

Aula 08 – Persistência de Dados

Cor verde: significa que está certo e o professor concorda com o trecho.

Cor amarela: correção.

Persistência de Dados: É a capacidade de manter um dado armazenado de forma durável.

Observação: Durante muito tempo, os bancos de dados relacionais foram amplamente utilizados — e ainda hoje são — em diversas aplicações. No entanto, com o passar do tempo, posteriormente a programação procedural, surge o paradigma da programação orientado a objetos. Nesse contexto, surgiram os bancos de dados orientados a objetos e o Mapeamento Objeto Relacional, que visava adaptar o modelo de persistência de dados ao novo paradigma, sem perder a possibilidade do uso do modelo relacional.

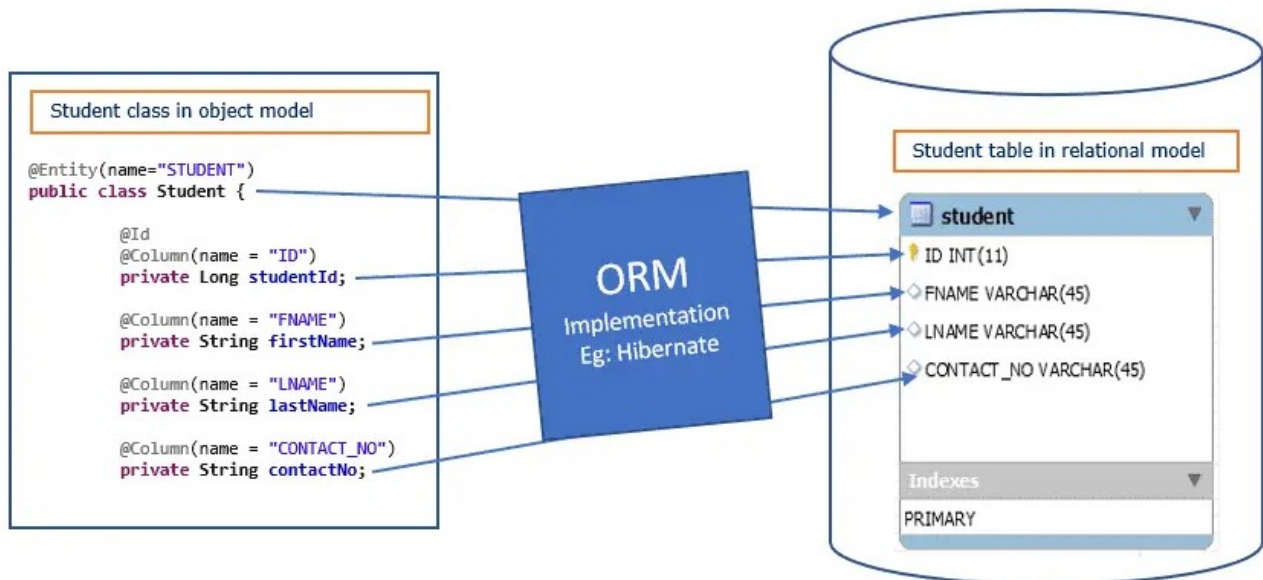
Relacional X Orientado a Objetos: Os bancos relacionais se utilizam de tabelas, onde os dados são comportados na estrutura de linha e coluna. No entanto, essa estrutura não é capaz de retratar a complexidade do paradigma orientado a objetos, sendo necessário a abordagem ORM. Por outro lado, temos a estrutura de bancos orientada a objetos, mais recente, que além do armazenamento de dados possui métodos e estruturas de relacionamentos complexos, inerentes aos objetos.

NoSQL: É uma abordagem utilizada para se referir a uma série de bancos que não utilizam o paradigma relacional, ou seja, a clássica estrutura rígida organizada em linha/coluna, normalizações e etc. Atualmente, o NoSQL é amplamente utilizado em sistemas que analisam/armazenam um grande volume de dados, a exemplo de big data, mineração de dados, inteligência artificial, já que uma abordagem mais flexível em comparação ao paradigma relacional. Alguns tipos de NoSql são: colunar, chave-valor, grafos e documentos.

Persistência Poliglota: É ideia de empregar diferentes abordagens de persistência de dados em um mesmo sistema, ou seja, usar o relacional, o não relacional e o orientado a objetos. Esse conceito de persistência de poliglota segue a mesma ideia dos padrões de arquitetura, onde, atualmente, são usados diversos padrões em uma mesma aplicação.

Mapeamento Objeto relacional (ORM): É uma abordagem que consiste em traduzir o modelo de classes (ou objetos) para o modelo relacional dos bancos de dados, ou vice-versa. Afinal, o paradigma relacional usa os relacionamentos baseados em chaves, os objetos, por sua vez, se utilizam do conceito de referências (ponteiros). Além disso, há diversas formas de efetuar esse mapeamento.

Representação:



Técnicas ORM – Foreign Key Mapping: No Mapeamento Objeto-Relacional, FKM representa um tipo de mapeamento (associação/composição/agregação) onde há um atributo que possui uma referência para outro objeto.

Associação: é um conceito em programação orientada a objetos onde um atributo possui uma referência para outro objeto. Na associação, os objetos envolvidos podem se comunicar e interagir, mas não precisam ser parte um do outro. *Por exemplo, um “cliente” pode ter vários “pedidos” e um “pedido” pode ser feito por um único “cliente”, mas ambos podem existir independentemente um do outro.*

Nota do professor : Isso, quando você cria um cliente não faz sentido criar junto seu pedido (new Pedido()) no construtor da classe Cliente). Da mesma forma, não faz sentido criar o cliente ao criar um pedido (new Cliente()) no construtor da classe Pedido) (o cliente deve ser criado antes, para depois ser associado ao pedido).

Composição: é um conceito em programação orientada a objetos em que um (objeto) é parte integrante do outro. Ou seja, um objeto composto não pode existir sem um objeto que o contenha. Mais ou menos assim : temos dois objetos, funcionário e dependentes. Se, por algum motivo, o funcionário for demitido, automaticamente, na composição, os dependentes terão que serem excluídos. Afinal, não há motivos para existir dependentes sem um objeto "funcionário".

Agregação: Na agregação, o objeto agregador (pai) contém ou utiliza o objeto agregado(filho), mas o objeto agregado pode existir sem o objeto agregador. Logo, a agregação é uma forma de relação mais fraca entre objetos do que a composição, onde um objeto pode ser parte do outro objeto, mas não é fundamental para a sua existência.

Técnica Lazy Initialization: é uma estratégia útil para adiar a inicialização de objetos até o momento em que são realmente necessários, economizando recursos computacionais e melhorando o desempenho da aplicação. Essa técnica utiliza métodos interceptadores, que executa automaticamente uma função quando uma propriedade não existente no objeto é requisitada.