

DOCUMENTATION TECHNIQUE BAREC AUTOMATISMES



Sommaire

<u>I. Analyse des données.....</u>	<u>2</u>
<u>II. Schéma Physique de la base de donnée.....</u>	<u>5</u>
<u>III. Alimentations de la table capteurs grâce a un script PHP et de fichiers XML.....</u>	<u>6</u>
<u>IV. La stratégie de mot de passe utilisé.....</u>	<u>7</u>

L'entreprise Barec Automatismes nous a demandé de créer une application web permettant l'accès à 3 différents outils :

Un outil permettant la gestion des PCB, un outil permettant la gestion MSL et un dernier outil permettant le suivi de la température et de l'humidité. Pour cela, nous avons dû commencer par créer une base de données regroupant toutes les informations nécessaires aux outils demandés de l'entreprise Barec telles que les références des différents PCB ou encore la liste des capteurs et encore.

I. Analyse des données

Voici la liste de toutes les données que nous devons insérer dans la base de données de l'entreprise Barec Automatismes :

niveau_msl		
Nom de la colonne	Description	Type MariaDB
niveau_msl	Identifiant du niveau MSL	varchar(2)
temps	le temps maximal où peut être exposé un des composants	int(11)
conditions	conditions pour stocker les composants	varchar(20)

reference_barec_pcb		
Nom de la colonne	Description	Type MariaDB
ref_barec	Identifiant du pcb	varchar(8)
nbre_couches	Nombre de couches	int(10)
type_rigidite	Type de la rigidite	varchar(20)

pcb_maintien		
Nom de la colonne	Description	Type MariaDB
id_lot_maintien	Identifiant du lot mis en maintien	int(11)
date_dernier_sortie	Date de dernière sortie d'asséchage ou nulle sinon	date
statut_armoire	Statut pour savoir s'il est dans l'armoire ou non	char(1)
seuil_alarme	le seuil d'alarme	int(11)

capteur		
Nom de la colonne	Description	Type MariaDB
id_capteur	permet d'identifier de manière unique le capteur	int(11)
date_enregist_donnee	date de l'enregistrement	date
heure_enregist_donnee	heure de l'enregistrement	time
zone_concernee	zone concernée par le relevé	char(5)
temperature_releve	température relevée	float
humidite_releve	humidité relevée	int(11)

type_zone		
-----------	--	--

Nom de la colonne	Description	Type MariaDB
code_type	permet d'identifier de manière unique le type de zone	char(3)
nom_zone	nom du type de zone	varchar(30)
temperature_min	température minimale	float
temperature_max	température maximale	float
humidite_min	humidité minimale	int(11)
humidite_max	humidité maximale	int(11)

type_assechage

Nom de la colonne	Description	Type MariaDB
code_type_assechage	Identifiant du type d'asséchage	varchar(4)
degre	Niveau de degré de ce type	int(11)
duree_max	Dure maximum de l'asséchage	int(11)

entrepot

Nom de la colonne	Description	Type MariaDB
id_entrepot	permet d'identifier de manière unique l'entrepot	int(11)
nom_entrepot	nom de l'entrepot	varchar(30)

utilisateur

Nom de la colonne	Description	Type MariaDB
id_utilisateur	identifiant de l'utilisateur	int(11)
mdp	mot de passe de l'utilisateur hacher avec sha512	varchar(128)
prenom	prenom de l'utilisateur	varchar(36)
nom	nom de l'utilisateur	varchar(30)
mail	mail de l'utilisateur	varchar(120)

alarme

Nom de la colonne	Description	Type MariaDB
id_alarme	permet d'identifier de manière unique l'alarme	int(11)
date_heure_declenche	date de déclenchement	datetime
duree_alarme	durée de déclenchement de l'alarme	time
zone_concernee	zone concernée par le déclenchement	char(5)
type_declencheur	l'alarme a été déclenché par la température ou l'humidité	char(1)
temperature_releve	température relevée	float
humidite_releve	humidité relevée	int(11)

lot_pcb_armoire

Nom de la colonne	Description	Type MariaDB
id_lot_pcb_armoire	identifiant du lot pcb	int(11)
ref_pcb	référence vers le pcb	varchar(8)
qte_composant	quantité de composants dans le lot	int(11)

id_armoire	identifiant vers l'armoire ou est stocké le lot	int(11)
niveau_msl	l'identifiant du niveau msl	varchar(2)
duree_utilisation_restante	La durée qu'il reste avant épuisement du lot	int(11)
date_ouverture	la date à la quelle il a été ouvert	date

zone

Nom de la colonne	Description	Type MariaDB
code_zone	identifiant de manière unique la zone	char(5)
type_zone	type de zone	char(3)
temperature_min	température minimum dans la zone	float
temperature_max	température maximum dans la zone	float
humidite_mini	humidité minimum dans la zone	int(11)
humidite_maxi	humidité maximum dans la zone	int(11)
duree_alarme_temperature	temps de déclenchement de l'alarme température	int(11)
duree_alarme_humidite	temps de déclenchement de l'alarme humidité	Int(11)

pcb_assechage

Nom de la colonne	Description	Type MariaDB
id_lot_assechage	Identifiant du lot mis en assechage	int(11)
date_heure_mise_etuve	date a la quel le lot est mis en étuve	datetime
duree_assechage_requis	dure minimal de l'assechage	int(11)
date_de_dispo	date/heure de disponibilité	date
type_assechage	Identifiant du type d'assechage qui va être fait	varchar(4)

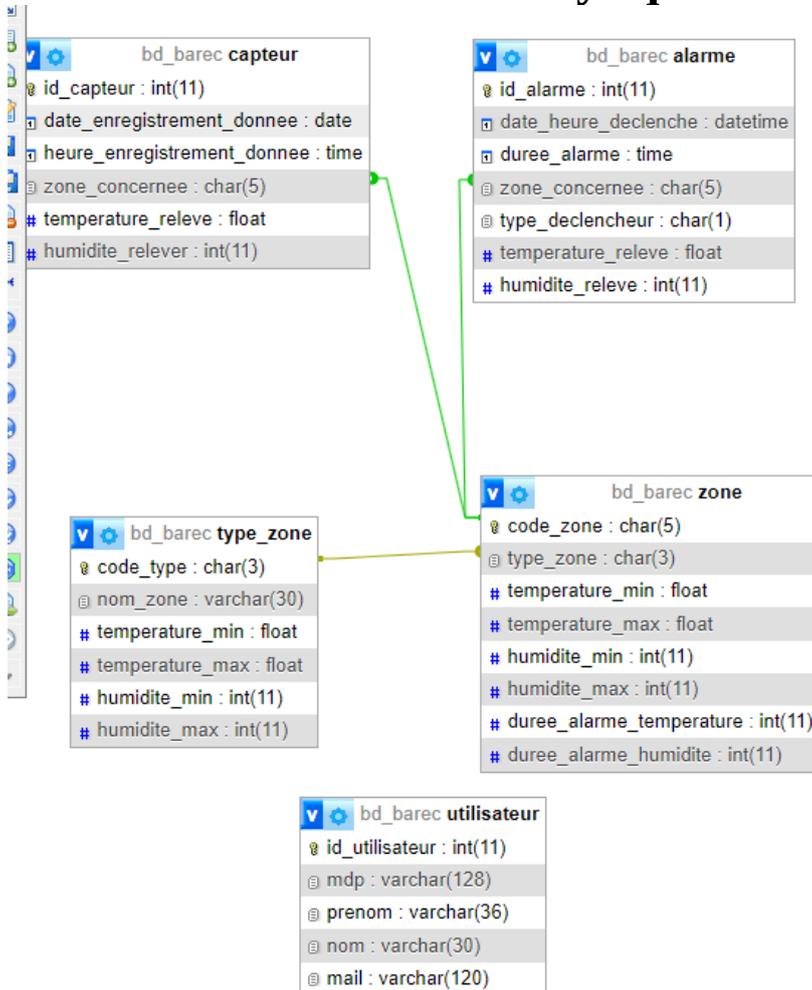
armoire

Nom de la colonne	Description	Type MariaDB
id_armoire	permet d'identifier de manière unique l'armoire	int(11)
nom_armoire	nom de l'armoire	varchar(30)

lot_pcb

Nom de la colonne	Description	Type MariaDB
ref_pbc	ref pbc	varchar(8)
id_lot_pcb	identifiant du lot pcb	int(11)
date_fin_validite	date de fin de validité	date
qte_pcb	quantité de composants	int(11)
etat	etat du lot	varchar(2)
localisation	localisation de l'entrepot	int(11)

II. Schéma Physique de la base de données



Les capteurs sont situés dans une zone.

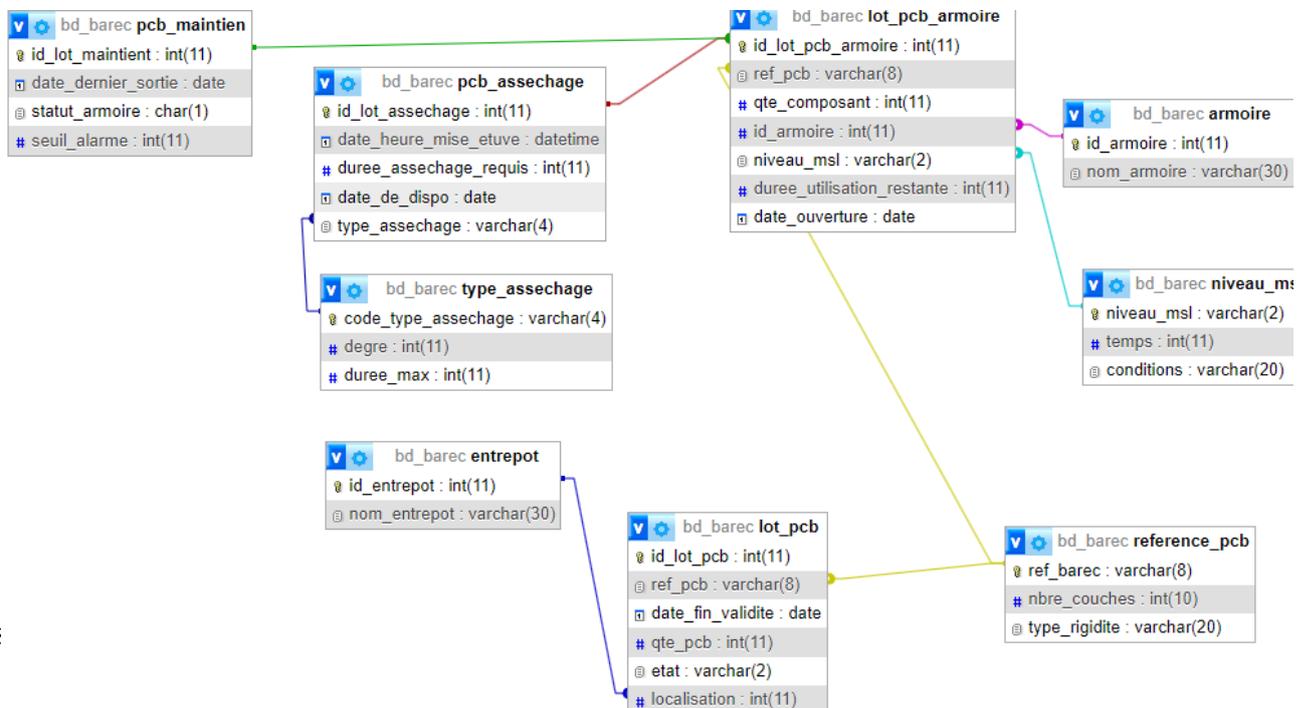
Une alarme se déclenche dans une zone.

Une zone correspond a un type.

Un lot PCB est entreposer dans un entrepôt et a une référence PCB.

Un lot PCB ouvert peut être stocker dans une armoire, si oui il a un niveau MSL., il peut aller en asséchage ou en maintien.

L'asséchage a différents types



III. Alimentation de la table capteur grâce à un script PHP et de fichiers XML

Voici le script PHP que nous avons créé afin de remplir la table capteur de notre base de données :

```
<?php
/*
Script permettant d'insérer les données des capteurs dans notre table.
Script réalisé par Christophe Naudet & Trystan Le Marec
*/
include('PdoConnexion.php');
include('PdoBarec.php');
//création d'une variable id qui représentera l'id de notre enregistrement
$id = 0;
//le nom du dossier que l'on veut parcourir
$nom_dossier = 'capteur/';
//ouverture du dossier a parcourir
$dossier = opendir($nom_dossier);
//Tant que le fichier peut encore etre lu
while($fichier = readdir($dossier))
{
//Si le fichier est matérialiser
if($fichier != '.' && $fichier != '..')
```

```
{
//on charge notre fichier XML
$docCapteur = simplexml_load_file('capteur/'.$fichier);
//on parcourt le noeud de notre fichier XML
foreach($docCapteur -> enregistrement as $unCapteur)
{
//on ajoute 1 a notre id pour qu'ils soient bien uniques
$id = $id + 1;

//on appelle la méthode insertBarec de la classe PdoBarec avec les bonnes données
$insertion = PdoBarec::insertBarec($id,$docCapteur["date"],$unCapteur -> heurereleve, $docCapteur["zone_releve"],
$unCapteur -> tempereleve, $unCapteur -> humiditereleve);
//si l'insertion réussit
if($insertion)
{
echo "l'insertion a réussi";
}
else //si elle ne réussit pas
{
echo "l'insertion a échouer";
}
}
}
//on ferme le dossier
closedir($dossier);
```

Nous avons décidé de choisir la méthode **opendir** comparée à la méthode **scandir** car elle permet de lire le contenu du dossier un élément à la fois, ce qui peut être plus efficace en termes de consommation de mémoire si le dossier contient un grand nombre de fichiers et offre un meilleur contrôle sur la manière dont vous traitez chaque élément du dossier, car vous pouvez exécuter des opérations sur chaque fichier/dossier individuellement pendant que vous parcourez le dossier.

IV. La stratégie de mot de passe utilisé

Afin de protéger vos mots de passe, nous allons vous proposer une stratégie de mots de passe principalement basé sur les recommandations de l'ANSSI :

- 14 caractère minimum
- 1 Majuscule minimum
- 1 Minuscule minimum
- 1 Caractère spécial minimum (Tous les caractères spéciaux sont autorisés)
- 1 Chiffre minimum

Exemple : BrY3^Sx-7t7)f4

De plus, afin de protéger encore plus vos mots de passes, nous allons mettre en place une fonction de hachage pour permettre de cacher les mots de passe si jamais un attaquant parvenez à s'introduire dans la base de données où les mots de passes sont stockés. La fonction de hachage que nous allons utiliser est SHA512.