CATRICE Grégoire

Stephanne Ferreira

HU Xian mei

IFT2935 – Bases de données

LEARNING VILLAGE

1. Description du Projet

2. Modèle entité-relations

3. Analyse du modèle ER

4. définition de la base dans le LDD

5. Les requêtes

6. fonctionnalités

7. Pour accéder au site

1. Description du Projet

Nous voulons créer un mini réseau social d'apprentissage, appelé Learning village. Dans ce réseau des utilisateurs peuvent se connecter en tant qu'apprenants ou instructeurs. Les apprenants peuvent s'intéresser à des thèmes particuliers d'apprentissage en informatique (bases de données, systèmes informatiques, algorithmiques et programmations,...). Chaque thème contient un ensemble de leçons avec des tutoriels  (sous forme de textes par exemple, ou autres...) et des exercices (QCM, problèmes à résoudre,...). Un tutoriel peut avoir un niveau (débutant, expert). De même, les exercices peuvent avoir un certain nombre d'étoiles pour qualifier leurs niveaux de difficulté.

Dans ce réseau, les apprenants peuvent consulter des tutoriels, faires des exercices, ou répondre aux questions relatives à un thème. Ils peuvent aussi poser des questions sur leurs difficultés ou sous forme de défis (challenges) aux autres étudiants. Les instructeurs, en revanche, proposent des tutoriels ou des exercices, répondent aux questions des apprenants, et évaluent des apprenants (évaluer les réponses des apprenants sur les exercices). Les apprenants peuvent aussi évaluer les instructeurs (donner un nombre d'étoiles par exemple). Un étudiant peut être connecté avec un ou plusieurs étudiants (workgroup) pour travailler ensemble sur un thème. Seuls les apprenants du même groupe peuvent voir les réponses de leurs collègues sur les questions. Un apprenant peut appartenir à plusieurs workgroups. Chaque workgroup possède un nom et peut posséder au moins deux membres. Learning village permet aussi de faire des recherches sur les: instructeurs, apprenants, exercices et tutoriels selon des critères relatifs: par exemple le groupe qui a le plus grand nombre d'adhérant, etc...

2. Modèle Entités-Relations

**(le fichier ci-joint : modeleER.png est l’image en taille réel)**

3. Analyse du modèle ER

Ce modèle ER est la dernière version de notre base de données. Nous avons intégrer la totalité des fonctions demandées ainsi que plusieurs supplémentaire comme le chat.

De façon général, l’implémentation du modèle est assez logique cependant il y a certaines contraintes que nous avons dû interpréter par nous même. Dans une première page, l’utilisateur devra donner ses informations personnelles c’est ensuite, qu’il sera différencié soit en PROF soit en ELEVE et ainsi accéder aux fonctionnalités qui lui sont propres.

« *Les apprenants peuvent consulter des tutoriels, faires des exercices, ou répondre aux questions relatives à un thème »* nous avons interprété cette formalité en créant l’entité INTERESSE. Si un ELEVE est intéressé par un THEME, il pourra avoir accès à l’intégralité du contenu du THEME (LECON, TUTO, EXO). Lorsqu’un PROF est responsable d’un THEME, il peut tout modifier à l’intérieur (les LECONs, les TUTOs, les EXO et corriger et noter les EXO fait par les ELEVES lié à son THEME)

*« Ils peuvent aussi poser des questions sur leurs difficultés ou sous forme de défis (challenges) aux autres étudiants »*Ici, nous avons créé une seule relation QUESTION de ELEVE vers UTILISATEUR par conséquent, l’ELEVE pourra poser sa question à n’importe qui. Dans le cas où l’étudiant fait parti d’un WORKGROUP, il pourra poser sa question dans le mini forum de son WORKGROUP mais cette fonctionnalité implique des modifications dans le modèle que nous implémenterons si le temps nous le permet.

Lorsqu’un ELEVE fait (par l’intermédiaire de FAIRE ) un exercice, il ne remplit que l’attribut ‘rep’. Les autres attributs sont NULL. Ensuite, lorsque le PROF a donné une note, l’ELEVE a accès à la correction de l’EXO. Nous avons considéré que les EXO étaient des composants des TUTO ce qui permettra à l’ELEVE de tester ses connaissances au cours du TUTO.

Nous avons réunis tous les types de question en un seul : QUESTION. C’est en fonction du type de destinataire que le type de la question sera déterminé. Si c’est un autre ELEVE alors il s’agit d’un défi (il y aura un attribut booléen dans QUESTION : ‘défi’ qui déterminera s’il s’agit d’un défi ou d’une question à un prof). L’ELEVE répondant aura alors un temps pour répondre. S’il répond au défi assez rapidement, il augmentera son compteur de défi réussis. S’il s’agit d’un prof, la QUESTION prend la forme d’un simple message.

4. définition de la base dans le LDD

Le LDD suivant est celui correspondant à notre base et nous en avons vérifié la normalité.

CREATE TABLE UTILISATEUR(

iduser INTEGER NOT NULL AUTO\_INCREMENT, PRIMARY KEY(iduser), UNIQUE(iduser),

nom VARCHAR(12) NOT NULL,

prenom VARCHAR(12) NOT NULL, UNIQUE(nom, prenom),

mail VARCHAR(24) NOT NULL,

pwd VARCHAR(16) NOT NULL,

prof BOOLEAN,

date\_inscription DATE,

last\_co DATE,

co BOOLEAN

ip TEXT);

CREATE TABLE THEME(

idtheme INTEGER NOT NULL AUTO\_INCREMENT, PRIMARY KEY(idtheme), UNIQUE(idtheme),

nomtheme VARCHAR(16) NOT NULL,

def VARCHAR(150) NOT NULL);

CREATE TABLE ESTFAITDE(

idetd INTEGER AUTO\_INCREMENT ,PRIMARY KEY(idetd),

idlecon INTEGER NOT NULL ,

idtuto INTEGER NOT NULL );

ALTER TABLE ESTFAITDE

ADD CONSTRAINT idlecon

FOREIGN KEY (idlecon) REFERENCES LECON(idlecon) ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE MODIFIE(

idm INTEGER NOT NULL AUTO\_INCREMENT, PRIMARY KEY(idm), UNIQUE(idm),

idtheme INTEGER NOT NULL,

iduser INTEGER NOT NULL);

ALTER TABLE MODIFIE

ADD CONSTRAINT idtheme

FOREIGN KEY (idtheme) REFERENCES THEME(idtheme) ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE LECON(

idlecon INTEGER NOT NULL AUTO\_INCREMENT, PRIMARY KEY(idlecon), UNIQUE(idlecon),

nomlecon VARCHAR(16) NOT NULL,

text TEXT NOT NULL,

idc INTEGER);

ALTER TABLE LECON

ADD CONSTRAINT idc

FOREIGN KEY (idc) REFERENCES CONTIENT(idc) ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE CONTIENT(

idc INTEGER AUTO\_INCREMENT ,PRIMARY KEY(idc),

idtheme INTEGER NOT NULL ,

idlecon INTEGER NOT NULL);

ALTER TABLE CONTIENT

ADD CONSTRAINT idtheme

FOREIGN KEY (idtheme) REFERENCES THEME(idtheme) ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE CONTIENT

ADD CONSTRAINT idlecon

FOREIGN KEY (idlecon) REFERENCES LECON(idlecon) ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE TUTO(

idtuto INTEGER NOT NULL AUTO\_INCREMENT, PRIMARY KEY(idtuto), UNIQUE(idtuto),

nomtuto VARCHAR(16) NOT NULL,

text VARCHAR(150) NOT NULL,

niveau VARCHAR(10),

idetd INTEGER);

ALTER TABLE TUTO

ADD CONSTRAINT idetd

FOREIGN KEY (idetd) REFERENCES ESTFAITDE(idetd) ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE COMPOSE(

idcmp INTEGER AUTO\_INCREMENT ,PRIMARY KEY(idcmp),

idtuto INTEGER NOT NULL ,

idexo INTEGER NOT NULL);

ALTER TABLE COMPOSE

ADD CONSTRAINT idtuto

FOREIGN KEY (idtuto) REFERENCES TUTO(idtuto) ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE EXO(

idexo INTEGER NOT NULL AUTO\_INCREMENT, PRIMARY KEY(idexo), UNIQUE(idexo),

nomexo VARCHAR(16) NOT NULL,

text VARCHAR(150) NOT NULL,

nbetoile FLOAT

idcmp INTEGER) ;

ALTER TABLE EXO

ADD CONSTRAINT idcmp

FOREIGN KEY (idcmp) REFERENCES COMPOSE(idcmp) ON DELETE CASCADE

ON UPDATE CASCADE; (exécuté sql ?)

CREATE TABLE WORKGROUP(

idw INTEGER NOT NULL AUTO\_INCREMENT, PRIMARY KEY(idW), UNIQUE(idW),

nomw VARCHAR(16) NOT NULL,

text VARCHAR(150)

iduser VAR NOT NULL);

CREATE TABLE ESTCONNECTE(

iduser INTEGER NOT NULL AUTO\_INCREMENT,

idw INTEGER NOT NULL,

PRIMARY KEY(iduser, idw));

ALTER TABLE ESTCONNECTE

ADD CONSTRAINT iduser

FOREIGN KEY (iduser) REFERENCES UTILISATEUR(iduser) ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE ESTCONNECTE

ADD CONSTRAINT idw

FOREIGN KEY (idw) REFERENCES WORKGROUP(idw) ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE CONCERNE(

idw INTEGER NOT NULL,

idtheme INTEGER NOT NULL,

PRIMARY KEY(idw, idtheme));

ALTER TABLE CONCERNE

ADD CONSTRAINT idw

FOREIGN KEY (idw) REFERENCES WORKGROUP(idw) ON DELETE CASCADE

ON UPDATE CASCADE; (exécuté sql ?)

ALTER TABLE CONCERNE

ADD CONSTRAINT idtheme

FOREIGN KEY (idtheme) REFERENCES THEME(idtheme) ON DELETE CASCADE

ON UPDATE CASCADE; (exécuté sql ?)

CREATE TABLE INTERESSE(

iduser INTEGER NOT NULL AUTO\_INCREMENT,

idtheme INTEGER NOT NULL,

PRIMARY KEY(iduser, idtheme));

ALTER TABLE INTERESSE

ADD CONSTRAINT idtheme

FOREIGN KEY (idtheme) REFERENCES THEME(idtheme) ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE EVALUE(

idu1 INTEGER NOT NULL,

idu2 INTEGER NOT NULL,

note INT

PRIMARY KEY(idu1, idu2));

CREATE TABLE WORKGROUPT(

idw INTEGER NOT NULL AUTO\_INCREMENT, PRIMARY KEY(idw), UNIQUE(idw),

nomw VARCHAR(16) NOT NULL,

text VARCHAR(150)

iduser INT NOT NULL,

idt INT);

CREATE TABLE QUESTION(

idq INTEGER NOT NULL AUTO\_INCREMENT, PRIMARY KEY(idq), UNIQUE(idq),

iduser1 INTEGER NOT NULL , FOREIGN KEY(iduser1) REFERENCES UTILISATEUR(iduser) ,

iduser2 INTEGER NOT NULL , FOREIGN KEY(iduser2) REFERENCES UTILISATEUR(iduser) ,

texte TEXT NOT NULL,

sujet TEXT NOT NULL,

rep BOOLEAN DEFAULT FALSE );

CREATE TABLE WORKGQ(

idwg INTEGER NOT NULL AUTO\_INCREMENT, PRIMARY KEY(idwg), UNIQUE(idwg),

idg1 INTEGER NOT NULL ,

idg2 INTEGER NOT NULL ,

texte TEXT NOT NULL,

sujet TEXT NOT NULL );

ALTER TABLE WORKGQ

ADD CONSTRAINT idg2

FOREIGN KEY (idg3) REFERENCES WORKGROUP(idw) ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE CHAT(

id INTEGER NOT NULL AUTO\_INCREMENT, PRIMARY KEY(id), UNIQUE(id),

idu INTEGER NOT NULL ,

nom TEXT NOT NULL ,

texteTEXT NOT NULL );

5. Les requêtes

$sql = "SELECT nomw,text,iduser,idt FROM WORKGROUPT WHERE idw = '".$\_GET['id']."' ";

$sql = "DELETE FROM WORKGROUPT WHERE idw = '".$\_GET['id']."'";

$req = $bdd->prepare("INSERT INTO WORKGROUP(nomw,text,iduser) VALUES( :nomw, :text , :iduser)");

$sql = "SELECT idw FROM WORKGROUP WHERE nomw = '$nomw' ";

$req = $bdd->prepare("INSERT INTO ESTCONNECTE(iduser,idw) VALUES(:iduser, :idw)");

$req = $bdd->prepare("INSERT INTO CONCERNE(idtheme,idw) VALUES(:idtheme, :idw)");

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT iduser FROM WORKGROUP WHERE idw = '".$\_GET['id']."' ";

$req = $bdd->prepare("INSERT INTO ESTCONNECTE (idw,iduser) VALUES(:idw, :iduser)");

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT nom FROM UTILISATEUR WHERE iduser = '".$\_SESSION['userid']."' ";

$req = $bdd->prepare('INSERT INTO CHAT (nom,text,idu) VALUES(:nom,:text,:idu)');

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$req = $bdd->prepare("INSERT INTO WORKGROUPT(nomw,text,iduser,idt) VALUES( :nomw, :text , :iduser,:idt)");

$sql = "SELECT idw, FROM WORKGROUPT WHERE nomw = '".$nomw."' ";

$req = $bdd->prepare("INSERT INTO QUESTION(iduser1,iduser2,sujet,texte,rep) VALUES(:iduser1,:iduser2,:sujet,:texte,:rep)");

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT iduser,nom,prenom FROM UTILISATEUR WHERE prof = 0 ORDER BY nom";

$req = $bdd->prepare("INSERT INTO QUESTION(iduser2,texte,iduser1,sujet,date\_question) VALUES(:iduser2,:texte,:iduser1,:sujet,:date\_question)");

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT idw FROM WORKGROUP WHERE nomw = '".$idg1."' ";

$req = $bdd->prepare("INSERT INTO WORKGQ(idg3,texte,idg1,sujet) VALUES(:idg3,:texte,:idg1,:sujet)");

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT iduser FROM MODIFIE WHERE idtheme = '".$\_GET['id']."' ";

$sql = "SELECT nomexo,text,nbetoile FROM EXO WHERE idexo = '".$\_GET['idt']."'";

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT iduser1,idexo FROM FAIRE WHERE idexo = '".$\_GET['ide']."' ";

$sqll = "SELECT UTILISATEUR.iduser,nom,FAIRE.idf FROM FAIRE INNER JOIN UTILISATEUR ON FAIRE.iduser1 WHERE FAIRE.idexo = '".$dn['idexo']."' AND FAIRE.iduser1 = UTILISATEUR.iduser";

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sqll = "SELECT idexo,note,remarque FROM FAIRE WHERE FAIRE.iduser1 = '".$\_SESSION['userid']."' AND FAIRE.note != '-1' ";

$sql = "SELECT nomexo FROM EXO WHERE idexo = '".$data['idexo']."' ";

$sql = "SELECT note FROM EVALUE WHERE idu1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT iduser FROM MODIFIE WHERE idtheme = '".$\_GET['id']."' ";

$sql = "SELECT nomexo,nbetoile,text FROM EXO WHERE idexo = '".$\_GET['ide']."'";

$req = $bdd->prepare('INSERT INTO FAIRE(rep,iduser1,idexo) VALUES(:rep, :iduser1, :idexo)');

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT nomexo,text,nbetoile FROM EXO WHERE idexo = '".$\_GET['id']."' ";

$sql = "UPDATE EXO SET nomexo='".$nomexo."' , text = '".$text."' ,nbetoile = '".$nbetoile."' WHERE idexo = '".$\_GET['id']."' ";

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "DELETE FROM EXO WHERE idexo = '".$\_GET['id']."'";

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$req = $bdd->prepare("INSERT INTO EXO(nomexo,text,nbetoile) VALUES(:nomexo, :text, :nbetoile)");

$sql = "SELECT idexo FROM EXO WHERE nomexo = '$nomexo' ";

$req = $bdd->prepare("INSERT INTO COMPOSE(idtuto,idexo) VALUES(:idtuto, :idexo)");

$sql = "SELECT idcmp,idexo FROM COMPOSE WHERE idexo = '".$dn['idexo']."' ";

$sql = "UPDATE EXO SET idcmp='".$dn['idcmp']."' WHERE idexo = '".$dn['idexo']."' ";

$sql = "SELECT nomtheme FROM THEME WHERE idtheme = '".$\_GET['idt']."' ";

$sql = "SELECT iduser FROM INTERESSE WHERE idtheme = '".$\_GET['idt']."' ";

$req = $bdd->prepare("INSERT INTO QUESTION(iduser1,iduser2,sujet,texte,rep) VALUES(:iduser1,:iduser2,:sujet,:texte,:rep)");

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$req = $bdd->prepare('INSERT INTO UTILISATEUR(nom,prenom,pwd,mail,prof,date\_inscription) VALUES(:nom, :prenom, :passe, :email, :prof, :date)');

$sql = "SELECT iduser FROM MODIFIE WHERE idtheme = '".$\_GET['idt']."' ";

$sql = "SELECT nomtheme,def FROM THEME WHERE idtheme = '".$\_GET['idt']."'";

$sql = "SELECT nomlecon,text FROM LECON WHERE idlecon = '".$\_GET['idl']."'";

$sql = "SELECT TUTO.idtuto,nomtuto,niveau FROM ESTFAITDE INNER JOIN TUTO ON ESTFAITDE.idtuto WHERE ESTFAITDE.idlecon = '".$\_GET['idl']."' AND ESTFAITDE.idtuto = TUTO.idtuto";

$sql = "SELECT nomlecon,text FROM LECON WHERE idlecon = '".$\_GET['id']."' ";

$sql = "UPDATE LECON SET nomlecon ='".$nomlecon."' , text = '".$text."' WHERE idlecon = '".$\_GET['id']."' ";

$sql = "DELETE FROM LECON WHERE idlecon = '".$\_GET['id']."'";

$sql = "UPDATE QUESTION SET rep ='1' WHERE idq = '".$\_GET['idq']."' ";

$sqll = "SELECT nom,prenom FROM UTILISATEUR WHERE iduser = '".$\_GET['id']."' ";

$sqll = "SELECT sujet,date\_question,texte FROM QUESTION WHERE idq = '".$\_GET['idq']."' AND iduser1 = '".$\_SESSION['userid']."'";

$sql = "SELECT UTILISATEUR.iduser FROM ESTCONNECTE INNER JOIN UTILISATEUR ON ESTCONNECTE.iduser WHERE ESTCONNECTE.idw = '".$\_GET['id']."' AND ESTCONNECTE.iduser = UTILISATEUR.iduser AND ESTCONNECTE.iduser = '".$\_SESSION['userid']."' ";

$sql = "SELECT sujet FROM WORKGQ WHERE idwg = '".$\_GET['idw']."' ";

$sqll = "SELECT nomw,texte FROM WORKGROUP WHERE idw = '".$dn['idg3']."' ";

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT iduser2,idq,sujet,date\_question FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sqll = "SELECT nom,prenom FROM UTILISATEUR WHERE iduser = '".$data['iduser2']."' ";

$sql = "DELETE FROM QUESTION WHERE idq= '".$\_GET['id']."'";

$req = $bdd->prepare("INSERT INTO EVALUE(idu1,idu2,note) VALUES( :idu1,:idu2,:note)");

$sql = "SELECT iduser,nom,prenom, date\_inscription FROM UTILISATEUR WHERE prof = 1 ORDER BY nom";

$sql = "SELECT prof,prenom,nom,iduser,date\_inscription,last\_co FROM UTILISATEUR WHERE iduser = '".$\_GET['id']."' ";

$sqlll = "SELECT idw FROM ESTCONNECTE WHERE iduser = '".$\_GET['id']."'";

$sqll = "SELECT nomw FROM WORKGROUP WHERE idw = '".$dat['idw']."'";

$sql = "SELECT idtheme FROM INTERESSE WHERE iduser = '".$\_GET['id']."' ";

$sqll = "SELECT nomtheme FROM THEME WHERE idtheme = '".$idtheme."'";

$sql = "DELETE FROM WORKGROUPT WHERE idw = '".$\_GET['id']."'";

$req = $bdd->prepare('INSERT INTO INTERESSE(iduser,idtheme) VALUES(:iduser, :idtheme)');

$reponse = $bdd->query('SELECT nom,text FROM CHAT ORDER BY ID DESC LIMIT 0, 10');

$sql = "SELECT pwd,iduser FROM UTILISATEUR WHERE mail = '".$username."'";

$sql = "UPDATE UTILISATEUR SET last\_co = '$last\_co' WHERE iduser = '$idu' ";

$sql = "UPDATE UTILISATEUR SET ip = '".$\_SERVER['REMOTE\_ADDR'].' '.$dn['ip']."' WHERE iduser = '$idu' ";

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT idtheme,nomtheme,def FROM THEME ORDER BY nomtheme";

$req = $bdd->prepare('INSERT INTO THEME(nomtheme,def) VALUES(:nomtheme, :def)');

$sql = "SELECT idtheme FROM THEME WHERE nomtheme = '$nomtheme' ";

$req = $bdd->prepare("INSERT INTO MODIFIE(iduser,idtheme) VALUES(:iduser, :idtheme)");

$req = $bdd->prepare("INSERT INTO LECON(nomlecon,text) VALUES(:nomlecon, :text)");

$sql = "SELECT idlecon FROM LECON WHERE nomlecon = '$nomlecon' ";

$req = $bdd->prepare("INSERT INTO CONTIENT(idtheme,idleconn) VALUES(:idtheme, :idleconn)");

$sql = "SELECT idc FROM CONTIENT WHERE idleconn = '".$dn['idlecon']."' ";

$sql = "UPDATE LECON SET idc='".$dn['idc']."' WHERE idlecon = '".$idleconn."' ";

$sql = "SELECT nomtheme FROM THEME WHERE idtheme = '".$\_GET['id']."' ";

$sql = "SELECT iduser FROM INTERESSE WHERE idtheme = '".$\_GET['id']."' ";

$req = $bdd->prepare("INSERT INTO QUESTION(iduser1,iduser2,sujet,texte,rep) VALUES(:iduser1,:iduser2,:sujet,:texte,:rep)");

$req = $bdd->prepare("INSERT INTO MODIFIE(iduser,idtheme) VALUES(:iduser, :idtheme)");

$sql = "SELECT iduser FROM MODIFIE WHERE idtheme = '".$\_GET['id']."' ";

$sql = "SELECT nomtheme,def FROM THEME WHERE idtheme = '".$\_GET['id']."' ORDER BY nomtheme";

$sql = "SELECT LECON.idlecon,LECON.nomlecon FROM CONTIENT INNER JOIN LECON ON CONTIENT.idc WHERE CONTIENT.idtheme = '".$\_GET['id']."' AND CONTIENT.idc = LECON.idc ";

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "UPDATE THEME SET nomtheme='".$nomtheme."' , def = '".$def."' WHERE idtheme = '".$\_GET['id']."' ";

$sql = "DELETE FROM THEME WHERE idtheme = '".$\_GET['id']."'";

$sql = "SELECT iduser FROM MODIFIE WHERE idtheme = '".$\_GET['id']."' ";

$sql = "SELECT nomtuto,niveau FROM TUTO WHERE idtuto = '".$\_GET['idt']."'";

$sql = "SELECT EXO.idexo,nomexo,nbetoile FROM COMPOSE INNER JOIN EXO ON COMPOSE.idexo WHERE COMPOSE.idtuto = '".$\_GET['idt']."' AND EXO.idexo = COMPOSE.idexo";

$sql = "SELECT nomtuto,text,niveau FROM TUTO WHERE idtuto = '".$\_GET['id']."' ";

$sql = "UPDATE TUTO SET nomtuto ='".$nomtuto."' , text = '".$text."' , niveau = '".$niveau."' WHERE idlecon = '".$\_GET['id']."' ";

$sql = "DELETE FROM TUTO WHERE idtuto = '".$\_GET['id']."'";

$req = $bdd->prepare("INSERT INTO TUTO(nomtuto,text,niveau) VALUES(:nomtuto, :text, :niveau)");

$sql = "SELECT idtuto FROM TUTO WHERE nomtuto = '$nomtuto' ";

$req = $bdd->prepare("INSERT INTO ESTFAITDE(idlecon,idtuto) VALUES(:idlecon, :idtuto)");

$sql = "SELECT idetd,idtuto FROM ESTFAITDE WHERE idtuto = '".$dn['idtuto']."' ";

$sql = "UPDATE TUTO SET idetd='".$dn['idetd']."' WHERE idtuto = '".$dn['idtuto']."' ";

$sql = "SELECT nomtheme FROM THEME WHERE idtheme = '".$\_GET['idt']."' ";

$sql = "SELECT iduser FROM INTERESSE WHERE idtheme = '".$\_GET['idt']."' ";

$req = $bdd->prepare("INSERT INTO QUESTION(iduser1,iduser2,sujet,texte,rep) VALUES(:iduser1,:iduser2,:sujet,:texte,:rep)");

$sql = "SELECT rep FROM QUESTION WHERE iduser1 = '".$\_SESSION['userid']."' ";

$sql = "SELECT rep FROM FAIRE WHERE idf= '".$\_GET['idf']."' AND iduser1 = '".$\_GET['idu']."' ";

$sql = "UPDATE FAIRE SET note ='".$note."' , remarque = '".$remarque."' , iduser2 = '".$iduser2."' WHERE iduser1 = '".$\_GET['idu']."' AND idf = '".$\_GET['idf']."'";

$sql = "SELECT idw,nomw FROM WORKGROUP";

$sqll = "SELECT UTILISATEUR.iduser,nom FROM ESTCONNECTE INNER JOIN UTILISATEUR ON ESTCONNECTE.iduser WHERE ESTCONNECTE.idw = '".$dat['idw']."' AND ESTCONNECTE.iduser = UTILISATEUR.iduser";

$sql = "SELECT idw,nomw FROM WORKGROUP WHERE idw IN (SELECT idw FROM ESTCONNECTE WHERE iduser='".$\_SESSION['userid']."')";

$sql = "SELECT nomw,text FROM WORKGROUP WHERE idw = '".$\_GET['id']."' ";

$sql = "UPDATE WORKGROUP SET nomw ='".$nomw."' , text = '".$text."' WHERE idw = '".$\_GET['id']."' ";

$sql = "SELECT iduser FROM WORKGROUP WHERE idw = '".$\_GET['id']."' ";

$sql = "SELECT now,text FROM WORKGROUP WHERE idw = '".$\_GET['id']."' ";

$sql = "SELECT UTILISATEUR.iduser,nom FROM ESTCONNECTE INNER JOIN UTILISATEUR ON ESTCONNECTE.iduser WHERE ESTCONNECTE.idw = '".$\_GET['id']."' AND ESTCONNECTE.iduser = UTILISATEUR.iduser";

$sql = "SELECT UTILISATEUR.iduser FROM ESTCONNECTE INNER JOIN UTILISATEUR ON ESTCONNECTE.iduser WHERE ESTCONNECTE.idw = '".$\_GET['id']."' AND ESTCONNECTE.iduser = UTILISATEUR.iduser AND ESTCONNECTE.iduser = '".$\_SESSION['userid']."' ";

$sql = "SELECT idg3,igw,sujet FROM WORKGQ WHERE idg1 = '".$\_GET['id']."' ";

$sqll = "SELECT nomw FROM WORKGROUP WHERE idw = '".$data['idg3']."' ";

$sql = "DELETE FROM WORKGROUP WHERE idw = '".$\_GET['id']."'";

6. fonctionnalités

Comme demandé dans le sujet, notre site est donc un petit réseau social qui relie des profs et des élèves. Les fonctionnalités qui ont été implémentées sont les suivantes :

-connexion en temps que prof ou en temps qu’élève

-possibilité de s’intéresser à un thème. Ces thèmes sont créés par un ou plusieurs professeurs qui les administrent.

-un thème est composé de leçons et de tutoriels eux même composé d’exercices

-Les tutoriels ainsi que les exercices possèdent un niveau de difficulté

-l’apprenant peut répondre à des questions relatives à un thème

-Un système de messagerie permet de pouvoir poser des questions à des professeurs comme à d’autres apprenants

-Les professeurs corrigent les élèves qui ont répondu à des exercices relatifs aux thèmes qu’ils administrent.

-Les Workgroups peuvent être créés comme définis dans le cahier des charges

Pour résumer, toutes les fonctionnalités demandées ont été implémentées. Il reste toujours quelques petites imperfections à notre site que nous n’avons pas su corriger mais celles-ci ne dérangent pas le bon fonctionnement. De plus, nous avons ajouté quelques fonctionnalités au cahier des charges :

-un mini chat

-Un professeur ayant créé un thème peut ajouter d’autres professeurs en temps qu’administrateurs. Ce qui permet de gérer les tutoriels et les exercices liés à ce thème

-la possibilité de pouvoir supprimer les composants d’un thème, le thème lui-même, les messages et les Workgroups sans endommager la base de données.

-Lors de la modification d’un thème, les personnes intéressées reçoivent un message pour les en informer

-les Workgroups peuvent aussi interagir entre eux via la messagerie.

**Quelques requêtes de recherches :**

Le nom du groupe avec le plus de membre

SELECT DISTINCT nomw

FROM WORKGROUP WHERE idw = (select idw from ESTCONNECTE group by idw ORDER BY count(iduser) DESC LIMIT 0 , 1)

Rechercher un nom :

**SELECT** \*

**FROM** `UTILISATEUR`

**WHERE** nom **= 'XXXX'**

**LIMIT** 0 , 30

Toute les notes qui ont la moyenne ou plus

**SELECT** \*

**FROM** `FAIRE`

**WHERE** note **LIKE** '1\_'

**LIMIT** 0 , 30

Afficher les 3 meilleurs note :

**SELECT** note

**FROM** `FAIRE`

**ORDER** **BY** note **DESC**

**LIMIT** 0 , 3

Afficher les X meilleurs nom/ prenom /note :

**SELECT** nom, prenom, note

**FROM** UTILISATEUR

**INNER** **JOIN** FAIRE

**WHERE** FAIRE.iduser1 = UTILISATEUR.iduser

**ORDER** **BY** note **DESC**

**LIMIT** 0 , X

Afficher les noms de tous les groupes dans lesquels il y a au moins un éleve qui a une note en dessous de 10 :

SELECT DISTINCT nomw

FROM WORKGROUP

INNER JOIN ESTCONNECTE ON WORKGROUP.idw =  ESTCONNECTE.idw

WHERE ESTCONNECTE.iduser IN

(SELECT iduser1

FROM FAIRE WHERE note < 10)

Afficher le nombre de personne que contient le groupe X :

**SELECT** COUNT( idw ) **AS** nb

**FROM** ESTCONNECTE

**WHERE** idw = 'X'

7. Pour accéder au site :

L’url du site est : http:// soozig.com/Learningvillage/index.php

Pour se connecter, en temps que professeur, il faut un mot de passe. Ce mot de passe est ‘prof’. Les fonctionnalités qui fonctionnent sont celles qui ont été décrites dans le cahier des charges.

Bonne visite.