

# Fundamentos de Banco de Dados

## Normalização



- **Objetivo**
  - Apresentar uma abordagem de projeto de banco de dados, denominada de Normalização, a qual permite analisar a qualidade das relações, bem como elevar a sua qualidade.
- **Principais tópicos**
  - Anomalias
  - Tuplas espúrias
  - Abordagens de Projeto de Banco de Dados
  - Dependências Funcionais
  - Regras de Inferência para DFs



- Principais tópicos (*Continuação*)
  - Formas Normais com base em Chaves Primárias
  - Definição Geral de Formas Normais
  - BCNF (Boyce-Codd Normal Form)
  - Dependências Multivaloradas
  - Quarta Forma Normal (4FN)





- Top-down
  - Iniciar com o agrupamento dos atributos obtidos a partir do projeto conceitual de mapeamento
  - Isso é chamado de projeto por análise
- Bottom-up
  - Considerar os relacionamentos entre atributos
  - Construir as relações
  - Isso é chamado projeto pela síntese
- Nossa Abordagem
  - Utilizar a abordagem Top-down para obter as relações
  - Utilizar a abordagem Bottom-up para melhorar a qualidade das relações obtidas anteriormente



- Cuidado com redundância de informação

**EMP\_DEP**

<u>NSS</u>	NOME	DTANIV	DNUMERO	DNOME	GERENTE
21	AA	-	5	CV	91
22	BB	-	5	CV	91
23	CC	-	6	TS	93
24	DD	-	7	OS	94
25	EE	-	7	OS	94

— Anomalias de Inserção:

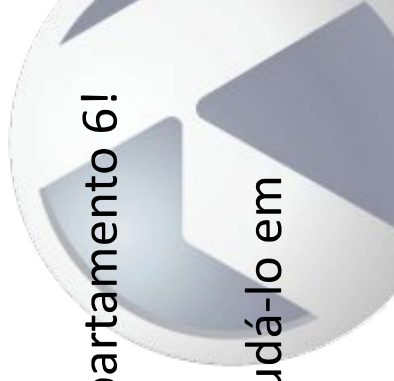
- Como inserir novo departamento sem que exista empregados?
- Inserir empregados é difícil quando informações de departamento devem ser inseridas corretamente.

— Anomalias de Remoção:

- O que acontece quando removemos CC? Perdemos o departamento 6!

— Anomalias de Alteração:

- Se mudarmos o gerente do departamento 5, devemos mudá-lo em todas as tuplas com DNUMERO = 5.



- Não quebre uma relação em relações que possam gerar tuplas espúrias (falsas)

DNUMERO	NOME	PNOME	PLOCALIZAÇÃO
123	XX	Compras	São Paulo
123	XX	Vendas	Rio de Janeiro
124	YY	Logística	São Paulo

- A relação pode ser quebrada em

DNUMERO	NOME	PNOME
123	XX	Compras
123	XX	Vendas
124	YY	Logística

DNUMERO	PLOCALIZAÇÃO
123	São Paulo
123	Rio de Janeiro
124	São Paulo

- Quando fazemos o Join, obtemos NOVAS TUPLAS!

<u>DNUMERO</u>	NOME	PNOME	PLOCALIZAÇÃO
123	XX	Compras	São Paulo
123	XX	Compras	Rio de Janeiro
123	XX	Vendas	São Paulo
123	XX	Vendas	Rio de Janeiro
124	YY	Logística	São Paulo

Após o Join, o resultado não foi a relação original. Assim, houve perda de informações. Conclui-se que houve uma decomposição com perdas.



- Quando o valor um determinado atributo (**Y**) dentro de uma relação (**R**) pode ser determinado pelo valor de outro atributo (**X**) podemos dizer que eles possuem uma dependência funcional, sendo que **X** determina funcionalmente **Y**:

$X \rightarrow Y$  (notação de dependência funcional)

$X$  = Determinante

$Y$  = Dependente

- **Exemplos:**

- $\{\text{CEP}\} \rightarrow \{\text{CIDADE, ESTADO}\}$
- $\{\text{CPF}\} \rightarrow \{\text{NOME, DATA DE NASCIMENTO}\}$
- $\{\text{RG, UF}\} \rightarrow \{\text{NOME}\}$
- $\{\text{Nome Jogo}\} \rightarrow \{\text{Produtor, Distribuidor}\}$



- Dependências funcionais (DFs) são usadas para medir formalmente a qualidade do projeto relacional
- As DFs e chaves são usadas para definir formas normais de relações



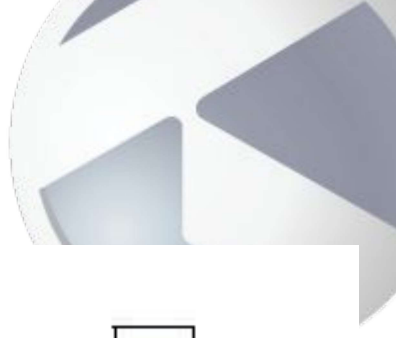
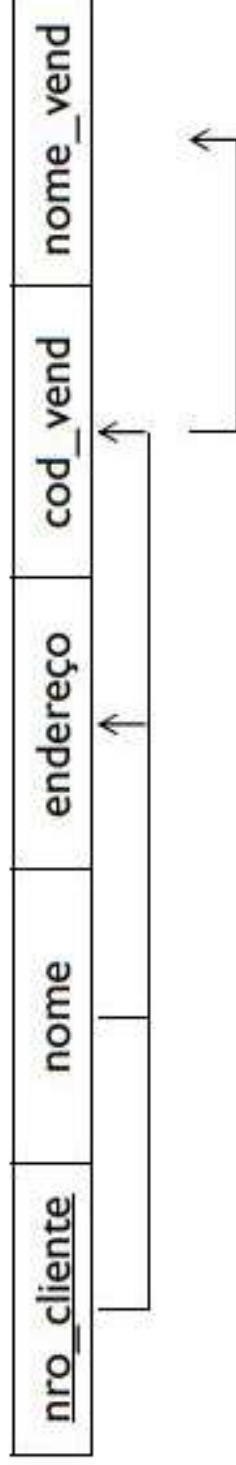
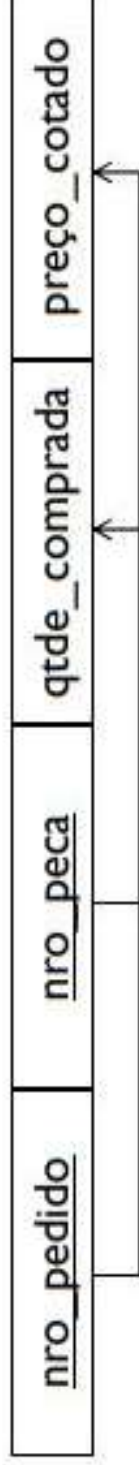
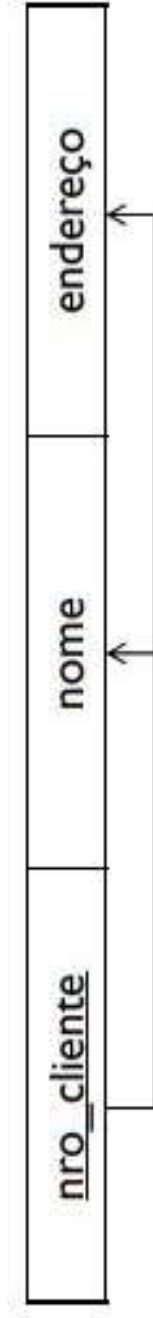


- A chave primaria determina funcionalmente todos os atributos da relação
- Se  $X \rightarrow Y$ 
  - Se duas tuplas tiverem o mesmo valor para X, elas devem ter o mesmo valor para Y. Ou seja:
  - Se  $X \square Y$  então, para quaisquer tuplas t1 e t2 de r(R):
  - Se t1[código departamento] = t2[código departamento]
  - Se t1[departamento] = t2[departamento]
  - As DFs são derivadas das restrições do mundo real e não de uma extensão específica da relação R

<u>CODIGO FUNCIONARIO</u>	NOME	DATA NASCIMENTO	COD DEPARTAMENTO	DEPARTAMENTO
100	FERNANDA AUGUSTA	01/04/1988	10	FINANCEIRO
200	FELIX DAMACENO	02/04/1988	10	FINANCEIRO
300	JOÃO MARQUES	03/04/1988	20	TECNOLOGIA
400	CARLOS FERNANDI	03/04/1988	20	TECNOLOGIA



## Notação Diagramática para Dependências funcionais



# Formas Normais com base em Chaves Primárias

---

- Normalização de Relações
- Uso prático de Formas Normais
- Definições de Chaves e de Atributos que participam de Chaves
- Primeira Forma Normal
- Segunda Forma Normal
- Terceira Forma Normal





- Norma é um conjunto de padrões, regras e diretrizes colocadas de forma simples, que contém orientações comportamentais ou especificações técnicas
- Normalização em banco de dados
  - Processo de decompor relações “ruins” dividindo seus atributos em relações menores e “melhores”
- Forma Normal
  - Indica o nível de qualidade de uma relação
- 1FN
  - Definição de relação. Atributos atômicos (indivisíveis).
- 2FN, 3FN, BCNF
  - Baseiam-se em chaves e DFs de uma relação esquema
- 4FN e 5FN
  - Baseiam-se em chaves e dependências multivaloradas



# Uso Prático das Formas Normais

---

- Na prática, a normalização é realizada para obter projetos de alta qualidade
- Os projetistas de bancos de dados não precisam normalizar na maior forma normal possível.
- Desnormalização
  - Processo de armazenar junções de relações de forma normal superior como uma relação base que está numa forma normal inferior





- Uma superchave é um conjunto de atributos que identificam univocamente uma tupla da outra.
- Superchave mínima é um conjunto mínimo de atributos que identificam uma tupla.
- Uma superchave, é uma chave se for uma superchave mínima. Ou seja, se em uma superchave for possível retirar algum atributo e mesmo assim conseguirmos identificar uma linha da outra, não serve pra ser chave.
- Se uma relação esquema tiver mais de uma chave, cada chave será chamada de chave-candidata. Uma das chaves-candidatas é arbitrariamente escolhida para ser a chave-primária e as outras são chamadas de chaves-secundárias.



- Um atributo primo (ou primário) é membro de alguma chave-candidata
- Um atributo não-primo é um atributo que não é primo – isto é, não é membro de qualquer chave-candidata

## TRABALHA\_EM

<u>NSS</u>	<u>PNRO</u>	HORAS
↓	↓	↓
Primo	Primo	Não-Primo



- Proíbe que relações tenham
  - Atributos compostos
  - Atributos multivalorados
  - Relações aninhadasOu seja
  - Permite apenas atributos que sejam atômicos
- Considerado como parte da definição de relação





## EMPREGADO

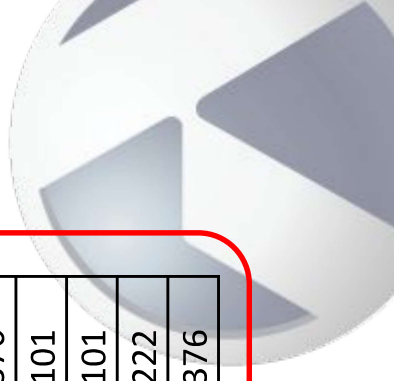
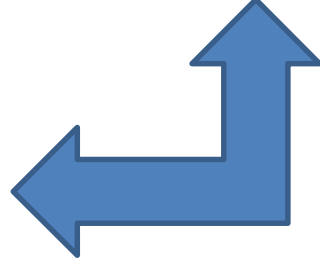
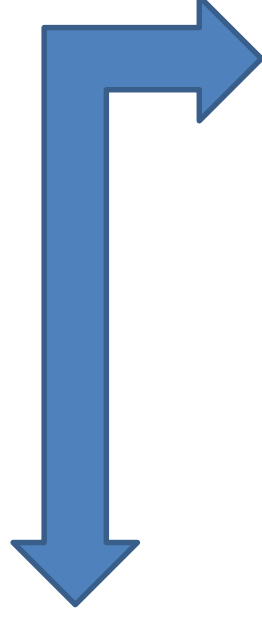
ENAME	<u>NSS</u>	Telefones
Joaquim	305	555-444
Katarina	381	555-333
Daví	422	555-101
Carlos	489	555-376
Bárbara	533	555-222
		555-376

## TELEFONE

<u>NSS</u>	<u>NÚMERO</u>
305	555-444
381	555-333
489	555-376
533	555-101
381	555-101
489	555-222
489	555-376

## EMPREGADO

ENAME	<u>NSS</u>
Joaquim	305
Katarina	381
Daví	422
Carlos	489
Bárbara	533



EMPREGADO\_PROJETO

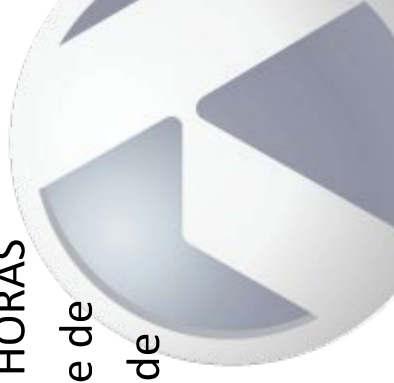
ENOME	NSS	PROJETOS	
		PNUMERO	HORAS
Joaquim	305	1	32
Joaquim	305	2	8
Daví	422	3	40
Carlos	489	1	20
Carlos	489	3	20

ENOME	<u>NSS</u>
Joaquim	305
Daví	422
Carlos	489

PROJETO

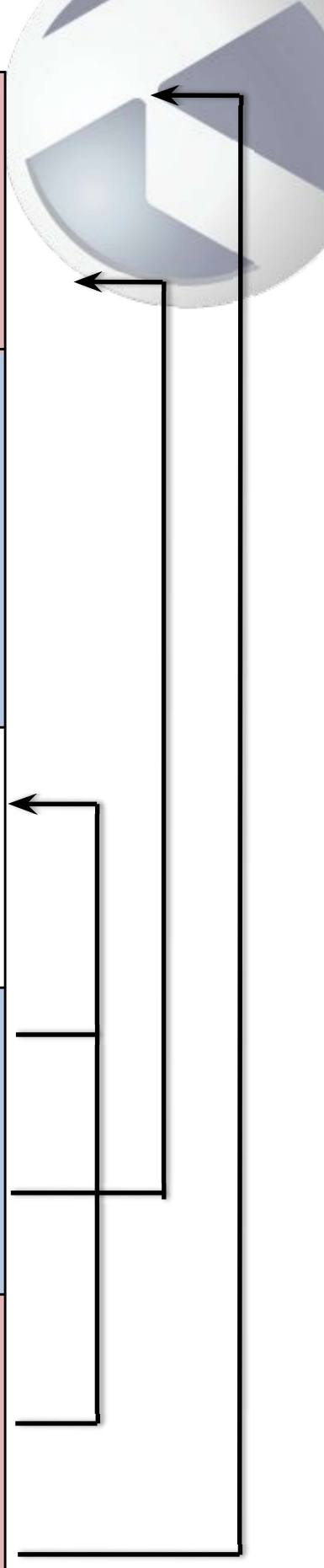
<u>NSS</u>	<u>PNUMERO</u>	HORAS
305	1	32
305	2	8
422	3	40
489	1	20
489	3	20

- Para entender a 2FN precisamos entender:
  - Dependência Funcional
  - Chave-primária
  - Atributo Não-Primo
  - Dependência funcional total
- Uma DF,  $Y \twoheadrightarrow Z$ , onde a remoção de qualquer atributo de  $Y$  invalida a DF.
  - Exemplos:
    - $\{ \text{NSS}, \text{PNUMERO} \} \twoheadrightarrow \text{HORAS}$  é dependente totalmente de
      - $\{ \text{NSS}, \text{PNUMERO} \}$ , uma vez que
      - NSS não determina HORAS e nem PNUMERO determina HORAS
    - $\{ \text{NSS}, \text{PNUMERO} \} \twoheadrightarrow \text{ENOME}$  não é dependente totalmente de
      - $\{ \text{NSS}, \text{PNUMERO} \}$ ; ENOME é dependente parcialmente de
        - $\{ \text{NSS}, \text{PNUMERO} \}$ , pois  $\text{NSS} \twoheadrightarrow \text{ENOME}$



- Para entender a 2FN precisamos entender:
  - Dependência Funcional
  - Chave-primária
  - Atributo Não-Primo
  - Dependência funcional total

<u>CODIGO PROJETO</u>	<u>CODIGO FUNCIONARIO</u>	HORAS	NOME FUNCIONARIO	NOME PROJETO
10	100	200	CAIO FERNANDO	ALFA
20	100	50	CAIO FERNANDO	BETA
10	200	120	MARIA ANTONIA	ALFA
20	200	140	MARIA ANTONIA	BETA





# Segunda Forma Normal

---

- Uma relação esquema R está na 2FN se estiver na 1FN e todos os atributos não-primos forem totalmente dependentes da chave-primária.
- Podemos decompor a relação em relações que estejam na 2 FN através do processo de normalização.
- Se uma relação estiver na 1FN e só possuir uma chave simples, podemos dizer que ela já está na 2FN, já que todos os atributos não primos são totalmente dependente da chave-primária



Horas projeto:

CODIGO PROJETO ce	CODIGO FUNCIONARIO ce	HORAS
10	100	200
20	100	50
10	200	120
20	200	140

Projeto:

<u>CODIGO PROJETO</u>	NOME PROJETO
10	ALFA
20	BETA

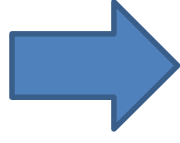
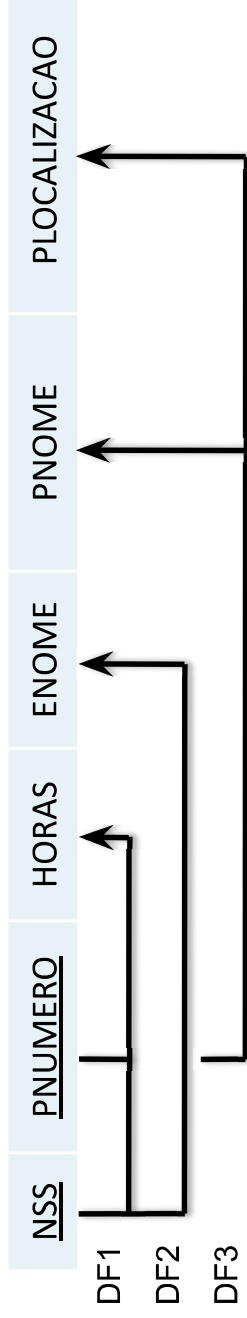
Funcionário:

<u>CODIGO FUNCIONARIO</u>	NOME FUNCIONARIO
100	CAIO FERNANDO
200	MARIA ANTONIA



# Normalização para a 2FN e 3FN

EMPREGADO\_PROJETO



EMPREGADO\_PROJETO

<u>NSS</u>	<u>PNUMERO</u>	HORAS
------------	----------------	-------



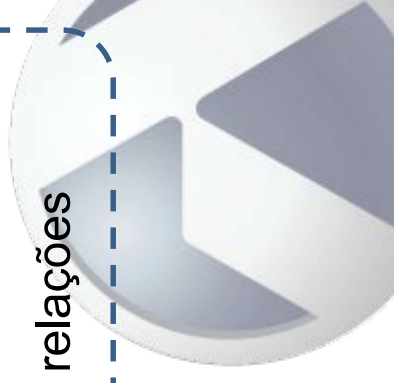
Não-Primo

Depende totalmente  
da chave primária

EMPREGADO PROJETO

<u>NSS</u>	ENOME	<u>PNUMERO</u>	PNOME	PLOCALIZACAO
------------	-------	----------------	-------	--------------

O mesmo ocorre com as outras relações



- Para entender a 3FN precisamos entender:
  - 2FN
  - Atributo Não-Primo
  - Dependência funcional transitiva
    - Se  $X \twoheadrightarrow Y$  e  $Y \twoheadrightarrow Z$  então  $X \twoheadrightarrow Z$







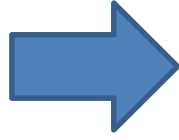
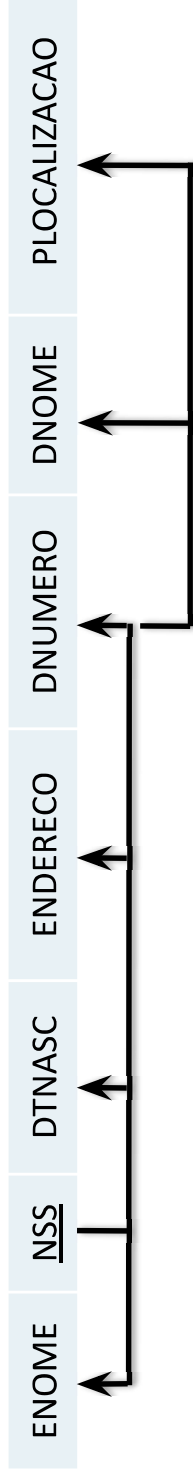
- Uma relação esquema R está na 3FN se ela estiver na 2FN e nenhum atributo não-primo for transitivamente dependente da chave-primária.
- Pode ser decomposto em relações que estejam na 3FN via o processo de normalização.
- NOTA:
  - Em  $X \twoheadrightarrow Y$  e  $Y \twoheadrightarrow Z$ , sendo X a chave-primária, pode ser considerado um problema se, e somente se, Y não for uma chave-candidata. Quando Y é uma chave-candidata, não existe problema com a dependência transitiva
  - Por exemplo, considere EMP (NSS, Emp#, Salario ).
    - Aqui, NSS  $\twoheadrightarrow$  Emp#  $\twoheadrightarrow$  Salario e Emp# é uma chave-candidata



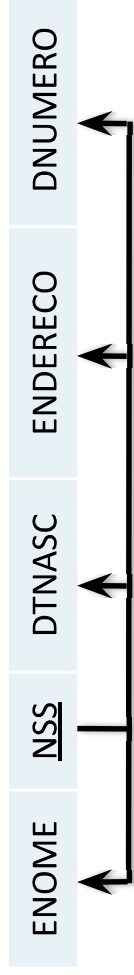


# Normalização para a 2FN e 3FN

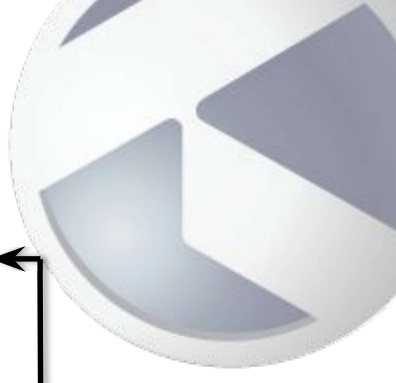
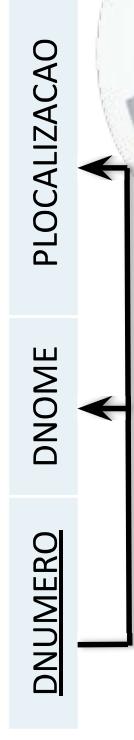
## EMPREGADO\_PROJETO



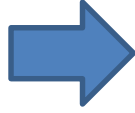
## EMPREGADO



## PROJETO



ID JOGO	NOME JOGO	PRODUTORA	PLATAFORMA	FABRICANTE
1	MORTAL KOMBAT 11	NETHER REALM	PLAYSTATION 4	SONY
2	MORTAL KOMBAT 11	NETHER REALM	XBOX ONE	MICROSOFT
3	RESIDENT EVIL 2	CAPCOM	PLAYSTATION 4	SONY
4	RESIDENT EVIL 2	CAPCOM	XBOX ONE	MICROSOFT
5	DARK SOULS 3	FROMSOFT	PLAYSTATION 4	SONY
7	DARK SOULS 3	FROMSOFT	XBOX ONE	MICROSOFT



JOGO PLATAFORMA

ID JOGO	NOME JOGO	PLATAFORMA
1	MORTAL KOMBAT 11	PLAYSTATION 4
2	MORTAL KOMBAT 11	XBOX ONE
3	RESIDENT EVIL 2	PLAYSTATION 4
4	RESIDENT EVIL 2	XBOX ONE
5	DARK SOULS 3	PLAYSTATION 4
7	DARK SOULS 3	XBOX ONE

PLATAFORMA

PLATAFORMA	FABRICANTE
PLAYSTATION 4	SONY
XBOX ONE	MICROSOFT

JOGO

NOME JOGO	PRODUTORA
MORTAL KOMBAT 11	NETHER REALM
RESIDENT EVIL 2	CAPCOM
DARK SOULS 3	FROMSOFT

## ■ Referências Bibliográficas

1. Elmasri, R.; Navathe, S. B. [Trad.]. Sistemas de bancos de dados. Traduzido do original: FUNDAMENTALS OF DATABASE SYSTEMS. São Paulo: Pearson(Addison Wesley), 2005. 724 p. ISBN: 85-88639-17-3.
2. Korth, H.; Silberschatz, A. Sistemas de Bancos de Dados. 3a. Edição, Makron Books, 1998.
3. Raghu Ramakrishnan e Johannes Gehrke, Database Management Systems, Second Edition, McGraw-Hill, 2000.
4. Teorey, T.; Lightstone, S.; Nadeau, T. Projeto e modelagem de bancos de dados. Editora Campus, 2007.

## ■ Referências Web

1. Takai, O.K; Italiano, I.C.; Ferreira, J.E. Introdução a Banco de Dados. Apostila disponível no site: <http://www.ime.usp.br/~jef/apostila.pdf>. (07/07/2005).



- **Referências Web**

2 - Rocha A. Santanché A. Dependências Funcionais e Normalização

<https://www.ic.unicamp.br/~rocha/teaching/2012s2/mc536/aulas/aula-08-final.pdf>

L.pdf 2012



---

# Obrigado!

