

Ficou curioso?

Os tipos de dados apresentados na aula podem ser encontrados em meio ao portfólio da AWS, o relacional é coberto pelo Amazon RDS (*Relational Database System*), o não relacional de nome DynamoDB e o sistema de arquivos data storage sendo o S3. Como já dito, cada um tem um uso para o qual é mais indicado, cabe a quem trabalha com engenharia de dados definir no escopo.

Trabalhar com bancos é algo realmente interessante. Levando em conta os bancos relacionais por exemplo, as soluções podem variar de acordo com o processo em si e também da forma como são consumidas. Explicando um pouco melhor, vemos que nosso exemplo da clínica que criamos uma tabela apenas para pacientes e uma para médicos além de uma apenas para fazer a relação entre as duas. Por que?

No banco relacional, e não só nele, quanto menos dados temos que percorrer, melhor, além de que quanto maior a massa dessas informações, maior será o consumo de armazenamento, por isso, o fato de registrarmos dados do cliente apenas uma vez em uma tabela e nas outras usar apenas um número que o referencia é um ótimo começo. Bom, para um cliente talvez o ganho não seja tão grande, mas imagina um médico que faça dez consultas três vezes por semana? Ter que repetir seu telefone em todas linhas de uma tabela unificada é um desperdício.

Além disso, esse tipo de otimização permite uma manutenção muito mais fácil. Como a consulta a esse tipo de banco acontece por referência, atualizar os dados do médico e do paciente só precisa ser feito em apenas um local, sendo essa vantagem do relacional. Se esse procedimento tivesse que ser feito em um não relacional, todas chaves teriam que ser percorridas.

Sugiro que você pesquise também por índices em bancos de dados relacionais, eles são um artifício para agilizar e localização de valores de uma coluna em específico. A título de exemplo (aqui vemos a importância de se conhecer a utilização): se a principal ferramenta da nossa clínica é lista todas as consultas diárias, talvez seja uma boa ideia indexarmos por data.

Mas só para lembrar! O relacionamento dos dados tem um custo, trocar or ou excluir tabelas podem gerar dor de cabeça por que podem gerar inconsistências no banco, como o mesmo procurar a referência de tabelas de e exames e a mesma ter sido excluída.

Já para o não relacional tivemos o exemplo de ter dados que não tem uma forma fixa. Esse tipo de característica, como podemos inferir, é feita em troca da relação entre as tabelas, afinal, não sabemos se a chave do relacionamento se encontra lá. Esse banco também é útil para trazer blocos de informações, por exemplo, se quiséssemos pegar os dados de um exame, bastaria passar sua "chave" e ele seria retornado para nós. O mesmo se aplica para *datastores*, porém ao invés de texto, poderíamos retornar a imagem desse exame.

Espero que vocês tenham gostado do curso! Fiz essa seção para passar como o guia de onde pode-se ir para continuar aprendendo.

Abraço!