

avant de commencer...

## But de la séance :

Ce TP n'est pas un projet, mais y ressemble beaucoup. Le modèle des données est imposé, ainsi qu'un contenu initial pour la base de données. Les étapes décrites ici correspondent au démarrage : installation, création de la base, peuplement des tables, versioning initial, test d'une première fonctionnalité.

## Pré-requis :

- Disposer d'une configuration LAMP fonctionnelle et paramétrée pour CakePHP
- Connaître les paramètres du SGBD
- Composer installé
- Git installé

## Références :

- CakePHP Cookbook Documentation v3.8  
[https://book.cakephp.org/3/\\_downloads/fr/CakePHPCookbook.pdf](https://book.cakephp.org/3/_downloads/fr/CakePHPCookbook.pdf)

## Sommaire

1	Présentation du projet	1
2	Modèle des données	3
3	Création du projet	3
4	Connexion à la base de données	4
5	Génération du MPD	4
6	Insertion de données dans les tables	4
7	Premier essai : affichage de la liste des formateurs	5
8	Poursuite du développement	5

## 1 Présentation du projet

On veut réaliser un outil de suivi des charges d'enseignement pour un établissement. Le but de cet outil est de savoir qui fait quoi, par rapport à un référentiel donné, et de pouvoir lister une fiche de charges (liste des cours réalisés par le formateur), comportant éventuellement des enseignements faits en dehors du référentiel (pour le compte d'autres départements).

Sur l'année 2019-2020 par exemple, l'établissement *UFR Sciences et Techniques*, à travers son département *Département d'Informatique*, propose les formations suivantes :

- *Licence Sciences, Technologie, Santé*, mention *Informatique*, années 1, 2, et 3, sur les sites de *Tours* et de *Blois* ;
- *Master Sciences, Technologie, Santé*, mention *Informatique*, parcours *Big Data Management and Analytics*, années 1, 2, site de *Blois* ;
- *Master Sciences, Technologie, Santé*, mention (*toutes*), parcours *Compétences complémentaires en Informatique*, année 2, site de *Tours*

Un établissement compte plusieurs départements. Chacun propose plusieurs formations. Une formation peut avoir plusieurs mentions. Une mention peut avoir plusieurs parcours. La formation peut être faite sur différents sites (l'unité pour le site est l'année de formation). Pour simplifier, on considèrera qu'une mention compte au minimum un parcours (appelé par défaut *voie générale*). Il est possible également, au même niveau, de distinguer des dispositifs ayant un impact sur la liste des cours suivis. Par exemple, en concurrence de *voie générale*, on pourrait trouver *aide à la réussite* par exemple.

Un parcours consiste en une liste d'enseignements à suivre. Un enseignement peut être commun à plusieurs parcours. Par ex. l'enseignement *Bases de données & Web* existe dans les parcours *voie générale*, et *aide à la réussite* des mentions *Informatique* et *Mathématiques* de la formation *Licence Sciences, Technologie, Santé*. Les enseignements sont regroupés en modules, au sein de périodes (habituellement des semestres). Ainsi, la mention *Informatique*, parcours *voie générale*, compte 6 périodes : *semestre 1* à *semestre 6*. Au cours du semestre 5, il y a 4 modules : *Structuration et analyse de données*, *Programmation Web et Théorie Informatique*, *Génie Logiciel*, et *Compétences transversales*. Chaque module a aussi un numéro, relatif au semestre (il y a donc plusieurs modules 1 par exemple). Pour prolonger l'exemple, dans le module *Génie Logiciel* il y a 2 enseignements : *Conception ergonomique d'interfaces* et *Génie logiciel et gestion de projet*. Chaque enseignement a un numéro, relatif au module et possède un code, unique. Les modules sont habituellement nommés *M1*, *M2* ... et les enseignements *EP1*, *EP2* ...

Chaque enseignement consiste en un certain nombre d'heures de cours (CM), d'heures de travaux dirigés (TD), et d'heures de travaux pratiques (TP). L'ensemble de ces données est fixe sur une certaine période de temps, qui dure en général 4 ans (parfois c'est moins, parfois c'est plus).



On doit pouvoir créer, modifier, supprimer toutes ces données, c'est à dire définir et mettre à jour toute une structure de formation. On peut extraire de cette base de données, par exemple, pour une formation donnée, la liste de tous ses enseignements, pour une année donnée, ordonnés par période, module et numéro d'enseignement, avec les volumes horaires détaillés, et les totaux au niveau module, période, formation.

L'établissement forme des étudiants. C'est à dire qu'elle ouvre des sessions de formation, conformes au catalogue défini ci-dessus. Il y a donc, par enseignement, des sessions de formation. Chacune a une date de début une date de fin, un nombre de groupe d'étudiants suivant les heures CM, un nombre de groupe d'étudiants suivant les heures TD, et un nombre de groupe d'étudiants suivant les heures TP. Ces valeurs sont applicables, par défaut, à tous les enseignements formant un parcours de formation. Attention, il peut y avoir des options certaines années et pas d'autres (selon le nombre de participants), donc, en réalité, c'est chaque session qui doit posséder ces paramètres.



On doit pouvoir créer, modifier, supprimer ces données. Il est alors possible, pour un parcours donné, d'avoir la liste des sessions ouvertes, avec, par période, module, enseignement, catégorie (CM, TD, TP), le nombre d'heures total à effectuer. On peut cumuler ces totaux au niveau du module, de la période, du parcours ... On peut aussi agréger ces totaux de deux manières :  $HE = CM + TD + TP$ , et  $HETD = 1,5 * CM + TD + \frac{2}{3} TP$ . On peut donc avoir en HE et en HETD le nombre d'heures total d'un enseignement, de tous les enseignements d'un module, de tous les modules d'une période, de toutes les périodes d'un parcours, de tous les parcours d'une mention, de toutes les mentions d'une formation, de toutes les formations d'un établissement, de tous les établissements.

L'établissement fait appel à des formateurs. Pour chacun on connaît son nom, prénom, email, statut.



On doit pouvoir créer, modifier, supprimer ces données. La machine peut nous éditer une liste des formateurs.

Chaque formateur est associé, pour un certain nombre d'heures CM, TD, et TP, à des sessions. Plusieurs formateurs peuvent collaborer sur la même session, avec la contrainte que la somme des heures qu'il font, par catégorie, ne soit pas supérieure aux nombres d'heures exigés par la session. Chaque session de formation a un formateur responsable. Un responsable peut avoir plusieurs sessions en responsabilité.



On doit pouvoir créer, modifier, supprimer ces données. On peut notamment éditer :

- une liste des enseignements pour lesquels toutes les heures ne sont pas effectuées ;
- un récapitulatif des enseignements, par parcours/module, pour une période de formation donnée, précisant le détail des heures à faire et les formateurs affectés avec les heures faites par chacun ;
- par formateur, le détail heures effectuées dans chaque enseignement auquel il participe, pour une période donnée

**R** Dans ce sujet, on parle souvent d'édition. Cela peut signifier "afficher à l'écran", mais aussi imprimer, voir exporter au format CSV.

## 2 Modèle des données

Le MCD est présenté en plusieurs parties, et en essayant de respecter les conventions de CakePHP. Les schémas ont été faits sous JMerise. La figure 1 montre la partie référentiel : institution, départements, sites, formations, mentions (*domains*) et parcours (*specialties*).

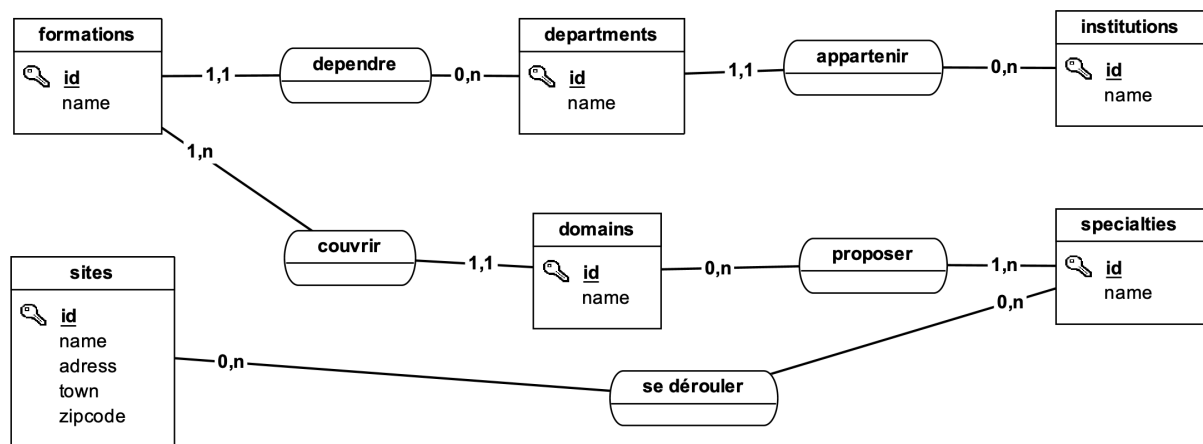


FIGURE 1 – Modèle des données du référentiel

La figure 2 montre la partie maquette (modules, enseignements, heures). L'entité *specialties* est le point de raccordement avec le schéma précédent. Dans l'entité *blocks* qui représente les modules, le champ *code* correspond au code module dans la période ; dans l'entité *courses* qui représente les enseignements, le champ *code* correspond au code établissement (appelé *code Apogée* pour une université).

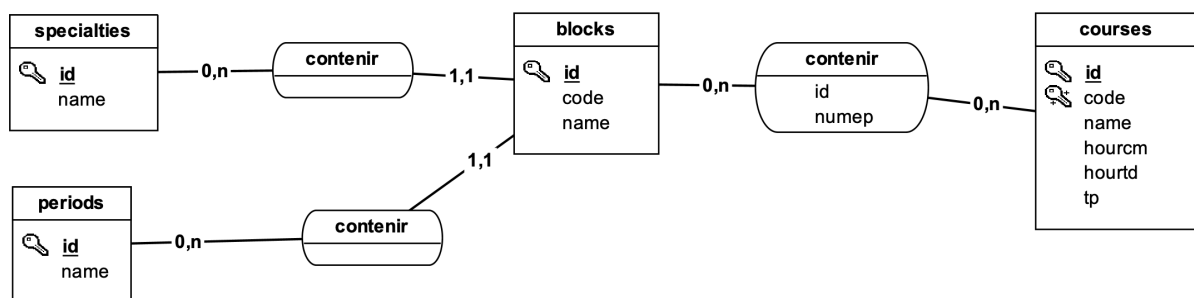


FIGURE 2 – Modèle des données des maquettes

La figure 3 montre la partie réalisation des enseignements. L'entité *courses* est le point de raccordement avec le schéma précédent.

## 3 Création du projet

Définir, à la racine de la zone servie par votre serveur web, un dossier *gestcharge*. Créer ensuite, *via* Composer, le projet<sup>1</sup>. Tester l'application (cf. doc en ligne sur Celene) pour vérifier qu'au moins elle se lance, et corriger les problèmes en conséquence (cf. le manuel en ligne).

1. Il est aussi possible de récupérer une archive à décompresser sur Github : <https://github.com/cakephp/cakephp/releases/tag/3.8.6>

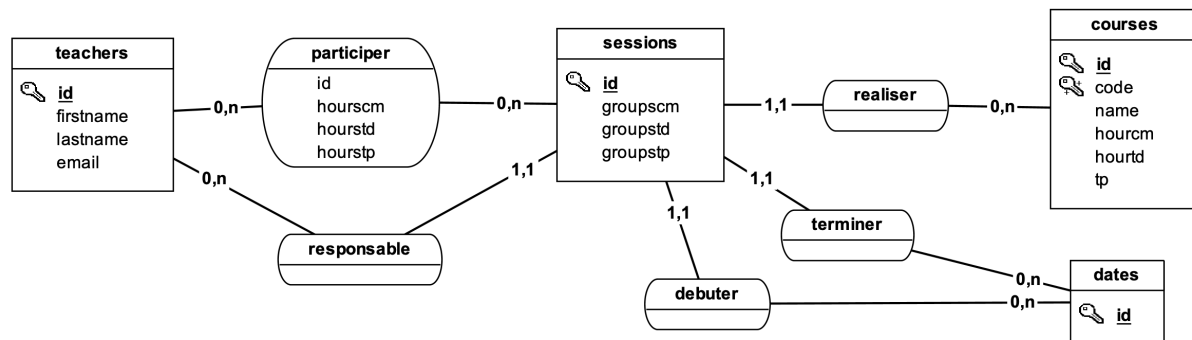


FIGURE 3 – Modèle des données des sessions de cours

## 4 Connexion à la base de données

Créez la base de données *gestcharge* : connectez vous en ligne de commande sur votre serveur (`mysql -u root -p` en général), créez la base de données (`create database gestcharge;`), et connectez-y l'application CakePHP (là aussi, voir le manuel en ligne<sup>2</sup>) : dans le répertoire *gestcharge/config* dupliquez le `app_local.example.php`, renommez-le en `app_local.php`, éditez-le pour mettre à jour les infos concernant la base de données et enregistrez. Testez avec le navigateur (`http://localhost:8765/`).

Si cela fonctionne, créez, depuis `mysql`, un utilisateur *gestcharge*<sup>3</sup> qui aura les droits sur la base<sup>4</sup>, et mettez à jour l'application avec cet utilisateur (le *root* est réservé à l'administrateur!!).

Aller ensuite, au moyen d'un shell, dans ce dossier et lancer *git* afin de faire une image de la version initiale<sup>5</sup>. Vous penserez à ne pas tout archiver (inutile d'avoir le répertoire `/tmp`, ni le répertoire `/logs`, ni le fichier `app_local.php` qui contient les paramètres de la base de données de développement, ne le fichier `app.php` qui contient les paramètres des serveurs d'exploitation. Vérifiez avec `git log` que cela s'est bien passé. Créez une branche *dev* et placez vous dedans.

## 5 Génération du MPD

Il est facile d'obtenir un MPD depuis l'outil de modélisation. On reprend ensuite le script généré de façon à imposer les conventions de CakePHP. Cela concerne principalement les noms des tables N-N et les noms des clés étrangère, la vérification de la présence d'un identifiant si la relation était porteuse (cela a déjà été fait sur le MCD), et une relecture globale pour voir si on a rien oublié. Les scripts sont téléchargeables en ligne. Insérez le schéma<sup>6</sup> dans la base et vérifiez que cela est correct<sup>7</sup>.

## 6 Insertion de données dans les tables

Le *peuplement* initial des tables se fait souvent au moyen de fichiers à plat, générés par un tableur, sous la forme de fichiers CSV. Il est assez facile de mettre des données dans les tables terminales, et de recopier, dans les autres, les clés primaires de ces tables. Le tableur offre aussi l'avantage de pouvoir calculer et générer aléatoirement, des données. Il reste ensuite à insérer les données dans la base. Cela peut se faire au moyen d'une interface graphique, ou en ligne de commandes<sup>8</sup> par `load data local infile 'nom_complet_fichier.csv' into table gestcharge.nom_table character set utf8mb4 fields terminated by ',';` (environ). Juste vérifier que le délimiteur est bien le point-virgule, et que le nom du fichier est bien absolu. On a autant de fichiers CSV que de tables. On scripte tout ça en SQL, chargé ensuite avec un ordre *source* dans la console `mysql`. Vous vérifierez notamment l'encodage des caractères, car sinon par la suite ça risque d'être un peu délicat. Les fichiers fournis ont l'extention `.txt`, mais le bon format.

2. <https://book.cakephp.org/3/en/quickstart.html#cms-tutorial-creating-the-database>

3. par exemple : `CREATE USER 'gestcharge'@'localhost' IDENTIFIED BY 'abcdef';`

4. `GRANT ALL PRIVILEGES ON gestcharge.* TO 'gestcharge'@'localhost'; FLUSH PRIVILEGES;`

5. <https://git-scm.com/book/fr/v2>

6. En ligne de commande `mysql: source chemin_absolu/nomfichier.sql`

7. `show tables;` et des trucs du genre `describe blocks;` par exemple

8. <https://dev.mysql.com/doc/refman/5.7/en/load-data.html>

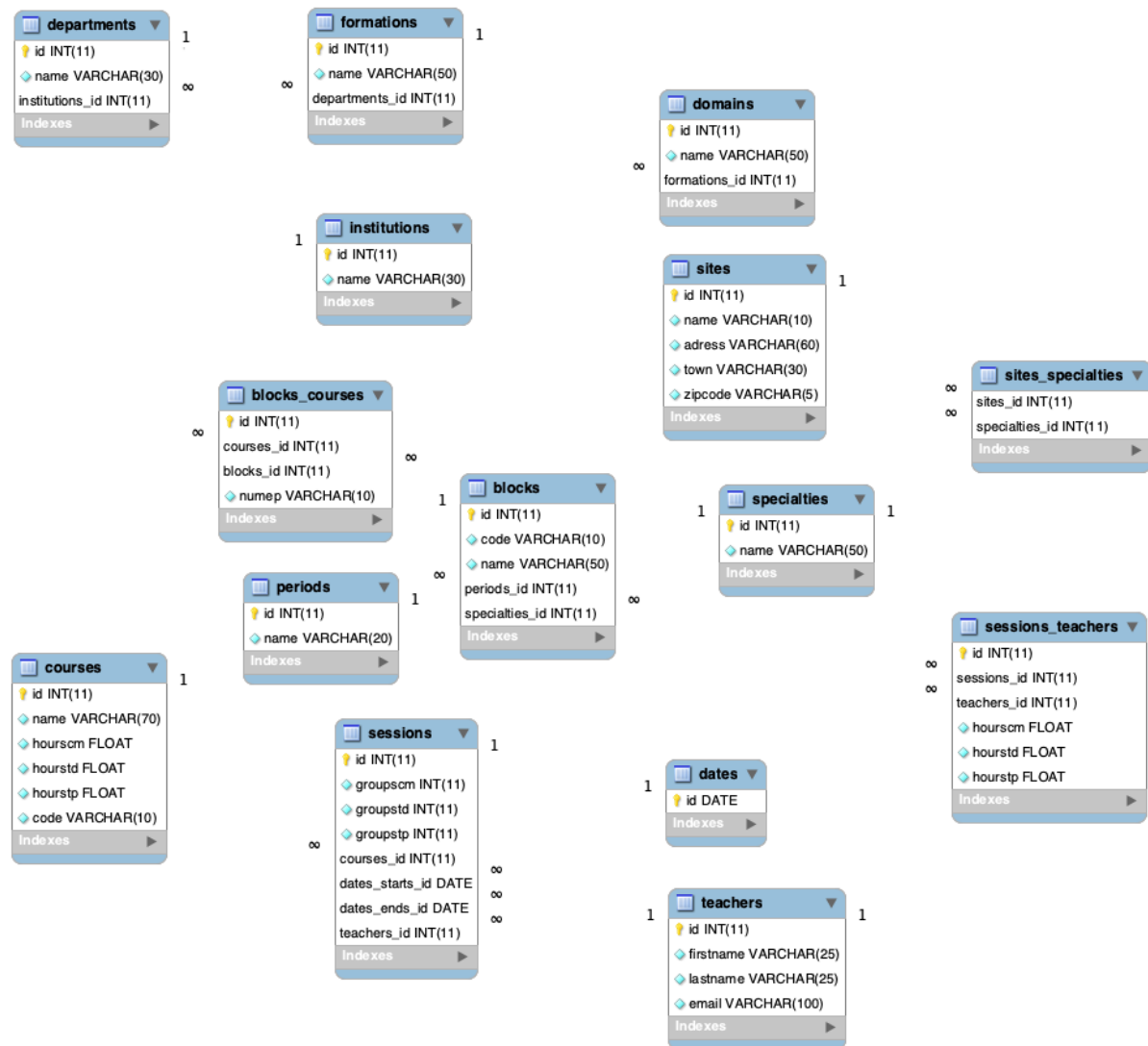


FIGURE 4 – Modèle logique des données (MySQL Workbench)

## 7 Premier essai : affichage de la liste des formateurs

Arrivés à ce point, vous avez une application CakePHP de base, connectée à une base de données, laquelle a un schéma et des données. On va donc vérifier si tout cela est correct, et si l'ORM est si pratique que cela.

En reprenant le schéma des données, et en consultant le manuel de CakePHP en ligne<sup>9</sup>, créez les éléments nécessaires (modèle, vue et contrôleur) de façon à faire afficher la table des formateurs. Vous aurez donc à créer les fichiers :

- `src/Controller/TeachersController.php`, pour lancer l'action associée à l'url ;
- `src/Templates/Teachers/index.ctp`, pour afficher les données

L'url `http://localhost/gestcharge/teachers/` devrait afficher la liste des formateurs, telle que visible figure 5. Quand cela est au point, faites une version.

## 8 Poursuite du développement

Poursuivez le tutoriel en ligne. Les fonctionnalités à implémenter sont :

- ajouter un formateur ;
- supprimer un formateur ;
- modifier les données d'un formateur.

9. <https://book.cakephp.org/3/fr/quickstart.html#creation-du-premier-model>

Teachers		
Liste des formateurs		
Prénom	Nom	E-mail
Thierry	Brouard	brouard@univ-tours.fr
Aurore	Cabet	cabet@univ-tours.fr
Nizar	Messai	messai@univ-tours.fr
Yacine	Sam	sam@univ-tours.fr
Michel	Teguaia	teguaia@univ-tours.fr

FIGURE 5 – Premier affichage de la vue d'ensemble de la table *Teachers*

Dès qu'une fonctionnalité est en place, faites une version. Lorsque toutes les fonctionnalités concernant la gestion de base d'un formateur seront en place, fusionner la branche *dev* et la branche *master* de façon à créer une version stable officielle. Retournez sur la branche *dev* afin de poursuivre le développement. En ajoutant quelques icônes, on peut, en moins d'une heure, arriver à une version complète (sans gestion d'erreurs pour le moment) mais avec validation de la saisie des formulaires telle que le montre la figure 6.

Teachers

Liste des formateurs +



















Prénom	Nom	E-mail	Actions
Thierry	Brouard	thierry.brouard@univ-tours.fr	  
Aurore	Cabet	cabet@univ-tours.fr	  
Nizar	Messai	nizar.messai@univ-tours.fr	  
Yacine	Sam	sam@univ-tours.fr	  
Michel	Teguaia	teguaia@univ-tours.fr	  
Maxime	Bourreau	bourreau@univ-tours.fr	  

FIGURE 6 – En moins d'une heure on a un premier jet tout à fait fonctionnel.