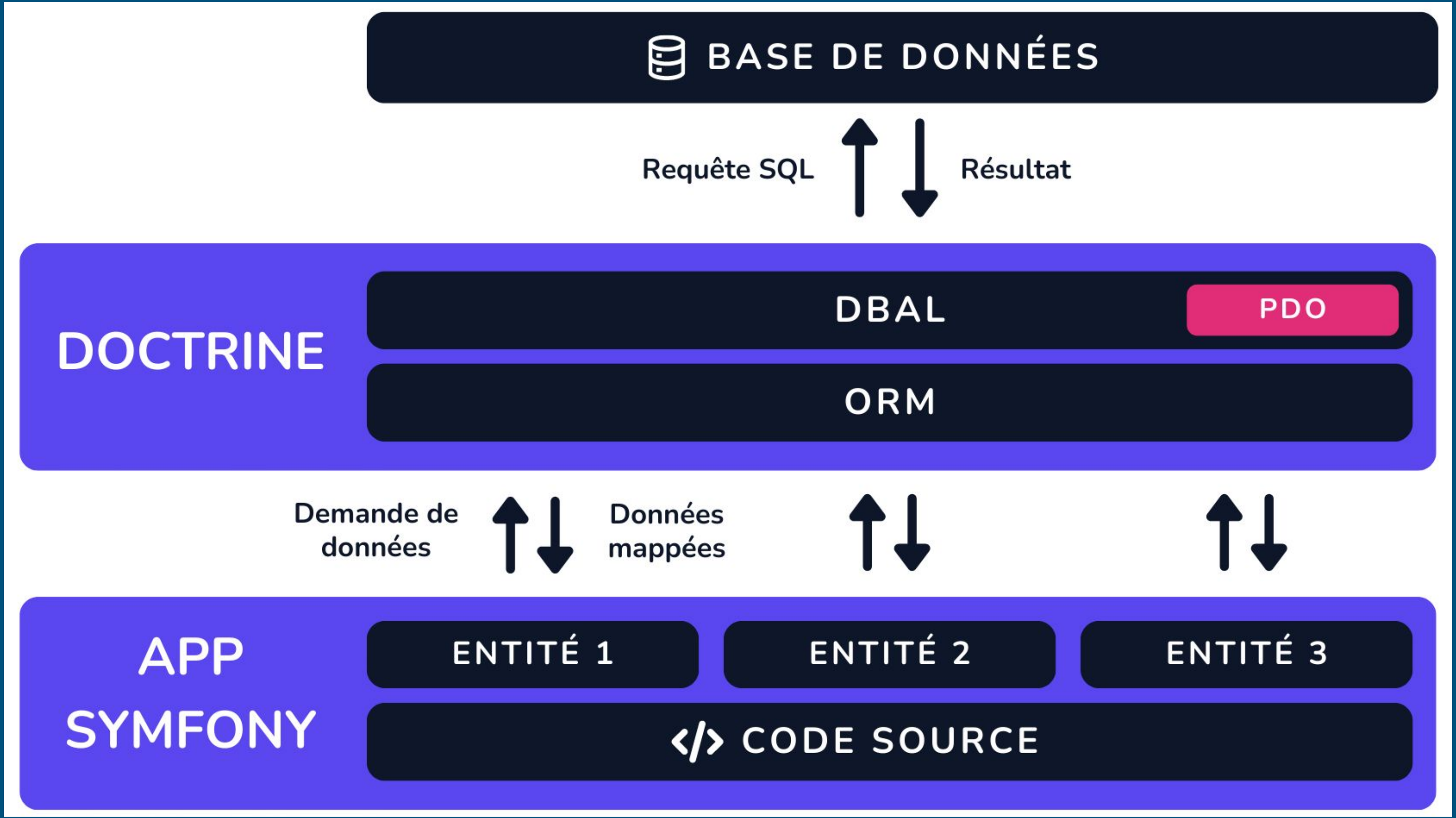
Doctrine

C’est quoi?

Librairie PHP conçue pour faciliter la manipulation des bases de données et le mapping des objets.
Il est qualifié de ORM et DBAL.



BASE DE DONNÉES

Requête SQL

Résultat

DOCTRINE

DBAL

PDO

ORM

Demande de données

Données mappées

APP SYMFONY

ENTITÉ 1

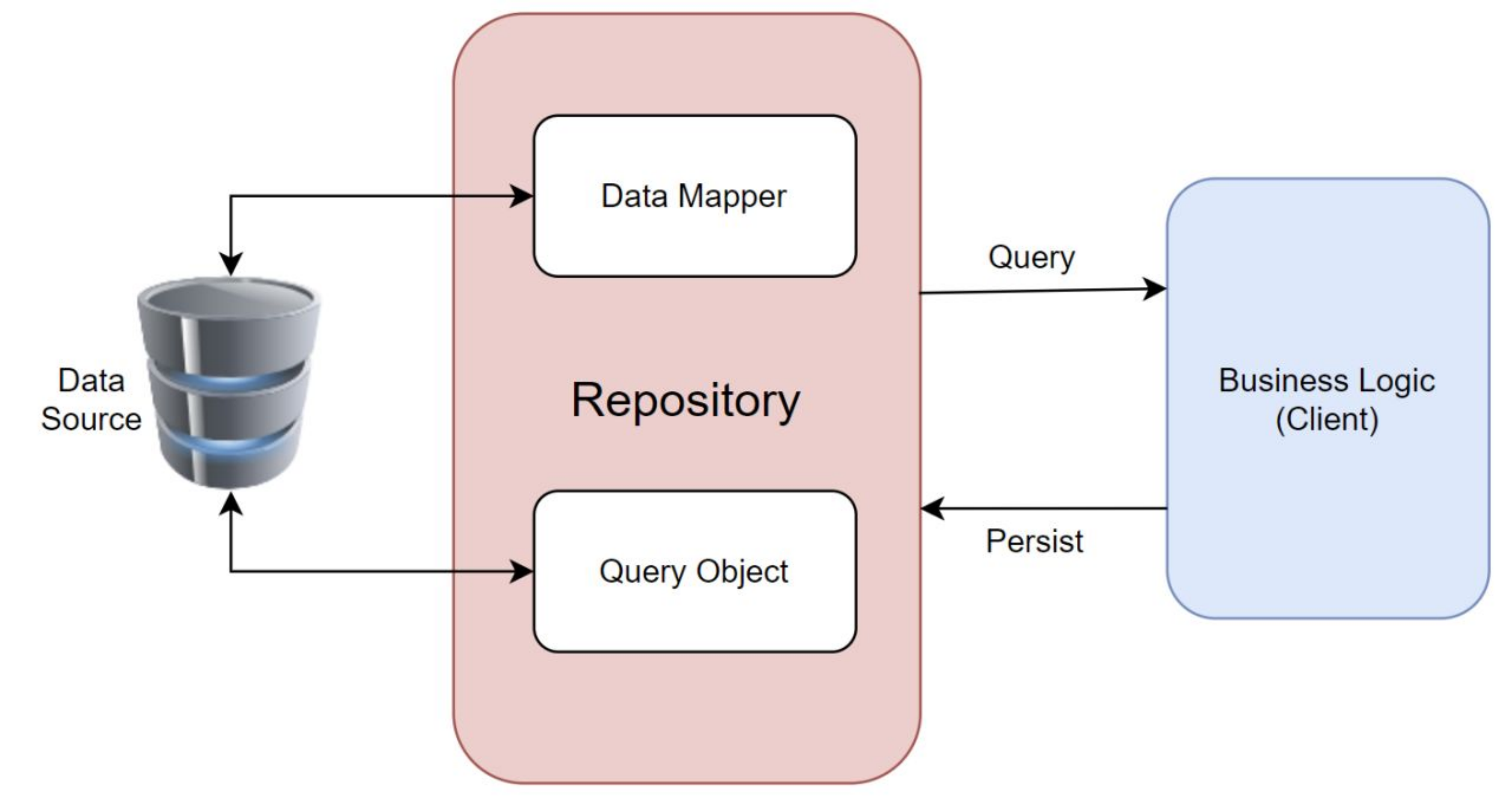
ENTITÉ 2

ENTITÉ 3

CODE SOURCE

2 patterns objets pour faire la correspondance ses objets PHP :

- Data Mapper
- Unit of Work


Data Mapper
Query
Data Source
Repository
Business Logic (Client)
Persist
Query Object

Création Base de Données

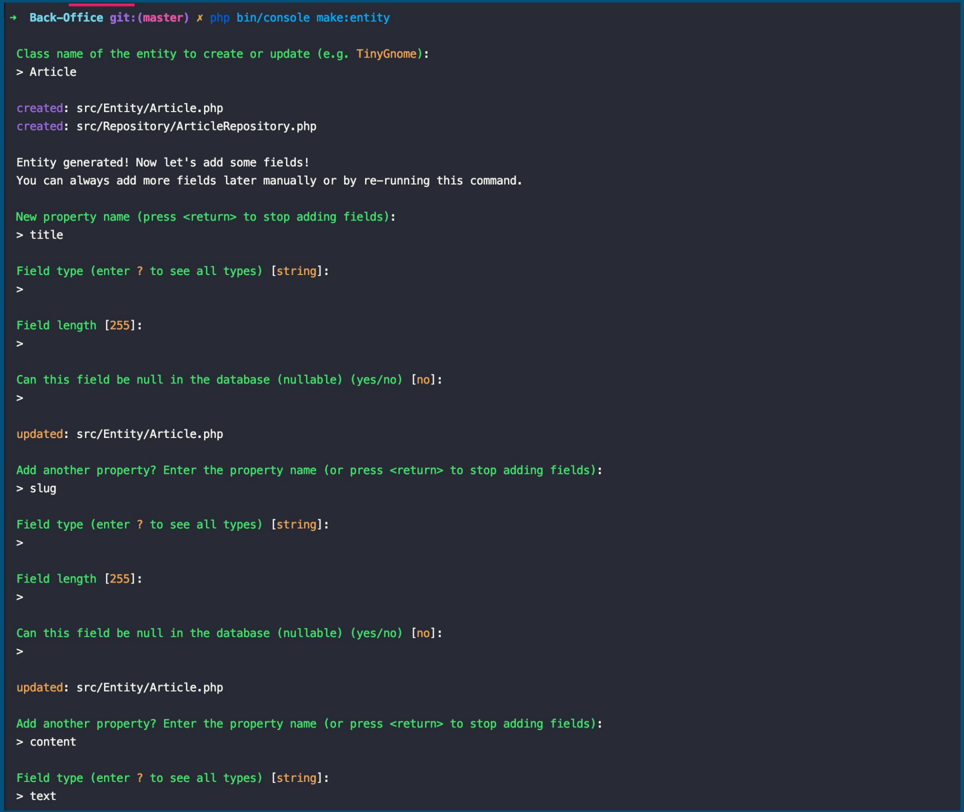
Fichier .env :

```
DATABASE_URL="mysql://app:!ChangeMe!@127.0.0.1:3306/app?serverVersion=8&charset=utf8mb4"
```

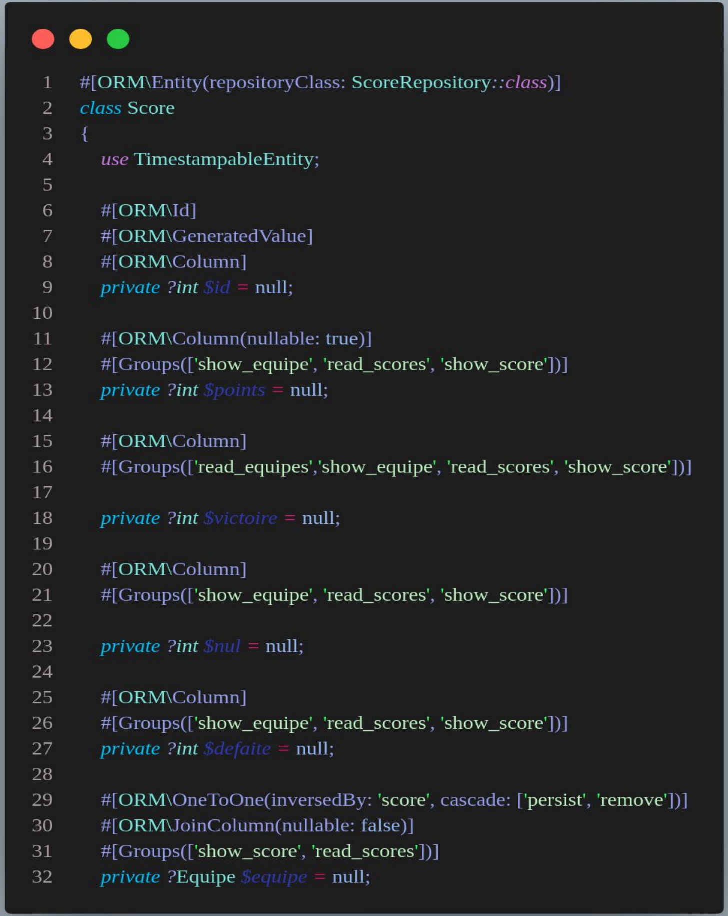
Commande :

```
symfony console doctrine:database:create
```

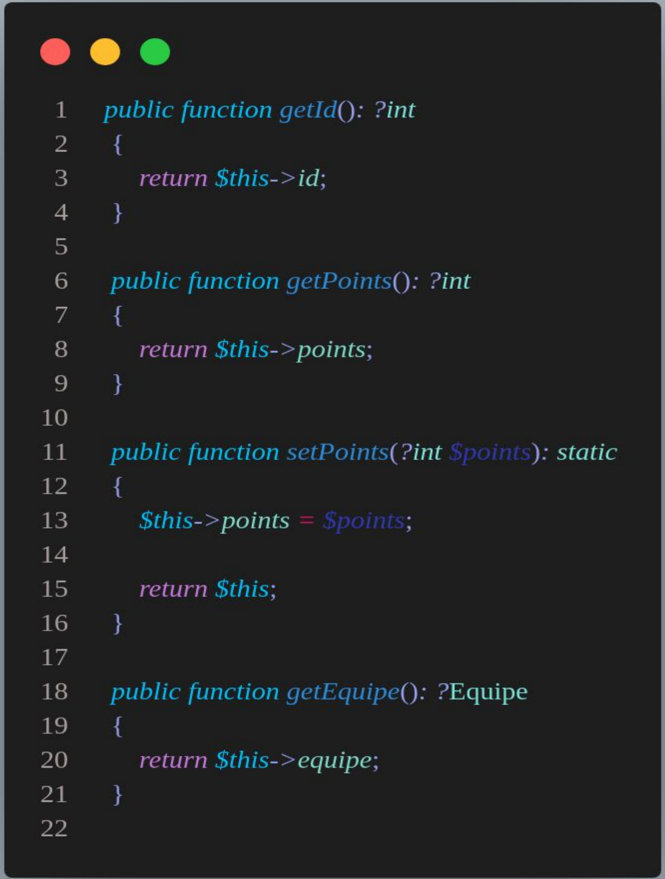
Création entité



```
➜  Back-Office git:(master) ✗ php bin/console make:entity

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

 created: src/Entity/Article.php
 created: src/Repository/ArticleRepository.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):
 > title

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

 Field length [255]:
 >

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

 updated: src/Entity/Article.php

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

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

 Field length [255]:
 >

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

 updated: src/Entity/Article.php

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

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

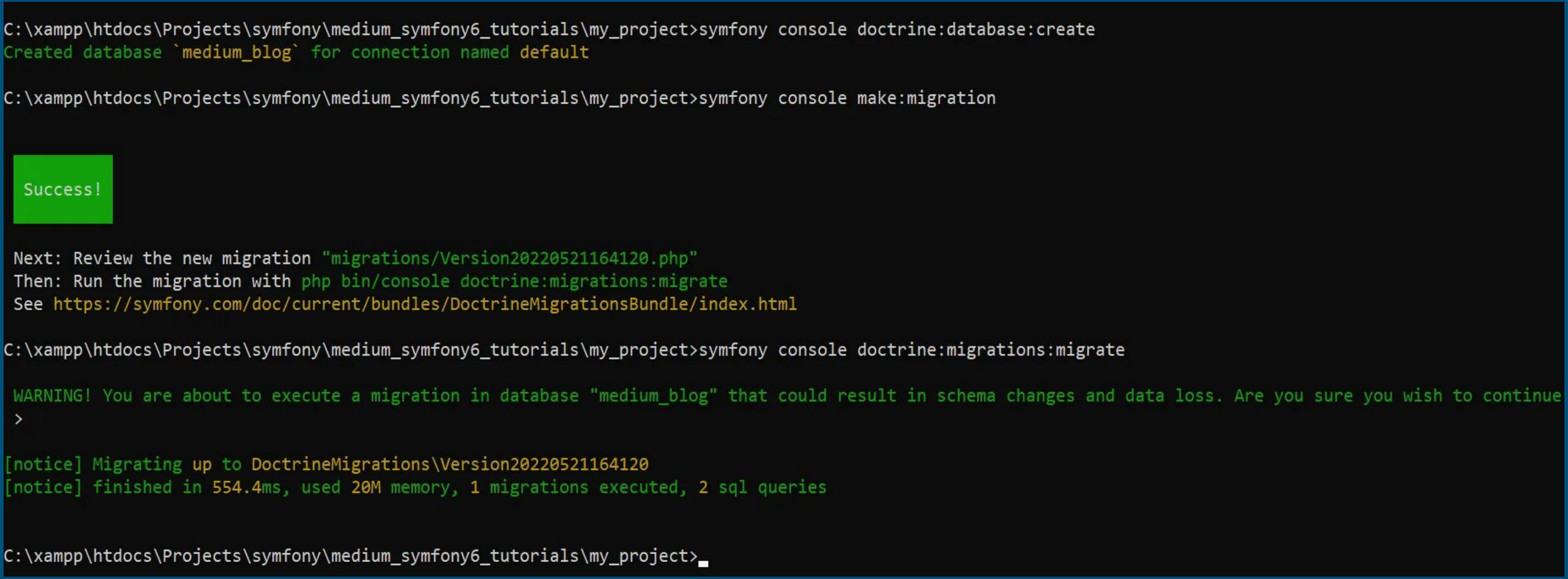


```php
#[ORM\Entity(repositoryClass: ScoreRepository::class)]
class Score
{
    use TimestampableEntity;

    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

    #[ORM\Column(nullable: true)]
    #[Groups(['show_equipe', 'read_scores', 'show_score'])]
    private ?int $points = null;

    #[ORM\Column]
    #[Groups(['read_equipes','show_equipe', 'read_scores', 'show_score'])]

    private ?int $victoire = null;

    #[ORM\Column]
    #[Groups(['show_equipe', 'read_scores', 'show_score'])]

    private ?int $nul = null;

    #[ORM\Column]
    #[Groups(['show_equipe', 'read_scores', 'show_score'])]
    private ?int $defaite = null;

    #[ORM\OneToOne(inversedBy: 'score', cascade: ['persist', 'remove'])]
    #[ORM\JoinColumn(nullable: false)]
    #[Groups(['show_score', 'read_scores'])]
    private ?Equipe $equipe = null;
```

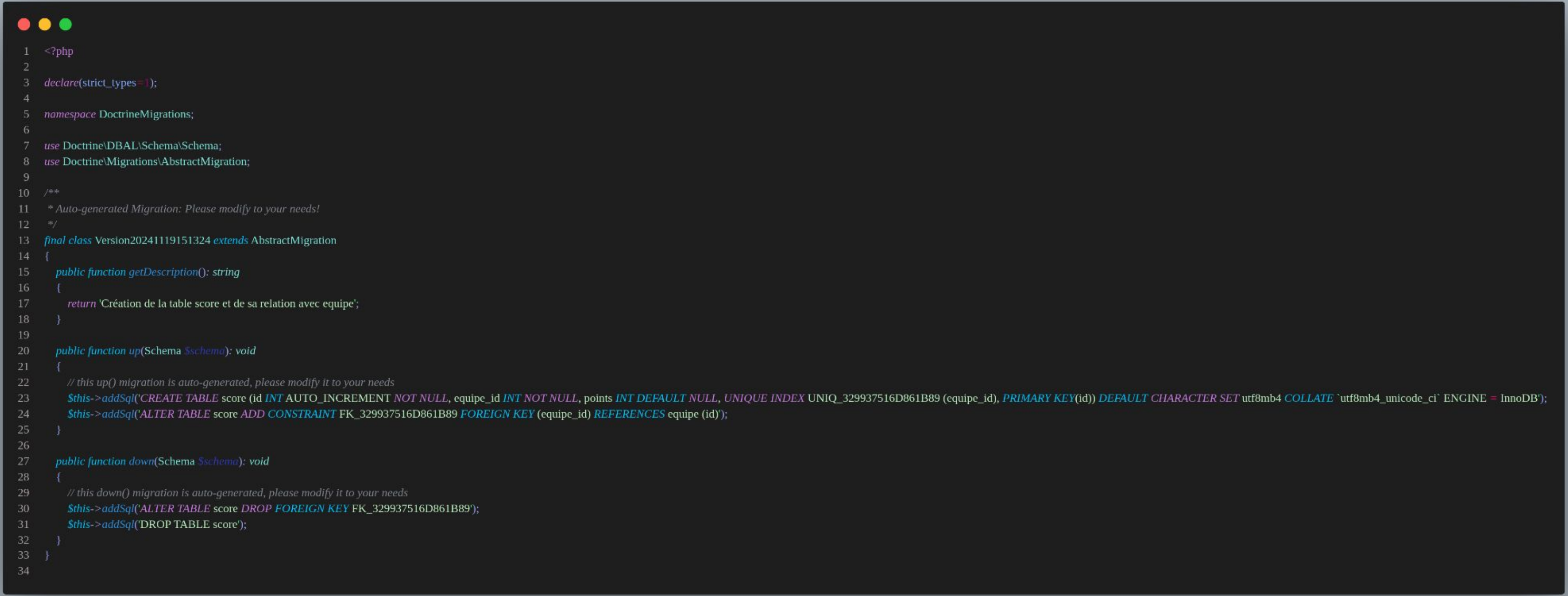


```php
public function getId(): ?int
{
    return $this->id;
}

public function getPoints(): ?int
{
    return $this->points;
}

public function setPoints(?int $points): static
{
    $this->points = $points;

    return $this;
}

public function getEquipe(): ?Equipe
{
    return $this->equipe;
}

```

Migration

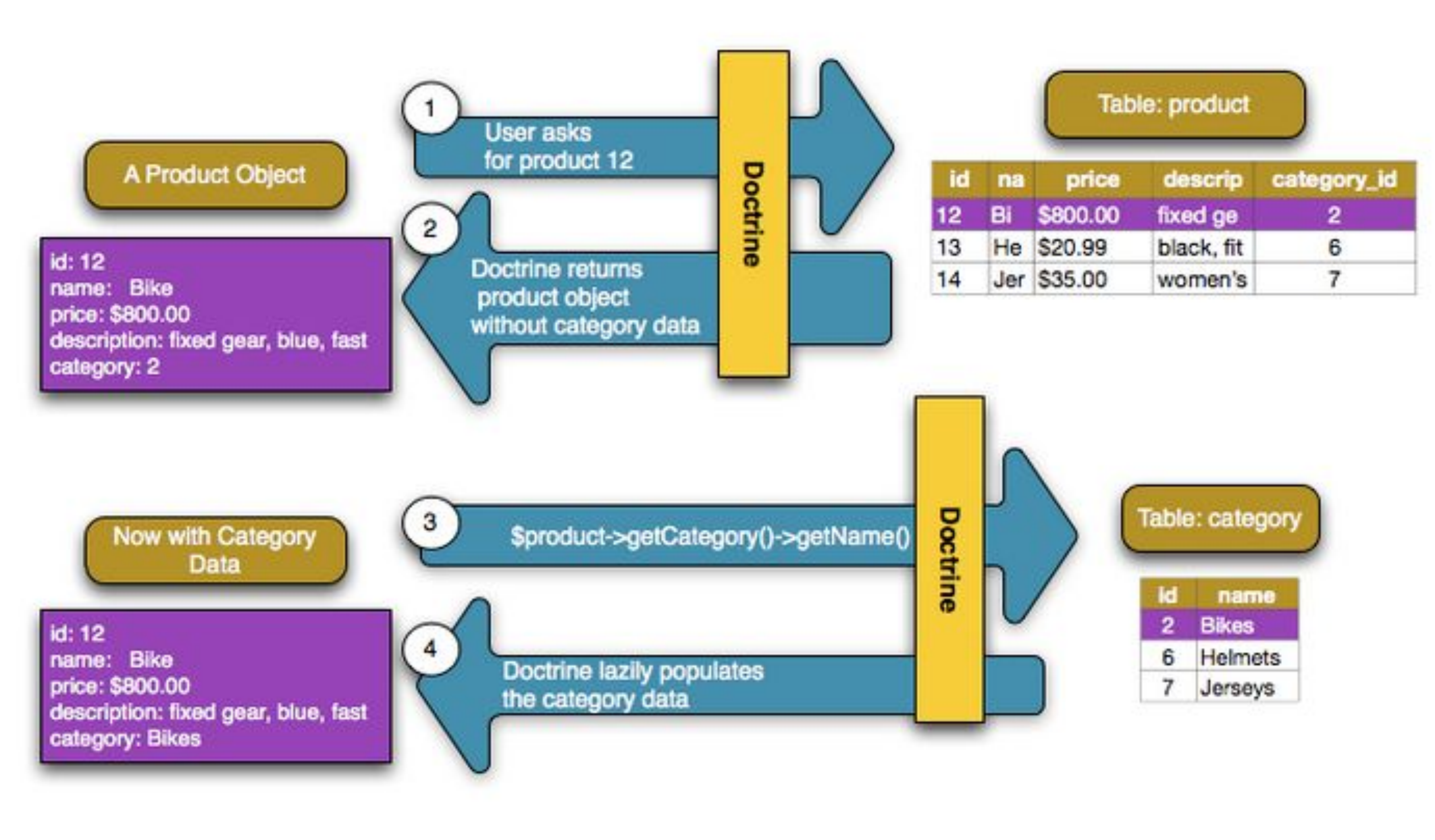


```
C:\xampp\htdocs\Projects\symfony\medium_symfony6_tutorials\my_project>symfony console doctrine:database:create
Created database `medium_blog` for connection named default

C:\xampp\htdocs\Projects\symfony\medium_symfony6_tutorials\my_project>symfony console make:migration


 Success!

 Next: Review the new migration "migrations/Version20220521164120.php"
 Then: Run the migration with php bin/console doctrine:migrations:migrate
 See https://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html

C:\xampp\htdocs\Projects\symfony\medium_symfony6_tutorials\my_project>symfony console doctrine:migrations:migrate

 WARNING! You are about to execute a migration in database "medium_blog" that could result in schema changes and data loss. Are you sure you wish to continue
 >

[notice] Migrating up to DoctrineMigrations\Version20220521164120
[notice] finished in 554.4ms, used 20M memory, 1 migrations executed, 2 sql queries


C:\xampp\htdocs\Projects\symfony\medium_symfony6_tutorials\my_project>
```



```php
<?php

declare(strict_types=1);

namespace DoctrineMigrations;

use Doctrine\DBAL\Schema\Schema;
use Doctrine\Migrations\AbstractMigration;

/**
 * Auto-generated Migration: Please modify to your needs!
 */
final class Version20241119151324 extends AbstractMigration
{
    public function getDescription(): string
    {
        return 'Création de la table score et de sa relation avec equipe';
    }

    public function up(Schema $schema): void
    {
        // this up() migration is auto-generated, please modify it to your needs
        $this->addSql('CREATE TABLE score (id INT AUTO_INCREMENT NOT NULL, equipe_id INT NOT NULL, points INT DEFAULT NULL, UNIQUE INDEX UNIQ_329937516D861B89 (equipe_id), PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');
        $this->addSql('ALTER TABLE score ADD CONSTRAINT FK_329937516D861B89 FOREIGN KEY (equipe_id) REFERENCES equipe (id)');
    }

    public function down(Schema $schema): void
    {
        // this down() migration is auto-generated, please modify it to your needs
        $this->addSql('ALTER TABLE score DROP FOREIGN KEY FK_329937516D861B89');
        $this->addSql('DROP TABLE score');
    }
}

```

symfony console doctrine:migrations:migrate



A Product Object

id: 12
name: Bike
price: $800.00
description: fixed gear, blue, fast
category: 2

1 User asks for product 12

2 Doctrine returns product object without category data

Doctrine

Table: product

id	na	price	descrip	category_id
12	Bi	$800.00	fixed ge	2
13	He	$20.99	black, fit	6
14	Jer	$35.00	women's	7

Now with Category Data

id: 12
name: Bike
price: $800.00
description: fixed gear, blue, fast
category: Bikes

3 $product->getCategory()->getName()

4 Doctrine lazily populates the category data

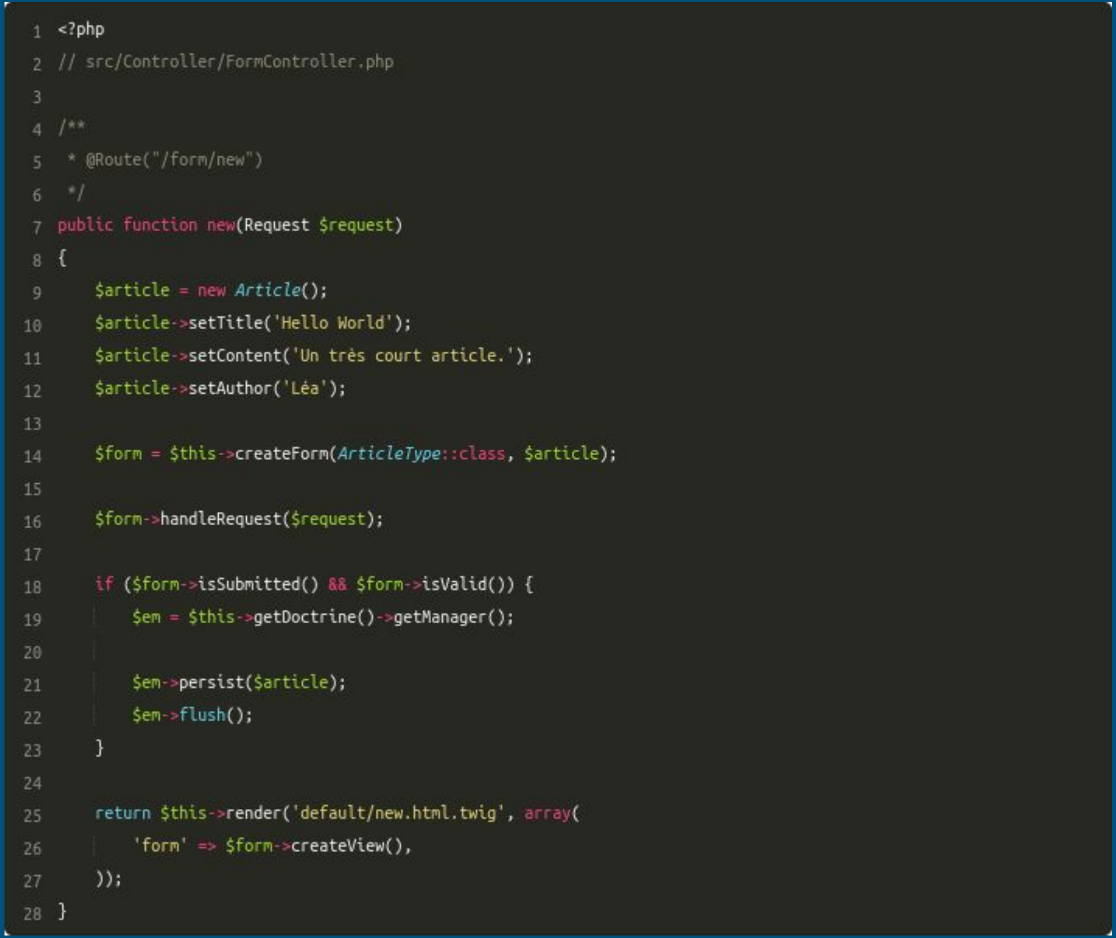
Doctrine

Table: category

id	name
2	Bikes
6	Helmets
7	Jerseys

Entity Manager

- Find
- Persist
- Flush
- Remove



```php
<?php
// src/Controller/FormController.php

/**
 * @Route("/form/new")
 */
public function new(Request $request)
{
    $article = new Article();
    $article->setTitle('Hello World');
    $article->setContent('Un très court article.');
    $article->setAuthor('Léa');

    $form = $this->createForm(ArticleType::class, $article);

    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $em = $this->getDoctrine()->getManager();

        $em->persist($article);
        $em->flush();
    }

    return $this->render('default/new.html.twig', array(
        'form' => $form->createView(),
    ));
}
```

Utilité

- Abstraction des détails de la base de données
- Productivité accrue
- Maintenabilité
- Sécurité

Inconvénients et Points de Vigilance

- **Surcouche et Performances**
- **Courbe d'apprentissage**
- **Flexibilité Limitée**

Sources

laconsole.dev (Formation Doctrine)
linkedin (Eric Venturino, qu'est qu'un ORM?)
symfony (Site officiel)
Boris Bruyere (Blog, développeur freelance)
OpenClassrooms