

Symfony 4

DÉVELOPPEMENT WEB AVANCÉ – UTILISATION D'UN FRAMEWORK

Doctrine – Qu'est-ce que c'est

2

- ▶ Librairie PHP
- ▶ Aucun code SQL (ou très peu...)
- ▶ Mapper des class PHP en table SQL

Liaison à la base de données

3

- ▶ .env
- ▶ DATABASE_URL=mysql://db_user:db_password@127.0.0.1:3306/db_name
- ▶ DATABASE_URL=mysql://root:password@127.0.0.1:3306/my_project

Création de la base de données

4

- ▶ `php bin/console doctrine:schema:create`

```
root@c4b87f4dce5d:/var/www/html# php bin/console doctrine:schema:create
```

```
!  
! [CAUTION] This operation should not be executed in a production environment!  
!
```

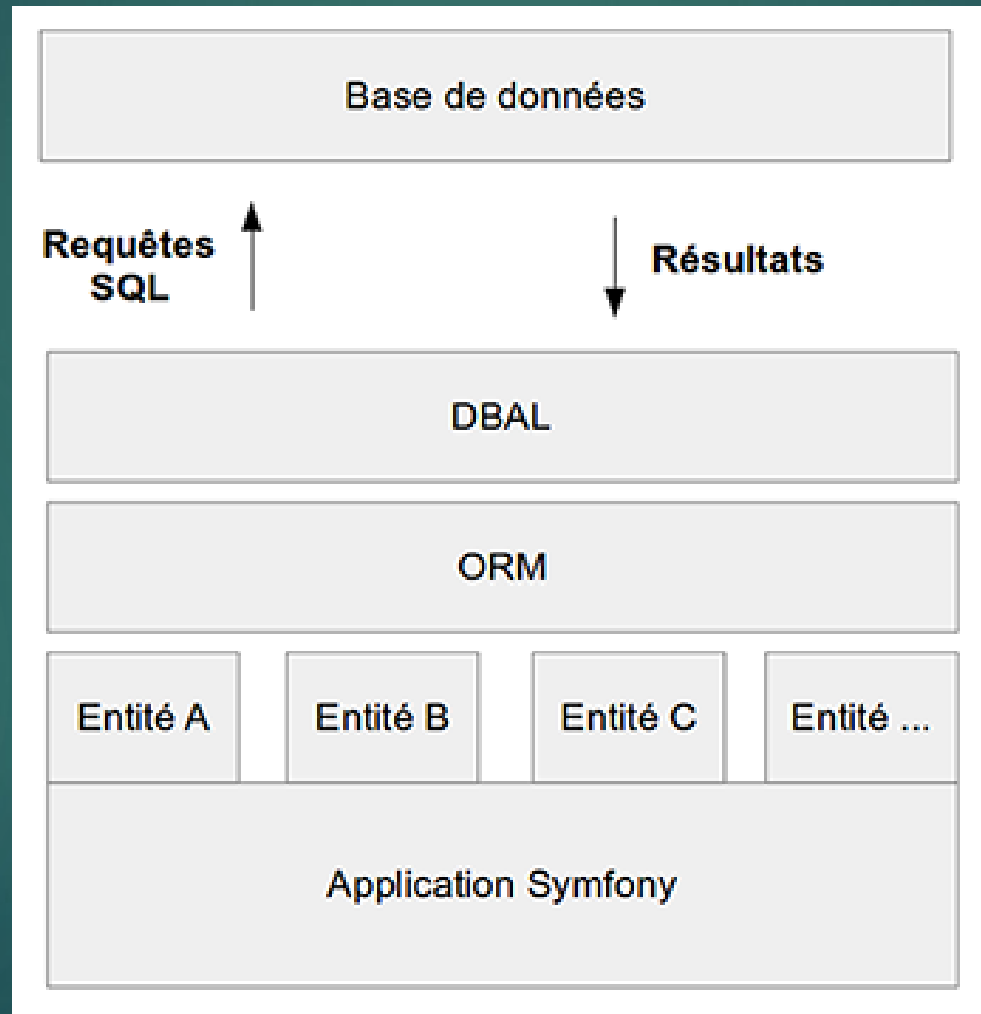
```
Creating database schema...
```

```
[OK] Database schema created successfully!
```

```
root@c4b87f4dce5d:/var/www/html# |
```

Doctrine – Kesako

5



Doctrine – Création d'une entité

6

```
<?php
```

```
namespace App\Entity;
```

```
use Doctrine\ORM\Mapping as ORM;
```

```
/**
```

```
 * @ORM\Entity(repositoryClass="App\Repository\ProductRepository")
```

```
 */
```

```
class Product
```

```
{
```

Doctrine – Création d'une entité

7

```
/**
 * @ORM\Id()
 * @ORM\GeneratedValue()
 * @ORM\Column(type="integer")
 */
private $id;

/**
 * @ORM\Column(type="string", length=255)
 */
private $name;

/**
 * @ORM\Column(type="decimal", precision=7, scale=2)
 */
private $price;

/**
 * @ORM\Column(type="boolean")
 */
private $active;
```

Doctrine – Création d'une entité

```
public function getName(): ?string
{
    return $this->name;
}

public function setName(string $name): self
{
    $this->name = $name;

    return $this;
}
```


Doctrine – Création d'une entité

```
/**  
 * @ORM\Column(type="string", length=255)  
 */
```

```
/**  
 * @ORM\Column(type="decimal", precision=7, scale=2)  
 */
```

```
/**  
 * @ORM\Column(type="string", length=40, unique=true)  
 */
```

Doctrine – Création d'une entité

10

- ▶ String
- ▶ Integer
- ▶ Boolean
- ▶ Datetime
- ▶ Array
- ▶ Float
- ▶ ...

Doctrine – Création d'une entité

11

```
/**  
 * @ORM\Id()  
 * @ORM\GeneratedValue()  
 * @ORM\Column(type="integer")  
 */  
private $id;
```

Doctrine – Relation

12

- ▶ Tiré de la méthode Merise
- ▶ Utilisé dans 99% des projets
- ▶ Toujours difficile à retenir...

Doctrine – ManyToOne Bidirectionnelle

13

```
/**
 * @ORM\ManyToOne(targetEntity="App\Entity\Category", inversedBy="products")
 * @ORM\JoinColumn(nullable=false)
 */
private $category;
```

```
public function getCategory(): ?Category
{
    return $this->category;
}

public function setCategory(?Category $category): self
{
    $this->category = $category;

    return $this;
}
```

Doctrine – ManyToOne Bidirectionnelle

14

```
/**  
 * @ORM\OneToMany(targetEntity="App\Entity\Product", mappedBy="category", orphanRemoval=true)  
 */  
private $products;
```

Doctrine – ManyToOne Bidirectionnelle

15

```
/**  
 * @return Collection|Product[]  
 */  
public function getProducts(): Collection  
{  
    return $this->products;  
}
```

Doctrine – ManyToOne Bidirectionnelle

16

```
public function addProduct(Product $product): self
{
    if (!$this->products->contains($product)) {
        $this->products[] = $product;
        $product->setCategory($this);
    }

    return $this;
}
```


Doctrine – ManyToOne Bidirectionnelle

17

```
public function removeProduct(Product $product): self
{
    if ($this->products->contains($product)) {
        $this->products->removeElement($product);
        // set the owning side to null (unless already changed)
        if ($product->getCategory() === $this) {
            $product->setCategory(category: null);
        }
    }

    return $this;
}
```

Doctrine – MakerBundle

18

- ▶ `php bin/console make:entity`

```
root@c4b87f4dce5d:/var/www/html# php bin/console make:entity
```

```
Class name of the entity to create or update (e.g. VictoriousPuppy):
```

```
> Category
```

```
created: src/Entity/Category.php
```

```
created: src/Repository/CategoryRepository.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):
```

```
> name
```

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

```
> |
```

Doctrine – Migration

19

- ▶ `php bin/console make:migration`
- ▶ Une migration est un simple fichier PHP qui note les changements entre votre base de données actuelle et vos fichiers PHP
- ▶ Il gère la possibilité d'upgrader ou de downgrader votre base de données, en automatique

Doctrine – Up et Down

20

```
public function up(Schema $schema) : void
{
    // this up() migration is auto-generated, please modify it to your needs
    $this->abortIf( condition: $this->connection->getDatabasePlatform()->getName() !== 'mysql', message: 'Migration can only be executed safely on \'mysql\'');

    $this->addSql( sql: 'ALTER TABLE product ADD category_id INT NOT NULL, ADD reference VARCHAR(40) DEFAULT NULL');
    $this->addSql( sql: 'ALTER TABLE product ADD CONSTRAINT FK_D34A04AD12469DE2 FOREIGN KEY (category_id) REFERENCES category (id)');
    $this->addSql( sql: 'CREATE INDEX IDX_D34A04AD12469DE2 ON product (category_id)');
}
```

```
public function down(Schema $schema) : void
{
    // this down() migration is auto-generated, please modify it to your needs
    $this->abortIf( condition: $this->connection->getDatabasePlatform()->getName() !== 'mysql', message: 'Migration can only be executed safely on \'mysql\'');

    $this->addSql( sql: 'ALTER TABLE product DROP FOREIGN KEY FK_D34A04AD12469DE2');
    $this->addSql( sql: 'DROP INDEX IDX_D34A04AD12469DE2 ON product');
    $this->addSql( sql: 'ALTER TABLE product DROP category_id, DROP reference');
}
```

Doctrine - PostUp

21

```
public function postUp(Schema $schema) : void
{
    $this->addSql( sql: "INSERT INTO category(name) VALUE('Ma catégorie')");
}
```

Doctrine – Migration migrate

22

- ▶ `php bin/console doctrine:migration:migrate`
- ▶ `php bin/console d:m:m`

Adminer – PHPMyAdmin – Command line

- ▶ Pour accéder à votre base de données, 3 possibilités :
 - ▶ Adminer : un seul fichier à déposer à la racine de votre projet :
 - ▶ <https://www.adminer.org/>
 - ▶ PHPMyAdmin : Outil complet
 - ▶ <https://www.phpmyadmin.net/>
 - ▶ Commande line : `mysql -u MonUser -p`