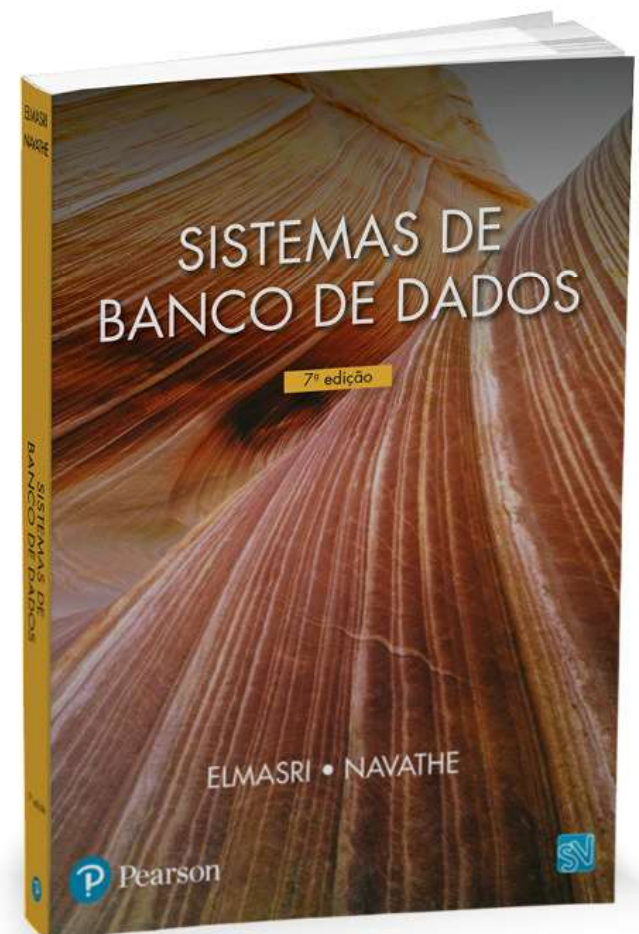


SQL básica



Definições e tipos de dados em SQL

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- A SQL usa os termos **tabela**, **linha** e **coluna** para os termos do modelo relacional formal relação, tupla e atributo, respectivamente.
- O principal comando SQL para a definição de dados é o CREATE, que pode ser usado para criar esquemas, tabelas (relações), tipos e domínios, bem como outras construções, como views, assertions e triggers.
- Um **esquema SQL** é identificado por um nome de esquema, e inclui um identificador de autorização para indicar o usuário ou a conta proprietária do esquema, bem como descritores para cada elemento.
- Um esquema é criado por meio da instrução CREATE SCHEMA, que pode incluir todas as definições dos elementos do esquema.

Definições e tipos de dados em SQL

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Observe que cada instrução em SQL termina com um ponto e vírgula.

```
CREATE SCHEMA EMPRESA AUTHORIZATION 'Jsilva';
```

- Em geral, nem todos os usuários estão autorizados a criar esquemas e elementos do esquema.
- A SQL usa o conceito de um **catálogo** — uma coleção nomeada de esquemas.
- Um catálogo sempre contém um esquema especial, chamado INFORMATION_SCHEMA, que provê informações sobre todos os esquemas no catálogo e todos os descritores de elemento nesses esquemas.

O comando CREATE TABLE em SQL

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- O comando CREATE TABLE é usado para especificar uma nova relação.
- Os atributos são especificados primeiro, e cada um deles recebe um nome, um tipo de dado para especificar seu domínio de valores e quaisquer restrições de atributo, como NOT NULL.
- As restrições de chave, integridade de entidade e integridade referencial podem ser especificadas na instrução CREATE TABLE, depois que os atributos forem declarados, ou acrescentadas depois, usando o comando ALTER TABLE.
- Em geral, o esquema SQL em que as relações são declaradas é especificado implicitamente no ambiente em que as instruções CREATE TABLE são executadas.

O comando CREATE TABLE em SQL

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Por exemplo, escrevendo

```
CREATE TABLE EMPRESA.FUNCIONARIO
```

- em vez de

```
CREATE TABLE FUNCIONARIO
```

- As relações declaradas por meio das instruções CREATE TABLE são chamadas de **tabelas de base** (ou **relações de base**).
- Isso significa que a relação e suas tuplas são realmente criadas e armazenadas como um arquivo pelo SGBD.
- As relações de base são distintas das **relações virtuais**, criadas por meio da instrução CREATE VIEW.

O comando CREATE TABLE em SQL

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Os tipos de dados básicos disponíveis para atributos são:
 1. numérico,
 2. cadeia ou sequência de caracteres,
 3. cadeia ou sequência de bits,
 4. booleano,
 5. data e hora.
- É possível especificar o tipo de dado de cada atributo diretamente; como alternativa, um domínio pode ser declarado e seu nome, usado com a especificação de atributo.

Especificando restrições de atributo e *defaults* de atributo

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Como a SQL permite NULLs como valores de atributo, uma restrição NOT NULL pode ser especificada se o valor NULL não for permitido para determinado atributo.
- Também é possível definir um valor default (ou padrão) para um atributo anexando a cláusula DEFAULT <valor> a uma definição de atributo.
- Outro tipo de restrição pode limitar valores de atributo ou domínio usando a cláusula CHECK após uma definição de atributo ou domínio.
- A figura a seguir ilustra um exemplo de especificação de um gerente default para um novo departamento e um departamento default para um novo funcionário.

Especificando restrições de atributo e *defaults* de atributo

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

```
CREATE TABLE FUNCIONARIO
(
    ...,
    Numero_departamento INT NOT NULL DEFAULT 1,
    CONSTRAINT CHAVEPRIMFUNCIONARIO
        PRIMARY KEY (Cpf),
    CONSTRAINT CHAVEESTRFUNC_SUPERVISOR
        FOREIGN KEY (Cpf_supervisor) REFERENCES FUNCIONARIO(Cpf)
            ON DELETE SET NULL ON UPDATE CASCADE,
    CONSTRAINT CHAVEESTRFUNC_DEPARTAMENTO
        FOREIGN KEY(Numero_departamento) REFERENCES DEPARTAMENTO(Numero_departamento)
            ON DELETE SET DEFAULT ON UPDATE CASCADE;
CREATE TABLE DEPARTAMENTO
(
    ...,
    Cpf_gerente CHAR(11) NOT NULL DEFAULT '88866555576',
    ...,
    CONSTRAINT CHAVEPRIMDEPARTAMENTO
        PRIMARY KEY(Numero_departamento),
    CONSTRAINT CHAVEUNICADEPARTAMENTO
        UNIQUE (Nome_departamento),
    CONSTRAINT CHAVEESTRDEPARTAMENTO_FUNC
        FOREIGN KEY (Cpf_gerente) REFERENCES FUNCIONARIO(Cpf)
            ON DELETE SET DEFAULT ON UPDATE CASCADE;
CREATE TABLE LOCALIZACOES_DEPARTAMENTO
(
    ...,
    PRIMARY KEY (Numero_departamento, Local),
    FOREIGN KEY (Numero_departamento) REFERENCES
        DEPARTAMENTO(Numero_departamento)
            ON DELETE CASCADE ON UPDATE CASCADE;
```


Especificando restrições de chave e integridade referencial

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- A cláusula **PRIMARY KEY** especifica um ou mais atributos que compõem a chave primária de uma relação.
- Se uma chave primária tiver um único atributo, a cláusula pode acompanhar o atributo diretamente.
- A cláusula **UNIQUE** especifica chaves alternativas (secundárias), também conhecidas como chaves candidatas.
- A integridade referencial é especificada por meio da cláusula **FOREIGN KEY** (chave estrangeira).
- A ação default que a SQL toma para uma violação de integridade é rejeitar a operação de atualização que causará uma violação, o que é conhecido como opção **RESTRICT**.

Especificando restrições de chave e integridade referencial

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Porém, o projetista do esquema pode especificar uma ação alternativa para ser tomada conectando uma cláusula de **ação de disparo referencial** a qualquer restrição de chave estrangeira.
- As opções incluem SET NULL, CASCADE e SET DEFAULT.
- Uma opção deve ser qualificada com ON DELETE ou ON UPDATE.
- A figura anterior também ilustra como uma restrição pode receber um nome de restrição, seguindo a palavra-chave **CONSTRAINT**.
- Os nomes de todas as restrições dentro de um esquema em particular precisam ser exclusivos.

Especificando restrições sobre tuplas usando CHECK

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Restrições baseadas em tupla se aplicam a cada tupla individualmente e são verificadas sempre que uma tupla é inserida ou modificada.
- Por exemplo, suponha que a tabela DEPARTAMENTO tivesse um atributo adicional Data_criacao, que armazena a data em que o departamento foi criado.
- Então, poderíamos acrescentar a seguinte cláusula CHECK ao final da instrução CREATE TABLE para a tabela DEPARTAMENTO para garantir que a data de início de um gerente seja posterior à data de criação do departamento.

CHECK (Data_criacao <= Data_inicio_gerente)

Consultas de recuperação básicas em SQL

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- A SQL tem uma instrução básica para recuperar informações de um banco de dados: a instrução **SELECT**.
- A forma básica do comando SELECT, às vezes chamada de **mapeamento** ou **bloco select-from-where**, é composta pelas três cláusulas SELECT, FROM e WHERE, e tem a seguinte forma:
 1. SELECT <lista atributos>
 2. FROM <lista tabelas>
 3. WHERE <condição>;
- Em SQL, os operadores básicos de comparação lógicos para comparar valores de atributo entre si e com constantes literais são =, <, <=, >, >= e <>.

Consultas de recuperação básicas em SQL

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Em SQL, o mesmo nome pode ser usado para dois (ou mais) atributos, desde que estes estejam em relações diferentes.
- Se isso acontecer, e uma consulta em múltiplas relações se referir a dois ou mais atributos com o mesmo nome, é preciso qualificar o nome do atributo com o nome da relação para evitar ambiguidade.
- Isso é feito prefixando o nome da relação ao nome do atributo e separando os dois por um ponto.
- Nomes de atributo totalmente qualificados podem ser usados por clareza mesmo que não haja ambiguidade nos nomes de atributo.
- Podemos usar o mecanismo de **nomeação** de apelidos em qualquer consulta SQL para especificar variáveis de tupla para cada tabela na cláusula WHERE.

Cláusula WHERE não especificada e uso do asterisco

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- A falta de uma cláusula WHERE indica que não há condição sobre a seleção de tuplas; logo, todas as tuplas da relação especificada na cláusula FROM se qualificam e são selecionadas para o resultado da consulta.
- Se mais de uma relação for especificada na cláusula FROM e não houver uma cláusula WHERE, então o PRODUTO CARTESIANO — todas as combinações de tuplas possíveis — dessas relações será selecionado.
- É extremamente importante especificar cada condição de seleção e junção na cláusula WHERE.
- Se alguma condição desse tipo for esquecida, o resultado poderá ser relações incorretas e muito grandes.

Cláusula WHERE não especificada e uso do asterisco

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Para recuperar todos os valores de atributo das tuplas selecionadas, não precisamos listar os nomes de atributo explicitamente em SQL; basta especificar um asterisco (*), que significa todos os atributos.
- O * também pode ser iniciado pelo nome da relação ou apelido; por exemplo, FUNCIONARIO.* refere-se a todos os atributos da tabela FUNCIONARIO.

```
C1C: SELECT *  
      FROM  FUNCIONARIO  
      WHERE Numero_departamento = 5;
```

```
C1D: SELECT *  
      FROM  FUNCIONARIO, DEPARTAMENTO  
      WHERE Nome_departamento = 'Pesquisa' AND DEPARTAMENTO.Numero_  
            departamento = FUNCIONARIO.Numero_departamento;
```

```
C10A: SELECT *  
      FROM  FUNCIONARIO, DEPARTAMENTO;
```

Tabelas como conjuntos em SQL

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- A SQL não elimina automaticamente tuplas duplicadas nos resultados das consultas, pelos seguintes motivos:
 1. A eliminação de duplicatas é uma operação dispendiosa.
 2. O usuário pode querer ver as tuplas duplicadas no resultado de uma consulta.
 3. Quando uma função agregada é aplicada às tuplas, na maioria dos casos não queremos eliminar duplicatas.
- Se quisermos eliminar tuplas duplicadas do resultado de uma consulta SQL, usamos a palavra-chave **DISTINCT** na cláusula **SELECT**, significando que apenas as tuplas distintas deverão permanecer no resultado.

Tabelas como conjuntos em SQL

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Existem operações de união de conjunto (**UNION**), diferença de conjunto (**EXCEPT**) e interseção de conjunto (**INTERSECT**).
- As relações resultantes dessas operações de conjunto são conjuntos de tuplas; ou seja, tuplas duplicadas são eliminadas do resultado.
- A SQL também possui operações multiconjunto correspondentes, que são acompanhadas da palavra-chave **ALL** (**UNION ALL**, **EXCEPT ALL**, **INTERSECT ALL**).
- Seus resultados são multiconjuntos (duplicatas não são eliminadas).
- Basicamente, cada tupla — seja ela uma duplicata ou não — é considerada uma tupla diferente ao aplicar essas operações.

Combinação de padrão de subcadeias e operadores aritméticos

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Um recurso permite condições de comparação apenas sobre partes de uma cadeia de caracteres, usando o operador de comparação LIKE.
- Por exemplo, considere a consulta a seguir.
- Recuperar todos os funcionários cujo endereço esteja em Belo Horizonte, MG.

```
SELECT Primeiro_nome, Ultimo_nome
```

```
FROM FUNCIONARIO
```

```
WHERE Endereco LIKE '%Belo Horizonte,MG%';
```

Combinação de padrão de subcadeias e operadores aritméticos

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Outro recurso permite o uso de aritmética nas consultas.
- Por exemplo:
- Mostrar os salários resultantes se cada funcionário que trabalha no projeto 'ProdutoX' receber um aumento de 10%.

```
SELECT F.Primeiro_nome, F.Ultimo_nome, 1.1 * F.Salario      AS  
Aumento_salario  
FROM FUNCIONARIO AS F, TRABALHA_EM AS T, PROJETO      AS P  
WHERE  F.Cpf    =  T.Cpf_funcionario  AND  T.Numero_projeto  =  
P.Numero_projeto  
AND P.Nome_projeto = 'ProdutoX';
```

Ordenação dos resultados da consulta

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- A SQL permite que o usuário ordene as tuplas no resultado de uma consulta pelos valores de um ou mais dos atributos que aparecem no resultado da consulta, usando a cláusula ORDER BY. Por exemplo:

```
SELECT D.Nome_departamento, F.Ultimo_nome,      F.Primeiro_nome,
P.Nome_projeto
FROM DEPARTAMENTO AS D, FUNCIONARIO AS F,      TRABALHA_EM AS
T, PROJETO AS P
WHERE      D.Numero_departamento      =      F.Numero_departamento
AND      F.Cpf=      T.Cpf_funcionario      AND      T.Numero_projeto      =
P.Numero_projeto
ORDER BY D.Nome_departamento,      F.Ultimo_nome,
F.Primeiro_nome;
```

O comando INSERT

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- Em sua forma mais simples, INSERT é usado para acrescentar uma única tupla (linha) a uma relação (tabela).
- Uma segunda forma da instrução INSERT permite que o usuário especifique nomes de atributo explícitos que correspondem aos valores fornecidos no comando INSERT.
- Uma variação do comando INSERT inclui várias tuplas em uma relação em conjunto com a criação da relação e sua carga com o resultado de uma consulta.
- A maioria dos SGBDs tem ferramentas de carregamento em massa que permitem que um usuário carregue dados formatados de um arquivo em uma tabela sem ter de escrever um grande número de comandos INSERT.

O comando DELETE

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- O comando DELETE remove tuplas de uma relação.
- Ele inclui uma cláusula WHERE, semelhante à que é usada em uma consulta SQL, para selecionar as tuplas a serem excluídas.
- Dependendo do número de tuplas selecionadas pela condição na cláusula WHERE, zero, uma ou várias tuplas podem ser excluídas por um único comando DELETE.
- Uma cláusula WHERE inexistente especifica que todas as tuplas na relação deverão ser excluídas; porém, a tabela permanece no banco de dados como uma tabela vazia.
- Temos de usar o comando DROP TABLE para remover a definição da tabela.

O comando UPDATE

SISTEMAS DE BANCO DE DADOS

7ª edição

ELMASRI • NAVATHE

- O comando UPDATE é usado para modificar valores de atributo de uma ou mais tuplas selecionadas.
- Assim como no comando DELETE, uma cláusula WHERE no comando UPDATE seleciona as tuplas a serem modificadas em uma única relação.
- A atualização de uma chave primária pode ser propagada para os valores de chave estrangeira das tuplas em outras relações se tal ação de disparo referencial for especificada nas restrições de integridade referencial da DDL.
- Uma cláusula SET adicional no comando UPDATE especifica os atributos a serem modificados e seus novos valores.