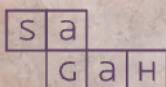


# CONSULTAS EM BANCO DE DADOS

Anderson Sene Gonçalves



SOLUÇÕES  
EDUCACIONAIS  
INTEGRADAS



# Diferença de conjuntos em SQL

## Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Conceituar a operação de diferença em álgebra relacional.
- Exemplificar a operação de diferença em banco de dados relacional.
- Demonstrar a operação de diferença em SQL.

## Introdução

Neste capítulo, você estudará sobre o operador de diferença de álgebra relacional e como pode ser aplicado na construção de consultas em banco de dados.

Para isso, inicialmente será realizada uma explanação a respeito do operador de diferença em álgebra relacional e sua aplicação em conjuntos de dados, bem como a sua representação seguindo o diagrama de Venn.

Em seguida, você estudará como esse operador pode ser usado na construção de consultas em bancos de dados e quais os retornos esperados após a sua aplicação.

Por fim, verá na prática como o operador de diferença pode ser empregado em consultas em bancos de dados por meio do comando `EXCEPT`.

## 1 Diferença em álgebra relacional

Quando aplicados em consultas em banco de dados, os conceitos de álgebra relacional contribuem para a tipos de consultas de maior complexidade.

Entre os operadores tradicionais de conjuntos, existe o operador de diferença, que, ao ser empregado na álgebra relacional, tem o objetivo de retornar os dados existentes apenas no primeiro conjunto e que não estejam nos conjuntos subsequentes.

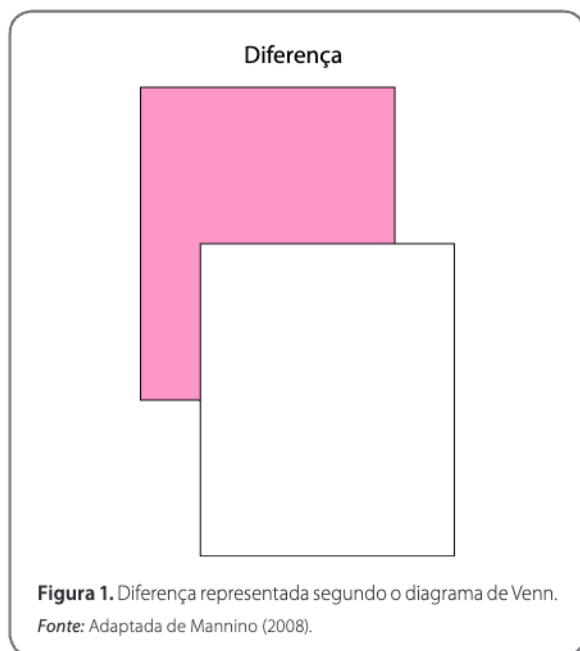
Na visão de Machado (2014), uma operação relacional age (opera) sobre um (ou mais) conjunto de dados e fornece um novo conjunto como resultado.

Assim, pode-se afirmar que um conjunto de dados é representado por uma única informação ou por um conjunto de informações, por exemplo, um conjunto de dados de automóveis pode dispor apenas da informação do nome, como também do modelo, do ano de fabricação, do valor de compra, etc.

A partir de dois conjuntos de dados, podemos usar um operador para gerar como resultado um novo conjunto de dados. Entre os operadores existentes, há o operador responsável por obter a diferença entre dois conjuntos de valores: o operador de diferença ( $-$ ). Segundo Siebra (2010), a diferença consiste na operação entre duas relações de “união compatível”, gerando como resultado o conjunto de todas as linhas que estão na primeira relação e não aparecem na segunda.

Já para Ramakrishnan e Gehrke (2008), a diferença de conjunto  $R - M$  retorna uma instância de relação contendo todas as tuplas que ocorrem em  $R$ , mas não em  $M$ . As relações  $R$  e  $M$  devem ser compatíveis à união, e o esquema do resultado é definido de forma idêntica ao esquema de  $R$ .

Na Figura 1, há uma representação do operador de diferença a partir do diagrama de Venn, contribuindo para compreender como se dá a sua aplicação.



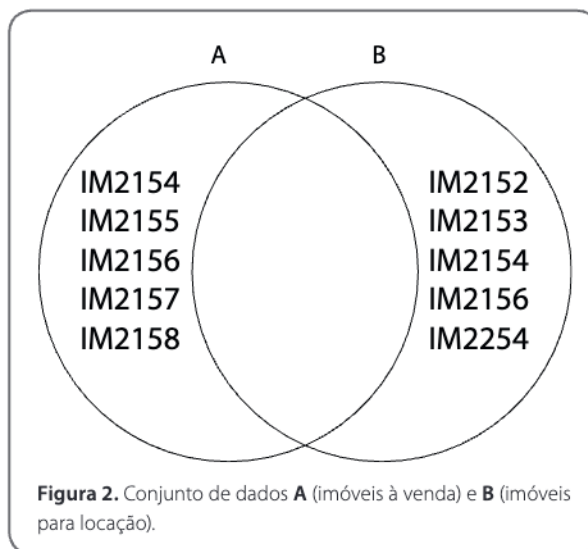
Com base na representação do diagrama de Venn, podemos observar que a aplicação do operador de diferença gerará um novo conjunto que tem apenas os dados existentes no primeiro conjunto sem uma correspondência no segundo conjunto de dados, além do fato de os valores pertencentes apenas ao segundo conjunto de dados não aparecerem no resultado dessa operação.



### Fique atento

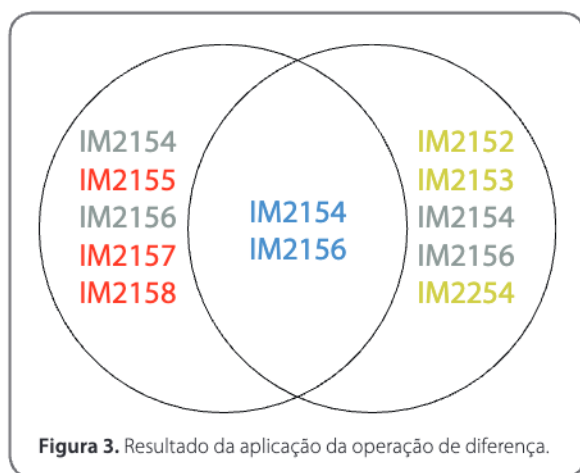
Conhecer os operadores de álgebra relacional possibilita encontrar soluções de consultas em bancos de dados para melhorar o tempo de *performance* e, sobretudo, encontrar exatamente as informações desejadas.

Considere os seguintes conjuntos de dados da Figura 2, em que se tem a relação dos imóveis à venda no conjunto de dados **A** e para locação no conjunto de dados **B**.



**Figura 2.** Conjunto de dados **A** (imóveis à venda) e **B** (imóveis para locação).

O resultado da aplicação do operador de diferença nos dois conjuntos de dados representados avaliará se cada valor dos conjuntos de dados **A** existe no conjunto de dados **B**. O resultado dessa operação será composto por um conjunto de dados de saída, em que este terá os valores existentes apenas no conjunto de dados **A** sem correlação no conjunto de dados **B**; ainda os valores existentes apenas no conjunto de dados **B** não aparecerão no resultado dessa operação (Figura 3).



Como mostrado na Figura 3, a operação de diferença fará com que apenas os elementos ilustrados em **vermelho** apareçam no conjunto de dados de saída. Os valores em **azul** são os valores presentes nos dois conjuntos (**A** e **B**), ou seja, não atendem ao critério da operação de diferença, assim como os valores em **amarelo**, que existem somente no conjunto de dados **B**.

## 2 Diferença aplicada em banco de dados

A partir de agora, você verá como o operador de diferença da álgebra relacional pode ser aplicado contextualmente nas operações de bancos de dados. O resultado obtido em álgebra relacional e na aplicação à base de dados será alcançado da mesma maneira.

Diante disso, pode-se afirmar que a operação de diferença aplicada à álgebra relacional visa a retornar somente as linhas existentes no primeiro conjunto. Para aplicar isso em uma consulta em banco de dados, faz-se necessário o uso de um comando que realize justamente essa operação.

Essa visão é reforçada por Machado (2014), que afirma que uma operação de diferença de tabelas resulta em linhas que estão em uma tabela, mas não em outra. A expressão `tabela1 - tabela2` resulta em uma tabela que contém todas as linhas que estão na `tabela1`, e não na `tabela2`.

Ramakrishnan e Gehrke (2008) apresentam um cenário em que há duas instâncias, intituladas M1 e M2, a partir das quais se aplica o operador de diferença para identificar todos os marinheiros que estão na instância M1 e que não têm uma correspondência em M2 ou que não existem apenas em M2, como mostrado na Figura 4.

id-marin	nome-marin	avaliação	idade
22	Dustin	7	45,5
31	Lubber	8	55,5
58	Rusty	10	35,0

id-marin	nome-marin	avaliação	idade
28	Yuppy	9	35,0
31	Lubber	8	55,5
44	Guppy	5	35,0
58	Rusty	10	35,0

**Figura 4.** Instâncias M1 e M2 de marinheiros.

*Fonte:* Adaptada de Ramakrishnan e Gehrke (2008).

O resultado após essa aplicação de operador está exibido na Figura 5, que retornou apenas uma tupla, já que as tuplas referentes ao `id-marin` de valores 31 e 58 existem na instância M2.

id-marin	nome-marin	avaliação	idade
22	Dustin	7	45

**Figura 5.** Resultado obtido após a aplicação do operador de diferença nas instâncias `M1 - M2`.

*Fonte:* Adaptada de Ramakrishnan e Gehrke (2008).

Mannino (2008) apresenta um exemplo de uma consulta no banco de dados de uma instituição de ensino, cuja proposta consiste em exibir os professores que não são estudantes (mas apenas professores), a fim de mostrar somente as colunas comuns nos resultados. Para obter esse resultado, deve-se trazer as mesmas quantidades de colunas e com os mesmos tipos de dados em duas consultas e aplicar o operador de diferença nesses dois conjuntos.

Parte das tuplas das tabelas `professor` e `aluno` está representada nas Figuras 6 e 7.

CPFProf	NomeProf	SobrenomeProf	CidadeProf	UFProf	DepoProf	Class.Prof	SalarioProf	SupervisorProf	DataAdmProf	CEPProf
098-76-5432	LEONARD	VINCE	SEATTLE	WA	MS	ASSISTENTE	\$35.000	654-32-1098	10-abr-1995	98111-9921
543-21-0987	VICTORIA	EMMANUEL	BOTHELL	WA	MS	CATEDRATICO	\$120.000		15-abr-1996	98011-2242
654-32-1098	LEONARD	FIBON	SEATTLE	WA	MS	ASSOCIADO	\$70.000	543-21-0987	01-mar-1994	98121-0094
789-43-2109	NICKI	MACON	BELLEVUE	WA	FINAN	CATEDRATICO	\$65.000		11-abr-1997	98015-9945
876-54-3210	CHRISTOPHER	COLAN	SEATTLE	WA	MS	ASSISTENTE	\$40.000	654-32-1098	01-mar-1999	98114-1332
987-65-4321	JULIA	MILLS	SEATTLE	WA	FINAN	ASSOCIADO	\$75.000	765-43-2109	15-mar-2000	98114-9954

**Figura 6.** Amostra da tabela `professor`.

Fonte: Adaptada de Mannino (2008).

Vale ressaltar que a tabela `professor` apresenta uma quantidade de atributos superior à da tabela `aluno`, como visto na Figura 7.

CPFAluno	NomeAluno	Sobrenome	Cidade	UF	CEP	Especializacao	Turma	MediaAluno
123-45-6789	HOMER	WELLS	SEATTLE	WA	98121-1111	SI	FR	3,00
124-56-7890	BOB	NORBERT	BOTHELL	WA	98011-2121	FINAN	JR	2,70
234-56-7890	CANDY	KENDALL	TACOMA	WA	99042-3321	CONTB	JR	3,50
345-67-8901	WALLY	KENDALL	SEATTLE	WA	98123-1141	SI	SR	2,80
456-78-9012	JOE	ESTRADA	SEATTLE	WA	98121-2333	FINAN	SR	3,20
567-89-0123	MARIAH	DODGE	SEATTLE	WA	98114-0021	SI	JR	3,60
678-90-1234	TESS	DODGE	REDMOND	WA	98116-2344	CONTB	SO	3,30
789-01-2345	ROBERT	MORALES	SEATTLE	WA	98121-2212	FINAN	JR	2,50
876-54-3210	CRIS	COLAN	SEATTLE	WA	98114-1332	SI	SR	4,00
890-12-3456	LUKE	BRAZZI	SEATTLE	WA	98116-0021	SI	SR	2,20
901-23-4567	WILLIAM	PILGRIM	BOTHELL	WA	98113-1885	SI	SO	3,80

**Figura 7.** Amostra da tabela `aluno`.

Fonte: Adaptada de Mannino (2008).

O resultado da operação de diferença entre as tabelas `professor` e `aluno` exibirá somente as tuplas presentes na tabela `professor` e que não estejam na tabela `aluno`. Novamente, é preciso ressaltar que, para que isso



aconteça, as duas consultas devem trazer as mesmas quantidades de atributos e tipos de dados.

Para alcançar esse resultado, foi necessário, inicialmente, selecionar apenas a mesma lista de atributos das duas tabelas considerando o respectivo atributo e o seu próprio tipo de dados.

Como se pode perceber, foram selecionados os atributos CPF, nome, sobrenome, cidade e UF das duas tabelas, ou seja, geram-se novos conjuntos de dados a partir das duas tabelas e, em seguida, aplica-se o operador de diferença. Veja a ilustração desses “conjuntos de dados” nas Figuras 8 e 9.

CPFProf	NomeProf	SobrenomeProf	CidadeProf	UFProf
098-76-5432	LEONARD	VINCE	SEATTLE	WA
543-21-0987	VICTORIA	EMMANUEL	BOTHELL	WA
654-32-1098	LEONARD	FIBON	SEATTLE	WA
765-43-2109	NICKI	MACON	BELLEVUE	WA
876-54-3210	CRISTOPHER	COLAN	SEATTLE	WA
987-65-4321	JULIA	MILLS	SEATTLE	WA

**Figura 8.** Novo conjunto de dados da tabela professor.

Fonte: Adaptada de Mannino (2008).

CPFAluno	NomeAluno	Sobrenome	Cidade	UF
123-45-6789	HOMER	WELLS	SEATTLE	WA
124-56-7890	BOB	NORBERT	BOTHELL	WA
234-56-7890	CANDY	KENDALL	TACOMA	WA
345-67-8901	WALLY	KENDALL	SEATTLE	WA
456-78-9012	JOE	ESTRADA	SEATTLE	WA
567-89-0123	MARIAH	DODGE	SEATTLE	WA
678-90-1234	TESS	DODGE	REDMOND	WA
789-01-2345	ROBERT	MORALES	SEATTLE	WA
876-54-3210	CRIS	COLAN	SEATTLE	WA
890-12-3456	LUKE	BRAZZI	SEATTLE	WA
901-23-4567	WILLIAM	PILGRIM	BOTHELL	WA

**Figura 9.** Novo conjunto de dados da tabela aluno.

Fonte: Adaptada de Mannino (2008).



Na Figura 10, exibimos o resultado dessa operação, em que se pode perceber que apenas uma linha da tabela `professor` não consta após a aplicação do operador de diferença nas duas tabelas citadas.

CPF	Nome	Sobrenome	Cidade	UF
098765432	LEONARD	VINCE	SEATTLE	WA
543210987	VICTORIA	EMMANUEL	BOTHELL	WA
654321098	LEONARD	FIBON	SEATTLE	WA
765432109	NICKI	MACON	BELLEVUE	WA
987654321	JULIA	MILLS	SEATTLE	WA

**Figura 10.** Resultado da operação de diferença.

*Fonte:* Adaptada de Mannino (2008).



### Saiba mais

Lembre-se de que, para aplicar o operador de diferença, você precisará compreender o cenário em que está atuando e realizar, se necessário, adaptações para alcançar o resultado desejado.

## 3 Diferença em SQL

O operador de diferença de álgebra relacional, quando usado em consultas em Structured Query Language (SQL), apresenta os mesmos conceito e aplicação.

A operação de diferença aplicada em álgebra relacional tem o objetivo de excluir dados existentes em dois conjuntos de valores.

No PostgreSQL, o Sistema Gerenciador de Banco de Dados (SGBD) que está sendo usado para construir as atividades, não existe o operador `MINUS`, como praticado em outros SGBD (p. ex., Oracle). Contudo, para realizar a operação de diferença no PostgreSQL, deve-se fazer uso do comando `EXCEPT`.

A sintaxe do comando `EXCEPT`, cuja finalidade visa a realizar justamente a operação de diferença entre dois conjuntos de dados, é dada por `Query EXCEPT [ALL] Query`, onde `query` refere-se a duas consultas na base de dados que retornem exatamente as mesmas quantidades de colunas e com os respectivos tipos de dados de cada uma delas.

Para entender a aplicação do operador de diferença, deve-se inicialmente compreender o objetivo de uso do comando `SELECT`: consultar as informações pertinentes de uma tabela. De acordo com Barboza e Freitas (2018), diferentes instruções SQL têm, em si mesmas, complexidades únicas; assim, uma consulta, por exemplo, não necessariamente trará resultados semelhantes em contextos distintos.

A sintaxe básica de uma consulta com a aplicação do operador `EXCEPT` é apresentada a seguir.

```
SELECT
    T1.atributo1
FROM
    tabela1 T1
EXCEPT
SELECT
    T2.atributo1
FROM
    tabela2 T2
```

Dessa forma, verificaremos quais são as tuplas referentes ao `atributo1` da `tabela1` que não constam nas tuplas do `atributo1` da `tabela2`. O resultado dessas tuplas mostrará o conjunto de tuplas que estão apenas a `tabela1` e que não estejam tanto nas duas tabelas quanto apenas na `tabela2`.

Deve-se também dizer que esses atributos podem ser comparados com mais de um atributo, ou seja, no exemplo apresentado, a aplicação foi feita com apenas com o `atributo1` das duas tabelas, mas pode haver aplicações, como a vista a seguir, nas quais são usados mais de um atributo das duas tabelas para aplicar o operador de diferença.

Para obter o resultado da tabela citada no exemplo anterior (professor e aluno), deve-se fazer uso da seguinte consulta SQL (RAMAKRISHNAN; GEHRKE, 2008):

```
SELECT CPFProf AS CPF
, NomeProf AS Nome
, SobrenomeProf AS Sobrenome
, CidadeProf AS Cidade
, UFProf AS UF
FROM professor
EXCEPT
SELECT CPFAluno AS CPF
, NomeAluno AS Nome
, SobrenomeAluno AS Sobrenome
, CidadeAluno AS Cidade
, UFAluno AS UF
FROM aluno
```

Ainda, as consultas que utilizarem EXCEPT podem ser reescritas de maneira similar usando NOT IN (RAMAKRISHNAN; GEHRKE, 2008).

Para alcançar o mesmo resultado sem usar o operador de diferença (EXCEPT), a consulta deve fazer uso de um atributo “chave” (único) a fim de comparar os valores das tuplas. A consulta reescrita com NOT IN é exibida a seguir (RAMAKRISHNAN; GEHRKE, 2008):

```
SELECT CPFProf AS CPF
, NomeProf AS Nome
, SobrenomeProf AS Sobrenome
, CidadeProf AS Cidade
, UFProf AS UF
FROM professor
WHERE
    CPFProf NOT IN
        (SELECT CPFAluno AS CPF
        FROM aluno)
```

Como o CPF é um valor único, o resultado dessa consulta com a condição `NOT IN` é o mesmo da consulta feita usando o operador `EXCEPT`.

Cabe ressaltar que, ao usar o `NOT IN` no lugar do operador `INTERSECT`, a consulta fica restrita a fazer uso de um `NOT IN` para cada condição necessária. Por isso, quando da necessidade de realizar a diferença entre dois ou mais conjuntos de dados, sugere-se o uso do operador `EXCEPT`.

Outra questão a se considerar reside no fato de que, ao usar o `NOT IN`, até mesmo as tuplas duplicadas da primeira tabela serão retornadas, o que, por padrão, não acontece ao usar o `EXCEPT`, já que, nesse caso, as tuplas duplicadas não aparecem no resultado da aplicação do operador de diferença.

Para obter o mesmo resultado usando o operador de diferença, deve-se fazer uso da cláusula opcional `ALL` do comando `EXCEPT`. Dessa maneira, até mesmo as tuplas duplicadas aparecerão no resultado da consulta.

Além disso, pode-se combinar nessa consulta o comando `WHERE` para aplicar condições no resultado da operação. Assim, por exemplo, pode-se filtrar o resultado para trazer apenas os residentes em Seattle:

```
SELECT CPFProf AS CPF
, NomeProf AS Nome
, SobrenomeProf AS Sobrenome
, CidadeProf AS Cidade
, UFProf AS UF
FROM professor
WHERE Cidade = 'SEATTLE'
EXCEPT
SELECT CPFAluno AS CPF
, NomeAluno AS Nome
, SobrenomeAluno AS Sobrenome
, CidadeAluno AS Cidade
, UFAluno AS UF
```

Com a aplicação do comando `WHERE`, conforme exemplificado, alcança-se o resultado apresentado na Figura 11, na qual, como podemos observar, são exibidas apenas as tuplas de que a cidade é igual a Seattle.

CPF	Nome	Sobrenome	Cidade	UF
098765432	LEONARD	VINCE	SEATTLE	WA
654321098	LEONARD	FIBON	SEATTLE	WA
987654321	JULIA	MILLS	SEATTLE	WA

**Figura 11.** Resultado da consulta com `WHERE`.

*Fonte:* Adaptada de Mannino (2008).

O modo como a consulta foi elaborada (com a condição no `SELECT` da `tabela1`) filtra a correspondência apenas no resultado das tuplas retornadas da `tabela1`. Dessa maneira, alcançou-se o resultado de exibir apenas os moradores de Seattle, como exibido na Figura 11.

Outra maneira de aplicar a condição de moradores de Seattle consiste em empregar o filtro no resultado da consulta. Assim, a própria consulta será tratada como um conjunto de dados e o filtro aplicado sobre ele.

```
SELECT * FROM
    (SELECT CPFProf AS CPF
      , NomeProf AS Nome
      , SobrenomeProf AS Sobrenome
      , CidadeProf AS Cidade
      , UFProf AS UF
    FROM professor
    EXCEPT
    SELECT CPFAluno AS CPF
      , NomeAluno AS Nome
      , SobrenomeAluno AS Sobrenome
      , CidadeAluno AS Cidade
      , UFAluno AS UF)
AS PESSOA P
WHERE
    P.Cidade = 'SEATTLE';
```

Com essa consulta, o resultado alcançado será o mesmo da consulta anterior, já que o operador de diferença retorna todas as tuplas da primeira tabela, exceto as que existem também na segunda tabela, ou seja, ao aplicar o filtro do modo como se fez na primeira consulta, os resultados serão somente da cidade de Seattle.



### Saiba mais

Na última consulta exibida, podemos perceber o uso do \* (asterisco) para trazer todos os atributos do conjunto de valores provenientes da outra consulta. No entanto, esta não compreende uma boa prática em consultas em bancos de dados. Deve-se sempre informar na lista de atributos **apenas** os atributos que são usados na consulta, uma abordagem que evita o tráfego de informações que não serão usadas, o que tornará a consulta mais rápida.

Nessa segunda *query*, a condição do WHERE está sendo aplicada no resultado do operador de diferença; logo, ele pode ser útil para aplicar condições que precisam ser condizentes com o resultado da operação de diferença.

```
SELECT CPFProf AS CPF
, NomeProf AS Nome
, SobrenomeProf AS Sobrenome
, CidadeProf AS Cidade
, UFProf AS UF
FROM professor
EXCEPT
SELECT CPFAluno AS CPF
, NomeAluno AS Nome
, SobrenomeAluno AS Sobrenome
, CidadeAluno AS Cidade
, UFAluno AS UF
ORDER BY 2, 3 DESC, 4;
```



A inclusão do `ORDER BY` nessa consulta fará com que o resultado do comando seja apresentado em ordem crescente pelo atributo `Nome`, decrescente pelo sobrenome e crescente pela `Cidade`. Cabe dizer também que essa ordenação está sendo realizada após a aplicação da diferença entre os conjuntos, motivo pelo qual a ordenação está sendo feita por indicadores numéricos de cada coluna.

Em suma, a aplicação do operador `EXCEPT` permite obter as informações provenientes apenas no conjunto de dados e que não tenham um par em outro conjunto de dados. Novamente, salientamos o impacto positivo de conhecer sobre a álgebra relacional. Por fim, pode-se combinar o uso desse operador juntamente a outras especificações provenientes de consulta em banco de dados.



## Referências

BARBOZA, F. F. M.; FREITAS P. H. C. *Modelagem e desenvolvimento de banco de dados*. Porto Alegre: Sagah, 2018.

MACHADO, F. N. R. *Banco de dados: projeto e implementação*. 3. ed. São Paulo: Érica, 2014.

MANNINO, M. V. *Projeto, desenvolvimento de aplicações e administração de banco de dados*. 3. ed. Porto Alegre: AMGH, 2008.

RAMAKRISHNAN, R.; GEHRKE, J. *Sistemas de gerenciamento de bancos de dados*. 3. ed. Porto Alegre: AMGH, 2008.

SIEBRA, S. A. *Banco de dados*. Recife: Universidade Federal Rural de Pernambuco, 2010.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

