**MCD**

**Container**

**- Id\_container**

- Nom

- Désignation

1-N

**Contenir**

0-N

**Page WEB**

**- Id\_web**

- Adresse URL

N-N

0-N

0-N

**AssocierContenu**

**AssocierTarif**

N-N

0-N

N-N

0-N

**Contenu**

**- Id\_contenu**

- Description

- Adresse\_image

- Contenu\_Texte

**Tarif**

**- Id\_tarif**

- Désignation

- Prix

0-N

0-N

**ModifierTarif**

Date

**ModifierContenu**

Date

N-N

N-N

0-N

0-N

**UserAdmin**

**- Id\_user**

- User

- Password

0-1

**ModifierProfil**

1-1

**Profil**

**- Id\_profil**

- Nom

- Prenom

- Adresse

- Adresse Mail

**Message**

**- Id\_message**

- Nom

- Prenom

- Adresse

- Adresse Mail

- Telephone

- Texte

- Statut

**Traiter** *Contrainte**(Lire plusieurs fois,* supprimer une fois)

Date

0-N

0-N

1-1

1-1

**Action**

**- Id\_action**

- Type (Lire, supprimer)

N-N

**MLD**

PageWeb(id\_web PK, adresse\_url);

Container(id\_container PK, nom, designation, id\_web FK);

Contenir(id\_web PK FK, id\_container PK FK);

Contenu(id\_contenu PK, description, adresse\_image, contenu\_texte);

AssocierContenu(id\_container PK FK, id\_contenu PK FK);

Tarif(id\_tarif PK, designation, prix);

AssocierTarif(id\_container PK FK, id\_tarif PK FK);

ModifierContenu(id\_user PK FK, id\_contenu PK FK, date\_modif PK);

ModifierTarif(id\_user PK FK, id\_tarif PK FK, date\_modif PK);

UserAdmin(id\_user PK, user, password);

Profil(id\_profil PK, nom, prenom, adresse, adresse\_mail);

ModifierProfil(id\_user PK FK, id\_profil UNIQUE FK);

Message(id\_message PK, nom, prenom, adresse, adresse\_mail, telephone, texte, statut);

Action(id\_action PK, type\_action);

Traiter(id\_user PK FK, id\_message PK FK, id\_action PK FK, date\_traitement PK);

**Script SQL version Chat GPT**

-- 1. Tables sans dépendances

CREATE TABLE Page\_web (

Id\_Page\_web INT PRIMARY KEY AUTO\_INCREMENT,

Adresse\_url\_Page\_web VARCHAR(255) NOT NULL

);

CREATE TABLE Container (

Id\_Container INT PRIMARY KEY AUTO\_INCREMENT,

Nom\_Container VARCHAR(100) NOT NULL,

Designation\_Container VARCHAR(255)

);

CREATE TABLE Contenu (

Id\_Contenu INT PRIMARY KEY AUTO\_INCREMENT,

Description\_Contenu TEXT,

Adresse\_image\_Contenu VARCHAR(255),

Contenu\_texte\_Contenu TEXT

);

CREATE TABLE Tarif (

id\_Tarif INT PRIMARY KEY AUTO\_INCREMENT,

Designation\_Tarif VARCHAR(255) NOT NULL,

Prix\_Tarif VARCHAR(50) NOT NULL

);

CREATE TABLE Message (

Id\_Message INT PRIMARY KEY AUTO\_INCREMENT,

Nom\_Message VARCHAR(100) NOT NULL,

Prenom\_Message VARCHAR(100) NOT NULL,

Adresse\_Message TEXT NOT NULL,

Adresse\_mail\_Message VARCHAR(150) NOT NULL,

Telephone\_Message VARCHAR(50) NOT NULL,

Texte\_Message TEXT NOT NULL,

Statut\_Message VARCHAR(50) NOT NULL

);

CREATE TABLE Action (

Id INT PRIMARY KEY AUTO\_INCREMENT,

Type\_Action VARCHAR(50) NOT NULL

);

-- 2. Table Profil (sans FK circulaire)

CREATE TABLE Profil (

Id\_Profil INT PRIMARY KEY AUTO\_INCREMENT,

Nom\_Profil VARCHAR(100) NOT NULL,

Prenom\_Profil VARCHAR(100) NOT NULL,

Adresse\_Profil TEXT NOT NULL,

Adresse\_mail\_Profil VARCHAR(150) NOT NULL

);

-- 3. User\_admin (après Profil)

CREATE TABLE User\_admin (

Id\_User\_admin INT PRIMARY KEY AUTO\_INCREMENT,

User\_User\_admin VARCHAR(100) NOT NULL,

Password\_User\_admin VARCHAR(255) NOT NULL,

profil\_id\_profil INT,

FOREIGN KEY (profil\_id\_profil) REFERENCES Profil(Id\_Profil)

);

-- 4. Ajouter la FK vers User\_admin (facultatif mais possible ici via ALTER si tu veux circulaire)

ALTER TABLE Profil

ADD CONSTRAINT fk\_profil\_user\_admin

FOREIGN KEY (Id\_Profil) REFERENCES User\_admin(profil\_id\_profil);

-- 5. Tables associatives

CREATE TABLE Contenir (

IdPageweb INT NOT NULL,

IdContainer INT NOT NULL,

PRIMARY KEY (Id\_Page\_web, Id\_Container),

FOREIGN KEY (Id\_Page\_web) REFERENCES Page\_web(Id\_Page\_web),

FOREIGN KEY (Id\_Container) REFERENCES Container(Id\_Container)

);

CREATE TABLE Associer\_Contenu (

IdContainer INT NOT NULL,

IdContenu INT NOT NULL,

PRIMARY KEY (Id\_Container, Id\_Contenu),

FOREIGN KEY (Id\_Container) REFERENCES Container(Id\_Container),

FOREIGN KEY (Id\_Contenu) REFERENCES Contenu(Id\_Contenu)

);

CREATE TABLE Associer\_Tarif (

IdContainer INT NOT NULL,

idTarif INT NOT NULL,

PRIMARY KEY (Id\_Container, id\_Tarif),

FOREIGN KEY (Id\_Container) REFERENCES Container(Id\_Container),

FOREIGN KEY (id\_Tarif) REFERENCES Tarif(id\_Tarif)

);

CREATE TABLE Traiter (

IdUserAdmin INT NOT NULL,

IdMessage INT NOT NULL,

IdAction INT NOT NULL,

Date\_Traiter DATETIME,

PRIMARY KEY (Id\_User\_admin, Id\_Message, Id\_Action),

FOREIGN KEY (Id\_User\_admin) REFERENCES User\_admin(Id\_User\_admin),

FOREIGN KEY (Id\_Message) REFERENCES Message(Id\_Message),

FOREIGN KEY (Id\_Action) REFERENCES Action(Id\_Action)

);

CREATE TABLE Modifier\_Contenu (

IdContenu INT NOT NULL,

IdUserAdmin INT NOT NULL,

Date\_Modifier\_Contenu DATETIME NOT NULL,

PRIMARY KEY (Id\_Contenu, Id\_User\_admin, Date\_Modifier\_Contenu),

FOREIGN KEY (Id\_Contenu) REFERENCES Contenu(Id\_Contenu),

FOREIGN KEY (Id\_User\_admin) REFERENCES User\_admin(Id\_User\_admin)

);

CREATE TABLE Modifier\_Tarif (

IdTarif INT NOT NULL,

IdUserAdmin INT NOT NULL,

Date\_Modifier\_Tarif DATETIME NOT NULL,

PRIMARY KEY (id\_Tarif, Id\_User\_admin, Date\_Modifier\_Tarif),

FOREIGN KEY (id\_Tarif) REFERENCES Tarif(id\_Tarif),

FOREIGN KEY (Id\_User\_admin) REFERENCES User\_admin(Id\_User\_admin)

);

-- La clé primaire garantit qu'un même UserAdmin ne peut effectuer qu'une seule fois

-- une même action sur un message donné

-- (ex: Lire une seule fois, Supprimer une seule fois)

-- Remarque importante :

-- - Plusieurs UserAdmin peuvent lire un même message (plusieurs lignes 'Lire')

-- - Un seul UserAdmin peut supprimer un message

-- - C'est la responsabilité de l'application de :

-- -> changer le Statut à 'supprimé' après une action de type 'Supprimer'

-- -> empêcher d'autres actions après suppression (côté application ou trigger)

**🔥 Remarques importantes**

* **Application** ou **Trigger** devra empêcher toute action de lecture après qu'un message soit **supprimé**.
* Si besoin, je peux aussi t’écrire un **exemple de trigger** en SQL pour empêcher automatiquement qu’on lise un message déjà supprimé (optionnel).

| **Élément** | **Géré par** | **Statut** |
| --- | --- | --- |
| Clé d'unicité sur Traiter | PK (id\_user, id\_message, id\_action) | ✅ |
| Possibilité de lecture multiple par plusieurs UserAdmin | Oui | ✅ |
| Suppression unique d'un message | Oui (action 'supprimer' unique par user+message) | ✅ |
| Évolution du statut du message | Par application ou trigger | ✅ |

Script SQL d'exemples d'insertion

-- Insertion d'Actions

INSERT INTO Action (Id\_action, Type) VALUES

(1, 'Lire'),

(2, 'Supprimer');

-- Insertion de Pages Web

INSERT INTO Page\_WEB (Id\_web, Adresse\_URL) VALUES

(1, 'https://example.com');

-- Insertion de Containers

INSERT INTO Container (Id\_container, Nom, Désignation) VALUES

(1, 'Bannière', 'Haut de page'),

(2, 'Pied de page', 'Bas de page');

-- Insertion de Contenus

INSERT INTO Contenu (Id\_contenu, Description, Adresse\_image, Contenu\_Texte) VALUES

(1, 'Image principale', 'image1.png', 'Bienvenue sur notre site');

-- Association entre Page Web et Contenu

INSERT INTO Contenir (Id\_web, Id\_contenu) VALUES

(1, 1);

-- Association entre Contenu et Container

INSERT INTO Associer (Id\_contenu, Id\_container) VALUES

(1, 1),

(1, 2);

-- Insertion de Profils

INSERT INTO Profil (Id\_profil, Nom, Prenom, Adresse, Adresse\_Mail) VALUES

(1, 'Dupont', 'Jean', '10 rue Paris', 'jean.dupont@email.com');

-- Insertion de UserAdmin

INSERT INTO UserAdmin (Id\_user, User, Password, Nom, Prenom, Adresse, Adresse\_Mail) VALUES

(1, 'admin1', 'password1', 'Martin', 'Paul', '20 avenue Lyon', 'paul.martin@email.com'),

(2, 'admin2', 'password2', 'Durand', 'Sophie', '30 boulevard Nice', 'sophie.durand@email.com');

-- Insertion de Messages

INSERT INTO Message (Id\_message, Nom, Prenom, Adresse, Adresse\_Mail, Telephone, Texte, Statut) VALUES

(1, 'Tremblay', 'Alice', '50 rue Bordeaux', 'alice.tremblay@email.com', '0601020304', 'Demande de renseignements', 'nouveau'),

(2, 'Bernard', 'Luc', '70 rue Marseille', 'luc.bernard@email.com', '0605060708', 'Problème de connexion', 'nouveau');

⚡ Trigger pour bloquer la lecture d'un message supprimé

DELIMITER $$

CREATE TRIGGER prevent\_read\_deleted\_message

BEFORE INSERT ON Traiter

FOR EACH ROW

BEGIN

DECLARE msg\_status VARCHAR(20);

-- On récupère le statut du message

SELECT Statut INTO msg\_status

FROM Message

WHERE Id\_message = NEW.Id\_message;

-- Si l'action est Lire et que le message est déjà Supprimé => Interdire

IF (NEW.Id\_action = (SELECT Id\_action FROM Action WHERE Type = 'Lire')) AND (msg\_status = 'supprimé') THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Impossible de lire un message supprimé.';

END IF;

-- Si l'action est Supprimer et que le message est déjà Supprimé => Interdire

IF (NEW.Id\_action = (SELECT Id\_action FROM Action WHERE Type = 'Supprimer')) AND (msg\_status = 'supprimé') THEN

SIGNAL SQLSTATE '45000'

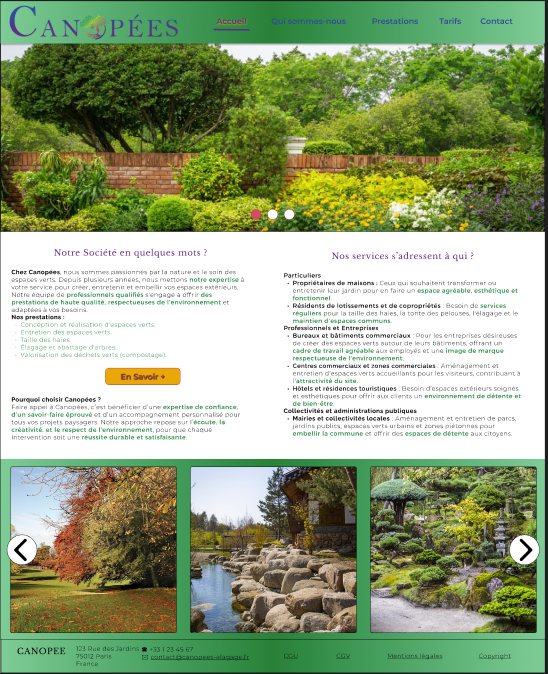
SET MESSAGE\_TEXT = 'Ce message est déjà supprimé.';

END IF;

END$$

DELIMITER ;

Mise a jour 14062025



**Accueil\_slider**

**Accueil\_carrousel**

**Accueil\_collectivite**

**Accueil\_profesionnel**

**Accueil\_particulier**

**Accueil\_marketing**

**Accueil\_prestation**

**Accueil\_presentation**



**Qui\_profil\_2**

**Qui\_profil\_1**

**Qui\_profil\_2**

**Qui\_profil\_1**



**Prestation\_5**

**Prestation\_5**

**Prestation\_4**

**Prestation\_4**

**Prestation\_3**

**Prestation\_3**

**Prestation\_2**

**Prestation\_2**

**Prestation\_1**

**Prestation\_1**



**Prestation\_realisation\_1 (ou 2 ou 3, 4,5)**



**Tarif\_Aménagement**

**Tarif\_Abattage**

**Tarif\_Entretien**

**Tarif\_Plantation**

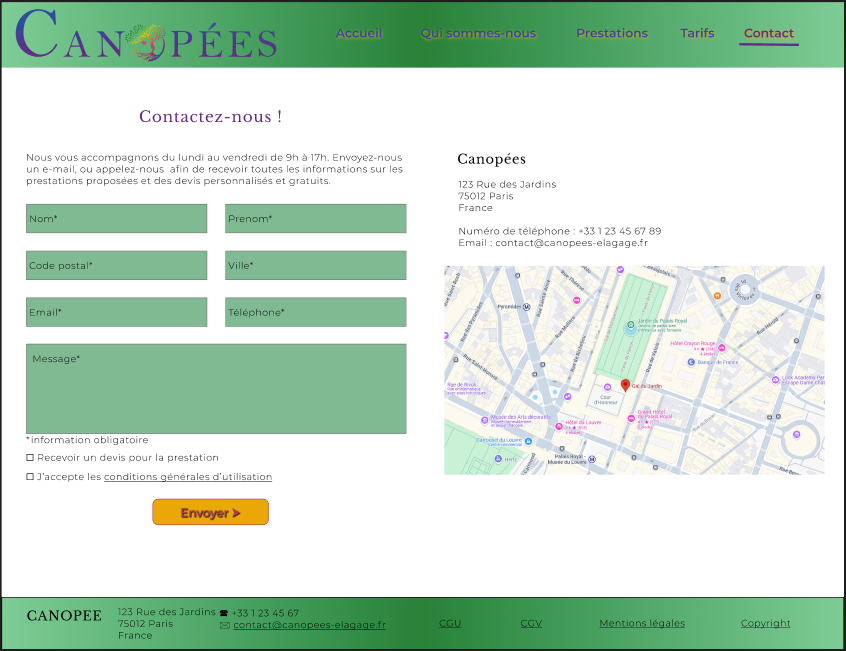
**Tarif\_Preparation**

**Tarif\_Etude**

**Tarif\_Etude**

**Tarif\_Etude**

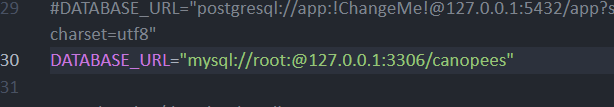
**Tarif\_Etude**



**Contact\_Message**

**Contact\_Formulaire**

Connexion a réaliser vers la base, dans le fichier env.

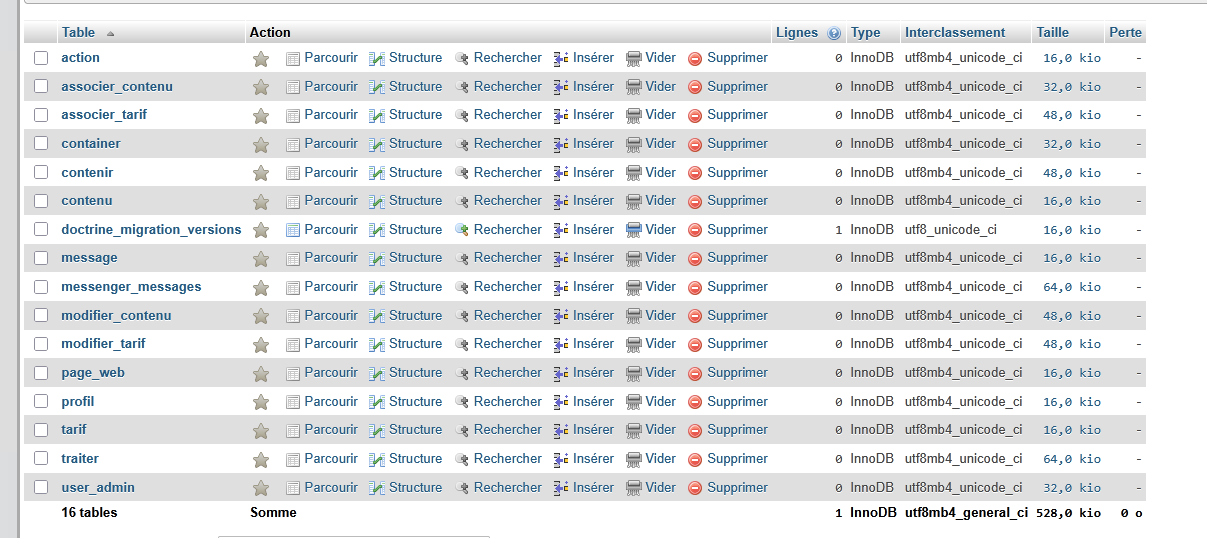


Symfony

Modification du fichier .env.

Création des entités

Création de la base et des tables



**Installer les outils :**

composer require api

composer require easycorp/easyadmin-bundle

composer require doctrine/doctrine-fixtures-bundle

composer require symfony/security-bundle

Installation fakerPHP

**Configuration de la base de données**

Dans le fichier .env :

DATABASE\_URL="mysql://root:password@127.0.0.1:3306/canopee?serverVersion=8.0"

Adapte selon ton utilisateur MySQL et ton mot de passe.

Modification du mot de passe dans MySQL

