|  |
| --- |
| SAE |
| Party Share |
| Documentation |

|  |
| --- |
| Dylan Carluccio  14/11/2019 |

Table des matières

[Base de données 2](#_Toc25590075)

[Modèle conceptuel 2](#_Toc25590076)

[Description de la base de données 2](#_Toc25590077)

[Conception de la base de données 3](#_Toc25590078)

[Utilisateurs 3](#_Toc25590079)

[Evènements 3](#_Toc25590080)

[Participer 3](#_Toc25590081)

[Changemotdepasse 3](#_Toc25590082)

[ConfirmationEmail 3](#_Toc25590083)

[PHP 4](#_Toc25590084)

[Example de fonction PHP(PDO) 4](#_Toc25590085)

[Fonction pour se connecter à la base de données. 4](#_Toc25590086)

[Fonction pour supprimer un évènement. (Delete) 4](#_Toc25590087)

[Fonction qui afficher les événements que l’utilisateur a rejoint. (Select) 4](#_Toc25590088)

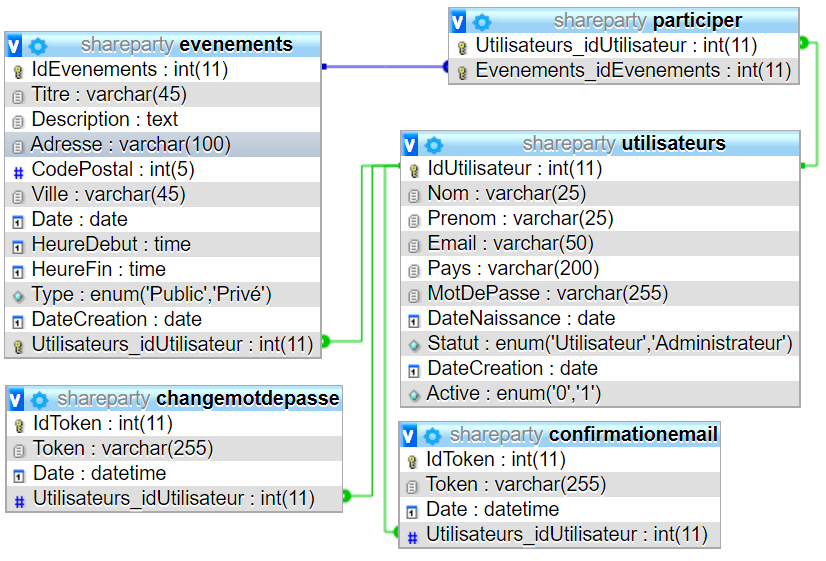
[Fonction pour rechercher dans la base de données. (Select) 4](#_Toc25590089)

[Fonction pour modifier un utilisateur. (Update) 5](#_Toc25590090)

[Fonction pour inscrire un utilisateur dans la base de données. (Insert) 5](#_Toc25590091)

# Base de données

## Modèle conceptuel



## Description de la base de données

Il y a une table « **utilisateurs** » qui contient les informations sur les utilisateurs. Qui est lié à « évènements ».

Il y a une table « **évènements** » qui contient les informations sur les événements.

Il y a une table « **participer** » qui lie les table « évènements » et « utilisateurs », elle contient l’id de l’évènements et l’id de l’utilisateur. Cette table est utilisée pour savoir si l’utilisateur participe à un évènement.

Il y a une table « **changemotdepasse** » qui contient les informations pour changer son mot de passe. Qui est lié à « utilisateurs ».

Il y a une table « **confirmationemail** » qui contient les informations pour confirmer son compte. Qui est lié à « utilisateurs ». Confirmer son compte sert à pouvoir publier des évènements.

## Conception de la base de données

### Utilisateurs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Champs** | **Type** | **Taille** | **PK** | **FK** | **AI** | **NULL** |
| IdUtilisateur | Int | 11 | X |  | X |  |
| Nom | Varchar | 25 |  |  |  |  |
| Prenom | Varchar | 25 |  |  |  |  |
| Email | Varchar | 50 |  |  |  |  |
| Pays | Varchar | 200 |  |  |  |  |
| MotDePasse | Varchar | 255 |  |  |  |  |
| DateNaissance | Date | / |  |  |  |  |
| Statut | Enum(‘utilisateur’,’Administrateur’) | / |  |  |  |  |
| DateCreation | Date | / |  |  |  |  |
| Active | Enum(‘0’,’1’) | / |  |  |  |  |

### Evènements

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Champs** | **Type** | **Taille** | **PK** | **FK** | **AI** | **NULL** |
| IdEvenements | Int | 11 | X |  | X |  |
| Titre | Varchar | 45 |  |  |  |  |
| Description | Text | / |  |  |  |  |
| Adresse | Varchar | 100 |  |  |  |  |
| CodePostal | Int | 5 |  |  |  |  |
| Ville | Varchar | 45 |  |  |  |  |
| Date | Date | / |  |  |  |  |
| HeureDebut | Time | / |  |  |  |  |
| HeureFin | Time | / |  |  |  |  |
| Type | Enum(‘Public’,’Privé’) | / |  |  |  |  |
| DateCreation | Date | / |  |  |  |  |
| Utilisateurs\_idUtilisateur | Int | 11 |  | X |  |  |

### Participer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Champs** | **Type** | **Taille** | **PK** | **FK** | **AI** | **NULL** |
| Utilisateurs\_idUtilisateur | Int | 11 | X | X |  |  |
| Evenements\_idEvenements | Int | 11 | X | X |  |  |

### Changemotdepasse

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Champs** | **Type** | **Taille** | **PK** | **FK** | **AI** | **NULL** |
| IdToken | Int | 11 | X |  | X |  |
| Token | Varchar | 255 |  |  |  |  |
| Date | Datetime | / |  |  |  |  |
| Utilisateurs\_idUtilisateur | Int | 11 |  | X |  |  |

### ConfirmationEmail

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Champs** | **Type** | **Taille** | **PK** | **FK** | **AI** | **NULL** |
| IdToken | Int | 11 | X |  | X |  |
| Token | Varchar | 255 |  |  |  |  |
| Date | Datetime | / |  |  |  |  |
| Utilisateurs\_idUtilisateur | Int | 11 |  | X |  |  |

# PHP

J’ai utilisé pour mon site du PHP (Version 7.3.9) et pour accéder à la base de données j’ai utilisé l’extension (PDO). J’ai utilisé le PDO pour une question de sécurité car on prépare les requêtes avant et aussi pour la facilité d’utilisation.

## Example de fonction PHP(PDO)

### Fonction pour se connecter à la base de données.

function dbConnect()  
{  
 global $dbc;  
 try {  
 $dbc = new PDO('mysql:host=' . *DB\_HOST* . ';dbname=' . *DB\_NAME*, *DB\_USER*, *DB\_PASSWORD*, array(PDO::*MYSQL\_ATTR\_INIT\_COMMAND* => "SET NAMES utf8"));  
 } catch (Exception $e) {  
 echo 'Erreur : ' . $e->getMessage() . '<br />';  
 echo 'N° : ' . $e->getCode();  
 die('Could not connect to MySQL');  
 }  
}

### Fonction pour supprimer un évènement. (Delete)

function SupprimerEvenement($idEvenements)  
{  
 global $dbc;  
 $req = $dbc->prepare('delete from evenements where idEvenements=:idEvenements');  
 $req->execute(array('idEvenements' => $idEvenements));  
 return $req->fetch(PDO::*FETCH\_ASSOC*);  
}

### Fonction qui afficher les événements que l’utilisateur a rejoint. (Select)

function AfficherEvenementsRejoint($idUtilisateur)  
{  
 global $dbc;  
 $req = $dbc->prepare("SELECT e.\* FROM utilisateurs u ,evenements e JOIN participer p WHERE p.Utilisateurs\_idUtilisateur=:idUtilisateur AND p.Utilisateurs\_idUtilisateur=u.IdUtilisateur AND u.IdUtilisateur!=e.Utilisateurs\_idUtilisateur AND e.IdEvenements=p.Evenements\_idEvenements;");  
 $req->execute(array('idUtilisateur' => $idUtilisateur));  
 return $req->fetchall(PDO::*FETCH\_ASSOC*);  
}

### Fonction pour rechercher dans la base de données. (Select)

function RechercherEvenements($recherche)  
{  
 global $dbc;  
 $req = $dbc->prepare("SELECT \* FROM evenements e WHERE e.Titre LIKE CONCAT('%', :recherche, '%') OR e.Description LIKE CONCAT('%', :recherche, '%')");  
 $req->execute(array(':recherche' => $recherche));  
 return $req->fetchAll(PDO::*FETCH\_ASSOC*);  
}

### Fonction pour modifier un utilisateur. (Update)

function ModifierUtilisateur($UtilisateurDonnees)  
{  
////modifie un utilisateur  
//--------------------------------------------------------------------------  
 global $dbc;  
 $req = $dbc->prepare('UPDATE utilisateurs set Nom = :Nom, Prenom = :Prenom, Pays = :Pays, DateNaissance = :DateNaissance WHERE IdUtilisateur = :IdUtilisateur;');  
  
 $req->execute($UtilisateurDonnees);  
  
}

### Fonction pour inscrire un utilisateur dans la base de données. (Insert)

function InscriptionUser($nom, $prenom, $email, $pays, $password, $dateNaissance, $statut, $dateCreation, $active)  
{  
//Inscription des utilisateurs  
//--------------------------------------------------------------------------  
 global $dbc;  
 $req = $dbc->prepare('INSERT INTO utilisateurs(Nom,Prenom,Email,Pays,MotDePasse,DateNaissance,Statut,DateCreation,Active) VALUES( :nom,:prenom,:email,:pays,:password,:dateNaissance,:statut,:dateCreation,:active)');  
 return $req->execute(array(  
 'nom' => $nom,  
 'prenom' => $prenom,  
 'email' => $email,  
 'pays' => $pays,  
 'password' => $password,  
 'dateNaissance' => $dateNaissance,  
 'statut' => $statut,  
 'dateCreation' => $dateCreation,  
 'active' => $active  
  
 ));  
}