

# Dependência Funcional e Normalização

Tiago Eugenio de Melo  
tmelo@uea.edu.br

Escola Superior de Tecnologia  
Universidade do Estado do Amazonas

31 de Julho de 2022

# Dependência Funcional

# Introdução



# Introdução

- As dependências funcionais e as formas normais estabelecem critérios de qualidade de projeto no Modelo Relacional.
  - Eles permitem detectar e prevenir a **redundância** e garantir a **consistência** da informação.

# Introdução

- As dependências funcionais e as formas normais estabelecem critérios de qualidade de projeto no Modelo Relacional.
  - Eles permitem detectar e prevenir a **redundância** e garantir a **consistência** da informação.
  - Fundamentam-se nas dependências entre os atributos das relações.

# Dependência Funcional

# Dependência Funcional

- Uma dependência funcional é um relacionamento muitos para um entre dois conjuntos de atributos de uma determinada relação R.





# Dependência Funcional

# Dependência Funcional

- Sejam os seguintes subconjuntos de atributos de um esquema  
 $T : A = (A_1, A_2, \dots, A_n)$  e  $B = (B_1, B_2, \dots, B_m)$ .

# Dependência Funcional

- Sejam os seguintes subconjuntos de atributos de um esquema  $T : A = (A_1, A_2, \dots, A_n)$  e  $B = (B_1, B_2, \dots, B_n)$ .
- Dizemos que  $B$  é dependente funcionalmente de um outro atributo  $A$  contido em  $T$  se a cada valor de  $A$  existir nas linhas da relação  $T$ , em que aparece, um único valor de  $B$ .

# Dependência Funcional

- Sejam os seguintes subconjuntos de atributos de um esquema  $T : A = (A_1, A_2, \dots, A_n)$  e  $B = (B_1, B_2, \dots, B_m)$ .
- Dizemos que  $B$  é dependente funcionalmente de um outro atributo  $A$  contido em  $T$  se a cada valor de  $A$  existir nas linhas da relação  $T$ , em que aparece, um único valor de  $B$ .
  - Notação:  $A \rightarrow B$

# Dependência Funcional

- Sejam os seguintes subconjuntos de atributos de um esquema  $T : A = (A_1, A_2, \dots, A_n)$  e  $B = (B_1, B_2, \dots, B_n)$ .
- Dizemos que  $B$  é dependente funcionalmente de um outro atributo  $A$  contido em  $T$  se a cada valor de  $A$  existir nas linhas da relação  $T$ , em que aparece, um único valor de  $B$ .
  - Notação:  $A \rightarrow B$
  - Lê-se:  $A$  determina funcionalmente  $B$ .

# Dependência Funcional

- Sejam os seguintes subconjuntos de atributos de um esquema  $T : A = (A_1, A_2, \dots, A_n)$  e  $B = (B_1, B_2, \dots, B_m)$ .
- Dizemos que  $B$  é dependente funcionalmente de um outro atributo  $A$  contido em  $T$  se a cada valor de  $A$  existir nas linhas da relação  $T$ , em que aparece, um único valor de  $B$ .
  - Notação:  $A \rightarrow B$
  - Lê-se:  $A$  determina funcionalmente  $B$ .
- DF é uma propriedade do projeto do BD, isto é, do seu esquema.

# Dependência Funcional

- Sejam os seguintes subconjuntos de atributos de um esquema  $T : A = (A_1, A_2, \dots, A_n)$  e  $B = (B_1, B_2, \dots, B_n)$ .
- Dizemos que  $B$  é dependente funcionalmente de um outro atributo  $A$  contido em  $T$  se a cada valor de  $A$  existir nas linhas da relação  $T$ , em que aparece, um único valor de  $B$ .
  - Notação:  $A \rightarrow B$
  - Lê-se:  $A$  determina funcionalmente  $B$ .
- DF é uma propriedade do projeto do BD, isto é, do seu esquema.
- O conhecimento advém do conhecimento da semântica dos dados armazenados na relação.



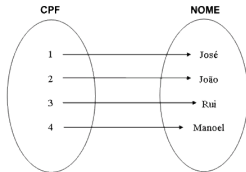
# Dependência Funcional

Observe os conjuntos abaixo:



# Dependência Funcional

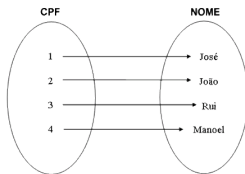
Observe os conjuntos abaixo:



- Existe uma dependência dentre os valores dos conjuntos *CPF* e *NOME*.

# Dependência Funcional

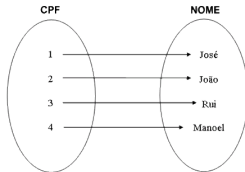
Observe os conjuntos abaixo:



- Existe uma dependência dentre os valores dos conjuntos *CPF* e *NOME*.
- A dependência pode ser expressa pela função  $f(CPF) = nome$ .

# Dependência Funcional

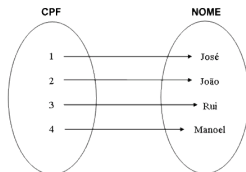
Observe os conjuntos abaixo:



- Existe uma dependência dentre os valores dos conjuntos *CPF* e *NOME*.
- A dependência pode ser expressa pela função  $f(CPF) = nome$ .
- Portanto, *nome* é função do *CPF*.

# Dependência Funcional

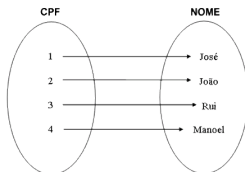
Observe os conjuntos abaixo:



- Existe uma dependência dentre os valores dos conjuntos *CPF* e *NOME*.
- A dependência pode ser expressa pela função  $f(CPF) = nome$ .
- Portanto, *nome* é função do *CPF*.
- Se eu tiver um número de CPF, poderei encontrar o nome correspondente da pessoa.

# Dependência Funcional

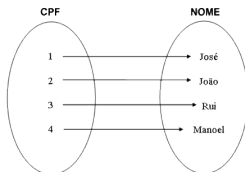
Observe os conjuntos abaixo:



- Existe uma dependência dentre os valores dos conjuntos *CPF* e *NOME*.
- A dependência pode ser expressa pela função  $f(CPF) = nome$ .
- Portanto, *nome* é função do *CPF*.
- Se eu tiver um número de CPF, poderei encontrar o nome correspondente da pessoa.
- Essa dependência é expressa como  $CPF \rightarrow NOME$ .

# Dependência Funcional

Observe os conjuntos abaixo:



- Existe uma dependência dentre os valores dos conjuntos *CPF* e *NOME*.
- A dependência pode ser expressa pela função  $f(CPF) = nome$ .
- Portanto, *nome* é função do *CPF*.
- Se eu tiver um número de CPF, poderei encontrar o nome correspondente da pessoa.
- Essa dependência é expressa como  $CPF \rightarrow NOME$ .
- Portanto, *nome* depende funcionalmente do *CPF*.



# Dependência Funcional

# Dependência Funcional

- Dependências triviais e não triviais

# Dependência Funcional

- Dependências triviais e não triviais
  - A redução do conjunto de dependências funcionais é feito através da eliminação das dependências triviais.

# Dependência Funcional

- Dependências triviais e não triviais
  - A redução do conjunto de dependências funcionais é feito através da eliminação das dependências triviais.
  - Uma dependência é trivial se não puder deixar de ser satisfeita.

# Dependência Funcional

- Dependências triviais e não triviais
  - A redução do conjunto de dependências funcionais é feito através da eliminação das dependências triviais.
  - Uma dependência é trivial se não puder deixar de ser satisfeita.
  - Dependências não triviais são as mais interessantes para o projeto de banco de dados, pois elas são as únicas que correspondem a restrições de integridade genuínas.



## Fecho de um conjunto de DFs

- O fecho (*closure*) de um conjunto de dependências funcionais  $F$  é o conjunto de todas as DFs que podem ser inferidas a partir de  $F$ .

# Fecho de um conjunto de DFs

- O fecho (*closure*) de um conjunto de dependências funcionais  $F$  é o conjunto de todas as DFs que podem ser inferidas a partir de  $F$ .
- Não é preciso listar todas as DFs impostas sobre um esquema.



## Fecho de um conjunto de DFs

- O fecho (*closure*) de um conjunto de dependências funcionais  $F$  é o conjunto de todas as DFs que podem ser inferidas a partir de  $F$ .
- Não é preciso listar todas as DFs impostas sobre um esquema.
- Normalmente especificamos um subconjunto dessas DFs e considerarmos o fecho desse subconjunto.

# Dependência Funcional

Exemplo de tabela de vendas.

F#	CIDADE	P#	QDE
F1	Londres	P1	100
F1	Londres	P2	100
F2	Paris	P1	200
F2	Paris	P2	200
F3	Paris	P2	300
F4	Londres	P2	400
F4	Londres	P4	400
F4	Londres	P5	400

# Dependência Funcional

Exemplo de tabela de vendas.

F#	CIDADE	P#	QDE
F1	Londres	P1	100
F1	Londres	P2	100
F2	Paris	P1	200
F2	Paris	P2	200
F3	Paris	P2	300
F4	Londres	P2	400
F4	Londres	P4	400
F4	Londres	P5	400

- *F#* é chave primária do fornecedor.

# Dependência Funcional

Exemplo de tabela de vendas.

F#	CIDADE	P#	QDE
F1	Londres	P1	100
F1	Londres	P2	100
F2	Paris	P1	200
F2	Paris	P2	200
F3	Paris	P2	300
F4	Londres	P2	400
F4	Londres	P4	400
F4	Londres	P5	400

- *F#* é chave primária do fornecedor.
- *Cidade* é a cidade do fornecedor.

# Dependência Funcional

Exemplo de tabela de vendas.

F#	CIDADE	P#	QDE
F1	Londres	P1	100
F1	Londres	P2	100
F2	Paris	P1	200
F2	Paris	P2	200
F3	Paris	P2	300
F4	Londres	P2	400
F4	Londres	P4	400
F4	Londres	P5	400

- *F#* é chave primária do fornecedor.
- *Cidade* é a cidade do fornecedor.
- *P#* é a chave primária do produto.

# Dependência Funcional

Exemplo de tabela de vendas.

F#	CIDADE	P#	QDE
F1	Londres	P1	100
F1	Londres	P2	100
F2	Paris	P1	200
F2	Paris	P2	200
F3	Paris	P2	300
F4	Londres	P2	400
F4	Londres	P4	400
F4	Londres	P5	400

- *F#* é chave primária do fornecedor.
- *Cidade* é a cidade do fornecedor.
- *P#* é a chave primária do produto.
- *QDE* representa a quantidade de produtos.

# Dependência Funcional

# Dependência Funcional

- Exemplos de dependências funcionais válidas:



# Dependência Funcional

- Exemplos de dependências funcionais válidas:
  - $\{F\# \} \rightarrow \{CIDADE\}$

# Dependência Funcional

- Exemplos de dependências funcionais válidas:
  - $\{F\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{QDE\}$

# Dependência Funcional

- Exemplos de dependências funcionais válidas:
  - $\{F\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{QDE\}$
  - $\{F\#, P\# \} \rightarrow \{CIDADE\}$

# Dependência Funcional

- Exemplos de dependências funcionais válidas:
  - $\{F\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{QDE\}$
  - $\{F\#, P\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{CIDADE, QDE\}$

# Dependência Funcional

- Exemplos de dependências funcionais válidas:
  - $\{F\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{QDE\}$
  - $\{F\#, P\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{CIDADE, QDE\}$
  - $\{F\#, P\# \} \rightarrow \{F\# \}$

# Dependência Funcional

- Exemplos de dependências funcionais válidas:
  - $\{F\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{QDE\}$
  - $\{F\#, P\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{CIDADE, QDE\}$
  - $\{F\#, P\# \} \rightarrow \{F\# \}$
  - $\{F\#, P\# \} \rightarrow \{F\#, P\#, CIDADE, QDE\}$

# Dependência Funcional

- Exemplos de dependências funcionais válidas:
  - $\{F\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{QDE\}$
  - $\{F\#, P\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{CIDADE, QDE\}$
  - $\{F\#, P\# \} \rightarrow \{F\# \}$
  - $\{F\#, P\# \} \rightarrow \{F\#, P\#, CIDADE, QDE\}$
  - $\{F\# \} \rightarrow \{QDE\}$

# Dependência Funcional

- Exemplos de dependências funcionais válidas:
  - $\{F\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{QDE\}$
  - $\{F\#, P\# \} \rightarrow \{CIDADE\}$
  - $\{F\#, P\# \} \rightarrow \{CIDADE, QDE\}$
  - $\{F\#, P\# \} \rightarrow \{F\# \}$
  - $\{F\#, P\# \} \rightarrow \{F\#, P\#, CIDADE, QDE\}$
  - $\{F\# \} \rightarrow \{QDE\}$
  - $\{QDE\} \rightarrow \{F\# \}$



# Propriedades funcionais

# Propriedades funcionais

- Sejam A, B e C subconjuntos arbitrários do conjunto de atributos de uma relação R, e considerando que  $AB$  é usada para indicar a união de A e B, teremos as seguintes propriedades (regras de inferência de Armstrong):

# Propriedades funcionais

- Sejam  $A$ ,  $B$  e  $C$  subconjuntos arbitrários do conjunto de atributos de uma relação  $R$ , e considerando que  $AB$  é usada para indicar a união de  $A$  e  $B$ , teremos as seguintes propriedades (regras de inferência de Armstrong):
  - **Reflexão:** se  $B$  é um subconjunto de  $A$ , então  $A \rightarrow B$ .

# Propriedades funcionais

- Sejam  $A$ ,  $B$  e  $C$  subconjuntos arbitrários do conjunto de atributos de uma relação  $R$ , e considerando que  $AB$  é usada para indicar a união de  $A$  e  $B$ , teremos as seguintes propriedades (regras de inferência de Armstrong):
  - **Reflexão**: se  $B$  é um subconjunto de  $A$ , então  $A \rightarrow B$ .
  - **Aumento**: se  $A \rightarrow B$ , então  $AC \rightarrow BC$ .

# Propriedades funcionais

- Sejam  $A$ ,  $B$  e  $C$  subconjuntos arbitrários do conjunto de atributos de uma relação  $R$ , e considerando que  $AB$  é usada para indicar a união de  $A$  e  $B$ , teremos as seguintes propriedades (regras de inferência de Armstrong):
  - **Reflexão**: se  $B$  é um subconjunto de  $A$ , então  $A \rightarrow B$ .
  - **Aumento**: se  $A \rightarrow B$ , então  $AC \rightarrow BC$ .
  - **Transitividade**: se  $A \rightarrow B$  e  $B \rightarrow C$ , então  $A \rightarrow C$ .

# Propriedades funcionais

- Sejam  $A$ ,  $B$  e  $C$  subconjuntos arbitrários do conjunto de atributos de uma relação  $R$ , e considerando que  $AB$  é usada para indicar a união de  $A$  e  $B$ , teremos as seguintes propriedades (regras de inferência de Armstrong):
  - **Reflexão**: se  $B$  é um subconjunto de  $A$ , então  $A \rightarrow B$ .
  - **Aumento**: se  $A \rightarrow B$ , então  $AC \rightarrow BC$ .
  - **Transitividade**: se  $A \rightarrow B$  e  $B \rightarrow C$ , então  $A \rightarrow C$ .Exemplo:

# Propriedades funcionais

- Sejam  $A$ ,  $B$  e  $C$  subconjuntos arbitrários do conjunto de atributos de uma relação  $R$ , e considerando que  $AB$  é usada para indicar a união de  $A$  e  $B$ , teremos as seguintes propriedades (regras de inferência de Armstrong):
    - **Reflexão**: se  $B$  é um subconjunto de  $A$ , então  $A \rightarrow B$ .
    - **Aumento**: se  $A \rightarrow B$ , então  $AC \rightarrow BC$ .
    - **Transitividade**: se  $A \rightarrow B$  e  $B \rightarrow C$ , então  $A \rightarrow C$ .
- Exemplo:
- $CPF \rightarrow \text{codigo\_cidade}$  e  $\text{codigo\_cidade} \rightarrow \text{nome\_cidade}$ , então  $CPF \rightarrow \text{nome\_cidade}$ .

# Propriedades funcionais

- Sejam  $A$ ,  $B$  e  $C$  subconjuntos arbitrários do conjunto de atributos de uma relação  $R$ , e considerando que  $AB$  é usada para indicar a união de  $A$  e  $B$ , teremos as seguintes propriedades (regras de inferência de Armstrong):
  - **Reflexão**: se  $B$  é um subconjunto de  $A$ , então  $A \rightarrow B$ .
  - **Aumento**: se  $A \rightarrow B$ , então  $AC \rightarrow BC$ .
  - **Transitividade**: se  $A \rightarrow B$  e  $B \rightarrow C$ , então  $A \rightarrow C$ .Exemplo:
  - $CPF \rightarrow \text{codigo\_cidade}$  e  $\text{codigo\_cidade} \rightarrow \text{nome\_cidade}$ , então  $CPF \rightarrow \text{nome\_cidade}$ .
  - **União**: se  $A \rightarrow B$  e  $A \rightarrow C$ , então  $A \rightarrow BC$ .



# Propriedades funcionais

- Sejam  $A$ ,  $B$  e  $C$  subconjuntos arbitrários do conjunto de atributos de uma relação  $R$ , e considerando que  $AB$  é usada para indicar a união de  $A$  e  $B$ , teremos as seguintes propriedades (regras de inferência de Armstrong):
  - **Reflexão**: se  $B$  é um subconjunto de  $A$ , então  $A \rightarrow B$ .
  - **Aumento**: se  $A \rightarrow B$ , então  $AC \rightarrow BC$ .
  - **Transitividade**: se  $A \rightarrow B$  e  $B \rightarrow C$ , então  $A \rightarrow C$ .Exemplo:
  - $CPF \rightarrow \text{codigo\_cidade}$  e  $\text{codigo\_cidade} \rightarrow \text{nome\_cidade}$ , então  $CPF \rightarrow \text{nome\_cidade}$ .
  - **União**: se  $A \rightarrow B$  e  $A \rightarrow C$ , então  $A \rightarrow BC$ .
  - **Decomposição**: se  $A \rightarrow BC$ , então  $A \rightarrow B$  e  $A \rightarrow C$ .

# Propriedades funcionais

- Sejam  $A$ ,  $B$  e  $C$  subconjuntos arbitrários do conjunto de atributos de uma relação  $R$ , e considerando que  $AB$  é usada para indicar a união de  $A$  e  $B$ , teremos as seguintes propriedades (regras de inferência de Armstrong):

- **Reflexão:** se  $B$  é um subconjunto de  $A$ , então  $A \rightarrow B$ .
- **Aumento:** se  $A \rightarrow B$ , então  $AC \rightarrow BC$ .
- **Transitividade:** se  $A \rightarrow B$  e  $B \rightarrow C$ , então  $A \rightarrow C$ .

Exemplo:

- $CPF \rightarrow \text{codigo\_cidade}$  e  $\text{codigo\_cidade} \rightarrow \text{nome\_cidade}$ , então  $CPF \rightarrow \text{nome\_cidade}$ .

- **União:** se  $A \rightarrow B$  e  $A \rightarrow C$ , então  $A \rightarrow BC$ .
- **Decomposição:** se  $A \rightarrow BC$ , então  $A \rightarrow B$  e  $A \rightarrow C$ .

Exemplo:



# Propriedades funcionais

- Sejam  $A$ ,  $B$  e  $C$  subconjuntos arbitrários do conjunto de atributos de uma relação  $R$ , e considerando que  $AB$  é usada para indicar a união de  $A$  e  $B$ , teremos as seguintes propriedades (regras de inferência de Armstrong):

- Reflexão:** se  $B$  é um subconjunto de  $A$ , então  $A \rightarrow B$ .
- Aumento:** se  $A \rightarrow B$ , então  $AC \rightarrow BC$ .
- Transitividade:** se  $A \rightarrow B$  e  $B \rightarrow C$ , então  $A \rightarrow C$ .

Exemplo:

- $CPF \rightarrow \text{codigo\_cidade}$  e  $\text{codigo\_cidade} \rightarrow \text{nome\_cidade}$ , então  $CPF \rightarrow \text{nome\_cidade}$ .

- União:** se  $A \rightarrow B$  e  $A \rightarrow C$ , então  $A \rightarrow BC$ .
- Decomposição:** se  $A \rightarrow BC$ , então  $A \rightarrow B$  e  $A \rightarrow C$ .

Exemplo:

- $CPF \rightarrow \text{nome, endereco}$ , então  $CPF \rightarrow \text{nome}$  e  $CPF \rightarrow \text{endereco}$ .

- Autoderminação:**  $A \rightarrow A$ .

# Propriedades funcionais

- Sejam  $A$ ,  $B$  e  $C$  subconjuntos arbitrários do conjunto de atributos de uma relação  $R$ , e considerando que  $AB$  é usada para indicar a união de  $A$  e  $B$ , teremos as seguintes propriedades (regras de inferência de Armstrong):

- Reflexão:** se  $B$  é um subconjunto de  $A$ , então  $A \rightarrow B$ .
- Aumento:** se  $A \rightarrow B$ , então  $AC \rightarrow BC$ .
- Transitividade:** se  $A \rightarrow B$  e  $B \rightarrow C$ , então  $A \rightarrow C$ .

Exemplo:

- $CPF \rightarrow \text{codigo\_cidade}$  e  $\text{codigo\_cidade} \rightarrow \text{nome\_cidade}$ ,  
então  $CPF \rightarrow \text{nome\_cidade}$ .

- União:** se  $A \rightarrow B$  e  $A \rightarrow C$ , então  $A \rightarrow BC$ .
- Decomposição:** se  $A \rightarrow BC$ , então  $A \rightarrow B$  e  $A \rightarrow C$ .

Exemplo:

- $CPF \rightarrow \text{nome, endereco}$ , então  $CPF \rightarrow \text{nome}$  e  
 $CPF \rightarrow \text{endereco}$ .

- Autoderminação:**  $A \rightarrow A$ .
- Composição:** se  $A \rightarrow B$  e  $C \rightarrow D$ , então  $AC \rightarrow BD$ .

# Propriedades funcionais

**União:** se  $A \rightarrow B$  e  $A \rightarrow C$ , então  $A \rightarrow BC$ .

# Propriedades funcionais

**União:** se  $A \rightarrow B$  e  $A \rightarrow C$ , então  $A \rightarrow BC$ .

Prova:

# Propriedades funcionais

**União:** se  $A \rightarrow B$  e  $A \rightarrow C$ , então  $A \rightarrow BC$ .

Prova:

- $A \rightarrow AB$  (regra aumento).



# Propriedades funcionais

**União:** se  $A \rightarrow B$  e  $A \rightarrow C$ , então  $A \rightarrow BC$ .

Prova:

- $A \rightarrow AB$  (regra aumento).
- $A \rightarrow C, AB \rightarrow BC$  (regra aumento).

# Propriedades funcionais

**União:** se  $A \rightarrow B$  e  $A \rightarrow C$ , então  $A \rightarrow BC$ .

Prova:

- $A \rightarrow AB$  (regra aumento).
- $A \rightarrow C, AB \rightarrow BC$  (regra aumento).
- $A \rightarrow AB$  e  $AB \rightarrow BC$ .

# Propriedades funcionais

**União:** se  $A \rightarrow B$  e  $A \rightarrow C$ , então  $A \rightarrow BC$ .

Prova:

- $A \rightarrow AB$  (regra aumento).
- $A \rightarrow C, AB \rightarrow BC$  (regra aumento).
- $A \rightarrow AB$  e  $AB \rightarrow BC$ .
- Logo,  $A \rightarrow BC$  (regra transitividade).

# Propriedades funcionais

**Decomposição:** se  $A \rightarrow BC$ , então  $A \rightarrow B$  e  $A \rightarrow C$ .

# Propriedades funcionais

**Decomposição:** se  $A \rightarrow BC$ , então  $A \rightarrow B$  e  $A \rightarrow C$ .

Prova:

# Propriedades funcionais

**Decomposição:** se  $A \rightarrow BC$ , então  $A \rightarrow B$  e  $A \rightarrow C$ .

Prova:

- $A \rightarrow AB$  e  $AB \rightarrow BC$  (regra aumento).

# Propriedades funcionais

**Decomposição:** se  $A \rightarrow BC$ , então  $A \rightarrow B$  e  $A \rightarrow C$ .

Prova:

- $A \rightarrow AB$  e  $AB \rightarrow BC$  (regra aumento).
- Logo,  $A \rightarrow BC$  (regra transitividade).

# Propriedades funcionais

**Composição:** se  $A \rightarrow B$  e  $C \rightarrow D$ , então  $AC \rightarrow BD$ .



# Propriedades funcionais

**Composição:** se  $A \rightarrow B$  e  $C \rightarrow D$ , então  $AC \rightarrow BD$ .

Prova: exercício!

# Exemplo de uso das propriedades funcionais

# Exemplo de uso das propriedades funcionais

- Considere a relação  $R$  com os atributos  $A, B, C, D, E, F$  e as DFs:

# Exemplo de uso das propriedades funcionais

- Considere a relação  $R$  com os atributos  $A, B, C, D, E, F$  e as DFs:
  - $A \rightarrow BC$

# Exemplo de uso das propriedades funcionais

- Considere a relação  $R$  com os atributos  $A, B, C, D, E, F$  e as DFs:
  - $A \rightarrow BC$
  - $B \rightarrow E$

# Exemplo de uso das propriedades funcionais

- Considere a relação  $R$  com os atributos  $A, B, C, D, E, F$  e as DFs:
  - $A \rightarrow BC$
  - $B \rightarrow E$
  - $CD \rightarrow EF$

## Exemplo de uso das propriedades funcionais

- Considere a relação  $R$  com os atributos  $A, B, C, D, E, F$  e as DFs:
  - $A \rightarrow BC$
  - $B \rightarrow E$
  - $CD \rightarrow EF$

Mostre que a dependência funcional  $AD \rightarrow F$  é válida para  $R$  e, portanto, é um membro do fecho do conjunto dado.

# Exemplo de uso das propriedades funcionais (resposta)

Solução:



# Exemplo de uso das propriedades funcionais (resposta)

Solução:

①  $A \rightarrow BC$  (dada)

# Exemplo de uso das propriedades funcionais (resposta)

Solução:

- 1  $A \rightarrow BC$  (dada)
- 2  $A \rightarrow C$  (1, decomposição)

# Exemplo de uso das propriedades funcionais (resposta)

Solução:

- 1  $A \rightarrow BC$  (dada)
- 2  $A \rightarrow C$  (1, decomposição)
- 3  $AD \rightarrow CD$  (2, aumento)

# Exemplo de uso das propriedades funcionais (resposta)

Solução:

- 1  $A \rightarrow BC$  (dada)
- 2  $A \rightarrow C$  (1, decomposição)
- 3  $AD \rightarrow CD$  (2, aumento)
- 4  $CD \rightarrow EF$  (dada)

# Exemplo de uso das propriedades funcionais (resposta)

Solução:

- 1  $A \rightarrow BC$  (dada)
- 2  $A \rightarrow C$  (1, decomposição)
- 3  $AD \rightarrow CD$  (2, aumento)
- 4  $CD \rightarrow EF$  (dada)
- 5  $AD \rightarrow EF$  (3 e 4, transitividade)

# Exemplo de uso das propriedades funcionais (resposta)

Solução:

- 1  $A \rightarrow BC$  (dada)
- 2  $A \rightarrow C$  (1, decomposição)
- 3  $AD \rightarrow CD$  (2, aumento)
- 4  $CD \rightarrow EF$  (dada)
- 5  $AD \rightarrow EF$  (3 e 4, transitividade)
- 6  $AD \rightarrow F$  (5, decomposição)

# Normalização

# Normalização



# Normalização

- O processo de Normalização, proposto primeiramente por Codd, faz uma série de testes para certificar se um Esquema Relacional satisfaz a uma Forma Normal (FN).

# Normalização

- O processo de Normalização, proposto primeiramente por Codd, faz uma série de testes para certificar se um Esquema Relacional satisfaz a uma Forma Normal (FN).
- Cada relação é avaliada e decomposta em novas relações, se necessário.

# Normalização

- O processo de Normalização, proposto primeiramente por Codd, faz uma série de testes para certificar se um Esquema Relacional satisfaz a uma Forma Normal (FN).
- Cada relação é avaliada e decomposta em novas relações, se necessário.
- Projeto relacional por análise.

# Normalização

- O processo de Normalização, proposto primeiramente por Codd, faz uma série de testes para certificar se um Esquema Relacional satisfaz a uma Forma Normal (FN).
- Cada relação é avaliada e decomposta em novas relações, se necessário.
- Projeto relacional por análise.
- Inicialmente, Codd propôs três formas normais.

# Normalização

# Normalização

Conseqüências:

# Normalização

Conseqüências:

- Problemas de anomalias e inconsistências diminuem.

# Normalização

Conseqüências:

- Problemas de anomalias e inconsistências diminuem.
- Relações simplificadas e estrutura regular.



# Normalização

Conseqüências:

- Problemas de anomalias e inconsistências diminuem.
- Relações simplificadas e estrutura regular.
- Aumento da integridade dos dados.

# Normalização

## Conseqüências:

- Problemas de anomalias e inconsistências diminuem.
- Relações simplificadas e estrutura regular.
- Aumento da integridade dos dados.
- Necessidade de realização de junções.

# Normalização

## Conseqüências:

- Problemas de anomalias e inconsistências diminuem.
- Relações simplificadas e estrutura regular.
- Aumento da integridade dos dados.
- Necessidade de realização de junções.
- Eventual queda na performance.

# Normalização - 1ª Forma Normal (1FN)

# Normalização - 1ª Forma Normal (1FN)

- Uma relação R está na 1FN se, e somente se, em todo valor válido dessa relação, cada tupla contém exatamente um valor para cada atributo.

## Normalização - 1ª Forma Normal (1FN)

- Uma relação R está na 1FN se, e somente se, em todo valor válido dessa relação, cada tupla contém exatamente um valor para cada atributo.
- Todo e qualquer atributo deve ter valor **ATÔMICO** e **INDIVISÍVEL**, ou seja, no modelo relacional não pode haver atributos multivalorados ou conjuntos de atributos.

# Normalização - 1ª Forma Normal (1FN)

- Uma relação R está na 1FN se, e somente se, em todo valor válido dessa relação, cada tupla contém exatamente um valor para cada atributo.
- Todo e qualquer atributo deve ter valor **ATÔMICO** e **INDIVISÍVEL**, ou seja, no modelo relacional não pode haver atributos multivalorados ou conjuntos de atributos.
- Esta FN é considerada parte da definição do Modelo Relacional.

# Normalização - 1ª Forma Normal (1FN)



# Normalização - 1ª Forma Normal (1FN)

Resumo:

# Normalização - 1ª Forma Normal (1FN)

Resumo:

- as linhas da tabela são unívocas.



# Normalização - 1ª Forma Normal (1FN)

Resumo:

- as linhas da tabela são unívocas.
- as linhas não contêm itens repetitivos.
- os atributos são atômicos.

# Normalização - 1ª Forma Normal (1FN) - Exemplo

# Normalização - 1ª Forma Normal (1FN) - Exemplo

Tabela: PEDIDO (não normalizada)

NumPed	DataEmis	Fornecedor	CGC	End	CodProd	NomeProd	Qtde	Preço
3	20/jan	Casa Software	1111111-11	Lapa Rua A	111	Prod 1	10	R\$100,00
					222	Prod 2	44	R\$150,00
					333	Prod 3	50	R\$120,00
4	10/fev	Computer	2222222-22	Itu 49	222	Prod 4	73	R\$150,00
					333	Prod 5	80	R\$120,00



# Normalização - 1ª Forma Normal (1FN) - Exemplo



# Normalização - 1ª Forma Normal (1FN) - Exemplo

Tabela: PEDIDO - sem itens repetidos

NumPed	DataEmis	Fornecedor	CGC	End	Cod Prod	NomeProd	QtdeProd	PreçoProd
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	111	Prod 1	10	R\$100.00
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	222	Prod 2	44	R\$150.00
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	333	Prod 3	50	R\$120.00
4	Feb 10	Computer	2222222-22	Itu 49	222	Prod 4	73	R\$150.00
4	Feb 10	Computer	2222222-22	Itu 49	333	Prod 5	80	R\$120.00

# Normalização - 1ª Forma Normal (1FN) - Exemplo

Tabela: PEDIDO - sem itens repetidos

NumPed	DataEmis	Fornecedor	CGC	End	Cod Prod	NomeProd	QtdeProd	PreçoProd
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	111	Prod 1	10	R\$100.00
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	222	Prod 2	44	R\$150.00
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	333	Prod 3	50	R\$120.00
4	Feb 10	Computer	2222222-22	Itu 49	222	Prod 4	73	R\$150.00
4	Feb 10	Computer	2222222-22	Itu 49	333	Prod 5	80	R\$120.00

Quais são os problemas que ainda persistem?

# Normalização - 1ª Forma Normal (1FN)

# Normalização - 1ª Forma Normal (1FN)

Processo para obtenção da 1FN:

# Normalização - 1ª Forma Normal (1FN)

Processo para obtenção da 1FN:

- Em cada tabela eliminar grupos repetitivos gerando novas linhas, uma para cada ocorrência de item repetitivo, mantendo os valores dos demais itens.

# Normalização - 1ª Forma Normal (1FN)

Processo para obtenção da 1FN:

- Em cada tabela eliminar grupos repetitivos gerando novas linhas, uma para cada ocorrência de item repetitivo, mantendo os valores dos demais itens.
- Transformar os atributos compostos em atômicos.

# Normalização - 1ª Forma Normal (1FN) - Exemplo

# Normalização - 1ª Forma Normal (1FN) - Exemplo

Tabela: PEDIDO - sem itens repetidos (antes)

NumPed	DataEmis	Fornecedor	CGC	End	Cod Prod	NomeProd	QtdeProd	PreçoProd
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	111	Prod 1	10	R\$100.00
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	222	Prod 2	44	R\$150.00
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	333	Prod 3	50	R\$120.00
4	Feb 10	Computer	2222222-22	Itu 49	222	Prod 4	73	R\$150.00
4	Feb 10	Computer	2222222-22	Itu 49	333	Prod 5	80	R\$120.00



# Normalização - 1ª Forma Normal (1FN) - Exemplo

Tabela: PEDIDO - sem itens repetidos (antes)

NumPed	DataEmis	Fornecedor	CGC	End	Cod Prod	NomeProd	QtdeProd	PreçoProd
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	111	Prod 1	10	R\$100.00
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	222	Prod 2	44	R\$150.00
3	Jan 20	Casa Software	1111111-11	Lapa Rua A	333	Prod 3	50	R\$120.00
4	Feb 10	Computer	2222222-22	Itu 49	222	Prod 4	73	R\$150.00
4	Feb 10	Computer	2222222-22	Itu 49	333	Prod 5	80	R\$120.00

Tabela: PEDIDO - com atributos atômicos (depois)

NumPed	DataEmis	Fornecedor	CGC	Bairro	Rua	Cod Prod	NomeProd	QtdeProd
3	Jan 20	Casa Software	1111111-11	Lapa	Rua A	111	Prod 1	10
3	Jan 20	Casa Software	1111111-11	Lapa	Rua A	222	Prod 2	44
3	Jan 20	Casa Software	1111111-11	Lapa	Rua A	333	Prod 3	50
4	Feb 10	Computer	2222222-22	Itu	Rua 49	222	Prod 2	73
4	Feb 10	Computer	2222222-22	Itu	Rua 49	333	Prod 3	80

# Normalização - 1ª Forma Normal (1FN)

# Normalização - 1ª Forma Normal (1FN)

Processo para obtenção da 1FN:

# Normalização - 1ª Forma Normal (1FN)

Processo para obtenção da 1FN:

- Em cada tabela eliminar grupos repetitivos gerando novas linhas, uma para cada ocorrência de item repetitivo, mantendo os valores dos demais itens.

# Normalização - 1ª Forma Normal (1FN)

Processo para obtenção da 1FN:

- Em cada tabela eliminar grupos repetitivos gerando novas linhas, uma para cada ocorrência de item repetitivo, mantendo os valores dos demais itens.
- Transformar os atributos compostos em atômicos.

# Normalização - 1ª Forma Normal (1FN)

Processo para obtenção da 1FN:

- Em cada tabela eliminar grupos repetitivos gerando novas linhas, uma para cada ocorrência de item repetitivo, mantendo os valores dos demais itens.
- Transformar os atributos compostos em atômicos.
- Definir as chaves candidatas e escolher a chave primária da tabela (unicidade nas linhas).

# Normalização - 1ª Forma Normal (1FN) - Exemplo

# Normalização - 1ª Forma Normal (1FN) - Exemplo

Tabela: PEDIDO - com chave primária NumPed + CodProd

NumPed	Cod Prod	DataEmis	Fornecedor	CGC	Bairro	Rua	NomeProd	QtdeProd
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 1	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 2	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 3	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 2	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 3	80



# Normalização - 1ª Forma Normal (1FN)

Deve-se observar que se uma relação R estiver apenas na 1FN (ou seja, não esteja na 2FN, e portanto também não está na 3FN) tem uma estrutura indesejável por uma série de razões.

## Normalização - 2ª Forma Normal (2FN)

Uma relação R está na 2FN se e somente se ela está em 1FN e todo atributo não chave é irreduzivelmente dependente da chave primária.

# Normalização - 2ª Forma Normal (2FN)

# Normalização - 2ª Forma Normal (2FN)

Segunda Forma Normal (2FN):

# Normalização - 2ª Forma Normal (2FN)

Segunda Forma Normal (2FN):

- Está na 1FN.

## Normalização - 2ª Forma Normal (2FN)

Segunda Forma Normal (2FN):

- Está na 1FN.
- Cada uma das colunas não pertencentes à chave primária não é dependente parcial dessa chave.

## Normalização - 2ª Forma Normal (2FN)

Segunda Forma Normal (2FN):

- Está na 1FN.
- Cada uma das colunas não pertencentes à chave primária não é dependente parcial dessa chave.
- Cada atributo não-chave é dependente de toda a chave primária.

## Normalização - 2ª Forma Normal (2FN)

Segunda Forma Normal (2FN):

- Está na 1FN.
- Cada uma das colunas não pertencentes à chave primária não é dependente parcial dessa chave.
- Cada atributo não-chave é dependente de toda a chave primária.
- A dependência parcial de uma chave só será possível se esta chave for definida com mais de uma coluna.



## Normalização - 2ª Forma Normal (2FN)

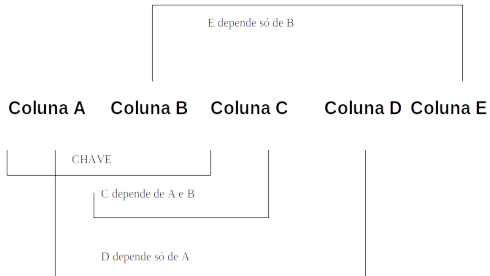
Segunda Forma Normal (2FN):

- Está na 1FN.
- Cada uma das colunas não pertencentes à chave primária não é dependente parcial dessa chave.
- Cada atributo não-chave é dependente de toda a chave primária.
- A dependência parcial de uma chave só será possível se esta chave for definida com mais de uma coluna.
- Dizemos que uma coluna é parcialmente dependente da chave se para que seu valor seja determinado não necessitamos conhecer a chave como um todo.

# Normalização - 2ª Forma Normal (2FN)

# Normalização - 2ª Forma Normal (2FN)

Tabela não normalizada:



# Normalização - 2ª Forma Normal (2FN)

Tabela não normalizada:

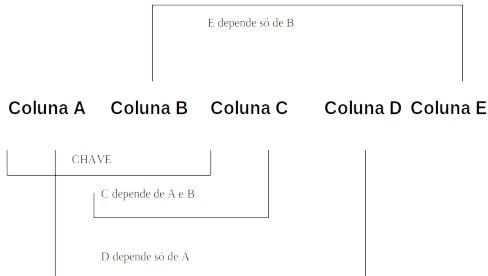


Tabela normalizada:

# Normalização - 2ª Forma Normal (2FN)

Tabela não normalizada:

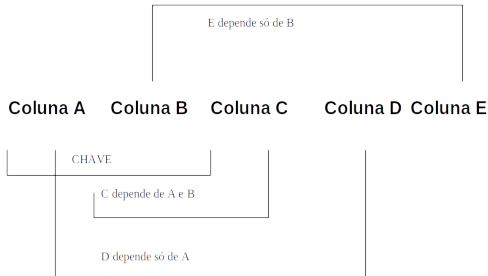


Tabela normalizada:

- $T_1 = \{\underline{ColunaA}, \underline{ColunaB}, ColunaC\}$

# Normalização - 2ª Forma Normal (2FN)

Tabela não normalizada:

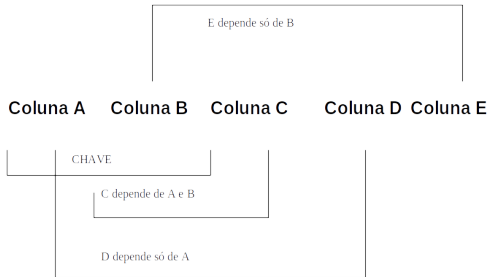


Tabela normalizada:

- $T_1 = \{\underline{ColunaA}, \underline{ColunaB}, ColunaC\}$
- $T_2 = \{\underline{ColunaA}, ColunaD\}$

# Normalização - 2ª Forma Normal (2FN)

Tabela não normalizada:

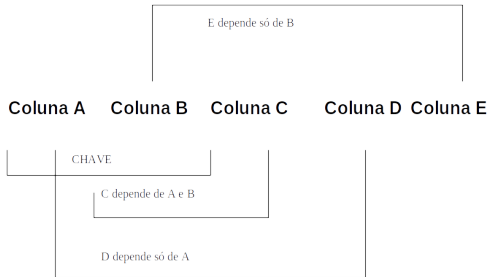


Tabela normalizada:

- $T_1 = \{\underline{ColunaA}, \underline{ColunaB}, ColunaC\}$
- $T_2 = \{\underline{ColunaA}, ColunaD\}$
- $T_3 = \{\underline{ColunaB}, ColunaE\}$

# Normalização - 2ª Forma Normal (2FN)



# Normalização - 2ª Forma Normal (2FN)

Processo para obtenção da 2FN:

## Normalização - 2ª Forma Normal (2FN)

Processo para obtenção da 2FN:

- Identificar as colunas que não participam da chave primária da tabela.

# Normalização - 2ª Forma Normal (2FN)

Processo para obtenção da 2FN:

- Identificar as colunas que não participam da chave primária da tabela.
- Para cada uma das colunas identificadas, analisar se seu valor é determinado por parte, ou pela totalidade da chave.

## Normalização - 2ª Forma Normal (2FN)

Processo para obtenção da 2FN:

- Identificar as colunas que não participam da chave primária da tabela.
- Para cada uma das colunas identificadas, analisar se seu valor é determinado por parte, ou pela totalidade da chave.
- Para as colunas dependentes parcialmente:

## Normalização - 2ª Forma Normal (2FN)

Processo para obtenção da 2FN:

- Identificar as colunas que não participam da chave primária da tabela.
- Para cada uma das colunas identificadas, analisar se seu valor é determinado por parte, ou pela totalidade da chave.
- Para as colunas dependentes parcialmente:
  - Criar novas tabelas onde a chave primária será(ão) a(s) coluna(s) da chave primária original que determinou o valor da coluna analisada.

## Normalização - 2ª Forma Normal (2FN)

Processo para obtenção da 2FN:

- Identificar as colunas que não participam da chave primária da tabela.
- Para cada uma das colunas identificadas, analisar se seu valor é determinado por parte, ou pela totalidade da chave.
- Para as colunas dependentes parcialmente:
  - Criar novas tabelas onde a chave primária será(ão) a(s) coluna(s) da chave primária original que determinou o valor da coluna analisada.
  - Excluir da tabela original as colunas dependentes parcialmente da chave.

# Normalização - 2ª Forma Normal (2FN) - Exemplo

# Normalização - 2ª Forma Normal (2FN) - Exemplo

Tabela na 1FN: pedidos

NumPed	Cod Prod	DataEmis	Fornecedor	CGC	Bairro	Rua	NomeProd	QtdeProd
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 1	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 2	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 3	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 2	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 3	80



# Normalização - 2ª Forma Normal (2FN) - Exemplo

Tabela na 1FN: pedidos

NumPed	Cod Prod	DataEmis	Fornecedor	CGC	Bairro	Rua	NomeProd	QtdeProd
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 1	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 2	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 3	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 2	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 3	80

Tabela na 2FN: Pedido

NumPed	Cod Prod	DataEmis	Fornecedor	CGC	Bairro	Rua	QtdeProd
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	80

# Normalização - 2ª Forma Normal (2FN) - Exemplo

Tabela na 1FN: pedidos

NumPed	Cod Prod	DataEmis	Fornecedor	CGC	Bairro	Rua	NomeProd	QtdeProd
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 1	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 2	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	Prod 3	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 2	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	Prod 3	80

Tabela na 2FN: Pedido

NumPed	Cod Prod	DataEmis	Fornecedor	CGC	Bairro	Rua	QtdeProd
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	80

Tabela na 2FN: Produto

Cod Prod	NomeProd	PreçoProd
111	Prod 1	R\$100,00
222	Prod 2	R\$150,00
333	Prod 3	R\$120,00

## Normalização - 3ª Forma Normal (3FN)

Uma relação R está na 3FN se e somente se ela está na 2FN e todo atributo não chave é dependente de forma não transitiva da chave primária.

# Normalização - 3ª Forma Normal (3FN)

# Normalização - 3ª Forma Normal (3FN)

Terceira Forma Normal (3FN):

# Normalização - 3ª Forma Normal (3FN)

Terceira Forma Normal (3FN):

- Está na 2FN.

# Normalização - 3ª Forma Normal (3FN)

Terceira Forma Normal (3FN):

- Está na 2FN.
- Nenhuma coluna não pertencente à chave fica determinada transitivamente por esta.

## Normalização - 3ª Forma Normal (3FN)

Terceira Forma Normal (3FN):

- Está na 2FN.
- Nenhuma coluna não pertencente à chave fica determinada transitivamente por esta.
- Cada atributo não chave é dependente não transitivo da chave primária.



## Normalização - 3ª Forma Normal (3FN)

Terceira Forma Normal (3FN):

- Está na 2FN.
- Nenhuma coluna não pertencente à chave fica determinada transitivamente por esta.
- Cada atributo não chave é dependente não transitivo da chave primária.
- A dependência transitiva de uma chave só será possível se a tabela tiver pelo menos duas colunas não pertencentes à chave.

## Normalização - 3ª Forma Normal (3FN)

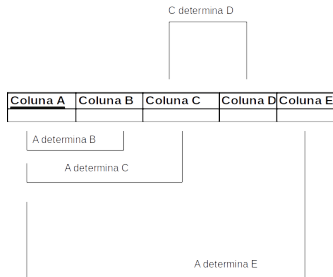
Terceira Forma Normal (3FN):

- Está na 2FN.
- Nenhuma coluna não pertencente à chave fica determinada transitivamente por esta.
- Cada atributo não chave é dependente não transitivo da chave primária.
- A dependência transitiva de uma chave só será possível se a tabela tiver pelo menos duas colunas não pertencentes à chave.
- Uma coluna depende transitivamente da chave se seu valor é determinado pelo conteúdo de uma coluna não chave.

# Normalização - 3ª Forma Normal (3FN)

# Normalização - 3ª Forma Normal (3FN)

Tabela não normalizada:



# Normalização - 3ª Forma Normal (3FN)

Tabela não normalizada:

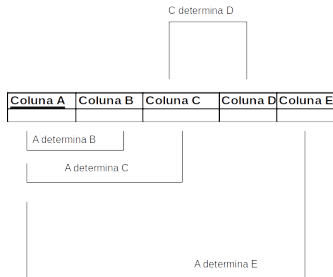


Tabela normalizada:

# Normalização - 3ª Forma Normal (3FN)

Tabela não normalizada:

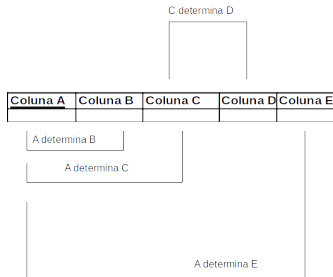


Tabela normalizada:

- $T_1 = \{\underline{ColunaA}, ColunaB, ColunaC, ColunaE\}$

# Normalização - 3ª Forma Normal (3FN)

Tabela não normalizada:

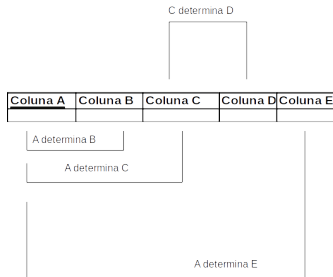


Tabela normalizada:

- $T_1 = \{\underline{ColunaA}, ColunaB, ColunaC, ColunaE\}$
- $T_2 = \{\underline{ColunaC}, ColunaD\}$

# Normalização - 3ª Forma Normal (3FN) - Exemplo



# Normalização - 3ª Forma Normal (3FN) - Exemplo

Tabela na 2FN: pedidos

NumPed	Cod Prod	DataEmis	Fornecedor	CGC	Bairro	Rua	QtdeProd
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	80

# Normalização - 3ª Forma Normal (3FN) - Exemplo

Tabela na 2FN: pedidos

NumPed	Cod Prod	DataEmis	Fornecedor	CGC	Bairro	Rua	QtdeProd
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	80

Tabela na 3FN: Pedido

NumPed	Cod Prod	DataEmis	CGC	QtdeProd
3	111	20/Jan	1111111-11	10
3	222	20/Jan	1111111-11	44
3	333	20/Jan	1111111-11	50
4	222	10/Fev	2222222-22	73
4	333	10/Fev	2222222-22	80

# Normalização - 3ª Forma Normal (3FN) - Exemplo

Tabela na 2FN: pedidos

NumPed	Cod Prod	DataEmis	Fornecedor	CGC	Bairro	Rua	QtdeProd
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	80

Tabela na 3FN: Pedido

NumPed	Cod Prod	DataEmis	CGC	QtdeProd
3	111	20/Jan	1111111-11	10
3	222	20/Jan	1111111-11	44
3	333	20/Jan	1111111-11	50
4	222	10/Fev	2222222-22	73
4	333	10/Fev	2222222-22	80

Tabela na 3FN: Produto

Cod Prod	NomeProd	PreçoProd
111	Prod 1	R\$100,00
222	Prod 2	R\$150,00
333	Prod 3	R\$120,00

# Normalização - 3ª Forma Normal (3FN) - Exemplo

Tabela na 2FN: pedidos

NumPed	Cod Prod	DataEmis	Fornecedor	CGC	Bairro	Rua	QtdeProd
3	111	Jan 20	Casa Software	1111111-11	Lapa	Rua A	10
3	222	Jan 20	Casa Software	1111111-11	Lapa	Rua A	44
3	333	Jan 20	Casa Software	1111111-11	Lapa	Rua A	50
4	222	Feb 10	Computer	2222222-22	Itu	Rua 49	73
4	333	Feb 10	Computer	2222222-22	Itu	Rua 49	80

Tabela na 3FN: Pedido

NumPed	Cod Prod	DataEmis	CGC	QtdeProd
3	111	20/Jan	1111111-11	10
3	222	20/Jan	1111111-11	44
3	333	20/Jan	1111111-11	50
4	222	10/Fev	2222222-22	73
4	333	10/Fev	2222222-22	80

Tabela na 3FN: Produto

Cod Prod	NomeProd	PreçoProd
111	Prod 1	R\$100,00
222	Prod 2	R\$150,00
333	Prod 3	R\$120,00

Tabela na 3FN: Fornecedor

CGC	Fornecedor	Bairro	Rua
1111111-11	Casa Software	Lapa	Rua A
2222222-22	Computer	Itu	Rua 49

# Normalização - FN de Boyce Codd (FNBC)

# Normalização - FN de Boyce Codd (FNBC)

- Existe uma restrição da 3FN que foi denominada de FNBC.

# Normalização - FN de Boyce Codd (FNBC)

- Existe uma restrição da 3FN que foi denominada de FNBC.
- Não é a 4FN, pois foi descoberta posteriormente.

## Normalização - FN de Boyce Codd (FNBC)

- Existe uma restrição da 3FN que foi denominada de FNBC.
- Não é a 4FN, pois foi descoberta posteriormente.
- Definição: Uma relação R está em FNBC se, para toda DF não trivial,  $X \rightarrow Y$ , válida para R, então X é uma superchave de R.



## Normalização - FN de Boyce Codd (FNBC)

- Existe uma restrição da 3FN que foi denominada de FNBC.
- Não é a 4FN, pois foi descoberta posteriormente.
- Definição: Uma relação R está em FNBC se, para toda DF não trivial,  $X \rightarrow Y$ , válida para R, então X é uma superchave de R.
- Portanto, as únicas dependências funcionais não triviais são derivadas de chaves da relação.

# Normalização - FN de Boyce Codd (FNBC)

# Normalização - FN de Boyce Codd (FNBC)

- Excepcionalmente, não se pode ter todas as relações de um BD na FNBC.

# Normalização - FN de Boyce Codd (FNBC)

- Excepcionalmente, não se pode ter todas as relações de um BD na FNBC.
- As exceções, porém, são muito raras.

# Normalização - FN de Boyce Codd (FNBC)

- Excepcionalmente, não se pode ter todas as relações de um BD na FNBC.
- As exceções, porém, são muito raras.
- Exemplo:

# Normalização - FN de Boyce Codd (FNBC)

- Excepcionalmente, não se pode ter todas as relações de um BD na FNBC.
- As exceções, porém, são muito raras.
- Exemplo:
  - Considere a seguinte relação:

## Normalização - FN de Boyce Codd (FNBC)

- Excepcionalmente, não se pode ter todas as relações de um BD na FNBC.
- As exceções, porém, são muito raras.
- Exemplo:
  - Considere a seguinte relação:
  - Endereços (Cidade, Rua, CEP), com DFs,  $CidadeRua \rightarrow CEP$  e  $CEP \rightarrow Cidade$ .

# Normalização - FN de Boyce Codd (FNBC)

- Excepcionalmente, não se pode ter todas as relações de um BD na FNBC.
- As exceções, porém, são muito raras.
- Exemplo:
  - Considere a seguinte relação:
  - Endereços (Cidade, Rua, CEP), com DFs,  $CidadeRua \rightarrow CEP$  e  $CEP \rightarrow Cidade$ .
  - O par (Cidade, Rua) determina o CEP, mas nenhum deles isoladamente determina o CEP.



## Normalização - FN de Boyce Codd (FNBC)

- Excepcionalmente, não se pode ter todas as relações de um BD na FNBC.
- As exceções, porém, são muito raras.
- Exemplo:
  - Considere a seguinte relação:
  - Endereços (Cidade, Rua, CEP), com DFs,  $CidadeRua \rightarrow CEP$  e  $CEP \rightarrow Cidade$ .
  - O par (Cidade, Rua) determina o CEP, mas nenhum deles isoladamente determina o CEP.
  - Dado um CEP, a Cidade fica determinada (mas não vice-versa, nem CEP determina Rua).

## Normalização - FN de Boyce Codd (FNBC)

- Excepcionalmente, não se pode ter todas as relações de um BD na FNBC.
- As exceções, porém, são muito raras.
- Exemplo:
  - Considere a seguinte relação:
  - Endereços (Cidade, Rua, CEP), com DFs,  $CidadeRua \rightarrow CEP$  e  $CEP \rightarrow Cidade$ .
  - O par (Cidade, Rua) determina o CEP, mas nenhum deles isoladamente determina o CEP.
  - Dado um CEP, a Cidade fica determinada (mas não vice-versa, nem CEP determina Rua).
  - (Cidade Rua) é chave da relação Endereços, mas ela não está na FNBC, pois CEP não é chave ou superchave.

# Normalização – 1FN (anomalias)

# Normalização – 1FN (anomalias)

Considere a tabela Empregados, sendo chave primária os campos Matrícula e CodProj.

Matrícula	Nome	CodCargo	NomeCargo	CodProj	DataFim	Horas
120	João	1	Programador	01	17/07/95	37
120	João	1	Programador	08	12/01/96	12
121	Hélio	1	Programador	01	17/07/95	45
121	Hélio	1	Programador	12	21/03/96	107
270	Gabriel	2	Analista	08	12/01/96	10
270	Gabriel	2	Analista	12	21/03/96	38
273	Silva	3	Projetista	01	17/07/95	22
274	Abraão	2	Analista	12	21/03/96	31

# Normalização – 1FN (anomalias)

## Normalização – 1FN (anomalias)

- Inserir: não é possível inserir um empregado sem que este esteja alocado num projeto, nem inserir um projeto sem que haja um empregado trabalhando nele.

## Normalização – 1FN (anomalias)

- Inserir: não é possível inserir um empregado sem que este esteja alocado num projeto, nem inserir um projeto sem que haja um empregado trabalhando nele.
- Remover: se for necessário remover um projeto, as informações de empregado que estiverem alocados naquele projeto serão perdidas.

## Normalização – 1FN (anomalias)

- Inserir: não é possível inserir um empregado sem que este esteja alocado num projeto, nem inserir um projeto sem que haja um empregado trabalhando nele.
- Remover: se for necessário remover um projeto, as informações de empregado que estiverem alocados naquele projeto serão perdidas.
- Atualizar: se um empregado for promovido de cargo, teremos que atualizar os atributos CodCargo e NomeCargo em todas as tuplas nas quais aquele empregado está presente.



# Normalização – 2FN (anomalias)

## EMPREGADO

<u>Matrícula</u>	Nome	CodCargo	NomeCargo
120	João	1	Programador
121	Hélio	1	Programador
270	Gabriel	2	Analista
273	Silva	3	Projetista
274	Abraão	2	Analista

## ALOCAÇÃO

<u>Matrícula</u>	<u>CodProj</u>	<u>Horas</u>
120	01	37
120	08	12
121	01	45
121	08	21
121	12	107
270	08	10
270	12	78
273	01	22
274	12	31

## PROJETO

<u>CodProj</u>	<u>DataFim</u>
01	17/07/95
08	12/01/96
12	21/03/96

# Normalização – 2FN (anomalias)

## Normalização – 2FN (anomalias)

- Inserir: só é possível criar cargos se houver empregados designados para eles.

## Normalização – 2FN (anomalias)

- Inserir: só é possível criar cargos se houver empregados designados para eles.
- Remover: se apagarmos um empregado que ocupa unicamente um cargo na empresa, perderemos a informação do cargo.

## Normalização – 2FN (anomalias)

- Inserir: só é possível criar cargos se houver empregados designados para eles.
- Remover: se apagarmos um empregado que ocupa unicamente um cargo na empresa, perderemos a informação do cargo.
- Atualizar: se um cargo mudar de nome, será necessário mudar todas as tabelas em que este cargo aparece.

# Normalização – 3FN (anomalias)

## EMPREGADO

<u>Matrícula</u>	Nome	CodCargo
120	João	1
121	Hélio	1
270	Gabriel	2
273	Silva	3
274	Abraão	2

## CARGO

CodCargo	Nome
1	Programador
2	Analista
3	Projetista

# Passos para o projeto de banco de dados relacionais<sup>1</sup>

---

<sup>1</sup><https://www.inf.ufsc.br/~r.fileto>

# Passos para o projeto de banco de dados relacionais<sup>1</sup>

- Guiado pelo bom-senso, construa um diagrama ER, agrupando os atributos nas tabelas que vão representar as entidades e os relacionamentos do seu banco de dados.

---

<sup>1</sup><https://www.inf.ufsc.br/~r.fileto>



# Passos para o projeto de banco de dados relacionais<sup>1</sup>

- Guiado pelo bom-senso, construa um diagrama ER, agrupando os atributos nas tabelas que vão representar as entidades e os relacionamentos do seu banco de dados.
- Construa o diagrama de dependências funcionais para as tabelas propostas no MER (ou um único diagrama de dependências funcionais considerando todos os atributos do seu banco de dados).

---

<sup>1</sup><https://www.inf.ufsc.br/~r.fileto>

# Passos para o projeto de banco de dados relacionais<sup>1</sup>

- Guiado pelo bom-senso, construa um diagrama ER, agrupando os atributos nas tabelas que vão representar as entidades e os relacionamentos do seu banco de dados.
- Construa o diagrama de dependências funcionais para as tabelas propostas no MER (ou um único diagrama de dependências funcionais considerando todos os atributos do seu banco de dados).
- Elimine os atributos repetitivos (se houver), de modo a obter um modelo de dados na 1FN.

---

<sup>1</sup><https://www.inf.ufsc.br/~r.fileto>

# Passos para o projeto de banco de dados relacionais<sup>1</sup>

- Guiado pelo bom-senso, construa um diagrama ER, agrupando os atributos nas tabelas que vão representar as entidades e os relacionamentos do seu banco de dados.
- Construa o diagrama de dependências funcionais para as tabelas propostas no MER (ou um único diagrama de dependências funcionais considerando todos os atributos do seu banco de dados).
- Elimine os atributos repetitivos (se houver), de modo a obter um modelo de dados na 1FN.
- Elimine as dependências parciais da chave primária em suas tabelas (se houver), obtendo um projeto na 2FN.

---

<sup>1</sup><https://www.inf.ufsc.br/~r.fileto>

# Passos para o projeto de banco de dados relacionais<sup>1</sup>

- Guiado pelo bom-senso, construa um diagrama ER, agrupando os atributos nas tabelas que vão representar as entidades e os relacionamentos do seu banco de dados.
- Construa o diagrama de dependências funcionais para as tabelas propostas no MER (ou um único diagrama de dependências funcionais considerando todos os atributos do seu banco de dados).
- Elimine os atributos repetitivos (se houver), de modo a obter um modelo de dados na 1FN.
- Elimine as dependências parciais da chave primária em suas tabelas (se houver), obtendo um projeto na 2FN.
- Elimine as dependências transitivas nas tabelas (se houver), obtendo um esquema na 3FN.

---

<sup>1</sup><https://www.inf.ufsc.br/~r.fileto>



# Questões para revisão

## Questões para revisão

- Qual a importância da normalização?

## Questões para revisão

- Qual a importância da normalização?
- Cite exemplos de normalização e descreva cada um.



## Questões para revisão

- Qual a importância da normalização?
- Cite exemplos de normalização e descreva cada um.
- O que você entende por atomicidade de atributos? Dê um exemplo.

# Exercício

# Exercício

Considere a tabela abaixo:

Código_cliente	Nome	Telefone	Endereço
C001	José	9563-6352 9847-2501	Rua Seis, 85 Morumbi 12536-965
C002	Maria	3265-8596	Rua Onze, 64 Moema 65985-963
C003	Janio	8645-8956 9598-6301	Praça Ramos Liberdade 68858-633

# Exercício

Considere a tabela abaixo:

Código_cliente	Nome	Telefone	Endereço
C001	José	9563-6352 9847-2501	Rua Seis, 85 Morumbi 12536-965
C002	Maria	3265-8596	Rua Onze, 64 Moema 65985-963
C003	<u>Janio</u>	8645-8956 9598-6301	Praça Ramos Liberdade 68858-633

Ela está na 1FN? Se não estiver, o que seria necessário ajustar?

# Exercício

Considere a relação  $R = \{\underline{A}, \underline{B}, C, D, E, F, G, H\}$ .

Considere também o conjunto de dependências funcionais abaixo:

- $\{A, B\} \rightarrow \{C, D, E, F\}$
- $\{B\} \rightarrow \{G, H\}$

Normalize a relação R. Mostre cada etapa.