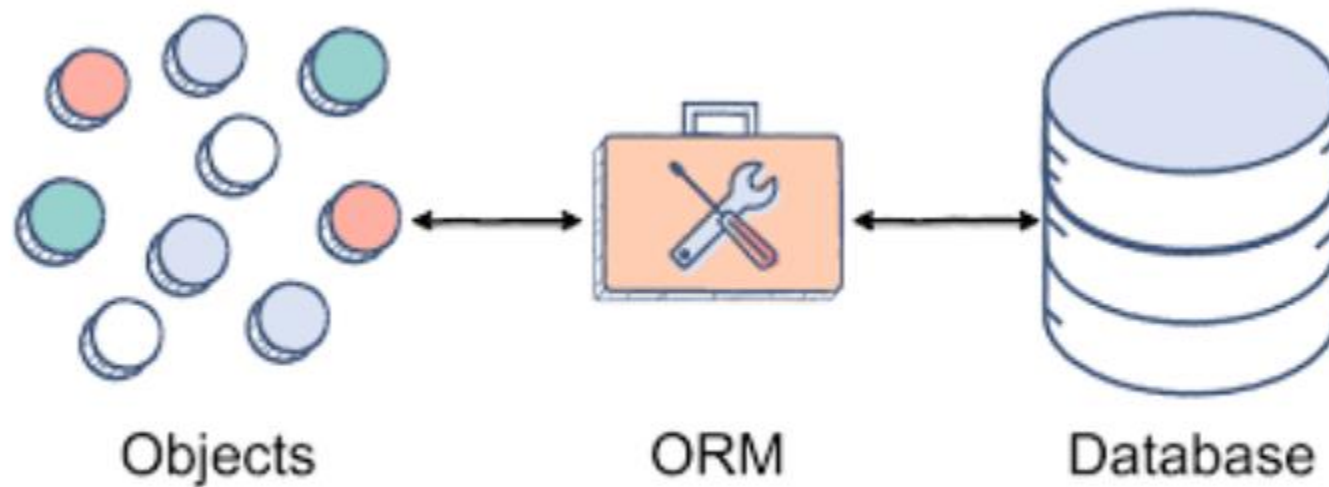


ORM

Mapeamento Relacional de Objetos

- Mapeamento Relacional de Objetos (ORM) é uma técnica usada para criar uma "ponte" entre programas orientados a objetos e, na maioria dos casos, [bancos de dados relacionais](#).



Como funciona:

- O ORM faz o mapeamento da sua classe para o banco de dados e cada ORM tem suas particularidades para gerar o SQL referente a inserção do objeto que corresponde a uma tabela no banco de dados e realizar a operação.



Como funciona:

- Ocorre conversão de classes para tabelas e vice-versa assim como as demais informações quando utilizamos um *framework* ORM em nossa aplicação. Isso facilita o desenvolvimento e evita a necessidade de *queries* (consultas) no código ou no banco de dados.
- Os ORMs usam dois padrões principais e diversas estratégias (que podem ser combinadas com um ou ambos os padrões) para mapear objetos de aplicativo para estruturas de banco de dados .

Padrão de registro ativo

Active record pattern

- Um padrão de registro ativo vincula um aplicativo a um esquema de banco de dados. Ele equipara tabelas a classes, linhas a objetos e colunas a atributos. Chaves estrangeiras (atributos que se referem a chaves primárias ou IDs de linha exclusivos em outra tabela) representam relacionamentos: elas servem como um equivalente a referências entre objetos no código.
- Cada classe fornece os métodos básicos para operações CRUD (criar, ler, atualizar, excluir), de modo que a classe é responsável por atualizar ou excluir a si mesma.

Padrão de registro ativo

Active record pattern

- Quando um aplicativo altera um objeto, o banco de dados atualiza automaticamente as tabelas correspondentes a essas alterações. E vice-versa, as alterações em um banco de dados são refletidas no código. Na verdade, não há uma camada adicional entre a lógica e o banco de dados. No entanto, um registro ativo ainda abstrai o mapeamento e gera SQL, que são características marcantes do ORM.

Padrão do mapeador Datta

Datta mapper pattern

- Um *Datta mapper pattern* separa classes e objetos da estrutura do banco de dados e não segue necessariamente a regra "um objeto = uma linha na tabela". Um objeto pode corresponder a várias tabelas, enquanto os programadores especificam como os objetos são serializados, desserializados e conectados ao banco de dados. Todas as operações do banco de dados são separadas dos objetos.
- Você pode trabalhar com coleções de tabelas e atualizá-las simultaneamente. Tudo isso dá aos programadores mais controle e flexibilidade, mas exige mais esforço.

Vantagens

- Aumento da produtividade.
- Menor risco de erros com SQL mal estruturado.
- Integração natural com POO.
- Facilita testes e manutenção

Desvantagens

- Menor controle sobre o SQL gerado.
- Pode gerar consultas ineficientes se mal configurado.
- Nem sempre é ideal para cenários altamente complexos.

Referências Bibliográficas

- [https://www.freecodecamp-org.translate.google.com/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/? x tr sl=en& x tr tl=pt& x tr hl=pt& x tr pto=tc](https://www.freecodecamp-org.translate.google.com/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/?x_tr_sl=en&x_tr_tl=pt&x_tr_hl=pt&x_tr_pto=tc)
- <https://renan04-marques.medium.com/oque-é-orm-3432889661f6>