

## Base de données de l'application de gestion de congés

Nom de la base de données : db\_app\_conges

**UTILISATEUR** (idUser, nomUser, mailUser, udp)

Clé primaire : idUser

**CONGES** (idConges, idUser, dateConges, heureDebut, heureFin, idType)

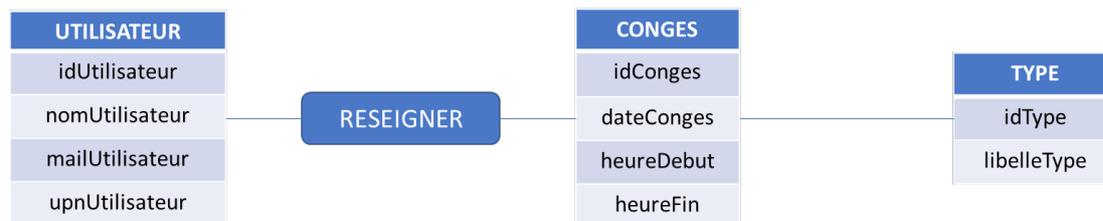
Clé primaire : idConges

Clé étrangère : idUser faisant référence à idUser d'UTILISATEUR

Clé étrangère : idType faisant référence à idType de TYPE

**TYPE** (idType, libelleType)

Clé primaire : idType



### Choix d'une solution pour l'authentification d'un utilisateur à l'application

- On pourrait créer un système d'authentification à l'aide de la base de données avec la table UTILISATEUR. On créera une page HTML avec un formulaire, puis on créera une page ASP.NET qui va comparer ce que l'utilisateur à rentrer dans le formulaire avec ce qui se trouve dans la table, et en fonction on autorisera l'accès ou non à l'application. Cependant, il faudra créer un formulaire pour que l'utilisateur puisse créer son mot de passe.
- Le plus simple serait d'utiliser le système d'authentification Azure AD, car la gestion de d'identifiant et de mot de passe (création, connexion, mot de passe oublié etc.) serait gérer par Azure AD et non le développeur. Il faudra juste adapter cette solution à l'application Recherche d'information. Il faut savoir maintenant si on doit utiliser un langage serveur ou client pour mettre cette solution en place.  
Solution choisie : authentification Azure, qui est possible coté client

### Choix d'une solution pour faire le lien entre la BDD et l'application de gestion de congés

Soit on trouve un langage client qui peut le gérer, soit on se retournera vers un langage serveur. Dans ce cas, on utilisera ASP.NET pour gérer les requêtes SQL.

Il n'y a pas l'air d'avoir de solution pour envoyer des requêtes SQL dans un langage client. On passera donc de JS à ASP.NET pour envoyer les requêtes SQL sur la base de données.

## Choix d'une solution de hachage pour le mot de passe

- MD5 : non, plus assez fiable
- SHA : Probable, un ou deux.

Si on choisit Azure AD en authentification, on ne fera pas ce choix (pas besoin de le gérer). Sinon, les mots de passes du formulaire et de la base de données seront hachés.

- 
- **Quelle solution devrais-je utiliser pour faire le lien entre la base de données et l'application ? (Un langage client ou serveur).**  
Un langage serveur, il me semble impossible de faire autrement (d'après mes recherches).
  - **Réfléchir à la conception de la base de données et la mettre en place.**  
Fait, pour l'instant il y a deux tables, une table UTILISATEUR et une table CONGES.
  - **Suite au passage de MS SharePoint à un base de données, je vais devoir plancher sur une autre méthode d'authentification. Je vais probablement faire une étude sur les solutions qui s'offre à moi.**  
On utilisera Azure AD. On peut le faire fonctionner en client. Cela va être beaucoup plus simple que de gérer l'authentification avec une base de données.

## Mis en place des solutions

1. **Comment faire pour intégrer Azure AD à l'application ?**
  2. **A-t-on vraiment besoin de la table UTILISATEUR, et si oui, comment l'utiliser ?**
  3. **Comment et quelle approche pour faire le lien entre la base de données et l'application : que va-t-on utiliser comme outils ?**
- 
1. Afin d'intégrer Azure AD à l'application, on utilisera des outils que Microsoft propose pour intégrer Azure AD à une application. Voici une façon d'y parvenir :  
On peut enregistrer son application sur le portail d'Azure. En l'enregistrant en tant que « application que ma société développe », on aura accès à des options pour configurer la connexion à Azure AD, afin que l'application puisse bénéficier de ce

mode d'authentification, et pour gérer les paramètres et droits de chaque utilisateur sur l'application. Si cela fonctionne réellement aussi simplement que ça, il faudra utiliser cette méthode à la fin, quand la partie « lien entre BDD et application » sera finie.

<https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-integrating-applications>

2. On utilisera la table UTILISATEUR pour identifier un utilisateur auprès de l'application : On utilisera l'UPN comme moyen de comparaison entre celui-ci et le mail de connexion : Si l'authentification est réussie, on affichera des données de l'utilisateur correspondant grâce aux relations entre la table UTILISATEUR et CONGES

**3. Comme dit précédemment, on utilisera un langage serveur pour faire le lien entre la base de données et l'application :**

- ASP.NET lien entre la BDD et l'application conges
- AJAX pour envoyer et récupérer les requêtes sans actualiser la page.

Avec tous ses outils, on devra faire :

- Mettre à jour la page app.js : Modifier la partie qui transmet les informations à SharePoint et faire en sorte de récupérer les informations que la page ASP.NET va échanger avec la BDD
- Créer une page de lien ASP.NET entre l'application et la base « app.conges »