

# PDO

- **PHP Data Objects**
- É uma extensão que disponibiliza uma interface padronizada para trabalhar com BD
- Independente do BD utilizado os métodos executados serão os mesmos. Ao trocar o BD pode-se manter grande parte do código PHP
- Facilita a portabilidade, não garante (funções avançadas)
- Precisa de um driver entre o código e a BD.



# Conexão

```
<?php
function conectaBD
{
    $servername = "localhost";
    $username = "username";
    $password = "password";

    try {
        $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
        // setar PDO error mode para exception...setAttribute(variável,valor)
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        return $conn
    }
    catch(PDOException $e)
    {
        echo "Erro na Conexao: " . $e->getMessage();
    }
}
?>
```

# Inserir, deletar, atualizar

```
$con = conectaBD();  
$nome=$POST["nome"]; $sn=$POST["sn"];  
$sql = "INSERT INTO cliente (nome, sobrenome) VALUES ('?', '?)";  
$stm = $con->prepare($sql);  
$stm->bindValue(1,$nome);  
$stm->bindValue(2,$sn);  
$res = $stm → execute();  
//no caso de insert recupera o id criado no bd, opcional  
//$last_id = $conn → lastInsertId();  
echo "Inserido";  
//fecha a conexão e statment  
$stm->closeCursor();  
$stm=NULL;  
$con=null;
```

# Selecionar

```
$con = conectaBD();  
$sql = "SELECT cod, nome, snome FROM cliente";  
$stm = $con->prepare($sql);  
$tabela=$stm->execute();  
if($tabela){  
    //PDO::FETCH_ALL, PDO::FETCH_ARRAY  
    while($linha = $stm->fetch(PDO::FETCH_ASSOC)){  
        echo "Nome: ".$linha['nome']. " Sobrenome: ".$linha['sn'];  
    }  
}  
$stm->closeCursor();  
$stm=NULL;  
$con = NULL;
```