Fatec

Faculdade de Tecnologia

ADM BD I

Administraçao de Banco de Dados I

Lúcia Satiko Murotani

WFernando Bastos

CONTEÚDO

Referência bibliográfica:

Felipe Nery Rodrigues Machado, Banco de Dados Projeto e Implementação – Editora Érica

Fanderuff, Damaris .Dominando o Oracle 9i – Modelagem e Desenvolvimento – Pearson Education – Makron Books

Poderoso, Celso Henrique. SQL Curso Prático

Apostila Professor Valter

CAPÍTULO 1 – PROJETO BANCO DE DADOS

BANCO DE DADOS

Definição : É uma coleção organizada de dados. É uma ferramenta desenvolvida para a manipulação eficiente de um grande conjunto de informações estruturadas e armazenadas de forma organizada e integrada.

Finalidade: Otimização dos sistemas.

SGBD (Sistema Gerenciador de Banco de Dados)

Tratando-se de dados informatizados utiliza-se o **SGBD (Sistema Gerenciador de Banco de Dados)** – é o software que permite a utilização simultânea de um banco de dados por múltiplos usuários e prove ferramentas para que eles possam criar e manipular as informações armazenadas no banco de dados.
Para ser considerado um SGBD devem seguir *algumas regras*, tais como: *auto-contenção, independência dos dados, abstração dos dados, visões, transações, acesso automático*, caso contrário, será um GA (Gerenciador de Arquivo) - é um programa de computador usado para criar e organizar pastas e ficheiros (arquivos) em sistemas operacionais. O acesso a informações em sistemas de processamentos de dados que não utilizam SGBDs é feito pelo acesso sequencial a um ou mais arquivos. Cabe ao desenvolvedor criar mecanismos de recuperação de informação.

Características de SGBD :

Integridade: consiste em impedir que determinado código ou chave em uma tabela não tenha correspondência em outra tabela. PK e FK.

Restrições ou consistência: informação confiável, o dado armazenado em um único local, com acesso descentralizado, compartilhado pelos vários sistemas.

Segurança ou privacidade: define para cada usuário o nível de acesso à tabela e /ou campo, seja de leitura, leitura e gravação ou sem acesso.

Restauração ou reorganização: apresentar facilidade para recuperar falhas de hardware e software, por meio da existência de arquivos de backup ou de outros recursos automáticos.

Controle de Redundância: consiste no armazenamento de uma mesma informação em locais diferentes, provocando inconsistências. Em um banco de dados as informações só se encontram armazenadas em um único local, controle no banco de dados.

Independência Física: imunidade das aplicações à estrutura de armazenamento e à estratégia de acesso às informações.

Padronização dos dado e esquematização: exemplificando, no campo sexo somente permite armazenamento ‘M’ ou ‘F’.

MODELAGEM DE DADOS

Existe de um lado as técnicas de Orientação a Objetos (que atualmente é indiscutível a vantagem obtida no projeto) e o grande mercado nacional e mundial utilizando os SGBD Relacionais. E como permitir que um projeto desenvolvido em O.O. seja facilmente inserido em um ambiente de banco de dados relacional? Denominamos como camada de persistência de dados.

O nosso objetivo é dominar as técnicas de modelagem de dados para banco de dados relacionais, completando com a implementação em projetos de análise estruturada ou com um projeto orientado a objetos.

Um dos principais problemas relacionados com banco de dados é a **redundância**, solucionamos com a criação de várias tabelas apesar de aumentar a complexidade. Quando se admite a redundância, é muito comum ter que repetir nomes, descrições, etc. Ao isolarmos essas informações em tabelas distintas e ao relacionarmos as tabelas por um código comum estamos **economizando espaço de armazenamento**.

A modelagem de dados busca representar a organização dos dados do mundo real. O objetivo de um modelo de dados é garantir que as informações existentes requeridos pela aplicação e pelo banco de dados estejam representados de forma simples para que tanto o usuário final entenda e o DBA (administrador do banco de dados) utilize para a criação da estrutura final do SGBD.

O analista abstrai do mundo real, ou seja, a parte do negócio que tem interesse em controlar, sem se preocupar nesse primeiro momento com qualquer conceito técnico. Obtemos assim um **modelo conceitual, tendo como resultado um esquema gráfico.**

O segundo passo é iniciar o modelo lógico, considerando as abordagens da tecnologia dos SGBD (relacional, hierárquica, rede ou Orientado a objeto) mas sem considerar, ainda, nenhuma característica específica de SGBD.

Somente no **modelo físico**, que será construído a partir do modelo lógico, deve obrigatoriamente definir o SGBD e descrever estruturas físicas de armazenamento de dados.

Projetando banco de dados

Para manter a estabilidade de todo o sistema e garantir um menor tempo que será despendido na manutenção do modelo é importante planejar a criação do banco de dados.

Metodologia Case

O processo de Análise de Dados pressupõe três fases distintas e integradas:

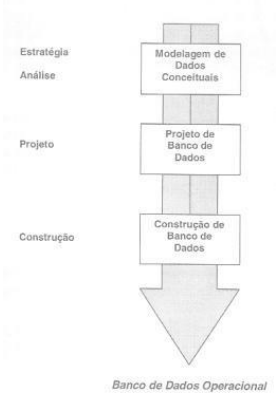


Figura 1

- Análise e modelagem utilizando modelo de entidade relacionamento e normalização de dados

- Definição de Tabela, índice, view, etc.

Estratégia

- Modelar o Negócio (Entrevista, Modelo de Dados “MER – simples “) , arquitetura (HD, link, etc) e plano de desenvolvimento (Cronograma Custo – R$);

Análise

- Detalhar a modelagem (MER – normalizado), especificações e estratégia de Implantação;

Projeto

- Arquitetura do Sistema, projetar o Banco de Dados, especificar módulos e rascunhar manuais do usuário;

Construção

- Criar os programas, criar o Banco de Dados (Linguagem SQL), planejar Implantação e Instalação de ambiente.

Esquema geral de modelagem de dados usando MER

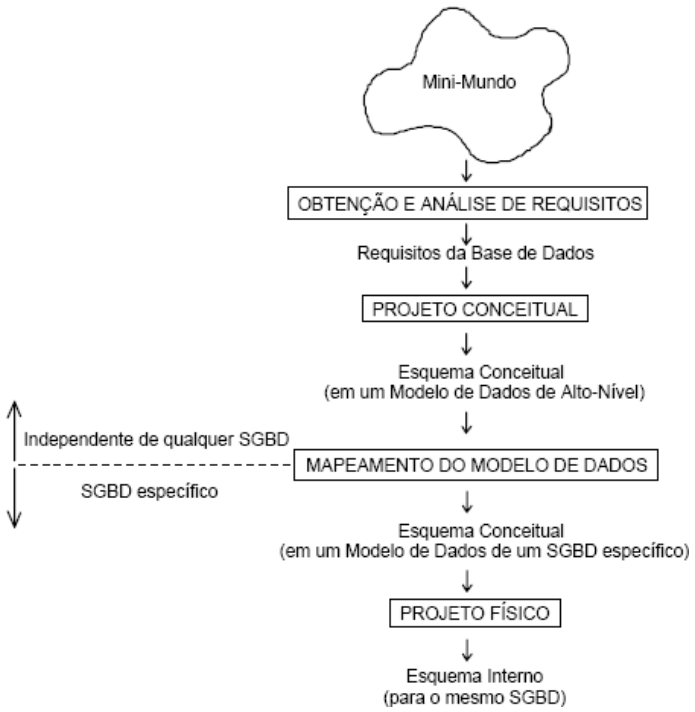


Figura 2

Modelo Conceitual

- Representação dos conceitos e características observados no ambiente;
- Ignorar particularidades de implementação.

Modelo Lógico

- Regras de Derivação:
  - Normalização das estruturas de dados
  - Derivação de estruturas de agregação e generalização-especialização
  - Derivação de relacionamentos

- Regras de Restrição:
  - Restrição de domínio
  - Restrição de Integridade
  - Restrição de Implementação

Modelo Físico

- Inclui a análise das características e recursos necessários para armazenamento e manipulação das estruturas de dados (estrutura de armazenamento, endereçamento, acesso e alocação física).

Modelo Entidade-Relacionamento

O Modelo Entidade-Relacionamento foi definido por Peter Chen em 1976, e teve como base a teoria relacional criada por E.F.Cood (1970). Segundo Chen, a visão de uma dada realidade, baseia-se no relacionamento entre entidades, os quais retratam os fatos que governam esta mesma realidade, e que cada um (entidade ou relacionamento) pode possuir atributos (qualificadores desta realidade).

O conceito de abstração permite ao analista separar da realidade em estudo, as partes que são realmente relevantes para o desenvolvimento do sistema de informações e excluir da modelagem todos os aspectos que não exercem influência sobre o ambiente a ser modelado. O objetivo da modelagem de dados é possibilitar a apresentação de uma visão única não redundante e resumida dos dados de uma aplicação. Também nos ajuda a entender a estrutura e o significado dos dados.

No desenvolvimento de aplicações em banco de dados, o Modelo Entidade Relacionamento (E-R) é o mais largamente utilizado para a representação e entendimento dos dados que compõem a essência de um sistema de informações.

A técnica de modelagem Entidade-Relacionamento proposta por Peter Chen está definida como uma notação orientada para o desenho do modelo conceitual, pois permite a descrição desse esquema conceitual sem a preocupação com problemas de implementação física ou de performance de banco de dados.

CAPITULO 2 - ENTIDADE, ATRIBUTOS, RELACIONAMENTOS

DER – DIAGRAMA ENTIDADE-RELACIONAMENTO

O diagrama ER é uma ferramenta para modelagem conceitual e lógica de um banco de dados amplamente utilizada no projeto de banco de dados, sendo considerado praticamente um padrão para modelagem, por ser de fácil compreensão e apresentar poucos conceitos.

Usaremos a notação gráfica original de Peter Chen com mínimas adaptações e extensões e comentários da Metodologia Case, James Martin. Para montar um modelo é necessário estudar detalhadamente os conceitos:

ENTIDADE

As entidades são o conjunto de objetos de mesma natureza, com as mesmas características ou atributos, abrigados sob um nome genérico. Entidade é algo que desempenha papel específico no sistema que está sendo modelado; é algo sobre o qual se deseja guardar informações. A existência e a identificação de uma entidade dependem inteiramente do contexto em que ela estiver inserida.

Uma entidade pode ser: um objeto real, como livro, máquina, lugar, etc;
Uma pessoa, como empregado, contribuinte, cliente, aluno, etc;
Um conceito abstrato, como curso, conta, etc;
Um acontecimento, uma situação em que algo está ocorrendo ou está planejado, como fornecimento de encomenda, casamento, etc;

Representação de uma entidade

Deve-se utilizar nomes breves e objetivos que identifiquem facilmente o conteúdo da entidade associada, sempre no singular; nome deve ser um ou mais **substantivos**, em caso de nome composto por mais de uma palavra utilizar o separador _ (underline) e não utilizar preposições, ou seja, em vez de HISTORICO_DE_COMPRA, utilizar HISTORICO_COMPRA.

Classificação das Entidades

Entidade forte ou primária: São entidades de dados que possuem grau de independência com relação a existência e identificação.

Entidade fraca ou dependente: Sua existência depende da existência de outra entidade, dita forte.

Exemplos: A Entidade *BANCO* é uma entidade forte (não depende de ninguém). A Entidade *AGENCIA* depende dela).

A Entidade *DISCIPLINA* é uma entidade forte e a Entidade *TURMA* é a Entidade fraca da tabela Disciplina. (a existência da primeira tabela é dependente da segunda)

Entidade associativa: Resulta da associação de duas ou mais entidades. Exemplo: PUBLICACAO_AUTOR é concebida da existência da entidade PUBLICACAO e AUTOR.

ATRIBUTOS

Atributos são informações referentes a uma entidade que necessitam ser conhecida ou até mesmo armazenadas. Atributos descrevem uma Entidade pela qualificação, identificação, quantificação ou expressão de estado da Entidade**. O que descreve uma Entidade?**

No caso de uma pessoa, um funcionário de um banco por exemplo, quais são os dados importantes que deverão ser mantidos?
FUNCIONARIO (matricula, nome, endereço, telefones, dataNascimento)

Atributos representam um tipo de descrição ou detalhe, não uma ocorrência ou exemplo.

Classificação dos Atributos

Atributo Chave: Identifica unicamente as instâncias das entidades do modelo;

Atributo Simples ou não chave: são os descritores, representa características simples;

Atributo Composto: tem outros atributos aninhados (subatributos);

Atributo Multivalorado: Abriga vários valores para uma única instância de uma entidade;

Atributo Derivado: Pode ser calculado a partir dos demais atributos da entidade;

Representação de um atributo

Nome de um atributo deve ser sempre no singular, não colocar acentuação;
Utilizar nomes breves e objetivos que identifiquem facilmente o conteúdo do atributo;

Se uma entidade têm atributos multivalorados temos de criar uma tabela com o atributo determinante dessa entidade e o atributo multivalorado ou se isso não se justificar definir um número fixo para esse atributo e desdobrá-lo nesse número de colunas.

ClientesTel (num_cliente, telefones) ⟹ Clientes (num_cliente, nome, tel1, tel2)

Se uma entidade têm atributos compostos temos de criar na tabela dessa entidade colunas referentes a esse atributo.

Clientes(N-contri, Nome, Telf1,Telf2, Rua, Cidade,Informações)

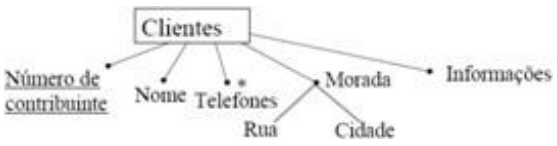


Figura 3 -Exemplo de atributo composto

IDENTIFICAR E MODELAR ENTIDADES

Seguir os passos seguintes para identificar e modelar Entidades a partir de um conjunto de notas feitas através de entrevistas:

- Examinar todos os substantivos. São coisas de significância ao negócio?
- Determine um nome para cada Entidade.
- Possui informações de interesse para a Entidade que necessitam ser conhecidas ou guardadas?
- Cada ocorrência da Entidade pode ser identificada unicamente? Qual atributo ou atributos servem como identificador único?
- Enumere exemplos, ocorrências, justificando os atributos levantados anteriormente. Por exemplo, “Pedro, Luana, Vanessa são todos empregados”.

DISTINGUIR ATRIBUTOS DE ENTIDADES

Se um Atributo tem Atributos de sua propriedade então ele é realmente uma Entidade.
Todas as Entidades são substantivos, mas nem sempre todos os substantivos são Entidades.

CARACTERÍSTICAS ENTIDADES	CARACTERISTICAS ATRIBUTOS
Qualquer objeto significante que possui informações que devem ser guardadas	Qualifica uma entidade
Possuiu um ou mais atributos	Não possui atributos de sua propriedade
Pode ter múltiplas ocorrências associadas com outras entidades via um relacionamento	Tem um único valor para cada ocorrência da entidade(não pode ter grupos de repetição)
Se uma entidade não tem atributos, ela pode ser um atributo e não uma entidade	Se um atributo possui atributo. Ele é uma Entidade ou não tem significância alguma.

Todo atributo multivalorado, torna-se uma Entidade

ANALISAR E MODELAR ATRIBUTOS

- Identificar um atributo canditado;
- Associar cada Atributo com uma Entidade;
- Nomear o Atributo
- Determinar a opcionalidade do Atributo;
- Validar o Atributo, ver se é realmente um Atributo ou não é uma Entidade;
- Ver a possibilidade de quebrar em vários Atributos
- Verificar que um Atributo possui valore singulares;
- Verificar que um Atributo não é um dado derivado.

OPCIONALIDADE DE UM ATRIBUTO [case/ PeterChen]

Obrigatórios: Um valor **dever ser** conhecido para cada ocorrência da Entidade. [*/ (1,1)]

Opcionais: Um valor **pode ser** conhecido para cada ocorrência da Entidade. [o/ (0,1)]

IDENTIFICADORES ÚNICOS PARA UM ATRIBUTO

Um identificador (UID) é qualquer combinação de Atributos e/ou Relacionamentos que serve para identificar unicamente cada ocorrência de uma Entidade. Cada ocorrência de Entidade deve ser identificada unicamente. [case]
único
#* único e obrigatório #*1

RELACIONAMENTOS

Um relacionamento pode ser entendido como uma associação entre instâncias de Entidades devido a regras de negócio. Normalmente ocorre entre instâncias de duas ou mais Entidades , podendo ocorrer entre instâncias da mesma Entidade (auto-relacionamento).

Por que o relacionamento é necessário ?

Quando existem várias possibilidades de relacionamento entre o par das entidades e se deseja representar apenas um;
Quando ocorrer mais de um relacionamento entre o par de entidades;
Para evitar ambiguidade;
Quando houver auto-relacionamento;

> ***Com a evolução e a criação de ferramentas CASE, foram criadas outros tipos de notação. Engenharia de Informações foi criado na década de 80 por James Martin. A maioria das ferramentas case de mercado não disponibilizam o tipo de notação Peter Chen.***

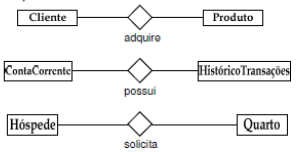
Representação gráfica (Peter Chen)

Entidades: Representada através de um retângulo

Atributos: conforme figura5

Relacionamento: Representada através de um segmento de reta ligando as classes cujos objetos se relacionam. Representadas através de um losango. Utilizar um verbo que identifique claramente o relacionamento existente;

Exemplos:



Grau de Relacionamento

O numero de conjuntos de entidades que participa de um conjunto de relacionamento é também o grau desse conjunto de relacionamento. Um conjunto de relacionamento binário é de grau dois; um relacionamento ternário é de grau três. O mais comum é o binário.

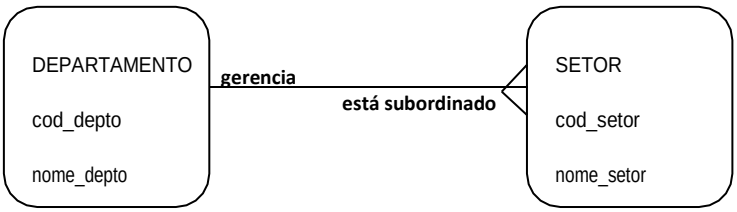
Ternário  Binário

Representação gráfica (Metodologia Case)

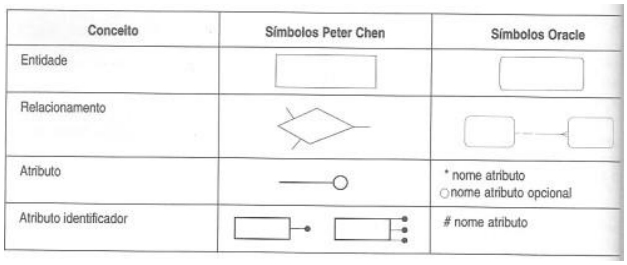
Entidade: Representada através de um retângulo com bordas arredondadas;

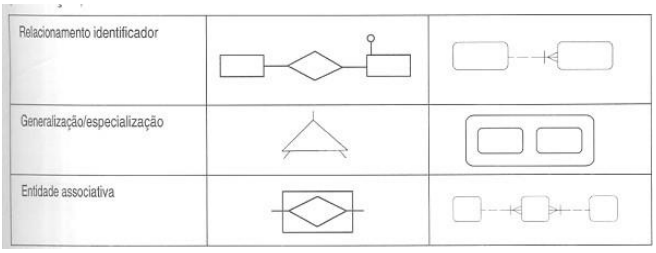
Atributos: Representados dentro do retângulo.

Relacionamento: Representada através de um segmento de reta ligando as classes cujos objetos se relacionam. Utilizar um verbo que identifique claramente o relacionamento existente de ambos os lados.



Comparação entre representações





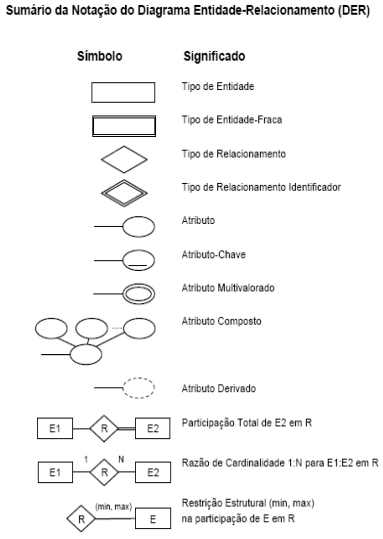


Figura 5

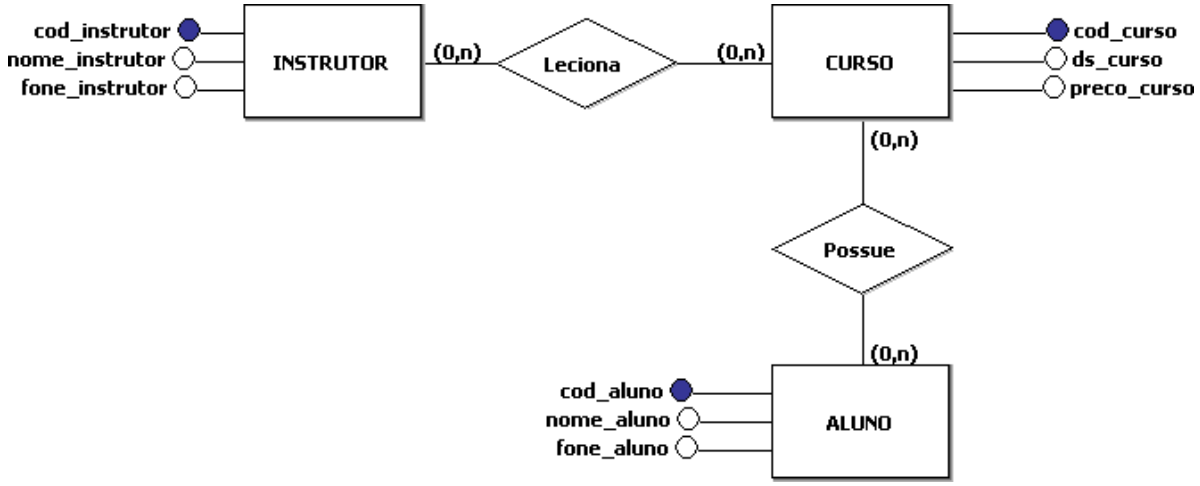
Vide exemplo do atributo composto na figura 1

Exemplo de um projeto de Banco de Dados

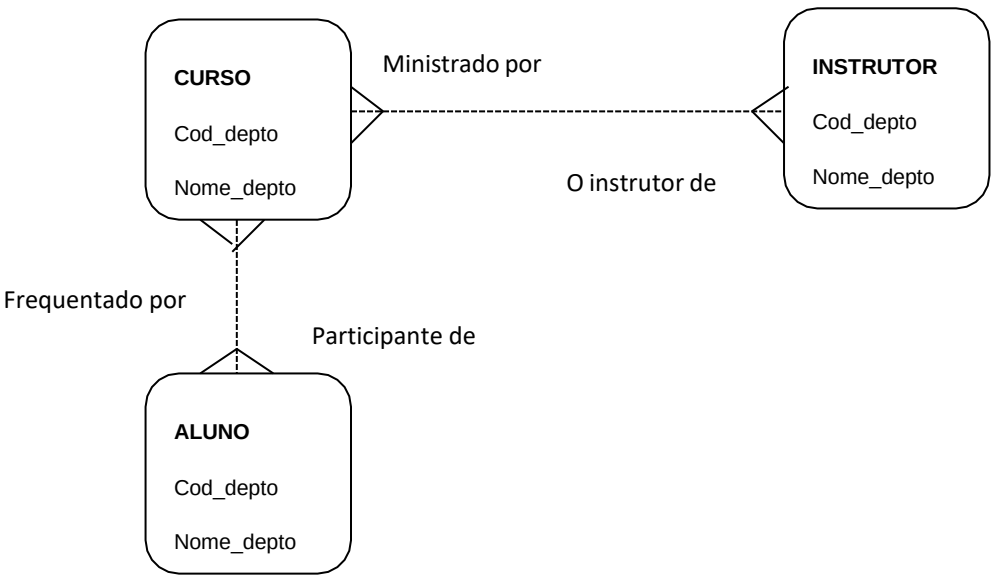
Entrevista: "Sou gerente de uma empresa de treinamento que ministra vários Cursos de caráter técnico. Ministramos vários cursos, que são identificados por um código, nome e preço. Os cursos "VB com SQLServer2008", "Delphi com Oracle", "Java e O.O." são alguns dos nossos cursos mais populares. A duração de cada curso pode variar de cinco a dez dias. Um instrutor pode ensinar vários cursos. Robert, Pedro e Rosimery são alguns de nossos instrutores. Mantemos aqui o nome e o telefone de cada instrutor. A gente cria um curso e aloca um instrutor. Os alunos (clientes) podem participar de vários cursos e, vários deles o fazem. O Joseph Brito, da SCA Informática, assiste a todo curso que oferecemos. Além do nome, mantemos também o número do telefone dos alunos. Alguns de nossos alunos e instrutores não possuem telefone"

Modelar o negócio: 1º Passo- identificação das possíveis Entidades, Atributos e Relacionamentos (Desenvolver o MER – "simples")

Peter Chen



Metodologia Case



CAPÍTULO 3 – CONCEITOS : RESTRIÇÕES DE INTEGRIDADE E ATRIBUTOS DE RELACIONAMENTO

RESTRIÇÕES DE INTEGRIDADE

As restrições de integridade no banco de dados:
Integridade -> correção, consistência e segurança dos dados armazenados.
Os aspectos de integridade básica do modelo relacional estão associados aos conceitos de chave de acesso. Ou seja, garantir a integridade de um esquema relacional significa garantir o acesso individualizado a todas as ocorrências de uma tabela, assim como garantir relacionamentos válidos e condizentes com a realidade.

Tipos de chave de acesso:

CHAVE PRIMÁRIA (primary key – pk) : Principal chave de acesso a uma tabela. Não permite duplicidade, visa manter a unicidade e consistência dos dados de uma tabela.

CHAVE COMPOSTA OU CONCATENADA: corresponde à combinação de duas ou mais chaves, e pode ser necessária para eliminar a ambiguidade, formando um identificador único;
Quando um único atributo da Entidade/Objeto de Dados não consegue personalizá-la, há a necessidade da junção com outros atributos para este objetivo. Este conjunto de atributos forma então uma **chave** que consegue personalizar a Linha da Entidade/Objeto de Dados**.** Esta **chave primária especial** tem o nome de **chave concatenada ou composta**.

CHAVE ÚNICA (chave candidata) : Além da chave primária, uma tabela pode possuir quantas chaves únicas forem necessárias.

CHAVE SECUNDÁRIA: É a chave auxiliar de acesso a uma tabela. A chave secundária também possui índices relacionados utilizados em campos onde se efetua constante pesquisa de acesso. Não precisam ser necessariamente únicos.

CHAVE ESTRANGEIRA (foreign key –fk): Permite o acesso e a validação de outras tabelas. Essa chave permite que se estabeleçam os relacionamentos em um banco de dados. A chave estrangeira deve ser compatível com sua correspondente em outra tabela, isso garante a integridade referencial.

Caracterizam as restrições nas quais os relacionamentos entre entidades estão submetidos (regras do negócio).

Exemplo:

“Todo empregado deve estar lotado num departamento”

“Cada aluno deve ser matriculado em um ou mais cursos”

“Toda Nota Fiscal deve ter pelo menos um item discriminado”

O esquema de Entidade Relacionamento de uma empresa pode definir certas restrições, as quais o conteúdo do banco de dados deve respeitar. Isso é feito utilizando o Mapeamento de Cardinalidade: expressa o número de entidades as quais outra entidade pode estar associada via um conjunto de relacionamentos**.**

GRAU DE CARDINALIDADE

O grau de cardinalidade representa a quantidade de ocorrências de cada entidade que estão envolvidas no relacionamento. Podem ser:

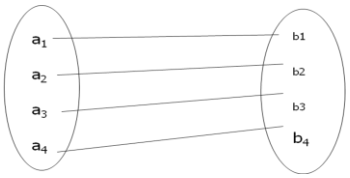
Em um projeto de BD é usada somente duas cardinalidades mínimas: a cardinalidade mínima 0 e a cardinalidade mínima 1.

Cardinalidade Mínima: especifica se a participação de todas as ocorrências das entidades no relacionamento é obrigatória ou opcional.

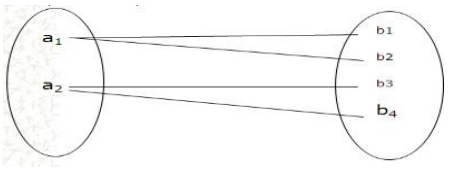
A cardinalidade **mínima 1** recebe a denominação de **"associação obrigatória".**
A cardinalidade **mínima 0** recebe a denominação de **"associação opcional".**
A cardinalidade mínima em um Diagrama é anotada junto a cardinalidade máxima.

Cardinalidade máxima: indica a qtde. máxima de ocorrências de entidades que podem estar associadas a uma ocorrência da outra entidade (1 ou N).

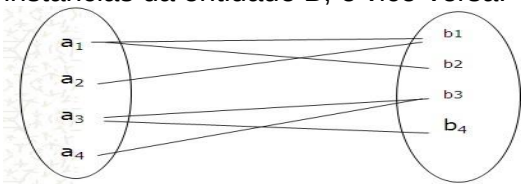
Um_para_Um (1:1) - Uma entidade em A está associada no máximo a uma entidade em B, e uma entidade em B está associada a no máximo uma entidade em A.



Um_para_Muitos (1:N) - Uma instância de uma entidade A está associada a qualquer número de instâncias da entidade B. Porém, uma instância da entidade B pode estar associada, no máximo, a uma instância da entidade A.



Muitos_para_Um (N:1) - uma instância da entidade A está associada a uma instância de B. Porém, uma instância de B pode estar associada a qualquer número de instâncias de A.
Muitos_para_Muitos(M:N) - uma instância da entidade A está associada a qualquer número de instâncias da entidade B, e vice-versa.



O uso de "zero" (0:1) ou (0:N) indica a totalidade do relacionamento.

Exemplos:

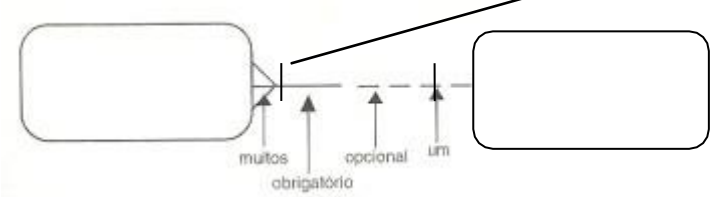
Pode haver um empregado que esteja alocado a uma mesa.
Pode haver uma mesa que não esteja sendo alocada por nenhum empregado.
Um empregado está alocado a uma, e somente uma mesa.

Pode haver um cliente que esteja associado a vários pedidos.
Pode haver um cliente que não esteja associado a pedido algum.
Um pedido está associado a um, e somente um cliente.

Conectividade	Em um extremo	No outro extremo
Um para um	(0,1) (1,1)	(0,1) (1,1)
Um para muitos	(0,1) (1,1)	(0,N) (1,N)
Muitos para muitos	(0,N) (1,N)	(0,N) (1,N)

Representação Crows Foot – James Martin

A barra indica que o Relacionamento com a entidade B faz parte do UID (identificador único) da entidade A.



Entidade A Entidade B

ATRIBUTOS DE RELACIONAMENTOS

Normalmente ocorrem quando duas ou mais entidades estão relacionadas, e é necessário manter informações sobre esta associação.
Atributos de relacionamentos são normalmente assinalados em relacionamentos muitos para muitos.
Nunca assinalamos atributos em relacionamentos 1-1 ou 1-N.

• identificação da existência de um atributo:

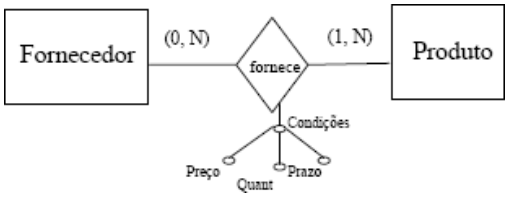
– fixa-se uma instância da uma das entidades, e observa-se o valor do atributo para cada mudança de instância na outra entidade.
– se o valor do atributo mudar, então ele não pertence à entidade que foi fixada.

Para melhor entendimento, a quem pertence o atributo CONDIÇÕES = PREÇO + QUANTIDADE + PRAZO ?

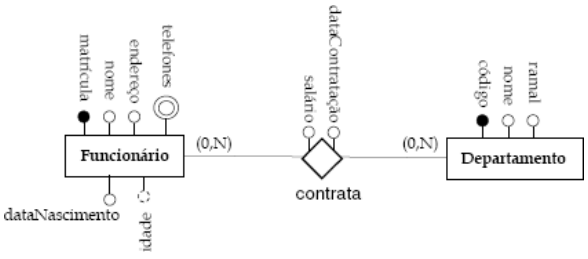
PREÇO, QUANTIDADE e PRAZO, não podem pertencer a PRODUTO, pois se fosse assim TODOS os FORNECEDORes deveriam praticar o mesmo preço.

PREÇO, QUANTIDADE e PRAZO, não podem pertencer a FORNECEDOR, pois se fosse assim TODOS os PRODUTOS de um FORNECEDOR teriam o mesmo preço.

Os atributos não pertencem nem a PRODUTO nem a FORNECEDOR, mas são relevantes para o FORNECE. Então, são do relacionamento.



Exemplo de Entidade, Atributos, Relacionamentos e Atributos de Relacionamentos:



(Peter Chen)

IDENTIFICAR E MODELAR RELACIONAMENTOS

Analisar e modelar Relacionamentos:

Determine a existência de relacionamentos entre as entidades;
Dê um nome a cada relacionamento identificado;
Determine a opcionalidade em cada direção do relacionamento;
Determine a cardinalidade (grau) em cada direção do relacionamento;
Leia e valide os relacionamentos identificados.

Após identificados todas as entidades e atributos, vamos melhorar o entendimento de "relacionamento"

DETERMINE A EXISTÊNCIA DE UM RELACIONAMENTO

Até que você tenha prática na identificação de relacionamentos, pode ser útil uma matriz de relacionamentos. Essa matriz nada mais é que a relação de todas as entidades em linha e coluna. Na intersecção das entidades, você deve identificar se há relacionamento. Se houver, coloque o verbo que
caracteriza o relacionamento. Assim:

	CD	GRAVADORA	MÚSICA	AUTOR
CD		Gravado por	Contém	
GRAVADORA	Grava			
MÚSICA	Está em			É escrita
AUTOR			Escreve	

NOMEAR O RELACIONAMENTO

Nomear cada direção de um Relacionamento. (voz ativa e voz passiva).

Pergunte um nome do Relacionamento:
Como é que a Entidade A está relacionada para a Entidade B?
Uma Entidade A é nome relacionamento a uma Entidade B.
Como é que a Entidade B está relacionada para uma entidade A?
Uma Entidade B é nome Relacionamento a uma Entidade A.

Exemplo:
Considerar o relacionamento entre departamento e empregado.

COMO É QUE UM DEPARTAMENTO ESTÁ RELACIONADO A UM EMPREGADO?
CADA DEPARTAMENTO É RESPONSÁVEL POR UM EMPREGADO.

COMO É QUE UM EMPREGADO ESTÁ RELACIONADO A UM DEPARTAMENTO?
CADA EMPREGADO É ATRIBUÍDO A UM DEPARTAMENTO.

Usar uma lista de nomes de Relacionamento para auxiliar em nomeação de Relacionamentos.

Baseado Em	Base Para
Descrição De	Para
Operado Por	Operador para
Representado Por	Representação De
Responsável Por	Responsabilidade De
Pertence a	Possui

Evite nomes genéricos. "Relacionado a, Associado com"

DETERMINAR A OPCIONALIDADE DE UM RELACIONAMENTO

Questione a opcionalidade de um Relacionamento?
Deve a Entidade A ser nome do Relacionamento Entidade B? **(Sempre?)**
Deve a Entidade B ser nome Relacionamento Entidade A?

Exemplo:
DEVE UM EMPREGADO SER ATRIBUÍDO A UM DEPARTAMENTO? **Sempre?**
EXISTE ALGUMA SITUAÇÃO EM QUE O EMPREGADO NÃO DEVERÁ SER ATRIBUIDO A UM DEPARTAMENTO?
NÃO *UM EMPREGADO SEMPRE DEVE SER ATRIBUÍDO PARA UM DEPARTAMENTO!*

DEVE UM DEPARTAMENTO SER RESPONSÁVEL POR UM EMPREGADO?
NÃO, UM DEPARTAMENTO NÃO TEM DE SER RESPONSÁVEL POR UM EMPREGADO. ELE PODE SER RESPONSÁVEL.

DETERMINAR O GRAU DE UM RELACIONAMENTO

Pergunte o grau de Relacionamento?
Pode a Entidade A ser nome do relacionamento mais que uma Entidade B?
Pode a Entidade B ser nome do Relacionamento mais que uma Entidade A?

Exemplo:
PODE UM EMPREGADO SER ATRIBUÍDO A MAIS QUE UM DEPARTAMENTO?
NÃO, UM EMPREGADO PODE SER ATRIBUÍDO A SOMENTE UM DEPARTAMENTO.

PODE UM DEPARTAMENTO SER RESPONSÁVEL POR MAIS QUE UM EMPREGADO?
SIM, UM DEPARTAMENTO PODE SER RESPONSÁVEL POR UM OU MAIS EMPREGADOS.

VALIDAR O RELACIONAMENTO

Ler em voz alta o Relacionamento.
A leitura do Relacionamento deve ter senso com relação ao negócio.
Exemplo:
CADA EMPREGADO DEVE SER ATRIBUÍDO A UM E SOMENTE UM DEPARTAMENTO.
CADA DEPARTAMENTO PODE SER RESPONSÁVEL POR UM OU MAIS EMPREGADOS.

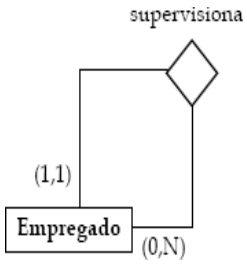
Devemos evitar dados derivados, pois são dados redundantes que podem gerar inconsistências no banco de dados. Relacionamento 1:1 são raros (pode ser a mesma entidade)

CAPÍTULO 5 – RELACIONAMENTOS REFLEXIVOS E EXTENSÕES DO MODELO E-R

AUTO-RELACIONAMENTO (REFLEXIVO OU RECURSIVO)

Um relacionamento recursivo é uma relação entre uma Entidade à ela mesma. No exemplo:
Cada empregado pode ser supervisionado por apenas um e somente um empregado.
Cada empregado pode ser gerente de um ou mais funcionários.

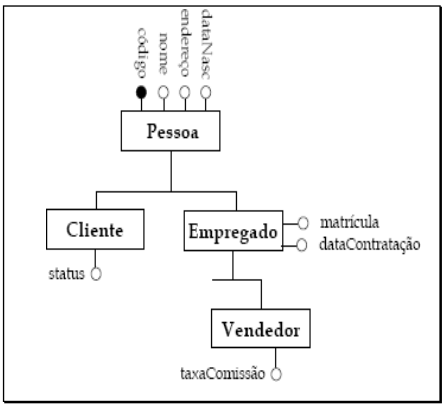
(Peter Chen)

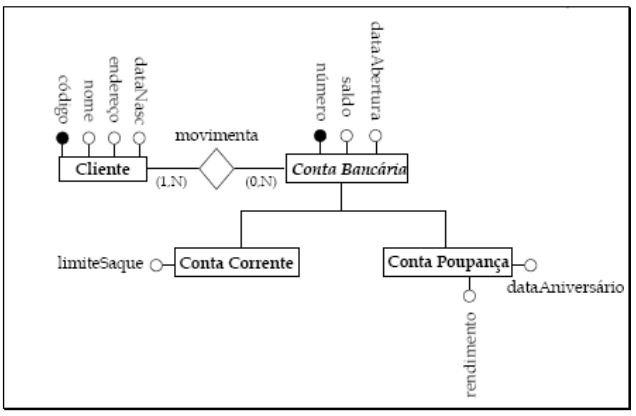


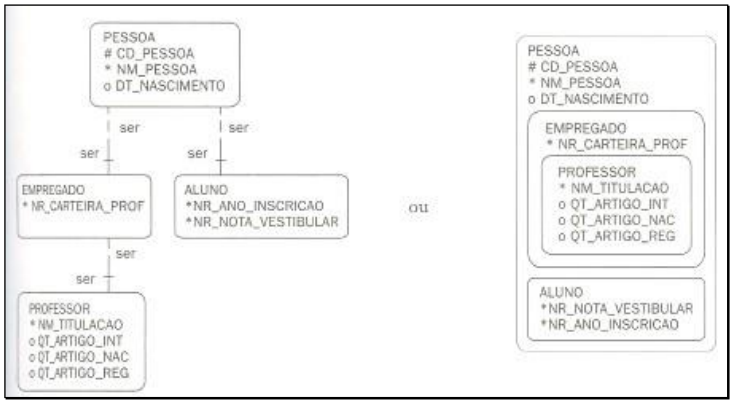
(Case).

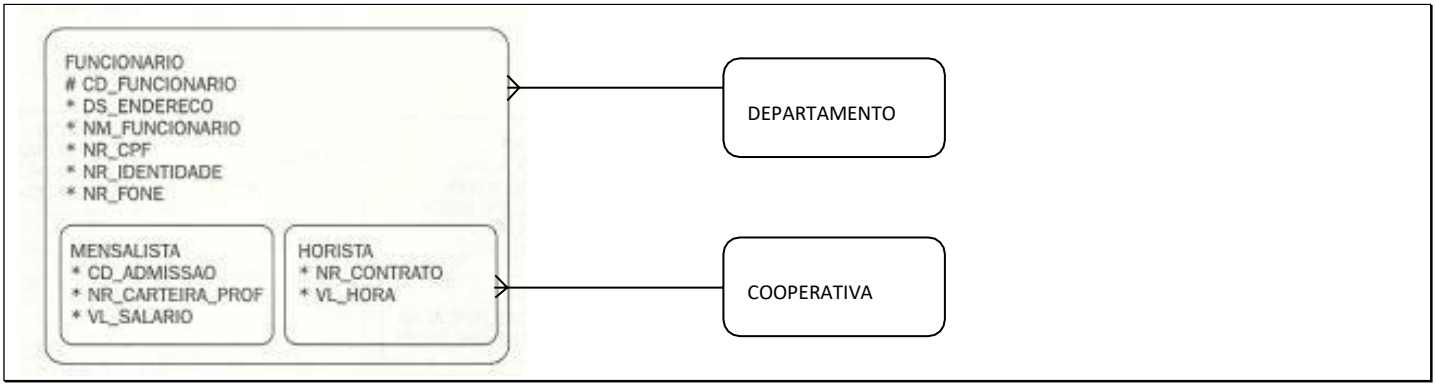
ESPECIALIZAÇÃO E GENERALIZAÇÃO (SUPERCLASSES E SUBCLASSES, SUPERTIPO E SUBTIPOS)

A generalização difere dos demais relacionamentos (entidade fraca, *n*-ários etc.) porque a primeira se trata de um relacionamento *entre entidades*.
Atributos e operações e relacionamentos são herdados pelas entidades filhas.









DOMÍNIO

A noção de domínio de um item de dado assemelha-se à noção de domínio de um conjunto na matemática. É um conjunto de regras de validação do negócio, restrições de formatos e outras propriedades que se aplicam a um grupo de atributos.
Por exemplo: Uma lista de valores (‘F’, ‘M’ para sexo, por exemplo); uma faixa de valores ([1,12]) para mês, por exemplo).
O domínio é definido uma única vez no dicionário de dados (banco de dados especificamente criado para guardar as definições dos dados e diagramas relacionados no banco de dados) e cada atributo será associado ao seu respectivo domínio.

TABELA

Objeto criado para armazenar os dados fisicamente. Os dados são armazenados em linhas (registros) e colunas (campos). Os dados de uma tabela normalmente descrevem um assunto tal como clientes, vendas, etc. Quando transporta ao modelo físico, uma tupla equivale a uma registro ou linha da tabela.

CAPITULO 6 – PROJETO LÓGICO DE BANCO DE DADOS

PROJETO LÓGICO DE BANCO DE DADOS

Quando da conclusão do modelo de dados da área de negócio, ou seja, identificadas todas as entidades e atributos, devemos iniciar o trabalho de modelagem de dados. Essa atividade deve ser iniciada com a aplicação de inspeção para a eliminação das redundâncias, e das técnicas de normalização, que serão introduzidas adiante. Em seguida, inicia-se o processo de identificação dos relacionamentos e das cardinalidades.

Com a conclusão da modelagem inicial, vamos dar início ao projeto inicial de Banco de Dados.
O projeto de Banco de Dados é realizado durante o estágio de Projeto do Ciclo de Desenvolvimento e é realizado concorrentemente com projetos de aplicações.

O projeto de Banco de Dados é realizado em duas atividades distintas:

1. Transcrever o Modelo ExR para Tabelas Relacionais para produzir um projeto inicial.
2. Redefinir projeto inicial para produzir um projeto completo de banco de dados.

Documentar para cada tabela Relacional em um formulário de Tabela, seguindo os passos:

- Transcrever simples Entidades para Tabelas;
- Transcrever Atributos para colunas e documentar exemplos dados;
- Transcrever identificadores únicos para Primary Keys;
- Transcrever Relacionamento para Foreign Keys;
- Escolher opções de Arco (Explícito ou Genérico – FK);
- Escolher opções de Subtipo (tabelas simples ou tabelas separadas).

PRIMARY KEY – PK (CHAVE PRIMÁRIA)

FOREIGN KEY – FK (CHAVE ESTRANGEIRA)

NOT NULL - NN (NÃO NULA, ou seja não pode estar vazio, o zero é um alfanumérico e espaço é um caractere para o banco de dados)

ÚNICA - U (ÚNICA, ou seja as informações não podem repetir na coluna especificada)

Toda Primary key deve ser obrigatoriamente Não Nula e única

Nem toda coluna única é uma chave primária

Metodologia Case

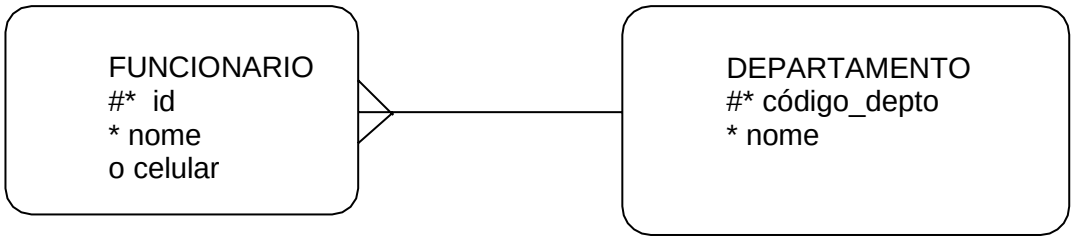


Tabela: FUNCIONARIO

Nome da Coluna	id	nome	celular	Cod_depto
Tipo de Chave	PK			FK
Único/ Nulo	NN.U	NN		NN
Dados Simples	101	Maria	7058-3166	10
	102	João		10

Peter Chen

Nome da Coluna	id	nome	sobrenome	Id_conjuge
Tipo de Chave	PK			FK
Único/ Nulo	NN.U	NN		U
Dados Simples	7450	Mary	Smith	5579
	5579	Bill	Jones	7450

CAPÍTULO 7 - O QUE É NORMALIZAÇÃO DE TABELAS?

NORMALIZAÇÃO DE TABELAS

Normalização é um conjunto de regras para minimizar redundância de dados.
Redundância de dados causam problemas de integridade. Atualizações e deleções podem não ser aplicadas de forma consistente para todas as cópias dos dados causando inconsistências de dados.
Normalização auxilia na identificação de Entidades, Relacionamentos e tabelas faltantes ou ausentes.

APLICAÇÃO DAS FORMAS NORMAIS

Primeira Forma normal

A normalização da tupla de forma que o relacionamento entre sua chave e os seus atributos seja unívoca, ou seja, para cada chave há a ocorrência de uma e somente uma informação de cada atributo.

CONCLUSÃO: Uma tabela está na 1FN se nenhum dos seus atributos tem domínio multivalorado.

OBJETIVO: Evitar que se tenha de reservar espaços para armazenar dados multivalorados, sendo que o espaço pode ser desperdiçado em um registro e se insuficiente em outro.

UTILIZAÇÃO: Projetam-se os atributos com domínio multivalorado para fora da tabela, levando um atributo (geralmente a chave da tabela original) como elo para refazer a ligação e recuperar o conteúdo da tabela original. PKFK / PK

Segunda Forma normal

A normalização da tupla que já submetida a primeira forma normal, apresenta chave concatenada que se relaciona de forma integral com todos os seus atributos.

CONCLUSÃO: Uma tabela está na 2FN quando está na 1FN e seus atributos dependem funcionalmente da totalidade da chave ou do atributo determinante. A 2FN aplica-se a tabelas onde a chave (atributo determinante) é composta por mais de um atributo

OBJETIVO: Evitar que se mantenham informações sobre um conjunto que tem intersecção com o conjunto representado na tabela, mas possui existência independente. Além da maior ocupação de espaço, a redundância aumenta a possibilidade de inconsistência.

UTILIZAÇÃO: Projetam-se os atributos que dependem funcionalmente da parte da chave para fora da tabela, levando parte da chave que determina como elo para refazer a ligação e recuperar o conteúdo da tabela original. PKFK / PKFK

Terceira Forma normal

Nenhuma coluna não chave pode ser funcionalmente depende de qualquer outra coluna não chave. Todo atributo não chave depende funcional e diretamente da chave primária.

CONCLUSÃO: Uma tabela está na 3FN quando está na 2FN e não há dependência funcional transitiva entre seus atributos. Dependência funcional transitiva é a situação em que um atributo depende de outro e este segundo depende de um terceiro.

OBJETIVO: Separar subconjuntos incertos em um subconjunto e evitar redundâncias nas informações. Além da repetição dos dados, a possibilidade de deterioração da qualidade de informação aumenta muito.

UTILIZAÇÃO: Projetam-se os atributos que dependem transitivamente da chave para fora da tabela, levando o seu determinante direto como elo para refazer a ligação e recuperar o conteúdo da tabela original. Dados de outra tabela que não seja identificada pela PK – cria uma nova tabela.

Pode-se normalizar durante a modelagem de dados. É dito que uma tabela está normalizada quando atende até a 3FN, porém se houver necessidade aplicar a 4FN e 5FN.

4FN – Quarta Forma Normal

4FUma tabela não deve possuir mais de uma D F Multivalorada.

Forma normal de boyce/ codd (FNBC)

As definições da 2fn e 3 fn, desenvolvidas por Codd, não cobriam certos casos. Essas inadequações foram apontadas por Raymond Boyce em 1974. Os casos não cobertos pelas definições de Cood somente ocorrem quando três condições aparecem juntas:

- A entidade tenha várias chaves candidatas;
- As chaves candidatas sejam concatenadas (mais de um atributo);
- As chaves concatenadas compartilhem pelo menos um atributo comum.

Uma entidade está na FNBC se e somente se todos os determinantes forem chaves candidatas.

Exemplo:

Entidade Cliente (cod_cliente, nome_cliente)

Entidade Agencia (cod_agencia, nome_agencia)

Entidade Empréstimo (cod_agencia, cod_cliente, nr_emprestimo, valor_emprestimo)

Aplicando a FNBC , teremos então a tabela Empréstimo da seguinte maneira:

Entidade Empréstimo (cod_agencia, nr_emprestimo, valor)

Entidade Devedor (cod_cliente, nr_emprestimo)

Exemplo:

Não normalizado:

Projeto						
cod_projeto	desc_projeto	cod_func	nome_func	cargo_func	sal_func	dt_inicio_proj
11	Alfa	1001	Antônio	Analista Sr.	1800	2/1/2009
11	Alfa	1004	Daniela	Analista Pl.	1200	5/1/2009
12	Beta	1003	Claudio	Analista Sr.	1800	10/2/2009

1º FN

ProjetoFunc					
cod_projeto	cod_func	nome_func	cargo_func	sal_func	dt_inicio_proj
11	1001	Antônio	Analista Sr.	1800	2/1/2009
11	1004	Daniela	Analista Pl.	1200	5/1/2009
12	1003	Claudio	Analista Sr.	1800	10/2/2009

Projeto	
cod_projeto	desc_projeto
11	Alfa
12	Beta

2º FN

Funcionario			
cod_func	nome_func	cargo_func	sal_func
1001	Antônio	Analista Sr.	1800
1004	Daniela	Analista Pl.	1200
1003	Claudio	Analista Sr.	1800

Projeto		
cod_projeto	cod_func	dt_inicio_proj
11	1001	2/1/2009
11	1004	5/1/2009
12	1003	10/2/2009

3º FN

Funcionario	
cod_func	nome_func
1001	Antônio
1004	Daniela
1003	Claudio

Cargo		
cod_cargo	cargo_func	sal_func
10	Analista Sr.	1800
20	Analista Pl.	1200
30	Analista Sr.	1800

Passando para a 4FN:

A tabela a seguir NÃO está na 4FN:

CodProj	CodFunc	CodEquip
11	1001	A10
11	1002	A10
11	1001	A20
11	1002	A20
12	1001	A10
12	1001	A20

Passando para 4FN:

ProjFun	
CodProj	CodFunc
11	1001
11	1002
11	1001
11	1002
12	1001
12	1001

ProjEquip	
CodProj	CodEquip
11	A10
11	A10
11	A20
11	A20
12	A10
12	A20

DESNORMALIZAÇÃO

Desnormalização significa que você propositadamente não efetuou o design do banco de dados até a 3º Forma Normal. Isto é feito visando maximizar a performance ou para simplificar a manipulação dos dados para o usuário final. Sempre que você desnormalizar um banco de dados, você deve estar disposto a perder os benefícios ganhos através da 3º Forma Normal.

Nota: Você deve iniciar a partir da 3º FN. Se o problema com performance continuar, volte para a 2º FN ou 1ºFN. Tenha em sua mente que quando você desnormaliza um banco de dados, você o faz para uma aplicação específica, por exemplo um Data Warehouse.

Performance

Um banco de dados na 3º FN pode requerer mais " joins" para processar "queries" que na 2º ou 1º FN. Estes "joins" adicionais podem ser dispendiosos em termos de CPU ou I/O de disco.

Técnicas

Dados Duplicados – Podem reduzir o número de "joins" requeridos para processar uma "querie", assim reduzindo o consumo de CPD e I/O de disco.
Dados Resumidos – Melhoram o desempenho de uma "querie" por reduzir ou eliminar os passos requeridos para a sumarização dos dados.
Partição Horizontal – É a divisão de uma tabela em duas novas tabelas levando em consideração o número de registros, assim reduzindo o número de linhas por tabela.

Partição Vertical – É a divisão de uma tabela em duas novas tabelas levando em consideração o número de colunas, assim reduzindo o número de colunas por tabela.