Curso Banco de Dados: Essencial – Geek University

**Seção 2: Introdução a Banco de Dados**

**1. Dado e Informação.**

Dado -> Elemento bruto geralmente proveniente de uma entrada do usuário ou de outros sistemas.

Informação -> Elemento processado (lapidado) constituído de dados.

**Banco de Dados Relacional x Não Relacional**

Bancos de dados Relacionais são conhecidos como SQL e os Não Relacionais são conhecidos como NoSQL.

SQL significa Structure Query Language – Linguagem de Consulta Estruturada.

NoSQL – Not Only SQL

**Banco de Dados Relacionais.**

Conceito criado por Edgar Frank Codd em 1970, sendo descrito no artigo “Relational Modelo f Data for Large Shared Data Banks”.

Baseado no Modelo Relacional, onde os dados estão guardados entre tabelas que por sua vez podem ou não estar relacionados.

**\*Pontos fortes:**

Os SGBDRs – Sistemas Gerenciadores de Bancos de Dados, conforme são chamados, oferecem aos usuários processos de validação, verificação e garantias de integridade dos dados. Além disso, oferecem controle de concorrência, recuperação de falhas de segurança, controle de transações, otimização de consultas, dentre outros.

**\*Pontos Fracos:**

Dificuldade em conciliar o modelo relacional com a demanda por escalabilidade cada vez mais frequente.

Dificuldade em se organizar os dados em um sistema distribuído trabalhando com particionamento de dados.

Escalabilidade = ação de elevar a quantidade de acessos simultâneos ao banco de dados.

**Banco de Dados Não Relacional**

O termo NoSQL surgiu em 1998, a partir de uma solução de banco de dados que não oferecia uma interface SQL, mas ainda assim estava baseado inicialmente na arquitetura relacional.

Posteriormente, o termo passou a representar soluções que promoviam uma alternativa ao modelo relacional, tornando-se uma abreviação de Not Only SQL – Não apenas SQL

*O grande diferencial aqui é a performance, pois por não utilizar um modelo estruturado em tabelas, consegue-se trabalhar com os dados de forma muito mais rápida.*

**Características em comum:**

Como são livres do modelo relacional, promovem alta disponibilidade e maior escalabilidade.

**Algumas implementações oferecem:**

- O particionamento e a replicação de dados;

- Sistemas baseados em armazenamento de chave/valor;

- Sistemas orientados a documentos

- Sistemas orientados a doluna;

- Sistemas baseados em grafos

- etc

**Modelagem Conceitual, Logica e Física**

Os modelos de banco de dados são usados para descrever, mais detalhadamente, a estrutura de um banco de dados. Eles servem então como parte importante da documentação dos sistemas que auxiliam não somente os desenvolvedores que estão trabalhando no projeto mas também servem como documentação que poder ser entregue ao cliente que contratou o serviço de desenvolvimento.

**Atenção:** Um modelo de dados não informa quais dados estão armazenados em um banco de dados, mas sim apenas quais e que tipos de dados contém.

Estes modelos são baseados em três níveis: **Conceitual, logico e físico.**

*Quando entramos nesses aspectos de modelagem logica/física/conceitual, nos ainda não estamos tratando dos dados em si, e sim da estrutura do banco de dados.*

**Modelo conceitual:** o nível conceitual é desenvolvido com alto nível de abstração, a partir dos requisitos do sistema, extraídos na fase de levantamento de requisitos pelos analistas de sistemas. Esse modelo pode ser elaborado de forma textual ou por meio de diagramas: Diagrama de Entidade e Relacionamento.

**Modelo Logico:** este modelo descreve com os dados serão armazenados no banco de dados e também seus relacionamentos. Neste modelo podemos definir também qual sera a tecnologia utilizada para armazenagem dos dados.

**Modelo Físico:** Também chamado de Modelo de Implementação, descreve, por meio de alguma linguagem (comumente SQL), como será feita a armazenagem do banco.

Neste nível se escolhe qual SGBD será utilizado, levando em consideração o modelo lógico adotado.

**Outros conceitos essenciais:**

**Entidade:** Objeto ou evento no mundo real sobre o qual desejamos manter registros em um banco de dados.

Ex: Aluno, Carro, Produto, Vendedor, Venda, etc.

*Entidade == tabela*

**Atributo:** Propriedade ou característica que descreve uma entidade. São as colunas (campos) das tabelas que irão armazenar os dados.

**Atributo Chave:** Também conhecido como chave primária, primary key ou pk, é um campo que deve possuir **valor único** (sem repetição) em todo o conjunto de dados da entidade. Este atributo **é usado para identificar unicamente** um registro na tabela.

**Relacionamentos:** Geralmente as entidades nunca estão sozinhas; normalmente estão associadas com outras entidades. Reconhecer e registrar os relacionamentos entre entidades fornece uma descrição muito mais rica do modelo.

Um relacionamento pode acontecer entre uma ou varias entidades.

**Chave estrangeira:** Também conhecido como foreign key ou fk, é um atributo presente em uma entidade que indica um relacionamento e representa a chave primária de uma outra entidade

**Grau de relacionamento:**

**Relacionamento Unário:** chamado também de grau 1 – uma entidade se relaciona com ela mesma. Ex: funcionário é gerenciado por outro funcionário.

**Relacionamento binário:** grau 2 – Uma entidade se relaciona com outra entidade. Ex: vendedor – vende – produto

**Relacionamento ternário:** chamado também de grau 3: Três entidades estão relacionadas por um mesmo relacionamento.

**Cardinalidade (Máxima):** Define a quantidade máxima de ocorrências de uma entidade que poderá estar associada a outra entidade.

Por exemplo:

- Um vendedor pode vender apenas um tipo de produto? Ou dois? Ou três?

- Um produto pode ser vendido por apenas um vendedor? Ou por todos?

***Relacionamento Binário um-pra-um*** *(****1:1) one-to-one*** *–* Indica que uma ocorrência da entidade A pode se relacionar exclusivamente com uma ocorrência da entidade B e vice-versa.

Ex: Um vendedor ocupa um único escritório e um escritório pode ser ocupado por um único vendedor.

***Relacionamento binário um-pra-muitos (1:n) one-to-many:*** Indica que uma ocorrência A pode se relacionar com várias ocorrências da entidade B, porém o inverso não é permitido.

Ex: Um vendedor atende muitos clientes, porém cada cliente é atendido por apenas um vendedor específico.

***Relacionamento binário Muitos-pra-Muitos (n:m) many-to-many –*** Indica que uma ocorrência da entidade A pode se relacionar com várias ocorrências da entidade B e vice versa.

**Normalização de Dados.**

O processo de normalização aplica uma série de regras sobre as tabelas de um banco de dados para verificar se estas estão corretamente projetadas.

**Primeira Forma Normal –** Uma Entidade estará na primeira forma normal (IFN) se todos os campos forem atômicos (simples) e não multivalorados (com múltiplos valores).

**Segunda Forma Normal –** Uma entidade estará na 2FN se ela já se encontrar na 1FN e todos os atributos não chave forem totalmente referentes a chave primária.

**Terceira Forma Normal –** Cada uma das formas normais tende a ir rtefinando a modelagem e deixando a estrutura de dados mais integra e exclusiva, evitando repetições desnecessárias e possíveis sobrecarga no gerenciador de banco de dados.

Uma tabela estará na 3FN se ela estiver na 2FN e se nenhuma coluna não-chave depender de outra coluna não-chave. Ou seja, na 3FN temos simplesmente que eliminar os campos que podem ser obtidos pela equação de outros campos da mesma tabela.

**MER – Modelo Entidade-Relacionamento**