

PHP e PDO: Trabalhando com bancos de dados

- O PDO (PHP Data Object) é uma extensão da linguagem PHP para acesso a banco de dados.
- SQLite é um gerenciador de banco de dados que não precisa de um servidor
- Para buscar dados no banco, utilizamos o método query.
- Os diversos métodos para recuperar dados utilizando o PDO:
 - fetch
 - fetchObject
 - fetchColumn
 - fetchAll
- O PHP recomenda que você envie os dados como string (PDO::PARAM_STR), que é o tipo padrão, e o banco tratará esses tipos.
- Prepared Statements: previne SQL Injection e ajuda na performance.
- Podemos informar o tipo de dado que estamos passando através do PDO, utilizando o terceiro parâmetro de bindValue e bindParam.
- Exemplo de código para atualizar uma tabela

```
public function update(Student $student): bool
{
    //escreve a query e faz o prepare() por causa dos '?'
    $updateQuery = 'UPDATE students SET name = ?, birth_date = ? WHERE id = ?';
    $stmt = $this->connection->prepare($updateQuery);

    //define os valores de '?'
    $stmt->bindValue(1, $student->name());
    $stmt->bindValue(2, $student->birthDate()->format('Y-m-d'));
    $stmt->bindValue(3, $student->id());

    //executa o statement preparado
    return $stmt->execute();
}
```

- Se usar \$connection->beginTransaction(); precisa ter um \$connection->commit(); no fim, se não nada vai acontecer.
- O padrão Aggregate, onde o acesso a objetos é controlado por outros objetos (acesso a Telefone controlado por Aluno)

Dúvidas e assuntos a pesquisar

- [Common Entity Framework Problems: N + 1 - Brent Ozar Unlimited®](#)

- [DDD Aggregate \(martinfowler.com\)](http://martinfowler.com/DDD/Aggregate.html).