO TYPENEMI VALUES ('Boss');

INSERT INTO ENNEMI VALUES ('Agent', 150, 1);
INSERT INTO ENNEMI VALUES ('Agent1', 150, 1);
INSERT INTO ENNEMI VALUES ('Smith', 500, 2);

INSERT INTO JOUEUR VALUES ('Morphéus', 'Kran23', 'passtest1');
INSERT INTO JOUEUR VALUES ('Néo', 'TCH666', 'oday203');

INSERT INTO ELIMINER VALUES (1, 1);
INSERT INTO ELIMINER VALUES (2, 2);
INSERT INTO ELIMINER VALUES (1, 2);
INSERT INTO ELIMINER VALUES (2, 1);


INSERT INTO SESSIONJEU VALUES ('2020-06-06 09:00', '2020-06-09 11:00');
INSERT INTO SESSIONJEU VALUES ('2020-07-07 09:00', '2020-06-09 11:00');

INSERT INTO PARTIE VALUES ('2020-06-06 09:00', '2020-06-09 11:00', 100, 2);
INSERT INTO PARTIE VALUES ('2020-06-06 09:00', '2020-06-09 11:00', 90, 1);
INSERT INTO PARTIE VALUES ('2020-06-06 23:00', '2020-06-09 23:30', 50, 1);

INSERT INTO CREER VALUES (1, 1);
INSERT INTO CREER VALUES (2, 2);
```

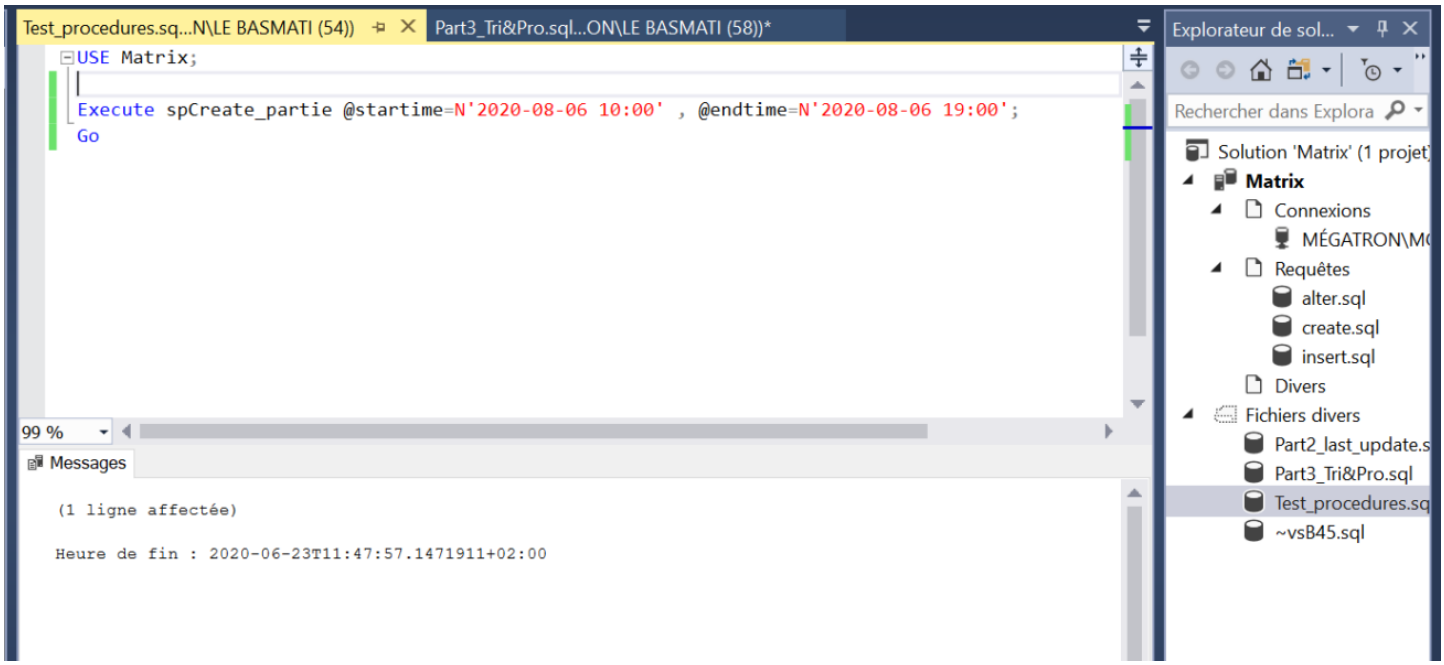
PARTIE 3 – La préparation de code dans la BD

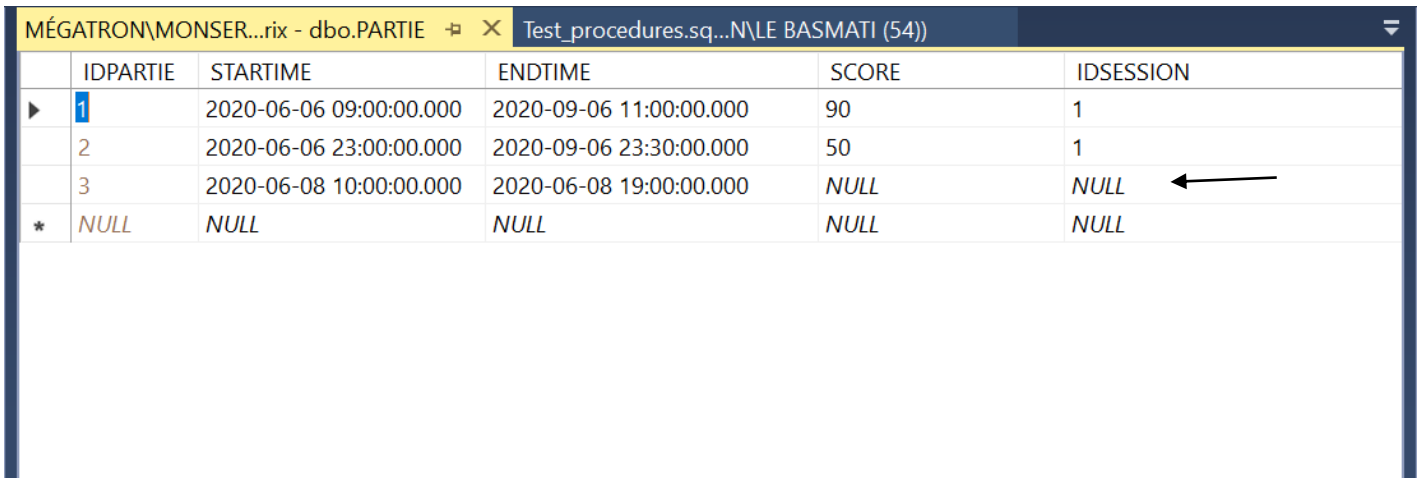
Procédures

```
Create Procedure spCreate_Session_jeu
	@starttime DateTime, @endtime DateTime

	AS
		DECLARE @start DateTime, @end DateTime,
			@startHour DateTime, @endHour DateTime;

		Select @start = SESSIONJEU.STARTIME FROM SESSIONJEU
			WHERE STARTIME = @starttime;

		Select @end = SESSIONJEU.ENDTIME FROM SESSIONJEU
			WHERE ENDTIME = @endtime;

		if @end < @start
			PRINT 'échec de création de jeu';
		else
			if @end = @start

			Select @startHour = FORMAT(CAST(SESSIONJEU.STARTIME AS datetime2), N'HH:mm')
			FROM SESSIONJEU
			WHERE STARTIME = @starttime;

			Select @endHour = FORMAT(CAST(SESSIONJEU.ENDTIME AS datetime2), N'HH:mm')
			FROM SESSIONJEU
			WHERE ENDTIME = @endtime;
```

```
                        if @endHour < @startHour
                                PRINT 'échec de création de jeu';
                        else
                                INSERT INTO SESSIONJEU VALUES (@starttime, @endtime);

                if @start < @end
                        INSERT INTO SESSIONJEU VALUES (@starttime, @endtime);
        Go

Create Procedure spCreate_partie
    @startime DateTime, @endtime Datetime

    AS
        DECLARE @start DateTime, @end DateTime;

        Select @start = FORMAT(CAST(PARTIE.STARTIME AS datetime2), N'HH:mm')
        FROM PARTIE
        WHERE STARTIME = @startime;

        Select @end = FORMAT(CAST(PARTIE.ENDTIME AS datetime2), N'HH:mm')
        FROM PARTIE
        WHERE ENDTIME = @endtime;

        if @end < @start
            PRINT 'échec de création de la partie';
        else
            INSERT INTO PARTIE(STARTIME, ENDTIME) VALUES (@startime, @endtime);

    Go
```







	IDPARTIE	STARTIME	ENDTIME	SCORE	IDSESSION
▶	1	2020-06-06 09:00:00.000	2020-09-06 11:00:00.000	90	1
	2	2020-06-06 23:00:00.000	2020-09-06 23:30:00.000	50	1
	3	2020-06-08 10:00:00.000	2020-06-08 19:00:00.000	*NULL*	*NULL*
*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

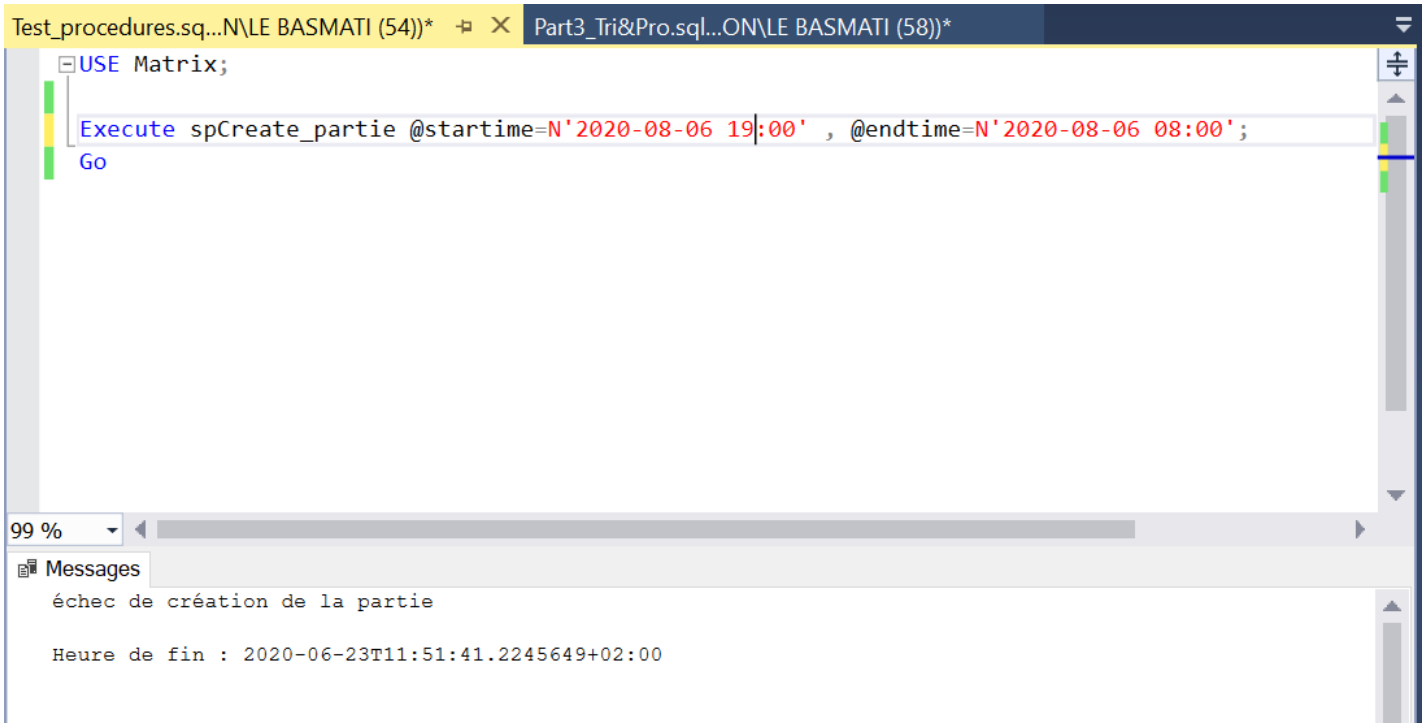
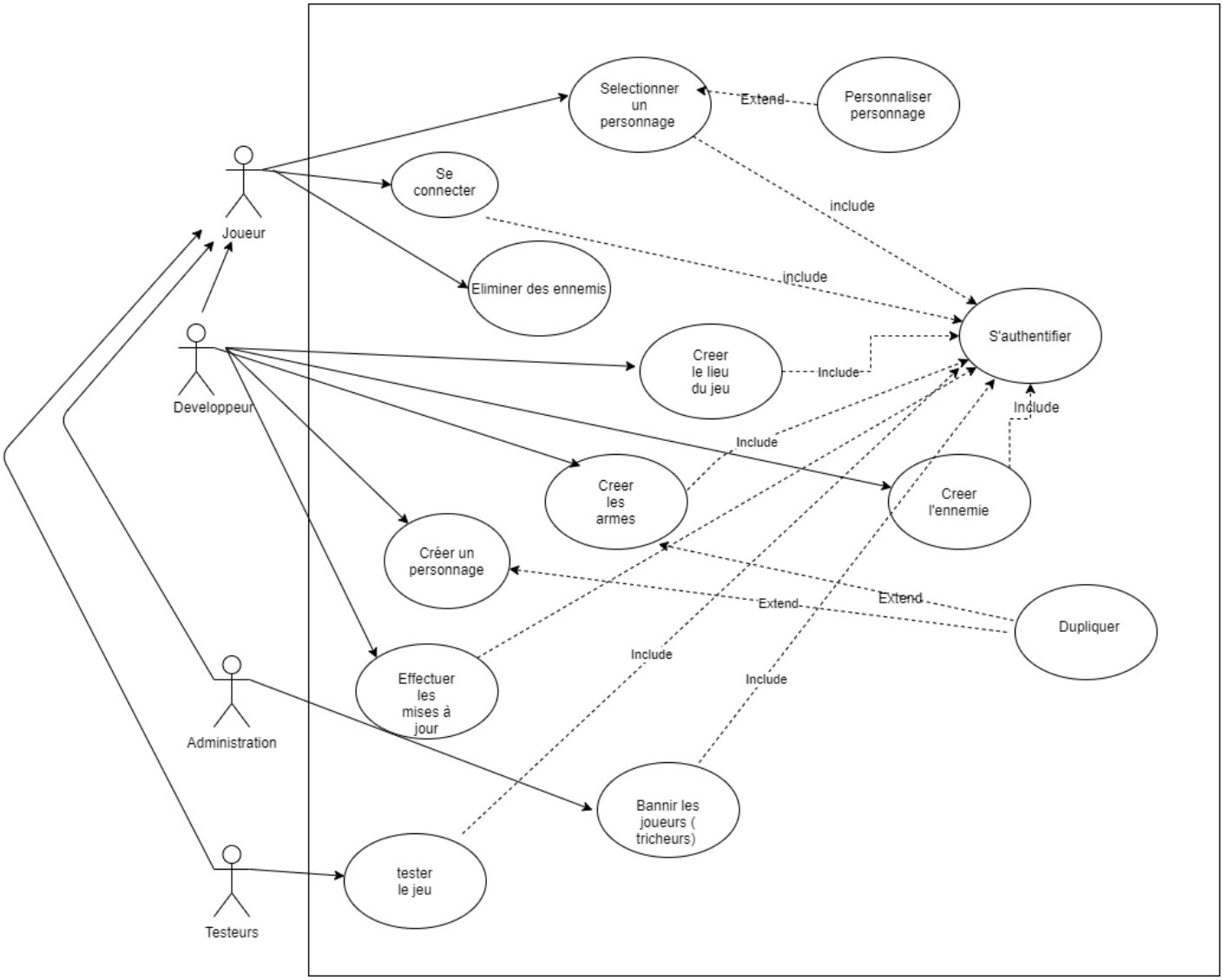


Figure i : cas où la date de fin n'est pas antérieure à la date de début

Triggers

PARTIE 4 – La sécurité de la BD

DIAGRAMME DES CAS D'UTILISATIONS



Acteurs :

- Developpeurs
- Administrateur
- Joueur
- Testeur
-

Tache	Tables à utiliser	Privilèges nécessaires
développeurs	Personnage/groupe/caracteristiques/categorie/lieu/role/ type/telephone/armes/catégories armes/ ennemi/type ennemi	Tous les accès à ces informations nécessaires
Administrateurs (s'occupe du respect des règles du jeu, effectue les mises à jour nécessaires au bon fonctionnement du jeu)	Sessions de Jeu/parties/joueur/personnages/ennemi/armes/lieu	Accès aux informations des sessions de jeu/parties, droits de voir les infos d'un joueur et de supprimer

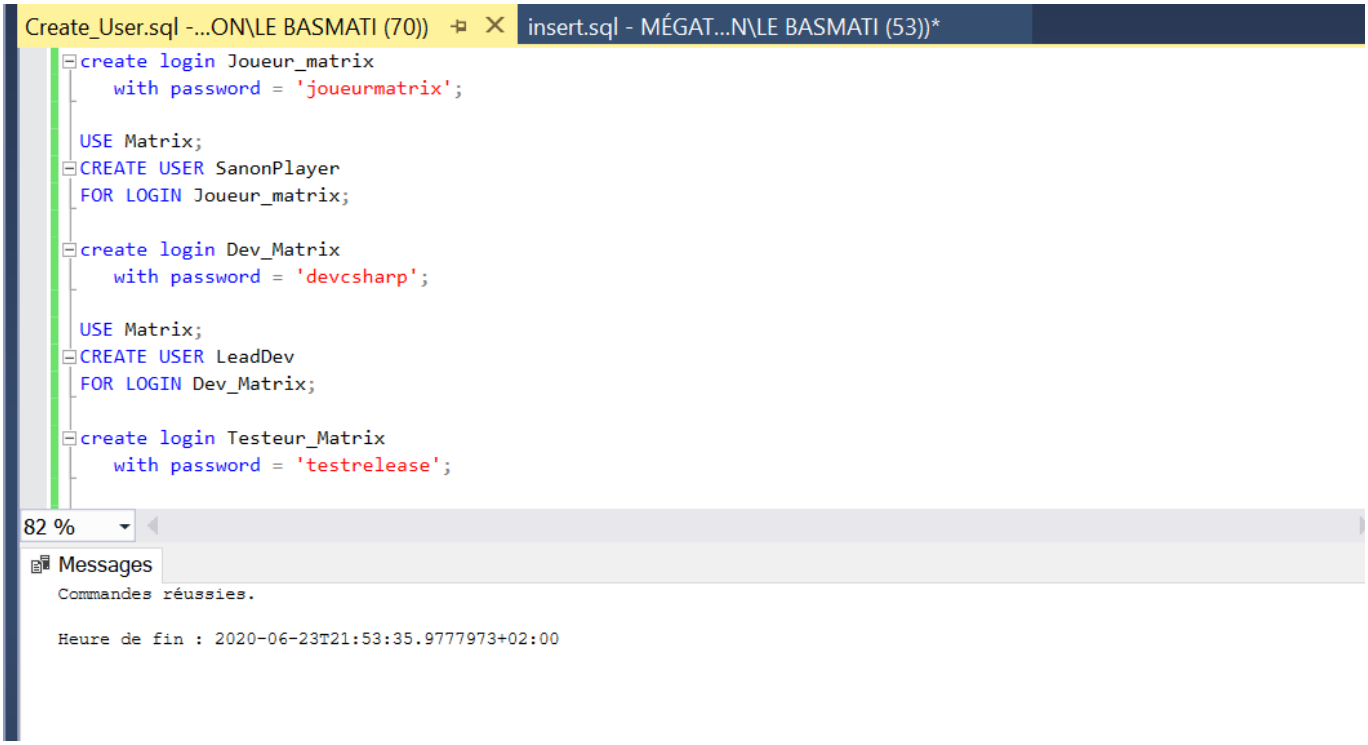
		un compte joueur
Joueur (utilisateur du jeu)	Joueur/personnage/caracteristiques/categories/armes/lieu	Peu voir certaine information sur le jeu/ a acces a toutes ses infos
testeur	Personnage/ennemi/armes/lieu	A la possibilite de voir certaines info du jeu

Tables > Acteurs	joueur	Session de jeu	partie	ennemi	personnage	téléphone	lieu	arme	Categories armes
joueur	suid	suid	suid		s	s	su	su	su
developpeurs	s	s	s	siud	siud	siud	siud	siud	siud
Administrateurs	sd	su	su	s	s	s	s	s	s
testeur	s	s	s	s	s	s	s	s	s

S= select I=insert U=update D= delete

Creation des users

```
create login Joueur_matrix
   with password = 'joueurmatrix';

USE Matrix;
CREATE USER SanonPlayer
FOR LOGIN Joueur_matrix;

create login Dev_Matrix
   with password = 'devcsharp';

USE Matrix;
CREATE USER LeadDev
FOR LOGIN Dev_Matrix;

create login Testeur_Matrix
   with password = 'testrelease';

USE Matrix;
CREATE USER TesterUI
FOR LOGIN Testeur_Matrix;

create login AdminMatrixProject
   with password = 'adminforGame';

USE Matrix;
CREATE USER Administrateur
       FOR LOGIN AdminMatrixProject;
```



Définition des accès

```
USE Matrix;

Grant select, insert, update, delete on JOUEUR to SanonPlayer;
Grant select, insert, update, delete on SESSIONJEU to SanonPlayer;
Grant select, update on PERSONNAGE to SanonPlayer;
Grant select, update on ARME to SanonPlayer;

Grant select on JOUEUR to LeadDev;
Grant select on SESSIONJEU to LeadDev;
Grant select, insert, update, delete on PERSONNAGE to LeadDev;
Grant select, insert, update, delete on ENNEMI to LeadDev;
Grant select, insert, update, delete on LIEU to LeadDev;
Grant select, insert, update, delete on ARME to LeadDev;

Grant select, insert, update, delete on JOUEUR to TesterUI;
Grant select, insert, update, delete on SESSIONJEU to TesterUI;
Grant select, insert, update, delete on PERSONNAGE to TesterUI;
Grant select, insert, update, delete on ENNEMI to TesterUI;
Grant select, insert, update, delete on LIEU to TesterUI;
Grant select, insert, update, delete on ARME to TesterUI;


Grant select, delete on JOUEUR to Administrateur;
Grant select, delete on SESSIONJEU to Administrateur;
Grant select, update, delete on PERSONNAGE to Administrateur;
Grant select, update, delete on ENNEMI to Administrateur;
Grant select, update, delete on LIEU to Administrateur;
Grant select, update, delete on ARME to Administrateur;
```

Access.sql - MÉGAT...N\LE BASMATI (62)) | Create_User.sql -...ON\LE BASMATI (70))

```
USE Matrix;

Grant select, insert, update, delete on JOUEUR to SanonPlayer;
Grant select, insert, update, delete on SESSIONJEU to SanonPlayer;
Grant select, update on PERSONNAGE to SanonPlayer;
Grant select, update on ARME to SanonPlayer;

Grant select on JOUEUR to LeadDev;
Grant select on SESSIONJEU to LeadDev;
Grant select, insert, update, delete on PERSONNAGE to LeadDev;
Grant select, insert, update, delete on ENNEMI to LeadDev;
Grant select, insert, update, delete on LIEU to LeadDev;
Grant select, insert, update, delete on ARME to LeadDev;

Grant select, insert, update, delete on JOUEUR to TesterUI;
Grant select, insert, update, delete on SESSIONJEU to TesterUI;
Grant select, insert, update, delete on PERSONNAGE to TesterUI;
Grant select, insert, update, delete on ENNEMI to TesterUI;
Grant select, insert, update, delete on LIEU to TesterUI;
Grant select, insert, update, delete on ARME to TesterUI;


Grant select, delete on JOUEUR to Administrateur;
Grant select, delete on SESSIONJEU to Administrateur;
```

82 %

Messages

```
Commandes réussies.

Heure de fin : 2020-06-23T22:18:42.4653140+02:00
```

Quelques requêtes d'utilisateurs

```
USE Matrix;


EXECUTE AS USER = 'LeadDev';
SELECT * FROM JOUEUR
```

Résultats | Messages

IDJOUEUR	NOM	PSEUDO	PASSWORD
0	Morphéus	Kran23	passtest1
1	Néo	TCH666	oday203

PARTIE 5 – La protection de données

Partie 5.1: la protection de données

Données sensibles:

- Informations personnels des utilisateurs
- Informations personnels des employés
- Données du jeu

Personnes concerne:

- Administrateurs
- Joueurs
- Developpeurs

Materiels de stockage sécurisé de base de données:

- Un Hardware Security Module (en français, boîte noire transactionnelle ou BNT) appareil considéré comme inviolable offrant des fonctions cryptographiques. Il s'agit d'un matériel électronique offrant un service de sécurité qui consiste à générer, stocker et protéger des clefs cryptographiques.

Software et type de communication :
Un HSM peut ainsi être utilisé comme une solution fonctionnelle pour sécuriser les transactions distantes opérées par un système télébillettique d'un opérateur de transport.

Certains SGBDR comme SQL Server, MySQL ou Oracle DB permettent d'utiliser un HSM pour chiffrer les données sensibles d'une base de données.

Liste des risques	Effet sur l'activité	Mesures preventives	Mesures curatives
Attaque externe (hack)	Non continuite de service/ perte de fiabilités de l'entreprise	Mise en place d'un firwall/antivirus/ Mise en place de mots de passe forts	Reboot du système Isoler /changer de port au matériel infectes
Attaque interne	Perte d'information/ perte de crédibilités de l'entreprise	Rappel du respect des règles de sécurités aux utilisateurs	
Pannes de materiel	Arrête l'activité / dérèglement de l'organisation	Mettre en place un site back up/ Effectuer des sauvegardes de la base de données / Redondance des matériaux importants	Faire marche le matériel de back up ou en redondance

injection sql	Perte/modification/suppression des données possibles Vole d'information, néfaste pour l'entreprise	Sécurisé le développement de la base de données Crypter certaine données	Avoir une base de données dupliquée pour récupérer ses données

Selon la loi RGPD, si les utilisateurs demandent la suppression immédiate de leurs données personnelles, les données devront donc être effacer directement. Ainsi, puisque le joueur n'a pas les acces et les droits d'accès a la base de données cela s'effectuera sous forme de demande auprès des administrateurs
En cas de piratage du système, l'ensemble des employés et des utilisateurs doivent être informe à travers un communique de la maintenance des serveurs , et dans certains cas selon la gravite de l'attaque d'un potentiel arrêt de l'activité.

Partie 5.2 :
Chartre de sécurité

- ne jamais confier son identifiant/mot de passe à un tiers ;
- Ne jamais laisser son ordinateur déverrouillé en cas d'absence ;
- ne pas modifier les paramétrages du poste de travail ;
- Mettre en place des mots de passe robustes et encore plus robustes lorsque l'utilisateur a beaucoup d'accès et manipule beaucoup d'informations tel que les administrateurs et les développeurs.
- ne pas installer, copier, modifier, détruire des logiciels sans autorisation Les recommandations pour les sauvegardes sont :
- chiffrer les sauvegardes elles-mêmes / les données à la source ;
- prévoir un stockage dans un lieu sécurisé, éventuellement sur un site extérieur, dans des coffres ignifugés et étanches ;
- Mettre en place des procédures de stockage automatises de la base de données (triggers) ;
- suivre des règles en adéquation avec la politique de sécurité pour le convoyage éventuel des sauvegardes.
- Effectuer des sauvegardes fréquentes pour éviter la perte d'information. Selon le volume d'informations à sauvegarder, il peut être opportun de prévoir des sauvegardes incrémentales avec une fréquence quotidienne et des sauvegardes complètes avec une fréquence moindre selon l'importance de celle-ci.
- Chiffre les sauvegardes lorsque les informations sont sensibles ;