**Curso Boson Treinamento**

**Modelagem de Dados**

**Níveis da Modelagem de Dados;**

\*Modelo Conceitual(alto nível) – MCD

\*Modelo Lógico – MLD

\*Modelo Físico(baixo nível) – MFD

**Modelo Conceitual**

Primeira Fase da modelagem, onde representaremos o mundo real por meio de uma visão simplificada dos dados e seus relacionamentos. Assim poderemos determinar quais informações serão armazenadas do BD.

Esse nível é independente do SGBD

Exemplo: Cadastro de produtos em uma loja

Dados necessário: Nome do produto, categoria de produto (limpeza, higiene, etc), código do fornecedor, tipo de de embalagem, tamanho, quantidade.

Modelo Conceitual = Neste nível, detalhes da implementação não aparecem, porém é suficientemente detalhado para o ponto de ser possível descrever os tipos de dados requeridos, seus relacionamentos entre si e regras de consistência.

**Modelo Lógico**

\*Possui conceitos que os usuários são capazes de entender, ao mesmo tempo em que não está distante do modelo físico do banco de dados.

\*Neste nível o projeto é independente de SGBD.

\*Consiste na especificação lógica dos dados em um formato adequado ao SGBD escolhido. Os tipos de dados são completamente definidos.

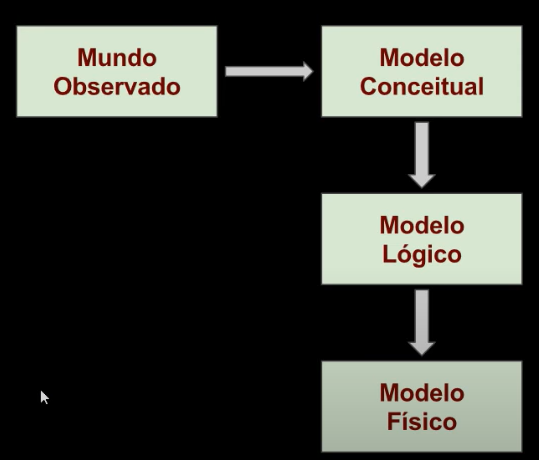
**Modelo Físico**



\*A partir de um modelo lógico nós derivamos o modelo físico, onde se detalham os componentes de estrutura física do banco de dados, incluindo as tabelas, campos, tipos de valores, restrições, etc.

\*Ao criarmos o modelo físico, poderemos partir para a implementação física do Banco de dados, utilizando o SGBD mais adequado.

**Arquitetura de Três Níveis**

****

**Esquema do Banco de Dados**

É uma definição do Banco de Dados especificada durante o projeto armazenada no Dicionário de Dados.

Um esquema (Schema) raramente muda durante a vida do BD.

Trata-se da organização dos dados em um plano que mostra como o bancp é construído.

O esquema define tabelas, campos, relacionamentos, visões, funções e muitos outros elementos que compõem o BD.

**Etapas do desenvolvimento de um BD**

As principais etapas no desenvolvimento de um BD são:

1.Especificação e Análise de Requisitos

a)os requisitos são documentados

2.Projeto Conceitual

a)Baseado nos requisitos

3.Projeto Lógico

a)Expresso em um modelo de dados, como o relacional

4.Projeto Físico

a)Especificações para armazenar e acessar o banco de Dados

b)Implementação do BD, inserção de dados reais e manutenção

**Tarefas para Modelagem**

As tarefas a seguir devem ser realizadas para que seja possível efetuar modelagem de dados e projeto de BD funcional:

\*Identificar os tipos de entidade

\*Identificar atributos

\*Identificar relacionamentos

\*Criar e associar chaves

\*Normalizar para reduzir redundância

\*Desnormalizar para aumentar performance

**Chaves**

Chave consiste em uma ou mais colunas de uma relação cujos valores são usados para identificar de forma exclusiva uma linha ou conjunto de linha.

Pode ser única (identifica uma única linha) ou não-única (identifica um conjunto de linhas).

Únicas(Unique): Candidata, Composta, Primária, Surrogada

Não-Única (Non-Unique): Estrangeira.

**Chave Candidata**

Atributo ou grupo de atributos com o potencial para se tornarem uma chave primária;

Uma chave candidata que não seja usada como chave primária será conhecida como **Chave Alternativa**.

Exemplo: Campos Num\_Matrícula e CPF em uma tabela de registro de alunos

**Chave Primária**

Chave principal da Relação , identifica de forma exclusiva os registros em uma tabela.

Primary Key / PK

**Chave Estrangeira**

Coluna de uma tabela que estabelece um Relacionamento com a Chave Primária (Pk) de outra tabela.

É a partir da chave estrangeira (Foreing Key/FK) que sabemos com qual registro em outra tabela um registro está relacionado.

**Chave composta**

Chave que é composta de dois ou mais atributos (colunas)

Geralmente empregada quando não é possível utilizar uma única coluna de uma tabela para identificar de forma exclusiva seus registros.

**Chave Surrogada / Substituta**

Valor numérico, único, adicionado a uma relação para servir como chave primária

Fica escondida nas aplicações e são frequentemente usadas no lugar de uma chave primária composta.

**O Conceito de Domínio**

\*Definir tipos de dados

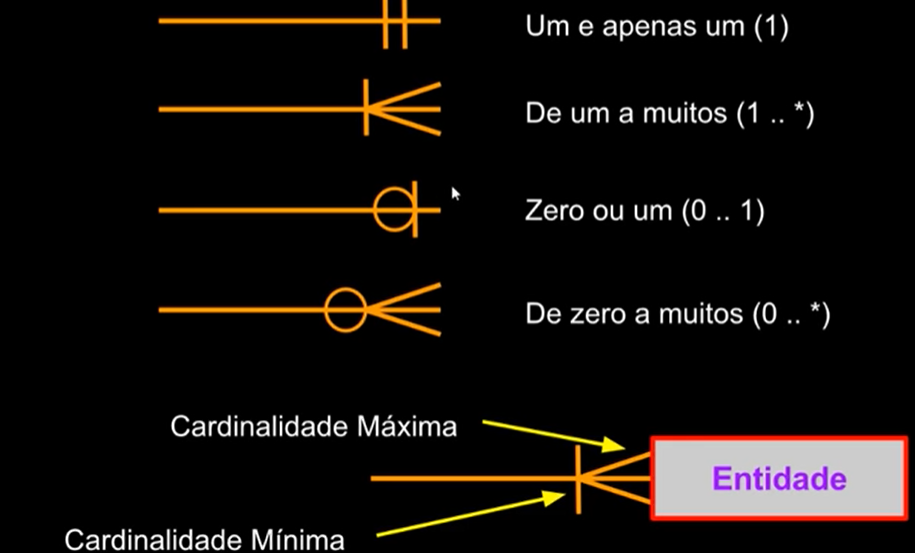
\*Especificar valores válidos em um campo

**Cardinalidades**

A cardinalidade diz respeito ao número de itens que se relacionam nas entidades.

A cardinalidade pode ser máxima ou mínima, significando respectivamente os números mínimo e máximo de instâncias de cada entidade associadas no relacionamento.

**Simbologia para Cardinalidades**



**Restrições de Integridade**

**Integridade de Dados**

Manutenção e garantia da consistência e precisão dos dados, sendo um aspecto crítico no design, implementação e uso de sistemas de armazenamentos de dados.

A integridade é atendida por meio da aplicação de Restrições de Integridade.

**Integridade de Domínio** = Valores inseridos em uma coluna devem sempre obedecer à definição dos valores que são permitidos para essa coluna – os valores do domínio.

Exemplo – Em uma coluna que armazena preço de mercadorias, os valores admitidos são do domínio numérico- ou seja, apenas número.

Fatores : Tipo de Dado do campo / Representação interna do tipo de Dado / Presença ou não do dado / Intervalos de valores no domínio / Conjuntos de valores discretos;

Exemplo : Atributo **Preço do Produto**: Valor Monetário

\*Valor Permitido : 25,33

\*Valor não Permitido: 25 Reais e 33 Centavos / -32,33

**Integridade Referencial**

Uma restrição de Integridade Referencial assegura que valores de uma coluna em uma tabela são validos baseados nos valores em uma outra tabela relacionada.

Exemplo: Se um produto de ID 523 foi cadastrado em uma tabela de Vendas, então um produto com ID 523 deve existir na tabela de Produtos relacionada.

Exemplo 2 : Atributo Nome\_Produto: Caracteres

\*Valores permitidos (produtos cadastrados):

Água

Refrigerante

Suco

\*Valores não permitido para venda ( não existente na tabela de produtos):

Cerveja

OBS: Se um registro for excluído em uma tabela, então os registros relacionados em outras tabelas que o referenciam talvez precisem ser excluídos. Caso contrário ocorrerá erro, o mesmo se dá com atualização de registros.

**Integridade de Vazio**

Este tipo de integridade informa se a coluna é obrigatória ou opcional – ou seja , se é possível não inserir um valor na coluna.

Uma coluna de chave primária, por exemplo, sempre deve ter dados inseridos, e nunca pode estar vazia, para nenhum registro.

**Integridade de Chave**

Os valores inseridos na coluna de chave primaria (PK) devem ser sempre únicos, não admitindo-se repetições nesses valores. Desta forma, as tuplas (registros) serão sempre distintas. Os valores de chave primária também não podem ser nulos.

**Integridade Definida pelo Usuário**

Diz respeito as regras de negócio específicas que são definidas pelo usuário do banco de dados.

Por exemplo, pode-se definir que uma coluna somente aceitará um conjunto restrito de valores.

**Dicionário de Dados**

Documento usado para armazenar informações sobre o conteúdo, formato e a estrutura de um banco de dados , assim como os relacionamentos entre os seus elementos.

É chamado de Repositório de Metadados

**Dependências**

Dependência Funcional = Aquele que o Atributo depende do outros. Ou seja exemplo X é ligado a Y.

O atributo que determina o valor é chamado Determinante.

Já o outro é chamado de Dependente.

**Dependência Funcional Total** = Em uma relação com uma PK composta, um atributo não-chave que dependa dessa PK como um todo, e não somente de parte dela, é dito como possuindo Dependência Funcional Total.

Dependência Funcional Parcial = É parcial quando os atributos não-chaves não dependem funcionalmente de toda PK quando está composta. Ou seja, existe uma dependência funcional, mas somente de uma parte da chave primária.

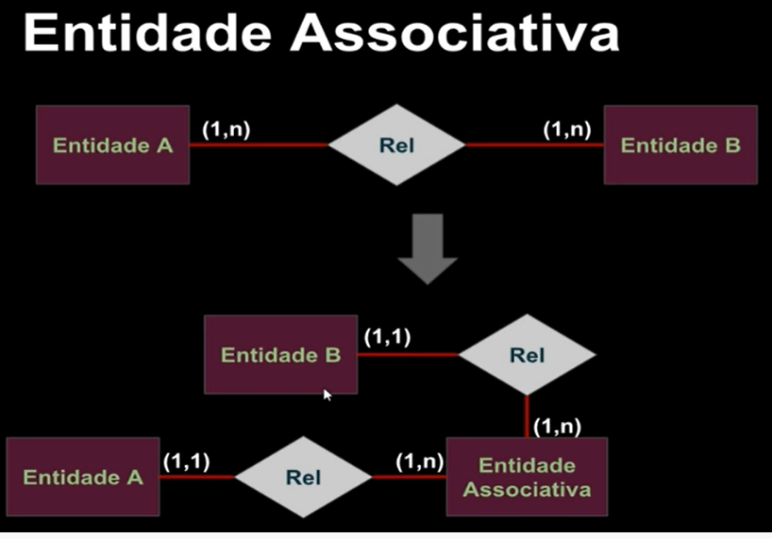
**ENTIDADE ASSOCIATIVA**

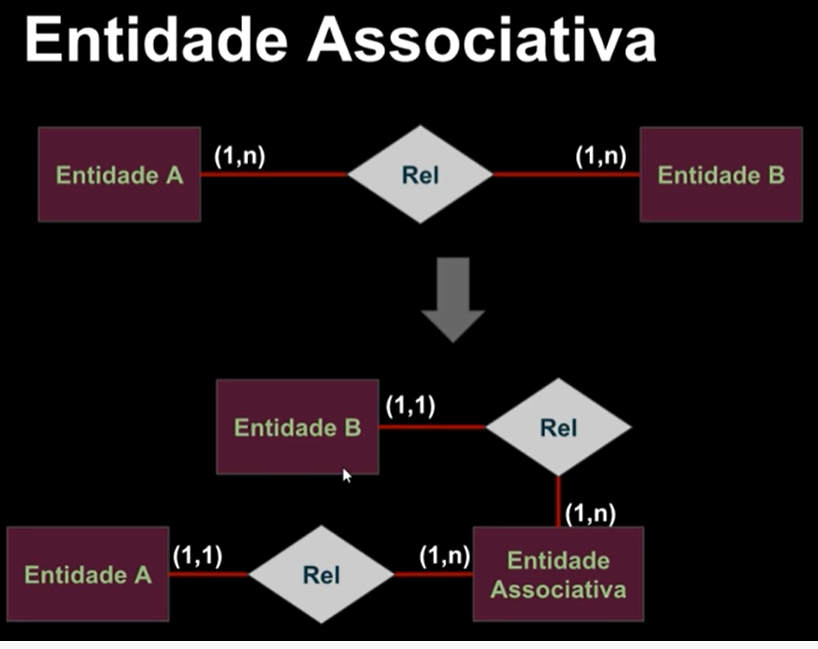
OBSERVAÇÃO: Uma entidade Associativa é implementada para resolver um relacionamento muitos-para-muitos (N:M).Gera uma tabela associativa, que permite mapear duas ou mais tabelas fazendo referência às chaves primárias de cada tabela. Contém chaves estrangeiras, cada uma em um relacionamento um-ara-muitos da tabela de junção para as tabelas de dados individuais.

Sua chave primária, no geral, é composta a partir das colunas de chaves estrangeiras em si.

Uma tabela associativa também pode ser chamada de: Tabela de Referência Cruzada, Tabela de Intersecção, Tabela de Junção, Tabela de Mapeamento, Tabela de Transição, ou ainda outros nomes.

**RELACIONAMENTOS MUITOS para MUITOS** (1,n) – (1,n) – Gera problemas de duplicidade de informações ou informações cruzadas ao inserir Dados no DATE BASE.





**Atributos**

**Atributo Simples** = É o que não possui características especial e não possui Chave, ou seja possui Valor Único como Nome, Rua.

**Atributo Composto** = É o que é formado por vários itens menores, exemplo **endereço** que possui vários campos menores.

**Atributo Multivalorado** = É o atributo que possui mais de um valor exemplo **Telefone**, a pessoa pode ter mas de um.

**Atributo Determinante =** É o que identifica de uma forma única e no Banco de Dados ele é dito como PK, como por exemplo **CPF,CNPJ,NUMERO DA INSCRIÇÃO** ou seja só possui um único Valor.

**Formas Normais**

Entendimento = Uma forma normal, no contexto de bancos de dados, refere-se à uma diretriz, uma convenção com o objetivo de prevenir anomalias e inconsistências em meio às atualizações frequentes que uma base irá sofrer, minimizando a redundância ao preço de uma menor eficiência entre consultas.

**(É uma forma de deixar o sistema de uma maneira exemplificada, fazendo com o mesmo futuramente ao fazer alterações não sofra e tenha consequências que tragam prejuízo a empresa).**

**1FN (1ª Forma Normal)**

Refere-se ao formato de um registro. Esta diretriz exclui a possibilidade de haverem campos que possuem mais de um atributo, ou seja, um vetor ou grupo de atributos.

(É a forma que só pode possuir atributo simples e não pode possuir atributos compostos e mais de uma PK)

**2FN (2ª Forma Normal)**

\*Está na 1FN

\*Todos os atributos não-chave são funcionalmente dependentes de todas as partes da chave primária.

\*Não existem dependências parciais, e atributos não dependem de chaves candidatas.

Caso contrário, deve-se gerar uma nova tabela com os dados.

Um atributo-chave é um que é uma PK ou parte de uma PK composta.

--------------------------------------------------------------------------

Esta forma normal refere-se ao relacionamento entre atributos dentro de uma determinada tabela, mais especificamente entre atributos chave e não-chave, tomaremos daqui em diante “chave” como chave primária, não considerando chaves estrangeiras.

Um esquema está na Segunda Forma Normal (2FN) se está na 1FN e se cada atributo não chave de uma tabela refere-se à chave da mesma **como um todo.**

(É a forma em que os campos na tabela precisão está ligado uma única PK e por contrario é necessário organizar e torna-la organizável).

**3FN (3ª Forma Normal)**

\*Estiver na 2FN

\*Não existirem dependências transitivas.

\*Uma tabela está na Terceira Forma Normal se ela estiver na segunda forma normal e se nenhuma coluna não-chave depender de outra coluna não-chave.

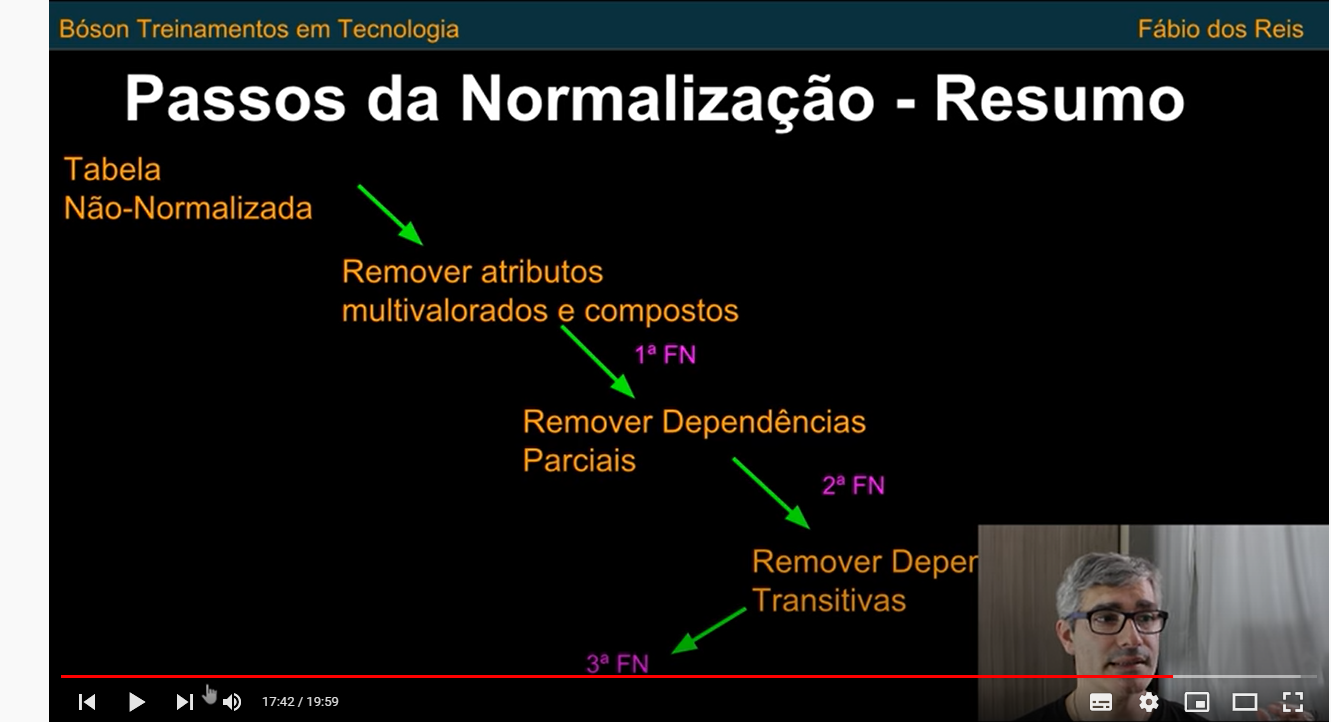
Dependência transitiva é uma dependência funcional entre dois ou mais atributos não-chave.

Caso contrário, deve-se gerar uma nova tabela.

A Terceira Forma Normal (3FN) é bastante semelhante a segunda e também refere-se a chaves. Além disso, ela também requer que antes a Primeira Forma Normal seja satisfeita. Esta diretriz é violada quando um atributo não chave refere-se a outro atributo que também não é chave dentro de uma tabela.

Um esquema está na Segunda e Terceira Forma Normal se todo campo é parte da chave ou provê um fato sobre toda a chave, nunca uma parte dela.

(É a forma quase idêntica a segunda , ela não fica formal a 3FN quando o atributo não chave esta com ligação a outro atributo na mesma tabela sendo que ele também não é PK. Pra ficar correto o campo deveria está ligado a PK e não metade).

****