Universite de Montpellier
Rapport de projet Base de données HLIN511

Hérault Events

Chakib ELHOUITI : 21813619
Massili KEZZOUL : 21815514
Groupe P

19 décembre 2019

1 Introduction

Dans le cadre d'un projet commun entre deux UE de la faculté des sciences de l'univérsite de Montpellier, nous avons réaliser une base de données associée à une application web permettant la publication d'événements culturels ou sportifs dans un département donné. Nous avons choisi le département de l'Hérault. L'application peut d'ailleur être visitée à cette adresse 'http ://webpeda.etu.umontpellier.fr/e20180011096' à condition d'être sur le réseau de l'univérsite de Montepellier.

2 Modèl entité-association

2.1 Schéma E/A

Tout d'abord nous avons modélisé notre base de données de la manière suivante :

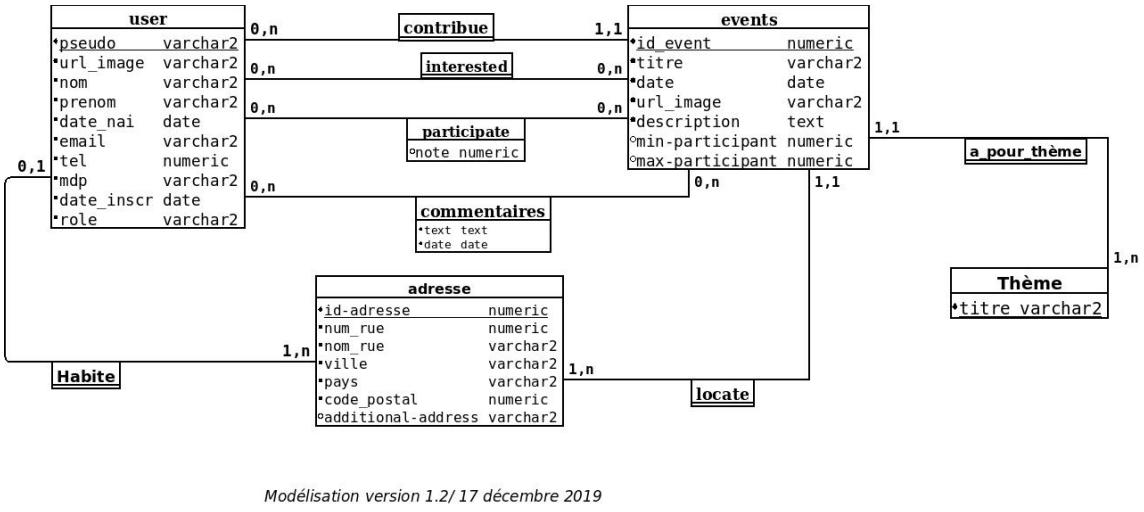

Schéma 1 – Modèl E/A

2.2 Explication du Schéma

Les tables

Adresse : Cette table permet de stocker l'ensemble des adresses qui seront utilisés par la table *User* et *Events*. On a choisi de ne pas stocker les coordonnées GPS des adresses car ils peuvent être calculés (par une API par example). On a decidé de le faire de cette manière aussi parcequ'il est préferable de ne pas demander des coordonnées GPS à un utilisateur lambda de l'application web.

User : La table *User* stock les informations personnelles des utilisateurs du site web. Un utilisateur est identifié par un pseudo qu'il renseigne à son inscription. Ce dernier doit bien évidement être unique.
Il existe trois types d'utilisateur :

- Visiteur ('visitor')
- Contributeur ('contributor')
- Administrateur ('admin')

Le type de chaque utilisateur est stocké dans l'attribut *role_user* qui est une énumération (de type ENUM). L'attribut *email*, et *tel* doivent être unique.

Events : Cette table se charge de stocker les informations relatifs à un événement. Un événement est défini par un identificateur numéric qui générer automatiquement à l'insertion d'un tuple (par AUTO_INCREMENT) et contient forcément un titre et une date (attribut de type DATETIME). Il peut aussi contenir un lien vers une image (local ou distante), une description (de type TEXT) et un nombre minimum et maximum de participant.

Theme : Chaque événement est organisé autour d'un théme donné, donc on a aussi modélisé une table thème afin de stocker tout les thèmes.

Les association

Participate : cette association permet de stocker les participation des utilisateurs à des évenements.C'est à dire un utlisateur peut participer à plusieurs évenements comme ne pas participer à aucun évenement et un évenement peut avoir aucun ou plusieurs participants. C'est pareil pour les autres associations commentaires et interested

contribuer : permet de stocker un utillisateur qui sera un contributeur dans un ou plusieurs évenements. C'est pareil pour les autres associations père/fils.

2.3 Modèle logique de données

Ensuite nous avons réaliser le modèle relationnel :
adresse (id_adresse,num_rue,nom_rue,ville,pays,code_postal,additional_adresse)
user(pseudo,nom,prenom,date_nai,email,tel,mdp,date_inscr,role,*id_adresse*)
events(id_event,titre,date,url_image,description,min-participant,max-participant,*pseudo_contributeur*,*id_*
thème(titre)
particiapte(*pseudo*,*id_event*,note)
interested(*pseudo*,*id_event*)
commentaires(*pseudo*,*id_event*,text,date)

2.4 Traduction du modèle relationnel en Langage sql (Définition des données)

Enfin on a écrit le script sql permettant de créer les tables du modèle relationnel (MYSQL).

3 Les procédures

throw_err : C'est une procédure .

c'est une fonction, elle prend en paramètres un id_event(un évenement) et calcul sa note moyenne.

nb_participate : c'est une fonction, elle prend en paramètres un id_event(un évenement) et calcul le nombre de particpants à cet évenement.

4 Les fonctions

note_event : c'est une fonction, elle prend en paramètres un id_event(un évenement) et calcul sa note moyenne.

nb_participate : c'est une fonction, elle prend en paramètres un id_event(un évenement) et calcul le nombre de particpants à cet évenement.

nb_interesses : c'est une fonction, elle prend en paramètres un id_event(un évenement) et calcul le nombre d'intéressés à cet évenement.

classement_event : c'est une fonction, elle prend en paramètres un id_event(un évenement) et retourne le classement de l'évenemets par note moyenne, c'est à dire son classement parmi tous les évenements, s'il y'a des évenements de même note, la fonction retourne son classement par note et par son id(du plus petit au plus grand).

5 Les triggers

Sur toutes les foncionnalitées demandées, nous les avons toutes implementées sauf :

— La visualisation de tout les événements en mode cartographique, néanmoins nous avons pu implementées dans la page d'un seul événement donné, sa position dans une carte (avec OpenLayers).

Par contre, nous avons implementé la possibilité pour un utilisateur, une fois inscrit, de modifier ses informations personnels, ajouter une photo pour son profil et aussi la possibilité de supprimer son compte.

On a aussi ajouté la possibilité pour un utilisateur de s'interessé à un événement avant d'y participer.

6 Conclusion

6.1 Les problèmes recontrés

Lors du développement du projet, nous n'avons rencontré aucun problème particulier. En revanche lors du déploiment de l'application sur le serveur de la faculté des sciences, nous avons recontrés deux problèmes que nous n'avons pas pu résoudre.

— La fonction PHP 'curl_exec()' qui permet de récuperer des données via des requêtes HTTP ne fonctionne pas. ce qui est génant car elle nous permet de récuperer les coordonnées GPS d'une adresse donnée. (la map ne fonctionne donc pas sur le serveur de la fac)

— Le serveur ne nous permet pas d'uploader des images. Il est donc impossible de mettre des images de profil et des images pour des événements.

6.2 Les compètences acquises

Pour conclure, à l'issue de ce projet nous avons réussi à réaliser un site web fonctionnel et prèt à l'utilisation.

Ce projet nous aura permis d'aprofondir nos connaissances en développement web et de compléter nos acquis sur les outils de base du web, tel que :HTML, CSS, PHP et JAVASCRIPT.