Normalização de Dados

É um conjunto de limitações impostas a uma estrutura de dados para que a mesma fique mais concisa e sem falhas estruturais.

* Codd -> 1970
* DATE -> 3FN (aperfeiçoada)
* FAGIN -> 4FN e 5FN -> 1977

DEPENDÊNCIA FUNCIONAL

Sejam E uma entidade, X e Y dois atributos quaisquer de E. Diz-se que Y é funcionalmente dependente de X se e somente se cada valor de X estiver associado a exatamente um valor de Y.

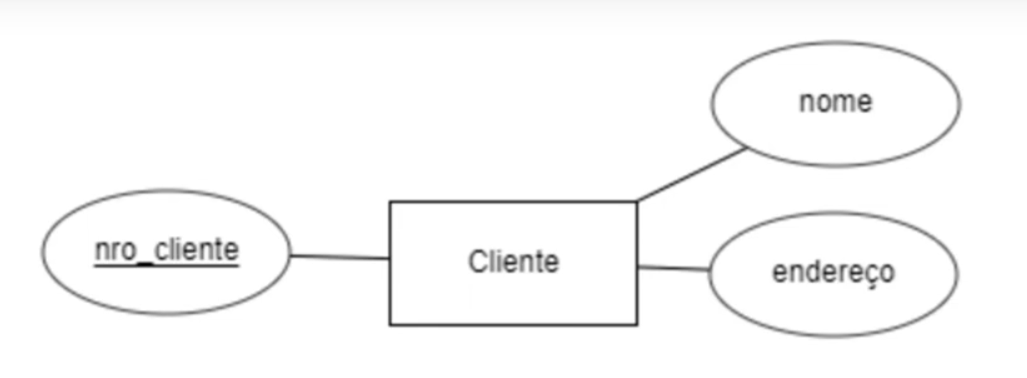
Notação:

X -> Y

Onde, X determina funcionalmente Y, ou Y depende funcionalmente de X.

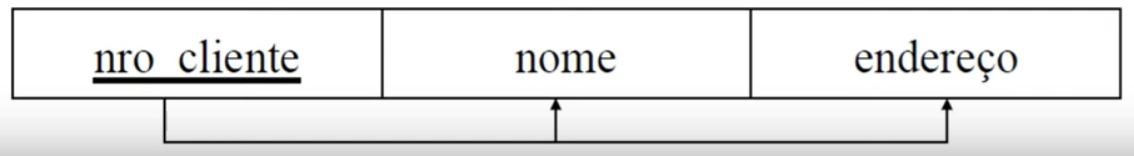
Exemplo -> numero\_pedido -> prazo\_entrega

Anotação gráfica desta relação:



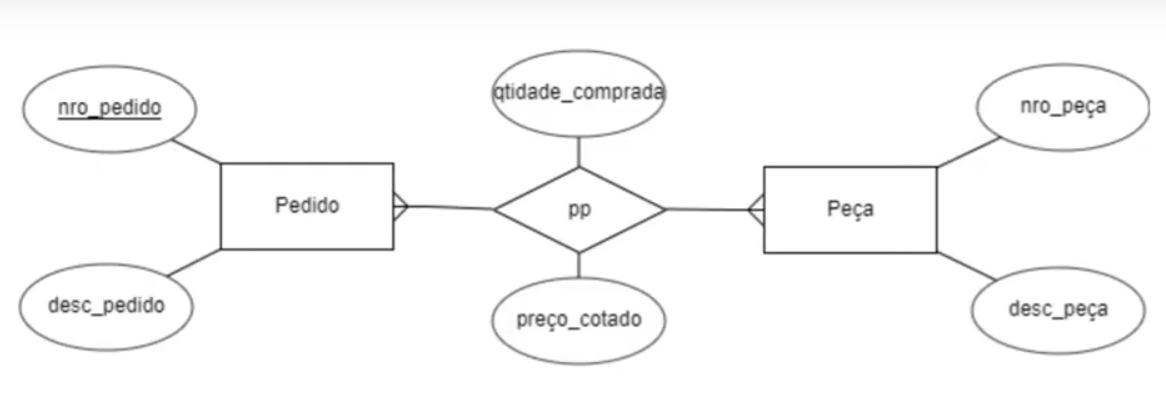
cliente(nro\_cliente, nome, endereço)

nro\_cliente -> {nome, endereço}



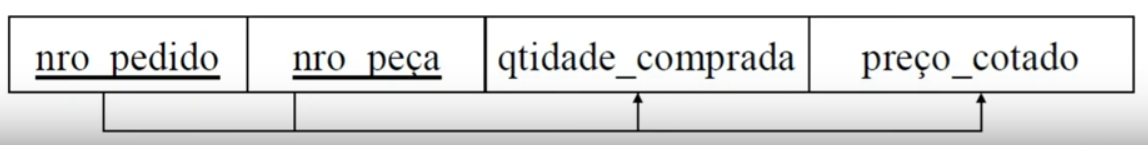
Dependência funcional total

Em uma relação com uma PK composta, um atributo não-chave que depende dessa PK como um todo, e não somente como parte dela, possui dependência funcional total.



pp (nro\_pedido, nro\_preça, qtidade\_comprada, preço\_cotado}

{nro\_pedido, nro\_peça} -> {qtidade\_comprada, preço\_cotado}



Dependência funcional parcial

Há dependência funcional parcial quando um atributo não-chave depende apenas de parte de uma chave primária composta.

(Há dependência funcional, mas ela é de parte parcial da chave, apenas)

Normalmente, ocorre quando há dados no lugar errado ou quando há redundância de dados

Dependência funcional transitiva

Ocorre quando um atributo não depende funcionalmente da chave primária, mas depende de outro atributo não-chave.

Normalmente, ocorre quando há má modelagem dos dados.

X -> Y e Y -> Z

Dependência multivalorada

Ocorre quando, para cada valor de um atributo A, existe um conjunto de valores de outros atributos B e C que estão associados a ele, mas que são independentes entre si.

Formalização: um atributo B de uma relação R é multidependente de um atributo A de R se um valor de A é associado a uma coleção de valores de B, independentemente de qualquer valor de outro atributo C de R.

Notação:

A ->-> B ou A ->> B

NORMALIZAÇÃO

Processo de análise que visa assegurr que uma relação está bem formada.

Ideia: decompor relações co manomalias para produzir relações menores e mais bem estruturadas.

Relações normalizadas -> inserção, remoção e atualização isentas de anomalias

Anomalias

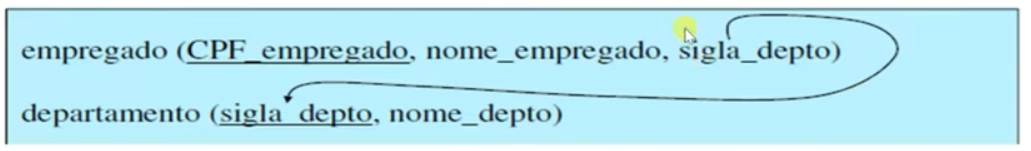
São problemas que ocorrem em BDs.

Decorrem de mau planejamento

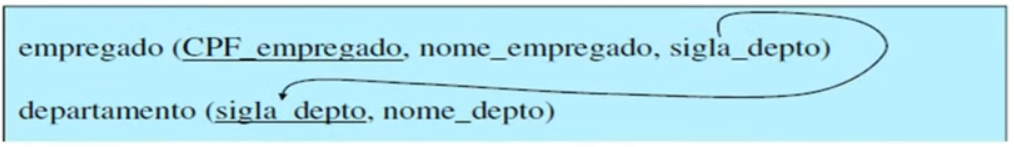
* Excesso de dados em uma mesma relação
* Dados em locais errados

Três tipos

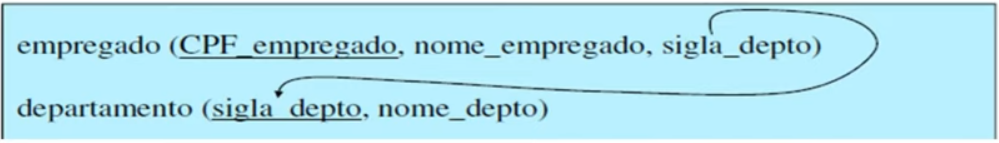
1. Anomalia de inserção



1. Anomalia de remoção



1. Anomalia de atualização



Normalização

Conceito introduzido em 1972 por Codd

São testes para certificar que uma relação satisfaz formas normais (FN).

Codd propôs que boas relações atendem a pelo menos 3 formas normais.

Existem outras formas:

- Forma normal de Boyce-Codd (FNBC)

- 4 FN

- 5 FN

1 FN

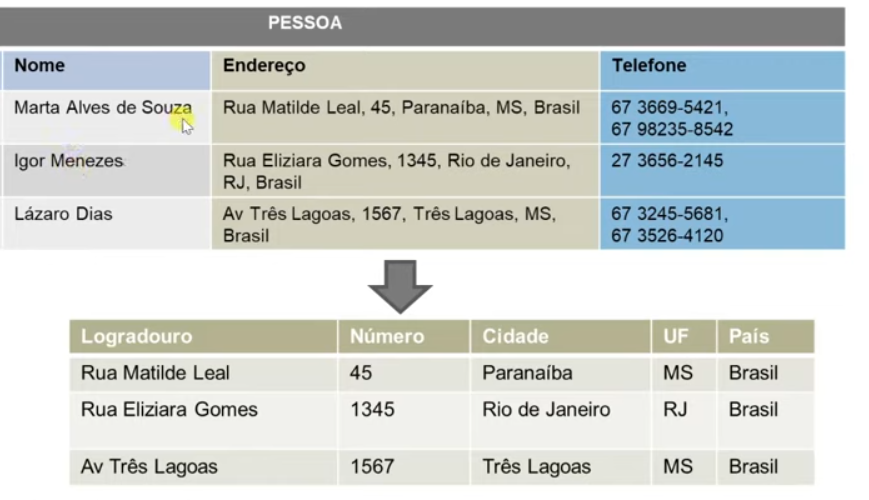
Reprova atributos multivalorados, compostos e suas combinações



Endereço: é um atributo composto, pois podemos separar os valores dele

Telefone: é multivalorado por ter mais de um valor no mesmo campo

1 Passo:



2 Passo:



Uma relação está na 1FN se:

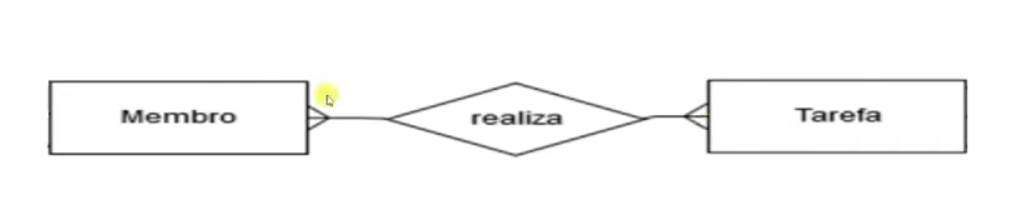
* Todos os valores são atômicos (indivisíveis)
* Há apenas um dado por coluna
* Existe uma chave primária (pelo menos)
* Não há relações aninhadas (tabelas dentro de tabelas)

2 FN

Uma relação está na 2FN se:

* Está na 1FN
* Não contém dependências parciais (Todas com dependência total)





N para N



Uma relação está em 2FN se:

- Se está em 1FN

- Se a chave primária for simples (apenas um atributo)

- Se nenhum atributo não-chave existe na relação (todos os atributos são parte da chave primária)

- Se todo atributo não-chave é dependente funcionalmente de todo o conjunto de atributos da chave primária

3 FN

Uma relação está na 3FN se está na 2FN e não há atributo não-chave determinado funcionalmente por outro atributo não-chave.

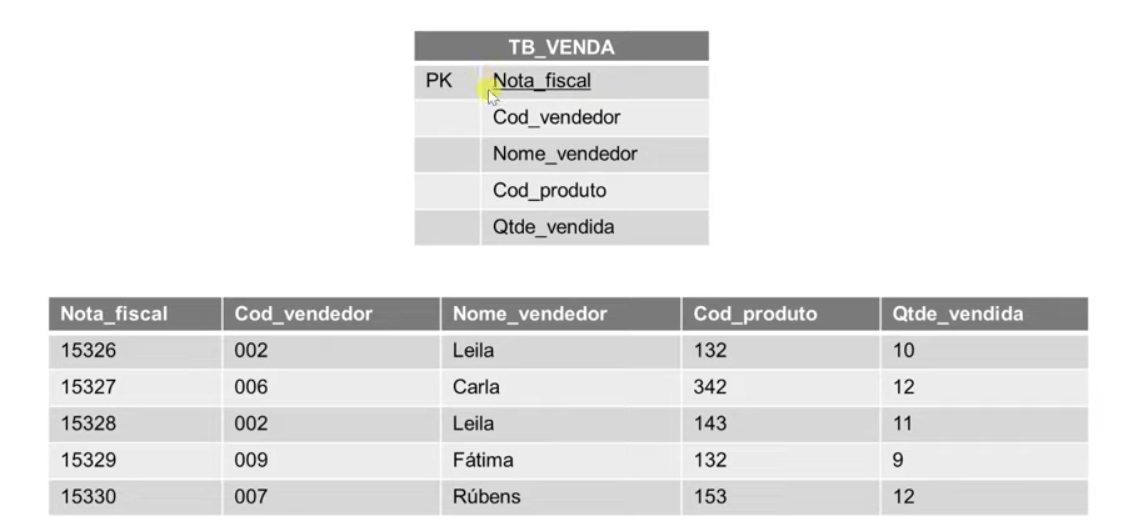
- Não há dependência transitiva

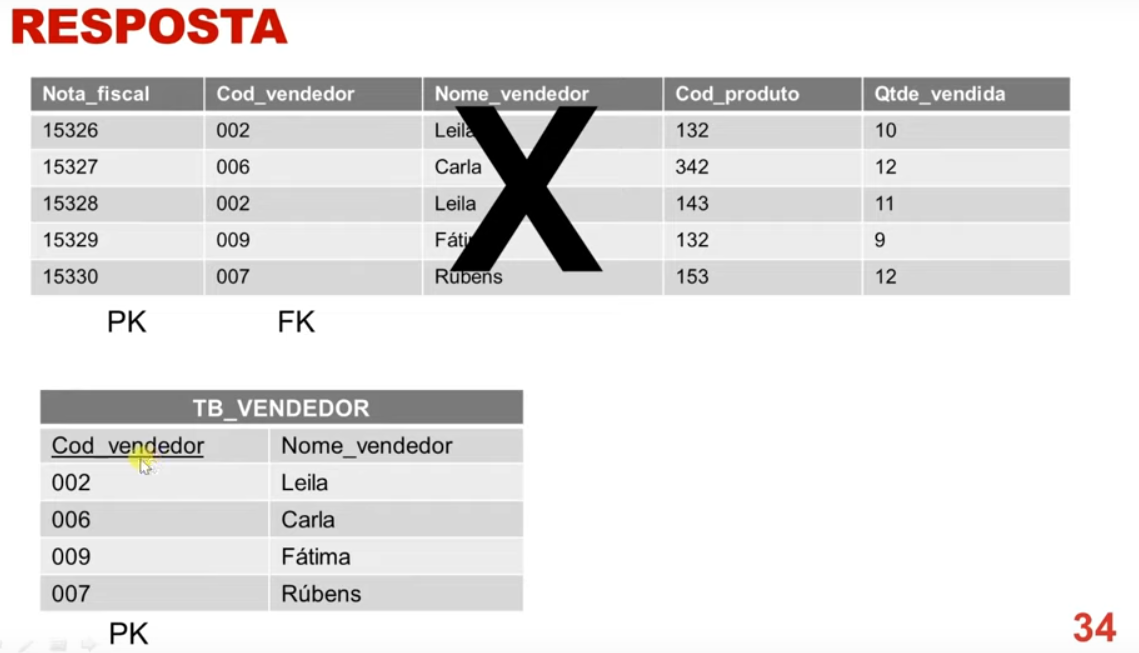
Solução para relações fora da 3FN

- Para cada atributo não-chave que determina funcionalmente outro atributo, crie uma nova tabela.

Este atributo não-chave será a PK desta nova tabela.

Mova todos os atributos dependentes funcionalmente da nova PK para a nova tabela.





SQL {DDL, DML, DQL, DCL, DTL}

Desenvolvida nos anos 70 pela IBM

DDL – Data Definition Language

DML – Data Manipulation Language

DQL – Data Query Language

DCL – Data Control Language

DTL – Data Transaction Language

DDL

- CREATE

- DROP

- ALTER

- TRUNCATE

- COMMENT

- RENAME

DML

- INSERT

- DELETE

- UPDATE

- CALL

- EXPLAINPLAN

- LOCK TABLE

DQL

- SELECT

DCL

- GRANT

- REVOQUE

- DENY

DTL

- COMMIT

- SAVEPOINT

- ROLLBACK