

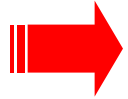


FACULDADE DE TECNOLOGIA DE SÃO PAULO

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS - ADS -**

**DISCIPLINA: BANCO DE DADOS
(Notas de Aula)**

Prof. Dr. Napoleão Verardi Galegale



Aula/Semana 10 (Teoria):

5. Modelo Relacional:

- a) Definição de relação, tupla e domínio
- b) Integridade relacional
- d) Normalização (Primeira, Segunda e Terceira Formas Normais)
- e) Transposição do MER para o Modelo Relacional
 - Exemplo de Estudo de Caso de Normalização
 - Exercício sobre modelo relacional e normalização

- ✓ O modelo relacional é um modelo de dados que se baseia no princípio em que todos os dados estão organizados, matematicamente falando, em relações. Sua definição é teórica e baseada na lógica de predicados e na teoria dos conjuntos.
- ✓ O conceito foi criado por E. F. Codd em 1970, sendo descrito no artigo *Relational Model of Data for Large Shared Data Banks* (Communications of the ACM, Vol. 13, No. 6, June 1970, pp. 377-387).
- ✓ Um **banco de dados relacional** é composto por **relações**.
- ✓ Uma **relação** é um conjunto não ordenado de ***n*-tuplas**.
- ✓ Uma **tupla** é composta por uma série de **domínios**.
- ✓ Cada **domínio** é identificado por um **nome de domínio**.
- ✓ As ***n*-tuplas** de uma **relação** não têm ordenação.
- ✓ **Chave primária** é um ou mais domínios que **identificam** uma **tupla** da **relação**.
- ✓ **Chave estrangeira** é um ou mais domínios que aparecem como **chave primária** em uma outra relação, permitindo a implementação de **relacionamentos**.

✓ Notação do modelo relacional:

Uma relação R de grau n é indicada por $R (D_1, D_2, D_3, \dots, D_n)$

Uma n -tupla t em uma relação $r(R)$ é indicada por $t = (v_1, v_2, \dots, v_n)$, em que v_i é o valor correspondente ao domínio D_i

Empregado (CodEmp, NomeEmp, CodDepto, DtNascEmp)

Empregado

CodEmp	NomeEmp	CodDepto	DtNascEmp
E3	João	D1	10/05/1973
E2	José	D1	20/02/1965
E1	Pedro	D2	19/02/1981
E4	Antonio	D2	06/09/1984

← n -tupla t

↖
Chave
Primária

↖
Chave
Estrangeira

✓ **INTEGRIDADE DE DOMÍNIO:**

O valor de um domínio deve obedecer a definição de valores admitidos para o domínio (número inteiro, real, alfanumérico de tamanho definido, data, etc.).

✓ **INTEGRIDADE DE VAZIO:**

Restrição que define se os valores de um domínio podem ou não ser vazios (obrigatório ou opcional). Domínios que compõem a chave primária devem ser não vazios.

✓ **INTEGRIDADE DE CHAVE:**

Restrição que define se os valores da chave primária e/ou chave estrangeira devem ser únicos.

✓ **INTEGRIDADE REFERENCIAL:**

Restrição que define se os valores dos domínios que compõem uma chave estrangeira devem também aparecer na chave primária da tabela referenciada.

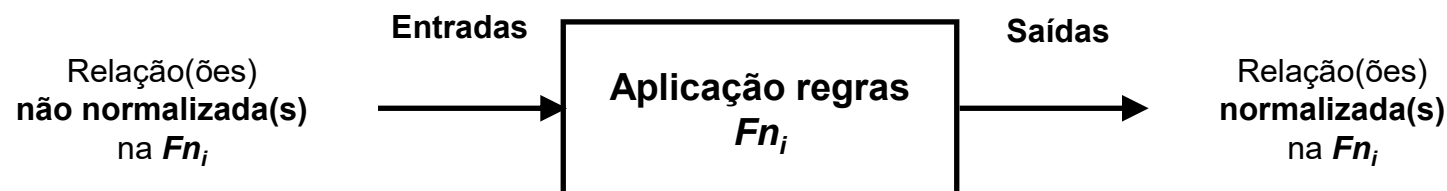
✓ DEFINIÇÃO:

A normalização de relações foi proposta por E. Codd em 1970, juntamente com o próprio modelo de dados relacional, como uma técnica para eliminar redundâncias de informações e evitar anomalias causadas pela inserção, atualização e remoção de tuplas.

✓ PROCESSO DE NORMALIZAÇÃO:

Baseado no conceito de formas normais (processo sistemático de geração de tabelas). Uma relação R está em uma forma normal se atende uma certa regra:

“se R está na Forma Normal $(FN)_i$, então R está automaticamente nas $FN_{i-1, i-2, \dots}$ ”.



✓ PRIMEIRA FORMA NORMAL (1FN):

Uma relação R está na **1FN**, se e somente se, todos os seus domínios são domínios simples, ou seja, não correspondem a itens estruturados e/ou repetitivos (aninhados).

Alternativas:

- a) Construir uma única relação com redundância de dados (maior correção do processo de normalização)
- b) Construir uma relação para cada relação aninhada (mais utilizada)

✓ **Exemplo 1FN: Uma única relação com redundância de dados**

Relação não normalizada:

Projeto (CodProj, NomeProj, CodEmp, NomeEmp, CodFaixa, ValorFaixa, HorasProj)

1FN:

Projeto' (CodProj, NomeProj, CodEmp, NomeEmp, CodFaixa, ValorFaixa, HorasProj)

CodProj	NomeProj	CodEmp	NomeEmp	CodFaixa	ValorFaixa	HorasProj
P1	Financeiro	E1	Pedro	JR	4.000,00	160
P1	Financeiro	E2	José	SR	12.000,00	80
P1	Financeiro	E3	João	PL	8.000,00	120
P2	Automação	E4	Antonio	JR	4.000,00	160
P2	Automação	E5	Alfredo	PL	8.000,00	160
P2	Automação	E2	José	SR	12.000,00	80

✓ Exemplo 1FN: Uma relação para cada relação aninhada

Relação não normalizada:

Projeto (*CodProj*, *NomeProj*, *CodEmp*, *NomeEmp*, *CodFaixa*, *ValorFaixa*, *HorasProj*)

1FN:

Projeto' (*CodProj*, *NomeProj*)

Alocação (*CodProj*, *CodEmp*, *NomeEmp*, *CodFaixa*, *ValorFaixa*, *HorasProj*)

CodProj	NomeProj	CodEmp	NomeEmp	CodFaixa	ValorFaixa	HorasProj
P1	Financeiro	E1	Pedro	JR	4.000,00	160
		E2	José	SR	12.000,00	80
		E3	João	PL	8.000,00	120
P2	Automação	E4	Antonio	JR	4.000,00	160
		E5	Alfredo	PL	8.000,00	160
		E2	José	SR	12.000,00	80

Itens estruturados
(aninhados)

✓ DEPENDÊNCIA FUNCIONAL:

Um domínio D_2 *depende funcionalmente* de um domínio D_1 (ou um domínio D_1 *determina* um domínio D_2) quando, em todas as tuplas da relação, para cada valor de D_1 que aparece na relação, aparece o mesmo valor de D_2 .

Exemplo:

D_1	D_2	D_3	D_4
a	1	x	10
b	3	z	30
d	5	y	25
a	1	x	10
k	7	w	15
d	5	y	25
b	3	z	30

$$D_1 \longrightarrow D_4$$

$$D_1 \longrightarrow D_2$$

$$(D_1, D_2) \longrightarrow D_3 \text{ (Dependência Funcional composta)}$$

✓ **DEPENDÊNCIA FUNCIONAL TOTAL:**

Se um domínio D_x depende funcionalmente de todos os domínios que compõem a chave primária de uma relação R , então D_x possui dependência funcional total da chave primária de R .

✓ **DEPENDÊNCIA FUNCIONAL PARCIAL:**

Se um domínio D_x depende funcionalmente de apenas alguns domínios (não todos) que compõem a chave primária de uma relação R , então D_x possui dependência funcional parcial da chave primária de R .

✓ SEGUNDA FORMA NORMAL (2FN):

Uma relação **R** está na **2FN**, se e somente se, ela estiver na 1FN e não possuir dependências funcionais parciais. Relações com chave primária composta de apenas um domínio já estão na 2FN.

Exemplo 1FN:

Projeto' (CodProj, NomeProj, CodEmp, NomeEmp, CodFaixa, ValorFaixa, HorasProj)

$CodProj \rightarrow NomeProj$

$CodEmp \rightarrow NomeEmp, CodFaixa, ValorFaixa,$

$(CodProj, CodEmp) \rightarrow HorasProj$

2FN:

Projeto'' (CodProj, NomeProj)

Empregado (CodEmp, NomeEmp, CodFaixa, ValorFaixa)

Alocação (CodProj, CodEmp, HorasProj)

✓ TERCEIRA FORMA NORMAL (3FN):

Uma relação **R** está na **3FN**, se e somente se, ela estiver na 2FN e não possuir dependências funcionais transitivas. Relações que possuem nenhum ou apenas um domínio que não faz parte da chave primária já estão na 3FN

Exemplo 2FN:

Projeto (CodProj, NomeProj)

Empregado (CodEmp, NomeEmp, CodFaixa, ValorFaixa)

Alocação (CodProj, CodEmp, HorasProj)

$CodEmp \rightarrow CodFaixa$

$CodEmp \rightarrow ValorFaixa$

$CodFaixa \rightarrow ValorFaixa$

✓ TERCEIRA FORMA NORMAL (3FN) cont.:

3FN:

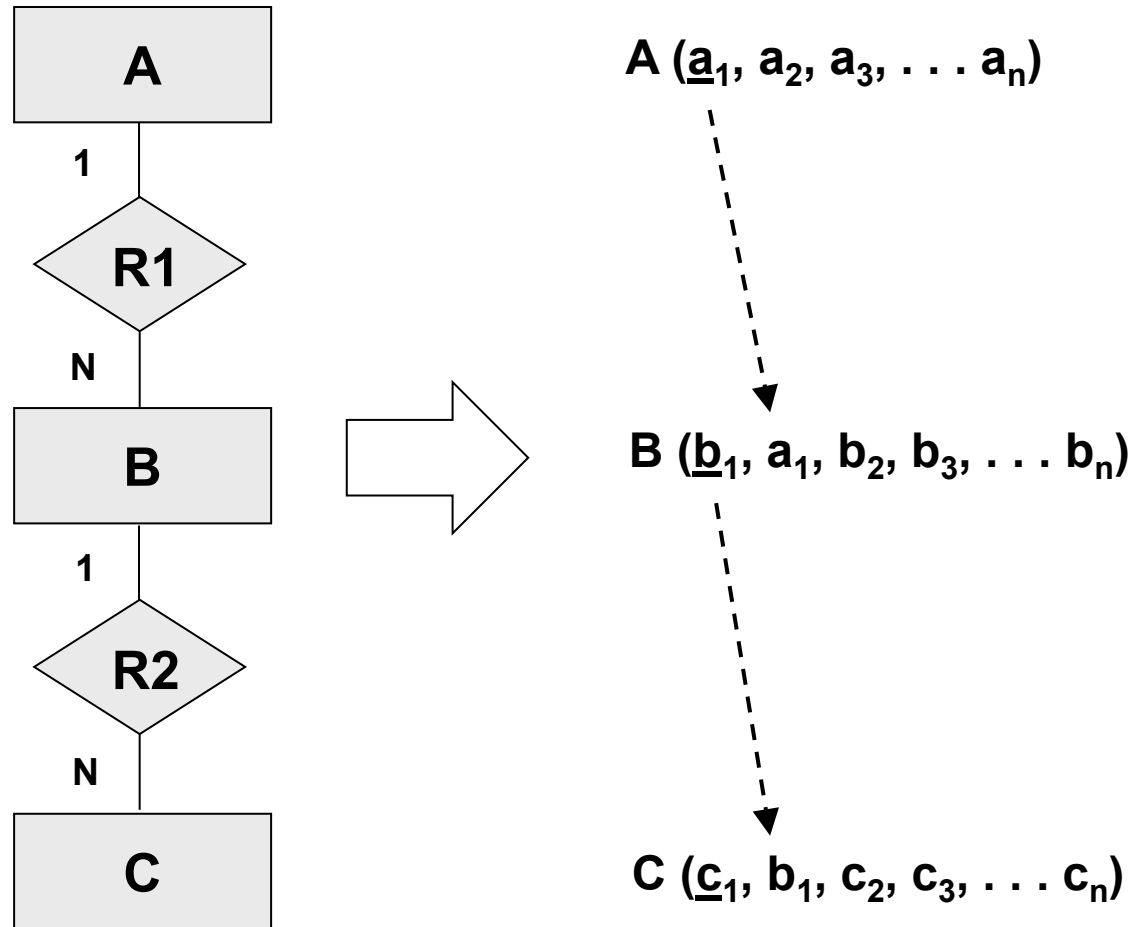
Projeto'' (CodProj, NomeProj)

Empregado' (CodEmp, NomeEmp, CodFaixa)

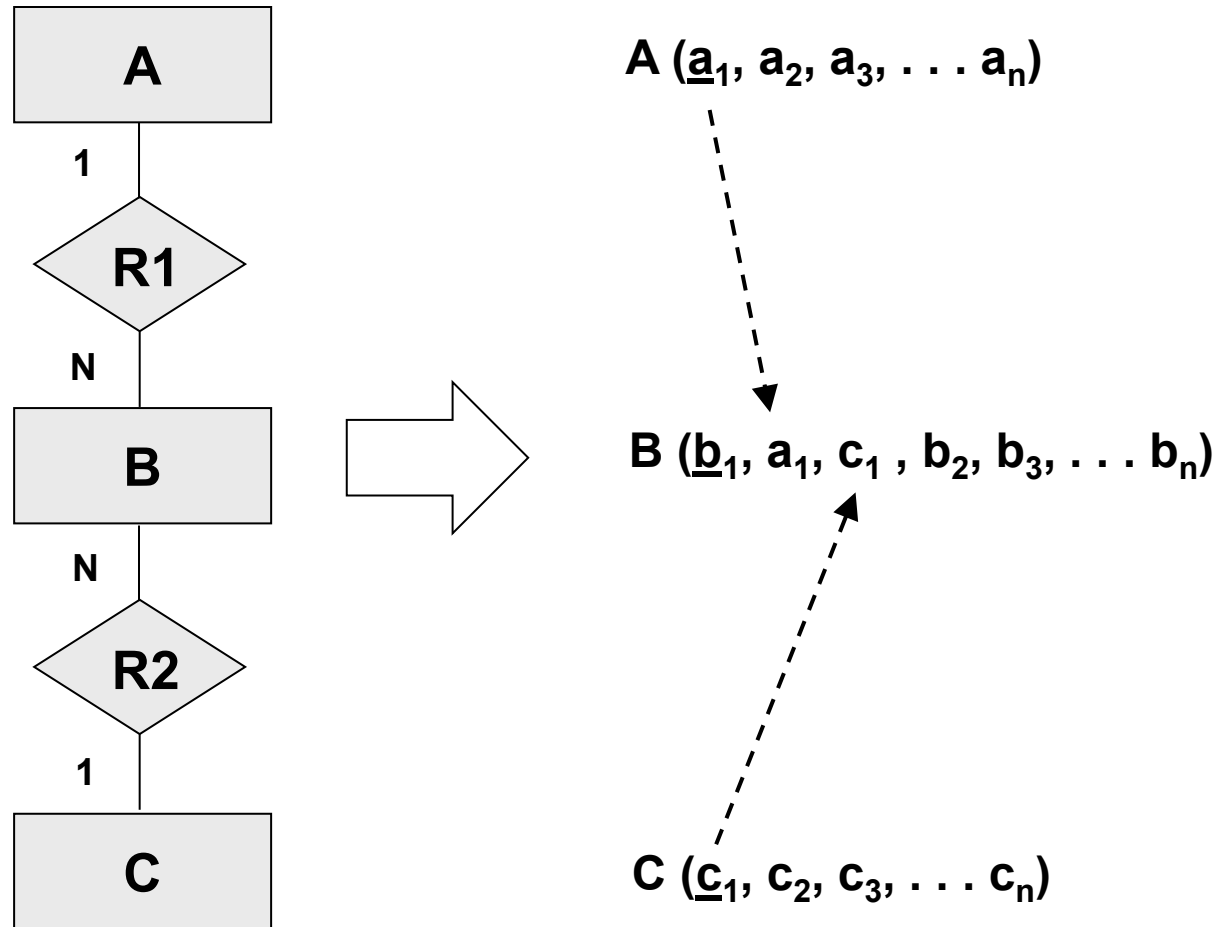
Alocação (CodProj, CodEmp, HorasProj)

Faixa (CodFaixa, ValorFaixa)

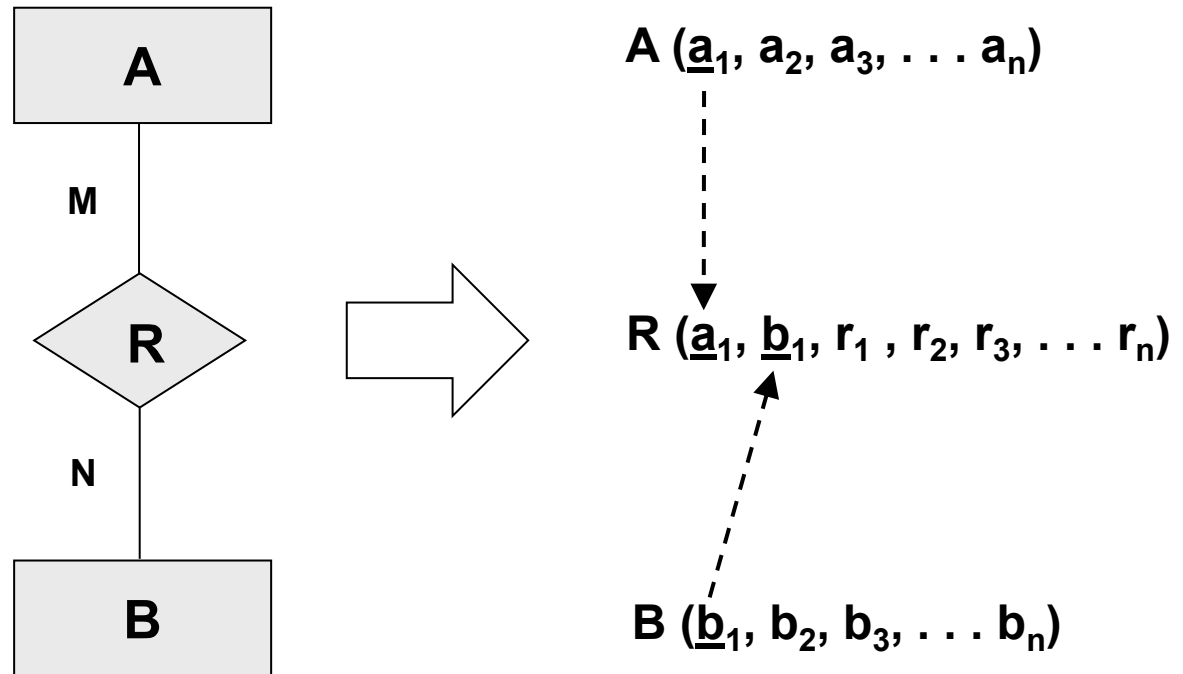
✓ CARDINALIDADES 1:N HIERÁRQUICAS:



✓ CARDINALIDADES 1:N NÃO HIERÁRQUICAS:



✓ CARDINALIDADES M:N



NORMALIZAÇÃO - EXEMPLO

(Exercício de Normalização – SCA – Sistema de Controle de Acessos)

Cod_Depto	Nome_Depto	Matricula	Nome_Usu	Sigla_Sist	Nome_Sist	Tipo_Acesso	Data_Ini	DataFim
RH	Rec. Humanos	1010	Maria	C1	Cad Func	G	2.1.2023	2.6.2023
RH	Rec. Humanos	1020	Paulo	C1	Cad Func	O	2.1.2023	2.2.2023
RH	Rec. Humanos	1030	Bia	C1	Cad Func	O	2.1.2023	2.2.2023
TI	Tecnolog Info	1040	Jose	C1	Cad Func	G	2.1.2023	2.2.2023
TI	Tecnolog Info	1040	Jose	E1	Estoque	G	2.1.2023	2.3.2023
TI	Tecnolog Info	1050	Ana	E1	Estoque	O	2.1.2023	2.2.2023
CP	Compras	1060	Pedro	E1	Estoque	G	2.1.2023	2.6.2023
CP	Compras	1060	Pedro	C1	Cad Func	O	2.1.2023	2.2.2023
TI	Tecnolog Info	1050	Ana	C1	Cad Func	O	2.2.2023	2.3.2023
CP	Compras	1070	Luzia	C1	Cad Func	O	2.1.2023	2.2.2023
CP	Compras	1070	Luzia	E1	Estoque	O	2.1.2023	2.2.2023
CP	Compras	1080	Carlos	E1	Estoque	S	2.1.2023	2.4.2023
CP	Compras	1080	Carlos	C1	Cad Func	O	2.1.2023	2.2.2023
TI	Tecnolog Info	1050	Ana	S1	Sist SQL	S	2.1.2023	2.3.2023
TI	Tecnolog Info	1040	Jose	S1	Sist SQL	G	1.2.2023	2.4.2023

G = Gestão; S = Supervisão; O = Operação

Pede-se:

- 1) Aplicar as regras de normalização para se obter as relações na terceira forma normal (3FN). Apresentar as formas normais: 1FN, 2FN e 3FN.
- 2) Desenvolver os diagramas E-R lógico e conceitual a partir das relações obtidas na 3FN.

(Exercício de Normalização – SCA – Sistema de Controle de Acessos)

Forma Não-Normal (desnormalizada)

(Cod_Depto, Nome_Depto, Matrícula, Nome_Usu, Sigla_Sist, Nome_Sist, Tipo_Acesso, Data_Ini, DataFim)

Solução

1FN:

R1 = (Cod_Depto, Nome_Depto, Matrícula, Nome_Usu, Sigla_Sist, Nome_Sist, Tipo_Acesso, Data_Ini, DataFim)

2FN:

R1a = (Matrícula, Nome_Usu, Cod_Depto, Nome_Depto)

R1b = (Sigla_Sist, Nome_Sist)

R1c = (Matrícula, Sigla_Sist, Tipo_Acesso, Data_Ini, DataFim)

3FN:

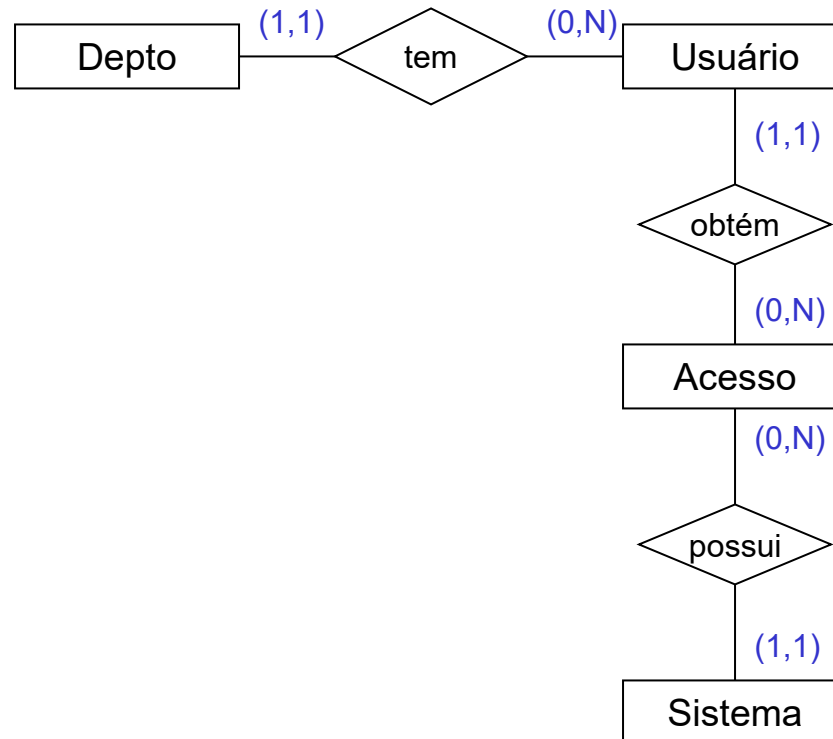
R1a2 = (Cod_Depto, Nome_Depto) Departamento

R1a1 = (Matrícula, Nome_Usu, Cod_Depto) Usuário

R1b = (Sigla_Sist, Nome_Sist) Sistema

R1c = (Matrícula, Sigla_Sist, Tipo_Acesso, Data_Ini, DataFim) Acesso

(Exercício de Normalização – SCA – Sistema de Controle de Acessos)



(Exercício de Normalização – SCA – Sistema de Controle de Acessos)

