

Projet SLAM

Table des matières

| | |
|--|---|
| 1) Mon projet..... | 2 |
| 2) Etapes du projet..... | 2 |
| 1) Créer un dictionnaire de données | 2 |
| 2) Concevoir le modèle conceptuel de données et le modèle logique de données | 2 |
| 3) Créer la base de données..... | 3 |
| 3) Explication des entités et des relations | 4 |
| 4) Diagramme de cas d'utilisation | 5 |

1) Mon projet

Mon premier projet consiste à créer une base de données pour une compétition de gymnastique.

2) Etapes du projet

1) Créer un dictionnaire de données

Je commence par créer un dictionnaire de données qui consiste à réfléchir à toutes les données dont on aura besoin (Voir Figure 1).

| TABLE | Nom | Nom logique | Type | Longueur | Observation |
|--------------|------------------------|----------------------|-----------|----------|--|
| GYMNASTES | licence_gymnaste | gym_licence_gymnaste | numérique | 11 | Clé primaire, identifiant unique pour chaque gymnaste |
| | nom_gymnaste | gym_nom | caractère | 50 | Nom de famille de la gymnaste |
| | prénom_gymnaste | gym_prenom | caractère | 50 | Prénom de la gymnaste |
| | date_naissance | gym_date_naissance | date | | Date de naissance de la gymnaste |
| | catégorie | gym_catégorie | caractère | 50 | Catégorie de la gymnaste (poussines, jeunes, aînées) |
| | club | gym_club | caractère | 100 | Nom du club auquel la gymnaste est affiliée |
| | identifiantéquipe | gym_idéquipe | numérique | 11 | Référence à l'équipe (peut être NULL si la gymnaste concourt en individuel) |
| | | | | | |
| EQUIPES | identifiantéquipe | equi_idéquipe | numérique | 11 | Clé primaire, identifiant unique pour chaque équipe |
| | noméquipe | equi_nom | caractère | 100 | Nom de l'équipe |
| JUGES | licence_juge | juge_licence | numérique | 11 | Clé primaire, identifiant unique pour chaque juge |
| | nom_juge | juge_nom | caractère | 50 | Nom de famille du juge |
| | prénom_juge | juge_prenom | caractère | 50 | Prénom du juge |
| | certification | juge_certification | caractère | 50 | Niveau de certification du juge (niveau 1, 2 ou 3) |
| EPREUVES | identifiantépreuve | epr_identifiant | numérique | 11 | Clé primaire, identifiant unique pour chaque épreuve |
| | nomépreuve | epr_nom | caractère | 50 | Nom de l'épreuve (sol, barres asymétriques, poutre, saut) |
| SCORES | identifiant_score | sco_identifiant | numérique | 11 | Clé primaire, identifiant unique pour chaque score |
| | note | sco_note | décimal | 5 | Note donnée par le juge (entre 0 et 10) |
| COMPETITIONS | identifiantcompétition | compet_identifiant | numérique | 11 | Clé primaire, identifiant unique pour chaque compétition |
| | nomcompétition | compet_nom | caractère | 50 | Nom de la compétition (championnat départemental, régional, national, international) |
| | typecompétition | compet_type | caractère | 50 | Type de compétition (individuel, par équipes) |
| | lieu | compet_lieu | caractère | 100 | Lieu où se déroule la compétition |
| | date | compet_date | date | | Date de la compétition |

Figure 1 Dictionnaire de données avec excel

2) Concevoir le modèle conceptuel de données et le modèle logique de données

Ensuite, j'utilise le logiciel Looping pour concevoir le modèle conceptuel de données (MCD) qui consiste à représenter les données sous forme d'entités et de relations entre eux (Voir Figure 2). Le logiciel génère aussi le modèle logique de données (MLD) qui détaille comment les entités et les associations seront représentées en tables et en relations (Voir Figure 3).

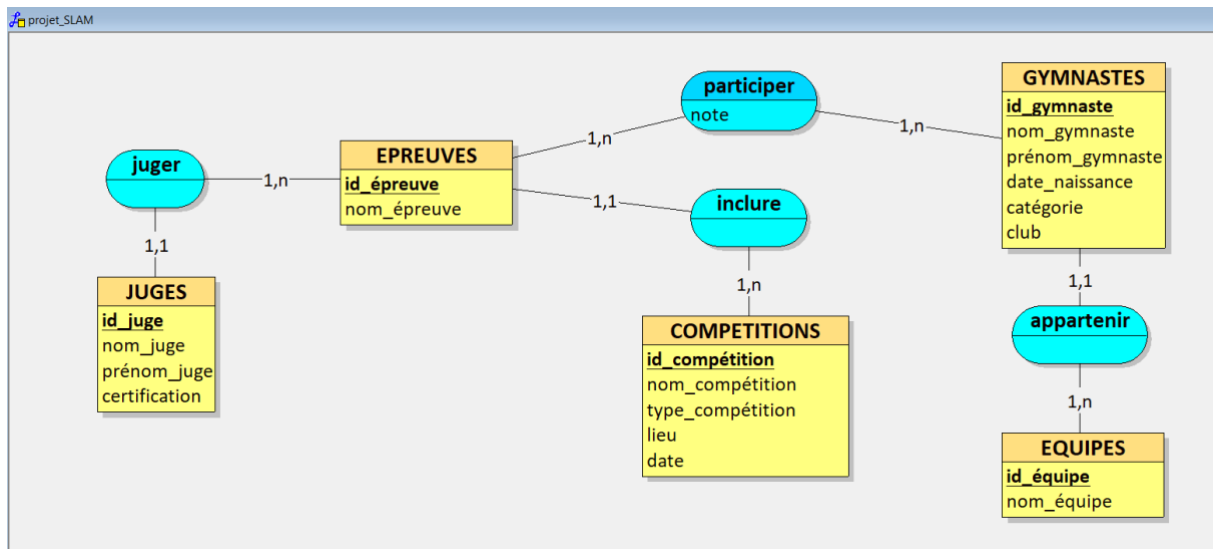


Figure 2 MCD sur Looping

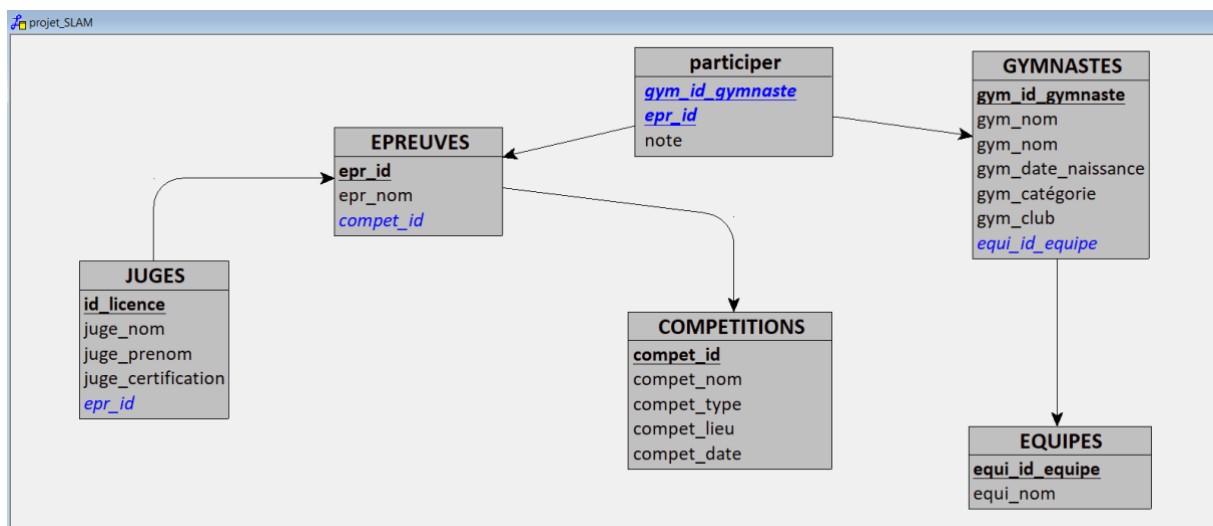


Figure 3 MLD sur Looping

3) Créer la base de données

Enfin, j'importe le script SQL généré par le logiciel Looping (Voir Figure 4 et 5) pour l'inclure dans PHPMyAdmin pour créer la base de données (Voir Figure 6). Avec PHPMyAdmin, on peut aussi générer le concepteur (Voir Figure 7).

```

SQL
CREATE TABLE EQUIPES(
    equi_id_equipe INT AUTO_INCREMENT,
    equi_nom VARCHAR(100),
    PRIMARY KEY(equi_id_equipe)
);

CREATE TABLE COMPETITIONS(
    compet_id INT AUTO_INCREMENT,
    compet_nom VARCHAR(50),
    compet_type VARCHAR(50),
    compet_lieu VARCHAR(100),
    compet_date DATE,
    PRIMARY KEY(compet_id)
);

CREATE TABLE GYMNASTES(
    gym_id_gymnaste INT AUTO_INCREMENT,
    gym_nom VARCHAR(50),
    gym_nom_1 VARCHAR(50),
    gym_date_naissance VARCHAR(50),
    gym_categorie VARCHAR(50),
    gym_club VARCHAR(50),
    equi_id_equipe INT NOT NULL,
    PRIMARY KEY(gym_id_gymnaste),
    FOREIGN KEY(equi_id_equipe) REFERENCES EQUIPES(equi_id_equipe)
);

```

Figure 5 Script SQL avec Looping

```

SQL
CREATE TABLE EPREUVES(
    epr_id INT AUTO_INCREMENT,
    epr_nom VARCHAR(50),
    compet_id INT NOT NULL,
    PRIMARY KEY(epr_id),
    FOREIGN KEY(compet_id) REFERENCES COMPETITIONS(compet_id)
);

CREATE TABLE JUGES(
    id_licence INT AUTO_INCREMENT,
    juge_nom VARCHAR(50),
    juge_prenom VARCHAR(50),
    juge_certification VARCHAR(50),
    epr_id INT NOT NULL,
    PRIMARY KEY(id_licence),
    FOREIGN KEY(epr_id) REFERENCES EPREUVES(epr_id)
);

CREATE TABLE participer(
    gym_id_gymnaste INT,
    epr_id INT,
    note DECIMAL(15,3),
    PRIMARY KEY(gym_id_gymnaste, epr_id),
    FOREIGN KEY(gym_id_gymnaste) REFERENCES GYMNASTES(gym_id_gymnaste),
    FOREIGN KEY(epr_id) REFERENCES EPREUVES(epr_id)
);

```

Figure 4 Script SQL avec Looping

| Table | Action | Lignes | Type | Interclassement | Taille | Perte |
|--------------|--|--------|--------|--------------------|-----------|-------|
| competitions | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_general_ci | 16,0 kio | - |
| epreuves | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_general_ci | 32,0 kio | - |
| equipes | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_general_ci | 16,0 kio | - |
| gymnastes | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_general_ci | 32,0 kio | - |
| juges | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_general_ci | 32,0 kio | - |
| participer | Parcourir Structure Rechercher Insérer Vider Supprimer | 0 | InnoDB | utf8mb4_general_ci | 32,0 kio | - |
| 6 tables | Somme | 0 | InnoDB | utf8mb4_general_ci | 160,0 kio | 0 o |

Figure 6 Base de données PHPMYAdmin

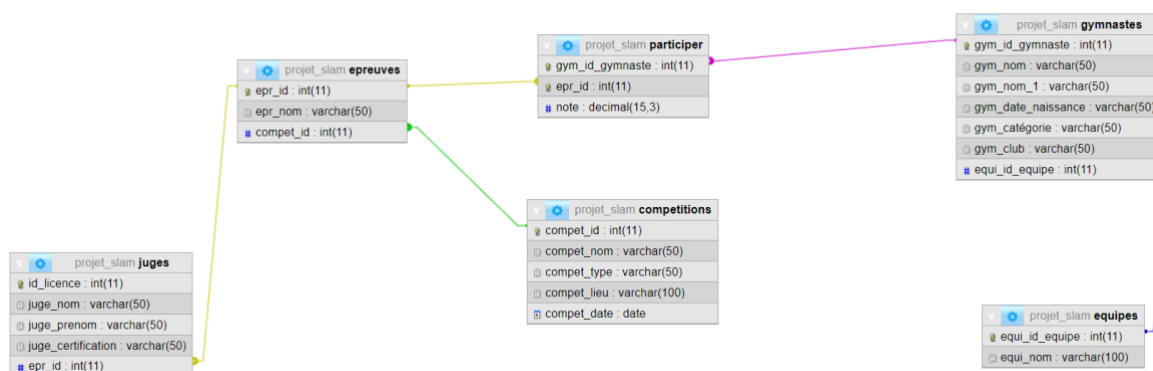


Figure 7 Concepteur sur PHPMYAdmin

3) Explication des entités et des relations

- JUGES – EPREUVES
 - o Un juge peut juger une épreuve
 - o Une épreuve peut être jugée par plusieurs juges
- COMPETITIONS – EPREUVES
 - o Une compétition peut inclure plusieurs épreuves
 - o Plusieurs épreuves peuvent être incluses dans une compétition
- GYMNASTES – EPREUVES
 - o Une gymnaste participe à plusieurs épreuves

- Une épreuve est participée par plusieurs gymnastes
- GYMNASTES – EQUIPES
 - Une gymnaste appartient à une équipe
 - Une équipe appartient à plusieurs gymnastes

4) Diagramme de cas d'utilisation

Ensuite, je crée un diagramme de cas d'utilisation. C'est un type de diagramme UML (Unified Modeling Language) qui représente les interactions entre les acteurs (utilisateurs ou systèmes) et les fonctionnalités principales d'un système (appelées cas d'utilisation). Il est utilisé pour décrire ce que le système doit accomplir du point de vue des utilisateurs et des objectifs métiers.

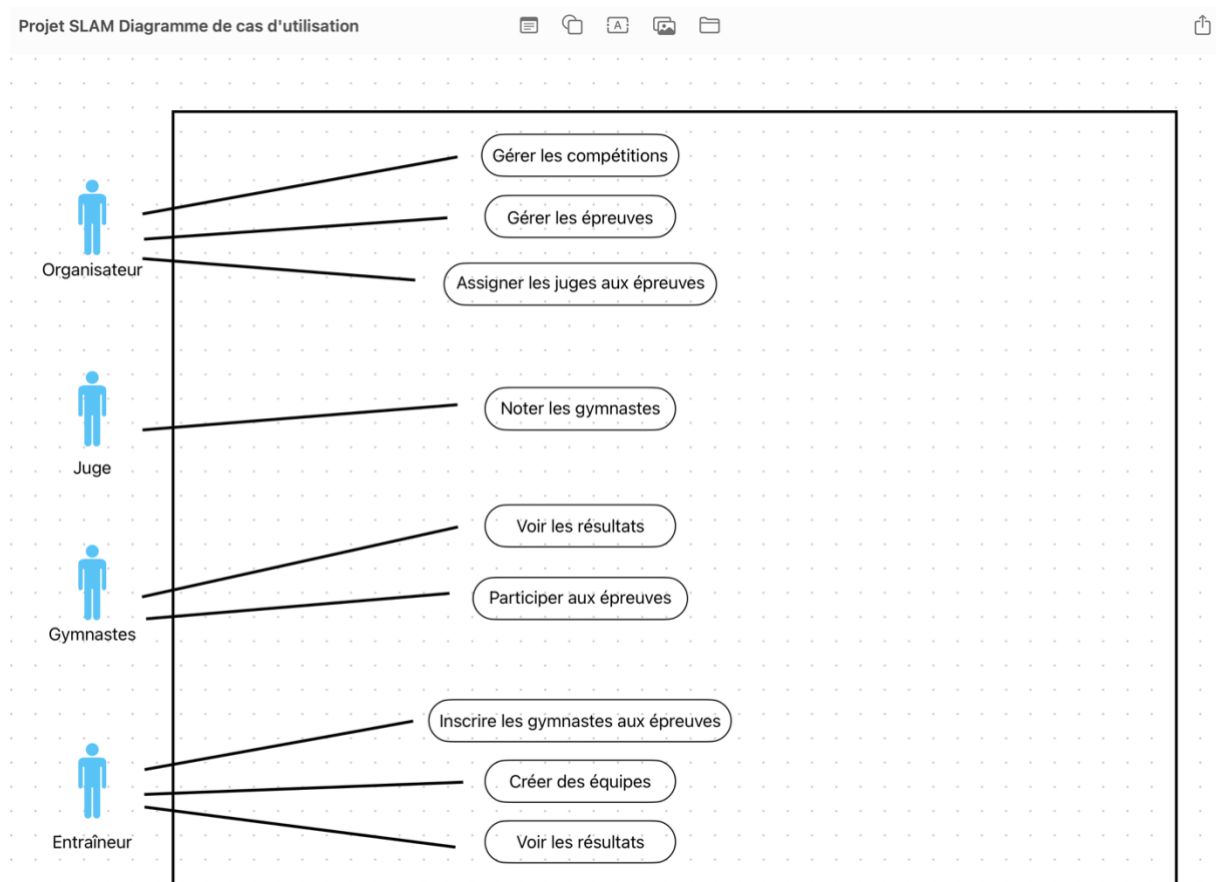


Figure 8 Diagramme de cas d'utilisation

