Módulo 1 – Introdução à Banco de Dados

Aula 02

Banco de dados é um conjunto de informações organizadas que podem estar em um sistema manual ou computadorizado.

O banco de dados é apenas um **arquivo físico**, que é gerenciado pelo SGBD (Sistema de Gerenciamento de Banco de Dados).

Modelo Conceitual

Representa ou descreve, de forma simples, a realidade de algum problema, negócio ou ambiente de modo a ser facilmente compreendido pelo usuário final. É um modelo com um desenho muito geral e abstrato, cujo objetivo é explicar a visão geral do negócio ou sistema.

Modelo Lógico

O modelo lógico é uma versão completo que inclui todos os detalhes sobre os dados. Explica quais dados são importantes, sua semântica, relacionamentos e restrições. Dentro da modelagem de dados para banco de dados, o modelo lógico serve para compilar os requisitos do negócio e representar as entidades como um modelo preliminar, que servirá para depois fazer o modelo final do banco de dados (modelo físico). Explica “**O que**”.

* Compila os requisitos de negócio em um modelo
* Define claramente as entidades do banco de dados
* Define relacionamentos
* Não está atrelado a um SGBD (MySQL, SQLServer)

Modelo Físico

É um modelo que implementa o modelo lógico. O modelo físico é um guia para o design do banco de dados real com base nos requisitos reunidos que constam no modelo lógico. Na modelagem física, é definido como será o esquema de banco de dados, e neste esquema pode ser definido a junção das entidades e suas relações, ou seja, é um modelo real de nosso banco.

Nesta etapa, são definidas as tabelas do banco e suas relações, são também definidas suas chaves primárias e suas chaves estrangeiras e as restrições com relação ao banco de dados. Os índices são definidos também nesta etapa.

É um esquema a ser implementado em um sistema de gerenciamento de dados. Explica “**como**”.

* Modelo do banco de dados com base no modelo lógico
* Define tabelas e colunas com base nas entidades e seus atributos
* Cria relacionamento entre as tabelas, chaves primárias e estrangeiras
* Está diretamente atrelado a um SGBD (MySQL, SQLServer)

Aula 03

Por que fazemos modelagem?

Para validar as interpretações que fazemos da realidade. Ou seja, para confirmar que realmente entendemos os conceitos mais importantes que devem ser armazenados. E também serve como **documentação** para trabalhar em equipe e entender os requisitos.

Quais elementos fazem parte da modelagem?

Entidades / Atributos / Chaves / Relações / Diagrama Entidade Relacionamento.

Tipo de Dados (Aula 05 – 17/08)

Dados do tipo números sem decimais

**TINYINT** => vai de -128 a 128 0 a 255

**SMALLINT** => vai de –32768 a 32767, 0 a 65535

**MEDIUMINT** => vai de -2147483648 a 2147483648, 0 a 4294967295

**BIGINT** => vai de -9223372036854775808 a 9223372036854775807, 0 a 18446744073709551615

Dados do tipo números com decimais

**FLOAT** => -3.402823466E+38 a -1.175494351E-38, 0, e de 1.175494351E-38 a 3.402823466E+38

**DOUBLE** => -1.7976931348623157E+308 to -2.2250738585072014E-308, 0, and 2.2250738585072014E-308 to 1.7976931348623157E+308.

**DECIMAL** => X indica a quantidade total de dígitos (precisão) e Y indica quantos deles estarão “depois da vírgula”;

O separador de decimais é o ponto (.);

X = Máximo de 65 casas;

Y = Máximo de 30 casas;

Booleanos

O MySQL salva os números booleanos por trás, como o zero ou como o um. Por motivos de desempenho, recomendamos não utilizar estes tipos de dados no MySQL.

Caso queira salvar valores "verdadeiros" e "falsos", podemos usar o tipo de dado tinyint e usar o 0 representando como false, e o 1 para representar true.

Dados do tipo texto

**CHAR (num)**

O número em parênteses indica a quantidade exata de caracteres. Exemplo: char (100) → 100 caracteres **sempre.**

**VARCHAR (num)**  
O número entre parênteses indica a quantidade máxima de caracteres. Exemplo: varchar (100) → 100 caracteres como máximo.

**TEXT**

Determina um dado do tipo texto sem limite de caracteres. Geralmente usado em postagens para blogs por exemplo.

Dados do tipo datas

**DATE** => Armazena as datas somente no formato YYYY-MM-DD.

**TIME** => Armazena as horas somente no formato HH:MM:SS.

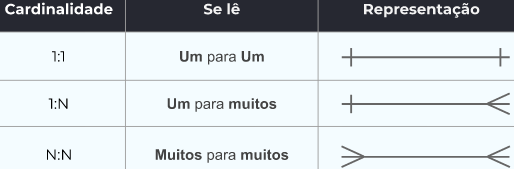
**DATETIME** => Corresponde a uma representação completo de data e hora, por exemplo: 3 de janeiro de 1967 às 08:00, que é armazena da seguinte forma: **1967-01-03 08:00:00**

Aula 07 (Relacionamento)

**Relacionamentos** indicam como duas tabelas devem ser relacionadas. Em um banco de dados, existem 3 tipos de relacionamento:

* Um a Um (1 x 1)
* Um para Muitos (1 x N)
* Muitos para Muitos (N x N)

**Cardinalidade** é a forma como as entidades se relacionam.



Um para Um (1 : 1)

Um usuário possui apenas um endereço. Um endereço pertence a apenas um usuário.

Para estabelecer a relação, colocamos a chave **primária do endereço** na tabela de usuários, indicando que este endereço está **associado** a esse **usuário (Chave Estrangeira).**



Um para muitos (1 : N)

Um usuário possui apenas um endereço. Um endereço pertence a apenas um usuário.

Para estabelecer a relação, colocamos a chave primária do endereço na tabela de usuários, indicando que este endereço está associado a esse usuário (Chave Estrangeira).



Muitos para muitos (N : N)

Um cliente pode comprar muitos produtos. Um produto pode ser comprado por vários clientes. Em relacionamentos N: N, no banco de dados, o próprio relacionamento se torna uma tabela.

Essa tabela intermediária - também conhecida como **tabela dinâmica** - pode conter 3 informações: uma chave primária (PK) e duas chaves estrangeiras (FK), cada uma referindo-se a cada **tabela no relacionamento**.



Com isso, no relacionamento de N:N, é necessário ter a tabelo dinâmica para fazer a associação das 2 tabelas.