Ao trabalharmos com tabelas em bancos de dados, as chaves primárias são essenciais. Geralmente, criamos um identificador único ou ID para representar essas chaves da melhor forma possível.

As chaves primárias servirão para que cada entidade seja salva de forma única. Se quando trabalhamos com Java a JVM se encarrega de separar cada objeto em um lugar da memória, no banco de dados isso precisa ser configurado. E utilizar um ID costuma ser uma boa forma de fazer isso.

Temos várias formas de gerar IDs no banco de dados, e a JPA também nos ajuda com isso. Basta utilizar a anotação GeneratedValue e escolher a estratégia de geração do ID. Posteriormente vamos falar sobre essas diversas estratégias existentes, ok?!

Confira:

**Estratégia AUTO**

A estratégia AUTO é a estratégia padrão de geração de ID JPA. Quando esta estratégia é utilizada, é a JPA que determina a melhor estratégia a ser usada, baseando-se no banco de dados específico que você está utilizando.

Aqui está um exemplo de como implementar a estratégia AUTO em um código:

@Entity

**public** **class** **Entidade** {

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

**private** Long id;

}

COPIAR CÓDIGO

Esta estratégia é bem prática, pois a JPA se encarrega de toda a lógica de geração de ID. No entanto, pode ser pouco flexível se você precisar de um maior controle sobre a geração de ID.

**Estratégia SEQUENCE**

A estratégia SEQUENCE utiliza um banco de dados de sequência para gerar IDs. Com essa estratégia, a sequência é incrementada cada vez que um ID é gerado, garantindo que cada ID seja único.

Aqui está um exemplo de como implementar a estratégia SEQUENCE:

@Entity

**public** **class** **Entidade** {

@Id

@GeneratedValue(strategy=GenerationType.SEQUENCE)

**private** Long id;

}

COPIAR CÓDIGO

Essa estratégia é particularmente útil para bancos de dados que suportam sequências, pois é muito eficiente e evita a possibilidade de colisões de ID.

**Estratégia IDENTITY**

A estratégia IDENTITY faz uso de colunas de auto incremento disponibilizadas por alguns bancos de dados. Cada vez que uma nova entidade é inserida no banco de dados, a coluna de ID auto incrementada é atualizada para gerar um novo ID.

Exemplo de implementação desta estratégia:

@Entity

**public** **class** **Entidade** {

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

**private** Long id;

}

COPIAR CÓDIGO

Esta estratégia é útil se você estiver usando um banco de dados que suporta colunas de auto incremento, como MySQL ou SQL Server.

**Estratégia TABLE**

Por último, temos a estratégia TABLE. Essa estratégia utiliza uma tabela de banco de dados separada para gerar IDs. Ela é menos eficiente do que as outras estratégias, mas tem a vantagem de ser portátil entre diferentes provedores de banco de dados.

Exemplo de implementação desta estratégia:

@Entity

**public** **class** **Entidade** {

@Id

@GeneratedValue(strategy=GenerationType.TABLE)

**private** Long id;

}

COPIAR CÓDIGO

Compreender essas estratégias e saber quando usar cada uma delas permitirá que você use a JPA de maneira mais eficiente, garantindo que seus dados sejam persistidos de maneira confiável e segura. Lembre-se, a escolha da estratégia de ID correta pode ter um grande impacto na eficiência da sua aplicação!