



Introdução a Banco de Dados

Cláudio Leones Bazzi



Cuiabá-MT
2013

Presidência da República Federativa do Brasil
Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Diretoria de Integração das Redes de Educação Profissional e Tecnológica

© Este caderno foi elaborado pela Universidade Tecnológica Federal do Paraná - PR para a Rede e-Tec Brasil, do Ministério da Educação em parceria com a Universidade Federal do Mato Grosso.

Equipe de Revisão
Universidade Federal de Mato Grosso –
UFMT

Coordenação Institucional
Carlos Rinaldi

**Coordenação de Produção de Material
Didático Impresso**
Pedro Roberto Piloni

Designer Educacional
Alceu Vidotti

Designer Master
Daniela Mendes Piloni

Revisão Final
Francisco Rodrigues dos Santos

Ilustração
Maurício José Mota

Diagramação
Tatiane Hirata

Revisão de Língua Portuguesa
Patrícia Rahuan

**Universidade Tecnológica Federal do
Paraná – PR**

Coordenação Geral e-TEC
Edilson Pontarolo

Coordenação de Tecnologia na Educação
Henrique Oliveira da Silva

Coordenação de Curso
Eliane Maria de Bortoli Fávero

Projeto Gráfico
Rede e-Tec Brasil / UFMT

Dados Internacionais de Catalogação na Publicação

B349 Bazzi, Cláudio Leones

Introdução a banco de dados / Cláudio Leones Bazzi. – Curitiba: Ed. UTFPR, 2013.
91 p. : il.

Inclui bibliografia
e-ISBN: 978-85-7014-114-9

1. Banco de dados. 2. Modelagem de dados. 3. SQL (linguagem de programação de computador).
I. Título.

CDD (22. ed.) 005.74

Bibliotecário: Adriano Lopes CRB 9/1429

Apresentação Rede e-Tec Brasil

Prezado(a) estudante,

Bem-vindo(a) à Rede e-Tec Brasil!

Você faz parte de uma rede nacional de ensino que, por sua vez, constitui uma das ações do Pronatec - Programa Nacional de Acesso ao Ensino Técnico e Emprego. O Pronatec, instituído pela Lei nº 12.513/2011, tem como objetivo principal expandir, interiorizar e democratizar a oferta de cursos de Educação Profissional e Tecnológica (EPT) para a população brasileira propiciando caminho de acesso mais rápido ao emprego.

É neste âmbito que as ações da Rede e-Tec Brasil promovem a parceria entre a Secretaria de Educação Profissional e Tecnológica (Setec) e as instâncias promotoras de ensino técnico, como os institutos federais, as secretarias de educação dos estados, as universidades, as escolas e colégios tecnológicos e o Sistema S.

A educação a distância no nosso país, de dimensões continentais e grande diversidade regional e cultural, longe de distanciar, aproxima as pessoas ao garantir acesso à educação de qualidade e ao promover o fortalecimento da formação de jovens moradores de regiões distantes, geograficamente ou economicamente, dos grandes centros.

A Rede e-Tec Brasil leva diversos cursos técnicos a todas as regiões do país, incentivando os estudantes a concluir o ensino médio e a realizar uma formação e atualização contínuas. Os cursos são ofertados pelas instituições de educação profissional e o atendimento ao estudante é realizado tanto nas sedes das instituições quanto em suas unidades remotas, os polos.

Os parceiros da Rede e-Tec Brasil acreditam em uma educação profissional qualificada – integradora do ensino médio e da educação técnica – capaz de promover o cidadão com capacidades para produzir, mas também com autonomia diante das diferentes dimensões da realidade: cultural, social, familiar, esportiva, política e ética.

Nós acreditamos em você!

Desejamos sucesso na sua formação profissional!

Ministério da Educação
Dezembro de 2013

Nosso contato
etecbrasil@mec.gov.br



Indicação de ícones

Os ícones são elementos gráficos utilizados para ampliar as formas de linguagem e facilitar a organização e a leitura hipertextual.



Atenção: indica pontos de maior relevância no texto.



Saiba mais: oferece novas informações que enriquecem o assunto ou “curiosidades” e notícias recentes relacionadas ao tema estudado.



Glossário: indica a definição de um termo, palavra ou expressão utilizada no texto.



Mídias integradas: remete o tema para outras fontes: livros, filmes, músicas, *sites*, programas de TV.



Atividades de aprendizagem: apresenta atividades em diferentes níveis de aprendizagem para que o estudante possa realizá-las e conferir o seu domínio do tema estudado.



Refleta: momento de uma pausa na leitura para refletir/escrever sobre pontos importantes e/ou questionamentos.



Palavra do Professor-autor

A disciplina de Banco de Dados é importante para qualquer profissional da área de informática, tendo em vista que os softwares normalmente utilizam recursos que permitem a inclusão, atualização, exclusão e consulta de dados.

Um banco de dados corresponde a uma forma organizada e padronizada de gerenciar e manter um conjunto de informações que na maioria das vezes são incluídos pelo próprio usuário do sistema.

Antes mesmo do surgimento da informática, os bancos de dados já existiam, mas de forma bem menos eficaz do que os atuais. Estes bancos correspondem à caderneta da padaria, ao fichário da loja de calçados, entre outros. Verifique que, de certa forma, os dados (aqueles das mercadorias compradas e não pagas na padaria, calçados vendidos, bem como os dados dos clientes) eram armazenados em listas ou fichas para posterior consulta e baixa, quando o cliente viesse a realizar o pagamento.

Perceba ainda que as informações não eram simplesmente anotadas. Havia um sistema de busca (alfabético, data) para que pudesse ser encontrada com facilidade a ficha de cada cliente. Estas técnicas são válidas até hoje. No entanto, a partir do surgimento da informática, foram facilitadas e cada vez mais vêm sendo aperfeiçoadas, objetivando principalmente a velocidade e a segurança dos dados, além da capacidade de obter informações sobre os dados armazenados.

Este material objetiva dar suporte para que o aluno possa elaborar modelos de dados e implementá-los para soluções práticas no desenvolvimento de softwares.



Apresentação da Disciplina

Inúmeros Bancos de Dados surgiram no mercado, objetivando dar suporte aos sistemas de informação. O Sistema de Gerenciamento de Banco de Dados (SGDB) é quem controla o acesso aos dados, autentica os usuários, gerencia a manipulação dos dados, dá suporte a consultas, permite realizar cópias de segurança (Backup), entre outras tarefas. É importante notar que o SGDB corresponde a somente uma interface para gerenciamento dos dados. Dependendo do caso de aplicação, os dados devem ser organizados de uma forma válida para aquela situação em específico, ficando a cargo do profissional de banco de dados, criar estas estruturas de forma lógica para atender a necessidade de cada exigência.

Para isso, existe a modelagem de dados, a qual permite ao profissional da área organizar os dados, criar estruturas e aplicá-las na solução de problemas reais. Desta forma, este material fornece suporte para que o aluno possa realizar a modelagem de bancos de dados a serem aplicados em sistemas de informação.

Bom curso e sucesso!



Sumário

Aula 1. Modelagem de dados	15
1.1 Modelagem.....	17
Aula 2. Modelo Entidade Relacionamento	27
2.1 Modelo Entidade Relacionamento - MER.....	27
2.2 Definição do tipo de dados.....	33
Aula 3. Dicionário de dados	39
3.1 Nomenclatura do dicionário de dados.....	40
3.2 Dicionário de dados (modelo de dados da biblioteca).....	41
Aula 4. Normalização de dados	53
4.1 Formas normais.....	54
4.2 Dependência funcional.....	55
4.3 Normalização de dados (nota fiscal).....	56
4.4 Normalização de dados (Formulário Remanejamento).....	62
Aula 5. Linguagem SQL	71
5.1 Comandos DDL (Linguagem de Definição de Dados).....	71
5.2 Comandos DML (Linguagem de Manipulação de Dados).....	73
5.3 Comandos DCL (Linguagem de Controle de Dados).....	74
5.4 Comandos DTL (Linguagem de Transação de Dados).....	75
Palavras Finais	76
Guia de Soluções	77
Referências	89
Obras Consultadas	90
Currículo do Professor-autor	91



Aula 1. Modelagem de dados

Objetivos:

- apresentar conceitos básicos de modelagem de bancos de dados;
- dar suporte para que o aluno consiga modelar um banco de dados;
- propiciar recursos para que o estudante possa interpretar e construir o Diagrama Entidade Relacionamento - DER.

Caro (a) estudante,

Em nossa primeira aula, é importante que você tenha ciência de que um banco de dados é uma parte importante de todo software e que independe de linguagem de programação. Antes mesmo de serem criadas as telas de um software, é fundamental que ela já possua um banco de dados construído. Veja que é muito importante que a estrutura do banco de dados criada seja suficiente para armazenar tudo o que o software exigir, pois após a implementação das telas, caso haja necessidade de inclusão de algum dado necessário e não levantado anteriormente, o trabalho a ser executado trará muitos transtornos. Pense na construção de uma casa onde o pedreiro esqueceu de colocar os canos de esgoto, mas já rebocou as paredes e já colocou o piso. Um trabalho simples pode tornar-se demorado por falta de uma boa análise e levantamento de requisitos que o sistema deverá atender.



Figura 1
Fonte: ilustrador



Como se pode perceber, a criação do banco de dados está intimamente ligada à disciplina de análise de sistemas, a qual busca obter todas as informações que precisam ser armazenadas no banco de dados e qual é a lógica do negócio ao qual se tem intenção de desenvolver um sistema.

Além de não deixar faltar nada, faz-se necessário organizar as estruturas que receberão os dados da melhor forma possível para que seja facilitada a manutenção, a inserção, alteração, deleção e consultas, além de não permitir a inconsistência, como por exemplo a duplicidade de dados ou permitir o armazenamento de dados incompletos, como um cliente sem CPF (Cadastro de Pessoa Física).



Figuras 2 e 3

Fonte: ilustrador

Desta forma, é importante que cada contexto onde o software será inserido deve ser conhecido e bem avaliado pelo projetista ou analista que irá construir a estrutura de armazenamento onde ficarão os dados.



1.1 Modelagem

A construção de um modelo de dados conciso e que responda de forma satisfatória ao cliente, fornecendo um desempenho adequado, necessitando do mínimo de requisitos de hardware, é uma tarefa difícil e normalmente as empresas não estão bem servidas de profissionais que atendam a essas necessidades.

Técnicas como a normalização de dados podem facilitar a modelagem, mas tendo em vista que é um aspecto subjetivo do ponto de vista prático, a experiência conta muito. A modelagem exige a participação permanente dos usuários envolvidos para que se possa especificar e viabilizar uma solução apropriada. Reuniões entre representantes de departamentos, funcionários que irão utilizar o software e analistas são fundamentais para que nada passe despercebido.



Figuras 4

Fonte: ilustrador

A representação inicial de um modelo de dados corresponde ao Diagrama Entidade Relacionamento (DER) que é uma representação gráfica de um modelo de banco de dados. É composto por entidades e relacionamentos, indicando visualmente as regras de determinado modelo.

Para quem não está habituado aos termos de banco de dados, neste ponto já deve estar se perguntando: O que é uma entidade? O que é um relacionamento e o que ele representa? Neste sentido, iremos conceituar de forma objetiva, exemplificando estes termos:



1.1.1 Entidade

Corresponde a tudo aquilo que se deseja guardar dados, podendo ser concreto ou abstrato, e que é composta pelas características ou atributos que deverão ser armazenados no banco de dados. Por exemplo, em uma loja de roupas, caso se deseje modelar um banco de dados para armazenar os dados de compra e venda, obrigatoriamente necessitaríamos armazenar informações sobre os clientes, tais como nome, CPF, RG, endereço, telefone. Neste caso, tem-se a entidade cliente, na qual estão inseridos seus atributos (dados sobre cada cliente). A representação da entidade é realizada através de um retângulo, com o nome da entidade localizado em seu centro.

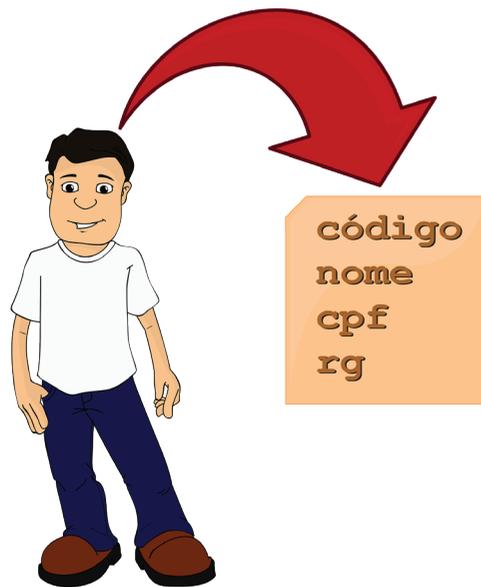


Figura 4

Fonte: ilustrador

Outra entidade neste exemplo corresponde aos dados das vendas. É evidente que o banco de dados armazene dados sobre todas as vendas realizadas. Sendo assim, outra entidade do modelo seria a entidade Venda, sendo esta uma entidade abstrata (gerada pelo contexto do negócio).

1.1.2 Atributo

O atributo corresponde a todas as características ou dados relacionados com a entidade e que se deseja guardar. Por exemplo, na entidade cliente é indispensável guardar o nome do cliente, o CPF, o endereço, etc. Perceba que cada um destes itens corresponde a um atributo da entidade cliente que será válido para cada cliente cadastrado. Em uma entidade os dados são organizados como em uma tabela, sendo que os atributos correspondem às colunas da tabela, e as linhas correspondem aos registros (um cliente por





exemplo), conforme apresentado na tabela 1.

Entidade Cliente

	Nome (atributo 1)	CPF (atributo 2)	Endereço (atributo 3)
Registro 1 ->	JOSÉ MARIA	904.343.333-00	RUA BAHIA
Registro 2 ->	JUCA LIMA	894.444.562-00	AVENIDA LISBOA
....
Registro 190 ->	WANDERLEI LIRA	878.444.332-09	RUA SÃO PAULO

Tabela 1 - Entidade Cliente com registros

Observe que cada coluna da entidade Cliente corresponde à representação de um dado relativo aos clientes. Neste caso, há três atributos (Nome, CPF, Endereço). Perceba que cada linha da entidade corresponde a um cliente diferente. Isto significa que os registros são representados por linhas e, a cada novo cliente, será inserida uma nova linha.

1.1.3 Relacionamento

Corresponde à representação que indica qual é a relação entre uma entidade e outra. Por exemplo, indicamos que teríamos a entidade cliente e a entidade venda para o exemplo da loja. O relacionamento indica justamente que a entidade cliente tem alguma relação com a entidade venda. Neste ponto, faz-se necessário o entendimento de outro conceito importante em banco de dados: o conceito de Cardinalidade.

1.1.4 Cardinalidade

Corresponde ao grau de relação entre duas entidades. No exemplo, pode-se definir que um cliente pode ter várias vendas relacionadas com ele e que uma venda é realizada para um único cliente. Perceba que a cardinalidade se baseia na seguinte questão: uma linha (registro) de uma tabela (Cliente) está relacionada com quantas linhas da outra tabela (Vendas). Como se pode verificar, dado que cada venda é registrada em uma linha da entidade Vendas, então pode-se dizer que um cliente está relacionado com várias vendas.

Visualizando o mesmo relacionamento, mas por outro ângulo, tem-se que uma linha (ou registro) de venda relaciona-se somente com um cliente. Ou seja, em uma venda, podemos relacionar somente um cliente, e não mais do que isso. Por este motivo, dizemos que temos uma relação “um para um”, em que uma venda se relaciona com somente um cliente. Perceba que o relacionamento entre duas entidades é dado sempre por um atributo. Neste



caso é utilizado o atributo **Codigo_Cliente**. O relacionamento e a cardinalidade é representada na figura 5.

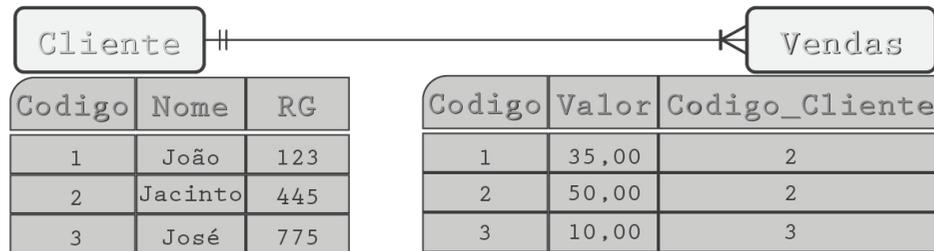


Figura 5 - Relacionamento entre a entidade Cliente e a Entidade Vendas.

Fonte: autor (adaptado pelo ilustrador)

Conforme apresentado na figura 6, têm-se as duas entidades (cliente e vendas) representadas por retângulos. A linha que interliga as duas entidades é chamada de relacionamento, e o conjunto todo, juntamente com a representação da cardinalidade, é definido como Diagrama Entidade Relacionamento – DER. Perceba que as entidades não são representadas com seus atributos inclusos. Caso isso ocorra, define-se o diagrama como Modelo Entidade Relacionamento (MER).



Figura 6 - Estrutura de um Diagrama Entidade Relacionamento - DER

Fonte: autor (adaptado pelo ilustrador)

O DER tem como objetivo identificar as entidades essenciais na organização dos dados e a forma com que elas estão relacionadas umas com as outras. Pelo que podemos perceber, corresponde a uma importante ferramenta de modelagem, usada para definir as informações necessárias ao modelo de entidade-relacionamento (MER), o qual é uma visualização mais real do modelo e que será visto num passo posterior.

O DER permite uma visualização ampla do projeto do banco e das formas de armazenamento dos dados, evitando a redundância, além de facilitar a identificação das entidades e relacionamentos existentes entre elas.

Em 1976, Peter Chen desenvolveu a E-R (Entidade e Relacionamento) que surgiu da necessidade de representar um Banco de Dados de uma forma visível mostrando todas as entidades, seus atributos e cardinalidade. Este método trata da simbologia geral do BD, sendo:



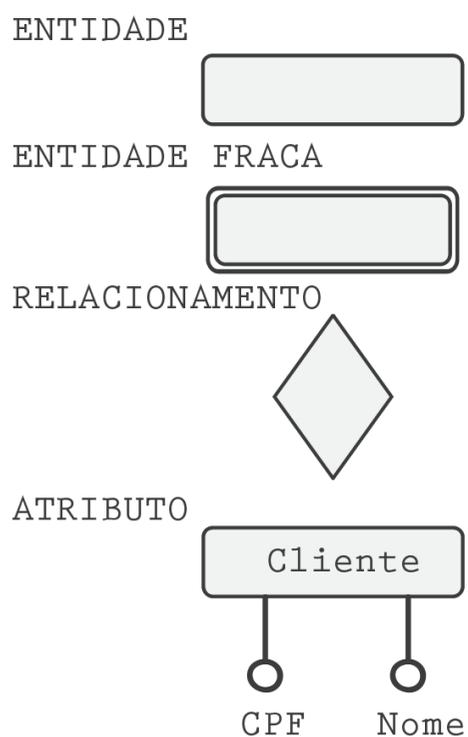


Figura 7

Fonte: autor (adaptado pelo ilustrador)

ENTIDADE FRACA – São entidades que só passarão a existir sob influência de outra entidade. Exemplo:



Figura 8

Fonte: autor (adaptado pelo ilustrador)

Neste caso, o dependente só passará a existir no sistema se ele estiver relacionado com um funcionário, sendo impossível a existência dele por outros meios.

As interações que existem entre duas entidades são descritas dentro do relacionamento (losango), dando um entendimento maior sobre a influência que uma entidade exerce sobre a outra (figura 9).

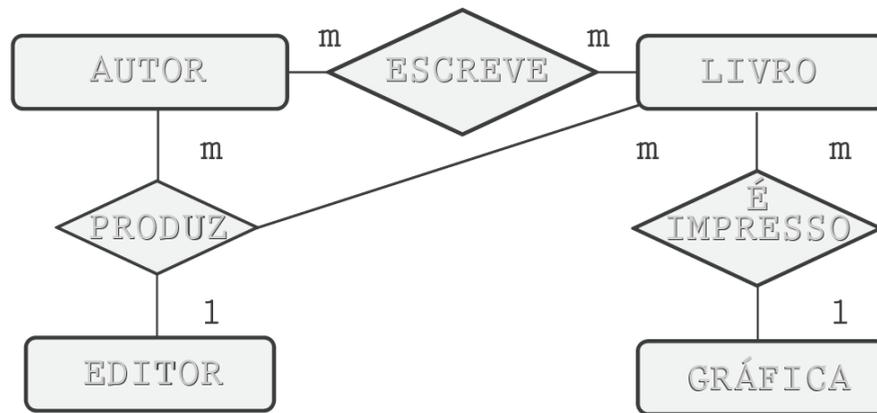


Figura 9 - Exemplo do DER apresentado por Peter Chen

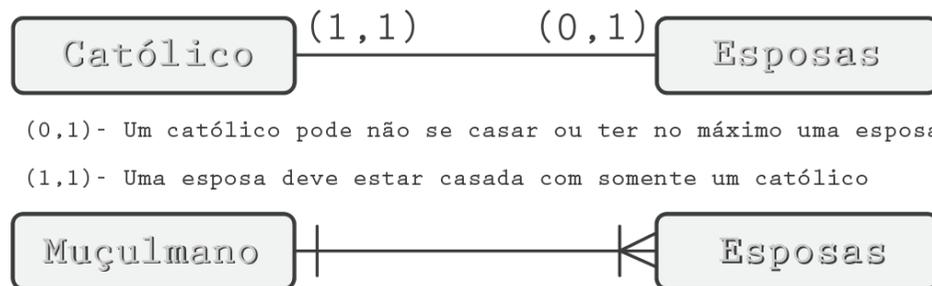
em que:

m - m = muitos para muitos, ou seja, uma linha de uma tabela A deve se relacionar com várias colunas da tabela B e uma linha da tabela B deve se relacionar com várias linhas da tabela A;

1 - m = um para muitos, ou seja, uma linha da tabela A deve se relacionar com várias linhas da tabela B, mas uma linha da tabela B se relaciona com uma única linha da tabela A;

Fonte: autor (adaptado pelo ilustrador)

Na década de 80, Charles Bachman e James Martin desenvolveram uma nova simbologia para realizar a representação do relacionamento extraíndo o losango. Neste caso, simplifica-se o Diagrama Entidade Relacionamento, em que o relacionamento é representado somente por uma linha. A cardinalidade é representada próxima à conexão da linha nos retângulos que representam as entidades.



(0,1)- Um católico pode não se casar ou ter no máximo uma esposa

(1,1)- Uma esposa deve estar casada com somente um católico

- Um muçulmano pode ter uma ou várias esposas

- Uma esposa deve estar casada com apenas um muçulmano

Figura 10 - Representação de cardinalidade mínima e máxima no modelo proposto por Charles e James

Fonte: autor (adaptado pelo ilustrador)

A representação de cardinalidade pode ser dada de várias maneiras, dependendo da relação existente entre as entidades, sendo:

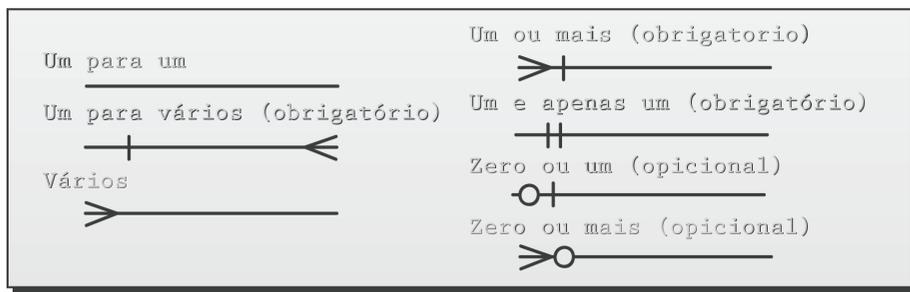


Figura 11

Fonte: autor (adaptado pelo ilustrador)

A figura 12 mostra um modelo simples para um banco relativo a uma videolocadora, em que se apresenta a aplicação dos tipos de cardinalidade existentes. É importantíssimo notar que um relacionamento nunca deve possuir uma relação de muitos para muitos, pois, para os conceitos apresentados neste caso, devem ser criadas entidades associativas, como é o caso da entidade **Item_Reserva** e **Item_Locacao**.

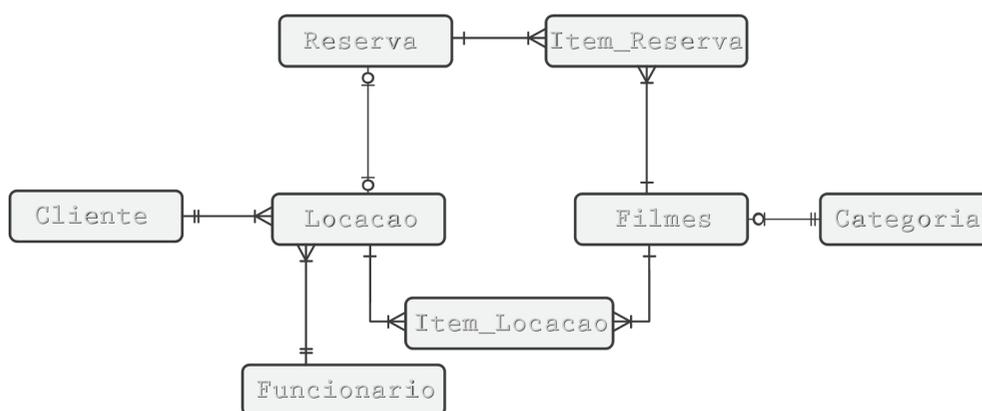


Figura 12 - Modelo de dados de uma videolocadora

Fonte: autor (adaptado pelo ilustrador)

Sobre o modelo construído para a videolocadora, pode-se dizer que:

- Um funcionário é responsável por uma ou várias locações, ao passo que uma locação é relacionada com apenas um funcionário;
- **Item_Locacao** é uma tabela associativa, onde haverá um código de locação e um código de filme para cada linha da tabela, podendo relacionar vários filmes com uma única locação, e um filme com varias locações;



- **Item_Reserva** trata-se da mesma situação (**item_locacao**). Um filme pode ser reservado várias vezes, e uma reserva pode conter vários filmes;
- Uma locação pode estar ou não relacionada com uma reserva.

Resumo

Nesta aula você aprendeu sobre conceitos de estruturas importantes presentes nos bancos de dados: entidades correspondem a tudo o que se deseja guardar dados; os relacionamentos correspondem às relações existentes entre as entidades; a cardinalidade diz qual é o grau de relação no relacionamento de duas entidades; os atributos são as características relativas ao que se quer guardar dados; e os registros são os dados armazenados em uma entidade. É importante perceber que são com estes conceitos que um banco de dados é construído, adequando-os a cada contexto que se deseja criar um banco de dados. A organização do contexto que se deseja modelar, considerando os conceitos apresentados, fornecem suporte para elaboração do Diagrama Entidade Relacionamento – DER que é um esboço pouco refinado do banco de dados final, e ao Modelo Entidade Relacionamento – MER que corresponde ao modelo que será implementado ao software, contendo os atributos de cada entidade, assim como as características apresentadas pelo DER.



Atividades de Aprendizagem

1. Defina entidade e apresente dois exemplos práticos.
2. Defina relacionamento e apresente dois exemplos práticos.
3. Quais entidades você definiria para a construção de um diagrama de dados para uma oficina mecânica?
4. Construa um Diagrama Entidade Relacionamento (DER), para servir de base para um sistema de uma loja de calçados. Sabe-se que a loja só vende calçados, tem cadastro de clientes, os funcionários da loja vendem à vista e à prazo (30, 60 e 90 dias). Os clientes inadimplentes não podem comprar se possuem contas em atraso.

Ao fim desta aula, fica clara a importância do analista no processo de criação de um banco de dados, o qual deve ter envolvimento com o processo ou contexto. Por meio da análise, deverão ser definidos todos os itens que se deseja guardar dados (entidade) e avaliar como se relacionam. Fica claro que



erros no projeto do banco de dados acabam por prejudicar toda a estrutura e desenvolvimento do mesmo, devendo-se ter o máximo de cuidado para sua criação. A modelagem corresponde à transformação do contexto onde o sistema irá ser inserido em entidades, relacionamentos, criando o DER. Na próxima aula trataremos da elaboração do Modelo Entidade Relacionamento - MER, que complementa o DER.





Aula 2. Modelo Entidade Relacionamento

Objetivos:

- definir o Modelo Entidade Relacionamento (MER) utilizando-se de casos reais; e
- dar suporte ao entendimento de conceitos e procedimentos para elaboração de modelos de dados.

Caro (a) estudante,

Nesta aula daremos continuidade ao conteúdo de modelagem de dados, incorporando ao Diagrama Entidade Relacionamento - DER características relativas ao contexto que se deseja modelar. Ao final desta aula, o aluno deverá estar apto a construir um Modelo Entidade Relacionamento (MER) para um contexto onde pretende-se realizar a construção de um banco de dados.

O Modelo Entidade Relacionamento (MER) corresponde a um diagrama mais detalhado ou mais próximo do modelo real que será implementado. Como vimos, o DER é uma forma gráfica de representação e está limitado a apresentar um esquema de todas as entidades, seus relacionamentos e a cardinalidade entre eles.

Perceba que o DER não contempla os atributos do modelo, ou seja, o MER complementa o DER, informando o nome dos atributos que cada entidade contempla, bem como quais atributos são utilizados para realizar o relacionamento entre eles.

2.1 Modelo Entidade Relacionamento - MER

Antes de iniciarmos, é preciso entender dois conceitos importantes sobre chave primária e chave estrangeira, que fazem parte do MER.



2.1.1 Chave primária

Relembrando que as estruturas de armazenamento do banco de dados são dadas como tabelas (com linhas e colunas), a chave primária corresponde ao atributo da entidade (coluna da tabela) que representa de forma única um registro, ou seja, nunca para uma entidade teremos dois registros (linhas da tabela) com o mesmo valor para o atributo chave primária. Por exemplo, no Brasil, temos o CPF como identificador único para cada pessoa. Considerando que tivéssemos a entidade **BRASILEIROS**, como não possuímos duas pessoas com o mesmo CPF, poderíamos ter o CPF como chave primária para esta entidade. Neste caso, cada brasileiro cadastrado no banco ocuparia uma linha da tabela e todas as linhas teriam um número de CPF diferente, pois cada linha representa um brasileiro.

Perceba que não poderíamos utilizar o nome de brasileiros como chave primária, tendo em vista que podemos ter várias pessoas com o mesmo nome e, quando falássemos do brasileiro “**João da Silva**”, não saberíamos realmente a qual linha do banco de dados ele está associado, considerando que poderíamos ter vários brasileiros com este nome.

Você deve estar se perguntando: posso ter vários “**João da Silva**”, mas moram em lugares diferentes, por meio do local de moradia, pode-se distinguir um do outro! Não poderia? A resposta para essa pergunta é **SIM**. Com certeza, você poderia relacionar com outros atributos e identificar qual “**João da Silva**” você está procurando.

Neste caso, utiliza-se o conceito de chave primária composta. A chave primária composta visa relacionar vários atributos para obter uma chave primária. Por exemplo, à entidade **BRASILEIROS** poderíamos utilizar os atributos **NOME, DATA DE NASCIMENTO**. Mas pensando bem, não seria suficiente, pois pode-se ter dois “**João da Silva**” nascidos no mesmo dia. Poderíamos então, utilizar mais atributos, tais como **NATURALIDADE** e **NOME DA MÃE**. Neste caso, verifique que nunca teríamos duas pessoas, nascidas no mesmo dia, na mesma cidade e com nomes de mãe idênticos. Pode-se, então, definir este conjunto de atributos como chave primária. A chave primária é representada por um asterisco (*).

2.1.2 Chave estrangeira

A chave estrangeira é aquela utilizada para realizar o relacionamento entre entidades. Supondo duas entidades A e B relacionadas, um atributo de A é incluído dentro da entidade B, registrando que este atributo corresponde a





uma referência para indicar um registro de A que faz referência em B. Por exemplo: supondo a entidade A (TB_ENDERECO), relacionada com a entidade B (TB_CLIENTE), dizemos que o relacionamento é realizado através do atributo **END_CODIGO** da tabela A, sendo referenciado na tabela B como **CLI_CODENDERECO**. Tem-se então, que quando cadastrado um cliente, deve-se referenciar o código do endereço previamente cadastrado na tabela TB_ENDERECO.

Outro exemplo, supondo o caso de uma videolocadora. Temos as entidades **Cliente** e **Locacao** que são relacionadas (figura 13).



Figura 13 - Relacionamento entre as entidades Cliente e Locação

Fonte: autor (adaptado pelo ilustrador)

É importante verificar que o relacionamento é realizado através de atributos e que, em nosso exemplo corresponde ao atributo **CLI_CODIGO**, que é chave primária na entidade **CLIENTE**. Para que seja realizada uma Locação, é necessário indicar quem é o cliente que a está realizando. Neste caso, para que isso seja possível, incluímos na entidade **LOCACAO** um atributo chamado **LOC_CODCLIENTE**, o qual indica o cliente, incluído na tabela de **CLIENTE** que está realizando a locação. Perceba que este atributo (**LOC_CODCLIENTE**) corresponde ao atributo de relacionamento entre as entidades. Como o atributo **LOC_CODCLIENTE** corresponde a um atributo disposto na outra entidade, este atributo corresponde a uma chave estrangeira. Uma chave estrangeira é normalmente representada pelo símbolo (#).

Considerando o exemplo da videolocadora, iremos agora definir quais atributos cada uma das entidades possui, bem como quais são chave primária e estrangeira.

Na entidade Cliente, como chave primária, utilizaremos um atributo **CLI_CODIGO**, que será gerado sequencialmente a cada novo cliente cadastrado, sendo o primeiro cliente com código 1, o segundo como 2, e assim por diante. Outro atributo importante para a entidade corresponde ao **CLI_NOME**, que armazenará o nome do cliente cadastrado. Entre os demais atributos, tem-se, **CLI_RG**, **CLI_CPF**, **CLI_FONE**, **CLI_CELULAR**. Outro atributo que se faz necessário é referente ao endereço do cliente. Mas é preciso tomar cui-



dado com essa situação. Perceba que o endereço do cliente está atrelado à rua em que ele mora, ao bairro, à cidade e ao número da casa. Note ainda que, em uma rua, podem-se ter vários clientes; no mesmo bairro e na mesma cidade também isso ocorre. Neste sentido, como uma rua, um bairro e uma cidade são atributos que podem se repetir várias vezes (em várias linhas) na entidade **CLIENTE**, define-se uma nova entidade para cada um destes componentes do endereço do cliente e as relacionamos com cada cliente cadastrado. Desta forma, não necessitaríamos digitar várias vezes o nome da "Rua Bahia", considerando que tivéssemos vários clientes que moram nela. Incluiríamos o registro de nome "RUA BAHIA" na entidade Endereço e relacionaria a mesma ao cliente através do atributo END_CODIGO (chave primária), na entidade **CLIENTE**.

Dando continuidade a este raciocínio, devemos criar as entidades **ENDE-RECO**, **BAIRRO**, **CIDADE** e **CLIENTE**, com os atributos relacionados, como podemos verificar na figura 14.

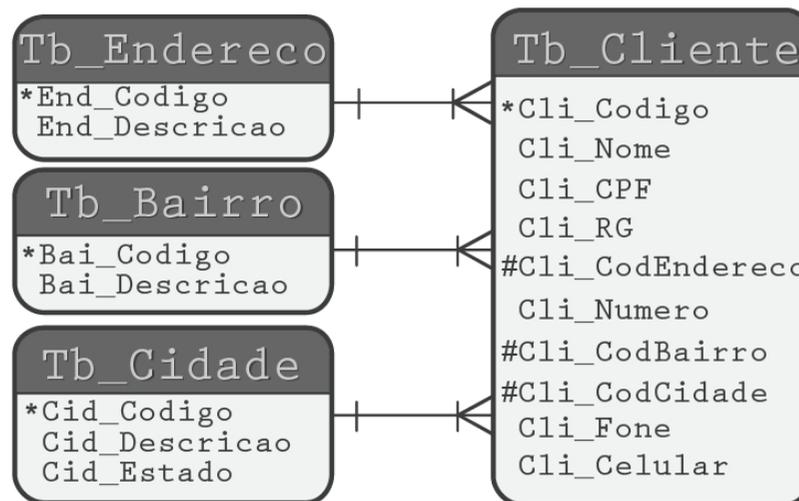


Figura 14 - Entidades, atributos e relacionamentos

Fonte: autor (adaptado pelo ilustrador)

A entidade Locação também é importante, tendo em vista que nesta serão realizados os cadastros de todas as locações realizadas. Vale a pena observar que temos também relacionada com ela outra entidade chamada **ITEM_LOCACAO**, a qual irá armazenar os itens locados. Isso se faz necessário, pois uma locação pode ter vários itens locados, e a entidade **LOCACAO** não teria condições de armazenar mais de um item locado, tendo em vista que não se sabe quantos itens uma locação terá e não se pode destinar várias colunas para esse tipo de registro, considerando que não se tem uma certeza de

quantos itens podemos ter. Neste caso, cada novo ítem de uma locação será incluído na entidade **ITEM_LOCACAO**, considerando que, para cada novo ítem incluso, teremos um novo registro na tabela (uma nova linha).

Para identificação dos itens a serem locados, devemos possuir uma entidade chamada **FILMES**, considerando que serão armazenados todos os filmes que a locadora possui. Perceba que não se terá um controle de cada exemplar que a locadora possui, e sim somente um cadastro referenciando qual é o filme e qual é a quantidade de exemplares. Verifique como está nosso modelo até o momento (figura 15).

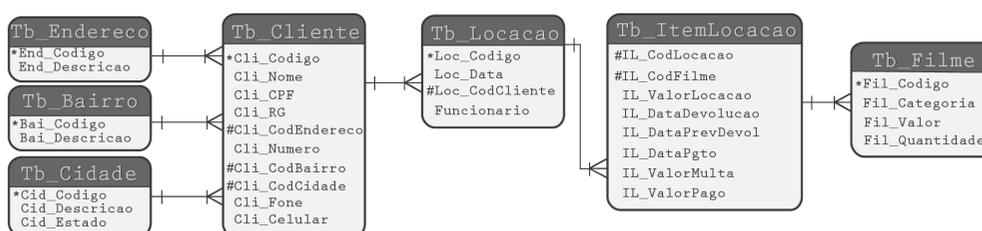


Figura 15 - Modelo de dados parcial.

Fonte: autor (adaptado pelo ilustrador)

Trabalharemos incluindo os dados gerais de uma locação na tabela **TB_LOCACAO** e os dados referentes aos filmes locados deverão ser inseridos na tabela **TB_ITEMLOCACAO**, considerando que a cada novo registro, teremos um novo ítem incluso na locação.

Perceba que a tabela **TB_LOCACAO** possui um atributo (*Loc_CodCliente*) que faz o relacionamento com a entidade **TB_CLIENTE**, considerando que se faz necessário, tendo em vista que, na entidade locação, necessitamos indicar qual é o cliente que está locando filmes. O atributo **Funcionario** na entidade **TB_LOCACAO** é importante para indicar quem foi o funcionário que realizou a locação. Perceba ainda que a entidade **TB_LOCACAO** não possui o valor total da locação, considerando que como se trata de um campo calculado a partir dos registros da tabela **TB_ITEMLOCACAO**, não há necessidade de criar campos adicionais.

A entidade **TB_FILME** é utilizada para registrar dados de todos os filmes da locadora, sendo o campo quantidade utilizado para indicar qual é a quantidade de exemplares de determinado filme. O campo **Valor** indica qual é o valor corrente do filme e o atributo **Categoria** indica qual é a categoria do filme.



A entidade **TB_ITEMLOCACAO** indica quais os itens locados em uma determinada locação, fazendo a relação entre as entidades **TB_LOCACAO** e **TB_FILME**, através dos atributos **IL_CodLocacao** e **IL_CodFilme**. O atributo **IL_Valor** corresponde ao valor do filme locado no momento da efetivação da locação. Perceba que esse valor é praticamente o mesmo encontrado na entidade **TB_FILME**, mas se diferencia por ser um valor fixo e inalterável, enquanto o atributo presente na entidade **TB_ITEMLOCACAO** corresponde ao valor corrente de aluguel de determinado filme. Na entidade **TB_ITEMLOCACAO**, foi incluído um atributo **Loc_DataPgto** que indica a data em que cada ítem da locação foi pago, considerando que não necessariamente necessita ser paga no mesmo dia em que ocorreu a locação e cada ítem pode ter sido devolvido e pago em uma data distinta. Por meio deste atributo, pode-se verificar que as locações que ainda não foram pagas, terão este atributo definido como NULL (vazio).

Perceba que, na entidade **TB_LOCACAO**, tem-se o atributo **Funcionario**, que, se for atribuído a um campo texto, pode-se digitar o nome do mesmo. Verifique que, se em toda locação for necessário digitar o nome do funcionário, se tornará uma operação incômoda e possivelmente inconsistente, tendo em vista que poderia ser digitado o nome erroneamente, algumas vezes, ou poderia ainda incluir nomes com todas as letras maiúsculas ou minúsculas. Neste sentido, para facilitar o trabalho e diminuir o espaço gasto para armazenar o nome do funcionário que fez uma locação, iremos criar uma entidade chamada **TB_FUNCIONARIO** e passaremos a relacioná-la com a entidade **TB_LOCACAO** pelo atributo **Loc_CodFuncionario**, localizado dentro da entidade **TB_LOCACAO**.

Iremos fazer isso também na entidade **TB_FILME**, em que iremos criar outra entidade chamada **TB_CATEGORIA** para cadastrar todas as categorias possíveis e relacioná-las através do atributo **Fil_CodCategoria** localizado na entidade **TB_FILME** como podemos verificar na figura 10.

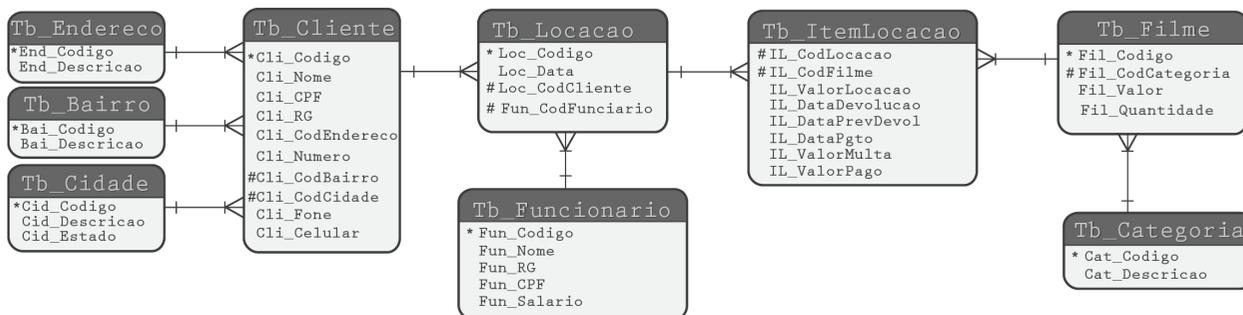


Figura 16 - Modelo de dados final

Fonte: autor (adaptado pelo ilustrador)





2.2 Definição do tipo de dados

Depois de elaborado o modelo de dados, é hora de definir qual é o tipo de dados que deverá ser utilizado para cada um dos atributos definidos no modelo. Esta é uma tarefa trabalhosa, necessária antes da criação do banco de dados, efetivamente.

Durante a criação das tabelas de dados, deve-se definir o tipo de dados para cada coluna e a escolha incorreta pode ocasionar problemas sérios, talvez não momentâneos, mas a partir de um tempo ou com mudanças na estrutura da empresa.

Dependendo do Sistema Gerenciador de Banco de Dados, certos tipos de atributos não são suportados. Neste sentido, é importante ater-se neste detalhe, assinalando que os tipos de dados apresentados neste material dizem respeito ao Sistema Gerenciador de Banco de Dados Oracle 10g.

2.2.1 Dados alfanuméricos

O tipo de dados alfanuméricos são os mais comumente utilizados, pois além de permitirem o armazenamento de dados referentes a caracteres alfanuméricos, incluindo números, acentuação e pontuação. Os tipos de dados alfanuméricos, conforme Oracle (2006), correspondem a:

- VARCHAR2 - Caracteres de dados de tamanho variado: normalmente o mais utilizado e possui uma característica de capacidade importante, podendo armazenar de 1 byte a 4 KB;
- NVARCHAR2 – Mesmas características do VARCHAR2, mas com capacidade de armazenar caracteres especiais como os caracteres chineses, os quais são diferentes do padrão Unicode;
- CHAR – Caracteres de dados de tamanho fixo: utilizado para armazenar dados de tamanho fixo, tais como siglas de estado (com dois caracteres) e possui capacidade de armazenamento de 1 byte a 2 KB.

Para representação de dados do tipo binário é utilizado o tipo RAW (Dados binários de tamanho variável) que não segue os padrões VARCHAR2 e NVARCHAR2, não mantendo padrão nenhum tipo de caractere em especial.



2.2.2 Dados numéricos

Segundo Oracle (2006), para armazenamento de dados numéricos, temos todos os tipos de tamanho variável:

- NUMBER – Dados numéricos: pode-se identificar a precisão e a escala, sendo de 1 a 38 caracteres e uma escala de 127 caracteres;
- FLOAT – Dados ponto flutuante: pode representar valores ponto flutuante com precisão de 38 caracteres. O Oracle provê os tipos BINARY_FLOAT e BINARY_DOUBLE como tipos alternativos;
- INTEGER – Dados inteiros: equivalente ao NUMBER, mas com escala zero.

2.2.3 Dados tipo data

Para tratar de dados referentes a datas, a Oracle (2006) prevê várias situações úteis para os usuários de bancos de dados. Além dos tradicionais tipos de dados Date e Timestamp, são previstos dados do tipo Timestamp com Timezone e intervalos entre datas. Para armazenamento de dados do tipo data e hora, temos:

- DATE – Data: possui tamanho zero se o campo estiver vazio, mas sete bytes quando estiver completo. O tipo de dado DATE possui informações sobre século, ano, mês, dia, hora, minuto e segundo. O intervalo válido é de primeiro de janeiro de 4712 antes de cristo e 9999 dC.
- TIMESTAMP – Data e hora: possui tamanho zero quando o campo estiver vazio, mas pode conter até onze bytes, de acordo com a precisão especificada. Praticamente igual a DATE, mas a precisão de até nove casas decimais para o segundo, sendo seis casas o padrão.
- TIMESTAMP COM TIMEZONE – Como o TIMESTAMP, mas com os dados é guardado um registro que armazena o fuso horário a que se refere. O tamanho pode chegar a treze bytes, dependendo da precisão. Permite identificar a diferença de dois tempos, mesmo estando em fusos horários diferentes.
- TIMESTAMP COM TIMEZONE LOCAL – Parecido com o TIMESTAMP, mas os dados são normalizados para um fuso horário padrão.
- INTERVALO DE ANOS PARA MESES – Utilizado para armazenar registros





referentes a um período de anos em meses para dois dados DATE ou TIMESTAMP.

- INTERVALO DE DIAS EM SEGUNDO - Utilizado para armazenar registros referentes a um período de dias em segundos para dois dados DATE ou TIMESTAMP.

2.2.4 Dados de tamanho elevado

Para armazenamento de dados grandes, são utilizados dados com características diferenciadas, apesar de armazenarem dados comuns, como por exemplo alfanuméricos. A Oracle (2006) definiu alguns tipos de dados para tratar esse tipo de problema:

- CLOB – dados do tipo caractere, que permitem a utilização de grandes quantidades de dados, sendo até 4GB.
- NLOB – dados parecidos com o CLOB, mas utilizado para caracteres especiais diferentes do padrão Unicode.
- BLOB – parecido com o CLOB, mas utilizado para dados binários.
- BFILE – ponteiro de localização que aponta para um arquivo de até 4 GB armazenado no servidor.
- LONG – dados de caractere, mas para grande quantidade de dados. Cada tabela pode ter somente um campo deste tipo.
- LONGRAW – parecido com o tipo LONG, mas para dados binários. Uma coluna deste tipo pode ser convertida para BLOB.
- ROWID – corresponde a um valor codificado na base 64, que é um ponteiro em uma linha na tabela. É o endereço físico e possui uma propriedade que não é visível, salvo quando especificamente selecionado.

Resumo

Tivemos a oportunidade de aprender nesta aula que o Modelo Entidade Relacionamento (MER) corresponde a um diagrama mais detalhado ou mais próximo do modelo real que será implementado. Ressaltamos que o DER é uma forma gráfica de representação e está limitado a apresentar um esquema de todas as entidades, seus relacionamentos e a cardinalidade entre eles.





Conceituamos chave primária, que corresponde ao atributo da entidade (coluna da tabela) que representa de forma única um registro, ou seja, nunca para uma entidade temos dois registros (linhas da tabela) com o mesmo valor para o atributo chave primária. Por exemplo, no Brasil, temos o CPF como identificador único para cada pessoa.

Observamos, também, o conceito de chave estrangeira, que é aquela utilizada para realizar o relacionamento entre entidades. Supondo duas entidades A e B relacionadas, um atributo de A é inserido na entidade B, registrando que este atributo corresponde a uma referência para indicar um registro de A que faz referência em B. Por exemplo: supondo a entidade A (TB_ENDERECO), relacionada com a entidade B (TB_CLIENTE), dizemos que o relacionamento é realizado através do atributo **END_CODIGO** da tabela A, sendo referenciado na tabela B como **CLI_CODENDERECO**. Tem-se, então, que quando cadastrado um cliente, deve-se referenciar o código do endereço previamente cadastrado na tabela TB_ENDERECO. Por fim, vimos que depois de elaborado o modelo de dados, é preciso definir qual é o tipo de dados que deverá ser utilizado para cada um dos atributos definidos no modelo. No entanto, dependendo do Sistema Gerenciador de Banco de Dados, certos tipos de atributos não são suportados. Neste sentido, é importante ater-se neste detalhe, assinalando que os tipos de dados apresentados neste material dizem respeito ao Sistema Gerenciador de Banco de Dados Oracle 10g.



Atividades de Aprendizagem

1. Construir os modelos de dados conforme casos hipotéticos:

a) Caso hipotético 1:

Em uma clínica trabalham médicos e existem pacientes internados. Cada médico é identificado pelo seu CRM, possui um nome e recebe um salário na clínica. Um médico tem formação em diversas especialidades (ortopedia, traumatologia, etc), mas só exerce uma delas na clínica. Para todo paciente internado na clínica são cadastrados alguns dados pessoais: nome, RG, CPF, endereço, telefone(s) para contato e data do nascimento. Um paciente tem sempre determinado médico como responsável (com um horário de visita diário predeterminado), porém, vários outros médicos podem participar de seu tratamento. Pacientes estão sempre internados em quartos individuais, que são identificados por um número e ficam em um andar da clínica.





b) Caso hipotético 2:

Elabore um modelo de dados para suprir a necessidade de um sistema que vise atender a uma oficina mecânica, onde são realizados orçamentos e serviços. Os clientes normalmente realizam o orçamento; sendo autorizado pelo cliente, o serviço é efetuado. O sistema deve controlar as peças (estoque), bem como manter, em banco, a relação de compras efetuadas. Um serviço ou uma peça podem ter garantia, ressaltando que o sistema deve controlar esse fato. Os mecânicos recebem comissão sobre seus serviços e possuem um salário fixo também. Deve-se ter uma forma de manter um histórico de tudo o que foi elaborado em determinado veículo para facilitar a identificação de novos problemas.

Nesta etapa, você deve estar refletindo como o banco de dados ficará estruturado quando implementado. Perceba que apesar da fácil interpretação dos modelos criados, são muitos detalhes que precisam ser seguidos para implementação e manipulação dos dados a serem armazenados em um banco de dados. Pense agora em modelar um sistema com vinte ou trinta entidades, dar suporte ao mesmo por um período e passar a tarefa para outro funcionário da empresa por você ter sido promovido. É impossível que este novo funcionário saiba todos os detalhes do banco que você implementou e até com o passar do tempo, fica difícil para você mesmo relembrar de detalhes importantes, como tipos de dados, nome de atributos e entidades, restrições de integridade, entre outros. Neste sentido, é de suma importância que sejam gravados dados sobre os dados (também chamados de metadados) e que correspondem a relatar cada detalhe do banco, incluindo entidades, atributos, relacionamentos, restrições de integridade, tipos de dados, entre outras coisas. Desta forma, qualquer pessoa que estiver em dúvida sobre algum detalhe do modelo, poderá consultar esta documentação e sanar suas dúvidas. O próximo capítulo trata do dicionário de dados e expõe uma forma organizada de documentar o modelo de dados.



Aula 3. Dicionário de dados

Objetivos:

- entender a importância de um Dicionário de Dados, tanto na criação como na manutenção de um banco de dados; e
- tornar o aluno capaz de elaborar dicionário de dados para modelos de dados aplicados a situações reais.

Caro (a) estudante,

Na definição das entidades, atributos e restrições para o modelo de dados, são elencados nomes que normalmente sugerem o que se deseja armazenar. Apesar disso, nomes tidos como "inconfundíveis" podem confundir um profissional que não esteve totalmente inserido na criação do modelo. Uma entidade neste contexto refere-se a Cliente, que mesmo para um leigo, deve armazenar dados de um cliente. Mas agora pense em uma situação inusitada! No caso de modelarmos um banco de dados para atender a necessidade de uma funerária. Quem é o cliente neste caso? O defunto? O parente mais próximo que deu entrada na funerária? Perceba que se torna uma situação confusa para quem necessita entender o modelo de dados criado. Neste sentido, o dicionário de dados permite você documentar todo o modelo, fazendo com que não reste dúvida de quem é o Cliente neste caso.

O dicionário de dados corresponde a uma descrição detalhada da estrutura que o banco de dados possui. Nesta aula apresentaremos um modelo de Dicionário de Dados, tendo como base um modelo de dados de uma Biblioteca, cujo Modelo Entidade Relacionamento (MER) é representado de acordo com a figura 17. Como dito, este é um modelo e pode sofrer alterações conforme a necessidade.

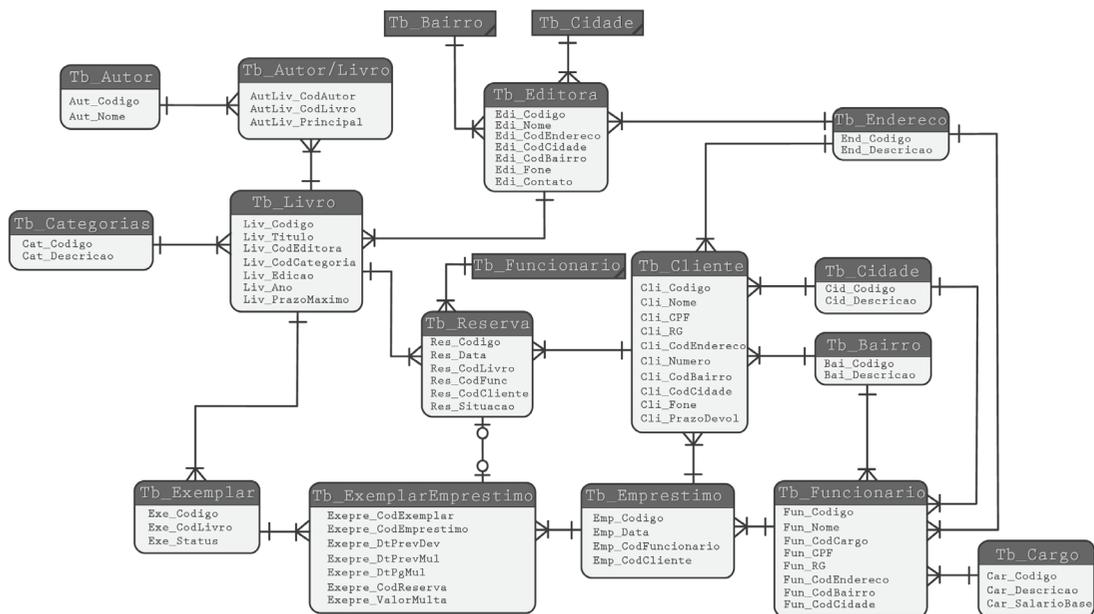


Figura 17 - Modelo de dados (biblioteca)

Fonte: autor (adaptado pelo ilustrador)

3.1 Nomenclatura do dicionário de dados

Antes de nos aprofundarmos no exemplo da figura 11 (biblioteca), é preciso entender que um dicionário de dados apresenta formas de caracterizar cada um dos principais elementos do Modelo Entidade Relacionamento (MER), relativos as entidades, relacionamentos e atributos. Neste sentido, para cada Entidade do modelo são apresentados três comentários sobre ela, devendo compreender:

a) Entidade: Caracterização da entidade de forma geral e abrangente.

Nome	Nome da entidade
Sigla	Sigla ou nome abreviado
Descrição	Descrição do conteúdo da Entidade no contexto do projeto, deixando evidente o que poderá ser registrado na entidade e o que não poderá estar, caso isso seja necessário.

b) Relacionamentos: Indicação de quais relacionamentos a entidade em questão possui, indicando características de cardinalidade, atributo e nome do relacionamento (quando for o caso)

Com a Entidade	Nome abreviado da entidade com a qual existe um relacionamento
Cardinalidade	Tipo do relacionamento, indicativo da cardinalidade do relacionamento, no formato X:Y em que X = cardinalidade na entidade em descrição, podendo ser: 0, 1 ou M (muitos) Y = cardinalidade na outra entidade, podendo ser 0, 1 ou M



Atributo	Nome do atributo que estabelece o relacionamento na entidade que está sendo descrita, ou na entidade relacionada.
Nome	Nome do relacionamento e sua descrição (o que representa no contexto do negócio).

c) Elementos de dados: Indicação dos atributos da entidade, indicando sobre características de nome, tipo, tipo de dado, descrição e se corresponde a chaves primárias ou estrangeiras.

Nome	Nome do atributo, conforme será utilizado pelos programas e linguagem SQL.
Característica	Simple, composto, identificador, único, multivalorado, derivado...
Tipo	Tipo do dado: Varchar2= Conjunto de caracteres N = Number D = Date.
Chave	Indicador de campo chave CP → chave primária CE → chave estrangeira CU → chave única
Descrição	Descrição estendida do atributo. Todos os detalhes referentes ao atributo devem ser relacionados nesta coluna. Caso possua máscara de edição, esta deve ser indicada nesta coluna, tais como: a) regras de validação b) valor padrão c) se pode ser nulo ou é requerido.

Perceba que após a descrição de todo o modelo, relacionado todas as entidades e seus respectivos atributos e relacionamentos, o dicionário de dados poderá ser utilizado para solucionar quaisquer dúvidas sobre o modelo criado.

3.2 Dicionário de dados (modelo de dados da biblioteca)

Agora iremos explicar a montagem do dicionário de dados, baseado no modelo de dados da biblioteca conforme a figura 11. Iniciaremos pela entidade Tb_Autor, seguindo a sugestão de falarmos inicialmente sobre a entidade de modo geral, sobre os relacionamentos e somente depois sobre os atributos da entidade. Procure padronizar esta ordem para facilitar o processo de busca pela informação quando necessário.





Tb_Autor	
Entidade:	
Nome da Entidade: Tb_Autor	Sigla: Aut
Descrição: Refere-se aos dados dos Autores das obras cadastradas.	

Relacionamentos:				
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado	
Tb_AutorLivro	1:M	AutLiv_CodAutor	Representa o código do autor, para indicar que o autor está associado a um livro.	
Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Aut_Codigo	Identificador	Number	CP	Refere-se ao código do autor. Não nulo.
Aut_Nome	Simples	Varchar2(60)	-	Representa o nome do autor. Não nulo.

Tb_AutorLivro	
Entidade:	
Nome da Entidade: Tb_AutorLivro	Sigla: AutLiv
Descrição: Entidade Associativa, onde são registrados e associados todos os autores de determinado livro.	

Relacionamentos:				
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado	
Tb_Autor	M:1	AutLiv_CodAutor	Representa o autor no relacionamento.	
Tb_Livro	M:1	AutLiv_CodLivro	Representa o livro no relacionamento.	
Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
AutLiv_CodAutor	Identificador	Number	CP	Representa o código do autor do livro. Não nulo.
AutLiv_CodLivro	Simples	Number	CP	Refere-se ao código do livro do autor. Não nulo.
AutLiv_Principal	Simples	Varchar2(3)	-	Indica que o autor que está sendo associado ao livro corresponde ao autor principal. Não nulo.

Tb_Categoria	
Entidade:	
Nome da Entidade: Tb_Categoria	Sigla: Cat
Descrição: Corresponde à nomenclatura de todas as áreas de interesse. Por exemplo, pode-se ter a categoria referente à informática, à física, à química, entre outros.	

Relacionamentos:			
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado



Tb_Livro	1:M	Liv_CodCategoria	Representa o código da categoria com que o livro se relaciona.	
Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Cat_Codigo	Identificador	Number	CP	Refere-se ao código da categoria. Não nulo.
Cat_Descricao	Simples	Varchar2(60)	-	Representa a descrição da categoria. Não nulo.

Tb_Livro	
Entidade:	
Nome da Entidade: Tb_Livro	Sigla: Liv
Descrição: São armazenados nesta tabela, dados referentes a cada obra disponível na biblioteca.	

Relacionamentos:			
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado
Tb_AutorLivro	1:M	AutLiv_CodAutor	Representa o código do livro no relacionamento.
Tb_Categoria	M:1	Liv_CodCategoria	Representa o código da categoria que o livro se relaciona.
Tb_Editora	M:1	Liv_CodEditora	Representa o código da editora com a qual o livro está registrado.
Tb_Reserva	1:M	Res_CodReserva	Representa o código do livro em uma reserva.
Tb_Exemplar	1:M	Exe_CodLivro	Representa o código do livro do exemplar

Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Liv_Codigo	Identificador	Number	CP	Representa o código do livro cadastrado. Campo não nulo.
Liv_Titulo	Simples	Varchar2 (60)	-	Refere-se ao título do livro. Campo não nulo.
Liv_CodEditora	Simples	Number	-	Refere-se ao código da editora na qual o livro foi editado. Campo não nulo.
Liv_CodCategoria	Simples	Number	-	Representa a categoria pela qual o livro está inserido. Campo não nulo.
Liv_Edicao	Simples	Varchar2 (20)	-	Refere-se ao número da edição do livro. Campo não nulo.
Liv_Ano	Simples	Number	-	Representa o ano de lançamento do livro. Campo não nulo.
Liv_PrazoMinimo	Simples	Number	-	Refere-se ao prazo que o livro ficará emprestado. Campo nulo. Este dado é expresso em dias.



Tb_Exemplar

Entidade:**Nome da Entidade:** Tb_Exemplar**Sigla:** Exe

Descrição: Refere-se ao exemplar do livro dentro do acervo da biblioteca. Neste caso, pode-se verificar que um livro pode conter vários exemplares. O que deve ficar bem claro é que o exemplar se faz necessário para identificar, de forma única, cada um dos exemplares de cada obra que será utilizado para realização dos empréstimos. Deve-se ter em mente que, quando é realizado um empréstimo, empresta-se um exemplar do livro para saber, exatamente, qual é o exemplar que foi emprestado para determinado leitor.

Relacionamentos:

Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado
Tb_Livro	M:1	Exe_CodLivro	Representa o livro ao qual corresponde o exemplar.
Tb_ExemplarEmprestimo	1:M	Exepre_Codexemplar	Representa o código do exemplar do livro que está sendo emprestado.

Elementos de dados:

Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Exe_Codigo	Identificador	Number	CP	Refere-se ao código do exemplar no acervo da biblioteca. Campo não nulo.
Exe_CodLivro	Simple	Number	-	Representa o código do livro registrado. Não nulo.
Exe_Status	Simple	Varchar2(10)	-	Refere-se à situação atual do exemplar. Não nulo.

Tb_Editora

Entidade:**Nome da Entidade:** Tb_Editora**Sigla:** Edi

Descrição: Representa a editora pela qual um livro foi editado.

Relacionamentos:

Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado
Tb_Bairro	M:1	Edi_CodBairro	Representa o bairro onde a editora está localizada.
Tb_Cidade	M:1	Edi_CodCidade	Representa a cidade em que a editora se localiza.
Tb_Endereco	M:1	Edi_CodEndereco	Representa a rua da editora.
Tb_Livro	1:M	Liv_CodEditora	Representa a editora que editou o livro

Elementos de dados:

Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Edi_Codigo	Identificador	Number	CP	Refere-se ao código da editora. Não nulo.
Edi_Nome	Simple	Varchar2 (60)	-	Representa o nome da editora. Não nulo.



Edi_CodEndereco	Simples	Number	-	Refere-se ao código do endereço da editora. Não nulo.
Edi_CodCidade	Simples	Number	-	Refere-se ao código da cidade da editora. Não nulo.
Edi_CodBairro	Simples	Number	-	Representa o código do bairro da editora. Não nulo.
Edi_Fone	Simples	Varchar2 (14)	-	Refere-se ao número telefônico da editora. Não nulo.
Edi_Contato	Simples	Varchar2 (14)	-	Representa uma pessoa da empresa que possa ser contatada. Nulo.

Tb_Reserva

Entidade:

Nome da Entidade: Tb_Reserva

Sigla: Res

Descrição: Refere-se à reserva de livros no acervo da biblioteca. Caso o livro não esteja disponível no momento, o livro pode ser reservado pelo leitor.

Relacionamentos:

Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado
Tb_Livro	M:1	Res_CodLivro	Representa o código do livro da reserva.
Tb_Funcionario	M:1	Res_CodFuncionario	Representa o código do funcionário responsável pela reserva.
Tb_Cliente	M:1	Res_CodCliente	Representa o código do cliente da reserva.
Tb_ExemplarEmprestimo	1:N	Exepre_CodReserva	Representa o código da reserva registrada no banco de dados.

Elementos de dados:

Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Res_Codigo	Identificador	Number	CP	Refere-se ao código da reserva. Não nulo.
Res_Data	Simples	Date	-	Representa a data que a reserva foi solicitada. Não nulo.
Res_CodLivro	Simples	Number	-	Representa o código do livro. Não nulo.
Res_CodFunc	Simples	Number	-	Refere-se ao código do funcionário que realizou a reserva. Não nulo.
Res_CodCliente	Simples	Number	-	Representa o código do cliente que realizou a reserva. Não nulo.
Res_Situacao	Simples	Varchar2 (10)	-	Refere-se ao status da reserva. Nulo.



Tb_Cargo	
Entidade:	
Nome da Entidade: Tb_Cargo	Sigla: Car
Descrição: Entidade que tem por função armazenar a descrição dos cargos dos funcionários que trabalham na biblioteca, juntamente com o salário-base equivalente.	

Relacionamentos:				
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado	
Tb_Funcionario	1:M	Fun_CodCargo	Representa o cargo do funcionário	
Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Car_Codigo	Identificador	Number	CP	Refere-se ao código do cargo. Não nulo.
Car_Descricao	Simple	Varchar2 (60)	-	Representa o nome do cargo. Não nulo.
Car_SalarioBase	simple	Number	-	Representa o salário-base, referente ao cargo. Não nulo.

Tb_Funcionario	
Entidade:	
Nome da Entidade: Tb_Funcionario	Sigla: Fun
Descrição: Representa os dados pessoais do funcionário juntamente com sua função na biblioteca.	

Relacionamentos:				
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado	
Tb_Cargo	M:1	Fun_CodCargo	Representa o cargo do funcionário.	
Tb_Bairro	M:1	Fun_CodBairro	Representa o bairro do funcionário.	
Tb_Endereco	M:1	Fun_CodEndereco	Representa o endereço do funcionário.	
Tb_Cidade	M:1	Fun_CodCidade	Representa a cidade do funcionário.	
Tb_Emprestimo	1:M	Emp_CodFuncionario	Representa o código do funcionário na tabela de empréstimos. Mantém o controle de qual funcionário efetuou o empréstimo.	
Tb_Funcionario	1:M	Res_CodFuncionario	Representa o código do funcionário responsável pela reserva.	
Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Fun_Codigo	Identificador	Number	CP	Código do funcionário. Não nulo.
Fun_Nome	Simple	Varchar2(60)	-	Nome do funcionário. Não nulo
Fun_CodCargo	Simple	Number	CE	Cargo do funcionário. Não nulo
Fun_CPF	Simple	Varchar2 (14)	-	CPF do funcionário. Não nulo.
Fun_RG	Simple	Varchar2 (20)	-	RG do funcionário. Não nulo.



Fun_CodEndereco	Simples	Number	CE	Endereço do funcionário. Não nulo.
Fun_CodBairro	Simples	Number	CE	Bairro do funcionário. Não nulo.
Fun_CodCidade	Simples	Number	CE	Cidade do funcionário. Não nulo.

Tb_Bairro

Entidade:

Nome da Entidade: Tb_Bairro

Sigla: Bai

Descrição: Representa os bairros cadastrados no banco de dados.

Relacionamentos:

Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado
Tb_Funcionario	1:M	Fun_CodBairro	Bairro onde o funcionário reside.
Tb_Cliente	1:M	Cli_CodBairro	Bairro onde o cliente reside.
Tb_Editora	1:M	Edi_CodBairro	Bairro onde localiza-se a editora.

Elementos de dados:

Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Bai_Codigo	Identificador	Number	CP	Refere-se ao código do bairro. Não nulo
Bai_Descricao	Simples	Varchar2 (60)	-	Representa o nome do bairro. Não nulo.

Tb_Cidade

Entidade:

Nome da Entidade: Tb_Cidade

Sigla: Cid

Descrição: Refere-se às cidades cadastradas no banco de dados

Relacionamentos:

Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado
Tb_Funcionario	1:M	Fun_CodCidade	Cidade onde o funcionário reside.
Tb_Cliente	1:M	Cli_CodCidade	Cidade onde o cliente reside.
Tb_Editora	1:M	Edi_CodCidade	Cidade onde localiza-se a editora.

Elementos de dados:

Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Cid_Codigo	Identificador	Number	CP	Refere-se ao código da cidade. Não nulo
Cid_Descricao	Simples	Varchar2 (60)	-	Representa o nome da cidade. Não nulo.



Tb_Endereco

Entidade:**Nome da Entidade:** Tb_Endereco**Sigla:** End**Descrição:** Representa os endereços cadastrados no banco de dados**Relacionamentos:**

Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado
Tb_Funcionario	1:M	Fun_CodEndereco	Endereço onde o funcionário reside.
Tb_Cliente	1:M	Cli_CodEndereco	Endereço onde o cliente reside.
Tb_Editora	1:M	Edi_CodEndereco	Endereço onde localiza-se a editora.

Elementos de dados:

Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
End_Codigo	Identificador	Number	CP	Refere-se ao código do endereço. Não nulo
End_Descricao	Simple	Varchar2 (60)	-	Representa o nome do endereço. Não nulo.

Tb_Cliente

Entidade:**Nome da Entidade:** Tb_Cliente**Sigla:** Cli**Descrição:** Representa os dados pessoais do cliente.**Relacionamentos:**

Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado
Tb_Reserva	1:M	Res_CodCliente	Representa o cliente que fez a reserva.
Tb_Emprestimo	M:1	Emp_CodCliente	Representa o cliente que fez o empréstimo.
Tb_Bairro	M:1	Cli_CodBairro	Bairro onde o cliente reside.
Tb_Cidade	M:1	Cli_CodCidade	Cidade onde o cliente reside.
Tb_Endereco	M:1	Cli_CodEndereco	Endereço onde o cliente reside.

Elementos de dados:

Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Cli_Codigo	Identificador	Number	CP	Refere-se ao código do cliente. Não nulo.
Cli_Nome	Simple	Varchar2(60)	-	Representa o nome do cliente. Não nulo
Cli_RG	Simple	Varchar2(20)	-	Representa o RG do cliente. Não nulo.
Cli_CPF	Simple	Varchar2(14)	-	Representa o CPF do cliente. Nulo.
Cli_CodEndereco	Simple	Number	CE	Endereço onde o cliente reside. Não nulo.
Cli_Numero	Simple	Number	-	Representa o número do endereço no qual o cliente reside. Nulo.
Cli_CodBairro	Simple	Number	CE	Bairro onde o cliente reside. Não nulo.

Cli_CodCidade	Simples	Number	CE	Cidade onde o cliente reside. Não nulo.
Cli_Fone	Simples	Varchar2(14)	-	Representa telefone para contato do cliente. Não nulo.
Cli_PrazoDevol	Simples	Number	-	Representa o prazo em dias que o cliente tem para devolver o bem emprestado.

Tb_Emprestimo

Entidade:

Nome da Entidade: Tb_Emprestimo

Sigla: Emp

Descrição: Tem por função controlar os empréstimos, armazenar qual funcionário emprestou, e qual cliente tomou emprestado e a data de realização do mesmo.

Relacionamentos:

Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado
Tb_Funcionario	M:1	Emp_CodFunc.	Mantém o controle de qual funcionário efetuou o empréstimo.
Tb_Cliente	1:M	Emp_CodCliente	Representa o código do cliente quando é realizado um empréstimo.
Tb_ExemplarEmpréstimo	1:M	Exempre_Codemp.	Refere-se ao código do empréstimo na tabela associativa. Mantém o controle de quais exemplares foram emprestados, em determinado empréstimo.

Elementos de dados:

Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Emp_Codigo	Identificador	Number	CP	Código do empréstimo. Não nulo.
Emp_Data	Simples	Date	-	Refere-se à data em que foi realizado o empréstimo. Não nulo.
Emp_CodFuncionario	Simples	Number	-	Funcionário responsável pelo empréstimo. Não nulo.
Emp_CodCliente	Simples	Number	-	Cliente que realizou o empréstimo. Não nulo.



Tb_ExemplarEmprestimo				
Entidade:				
Nome da Entidade: Tb_ExemplarEmprestimo			Sigla: Exepre	
Descrição: Tabela associativa. Identifica quais exemplares foram emprestados, controlando também os prazos e multas.				
Relacionamentos:				
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado	
Tb_Emprestimo	M:1	Exepre_Codemprestimo	Código do empréstimo do relacionamento.	
Tb_Reserva	0:1	Exepre_CodReserva	Representa o código reserva que motivou um empréstimo.	
Tb_Exemplar	M:1	Exepre_CodExemplar	Representa o código do exemplar registrado no banco de dados.	
Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Exepre_CodExemplar	Identificador	Number	CP	Refere-se ao código do exemplar. Não nulo.
Exepre_CodEmprestimo	Identificador	Number	CP	Refere-se ao código do empréstimo. Não nulo.
Exepre_DtPrevDev	Simple	Date	-	Refere-se à data prevista para a devolução do exemplar. Não nulo.
Exepre_DtDevMul	Simple	Date	-	Refere-se à data de devolução em que foi cobrada multa. Nulo.
Exepre_DtPgMul	Simple	Date	-	Refere-se à data em que foi paga a multa. Nulo.
Exepre_CodReserva	Simple	Number	CE	Refere-se ao código da reserva. Não nulo.
Exepre_ValorMulta	Simple	Number	-	Refere-se ao valor cobrado em caso de atraso da devolução. Não nulo.

Perceba que acaba por ser cansativo o trabalho de descrição de todas as regras do modelo de dados para deixá-lo o mais explicativo possível. É importante lembrar que apesar de cansativo, o dicionário de dados é fundamental para consultas de projetistas e desenvolvedores, fazendo com que seja facilitado o trabalho destes.

Resumo

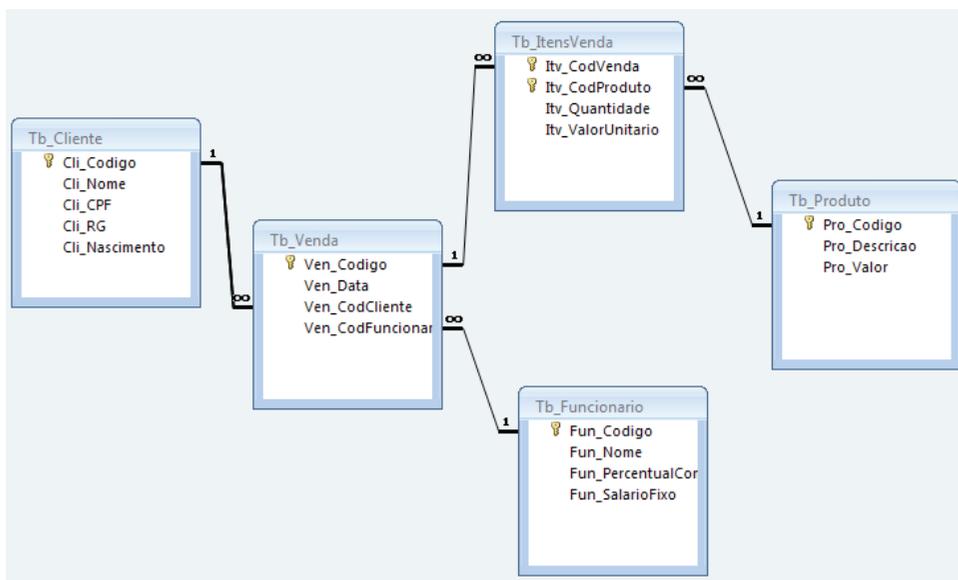
Observamos nesta aula que o dicionário de dados corresponde a uma descrição detalhada da estrutura que o banco de dados possui. Baseado em um modelo de dados de uma Biblioteca, demonstramos como um dicionário de dados é montado, entendendo que ele apresenta formas de caracterizar cada um dos principais elementos do Modelo Entidade Relacionamento



(MER), relativas as entidades, relacionamentos e atributos.

Atividade de Aprendizagem

1. Elaborar o Dicionário de dados para o Modelo Entidade Relacionamento abaixo:



Ao final da construção do dicionário de dados você deve estar refletindo no quanto é trabalhoso o papel do profissional de Banco de Dados. Apesar do trabalho, verifique que este processo é realizado somente uma vez para cada modelo de dados criado, e que este poderá ser utilizado por muitos anos, compensando assim esta etapa. Pense em que passados dez anos, ao ler o dicionário de dados criado, você ficará a par de tudo o que foi pensado na época para o desenvolvimento de tal modelo, e poderá sugerir alterações ou melhoras de forma confiável e com o mínimo de risco. Na próxima aula nosso tema será Normalização de dados. Prossiga com atenção.



Aula 4. Normalização de dados

Objetivos:

- aprender a construir o Banco de Dados em sua estrutura; e
- produzir um projeto de Banco de Dados utilizando as técnicas de normalização.

Caro (a) estudante,

Nesta aula, aprenderemos a montar o Modelo Entidade Relacionamento (MER), considerando outro ponto de vista. Anteriormente, era apresentado o contexto em que tinha-se a ideia de modelar o banco e eram definidas as entidades, relacionamentos e atributos de forma bastante subjetiva. A normalização de dados trata da construção do modelo de dados, mas considera fatos reais, como documentos fiscais, formulários, planilhas, fichas, entre outras coisas. A ideia básica é que a partir destes documentos, aplicando regras previstas no processo de normalização, seja gerado o Modelo de Dados.

Para atingir os objetivos, podemos dizer que um banco de dados está relacionado a alguns conceitos fundamentais:

- Banco de Dados = Coleção de Arquivos;
- Arquivos = Coleção de Registros;
- Registro = Coleção de Campos (Tuplas);
- Campo = Coleção de Caracteres;
- Caractere = Alfa-numéricos ou símbolos.



De acordo com Machado (1995), existem nove técnicas que podem ser utilizadas no desenvolvimento de um banco de dados nas fases iniciais, as quais podemos citar:

1. Contextualização;
2. Objetivação;
3. Intitulação;
4. Especificação de Requisitos;
5. Normalização;
6. Modelagem;
7. Trigramação;
8. Dicionarização;
9. Auditoria das fases de Análise e de Projeto Lógico do Sistema de Banco de Dados.

Para este curso, estaremos focados no que diz respeito à Normalização como ferramenta para o desenvolvimento de um banco de dados.

Os conceitos apresentados neste material, em seu todo ou em partes, foram retirados do livro “Projeto de Banco de Dados: uma visão prática”, de Machado, F. N. R, e Abreu, M. P., Editora Érica, 1995, Capítulo 12, Normalização.

4.1 Formas normais

A Teoria da Normalização é expressa tradicionalmente através de um conjunto de Formas Normais, que otimizam a estrutura e o conteúdo das relações entre as entidades.

O conceito de normalização foi introduzido por E. F. Codd, em 1970, utilizando a primeira forma normal – 1FN. Esta técnica, pode-se dizer, é um processo matemático formal que fundamenta-se na teoria dos conjuntos.





Nesse processo podem ocorrer as seguintes Anomalias de Atualização:

- Inclusão - ao se incluir um novo cliente, por exemplo, ele tem que estar relacionado com uma venda.
- Alteração - caso o fabricante de um produto altere o preço de determinada classe de produto, será preciso percorrer toda a Entidade para se realizarem múltiplas alterações.
- Exclusão - Ao se excluir um cliente, por exemplo, os dados referentes às suas compras poderão ser perdidos.

Através do processo de normalização podemos substituir, de forma gradual, um conjunto de Entidades e Relacionamentos para um modelo mais “adequado”, em relação às Anomalias de Atualização (Inclusão, Alteração e Exclusão), as quais podem causar certos problemas:

- Grupos repetitivos (atributos multivalorados) de dados;
- Dependências parciais em relação a uma chave concatenada;
- Redundâncias de dados desnecessários;
- Perdas acidentais de informação;
- Dificuldade na representação de fatos da realidade observada;
- Dependências transitivas entre atributos.

Estes problemas podem ser minimizados através da aplicação da normalização, tornando o modelo de dados bastante estável e sujeito a poucas manutenções.

4.2 Dependência funcional

Para descrever as formas normais, faz-se necessária a introdução deste conceito importante na aplicação da normalização, no qual a maior parte da teoria de normalização foi baseada.

Dada uma entidade qualquer, dizemos que o atributo ou conjunto de atributos



tos de A é dependente funcionalmente de outro atributo de B; neste caso, há dependência funcional.

Em outras palavras, podemos dizer que, em uma entidade TB_ITENSNOTA, o atributo ITN_CODPRODUTO depende funcionalmente do atributo PRO_CODIGO, que encontra-se na entidade TB_PRODUTO.

A análise das relações existentes entre os atributos de uma entidade deve ser feita a partir de um conhecimento prévio da sistemática aplicada nas relações entre as entidades, para que esta possa ser modelada, como no exemplo anterior.

Para iniciar a normalização, faz-se necessária a descrição de todos os atributos do documento que se deseja normalizar (visando a criação do modelo de dados), para que posteriormente, possam ser aplicadas técnicas (formas normais) que irão dividir os atributos em entidades, as quais darão suporte à construção do modelo de dados.

Para facilitar o entendimento, este material irá apresentar a normalização com exemplos práticos.

4.3 Normalização de dados (nota fiscal)

Supondo a necessidade de uma empresa em modelar um banco de dados para emissão de uma nota fiscal simples, como apresentado no quadro abaixo. Verifique que há dados que são específicos da empresa, representados aqui como "Dados da Empresa" e que não há necessidade de manter-se no banco de dados, pois são impressos diretamente na nota fiscal encaminhada pela gráfica. O que nos interessa aqui é justamente os dados que necessitam ser preenchidos, como Data, Número da Nota Fiscal, Cliente, RG, CPF, Cidade, UF, Endereço, Número e Bairro, além dos dados dos produtos vendidos, inerentes ao Código do produto, Descrição, Unidade de Medida, Quantidade, Valor Unitário e Valor Total.



Dados da Empresa		Data: 01/01/0101		NF Nº 123	
CLIENTE: João da Silva			RG: 0000000000-00		
CPF:000000000-00		CIDADE: Medianeira		UF: PR	
ENDEREÇO: R. Bahia		Nº 1234	BAIRRO: Centro		
Código	Descrição	Und. de Medida	Quantidade	Vlr Unitário	Vlr Total
1	Arroz	kg	2	1,99	3,98
2	Sabonete	Unidade	1	0,79	0,79
3	Leite	Litro	3	2,12	6,36
				Valor TOTAL : 11,13	

Inicialmente, realiza-se a definição de todos os atributos que o documento possui, registrando que eles são tidos como atributos da entidade principal (documento), conforme segue.

Neste caso, a entidade principal foi definida como Nota e a ela foram relacionados todos os atributos oriundos a emissão de uma nota fiscal. Foi definida ainda a chave primária, única para cada nota fiscal a ser emitida.

Nota (número da nota, data, cliente, RG, CPF, cidade, UF (Unidade da Federação), endereço, nº, bairro, código_produto_1, código_produto_2, código_produto_3, ..., descrição_produto_1, descrição_produto_2, descrição_produto_3, ..., unidade_produto_1, unidade_produto_2, unidade_produto_3, quantidade_produto_1, quantidade_produto_2, quantidade_produto_3, ..., valor_unitário_produto_1, valor_unitário_produto_2, valor_unitário_produto_3, ..., valor_total_produto_1, valor_total_produto_2, valor_total_produto_3, ..., valor_total_da_nota);

- Os campos sublinhados indicam as chaves primárias.

4.3.1 Aplicação da 1ª Forma Normal – 1FN

Existem situações em que nos deparamos com algumas informações que se repetem (atributo multivalorado) dentro de uma única linha, ligada a uma chave primária.

A 1FN diz que cada ocorrência da chave primária deve corresponder a somente uma informação de cada atributo, ou seja, a entidade não deve conter atributos repetidos ou multivalorados, ou, ainda, os atributos não-chave deverão ser atômicos (únicos).





Ao observarmos que certo grupo de atributos não-chave não são atômicos, ao longo do processo de entrada de dados, verificamos que este atributo deverá ser decomposto em uma nova entidade.

Nas novas entidades criadas, a chave primária será a concatenação da chave primária da entidade original, o que mantém, desta forma, o relacionamento entre elas, como também o conjunto de atributos que se repetem.

Por exemplo:

A 1FN diz que todos os atributos devem conter um valor atômico (único), ou seja, os dados inclusos dentro do campo não devem se repetir.

Ex.: o campo cliente vai ser preenchido quantas vezes na nota acima? Apenas uma vez, certo? Então corresponde a um valor atômico. Já os campos que se referem aos itens da nota serão acrescidos uma ou muitas vezes. Desta forma, tem-se a necessidade de separação da entidade Nota, criando-se a nova entidade chamada Itens_Nota, ficando:

Nota (número da nota, data, cliente, RG, CPF, endereço, nº, bairro, cidade, UF, valor_total)

Os campos que se repetem são destinados a uma nova tabela, no nosso exemplo chamado de itens da nota. A chave principal também é inclusa na nova tabela para manter o relacionamento entre elas.

Itens_Nota (número da nota, código, descrição, unidade_de_medida, valor_unitário, quantidade, valor_total)

- Note: as tabelas que saem na 1FN são tabelas associativas.

4.3.2 Aplicação da 2ª Forma Normal – 2FN

A 2FN diz que todos os atributos que não forem chave têm que ser dependentes diretos da chave, ou seja, os atributos que estão contidos na nota têm que depender diretamente da chave **Número da nota**.

Inicialmente, observa-se a entidade que possui chave primária concatenada. Para aquelas que satisfazem esta condição, analisar se existe algum atributo ou conjunto de atributos com dependência direta à chave primária.





Ex.: o atributo cliente é dependente direto da nota, pois é o cliente desta nota. Já atributo RG, CPF são dependentes do cliente, não tendo nenhuma relação direta com a nota, ou seja, independentemente da nota, o CPF do cliente será o mesmo se soubermos o nome do cliente. Sabe-se também que o CPF do cliente nunca muda. Desta forma, tem-se:

Nota (número da nota, data, código_cliente, endereço, nº, bairro, código_cidade, valor_total);

- Os atributos adicionados acima “código_cliente e código_cidade” são para manter o relacionamento entre as entidades.
- Verifique que o campo endereço deve permanecer na entidade Nota, tendo em vista que, se, por acaso, pretender-se buscar uma nota emitida há cinco anos cujo cliente alterou seu endereço para outra rua, o endereço que deverá aparecer na nota emitida há cinco anos deve ser o correspondente ao endereço antigo do cliente, e não o novo endereço.

Cliente (código, nome, RG, CPF);

- Os campos dependentes do cliente são transferidos para a tabela cliente.

Cidade (código, descrição, UF);

- O atributo UF que depende apenas da cidade é atribuído para a tabela cidade.

Itens nota (número da nota, código produto, quantidade, valor_unitário, valor_total);

- O atributo “código_produto” é mantido na tabela associativa para manter o relacionamento entre Nota, Itens da Nota e Produtos.

Produto (código, descrição, unidade_de_medida, valor_unitário);

- Os atributos dependentes de produto são relacionados tabela produto. Repare que o atributo valor unitário foi duplicado, isso pela necessidade do valor unitário de um produto sofrer mudanças, em notas diferentes o valor unitário pode estar diferente também.



4.3.3 Aplicação da 3ª Forma Normal – 3FN

A 3FN diz que o atributo precisa estar na segunda forma normal, e todos os atributos que não são chave, não podem depender de outros atributos que também não são chave. Deve-se verificar ainda se o valor de um atributo pode se repetir em muitas tuplas, sendo que neste caso, é necessário inclusão de nova entidade.

Uma Entidade está na 3FN se nenhum de seus atributos possui dependência em relação a outros atributos da Entidade. Ao verificar a existência de dependência entre as Entidades e seus atributos, devemos criar outra entidade que contenha os atributos dependentes.

Nota (número da nota, data, código_cliente, código_endereço, código_bairro, nº, código_cidade);

- Note que o atributo valor total da nota foi retirado na 3FN. Todos os atributos que são gerados a partir do resultado de dois ou mais atributos são retirados, visando manter a consistência dos dados.

Endereço (código, descrição);

- Aplicando a 3FN, o endereço é retirado da tabela nota, pois o endereço não é um atributo chave e refere-se ao endereço do cliente nesta nota. Então, o endereço é incluso na tabela de clientes e relaciona-se tanto com a entidade Cliente como a entidade Nota.

Bairro (código, descrição);

- Aplicando a 3FN, o bairro é retirado da tabela nota, pois o bairro não é um atributo chave e refere-se ao bairro do cliente nesta nota. Então, o bairro é incluso também na tabela de clientes e relaciona-se tanto com a entidade Cliente como a entidade Nota.

Cliente (código, nome, RG, CPF, nº, código_endereço, código_cidade);

- Os campos chave código_endereço e código_cidade vêm para a tabela de clientes com o fim de fazer o relacionamento entre as tabelas.

Cidade (código, descrição, UF);

Itens nota (número da nota, código produto, quantidade, valor_unitário);

- O atributo valor total sai, pois ele é resultado de outros dois atributos: a quantidade vezes o valor unitário.

Produto (código, descrição, unidade_de_medida, valor_unitário);

A figura 18 representa o MER do exemplo normalizado anteriormente.

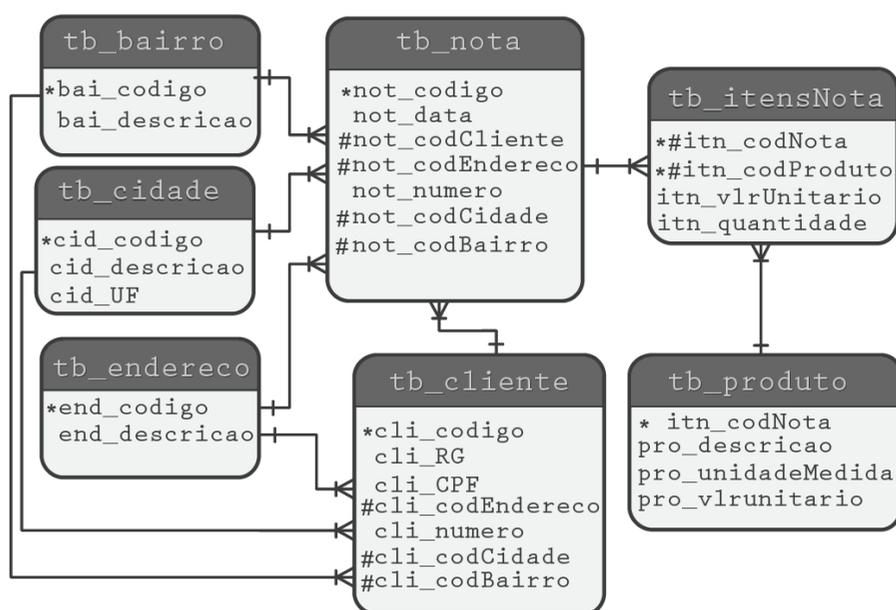


Figura 18 - Modelo gerado pela Normalização da Nota Fiscal

Fonte: autor (adaptado pelo ilustrador)



4.4 Normalização de dados (Formulário Remanejamento)

Dado o formulário de solicitação de remanejamento, será aplicada a normalização de dados visando obter um modelo de dados conciso:



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS MEDIANEIRA

REQUERIMENTO DE REMANEJAMENTO

ESPAÇO A SER PREENCHIDO PELO REQUISITANTE

Nome:.....

Sexo: M F Data de Nascimento:...../...../.....

Categoria Funcional:
.....

Lotação atual:
Data de admissão:...../...../.....
Cidade.....

Escolaridade
 1º Grau
 2º Grau - Curso
.....

3º Grau - Curso:
.....

Outros:
.....

Principais cursos realizados:

1.....
2.....
3.....
4.....
5.....

Experiências profissionais relevantes:
.....
.....
.....

Principais atribuições desenvolvidas ao setor:
.....
.....
.....

Figura 19

Fonte: Machado (1995)





Setores em que já trabalhou:

1Período:a

Motivo da saída:.....

2Período:a

Motivo da saída:.....

3Período:a

Motivo da saída:.....

Obs: se necessário, utilizar folha avulsa e anexar

Atribuições que gostaria de trabalhar, ordem de preferência:

1.....

2.....

3.....

4.....

5.....

Horário de preferência:

Setores/Divisões/ Diretorias em que gostaria de trabalhar, em ordem de preferência:

1.....

2.....

3.....

Qual(is) o(s) motivo(s) que o torna descontente com suas atividades?

- Falta de correlação com a formação acadêmica
- Atividade com complexidade mínima
- A falta de formação em relação à atividade desenvolvida prejudica sua execução
- Horário de trabalho
- O relacionamento com a chefia não é satisfatório
- O relacionamento com os colegas de trabalho não é bom
- Faz a mesma coisa há muitos anos
- Falta de expectativa de crescimento profissional
- A atividade desenvolvida está criando problemas de saúde
- Excesso de trabalho
- Poucos trabalho
- Ambiente excessivamente autocrático

Outros.....

Observações:

.....

.....

.....

Data de preenchimento:...../...../.....

Medianeira, de de.....

.....

Assinatura do servidor

Figura 20
 Fonte: Machado (1995)





ESPAÇO A SER PREENCHIDO PELA CHEFIA IMEDIATA

O(a) servidor(a) tem demonstrado interesse pela atividade atualmente desenvolvida?

.....

As tarefas que lhe são confiadas são cumpridas a contento?

.....

Os horários cumpridos estão dentro do estabelecido pela chefia?

.....

O(a) servidor(a) respeita os regulamentos e age em conformidade com as normas internas?

.....

O(a) servidor(a) desempenha suas funções com seriedade e precisão?

.....

Qual a posição da chefia em relação à permanência do servidor no atual setor?

.....

Qual o relacionamento do(a) servidor(a) com o grupo de trabalho?

.....

Mediante as informações acima prestadas:

() Libero o(a) funcionário(a) sem reposição
 () Libero o(a) funcionário(a) mediante reposição
 () Não libero o(a) funcionário(a)

Justificativa:

.....

.....

Data:...../...../..... Assinatura e carimbo:

Espaço a ser preenchido pela Diretoria em que o servidor está lotado

Data:/...../..... Assinatura e carimbo:.....

Espaço a ser preenchido pela Divisão de Recrutamento, Seleção e Desenvolvimento de Pessoal

.....

Figura 21

Fonte: Machado (1995)





4.4.1 Definição dos atributos contidos no formulário

Requerimento (código, nome, sexo, data_nascimento, categoria_funcional, lotação_atual, data_admissão, cidade, escolaridade, grau, cursos_realizados, experiências_profissionais, atribuições_desenvolvidas, setor_que_trabalhou, data_entrada, data_saída, motivo_saída, atribuições_que_gostaria, horário, setores/divisões/diretorias_gostaria, motivos_descontente, observações, data_preenchimento, questão1, questão2, questão3, questão4, questão5, questão6, questão7, resultado, justificativa, data_resultado)

- Os campos sublinhados indicam as chaves primárias.

4.4.2 Aplicação da 1ª Forma Normal - 1FN

A 1FN diz que todos os atributos devem conter um valor atômico (único), ou seja, os dados inclusos dentro do campo não devem se repetir.

Ex.: o campo que representa os cursos feitos pelo requerente será preenchido quantas vezes no formulário? Depende de quantos cursos ele fez: na possibilidade de ser preenchido duas vezes, o campo é retirado para uma tabela associativa.

Remanejamento (código, nome, sexo, data_nascimento, categoria_funcional, lotação_atual, data_admissão, cidade, data_preenchimento, questão1, questão2, questão3, questão4, questão5, questão6, questão7, resultado, justificativa, data_resultado);

- Permaneceram na tabela remanejamento apenas os campos com valores únicos, os campos que podem ser preenchidos uma ou mais vezes são retirados para as tabelas associativas.

Remanejamento/escolaridade (código_remanejamento, escolaridade, grau);

Remanejamento/Cursos (código_remanejamento, cursos);

Remanejamento/experiências profissionais (código_remanejamento, experiências_profissionais);

Remanejamento/atribuições desenvolvidas (código_remanejamento, atribuições);





Remanejamento/setores que trabalhou (código_reanejamento, setor, data_entrada, data_saída, motivo_saída);

Remanejamento/atribuições que gostaria (código_reanejamento, atribuições, horário);

Remanejamento/setores-divisões-diretorias que gostaria (código_reanejamento, setor);

Remanejamento/motivos descontentamento (código_reanejamento, motivos, observações).

4.4.3 Aplicação da 2ª Forma Normal - 2FN

A 2FN diz que todos os atributos que não forem chave têm que ser dependentes diretos da chave, ou seja, os atributos que estão contidos no requerimento têm que depender diretamente da chave código requerimento.

Inicialmente, observa-se a entidade que possui chave primária concatenada. Para aquelas que satisfazem esta condição, analisar se existe algum atributo ou conjunto de atributos com dependência direta à chave primária.

Remanejamento (código, código_funcionário, categoria_funcional, lotação_atual, data_admissão, cidade, data_preenchimento, questão1, questão2, questão3, questão4, questão5, questão6, questão7, resultado, justificativa, data_resultado);

- Verifique que o campo categoria funcional, lotação atual e cidade devem permanecer na entidade Remanejamento, tendo em vista que, se, por acaso, se desejar buscar um formulário emitido há cinco anos, cujo funcionário alterou sua categoria funcional, lotação atual ou cidade, os dados que deverão aparecer no formulário preenchido há cinco anos devem ser o correspondente à categoria funcional, lotação atual ou cidade, preenchido antigamente, e não os dados novos.

Funcionário (código, nome, sexo, data_nascimento);

Remanejamento/escolaridade (código_reanejamento, escolaridade, grau);

Remanejamento/Cursos (código_reanejamento, cursos);





Remanejamento/experiências profissionais (código_reanejamento, experiências);

Remanejamento/atribuições desenvolvidas (código_reanejamento, atribuições);

Remanejamento/setores que trabalhou (código_reanejamento, setor, data_entrada, data_saida, motivo_saida);

Remanejamento/atribuições que gostaria (código_reanejamento, atribuições, horário);

Remanejamento/setores-divisões-diretorias que gostaria (código_reanejamento, setor);

Remanejamento/motivos descontentamento (código_reanejamento, motivos, observações);

4.4.4 Aplicação da 3ª Forma Normal – 3FN

A 3FN diz que o atributo precisa estar na segunda forma normal, e todos os atributos que não são chave, não podem depender de outros atributos que também não são chave.

Uma Entidade está na 3FN se nenhum de seus atributos possui dependência em relação a outros atributos da entidade. Ao verificar a existência de dependência entre as Entidades e seus atributos, devemos criar outra entidade que contenha os atributos dependentes.

Remanejamento (código, código_funcionário, código_categoria_funcional, código_lotação, data_admissão, código_cidade, data_preenchimento, questão1, questão2, questão3, questão4, questão5, questão6, questão7, resultado, justificativa, data_resultado);

Categoria Funcional (código, descrição);

Lotação (código, descrição);

Cidade (código, descrição);

Funcionário (código, nome, sexo, data_nascimento);





Remanejamento/escolaridade (código_remanejamento, código_escolaridade, grau);

Escolaridade (código, descrição);

Remanejamento/Cursos (código_remanejamento, código_cursos);

Cursos (código, descrição);

Remanejamento/experiências profissionais (código_remanejamento, código_experiências);

Experiências Profissionais (código, descrição);

Remanejamento/atribuições desenvolvidas (código_remanejamento, código_atribuições);

Atribuições (código, descrição);

Remanejamento/setores que trabalhou (código_remanejamento, código_setor, data_entrada, data_saída, motivo_saída);

Setores (código, descrição);

Remanejamento/atribuições que gostaria (código_remanejamento, código_atribuições, horário);

Remanejamento/setores-divisões-diretorias que gostaria (código_remanejamento, código_setor);

Remanejamento/motivos descontentamento (código_remanejamento, código_motivos, observações);

Motivo descontentamento (código, descrição).

A figura 22 demonstra o MER do exemplo anterior.

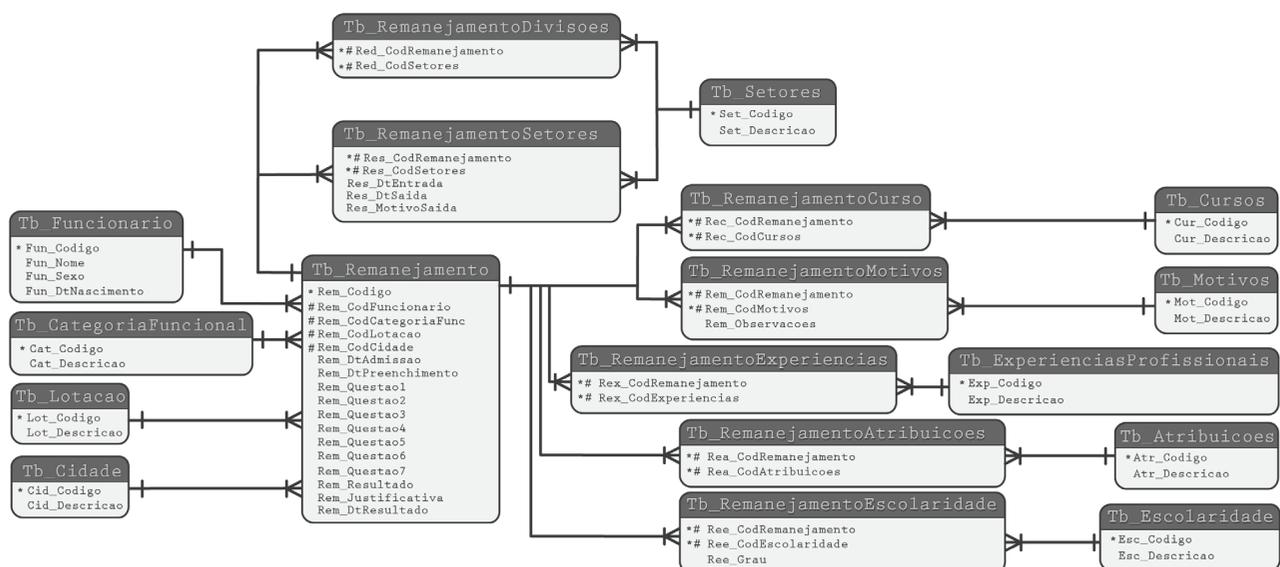


Figura 22 - Modelo de Dados aplicado ao Requerimento de Remanejamento

Fonte: autor (adaptado pelo ilustrador)

Resumo

Chegando ao fim deste capítulo, você deve ter percebido que a normalização de dados é uma ferramenta a mais, que propicia a criação de um Modelo de Dados conciso de forma facilitada. Lembre-se de que sempre há necessidade de seguir as regras "ao pé da letra", considerando que caso não faça isso, seu modelo gerado poderá apresentar problemas, como duplicidade de dados, inconsistência, entre outros. Fixe a ideia de que a primeira forma normal busca determinar todas as tabelas associativas que você terá. A segunda forma normal busca manter a integridade do banco, mantendo inclusive o aspecto histórico necessário e sua dependência com as entidades. A terceira forma visa criar entidades que normalmente não dependem de ninguém, mas que se tornam dependentes de outras entidades.



Atividade de Aprendizagem

1. Aplicar as 3 formas normais e criar o Modelo Entidade Relacionamento para o formulário de cadastro de pessoas em um determinado município:

CADASTRO INDIVIDUAL DE MUNÍCIPE

NOME:

CPF: RG: EMISSOR: UF:

DATA: TÍTULO: SEÇÃO: ZONA:

NASCIMENTO: NATURALIDADE:

NACIONALIDADE: SEXO: MASCULINO FEMININO

ESTADO CIVIL: SOLTEIRO CASADO SEPARADO CONCUBINATO VIÚVO DIVORCIADO

DEPENDENTES: ENDEREÇO: NÚMERO:

BAIRRO:

FONE: CELULAR: DATA CHEGADA:

COR: BRANCA NEGRA AMARELO PARDO

NOME DO PAI:

NOME DA MÃE:

OCUPAÇÃO:

ESCOLARIDADE: FORMAÇÃO:

ONDE ESTUDA? PRETENDE VOLTAR ESTUDAR? SIM NÃO

PORTADOR DE NECESSIDADES: FÍSICA MENTAL AUDITIVA VISUAL

ESTUDO PRETENDIDO: SOLICITA CUPOM FISCAL: SIM NÃO

ASSOCIAÇÃO: SINDICATO:

CLUBE: RELIGIÃO:

Prezado(a) estudante,

Esperamos que você tenha percebido que a normalização de dados corresponde a mais uma ferramenta para modelagem de bancos de dados. Neste momento do curso você já deve estar apto a trabalhar modelando bancos de dados para situações reais, adequando os modelos conforme necessidades específicas. A próxima aula trata de como os modelos criados são implementados no banco de dados e como é realizada a manipulação dos dados no banco. Para isso, normalmente utiliza-se a linguagem SQL (**Structured Query Language**) que está presente em praticamente todos os Sistemas Gerenciadores de Bancos de Dados e é tida como linguagem universal para bancos de dados.

Aula 5. Linguagem SQL

Objetivos:

- implementar modelos de dados utilizando a linguagem SQL (**Structured Query Language**); e
- operar a manipulação de informações em bancos de dados.

Após a modelagem do banco de dados, no qual há necessidade de definição de quais entidades, atributos, relacionamentos, tipos de dados e restrições que o mesmo deve conter, fica relativamente fácil implementá-lo no Sistema Gerenciador de Banco de Dados, o qual irá gerenciar os dados a serem armazenados. Para realizar a implementação do modelo de banco de dados, assim como manipular os dados (inserir, alterar, deletar) utiliza-se uma determinada linguagem, sendo que a mais conhecida e padronizada para praticamente todos os bancos de dados, corresponde à linguagem SQL (Structured Query Language).

A linguagem SQL foi desenvolvida para trabalhar com a manipulação de dados em bancos de dados e tornou-se um padrão internacional de acesso a bancos de dados.

Seus comandos são divididos em categorias, cada qual com funcionalidades específicas. Dependendo da literatura utilizada, possuímos quatro ou cinco categorias. Considerando o material fornecido pela Oracle (empresa que é líder do mercado no segmento de banco de dados), possuímos quatro classes distintas de comandos:

5.1 Comandos DDL (Linguagem de Definição de Dados)

São comandos utilizados para construção dos modelos de dados. Por exemplo, se se deseja criar uma tabela (entidade), deve-se utilizar o comando CREATE TABLE para construí-la. Deve ficar claro que os comandos DDL são



utilizados somente para criação de estruturas e não inserção de registros de dados.

Entre os comandos DDL temos:

5.1.1 Comando CREATE

Utilizado para criação de tabelas, índices, relacionamentos, entre outros. Como exemplo tem-se:

```
CREATE TABLE tb_cliente
(
  cli_codigo INTEGER NOT NULL,
  cli_nome VARCHAR2(60) NOT NULL,
  cli_rg VARCHAR2(15) NOT NULL,
  cli_cpf VARCHAR2(15) NOT NULL,
  cli_sexo CHAR(1) NOT NULL,
  PRIMARY KEY (cli_codigo)
);
```

Este exemplo apresenta a criação da entidade tb_cliente, com cinco atributos, em que são definidos os tipos de dados e a chave primária (cli_codigo) no final da instrução. O que indica o final da instrução corresponde ao (;) ponto e vírgula.

Para criação do índice único (não permite cadastrar dois clientes com o mesmo nome e com o mesmo CPF), um comando DDL deve ser utilizado, conforme o exemplo:

```
CREATE UNIQUE INDEX uk_cliente ON tb_cliente (cli_nome, cli_cpf);
```

5.1.2 Comando ALTER

É utilizado para alterar a estrutura de uma tabela, por exemplo. Através dele pode-se:

Adicionar colunas:

```
ALTER TABLE TB_CLIENTE ADD CLI_FONE VARCHAR2(14) NOT NULL;
```

Modificar colunas:

```
ALTER TABLE TB_CLIENTE MODIFY CLI_FONE VARCHAR2(18) NOT NULL;
```

Dropar colunas:

```
ALTER TABLE TB_CLIENTE DROP COLUMN CLI_FONE;
```

Marcar colunas como não utilizadas:

```
ALTER TABLE TB_CLIENTE SET UNUSED COLUMN CLI_FONE;
```

Renomear colunas:

```
ALTER TABLE TB_CLIENTE RENAME COLUMN CLI_FONE TO CLI_TELEFONE;
```

Definir tabelas como somente leitura:

```
ALTER TABLE TB_CLIENTE READ ONLY;
```

5.1.3 Comando DROP

Comando utilizado para apagar tabelas, colunas, índices, entre outros:

```
DROP TABLE tb_cliente;
```

5.2 Comandos DML (Linguagem de Manipulação de Dados)

Os comandos de manipulação de dados servem para incluir, alterar, deletar ou selecionar registro das estruturas de dados criadas, utilizando os comandos DDL.



5.2.1 Comando INSERT

Utilizado para inserir registros em uma entidade:

```
INSERT INTO tb_bairro (bai_codigo, bai_descricao) VALUES (1, 'CENTRO');
```

5.2.2 Comando UPDATE:

Utilizado para atualizar registros em uma entidade:

```
UPDATE tb_funcionario  
SET fun_salario = 2000  
WHERE fun_codigo = 10;
```

Perceba que este comando irá atualizar a tabela de funcionário, alterando o salário do funcionário de código = 10 para R\$ 2000,00.

5.2.3 Comando DELETE

Utilizado para apagar registro de uma entidade:

```
DELETE FROM tb_cliente WHERE cli_codigo = 5;
```

Neste caso, o comando apagará da tabela tb_cliente cujo código for 5;

5.2.4 Comando SELECT

Utilizado para selecionar registros em uma ou mais tabelas:

```
SELECT fun_codigo, fun_nome, fun_salario FROM tb_funcionario;
```

Este comando selecionará, na tabela tb_funcionario, todos os funcionários e seus respectivos códigos, nomes e salários.

5.3 Comandos DCL (Linguagem de Controle de Dados)

Permitem controlar quais usuários tem permissão para quais atribuições no banco de dados. O Comando GRANT dá permissão, o comando REVOKE exclui as permissões concedidas.

```
GRANT CREATE_TABLE ON USER_JOSE;  
REVOKE CREATE_TABEL ON USER_JOSE;
```

Respectivamente, dá-se permissão de criação de tabelas ao usuário USER_JOSE e retira-se a permissão concedida.

5.4 Comandos DTL (Linguagem de Transação de Dados)

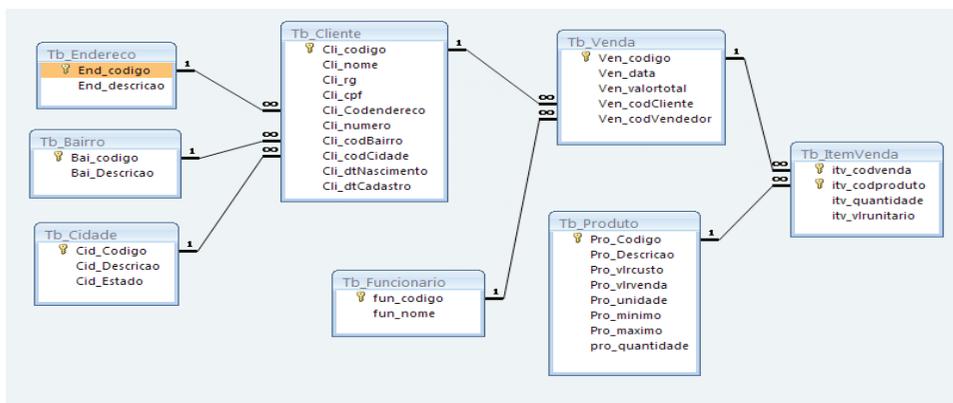
Estes comandos são necessários para que os dados sejam gravados fisicamente no disco, quando se confirma uma transação como concluída. Desta forma, o comando COMMIT, quando executado, grava toda a transação no disco, sem a possibilidade de retorno em uma situação anterior, a não ser através de mecanismos de backup. Quando se deseja desfazer uma transação, executa-se o comando ROLLBACK, e o sistema desfaz automaticamente tudo o que não foi “comitado” até o momento.

Resumo

Neste capítulo você pôde avaliar como é a estrutura da linguagem SQL, utilizada para construção e manipulação de bancos de dados. É importante que você tenha em mente a separação entre os comandos, que possuem particularidades. Lembre-se que comandos DDL são para serem utilizados para construção dos modelos de dados. Comandos DML são para manipulação de dados nos bancos de dados já criados. Comando DTL auxiliam na manipulação dos dados, indicando início e fim de transação e permitem que alterações sejam desfeitas. Por fim, comandos DCL são utilizados para controle de permissões de acesso e manipulação de dados e do banco, dadas ou retiradas dos usuários.

Atividade de Aprendizagem

1. Construir o modelo de dados utilizando comandos SQL, e inserir em cada entidade pelo menos um registro:





Palavras finais

Ao final de nosso curso, espero que você tenha tirado o máximo de proveito do conteúdo repassado e que este ensinamento seja o início de um futuro brilhante. Acredito que esta disciplina poderá colaborar para sua formação profissional e acadêmica e que será apenas o ponto de partida para que se torne um(a) profissional respeitado(a) e com orgulho do que faz, atuando sempre com simplicidade e dignidade.

Como professor, tenho consciência de que serei apenas mais um colaborador para que alcance seus objetivos, mas de qualquer forma, gostaria de dizer que não só como professor, mas como amigo, torço para que alcance seus objetivos e continue sempre na luta por aprender mais e buscar novos caminhos.





Guia de Soluções

Aula 1. Modelagem de Dados

1. Defina entidade e apresente dois exemplos práticos.

Resposta: Corresponde a tudo aquilo que se deseja guardar dados, podendo ser concreto ou abstrato. Ex. Cliente, com os atributos Código, nome, CPF, RG, data de nascimento. Neste caso teríamos um atributo concreto. Como abstrato, podemos ter a entidade Venda, que corresponde a um procedimento interno da empresa. Neste caso, teríamos que armazenar dados de nº da venda, data, cliente, funcionário, produtos, quantidades e valores unitários.

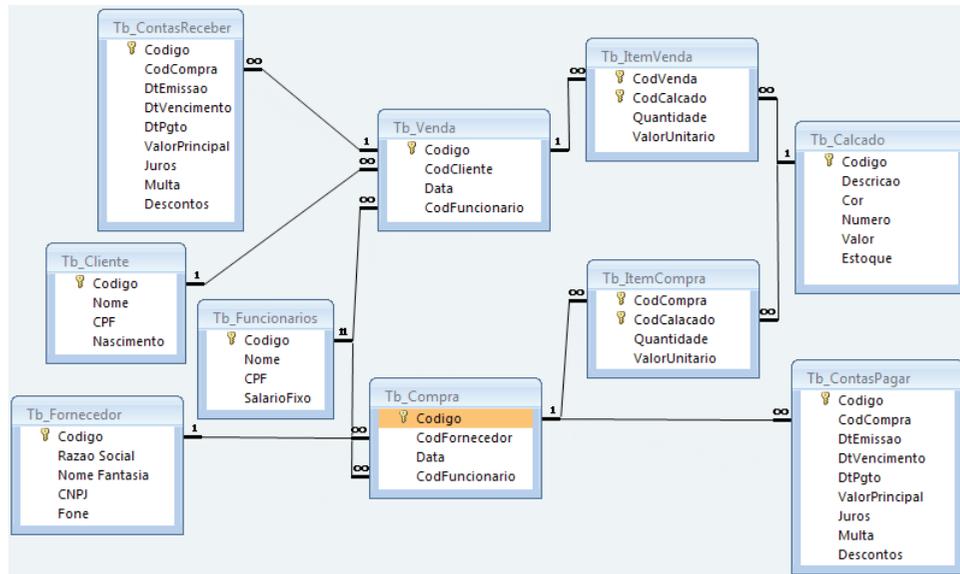
2. Defina relacionamento e apresente dois exemplos práticos.

Resposta: corresponde à ligação lógica que há entre duas entidades. A entidade Venda, por exemplo, tem ligação com o cliente para quem foi vendido algum item. Neste caso, tem-se um relacionamento entre a entidade Cliente e Funcionário. Outro exemplo corresponde a entidade Funcionário que se liga a entidade Venda por meio do atributo código_do_funcionário localizado na entidade venda e que representa qual funcionário realizou a venda. É importante notar que o Funcionário não tem relação direta (relacionamento) com a entidade Cliente, considerando que a entidade Venda faz a ligação entre as duas entidades.

3. Quais entidades você definiria para construção de um diagrama de dados para uma oficina mecânica?

Resposta: Cliente, Fornecedor Veículo, Orçamento, Ordem_Serviço, Peças, Serviços, Mecânicos, Serviços_Orçamento, Peças_Orçamento, Serviços_Ordem, Peças_Ordem, Compras, Peças_Compra, Contas a Pagar, Contas a Receber.

4. Construa um Diagrama Entidade Relacionamento (DER), para servir de base para um sistema de uma loja de calçados. Sabe-se que a loja só vende calçados, tem cadastro de clientes, os funcionários da loja vendem à vista e à prazo (30, 60 e 90 dias). Os clientes inadimplentes não podem comprar se possuem contas em atraso.

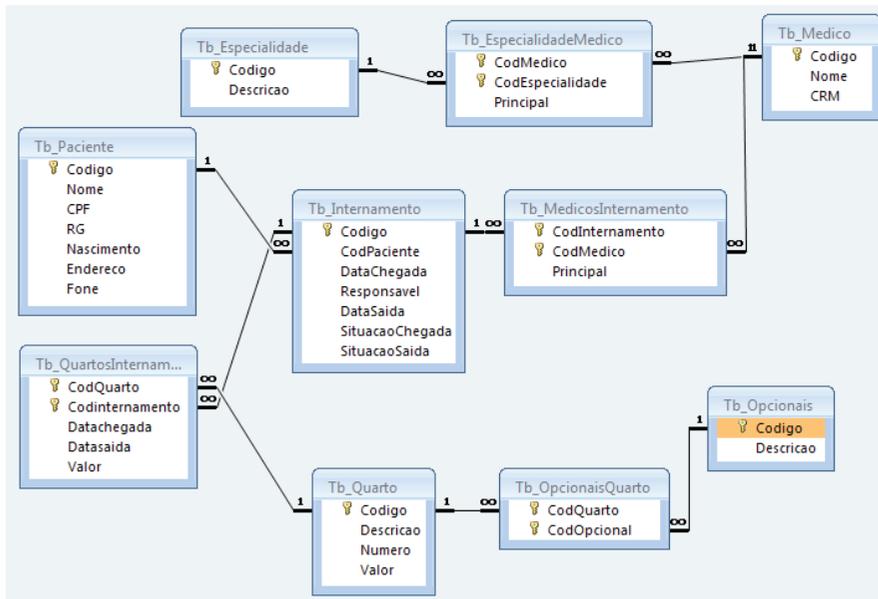


Aula 2 Modelo Entidade Relacionamento

1. Construir os modelos de dados conforme casos hipotéticos:

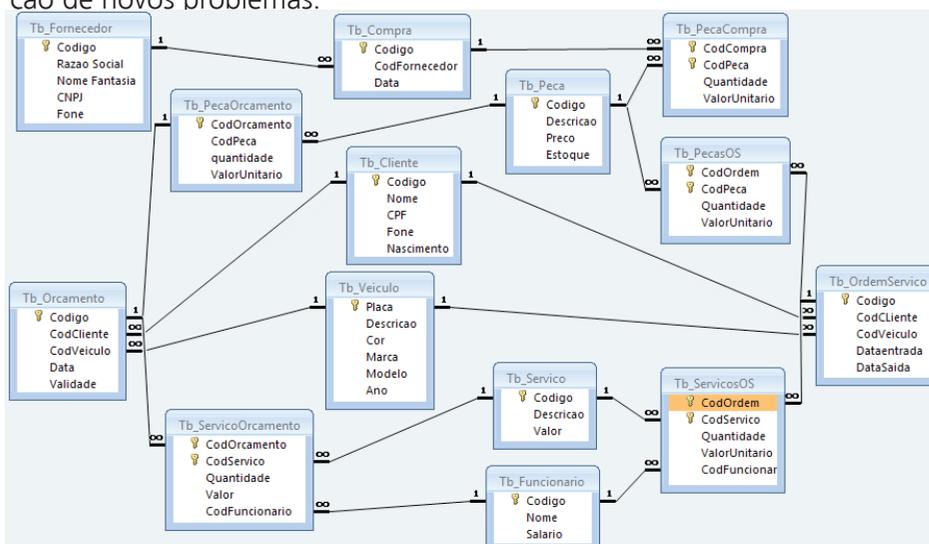
1.1 Caso hipotético 1:

Em uma clínica trabalham médicos e existem pacientes internados. Cada médico é identificado pelo seu CRM, possui um nome e recebe um salário na clínica. Um médico tem formação em diversas especialidades (ortopedia, traumatologia, etc), mas só exerce uma delas na clínica. Para todo paciente internado na clínica são cadastrados alguns dados pessoais: nome, RG, CPF, endereço, telefone(s) para contato e data do nascimento. Um paciente tem sempre determinado médico como responsável (com um horário de visita diário predeterminado), porém vários outros médicos podem participar de seu tratamento. Pacientes estão sempre internados em quartos individuais, que são identificados por um número e ficam em um andar da clínica.



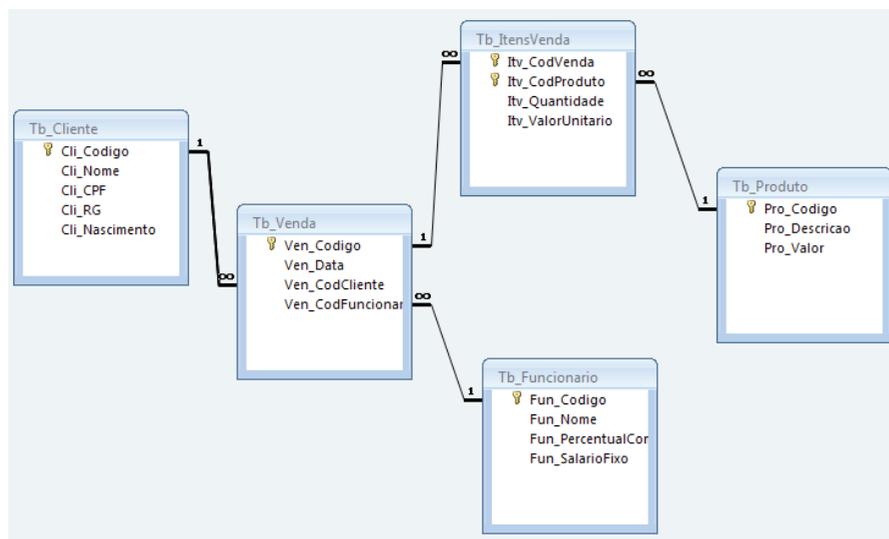
1.2 Caso hipotético 2:

Elabore um modelo de dados para suprir a necessidade de um sistema que vise atender a uma oficina mecânica, onde são realizados orçamentos e serviços. Os clientes normalmente realizam o orçamento; sendo autorizado pelo cliente, o serviço é efetuado. O sistema deve controlar as peças (estoque), bem como manter, em banco, a relação de compras efetuadas. Um serviço ou uma peça podem ter garantia, ressaltando que o sistema deve controlar esse fato. Os mecânicos recebem comissão sobre seus serviços e possuem um salário fixo também. Deve-se ter uma forma de manter um histórico de tudo o que foi elaborado em determinado veículo para facilitar a identificação de novos problemas.



Aula 3. Dicionário De Dados

1. Elaborar o Dicionário de dados para o Modelo Entidade Relacionamento abaixo:



Resposta:

Tb_Cliente				
Entidade:				
Nome da Entidade: Tb_Cliente				Sigla: Cli
Descrição: Refere-se aos dados dos Clientes a serem cadastrados				
Relacionamentos				
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado	
Tb_Venda	1:M	Ven_CodCliente	Chave primária do cliente faz o relacionamento com a entidade Venda	
Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Cli_Codigo	Identificador	Number	CP	Refere-se ao código do Cliente. Não nulo.
Cli_Nome	Simples	Varchar2(60)	CU	Representa o nome do Cliente. Não nulo.
Cli_CPF	Simples	Varchar2(14)	CU	Representa a identificação do Cadastro Nacional de Pessoa Física. Não nulo.
Cli_RG	Simples	Varchar2(15)		Representa a identificação do Cadastro Geral do Cliente.
Cli_Nascimento	Simples	Data		Representa a data de nascimento do cliente.



Tb_Venda				
Entidade:				
Nome da Entidade: Tb_Venda				Sigla: Ven
Descrição: Refere-se aos dados das vendas realizadas (dados de cabeçalho, ou seja, dados de quem vendeu, quando e para quem)				
Relacionamentos:				
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado	
Tb_Cliente	M:1	Ven_CodCliente	Chave primária do cliente faz o relacionamento com a entidade Venda	
Tb_Funcionario	M:1	Ven_CodFuncionario	Chave primária do funcionário faz o relacionamento com a entidade Venda	
Tb_ItensVenda	1:M	Ven_Codigo	Chave primária da Venda faz o relacionamento com a entidade Tb_ItensVenda representando o valor com que o item foi vendido.	
Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Ven_Codigo	Identificador	Number	CP	Refere-se ao código da venda. Não nulo.
Ven_Data	Simples	Data		Representa a data em que a venda foi realizada. Não nulo.
Ven_CodCliente	Simples	Number		Representa a identificação do cliente para quem foi realizada a venda.
Ven_CodFuncionario	Simples	Number		Representa a identificação do funcionário que realizou a venda.

Tb_ItensVenda				
Entidade:				
Nome da Entidade: Tb_ItensVenda				Sigla: Itv
Descrição: Refere-se aos dados referentes aos itens vendidos.				
Relacionamentos:				
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado	
Tb_Venda	M:1	Itv_CodVenda	Faz o relacionamento com a entidade Tb_Venda a qual indica o código da venda com que um item foi vendido.	
Tb_Produto	M:1	Itv_CodProduto	Faz o relacionamento com a entidade Tb_Produto, a qual indica o produto que foi vendido em determinada venda.	
Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Itv_CodVenda	Identificador	Number	CP	Refere-se ao código da venda. Não nulo.
Itv_CodProduto	Identificador	Number	CP	Refere-se ao código do produto. Não nulo.
Itv_Quantidade	Simples	Number		Representa a quantidade vendida de um item em determinada venda.



Itv_ValorUnitario	Simple	Number	Representa o valor de um item em determinada venda.
-------------------	--------	--------	---

Tb_Funcionario				
Entidade:				
Nome da Entidade: Tb_Funcionario				Sigla: Fun
Descrição: Refere-se aos dados dos funcionários				
Relacionamentos:				
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado	
Tb_Venda	1:M	Fun_Codigo	Representa a chave primária da entidade Tb_Funcionario, que é utilizada para se relacionar com a entidade Venda, indicando qual foi o funcionário que realizou determinada venda.	
Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Fun_Codigo	Identificador	Number	CP	Refere-se ao código do funcionário. Não nulo.
Fun_Nome	Simple	Varchar2(60)	CU	Refere-se ao nome do funcionário. Não Nulo.
Fun_PercentualComissao	Simple	Number		Representa o valor percentual que determinado funcionário possui sobre as vendas realizadas
Fun_SalarioFixo	Simple	Number		Representa o valor do salário fixo do funcionário.

Tb_Produto				
Entidade:				
Nome da Entidade: Tb_Produto				Sigla: Pro
Descrição: Refere-se aos dados relativos aos produtos para revenda				
Relacionamentos:				
Com a Entidade	Cardinalidade	Atributo(s)	Nome do relacionamento e significado	
Tb_ItensVenda	M:1	Itv_CodProduto	O atributo Pro_Codigo se relaciona com a entidade Tb_ItensVenda, indicando qual é o produto vendido em determinada venda.	
Elementos de dados:				
Nome do Atributo	Característica	Tipo	Chave	Descrição estendida e observações
Pro_Codigo	Identificador	Number	CP	Refere-se ao código do produto. Não nulo.
Pro_Descricao	Simple	Varchar2(60)	CU	Refere-se à descrição dos produtos cadastrados. Não nulo.
Pro_Valor	Simple	Number		Representa o valor unitário corrente de cada produto.



Aula 4. Normalização De Dados

1. Aplicar as 3 formas normais e criar o Modelo Entidade Relacionamento para o formulário de cadastro de pessoas em um determinado município:

CADASTRO INDIVIDUAL DE MUNÍCIPE

NOME:

CPF: RG: EMISSOR: UF:

DATA: TÍTULO: SEÇÃO: ZONA:

NASCIMENTO: NATURALIDADE:

NACIONALIDADE: SEXO: MASCULINO FEMININO

ESTADO CIVIL: SOLTEIRO CASADO SEPARADO CONCUBINATO VIÚVO DIVORCIADO

DEPENDENTES: ENDEREÇO: NÚMERO:

BAIRRO:

FONE: CELULAR: DATA CHEGADA:

COR: BRANCA NEGRA AMARELO PARDO

NOME DO PAI:

NOME DA MÃE:

OCUPAÇÃO:

ESCOLARIDADE: FORMAÇÃO:

ONDE ESTUDA? PRETENDE VOLTAR ESTUDAR? SIM NÃO

PORTADOR DE NECESSIDADES: FÍSICA MENTAL AUDITIVA VISUAL

ESTUDO PRETENDIDO: SOLICITA CUPOM FISCAL: SIM NÃO

ASSOCIAÇÃO: SINDICATO:

CLUBE: RELIGIÃO:

RESPOSTA:

MUNICIPE (Codigo, nome, cpf, rg, emissor, uf, data, titulo, secao, zona, nascimento, naturalidade, nacionalidade, sexo, estadocivil, dependentes, endereco, numero, bairro, fone, celular, dtchegada, cor, pai, mae, ocupacao, escolaridade, formacao, ondeestuda, pretendevoltarestudar, portador-necessidade, estudopretendido, socilitacupom, associacao, sindicato, clube, religiao)

1 FORMA NORMAL

MUNICIPE (Codigo, nome, cpf, rg, emissor, uf, data, titulo, secao, zona, nascimento, naturalidade, nacionalidade, sexo, estadocivil, dependentes, endereco, numero, bairro, fone, celular, dtchegada, cor, pai, mae, ocupacao, escolaridade, formacao, ondeestuda, pretendevoltarestudar, estudopretendido, socilitacupom, associacao, sindicato, clube, religiao)



NECESSIDADES/MUNICIPE(Codmunicipe, necessidadeespecial)

2 FORMA NORMAL

MUNICIPE (Codigo, nome, cpf, rg, emissor, uf, data, titulo, seção, zona, nascimento, naturalidade, nacionalidade, sexo, estadocivil, dependentes, endereco, numero, bairro, fone, celular, dtchegada, cor, pai, mae, ocupacao, escolaridade, formacao, ondeestuda, pretendevoltarestudar, estudopretendido, socilitacupom, associacao, sindicato, clube, religiao)

NECESSIDADES/MUNICIPE(Codmunicipe, necessidadeespecial)

3 FORMA NORMAL

MUNICIPE (codigo, nome, cpf, rg, codemissor, data, titulo, secao, zona, nascimento, codnaturalidade, codnacionalidade, sexo, codestadocivil, dependentes, codendereco, numero, codbairro, fone, celular, dtchegada, codcor, pai, mae, codocupacao, codescolaridade, codformacao, codondeestuda, pretendevoltarestudar, codestudopretendido, Solicitacupom, codassociacao, codsindicato, codclube, codreligiao)

RELIGIAO(Codigo, descricao)

CLUBE (Codigo, descricao)

SINDICATO (Codigo, descricao)

ASSOCIACAO (Codigo, descricao)

ESCOLA(Codigo, descricao)

FORMACAO (codigo, descricao)

ESCOLARIDADE (Codigo, descricao)

OCUPACAO (Codigo, descricao)

COR (Codigo, descricao)

BAIRRO (Codigo, descricao)



ENDERECO (Codigo, descricao)

ESTADOCIVIL (Codigo, descricao)

NACIONALIDADE (codigo, descricao)

NATURALIDADE (Codigo, descricao)

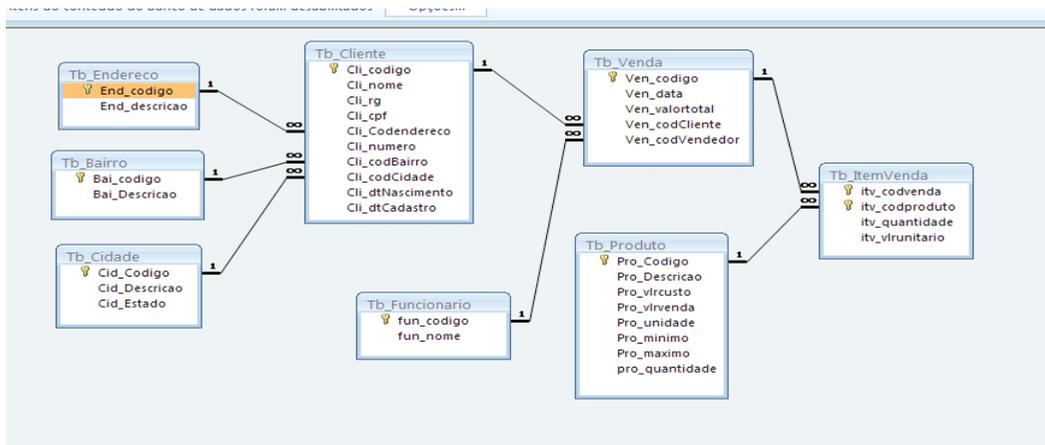
EMISSOR (Codigo, descricao, uf)

NECESSIDADES/MUNICIPE(Codmunicipe, Codnecessidadeespecial)

NECESSIDADEESPECIAL(Codigo, descricao)

Aula 5. Linguagem Sql

1. Construir o modelo de dados utilizando comandos SQL, e inserir em cada entidade pelo menos um registro:



Resposta

COMANDOS DDL (Linguagem de Definição de Dados):

```
CREATE TABLE TB_ENDERECO
```

```
(
  END_CODIGO INTEGER NOT NULL,
  END_DESCRICAO VARCHAR2(60) NOT NULL,
  PRIMARY KEY (END_CODIGO)
);
```



```
CREATE UNIQUE INDEX UK_ENDERECO ON TB_ENDERECO(END_DESCRICA  
CAO);
```

```
CREATE TABLE TB_BAIRRO  
(  
  BAI_CODIGO INTEGER NOT NULL,  
  BAI_DESCRICA VARCHAR2(60) NOT NULL,  
  PRIMARY KEY (BAI_CODIGO)  
);  
CREATE UNIQUE INDEX UK_BAIRRO ON TB_BAIRRO(BAI_DESCRICA);
```

```
CREATE TABLE TB_CIDADE  
(  
  CID_CODIGO INTEGER NOT NULL,  
  CID_DESCRICA VARCHAR2(60) NOT NULL,  
  CID_ESTADO CHAR(2) NOT NULL,  
  PRIMARY KEY (CID_CODIGO)  
);  
CREATE UNIQUE INDEX UK_CIDADE ON TB_CIDADE(CID_DESCRICA  
O, CID_ESTADO);
```

```
CREATE TABLE TB_CLIENTE  
(  
  CLI_CODIGO INTEGER NOT NULL,  
  CLI_NOME VARCHAR2(60) NOT NULL,  
  CLI_RG VARCHAR2(15) NOT NULL,  
  CLI_CPF VARCHAR2(14) NOT NULL,  
  CLI_CODENDERECO INTEGER NOT NULL REFERENCES TB_  
ENDERECO(END_CODIGO),  
  CLI_NUMERO INTEGER,  
  CLI_CODBAIRRO INTEGER NOT NULL REFERENCES TB_BAIRRO(BAI_  
CODIGO),  
  CLI_CODCIDADE INTEGER NOT NULL REFERENCES TB_CIDADE(CID_  
CODIGO),  
  CLI_DTNASCIMENTO DATE NOT NULL,  
  CLI_DTCADASTRO DATE NOT NULL,  
  PRIMARY KEY (CLI_CODIGO)  
);  
CREATE UNIQUE INDEX UK_CLIENTE ON TB_CLIENTE (CLI_CPF);
```

```
CREATE TABLE TB_FUNCIONARIO  
(  
  FUN_CODIGO INTEGER NOT NULL,  
  FUN_NOME VARCHAR2(60),  
  PRIMARY KEY (FUN_CODIGO)  
);
```

```
CREATE TABLE TB_PRODUTO  
(  
  PRO_CODIGO INTEGER NOT NULL,  
  PRO_DESCRICA VARCHAR2(60) NOT NULL,  
  PRO_VLRCUSTO DECIMAL(10,2) NOT NULL,  
  PRO_VLRVENDA DECIMAL(10,2) NOT NULL,  
  PRO_UNIDADE VARCHAR2(20) NOT NULL,
```



```
PRO_MINIMO DECIMAL(10,2),
PRO_MAXIMO DECIMAL(10,2),
PRO_QUANTIDADE DECIMAL (10,2),
PRIMARY KEY (PRO_CODIGO)
);
CREATE UNIQUE INDEX UK_PRODUTO ON TB_PRODUTO (PRO_DESCRICA
CAO);

CREATE TABLE TB_VENDA
(
VEN_CODIGO INTEGER NOT NULL,
VEN_DATA DATE NOT NULL,
VEN_VALORTOTAL DECIMAL (10,2),
VEN_CODCLIENTE INTEGER NOT NULL REFERENCES TB_CLIENTE(CLI_
CODIGO),
VEN_CODVENDEDOR INTEGER NOT NULL REFERENCES TB_
FUNCIONARIO(FUN_CODIGO),
PRIMARY KEY (VEN_CODIGO)
);

CREATE TABLE TB_ITEMVENDA
(
ITV_CODVENDA INTEGER NOT NULL REFERENCES TB_VENDA(VEN_
CODIGO),
ITV_CODPRODUTO INTEGER NOT NULL REFERENCES TB_PRODUTO(PRO_
CODIGO),
ITV_QUANTIDADE DECIMAL(10,2) NOT NULL,
ITV_VLRUNITARIO DECIMAL(10,2) NOT NULL,
PRIMARY KEY (ITV_CODVENDA, ITV_CODPRODUTO)
);
```

COMANDOS DML (LINGUAGEM DE MANIPULAÇÃO DE DADOS):

```
INSERT INTO TB_ENDERECO (END_CODIGO, END_DESCRICA) VALUES (1,
'RUA BAHIA');
```

```
INSERT INTO TB_BAIRRO (BAI_CODIGO, BAI_DESCRICA) VALUES (1,
'CONDÁ');
```

```
INSERT INTO TB_CIDADE (CID_CODIGO, CID_DESCRICA, CID_ESTADO)
VALUES (1, 'MEDIANEIRA', 'PR');
```

```
INSERT INTO TB_CLIENTE (CLI_CODIGO, CLI_NOME, CLI_RG, CLI_CPF, CLI_
CODENDERECO, CLI_NUMERO, CLI_CODBAIRRO, CLI_CODCIDADE, CLI_DT-
NASCIMENTO, CLI_DTCADASTRO) VALUES (1, 'CLAUDIO LEONES BAZZI',
'2324234-9', '009.098.943-22', 1, 1050, 1, 1, '02/07/1980', '20/03/2001');
```



```
INSERT INTO TB_FUNCIONARIO (FUN_CODIGO, FUN_NOME) VALUES (1,  
'JOSÉ DOS SANTOS');
```

```
INSERT INTO TB_PRODUTO (PRO_CODIGO, PRO_DESCRICA0, PRO_VLR-  
CUSTO, PRO_VLRVENDA, PRO_UNIDADE, PRO_MINIMO, PRO_MAXIMO,  
PRO_QUANTIDADE) VALUES (1, 'CAFÉ SOLÚVEL NESCAFÉ 200G', 2.90,  
4.50, 'FRASCO', 10, 50, 10);
```

```
INSERT INTO TB_VENDA (VEN_CODIGO, VEN_DATA, VEN_CODCLIENTE,  
VEN_CODVENDEDOR) VALUES (1, '10/10/2001', 1, 1);
```

```
INSERT INTO TB_ITEMVENDA (ITV_CODVENDA, ITV_CODPRODUTO, ITV_  
QUANTIDADE, ITV_VLRUNITARIO) VALUES (1, 1, 5, 4.5);
```





Referências

MACHADO, F. N. R, e ABREU, M. P. **Projeto de Banco de Dados**: uma visão prática. São Paulo: Editora Érica, 1995.

ORACLE. **Oracle Database 10g**: SQL Fundamentals. Student Guide. Oracle Corporation. Ed. 2, 2006.





Obras Consultadas

HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. Porto Alegre: Sagra-Luzzatto, 2000.

LIGHTSTONE, Sam; NADEAU, Tom; TEOREY, Toby. **Projeto e Modelagem de Bancos de Dados**. São Paulo: Campus, 2006.

MACHADO, Felipe N. R. **Projeto de Banco de Dados: Uma visão prática**. São Paulo: Érica, 2004.

SILVA, Ivan J. de M.; OLIVEIRA, Vivianne de. **Banco de dados: do modelo conceitual à implantação física**. Rio de Janeiro: Alta Books, 2005.

SILBERSCHATZ, Abraham. **Sistema de Banco de Dados**. São Paulo: Makron Books, 1999.





Currículo do Professor-autor



Cláudio Leones Bazzi, graduado em Tecnologia em Processamento de Dados pelo Centro de Ensino Superior de Foz do Iguaçu (2001), especialização em Computação Aplicada (2002), especialização em Redes de Computadores (2007), mestrado em Engenharia Agrícola pela Universidade Estadual do Oeste do Paraná (2007). Doutor em Engenharia Agrícola da Universidade Estadual do Oeste do Paraná (2011), professor Adjunto da Universidade Tecnológica Federal do Paraná, coordenador do curso de especialização em Bancos de Dados - Administração e Desenvolvimento e coordenador do curso de Bacharelado em Ciência da Computação do campus Medianeira da UTFPR. Tem experiência na área de Ciência da Computação, com ênfase em Desenvolvimento de Sistemas, atuando principalmente nos seguintes temas: computação aplicada, bancos de dados relacionais e bancos de dados geográficos, análise de dados.



