



# CLASSES - OBJETS

DAO

**BTS SIO1 – ANNÉE 2022/2023 – LYCEE PERGAUD**

# RAPPEL

## Diagramme de classes

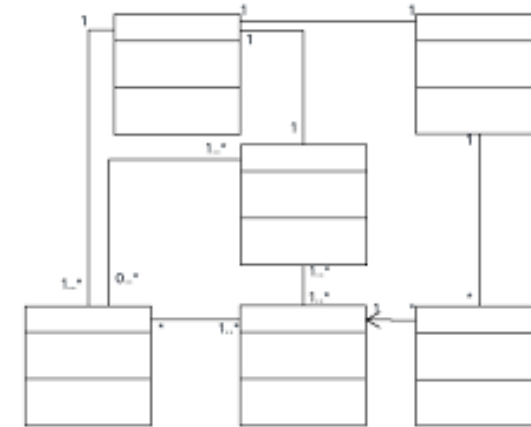


**1 analyser**

**2 modéliser**



**Application**



**3 implémenter**

**BD**

**Classes**

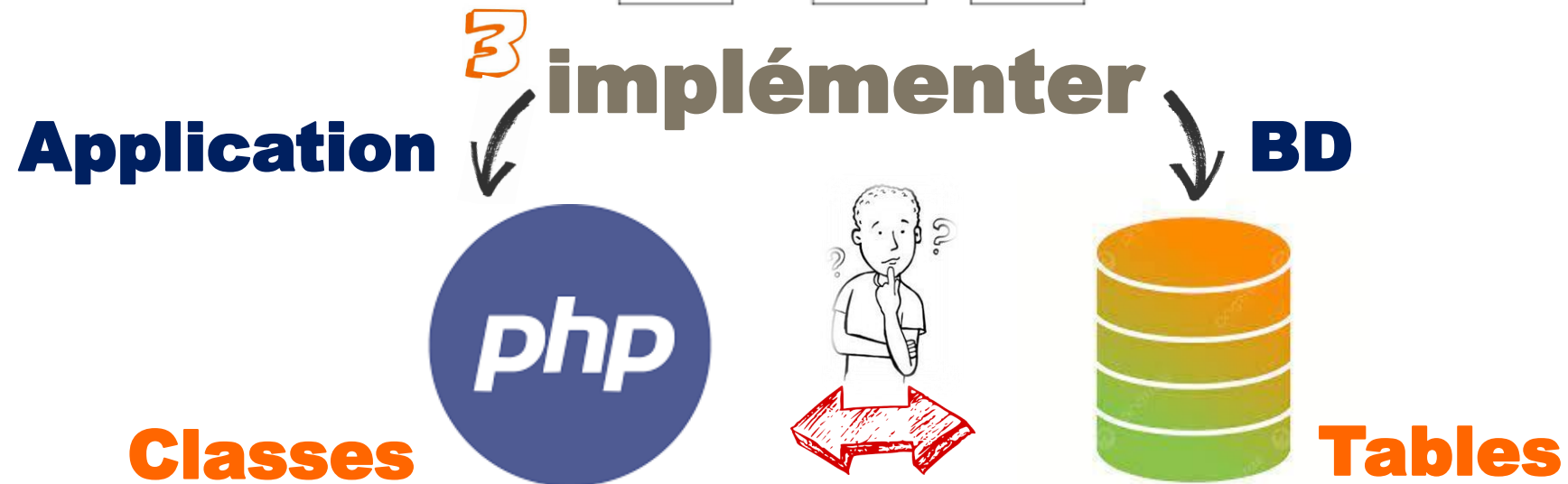
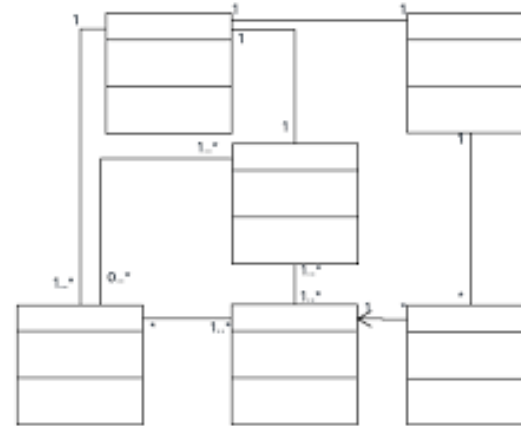


**Tables**



# RAPPEL

## Diagramme de classes



## 2 MODÈLES

### Modèle objet (représentation)

Application



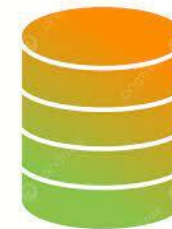
**Classes**



**Les données sont représentées  
sous forme d'objets**

### Modèle relationnel (représentation)

BD



MySQL

MariaDB

ORACLE

PostgreSQL

Microsoft  
SQL Server

**Tables**

**SGDBR**



**Les données sont représentées sous  
forme d'enregistrements**

# COMMUNICATION

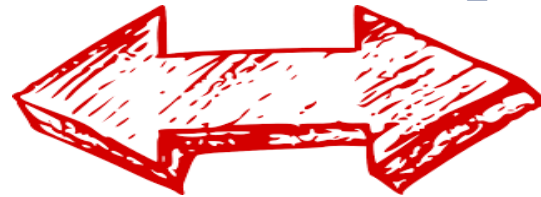
## Modèle objet

## Modèle relationnel



**Classes**

**Communiquer**



**Tables**

**Le client**



**Communication  
client/serveur**



**Le serveur**

# PDO

## Modèle objet

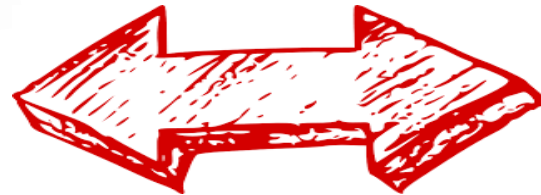
## Modèle relationnel



Classes



PDO (SQL)



Tables

Le client



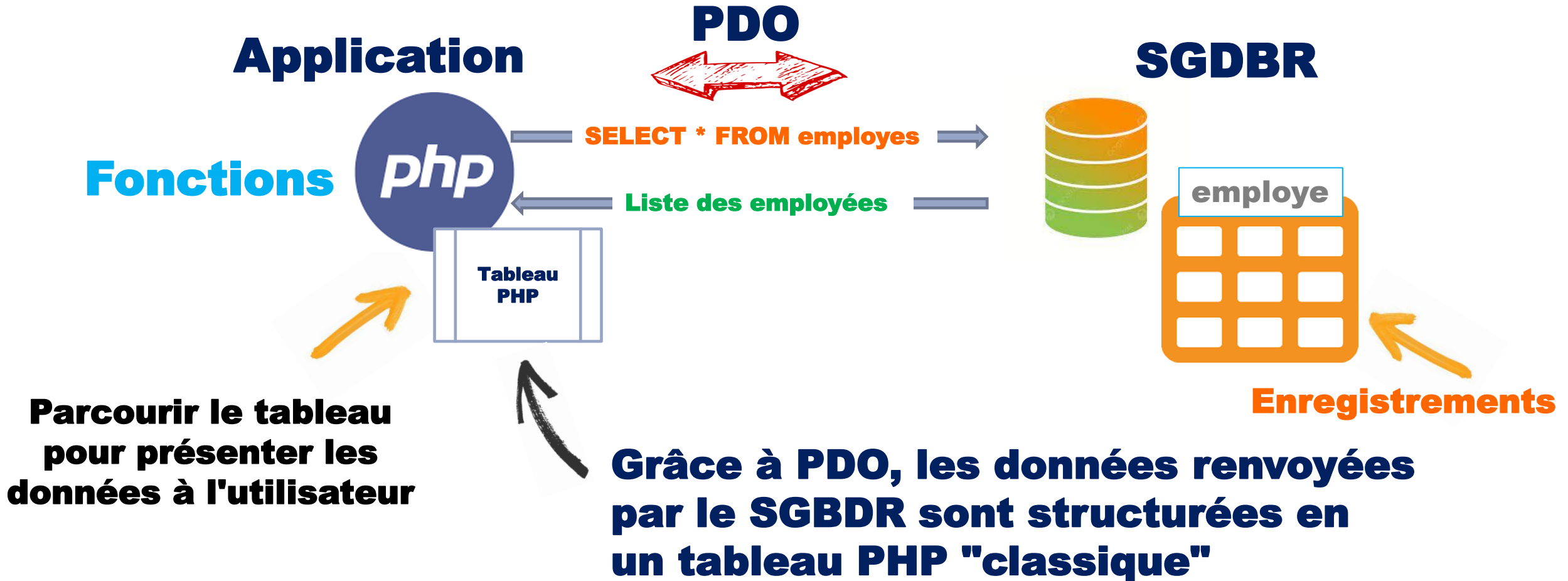
Communication  
client/serveur



Le serveur

# APPROCHE "NON OBJET"

## Modèle relationnel



# APPROCHE OBJET

→ Les données sont représentées de 2 manières différentes

## Modèle objet

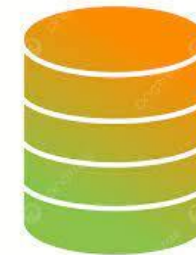
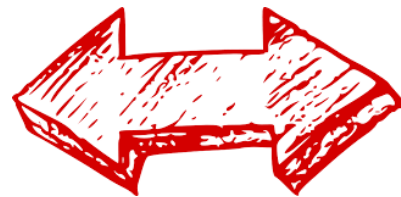
## Modèle relationnel

```
class ClasseA {  
    ...  
}  
class ClasseB {  
    ...  
}
```

**Classes**



**PDO**



**R1** [ a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub>, ..., #a<sub>N</sub> ]

**R2** [ a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub>, ..., #a<sub>N</sub> ]

**Tables**

**Objets** (instances)



**Enregistrements**



# APPROCHE OBJET

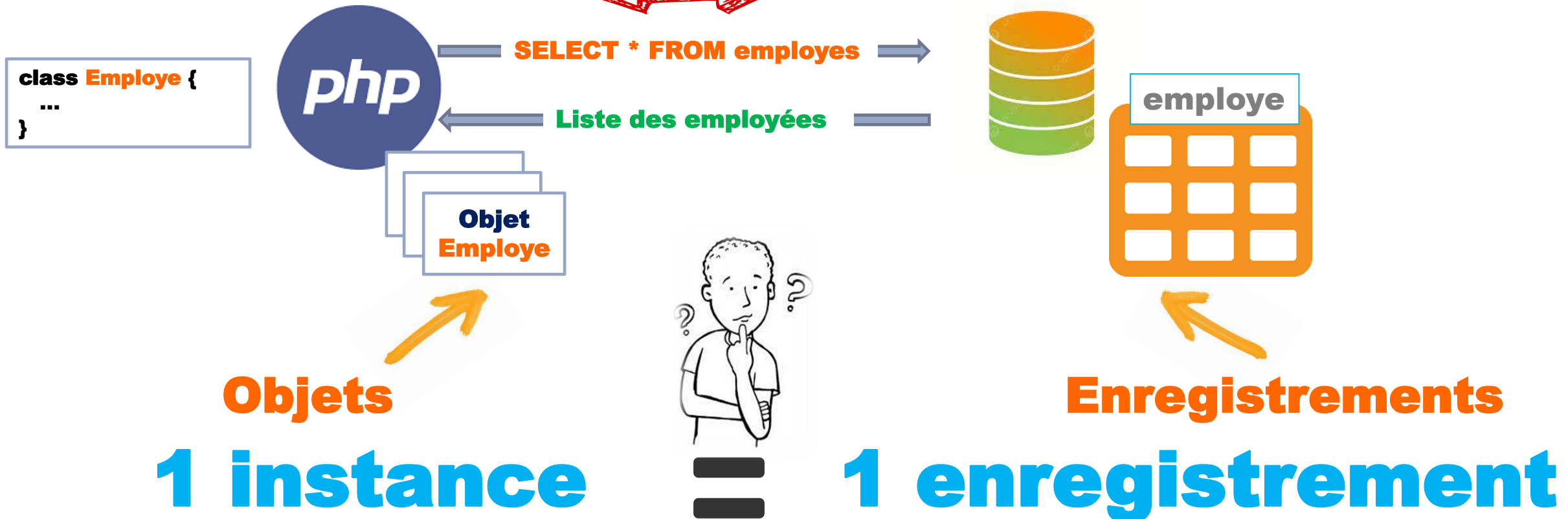
## Modèle objet

## Modèle relationnel

PDO

Application

SGDBR



# DAO

time to question

Comment faire **cohabiter** ces 2 représentations ?



Utiliser un **patron de conception**  
appelé **DAO**



**Patron de conception** : modèle permettant  
de concevoir une solution

**DAO** : **D**ata **A**ccess **O**bject

Modèle permettant de concevoir une solution d'accès  
à une BD depuis du code objet

# PRINCIPE DAO



**Créer une classe qui va faire le lien entre la base de données et le code de l'application PHP**



**Classe DAO**



**Proposer des méthodes permettant d'effectuer des requêtes SQL**



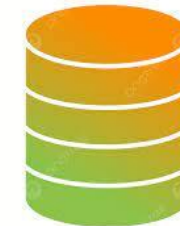
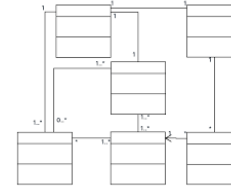
**Transformer les données reçues du SGBDR en objets utilisables dans l'application**

# MISE EN ŒUVRE

## Entité

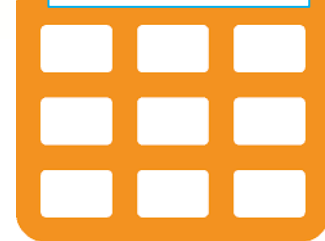
### Employe.php

```
class Employe {
    private int $matricule;
    private string $nom;
    private string $prenom;
    ...
}
```



matricule | nom | prenom

employe



### EmployeDAO.php

```
class EmployeDAO {
    public function creerEmploye(Employe $employe) {}
    public function rechercherEmployes() : array {}
    public function modifierEmploye(Employe $employe) {}
    public function supprimerEmploye(Employe $employe) {}
}
```

## Utiliser



PDO

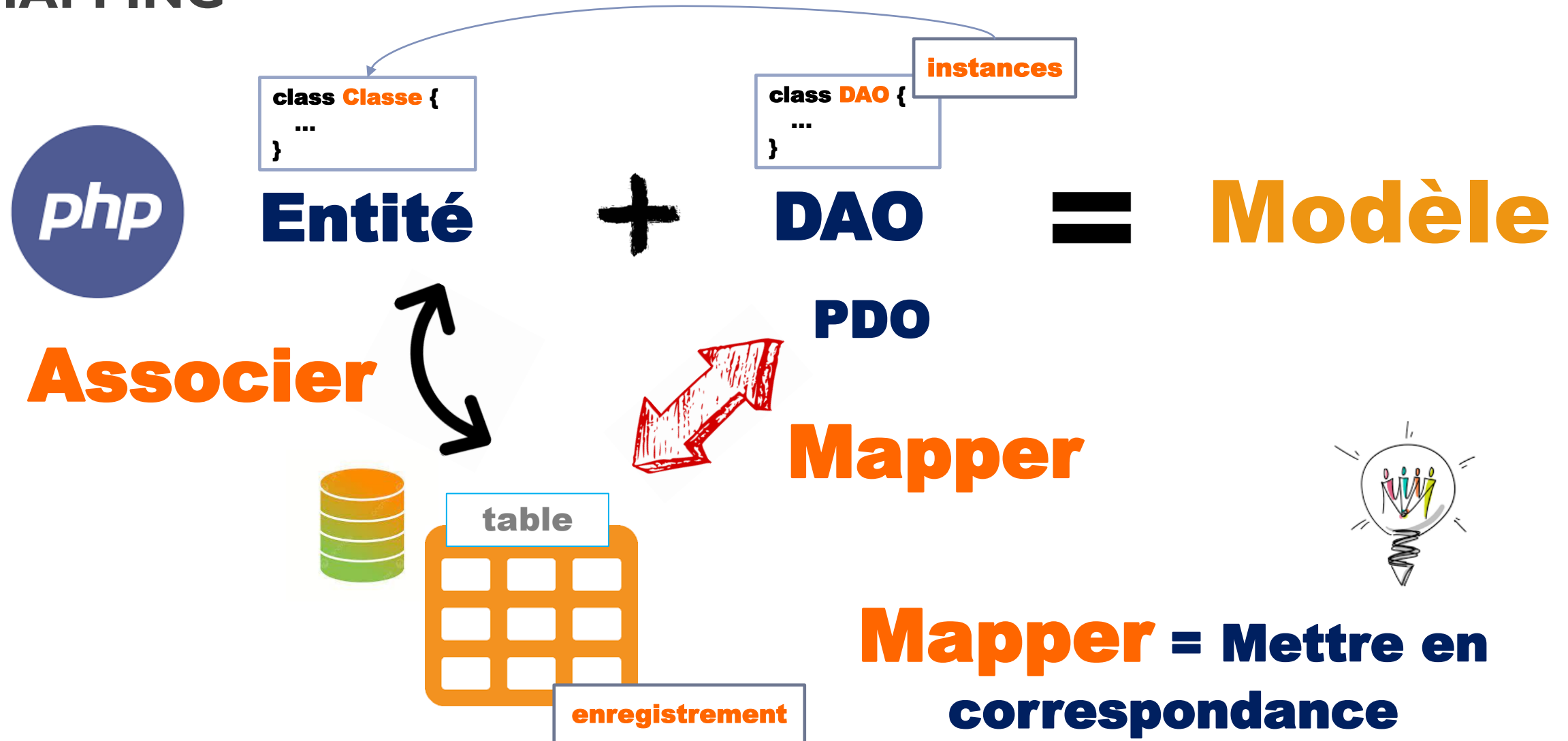
DAO



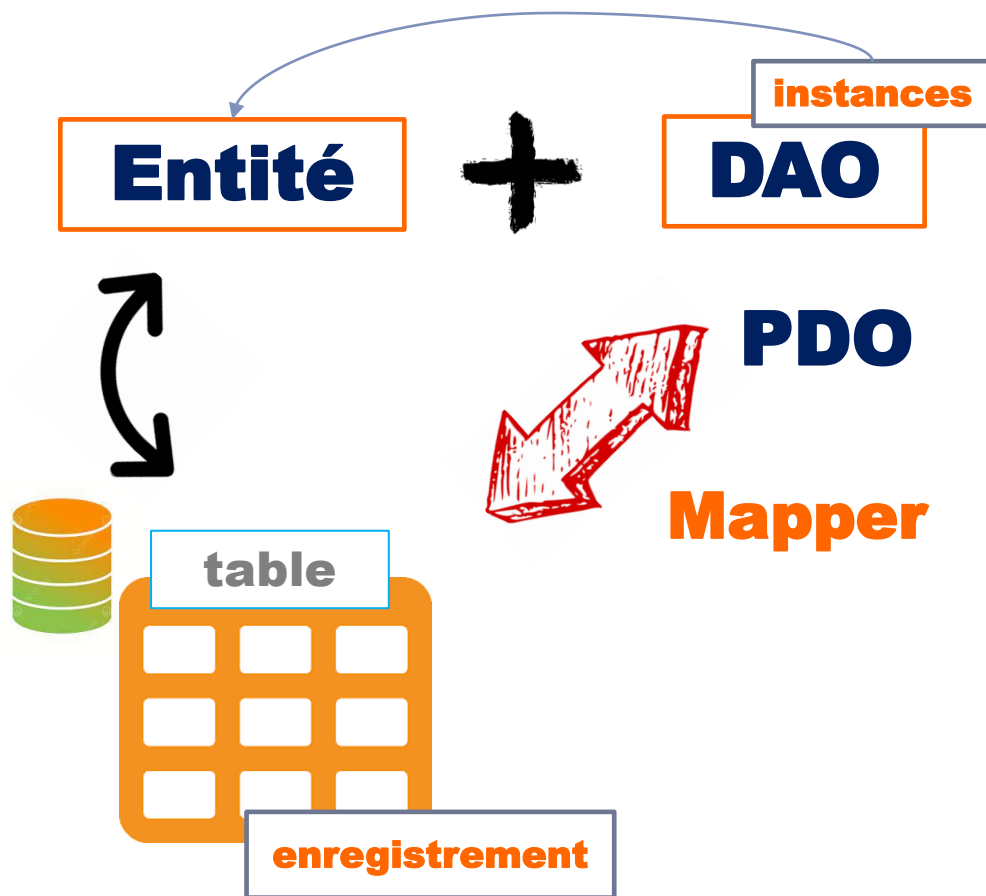
CREATE READ UPDATE DELETE

CRUD

# MAPPING



# SELECT



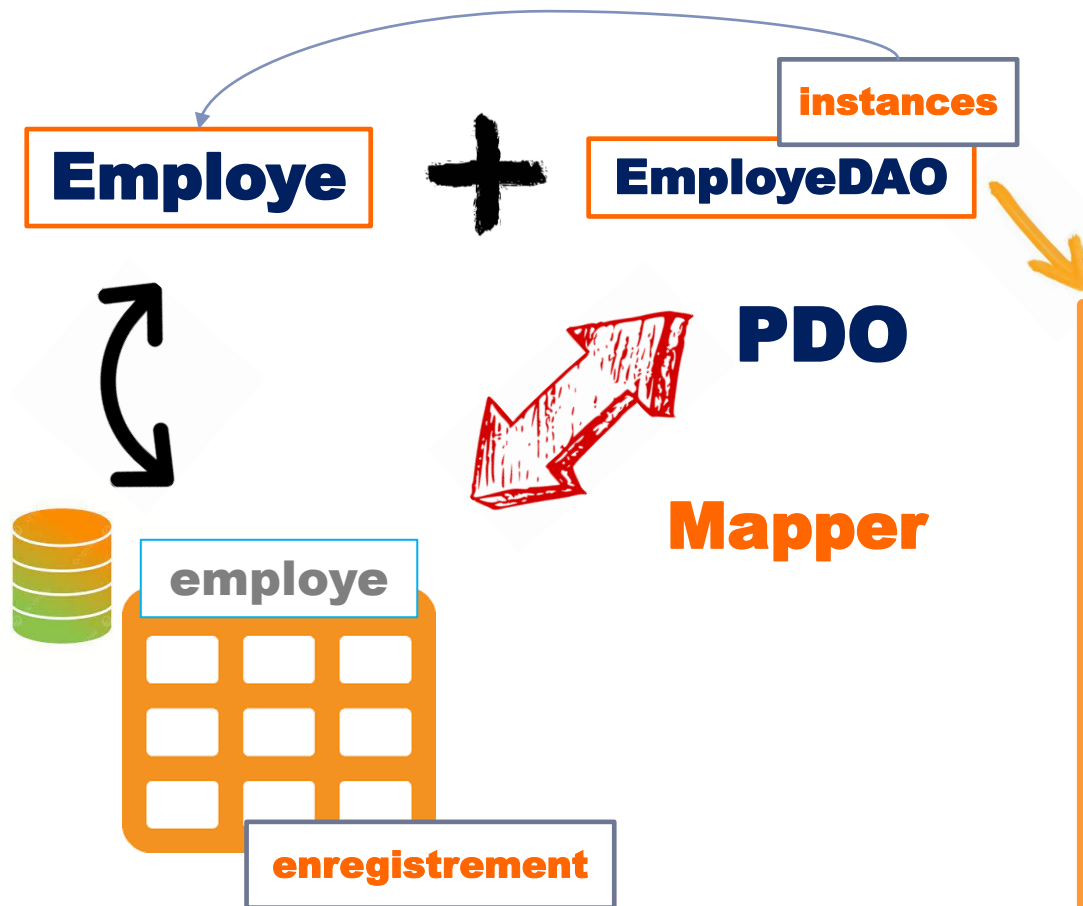
**Connecte** à la  en utilisant **PDO**

**Récupère** les **enregistrements**  
(requête **SELECT**)

**Convertit** les **enregistrements** en  
**instances**

**Retourne** les **instances**

SELECT



## Récupérer la liste des employés

**Connecte** à la  en utilisant **PDO**

**Récupère** les **enregistrements** (requête **SELECT**)

**Convertit** les **enregistrements** en **instances** de la **classe Employee**

**Retourne** les **instances**

## MAPPING



# Mapping Objet-Relationnel ORM



# LET'S GO!

