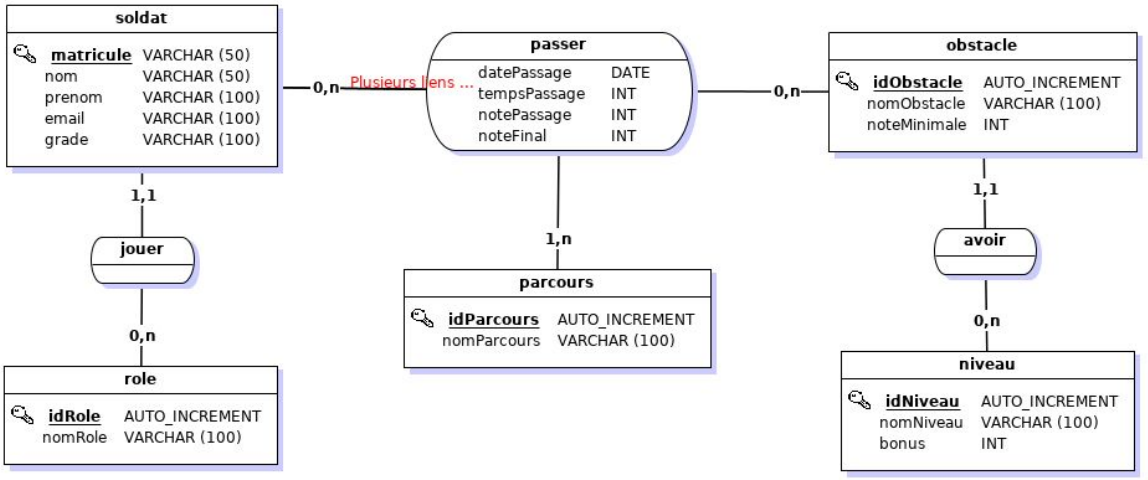
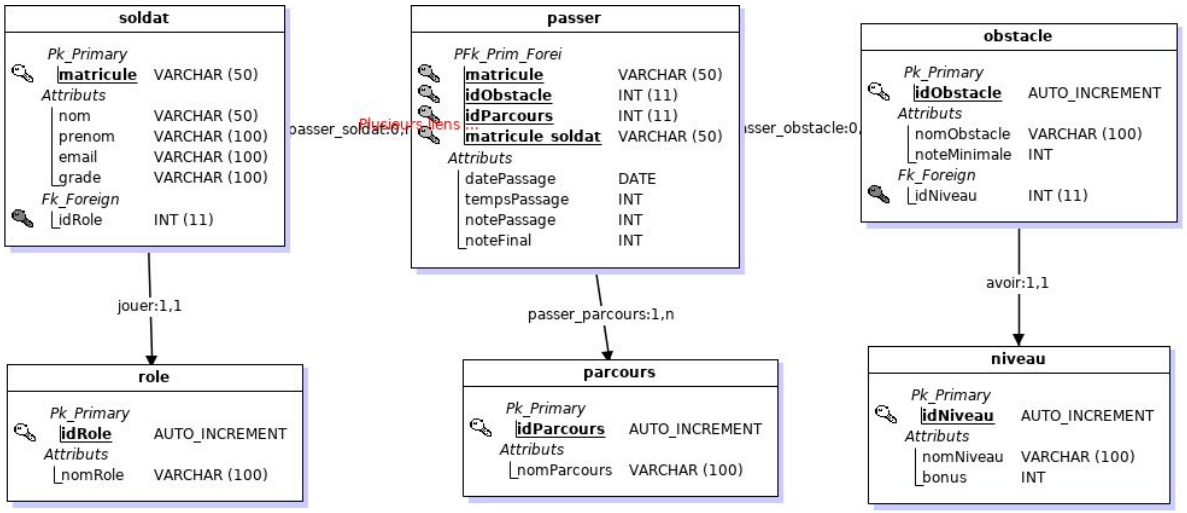
TAF: GESTION DES RÉSULTATS AU PARCOURS DU COMBATTANT

Modèle Entités-Associations



Modèle Logique de données



Ces modèles(**MLD et MCD**) sont réalisés grâce à application **JMerise**

Gestion de base de donnée

Création de base de donnée

```
mysql> create database parcours_combattant;
Query OK, 1 row affected (0.00 sec)

mysql> use parcours_combattant;
Database changed
mysql>
```

Créations des tables et les relations sous mysql

➢ Table Role

```
#------------------------------------------------------------
# Table: role
#------------------------------------------------------------

CREATE TABLE role(
        idRole  Int  Auto_increment  NOT NULL ,
        nomRole Varchar (100) NOT NULL
        ,CONSTRAINT role_PK PRIMARY KEY (idRole)
)ENGINE=InnoDB;
```

➢ Table Soldat

```
#------------------------------------------------------------
# Table: soldat
#------------------------------------------------------------

CREATE TABLE soldat(
        matricule Varchar (50) NOT NULL ,
        nom       Varchar (50) NOT NULL ,
        prenom    Varchar (100) NOT NULL ,
        email     Varchar (100) NOT NULL ,
        grade     Varchar (100) NOT NULL ,
        idRole    Int NOT NULL
        ,CONSTRAINT soldat_PK PRIMARY KEY (matricule)

        ,CONSTRAINT soldat_role_FK FOREIGN KEY (idRole) REFERENCES role(idRole)
)ENGINE=InnoDB;
```

➢ Table Niveau

```
#------------------------------------------------------------
# Table: niveau
#------------------------------------------------------------

CREATE TABLE niveau(
        idNiveau  Int  Auto_increment  NOT NULL ,
        nomNiveau Varchar (100) NOT NULL ,
        bonus     Int NOT NULL
        ,CONSTRAINT niveau_PK PRIMARY KEY (idNiveau)
)ENGINE=InnoDB;
```

➢ Table Obstacle

```
#------------------------------------------------------------
# Table: obstacle
#------------------------------------------------------------

CREATE TABLE obstacle(
        idObstacle   Int  Auto_increment  NOT NULL ,
        nomObstacle  Varchar (100) NOT NULL ,
        noteMinimale Int NOT NULL ,
        idNiveau     Int NOT NULL
        ,CONSTRAINT obstacle_PK PRIMARY KEY (idObstacle)

        ,CONSTRAINT obstacle_niveau_FK FOREIGN KEY (idNiveau) REFERENCES niveau(idNiveau)
)ENGINE=InnoDB;
```

➢ Table Parcours

```
#------------------------------------------------------------
# Table: parcours
#------------------------------------------------------------

CREATE TABLE parcours(
        idParcours  Int  Auto_increment  NOT NULL ,
        nomParcours Varchar (100) NOT NULL
        ,CONSTRAINT parcours_PK PRIMARY KEY (idParcours)
)ENGINE=InnoDB;
```

➢ Table Passer

```
#------------------------------------------------------------
# Table: passer
#------------------------------------------------------------

CREATE TABLE passer(
        matricule        Varchar (50) NOT NULL ,
        idObstacle       Int NOT NULL ,
        idParcours       Int NOT NULL ,
        matricule_soldat Varchar (50) NOT NULL ,
        datePassage      Date NOT NULL ,
        tempsPassage     Int NOT NULL ,
        notePassage      Int NOT NULL ,
        noteFinal        Int NOT NULL
        ,CONSTRAINT passer_PK PRIMARY KEY (matricule,idObstacle,idParcours,matricule_soldat)

        ,CONSTRAINT passer_soldat_FK FOREIGN KEY (matricule) REFERENCES soldat(matricule)
        ,CONSTRAINT passer_obstacle0_FK FOREIGN KEY (idObstacle) REFERENCES obstacle(idObstacle)
        ,CONSTRAINT passer_parcours1_FK FOREIGN KEY (idParcours) REFERENCES parcours(idParcours)
        ,CONSTRAINT passer_soldat2_FK FOREIGN KEY (matricule_soldat) REFERENCES soldat(matricule)
)ENGINE=InnoDB;
```

Les requêtes SQL

Nous avons inséré les données des différentes tables, tout en respectant les contraints d'intégrités. Cela a été fait via **phpmyadmin**
l'aide du langage SQL affichons les informations suivantes:

1. **Afficher les soldats qui ont passé un parcours donné (*)**

```
mysql> SELECT DISTINCT s.matricule,s.nom,s.prenom
    -> FROM soldat s,passer p
    -> WHERE s.matricule=p.matricule AND p.idParcours=1;
+-----------+--------+----------+
| matricule | nom    | prenom   |
+-----------+--------+----------+
| MAT-001   | Diallo | Moussa   |
| MAT-002   | Diagne | Ibrahima |
| MAT-003   | Fall   | Bamba    |
| MAT-005   | Pam    | Issa     |
+-----------+--------+----------+
4 rows in set (0.00 sec)
```

2. **Afficher les obstacles d'un parcours (*)**

```
mysql> SELECT DISTINCT o.idObstacle,o.nomObstacle,o.noteMinimale
    -> FROM obstacle o,passer p
    -> WHERE o.idObstacle=p.idObstacle AND p.idParcours=1;
+------------+------------------+--------------+
| idObstacle | nomObstacle      | noteMinimale |
+------------+------------------+--------------+
|          1 | Echelle de corde |           13 |
|          2 | Ramper           |           13 |
|          4 | Petit mur        |           11 |
|          6 | Fosse            |           10 |
|          8 | Mur d'assaut     |           10 |
|         10 | 3 Chicanes       |           10 |
+------------+------------------+--------------+
6 rows in set (0.00 sec)
```

3. **Afficher les soldats avec leurs moyennes pour un parcours(nom soldat, prénom, matricule, moyenne)**

```
mysql> SELECT s.matricule,s.prenom,s.nom,AVG(p.notePassage) as moyenne
    -> FROM soldat s,passer p
    -> WHERE s.matricule=p.matricule AND p.idParcours=1
    -> GROUP BY s.matricule;
+-----------+----------+--------+---------+
| matricule | prenom   | nom    | moyenne |
+-----------+----------+--------+---------+
| MAT-001   | Moussa   | Diallo | 12.1667 |
| MAT-002   | Ibrahima | Diagne | 10.6667 |
| MAT-003   | Bamba    | Fall   | 10.6667 |
| MAT-005   | Issa     | Pam    |  8.5000 |
+-----------+----------+--------+---------+
4 rows in set (0.05 sec)
```

4. **Afficher un soldat avec ses notes d'un parcours**

```
mysql> SELECT s.matricule,s.nom,s.prenom,p.notePassage,p.noteFinale
    -> FROM soldat s,passer p
    -> WHERE p.matricule='MAT-002' AND p.idParcours=1
    -> AND s.matricule=p.matricule;
+-----------+--------+----------+-------------+------------+
| matricule | nom    | prenom   | notePassage | noteFinale |
+-----------+--------+----------+-------------+------------+
| MAT-002   | Diagne | Ibrahima |          14 |         14 |
| MAT-002   | Diagne | Ibrahima |          12 |          0 |
| MAT-002   | Diagne | Ibrahima |          12 |         13 |
| MAT-002   | Diagne | Ibrahima |           9 |         11 |
| MAT-002   | Diagne | Ibrahima |           5 |          0 |
| MAT-002   | Diagne | Ibrahima |          12 |         14 |
+-----------+--------+----------+-------------+------------+
6 rows in set (0.00 sec)
```

5. **Afficher les soldats qui ont réussi un parcours donné**

```
mysql> SELECT s.matricule,s.prenom,s.nom
    -> FROM soldat s,passer p
    -> WHERE s.matricule=p.matricule AND p.idParcours=1
    -> GROUP BY s.matricule HAVING AVG(p.notePassage)>10;
+-----------+----------+--------+
| matricule | prenom   | nom    |
+-----------+----------+--------+
| MAT-001   | Moussa   | Diallo |
| MAT-002   | Ibrahima | Diagne |
| MAT-003   | Bamba    | Fall   |
+-----------+----------+--------+
3 rows in set (0.00 sec)
```

6. **Afficher les soldats qui ont réussi au moins 10 obstacles**

```
mysql> SELECT s.matricule,s.prenom,s.nom,count(p.noteFinale) as nbr_obs_reussi
    -> FROM soldat s,passer p
    -> WHERE p.matricule=s.matricule AND p.noteFinale>0
    -> Group by s.matricule having nbr_obs_reussi>10;
+-----------+--------+--------+----------------+
| matricule | prenom | nom    | nbr_obs_reussi |
+-----------+--------+--------+----------------+
| MAT-001   | Moussa | Diallo |             11 |
+-----------+--------+--------+----------------+
1 row in set (0.00 sec)
```

7. **Afficher les soldats qui ont réussi tous les obstacles**

```
mysql> SELECT s.matricule AS mat_soldat,s.prenom,s.nom,COUNT(p.noteFinale) as nbr_reussi

    -> FROM soldat s,passer p
    -> WHERE s.matricule=p.matricule and p.noteFinale>0
    -> GROUP BY s.matricule HAVING nbr_reussi=
    -> (SELECT COUNT(noteFinale) FROM passer WHERE mat_soldat=matricule);
+------------+--------+--------+------------+
| mat_soldat | prenom | nom    | nbr_reussi |
+------------+--------+--------+------------+
| MAT-001    | Moussa | Diallo |         11 |
+------------+--------+--------+------------+
1 row in set (0.00 sec)
```

8. **Afficher les soldats qui n'ont réussi aucun obstacle**

```
mysql> SELECT s.matricule AS mat_soldat,s.prenom,s.nom,COUNT(p.noteFinale) as nbr_echec
    -> FROM soldat s,passer p
    -> WHERE s.matricule=p.matricule and p.noteFinale=0
    -> GROUP BY s.matricule HAVING nbr_echec=
    -> (SELECT COUNT(noteFinale) FROM passer WHERE mat_soldat=matricule);
+------------+--------+-----+-----------+
| mat_soldat | prenom | nom | nbr_echec |
+------------+--------+-----+-----------+
| MAT-005    | Issa   | Pam |         6 |
| MAT-007    | Amadou | Ba  |         5 |
+------------+--------+-----+-----------+
2 rows in set (0.01 sec)
```

9. **Afficher les instructeurs ayant examiné des soldats pour un parcours donné**

```
mysql> SELECT DISTINCT s.matricule,s.nom,s.prenom
    -> FROM soldat s,passer p
    -> WHERE s.idRole=2 AND p.idParcours=2 AND
    -> s.matricule=p.matricule_instructeur;
+-----------+--------+---------+
| matricule | nom    | prenom  |
+-----------+--------+---------+
| MAT-008   | Dia    | Hamidou |
| MAT-010   | Ndiaye | Moctar  |
+-----------+--------+---------+
2 rows in set (0.00 sec)
```

10. **Pour chaque instructeur afficher le nombre de soldats examinés dans un parcours**

```
mysql> SELECT s.matricule,s.prenom,s.nom,COUNT(DISTINCT p.matricule)as nb_soldats
    -> FROM soldat s,passer p
    -> WHERE s.idRole=2 AND p.idParcours=1 AND
    -> s.matricule=p.matricule_instructeur
    -> GROUP BY p.matricule_instructeur;
+-----------+----------+--------+------------+
| matricule | prenom   | nom    | nb_soldats |
+-----------+----------+--------+------------+
| MAT-006   | Ibrahima | Dia    |          4 |
| MAT-008   | Hamidou  | Dia    |          3 |
| MAT-010   | Moctar   | Ndiaye |          2 |
+-----------+----------+--------+------------+
3 rows in set (0.00 sec)
```

11. **Pour chaque parcours afficher le nombre de soldats ayant participés**

```
mysql> SELECT p.idParcours,p.nomParcours,COUNT(DISTINCT pp.matricule) AS nb_soldat
    -> FROM parcours p,passer pp
    -> WHERE p.idParcours=pp.idParcours
    -> GROUP BY p.idParcours;
+------------+-------------+-----------+
| idParcours | nomParcours | nb_soldat |
+------------+-------------+-----------+
|          1 | Parcours1   |         4 |
|          2 | Parcours2   |         3 |
+------------+-------------+-----------+
2 rows in set (0.00 sec)
```

Projet : Reprendre le projet de fichier en utilisant une base de données mysql **lien du projet**