

GESTION D'UNE ASSOCIATION SPORTIVE

SPECIFICATION TECHNIQUE



LYCEE ST VINCENT

LAURA GONÇALVES - LUCAS BOGANIN - HUGO MONTEIRO

Table des matières

I. Maquettes	2-4
II. Modélisation	5
III. Schéma relationnel	6-8
➤ Table Utilisateur	6
➤ Table Comptable	6
➤ Table Administrateur	6
➤ Table Élève	6
➤ Table Evénement	6
➤ Table Classe	6
➤ Table Type de budget	7
➤ Table Budget	7
➤ Table Flux	7
➤ Table Accéder	7
➤ Table Responsabilité	7
➤ Table Gérer	8
IV. Script SQL	9-14

I-Maquettes



CONNEXION

Identifiant

Mot de passe

☐ Se souvenir de moi [Mot de passe oublié ?](#)



Détails des élèves

Ajouter

Nom	Prenom	Classe	Date de naissance	Sexe	Numéro élève	Numéro parent	E-mail	Mise à jour	
									<button>Editer</button> <button>Supprimer</button>
									<button>Editer</button> <button>Supprimer</button>
									<button>Editer</button> <button>Supprimer</button>
									<button>Editer</button> <button>Supprimer</button>
									<button>Editer</button> <button>Supprimer</button>



Ajouter un élève

Annuler

Identité

Nom

Prenom

Sexe

Naissance

Autres

Sweat ☐ Oui ☐ Non

Prelevement ☐ Oui ☐ Non

Informations

Téléphone

Tel Parent

Mail

Naissance





Gestion Comptabilité


Rechercher un élève :

Résultats

Nom	Prenom	Classe	Prelevement	Sweat	Montant Pre
-----	--------	--------	-------------	-------	-------------

Flux élève

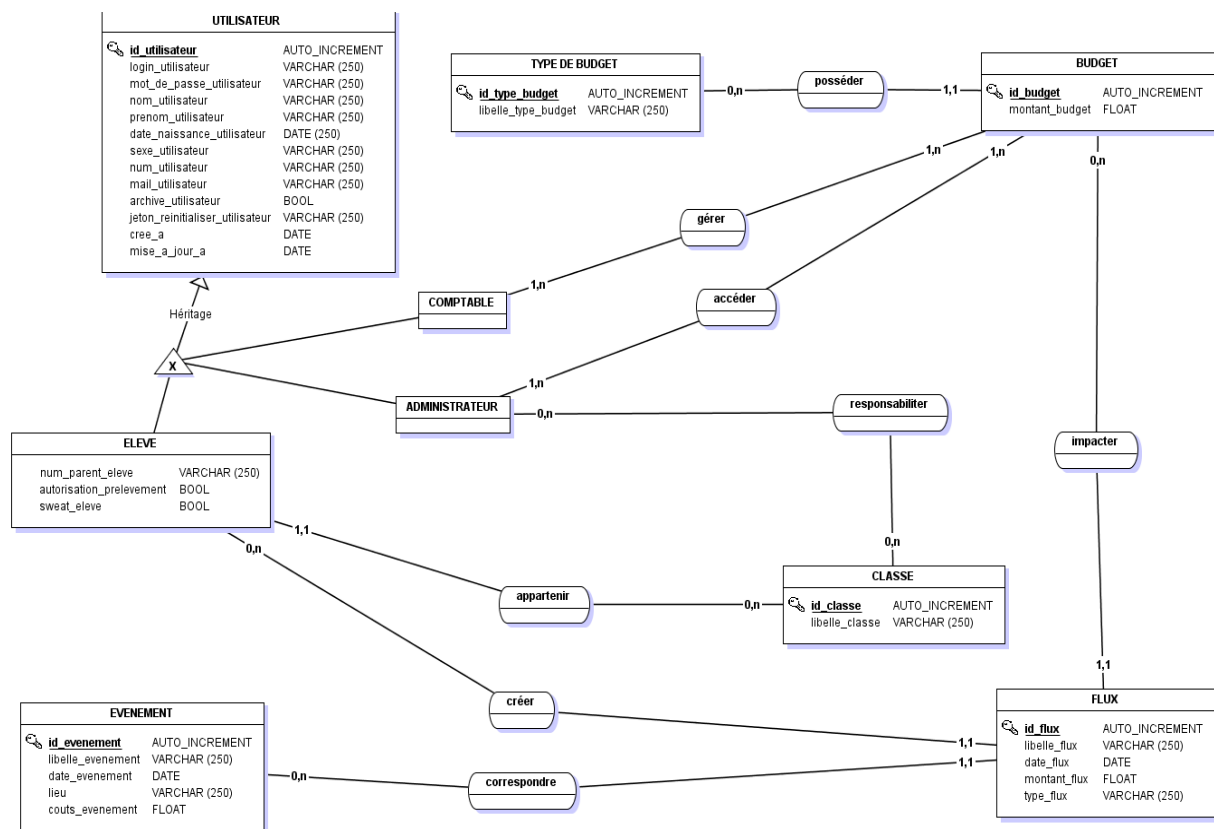
Libelle	Date	Montant	Type	Budget Impacte
---------	------	---------	------	----------------



Statistiques

Licenciés	Licence 10€ : 0			Licence 25€ : 0	
Participation Evenement	Nom	Date	Lieu	Montant Total	Nombre Total participants
Effectif Licencié par Evenement	Evenement : <input style="width: 100px;" type="text"/>			Nombre de licencié pour l'evenement x : 0	
Effectif Licencié Fille/Garçon	Licencié Fille : 0			Licence Garçon : 0	
Effectif Licencié par naissance	Année : <input style="width: 80px;" type="text"/>			Nombre de licencié pour l'année x : 0	
Statistique élève	Rechercher un élève : <input style="width: 250px;" type="text"/>				
	L'élève à participer à : 0 événement(s)				

I-Modélisation



I-Schéma relationnel

Table comptable

COMPTABLE (id_utilisateur, login_utilisateur, mot_de_passe_utilisateur, nom_utilisateur, prenom_utilisateur, date_naissance_utilisateur, sexe_utilisateur, num_utilisateur, mail_utilisateur, archive_utilisateur, jeton_reinitialisation_utilisateur, cree_a, mise_a_jour_a)

Clé primaire :id_utilisateur

Table administrateur

ADMINISTRATEUR (id_utilisateur, login_utilisateur, mot_de_passe_utilisateur, nom_utilisateur, prenom_utilisateur, date_naissance_utilisateur, sexe_utilisateur, num_utilisateur, mail_utilisateur, archive_utilisateur, jeton_reinitialisation_utilisateur, cree_a, mise_a_jour_a)

Clé primaire :id_utilisateur

Table élève

ELEVE (id_utilisateur, login_utilisateur, mot_de_passe_utilisateur, nom_utilisateur, prenom_utilisateur, date_naissance_utilisateur, sexe_utilisateur, num_utilisateur, mail_utilisateur, archive_utilisateur, jeton_reinitialisation_utilisateur, cree_a, mise_a_jour_a)

Clé primaire :id_utilisateur

Table évènement

EVENEMENT (id_evenement, libelle_evenement, date_evenement, lieu, cout_evenement)

Clé primaire :id_evenement

Table classe

CLASSE (id_classe, libelle_classe)

Clé primaire :id_classe

Table type de budget

TYPE DE BUDGET (id_type_de_budget, libelle_type_de_budget)

Clé primaire :id_type_de_budget

Table budget

BUDGET (id_budget, montant_budget, id_type_budget)

Clé primaire :id_budget

Clé étrangère :

id_type_budget en référence à id_type_budget de la table TYPE DE BUDGET

Table flux

FLUX (id_flux, libelle_flux, date_flux, montant_flux, type_flux, id_budget, id_utilisateur, id_evenement)

Clé primaire : id_flux

Clé étrangère :

id_budget en référence à id_budget de la table BUDGET

id_utilisateur en référence à id_utilisateur de la table ELEVE

id_evenement en référence à id_evenement de la table EVENEMENT

Table accéder

ACCEDER (id_budget, id_utilisateur)

Clé primaire : (id_utilisateur, id_budget)

Clé étrangère :

id_utilisateur en référence à id_utilisateur de la table ADMINISTRATEUR

id_budget en référence à id_budget de la table BUDGET

Table responsabiliter

RESPONSABILITER (id_classe, id_utilisateur)

Clé primaire : (id_utilisateur, id_classe)

Clé étrangère :

id_utilisateur en référence à id_utilisateur de la ADMINISTRATEUR

id_classe en référence à id_classe de la table CLASSE

Table gérer

GERER (id_budget, id_utilisateur)

Clé primaire : (id_utilisateur, id_budget)

Clé étrangère :

id_utilisateur en référence à id_utilisateur de la table ADMINISTRATEUR

id_budget en référence à id_budget de la table BUDGET

Pour les sous types de l'entité générique UTILISATEUR il y a partition car nous avons d'après le cahier des charges une couverture, les occurrences appartiennent toutes à entité spécifique qui l'entité générique UTILISATEUR. Il y a disjonction car les intersections des ensembles des entités spécifiques, les occurrences appartiennent à une et une seule entité spécifique ou peuvent elles peuvent répondre à plusieurs spécificités.

Nous avons choisi la stratégie d'implémentation qui consiste à créer une table spécifique pour chaque sous types de l'entité générique.

VI-SCRIPT SQL

Table: CLASSE

```
CREATE TABLE CLASSE(  
    id_classe    Int Auto_increment NOT NULL ,  
    libelle_classe Varchar (250) NOT NULL  
  
    ,CONSTRAINT CLASSE_PK PRIMARY KEY (id_classe)  
);
```

Table: EVENEMENT

```
CREATE TABLE EVENEMENT(  
    id_evenement    Int Auto_increment NOT NULL ,  
    libelle_evenement Varchar (250) NOT NULL ,  
    date_evenement  Date NOT NULL ,  
    lieu            Varchar (250) NOT NULL ,  
    couts_evenement Float NOT NULL  
  
    ,CONSTRAINT EVENEMENT_PK PRIMARY KEY (id_evenement) );
```

Table: ELEVE

```
CREATE TABLE ELEVE(  
    id_utilisateur      Int NOT NULL ,  
    num_parent_eleve    Varchar (250) NOT NULL ,  
    autorisation_prelevement Bool NOT NULL ,  
    sweat_eleve         Bool NOT NULL ,  
    login_utilisateur   Varchar (250) NOT NULL ,  
    mot_de_passe_utilisateur Varchar (250) NOT NULL ,  
    nom_utilisateur     Varchar (250) NOT NULL ,  
    prenom_utilisateur  Varchar (250) NOT NULL ,  
    date_naissance_utilisateur Date NOT NULL ,  
    sexe_utilisateur    Varchar (250) NOT NULL ,
```

GESTION D'UNE ASSOCIATION SPORTIVE
Spécification Techniques

```
num_utilisateur      Varchar (250) NOT NULL ,
mail_utilisateur     Varchar (250) NOT NULL ,
archive_utilisateur  Bool NOT NULL ,
jeton_reinitialiser_utilisateur Varchar (250) NOT NULL ,
cree_a               Date NOT NULL ,
mise_a_jour_a        Date NOT NULL ,
id_classe             Int NOT NULL
```

```
,CONSTRAINT ELEVE_PK PRIMARY KEY (id_utilisateur)
,CONSTRAINT ELEVE_UTILISATEUR_FK FOREIGN KEY (id_utilisateur)
REFERENCES UTILISATEUR(id_utilisateur)
,CONSTRAINT ELEVE_CLASSE0_FK FOREIGN KEY (id_classe) REFERENCES
CLASSE(id_classe)
```

```
);
```

Table: ADMINISTRATEUR

```
CREATE TABLE ADMINISTRATEUR(
```

```
id_utilisateur      Int NOT NULL ,
login_utilisateur   Varchar (250) NOT NULL ,
mot_de_passe_utilisateur Varchar (250) NOT NULL ,
nom_utilisateur     Varchar (250) NOT NULL ,
prenom_utilisateur  Varchar (250) NOT NULL ,
date_naissance_utilisateur Date NOT NULL ,
sexe_utilisateur    Varchar (250) NOT NULL ,
num_utilisateur     Varchar (250) NOT NULL ,
mail_utilisateur    Varchar (250) NOT NULL ,
archive_utilisateur Bool NOT NULL ,
jeton_reinitialiser_utilisateur Varchar (250) NOT NULL ,
cree_a              Date NOT NULL ,
mise_a_jour_a       Date NOT NULL
```

```
,CONSTRAINT ADMINISTRATEUR_PK PRIMARY KEY (id_utilisateur)
,CONSTRAINT ADMINISTRATEUR_UTILISATEUR_FK FOREIGN KEY
(id_utilisateur) REFERENCES UTILISATEUR(id_utilisateur));
```

Table: COMPTABLE

```
CREATE TABLE COMPTABLE(  
    id_utilisateur      Int NOT NULL ,  
    login_utilisateur   Varchar (250) NOT NULL ,  
    mot_de_passe_utilisateur Varchar (250) NOT NULL ,  
    nom_utilisateur     Varchar (250) NOT NULL ,  
    prenom_utilisateur  Varchar (250) NOT NULL ,  
    date_naissance_utilisateur Date NOT NULL ,  
    sexe_utilisateur    Varchar (250) NOT NULL ,  
    num_utilisateur     Varchar (250) NOT NULL ,  
    mail_utilisateur    Varchar (250) NOT NULL ,  
    archive_utilisateur Bool NOT NULL ,  
    jeton_reinitialiser_utilisateur Varchar (250) NOT NULL ,  
    cree_a              Date NOT NULL ,  
    mise_a_jour_a       Date NOT NULL  
  
    ,CONSTRAINT COMPTABLE_PK PRIMARY KEY (id_utilisateur)  
    ,CONSTRAINT COMPTABLE_UTILISATEUR_FK FOREIGN KEY (id_utilisateur)  
    REFERENCES UTILISATEUR(id_utilisateur)  
);
```

Table: TYPE DE BUDGET

```
CREATE TABLE TYPE_DE_BUDGET(  
    id_type_budget  Int Auto_increment NOT NULL ,  
    libelle_type_budget Varchar (250) NOT NULL  
  
    ,CONSTRAINT TYPE_DE_BUDGET_PK PRIMARY KEY (id_type_budget)  
);
```

Table: BUDGET

```
CREATE TABLE BUDGET(  
    id_budget  Int Auto_increment NOT NULL ,  
    montant_budget Float NOT NULL ,
```

id_type_budget Int NOT NULL

```
,CONSTRAINT BUDGET_PK PRIMARY KEY (id_budget)
,CONSTRAINT BUDGET_TYPE_DE_BUDGET_FK FOREIGN KEY
(id_type_budget) REFERENCES TYPE_DE_BUDGET(id_type_budget)
);
```

Table: FLUX

```
CREATE TABLE FLUX(
    id_flux      Int Auto_increment NOT NULL ,
    libelle_flux Varchar (250) NOT NULL ,
    date_flux    Date NOT NULL,
    montant_flux Float NOT NULL,
    type_flux Varchar (250) NOT NULL,
    id_budget Int (11) NOT NULL,
    id_utilisateur Int (11) NOT NULL,
    id_evenement Int (11) NOT NULL

    ,CONSTRAINT FLUX_PK PRIMARY KEY (id_flux)

    ,CONSTRAINT FLUX_BUDGET_FK FOREIGN KEY (id_budget) REFERENCES
    BUDGET(id_budget)

    ,CONSTRAINT FLUX_UTILISATEUR_FK FOREIGN KEY (id_utilisateur)
    REFERENCES ELEVE(id_utilisateur)

    ,CONSTRAINT FLUX_EVENEMENT_FK FOREIGN KEY (id_evenement)
    REFERENCES EVENEMENT(id_evenement)
);
```

Table: ACCEDER

```
CREATE TABLE ACCEDER(
    id_budget      Int Auto_increment NOT NULL ,
    id_utilisateur Int (11) NOT NULL ,

    ,CONSTRAINT FLUX_PK PRIMARY KEY (id_budget)
```

```
,CONSTRAINT FLUX_PK PRIMARY KEY (id_flux)
,CONSTRAINT ACCEDER_BUDGET_FK FOREIGN KEY (id_budget) REFERENCES
BUDGET(id_budget)
,CONSTRAINT ACCEDER_UTILISATEUR_FK FOREIGN KEY (id_utilisateur)
REFERENCES ADMINISTRATEUR(id_utilisateur)
);
```

Table: RESPONSABILITER

```
CREATE TABLE RESPONSABILITER(
    id_classe    Int Auto_increment NOT NULL ,
    id_utilisateur Int (11) NOT NULL ,

    ,CONSTRAINT FLUX_PK PRIMARY KEY (id_classe)
    ,CONSTRAINT FLUX_PK PRIMARY KEY (id_flux)
    ,CONSTRAINT ACCEDER_BUDGET_FK FOREIGN KEY (id_classe) REFERENCES
    CLASSE(id_classe)
    ,CONSTRAINT ACCEDER_UTILISATEUR_FK FOREIGN KEY (id_utilisateur)
    REFERENCES ADMINISTRATEUR(id_utilisateur)
);
```

Table: GERER

```
CREATE TABLE GERER(
    id_budget    Int Auto_increment NOT NULL ,
    id_utilisateur Int (11) NOT NULL ,

    ,CONSTRAINT FLUX_PK PRIMARY KEY (id_budget)
    ,CONSTRAINT FLUX_PK PRIMARY KEY (id_flux)
    ,CONSTRAINT ACCEDER_BUDGET_FK FOREIGN KEY (id_budget) REFERENCES
    BUDGET(id_budget)
    ,CONSTRAINT ACCEDER_UTILISATEUR_FK FOREIGN KEY (id_utilisateur)
    REFERENCES COMPTABLE(id_utilisateur)
);
```