

## La base de données

Tout d'abord on copie le fichier .env en .env.local afin de sauvegarder nos paramètres concernant la base de données. Le .env.local fait partie du .gitignore et ne sera donc pas envoyé lors du partage du code source.

➔ `cp .env .env.local`

Et dans ce fichier, nous modifierons la ligne DATABASE\_URL qui est en commentaire :

`DATABASE_URL=mysql://(login):(motDePasse)@127.0.0.1:3306/(baseDeDonnées)?  
serverVersion=(laVersion)`

/ !\ si on met un \_ dans le nom de la base de données, alors, elle sera créée dans un fichier db

Création et initialisation de la base de données

- ➔ `php bin/console doctrine:database:create`
- ➔ `php bin/console make:migration`
- ➔ `php bin/console doctrine:migrations:migrate`

Maintenant nous allons créer les tables, pour cela il suffit de suivre les instructions :

➔ `php bin/console make:entity` ( pour créer/modifier une nouvelle table )

Exemple type :

```
login4060@symfony4-4060:~/public/englearnys$ php bin/console make:entity

Class name of the entity to create or update (e.g. OrangeGnome):
> Categorie

created: src/Entity/Categorie.php
created: src/Repository/CategorieRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
> libelle

Field type (enter ? to see all types) [string]:
>

Field length [255]:
> 100

Can this field be null in the database (nullable) (yes/no) [no]:
> no

updated: src/Entity/Categorie.php

Add another property? Enter the property name (or press <return> to stop adding fields):
>

Success!

Next: When you're ready, create a migration with php bin/console make:migration

login4060@symfony4-4060:~/public/englearnys$
```

Ici nous venons de créer la table `Categorie` avec un libelle, type string et non null

```
login4060@symfony4-4060:~/public/englearnny$ php bin/console doctrine:migrations:migrate

WARNING! You are about to execute a database migration that could result in schema changes and data
loss. Are you sure you wish to continue? (yes/no) [yes]:
>

[notice] Migrating up to DoctrineMigrations\Version20201204100936
[notice] finished in 323.6ms, used 18M memory, 1 migrations executed, 1 sql queries

login4060@symfony4-4060:~/public/englearnny$
```

Et ici, nous venons d'envoyer les modifications au serveur.

Dans la capture d'écran ci-dessous, nous verrons comment nous relier la table `catégorie` à la table `mot` :

```
login4060@symfony4-4060:~/public/englearnny$ php bin/console make:entity

Class name of the entity to create or update (e.g. OrangePopsicle):
> Categorie

Your entity already exists! So let's add some new fields!

New property name (press <return> to stop adding fields):
> mot

Field type (enter ? to see all types) [string]:
> OneToMany

What class should this entity be related to?:
> Mot

A new property will also be added to the Mot class so that you can access and set the related Categ
orie object from it.

New field name inside Mot [categorie]:
>

Is the Mot.categorie property allowed to be null (nullable)? (yes/no) [yes]:
> no

Do you want to activate orphanRemoval on your relationship?
A Mot is "orphaned" when it is removed from its related Categorie.
e.g. $categorie->removeMot($mot)

NOTE: If a Mot may *change* from one Categorie to another, answer "no".

Do you want to automatically delete orphaned App\Entity\Mot objects (orphanRemoval)? (yes/no) [no]:
> yes

updated: src/Entity/Categorie.php
updated: src/Entity/Mot.php
```

On prend la table `Categorie`, on lui ajoute le champ « `mot` », lié en `OneToMany`, à la table « `Mot` », enfin, la propriété `categorie` dans la table `Mot` est non null.

Ci-dessous la création de la table RealiseTest se trouvant entre Test et Utilisateur :

```
login4060@symfony4-4060:~/public/englearnny$ php bin/console make:entity

Class name of the entity to create or update (e.g. BravePizza):
> RealiseTest

created: src/Entity/RealiseTest.php
created: src/Repository/RealiseTestRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
> realiseAujourdHui

Field type (enter ? to see all types) [string]:
> boolean

Can this field be null in the database (nullable) (yes/no) [no]:
>

updated: src/Entity/RealiseTest.php

Add another property? Enter the property name (or press <return> to stop adding fields):
> score

Field type (enter ? to see all types) [string]:
> integer

Can this field be null in the database (nullable) (yes/no) [no]:
> yes

updated: src/Entity/RealiseTest.php

Add another property? Enter the property name (or press <return> to stop adding fields):
>

Success!
```

Puis la liaison entre ces tables :

```
login4060@symfony4-4060:~/public/englearnny$ php bin/console make:entity

Class name of the entity to create or update (e.g. GentlePopsicle):
> RealiseTest

Your entity already exists! So let's add some new fields!

New property name (press <return> to stop adding fields):
> test

Field type (enter ? to see all types) [string]:
> ManyToOne

What class should this entity be related to?:
> Test

Is the RealiseTest.test property allowed to be null (nullable)? (yes/no) [yes]:
> no

Do you want to add a new property to Test so that you can access/update RealiseTest objects from it - e.g. $test->getRealiseTests()? (yes/no) [yes]:
>

A new property will also be added to the Test class so that you can access the related RealiseTest objects from it.

New field name inside Test [realiseTests]:
>

Do you want to activate orphanRemoval on your relationship?
A RealiseTest is "orphaned" when it is removed from its related Test.
e.g. $test->removeRealiseTest($realiseTest)

NOTE: If a RealiseTest may "change" from one Test to another, answer "no".

Do you want to automatically delete orphaned App\Entity\RealiseTest objects (orphanRemoval)? (yes/no) [no]:
> no

updated: src/Entity/RealiseTest.php
updated: src/Entity/Test.php
```

```

Add another property? Enter the property name (or press <return> to stop adding fields):
> utilisateur

Field type (enter ? to see all types) [string]:
> ManyToOne

What class should this entity be related to?:
> Utilisateur

Is the RealiseTest.utilisateur property allowed to be null (nullable)? (yes/no) [yes]:
> no

Do you want to add a new property to Utilisateur so that you can access/update RealiseTest objects from it - e.g. $utilisateur->getRealiseTests()? (yes/no) [yes]:
>

A new property will also be added to the Utilisateur class so that you can access the related RealiseTest objects from it.

New field name inside Utilisateur [realiseTests]:
>

Do you want to activate orphanRemoval on your relationship?
A RealiseTest is "orphaned" when it is removed from its related Utilisateur.
e.g. $utilisateur->removeRealiseTest($realiseTest)

NOTE: If a RealiseTest may *change* from one Utilisateur to another, answer "no".

Do you want to automatically delete orphaned App\Entity\RealiseTest objects (orphanRemoval)? (yes/no) [no]:
>

updated: src/Entity/RealiseTest.php
updated: src/Entity/Utilisateur.php

Add another property? Enter the property name (or press <return> to stop adding fields):
>

Success!

```

Voici pour la partie base de données.

Maintenant, nous pouvons récupérer la base de données créée puisque celle-ci a été créée sur la machine de Maxence. Pour cela il suffit de :

- ➔ git pull
- ➔ composer install
- ➔ php bin/console doctrine:database:create
- ➔ php bin/console make:migration
- ➔ php bin/console doctrine:migrations:migrate

Une fois ces 5 étapes effectuées, nous pouvons vérifier sur phpMyAdmin que tout s'est bien passé :

	Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/>	abonnement	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
<input type="checkbox"/>	categorie	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
<input type="checkbox"/>	doctrine_migration_versions	★ Parcourir Structure Rechercher Insérer Vider Supprimer	4	InnoDB	utf8_unicode_ci	16,0 kio	-
<input type="checkbox"/>	entreprise	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
<input type="checkbox"/>	liste	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
<input type="checkbox"/>	mot	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
<input type="checkbox"/>	mot_liste	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
<input type="checkbox"/>	realise_test	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
<input type="checkbox"/>	role	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
<input type="checkbox"/>	test	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	48,0 kio	-
<input type="checkbox"/>	theme	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	16,0 kio	-
<input type="checkbox"/>	utilisateur	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_unicode_ci	80,0 kio	-
	12 tables	Somme	4	InnoDB	utf8mb4_general_ci	416,0 kio	0 0