

**­­Gestion de Données (C4)**

S3.A.01 – Développement d’une application

Tuteur : Pr. Richard Chbeir

Projet : Hego Lagunak, une application de parrainage pour le BDE

Équipe 3 : BRIERRE Titouan (TP1), DARGAZANLI Nicolas (TP1), ERREZARET Leho (TP2) et MAURICE Alexandre (TP1), en BUT Informatique, 2022, Semestre 3, Parcours A.

Table des matières

[Identifier les différents besoins de notre projet 3](#_Toc123896699)

[Rappel de projet 3](#_Toc123896700)

[Identifications des besoins 4](#_Toc123896701)

[Schéma UML et dictionnaire de données 5](#_Toc123896702)

[Schéma relationnel 8](#_Toc123896703)

[Scripts SQL (Création) 9](#_Toc123896704)

[Scripts SQL (interrogation) 10](#_Toc123896705)

[Scripts SQL (Insertion) 10](#_Toc123896706)

[Technologies utilisées 11](#_Toc123896707)

# Identifier les différents besoins de notre projet

## Rappel de projet

Nous travaillons sur un site web permettant de gérer automatiquement les parrainages réalisés par le BDE.

Le parrainage est un évènement annuel réalisé par l'association étudiante Hego Berria (un BDE), dans lequel les étudiants de première année répondent d'abord à un questionnaire rédigé par les membres du BDE. Ensuite, les étudiants de deuxième année sont associés automatiquement (ou en choisissant, en fonction du paramétrage du questionnaire) à un étudiant de 1ère année, créant alors l'association entre le parrain et le filleul.

Le système permet aux utilisateurs de s'inscrire avec leur adresse mèl de l'UPPA et de conserver le même compte tout au long de leur cursus au sein de la structure. Le site internet est destiné aux étudiants et au BDE, qui ne dispose que d'un compte "administrateur".

Toute personne se connectant au site peut voir si le questionnaire est disponible ou non et modifier ses informations personnelles. On dispose alors de leur nom, leur prénom, leur adresse mèl étudiante et des réponses qu’ils donneront au questionnaire (uniquement pendant la durée de l’évènement).

Les questionnaires sont rédigés et ouverts aux réponses par le BDE. Ils sont caractérisés par des questions et des réponses de différents types (Questionnaire à Choix Multiples ou réponses libre), un type d'association (automatique, en faisant intervenir l'algorithme de l'application, ou manuelle, nécessitant l'intervention des étudiants de 2ème année). Enfin, une date d'ouverture et de fermeture sont spécifiés.

Le bureau des étudiants Hego Berria est une association avec une identité forte, qui commence à développer en parallèle (sans lien avec la SAE) un site web vitrine et une application de covoiturage.

Faciliter ce système de parrainage peut alors le rendre plus fiable et avenant, en proposant des fonctionnalités ludiques dans un environnement maîtrisé (sans devoir recommencer à chaque fois tous les processus de création du formulaire ou devoir utiliser des applications tierces comme Excel ou Forms de chez Google, dénaturant l'association).

## Identifications des besoins

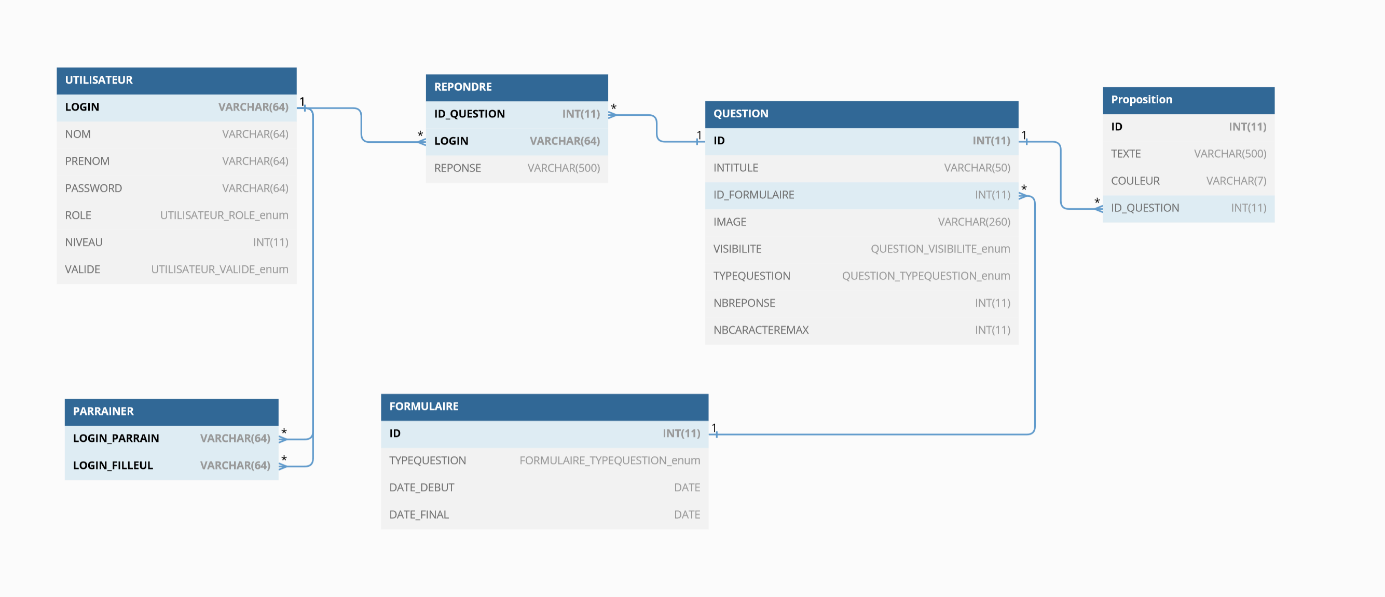
Notre application va accueillir des utilisateurs qui vont devoir s’identifier, il va donc falloir stocker les informations relatives à leurs compte (identifiant, mot de passe, état de validité) ainsi que leurs informations personnelles (nom, prénom, année de promotion)

De plus nous devrons mémoriser le questionnaire proposé par le BDE ainsi que les différentes questions de celui-ci, sans oublier toutes les informations relatives aux questions comme : les possibilités de réponses dans le cas d’une question QCM, ou le nombre totale de caractères disponibles d’une question libre.

Rajoutons à cela une relation entre les utilisateurs et ces questions qui formeront les réponses qui vont ensuite être traiter dans notre algorithme d’assortiment.

Finalement, une fois l’algorithme d’assortiment des étudiants effectué, nous nous retrouverons avec des couples d’étudiants qui seront des parrains/filleuls que nous devrons aussi stocker dans la base de données.

# Schéma UML et dictionnaire de données



*Note : « TYPEQUESTION » de la table « FORMULAIRE » est en réalité « TypeASSOS ».*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Utilisateur | | | | |
| Nom | Type | Contrainte | Description | Exemple |
| LOGIN | Chaine de caractères | Clé primaire | L’identifiant de l’utilisateur | ndargazan001 |
| NOM | Chaine de caractères | / | Nom de l’utilisateur | Nicolas |
| PRENOM | Chaine de caractères | / | Prénom de l’utilisateur | Dargazanli |
| PASSWORD | Chaine de caractères | / | Mot de passe hashé de l’utilisateur | $2y$10$TEbMLbFW  6C2NbQPupCFK  hO2fcNlm1zmKbMj  65be/7w45zWIO/vNZ2 |
| ROLE | enum | admin ou user  Défaut « user » | Le rôle de l’utilisateur | user |
| NIVEAU | entier | / | Niveau de la formation | 2 |
| VALIDE | enum | oui/non  Défault « non » | L’état de validité du compte | oui |

Nous avons décidé d’utiliser des énumérations comme type de données pour faciliter la compréhension dans le code plus tard.

Le niveau représente l’année de formation 2 = « 2eme année de formation »

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Formulaire | | | | |
| Nom | Type | Contrainte | Description | Exemple |
| ID | entier | Clé primaire | Identifiant du formulaire | 1 |
| TYPEASSOS | enum | auto/1/2 | Le mode d’association des étudiants | auto |
| DATE\_DEBUT | date | / | Date de d’ouverture du formulaire | 11/01/2023 |
| DATE\_FIN | date | Postérieur à la date de début | Date de fermeture du formulaire | 27/01/2023 |

Le type d’association permet de paramétrer l’algorithme d’association :

* Auto : L’algorithme se charge de trouver les meilleures correspondances
* 1 : Les premières années choisissent leurs parrains
* 2 : Les deuxièmes années choisissent leurs filleuls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Question | | | | |
| Nom | Type | Contrainte | Description | Exemple |
| ID | entier | Clé primaire | Identifiant de la question | 3 |
| INTITULE | Chaine de caractères | Non nulle | Intitule de la question | Aimez-vous les pâtes ? |
| ID\_FORMULAIRE | entier | Clé étrangère | Clé étrangère de Formulaire | 1 |
| IMAGE | Chaine de caractères | / | Chemin vers l’illustration de la question | ..\sources\images\imgplaceholder.jpg |
| VISIBILITE | | enum | All/1/2 | Qui peux voir la question | all |
| TYPEQUESTION | enum | QCM/LIBRE | Type de la question | QCM |
| NBREPONSE | entier | / | Nombre de réponse a la question (QCM) | 2 |
| NBCARACTEREMAX | entier | / | Nombre de caractère max à la réponse libre | 260 |

Nous avons décidé de faire qu’une seule table Question même s’il y a deux types, simplement car nous n’arrivons pas à maitriser le concept d’héritage dans la base de données suffisamment pour l’implémenter, cela va engendrer des attributs vide (par exemple le nombre de caractères max pour une question de type QCM).

La visibilité permettra de limiter la question uniquement à un niveau d’étudiant :

* All : Tout le monde
* 1 : seulement les premières années
* 2 : seulement les deuxièmes années

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposition | | | | |
| Nom | Type | Contrainte | Description | Exemple |
| ID | entier | Clé primaire | Identifiant de la proposition | 1 |
| TEXTE | Chaine de caractère | / | Le texte de la proposition | Oui |
| COULEUR | Chaine de caractères | / | Couleur de la proposition sur la carte | #808080 |
| ID\_QUESTION | entier | Clé étrangère | Clé étrangère de la question | 3 |

Seulement des propositions de questions de type QCM seront inséré dans cette table, car les questions libres n’auront pas de propositions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Repondre | | | | |
| Nom | Type | Contrainte | Description | Exemple |
| ID\_QUESTION | entier | Clé primaire  &  Clé secondaire | Identifiant de la Question référente | 3 |
| LOGIN | Chaine de caractères | Clé primaire  &  Clé secondaire | Identifiant de l’utilisateur auteur de la réponse | ndargazan001 |
| REPONSE | Chaine de caractères | / | Réponse de l’utilisateur | Oui |

La clé primaire est la combinaison de l’identifiant de la question et de l’identifiant de l’utilisateur permettant ainsi d’obtenir efficacement les réponses des utilisateurs et des questions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parrainer | | | | |
| Nom | Type | Contrainte | Description | Exemple |
| LOGIN\_PARRAIN | Chaine de caractères | Clé primaire  &  Clé étrangère | Identifiant du parrain | ndargazan001 |
| LOGIN\_FILLEUL | Chaine de caractères | Clé primaire  &  Clé étrangère | Identifiant du filleul | amaurice006 |

Etant donné que les parrains peuvent avoir plusieurs filleuls, cette table est indispensable.

# Schéma relationnel

*Note : « TYPEQUESTION » dans la table « question » est en réalité « TypeASSOS », que l’on voit aussi dans le schéma relationnel textuel.*

Utilisateur(login, nom, prenom, password, role, niveau, valide)

Parrainer(#login\_parrain, #login\_filleul)

Formulaire(id, TypeASSOS, date\_debut, date\_final)

Question(id, intitule, #id\_formulaire, image, visibilite, typeQuestion, nbReponse, nbCaractereMax)

Proposition(id, texte, couleur, #id\_question)

Repondre(#id\_question,#login, reponse)

# Scripts SQL (Création)

CREATE TABLE UTILISATEUR (

    LOGIN VARCHAR(64) PRIMARY KEY,

    NOM VARCHAR(64) DEFAULT NULL,

    PRENOM VARCHAR(64) DEFAULT NULL,

    PASSWORD VARCHAR(64) DEFAULT NULL,

    ROLE ENUM('admin', 'user') DEFAULT 'user',

    NIVEAU INT(11),

    VALIDE ENUM('oui', 'non') DEFAULT 'non');

CREATE TABLE Proposition(

    ID INT(11) AUTO\_INCREMENT PRIMARY KEY,

    TEXTE VARCHAR(500) NOT NULL,

    COULEUR VARCHAR(7) DEFAULT '#808080',

    ID\_QUESTION INT(11),

    FOREIGN KEY (ID\_QUESTION) REFERENCES QUESTION(ID));

CREATE TABLE REPONDRE (

    ID\_QUESTION INT(11),

    LOGIN VARCHAR(64),

    REPONSE VARCHAR(500) NOT NULL,

    PRIMARY KEY (ID\_QUESTION, LOGIN),

    FOREIGN KEY (ID\_QUESTION) REFERENCES QUESTION(ID),

    FOREIGN KEY (LOGIN) REFERENCES UTILISATEUR(LOGIN));

CREATE TABLE PARRAINER (

    LOGIN\_PARRAIN VARCHAR(64),

    LOGIN\_FILLEUL VARCHAR(64),

    PRIMARY KEY (LOGIN\_PARRAIN, LOGIN\_FILLEUL),

    FOREIGN KEY (LOGIN\_PARRAIN) REFERENCES UTILISATEUR(LOGIN),

    FOREIGN KEY (LOGIN\_FILLEUL) REFERENCES UTILISATEUR(LOGIN));

CREATE TABLE QUESTION (

    ID INT(11) AUTO\_INCREMENT PRIMARY KEY,

    INTITULE VARCHAR(50) NOT NULL,

    ID\_FORMULAIRE INT(11),

    IMAGE VARCHAR(260) DEFAULT '..\\sources\\images\\imgplaceholder.jpg',

    VISIBILITE ENUM('all', '1', '2'),

    TYPEQUESTION ENUM('QCM', 'LIBRE'),

    NBREPONSE INT(11) DEFAULT 1,

    NBCARACTEREMAX INT(11) DEFAULT 255,

    FOREIGN KEY (ID\_FORMULAIRE) REFERENCES FORMULAIRE(ID));

CREATE TABLE FORMULAIRE (

    ID INT(11) AUTO\_INCREMENT PRIMARY KEY,

    TYPEQUESTION ENUM('auto', '1', '2'),

    DATE\_DEBUT DATE,

    DATE\_FINAL DATE);

# Scripts SQL (interrogation)

------------------FORMULAIRE-------------

INSERT INTO Formulaire(typeAssociation,DateDebut,DateFin) VALUES(':type',':dateDebut',':dateFin');

UPDATE Formulaire SET typeAssociation = ':type', DateDebut = ':dateDebut', DateFin = ':dateFin';

DELETE FROM Formulaire WHERE idFormulaire = ':idFormulaire’;

------------------QUESTION---------------

INSERT INTO Question (INTITULE,ID\_FORMULAIRE,IMAGE,TYPEQUESTION,NBREPONSE,NBCARACTEREMAX) VALUES(':intitule',':idFormulaire',':image',':type',':nbReponse',':nbCaractereMax’);

UPDATE Question SET INTITULE = ':intitule', IMAGE = ':image', TYPEQUESTION = ':type', NBREPONSE = ':nbReponse', NBCARACTEREMAX = ':nbCaractereMax’;

DELETE FROM Question WHERE idQuestion = ':idQuestion';

# Scripts SQL (Insertion)

-----------------Utilisateur-----------

INSERT INTO Utilisateur(LOGIN,NOM,PRENOM,PASSWORD,ROLE,NIVEAU,VALIDITE) VALUES(':login',':nom',':prenom',':password',':role',':niveau',':validite');

-- L’utilisateur active son compte et défini son mot de passe

UPDATE Utilisateur SET password = :password, VALIDE = 1 WHERE login = :login;

-- Pour la connexion, on vérifie que l'utilisateur existe

SELECT \* FROM Utilisateur WHERE LOGIN = :login;

-- Si l'utilisateur souhaite supprimer son compte, on le supprime pas mais reset tout ses attributs

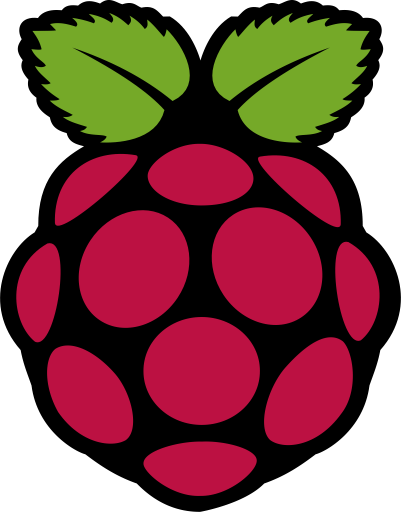
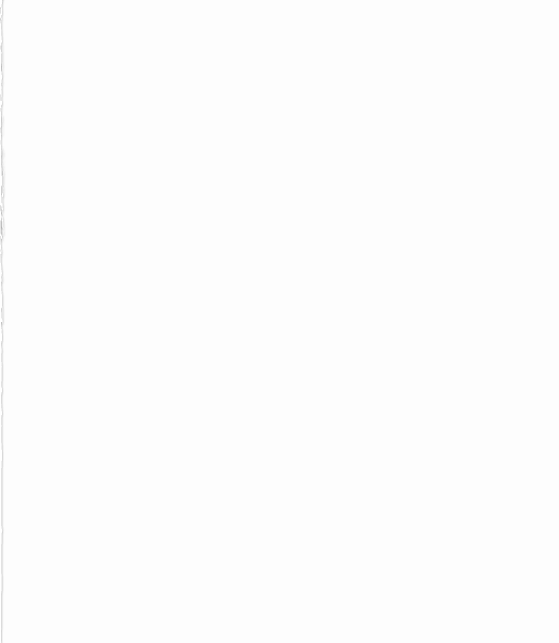
UPDATE Utilisateur SET VALIDE=0,NOM=NULL,PRENOM=NULL, PASSWORD='', role='user', NIVEAU='' WHERE LOGIN=:login;

# Technologies utilisées

Une image contenant texte, embarcation, navire de navigation

Description générée automatiquement





**Technologies**

* SQL
* MySQL
* PHP
* Serveur web

Nous avons décider d’utiliser le langage SQL avec le service MySQL ainsi que l’interface PhpMyAdmin pour simplifier l’interaction avec la base de données.