

## Sistemas de Banco de Dados

### Continuando na atividade de projeto

Segundo Yourdon (1994), comentando sobre o ciclo de vida de projeto estruturado, a maior parte do tempo das atividades do analista são gastas na atividade de análise de sistemas do ciclo de vida mencionado. Anteriormente, na atividade de levantamento, a viabilidade do projeto foi estabelecida e seu escopo foi traçado em linhas gerais, com intenso envolvimento dos usuários. A atividade seguinte, denominada *projeto*, trata de importantes tarefas que cabem ao analista, entre as quais a transformação de modelos de dados de entidades-relacionamentos em um projeto de banco de dados, como pode ser conferido na Apostila 2 (Oliveira, 2001).

Este assunto será tratado neste material, sem perder de vista a idéia de que não se trabalha em um modelo sequencial, ou seja, as fases não precisam ser encerradas para que se comece ou continue o trabalho de uma atividade seguinte (ou anterior). Assim, a atividade de análise será retomada, ainda, considerando outros aspectos, bem como outras tarefas pertinentes à atividade de projeto serão comentadas.

### Introdução ao modelo relacional

Neste material, será feita uma pequena introdução sobre o modelo de dados relacional, com comentários sobre sua estrutura básica, sobre o conceito de esquema de banco de dados, sobre chaves, além de um método para o mapeamento do DER para o modelo relacional e normalização de tabelas.

Segundo Korth e Silberschatz (1995, p.58), “um banco de dados relacional consiste em uma *coleção de tabelas*, cada qual designada por um nome único”. Tais tabela também são conhecidas como *relações*.

## Estrutura básica

Considerando a tabela denominada *depósito*, representada na Figura 1, podemos verificar a existência de *atributos*, os quais, no caso, são *nome-agência*, *número-conta*, *nome-cliente* e *saldo*. Cada atributo possui um *domínio*, que é um conjunto de valores permitidos (ou válidos). Qualquer linha da tabela *depósito* deverá ser constituída por um conjunto de quatro valores válidos, cada qual correspondente a um dos atributos da tabela. Então, uma linha de *depósito* contém (v1, v2, v3, v4), onde:

v1 = um nome de agência, contido no domínio de *nome-agência*;

v2 = um número de conta, contida no domínio de *número-conta*;

v3 = um nome de cliente, contido no domínio de *nome-cliente*;

v4 = um saldo, contido no domínio de *saldo*.

Uma linha de uma tabela ou relação qualquer é geralmente chamada de *tupla*.

## Esquema de banco de dados

Uma importante diferenciação precisa ser feita entre os conceitos de *esquema de banco de dados* (ou projeto lógico de um banco de dados) e uma *instância de um banco de dados*, “que é o dado dentro do banco de dados em um determinado instante” (Korth e Silberschatz, 1995, p.61). Ainda segundo os autores mencionados, é importante atribuir um nome qualquer a um esquema relacional, de maneira que, do exemplo citado até agora, resultaria:

*Esquema-depósito = (nome-agência, número-conta, nome-cliente, saldo)*

Ou seja:

$$\text{depósito} = (\text{Esquema-depósito})$$

Conforme a convenção recomendada pelos autores supramencionados, serão utilizadas iniciais *minúsculas* para relações e iniciais *maiúsculas* para esquemas de relações.

<i>nome-agência</i>	<i>número-conta</i>	<i>nome-cliente</i>	<i>saldo</i>
Downtown	101	Johnson	500
Mianus	215	Smith	700
Perryidge	102	Hayes	400
Round Hill	305	Turner	350
Perryidge	201	Williams	900
Redwood	222	Lindsay	700
Brighton	217	Green	750
Downtown	105	Green	850

Figura 1 – A relação depósito – Fonte: Korth e Silberschatz, 1995, p.60

O esquema da relação *cliente*, demonstrada na Figura 2, seria:

$$\text{Esquema-cliente} = (\text{nome-cliente}, \text{rua}, \text{cidade-cliente})$$

O atributo *nome-cliente* pode ser encontrado nos dois esquemas relacionais mencionados até agora. Isto pode ser bastante útil caso seja necessário relacionar tuplas de relações distintas.

<i>nome-cliente</i>	<i>rua</i>	<i>cidade-cliente</i>
Jones	Main	Harrison
Smith	North	Rye
Hayes	Main	Harrison
Curry	North	Rye

Lindsay	Park	Pittsfield
Turner	Putnam	Stamford
Williams	Nassau	Princeton
Adams	Spring	Pittsfield
Johnson	Alma	Palo Alto
Glenn	Sand Hill	Wooside
Brooks	Senator	Brooklyn
Green	Walnut	Stamford

Figura 2 – A relação cliente – Fonte: Korth e Silberschatz, 1995, p.62

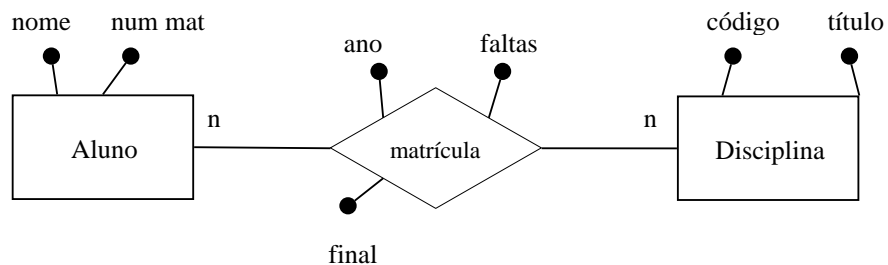
## Chaves

Este assunto já havia sido tratado anteriormente, na apostila anterior. As noções constantes neste material (superchave, chave candidata e chave primária) são plenamente aplicáveis ao modelo relacional. Nos exemplos utilizados até aqui, pode-se afirmar que *nome-cliente* é a chave primária para *Esquema-cliente*, enquanto *número-conta* é a chave primária para *Esquema-depósito*.

## Mapeamento do DER para o modelo relacional

- 1) Conjunto de entidades fortes: todo conjunto de entidades fortes do DER gera uma tabela no modelo relacional.

Exemplo:



*Esquema-aluno : (num-mat, nome)*

*Esquema-disciplina: (código, título)*

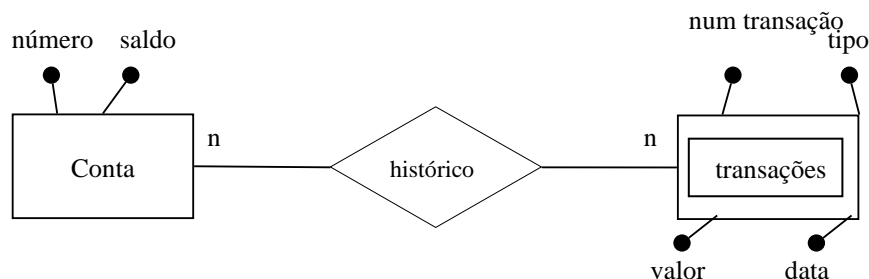
- 2) Conjunto de relacionamentos  $n$  para  $n$  ou que tenham atributos: os conjuntos de relacionamentos  $n$  para  $n$  sempre geram tabelas no modelo relacional, contendo as chaves primárias das entidades associadas mais os atributos próprios do conjunto de relacionamentos (se existirem).

Exemplo (considerando a figura do exemplo anterior):

*Esquema-matrícula: (num-mat, código, ano, faltas, final)*

- 3) Conjunto de entidades fracas: Os conjuntos de entidades fracas geram tabelas no modelo relacional, contendo seus atributos mais as chaves primárias dos conjuntos de entidades fortes dos quais dependem existencialmente (chaves estrangeiras).

Exemplo:



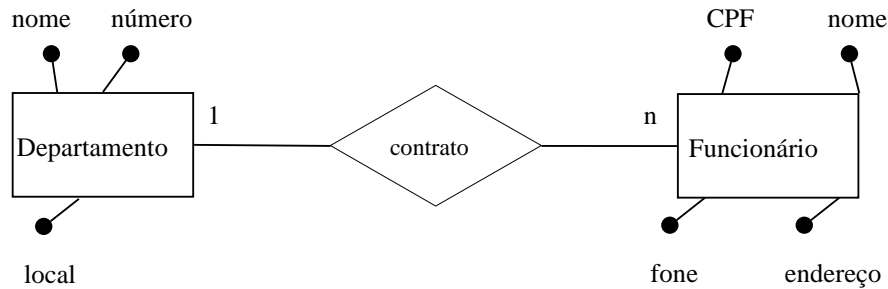
*Esquema-contas: (número, saldo)*

*Esquema-transações: (número, num-transação, tipo, valor, data)*

- 4) Conjunto de relacionamentos 1 para  $n$  (ou  $n$  para 1 ou, ainda, 1 para 1) sem atributos: Não geram tabelas. Utiliza-se o conceito de chave estrangeira, ou

seja, a entidade que só se associa a uma entidade do outro conjunto recebe a chave estrangeira.

Exemplo:



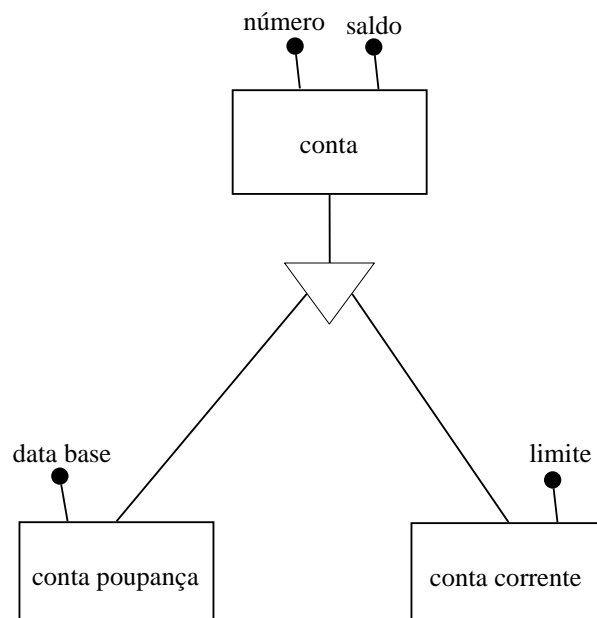
*Esquema-departamento: (número, nome, local)*

*Esquema-funcionário: (CPF, nome, endereço, fone, número)*

5) Generalização e especialização: Há duas formas para definir as tabelas para uma generalização ou especialização:

a) Gera-se uma tabela para a entidade de nível mais alto com seus atributos, mais uma tabela para cada entidade de nível mais baixo, com a chave primária da entidade de nível superior além de seus atributos próprios;

Exemplo:



*Esquema-conta: (número, saldo)*

*Esquema-conta-corrente: (número, limite)*

*Esquema-conta-poupança: (número, data-base)*

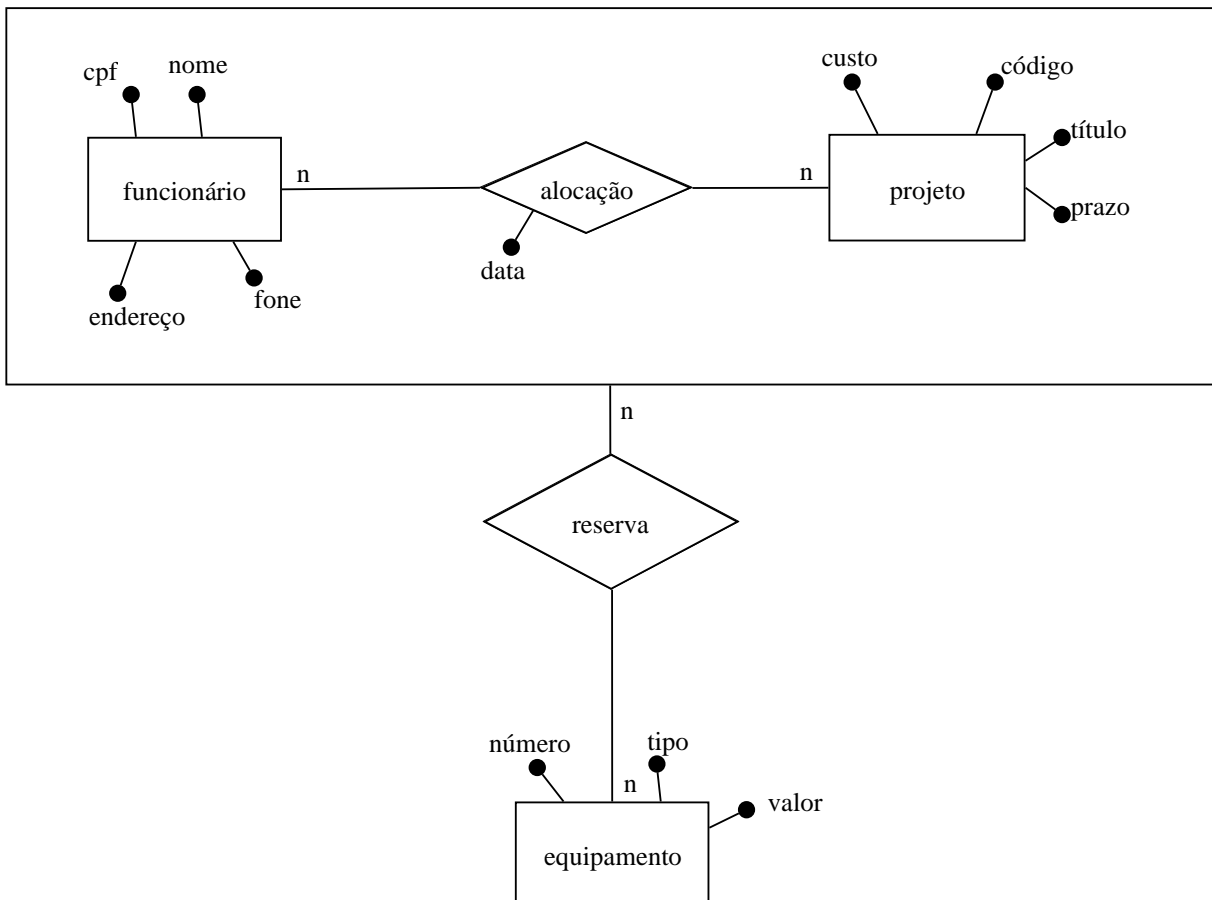
- b) Gera-se apenas as tabelas para os conjuntos de entidades de nível mais baixo, com todos os atributos herdados do conjunto de entidades do nível mais alto, mais os atributos próprios.

Exemplo:

*Esquema-conta-corrente: (número, saldo, limite)*

*Esquema-conta-poupança: (número, saldo, data-base)*

### Exemplos de Mapeamento do DER para o modelo relacional



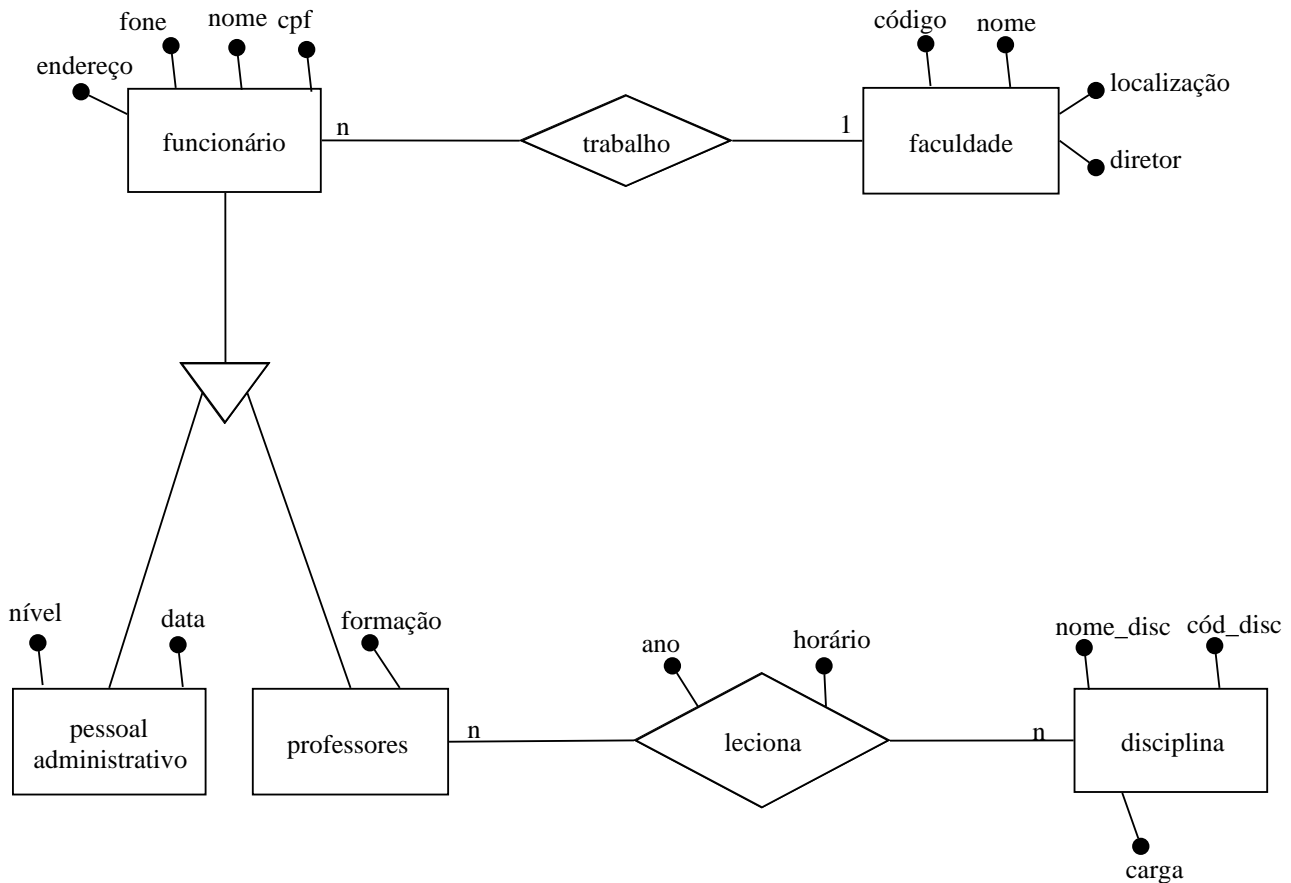
Esquema-funcionário: (cpf, nome, endereço, fone);

Esquema-projeto: (código, título, prazo, custo);

Esquema-alocação: (cpf, código, data);

Esquema-equipamento: (número, tipo, valor);

Esquema-reserva: (número, cpf, código, data).



Esquema-funcionário: (cpf, nome, endereço, fone, código);

Esquema-faculdade: (código, nome, localização, diretor);

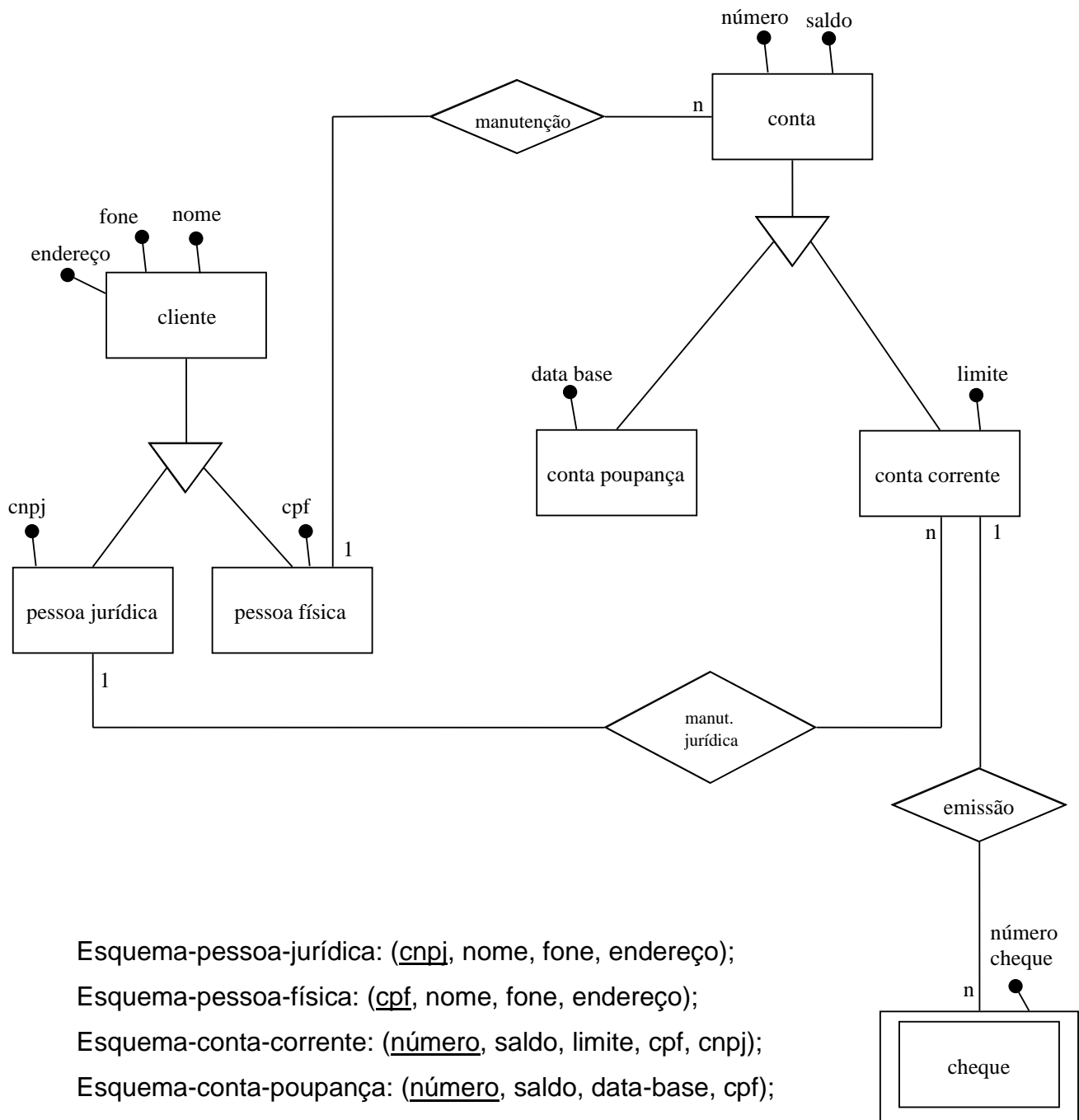
Esquema-pessoal-administrativo: (cpf, nível, data);

Esquema-professores: (cpf, formação);

Esquema-disciplina: (cód\_disc, nome\_disc, carga);

Esquema-leciona: (cód\_disc, cpf, ano, horário).





Esquema-pessoa-jurídica: (cnpj, nome, fone, endereço);

Esquema-pessoa-física: (cpf, nome, fone, endereço);

Esquema-conta-corrente: (número, saldo, limite, cpf, cnpj);

Esquema-conta-poupança: (número, saldo, data-base, cpf);

Esquema-cheque: (número, número-cheque).

## Normalização

De maneira resumida, pode-se dizer que a normalização é uma técnica para simplificar o projeto de um banco de dados relacional. Os principais objetivos desta técnica são:

- Eliminar redundâncias;
- Evitar problemas de atualização;
- Implementar estruturas de dados mais estáveis.

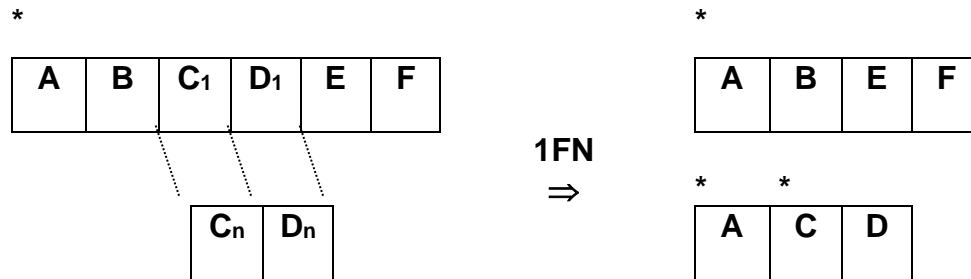
Exemplo:

### NOTA FISCAL

Número: 5897			Data: 20/02/2023	
CNPJ_dest: 999.999.999/0009-99				
Nome_dest: Maria da Silva				
Endereço: Rua X, 1000 – São Paulo – SP				
Telefone: (0xx11) 9999-9999				
Qtde	Cód	Descr	Preço_unit	Preço_item
10	001	prego	0,10	1,00
12	002	parafuso	0,12	1,44
05	003	arruela	0,05	0,25
Total				2,69

Esquema\_nota\_fiscal: (número, data, CNPJ\_dest, nome\_dest, endereço, telefone, qtde, cód, descr, preço\_unit, preço\_item, total)

Primeira forma normal (1FN): um modelo está na 1FN se é constituído por tabelas onde não existem atributos multivalorados (ou grupos repetitivos).



\* = chave primária

Para eliminar atributos multivalorados, gera-se uma nova tabela com a chave primária da tabela original mais o grupo de atributos multivalorados. Assim:

NOTA FISCAL (1FN):

Esquema\_nota\_fiscal: (número, data, CNPJ\_dest, nome\_dest, endereço, telefone, total);

Esquema\_itens: (número, qtde, cód, descr, preço\_unit, preço\_item).

Antes de mencionar as outras formas normais, imprescindível conhecer alguns conceitos:

- **Dependência funcional:** um atributo qualquer B é dependente funcional de outro atributo qualquer A ( $A \rightarrow B$ ) se, e somente se, para cada valor de A existe a correspondência de um único valor de B.

Exemplos:

$cód \rightarrow descr$

$cód \rightarrow preço\_unit$

$(cód, qtde) \rightarrow preço\_item$

- Dependência funcional total: ocorre quando um atributo depende de dois outros, mas não depende de um deles individualmente, ou seja,  $(A, B) \rightarrow C$ , mas  $A \not\rightarrow C$  e  $B \not\rightarrow C$ .

Exemplo:

$(\text{cód}, \text{qtde}) \rightarrow \text{preço\_item}$ , mas

$\text{cód} \not\rightarrow \text{preço\_item}$

$\text{qtde} \not\rightarrow \text{preço\_item}$

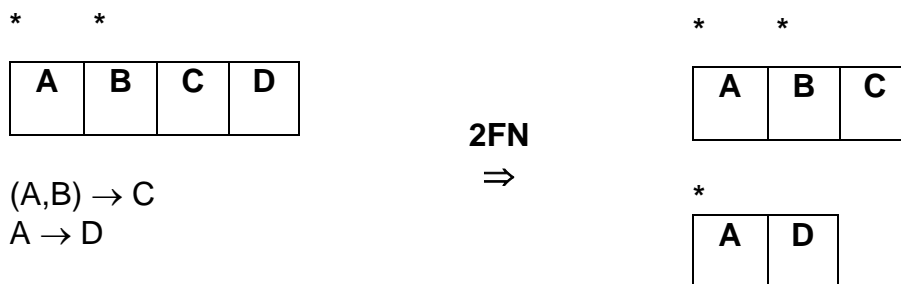
- Dependência funcional parcial: ocorre quando um atributo depende de dois outros, além de depender de um deles individualmente, também. Assim,  $(A, B) \rightarrow C$  e  $A \rightarrow C$  ou  $B \rightarrow C$ .

Exemplo:

$(\text{número}, \text{cód}) \rightarrow \text{preço\_unit}$ , mas

$\text{cód} \rightarrow \text{preço\_unit}$

Segunda forma normal (2FN): um modelo está na 2FN se, e somente se, estiver na 1FN e, para cada tabela, não existir nenhum atributo que depende de uma parte da chave primária (não exista, portanto, dependência funcional parcial com relação à chave primária). Se a tabela tem uma chave primária simples e está na 1FN, então também está na 2FN.

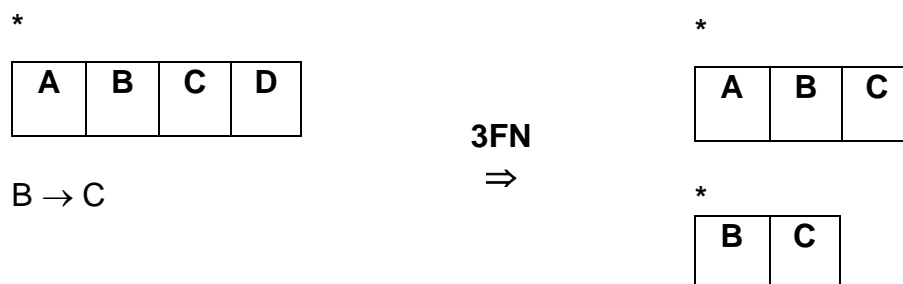


Esquema\_nota\_fiscal: (número, data, CNPJ\_dest, nome\_dest, endereço, telefone, total);

Esquema\_nota\_fiscal: (número, data, CNPJ\_dest, nome\_dest, endereço, telefone, total);

Esquema\_produtos: (cód, descr, preço\_unit).

Terceira forma normal (3FN): um modelo está na 3FN se, e somente se, estiver na 2FN e, para cada tabela, não existir nenhum atributo que dependa de algum outro atributo que não seja a chave primária.



#### NOTA FISCAL (1FN):

Esquema\_nota\_fiscal: (número, data, CNPJ\_dest, nome\_dest, endereço, telefone, total);

Esquema\_itens: (número, qtde, cód, descr, preço\_unit, preço\_item);

#### NOTA FISCAL (2FN):

Esquema\_nota\_fiscal: (número, data, CNPJ\_dest, nome\_dest, endereço, telefone, total);

Esquema\_itens: (número, qtde, cód, preço\_item);

Esquema\_produtos: (cód, descr, preço\_unit).

#### NOTA FISCAL (3FN):

Esquema\_nota\_fiscal: (número, data, CNPJ\_dest, total);

Esquema\_clientes: (CNPJ\_dest, nome\_dest, endereço, telefone);

Esquema\_itens: (número, qtde, cód, preço\_item);

Esquema\_produtos: (cód, descr, preço\_unit).

Em resumo, uma tabela está normalizada quando todo atributo depende da chave primária e somente dela, e não há atributos multivalorados.

## Exercícios de normalização

### 1) Livros de uma biblioteca:

Esquema\_livros: (cód livro, título, nome\_autor, nome\_editora, num\_pag, gênero, CNPJ\_forn, nome\_forn, end\_forn, fone\_forn, data\_aquisição, cód\_cli, nome\_cli, end\_cli, fone\_cli, data\_retirada, data\_devolução, custo\_empréstimo)

### 2) Turmas de uma escola de informática:

Esquema\_turmas: (cód turma, cód\_curso, descrição\_curso, carga\_horária, data\_início, horário, local, cód\_prof, nome\_prof, end\_prof, fone\_prof, cód\_aluno, nome\_aluno, end\_aluno, fone\_aluno, frequência, nota)

### 3) Seguros:

Esquema\_seguro: (CPF\_cliente, nome\_cliente, end\_cliente, fone\_cliente, chassi\_veículo, placas\_veículo, marca\_veículo, modelo\_veículo, cor\_veículo, ano\_veículo, nome\_corretor, cód corretor, fone\_corretor, num\_sinistro, data\_sinistro, tipo\_sinistro, local\_sinistro, extensão\_sinistro, hora\_sinistro, tipo\_contrato\_veic, cobertura\_contrato\_veic, custo\_contrato\_veic, data\_contrato\_veic, franquia\_contrato\_veic, validade\_contrato\_veic)

### 4) Manufatura:

Esquema\_manufatura: (cód prod, descrição\_prod, preço\_prod, qtde\_mp\_prod, CNPJ\_forn, nome\_forn, fone\_forn, ende\_forn, cód mp, descrição\_mp, cor\_mp, tamanho\_mp, data\_mp\_pedido, qtde\_mp\_pedido, preço\_mp\_pedido, número\_mp\_pedido)

**Bibliografia :**

KORTH, Henry F., SILBERSCHATZ, Abraham. *Sistema de Banco de Dados*. São Paulo: Makron Books, 1995.

YOURDON, Edward. *Análise Estruturada Moderna*. Rio de Janeiro: Campus, 1994.

OLIVEIRA, Gerson P. *Curso de Análise e Projeto de Sistemas*. Material de aula. São Paulo: FIAP, 2001.