

BASE DE DONNÉES

Documentation



Joseph Massé
Elizio Respringer
Léo Thomas

Barec Automatismes
Version 2

Table des matières

Table des matières.....	1
Contexte	2
Analyse des données	2
Création des tables de la base de données	4
Schéma de la base de données	5
Démarche d'alimentation de la table capteur	5
Scripts PHP nécessaires à l'alimentation de la table capteur	6
Politique pour les mots de passes	6

Contexte

Barec Automatismes est une entreprise créant des circuits imprimés. Ces circuits imprimés sont asséchés dans des conditions spécifiques nécessitant une certaine surveillance. Pour cela, Barec Automatismes nous a demandé de créer une application web afin de surveiller les capteurs et les durées de séchages des circuits imprimés ainsi que la gestion du stockage.

Analyse des données

Pour cela, nous avons analysé les données nécessaires pour créer la base de données relationnelle que nous avons mis tout d'abord sous forme de tableau comme suit.

<u>Table utilisateur</u>	<u>Type</u>	<u>Description</u>
id_utilisateur	int(11)	Clé primaire permettant d'identifier chaque utilisateur
mail	varchar(50)	Adresse de courriel électronique
mdp	varchar(128)	Contient les mots de passe hachés des utilisateurs avec le SHA-512
nom	varchar(25)	Contient le nom de la personne se connectant
prenom	varchar(25)	Contient le prénom de la personne se connectant

<u>Table lot_pcb</u>	<u>Type</u>	<u>Description</u>
id_lot_pcb	int(11)	Clé primaire ; numéro permettant d'identifier le lot de PCB
quantite	int(11)	Nombre de PCB dans le lot
etat	varchar(2)	A : actif ou NA : non actif
reference	varchar(8)	Renvoie à id_ref de la table Référence ; permet de connaître le type de PCB
entrepot	int(11)	Renvoie à id_armoire de la table Armoire ; Armoire où sont stockés les PCB
date_fin_validite	date	Date avant la fin de la validité du lot

<u>Table reference barec_pcb</u>	<u>Type</u>	<u>Description</u>
ref_barec	varchar(8)	Clé primaire ; Nom de la référence
nbre_couches	int(11)	Nombre de couches des PCB
type_rigidite	varchar(20)	Type de rigidité des PCB

<u>Table armoire</u>	<u>Type</u>	<u>Description</u>
id_armoire	int(11)	Clé primaire permettant d'identifier l'armoire
nom	varchar(20)	Nom de l'armoire

<u>Table</u> <u>lot_pcb_armoire</u>	<u>Type</u>	<u>Description</u>
id_lot_armoire	int(11)	clé primaire ; renvoie à id_lot de la table lot_pcb
ref_barec	int(11)	renvoie à ref_barec de la table reference_barec_pcb
quantite_composants	int(11)	quantite des composants
id_armoire	int(11)	clé étrangère(référant à la table armoire)
code_niveau	varchar(2)	clé étrangère (référant à la table msl)
duree_stockage	int(11)	Durée pendant laquelle le lot a été stocké
date_ouverture	date	Date d'ouverture de l'armoire contenant le lot

<u>Table entrepot</u>	<u>Type</u>	<u>Description</u>
id_entrepot	int(11)	Clé primaire identifiant l'entrepôt
nom	varchar(20)	Nom de l'entrepôt

<u>Table type_assechage</u>	<u>Type</u>	<u>Description</u>
code_type_assechage	char(4)	Clé primaire du type de l'assechage
degre	int(11)	Temperature de l'assechage en °C
duree	time	Durée en heures de l'assechage

<u>Table assechage</u>	<u>Type</u>	<u>Description</u>
id_lot_assechage	int(11)	Clé primaire permettant d'identifier l'assechage effectué
type_assechage	char(4)	Renvoie à id_type_assechage de la table type_assechage
date_heure_mise_en_etuve	datetime	Date et heure duquel l'assechage a commencé
date_heure_disponibilite	datetime	Date et heure duquel l'assechage est terminé
duree	int(11)	Durée en heures de(s) assechage(s) effectué(s) sur un lot

<u>Table lot_maintien</u>	<u>Type</u>	<u>Description</u>
id_lot_maintien	int(11)	clé primaire
date_derniere_sortie	date	date de la dernière sortie d'asséchage ou nulle si le lot n'est pas passé à l'asséchage
statut	char(1)	indique si un lot PCB est dans l'armoire "E" ou utilisé "S"
seuil_alarme	int(11)	seuil d'alarme en heures

<u>Table msl</u>	<u>Type</u>	<u>Description</u>
code_niveau	varchar(2)	Clé primaire permettant de connaître le niveau du MSL (Moisture Sensivity Level)
temps	int(11)	Condition des températures pour le stockage
conditions	varchar(20)	Condition d'humidité pour le stockage

<u>Table zone</u>	<u>Type</u>	<u>Description</u>
code_zone	char(5)	Clé primaire
type_zone	char(3)	type de zone (clé étrangère référant à la table type_zone)
mini_temperature	float	temperature minimum
max_temperature	float	temperature maximum
mini_taux_humidite	int(11)	taux d'humidite minimum
max_taux_humidite	int(11)	taux d'humidite maximum
temps_declenchement_alarme_temperature	int(11)	temps en minutes du déclenchement de l'alarme pour la température
temps_declenchement_alarme_humidite	int(11)	temps en minutes du déclenchement de l'alarme pour l'humidité

<u>Table type_zone</u>	<u>Type</u>	<u>Description</u>
code_type	char(3)	clé primaire
nom_zone	varchar(30)	nom de la zone
temperature_min	float	temperature minimum de la zone
temperature_max	float	temperature maximum de la zone
humidite_min	int(11)	humidité minimum
humidite_max	int(11)	humidité maximum

<u>Table capteur</u>	<u>Type</u>	<u>Description</u>
id_capteur	int(11)	clé primaire
date	date	date de l'enregistrement des données
heure	time	heure de l'enregistrement des données
id_zone	char(5)	zone concernée (clé étrangère de la table zone)
temperature	float(5)	temperature relevée
humidite	int(11)	humidite relevée

<u>Table alarme</u>	<u>Type</u>	<u>Description</u>
id_alarme	int(11)	clé primaire
date_heure	datetime	date et heure de déclenchement
duree	int(11)	durée de l'alarme
id_zone	char(5)	zone concernée (clé étrangère de la table zone)
type	char(1)	type (température ou humidité)
temperature	float(5)	temperature relevée
humidite_relevée	int(11)	humidité relevée

Création des tables de la base de données

Ces scripts vont permettre de créer la structure de la base de données, c'est-à-dire la création des tables, l'ajout de clés primaires et l'ajout des clés étrangères.

Pour correspondre au cahier des charges, nous avons créé des relations entre les bases de données. Les relations entre les tables references_barec_pcb avec lot_pcb_armoire et lot_pcb va permettre d'avoir une liste précise des références dans une table permettant une meilleure flexibilité mais les lots doivent avoir une référence, d'où la création de cette relation.

Le code_zone de la table zone permet de connaître le lieu où sont situés les capteurs et les alarmes. Les relations entre lot_pcb et entrepot permet de connaître le lieu dans lesquelles les circuits imprimés sont stockés avant d'être livrés.

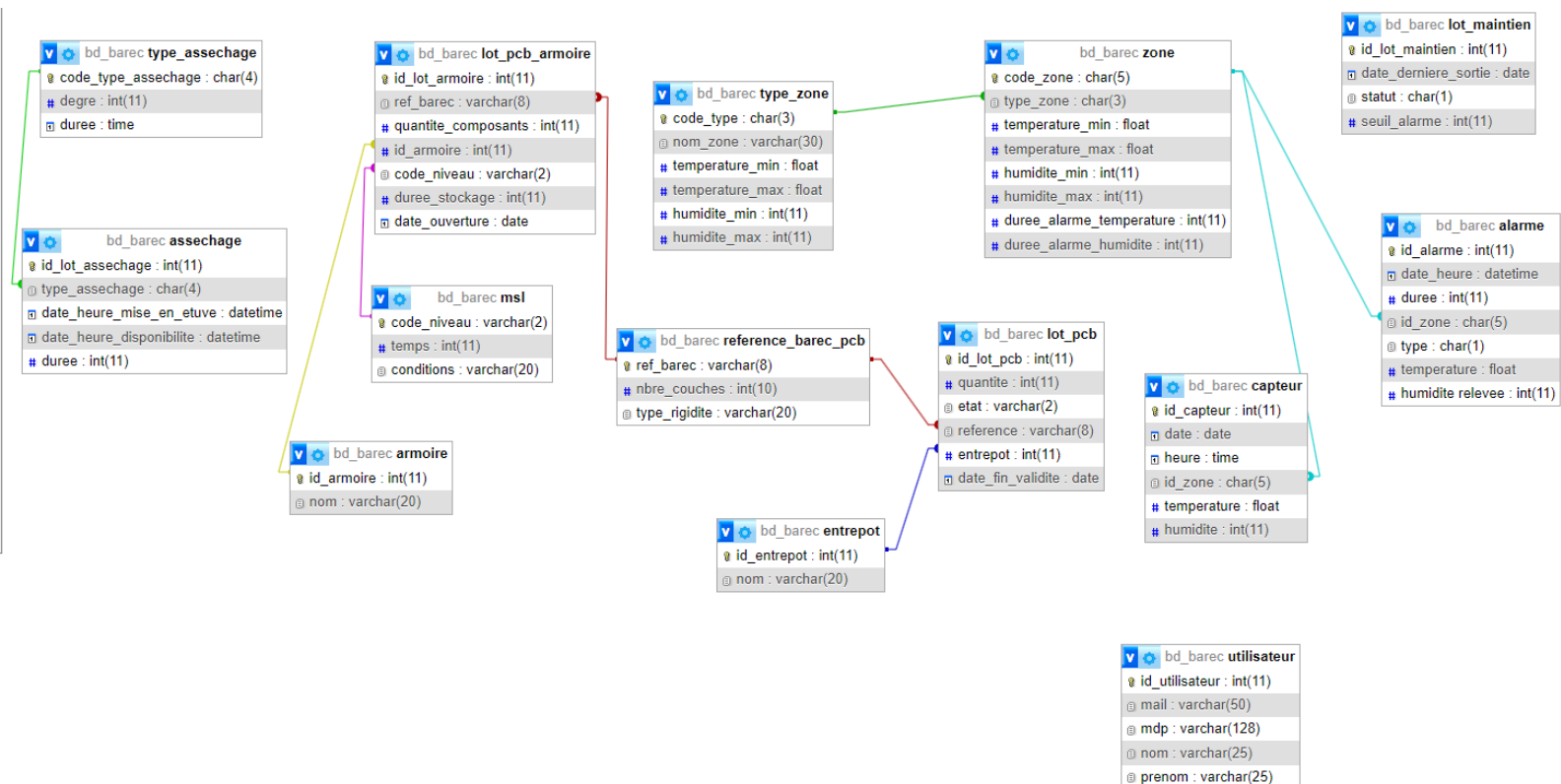
Pour ce qui en est d'entre lot_pcb_armoire et armoire permet de connaître où sont stockés les lots lorsqu'ils sont en armoire pour le stockage lorsque c'est l'étape de l'asséchage.

La relation entre zone et type zone permet de connaître les caractéristiques de chaque zone tout en pouvant avoir des zones avec les mêmes caractéristiques.

Dans le cas entre les tables lot_pcb_armoire et msl permet d'avoir des précisions sur le stockage en armoire sur les conditions d'humidité qui va être un paramètre clé sur le produit final.

Schéma de la base de données

Une fois cette analyse terminée, nous avons créé la base de données et avons conçu un schéma pour avoir une vision plus précise sur les clés primaires et les contraintes de clés étrangères.



Démarche d'alimentation de la table capteur

Afin de remplir la table capteur avec les fichiers XML, il est nécessaire d'avoir plusieurs programmes permettant :

- De se connecter à la base de données
- L'insertion des données dans la base de données
- De lire le dossier avec les fichiers XML ainsi que la lecture de ces documents.

Pour la lecture du dossier, nous avons utilisé la fonction opendir qui permet d'ouvrir le dossier contenant les fichiers XML puis nous lisons chaque fichier à l'aide de la fonction readdir et d'une boucle foreach pour accéder aux données à insérer via une fonction d'insertion préalablement créée.

Scripts PHP nécessaires à l'alimentation de la table capteur

Script de connexion (PdoConnexionCapteur.php) :

Ce script permet d'établir la connexion entre le programme d'insertion des données et la base de données MariaDB.

Script d'insertion (PdoInsertionCapteur.php) :

Ce script permet d'insérer les données contenues dans les fichiers XML dans la table capteur de la base de données.

Script de lecture des fichiers (InsertionDonnees.php) :

Ce script permet de lire les données contenues dans les fichiers XML pour les insérer dans la table capteur de la base de données grâce aux fonctions PdoInsertionCapteur et PdoConnexionCapteur.

Politique pour les mots de passes

Afin d'assurer la sécurité et la confidentialité des mots de passe, nous les stockons hachés avec la fonction de hachage SHA-512. De plus, nous recommandons de créer des mots de passe avec des majuscules, minuscules, caractères numériques et spéciaux d'au moins 12 caractères de long. Il est conseillé que le mot de passe ne soit pas des mots du dictionnaire.