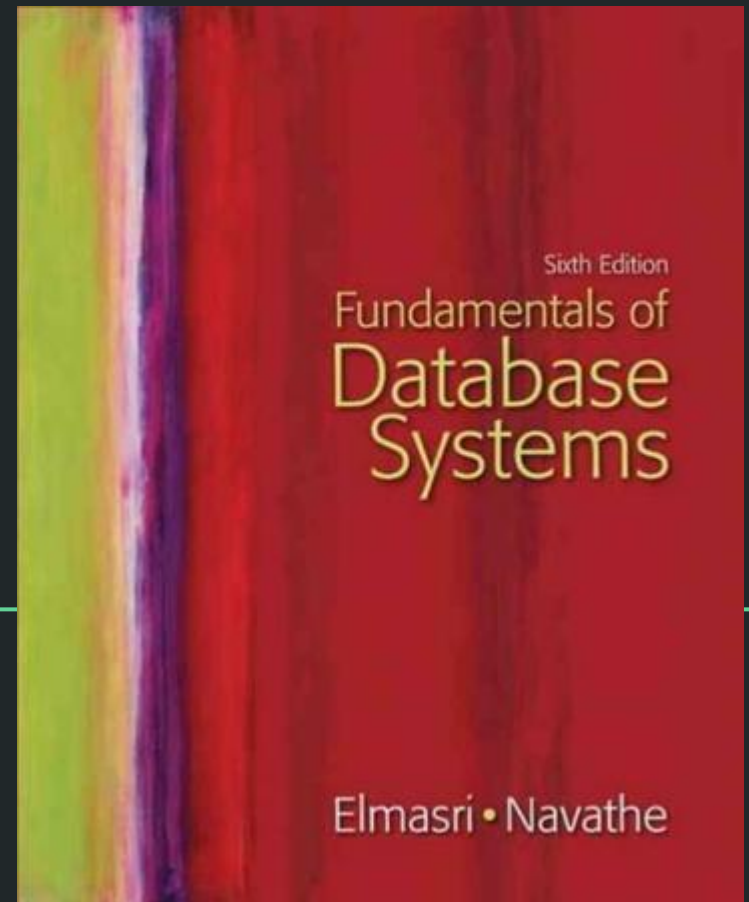


Capítulo 4

SQL básica



Tópicos

- Definições e tipos de dados em SQL
- Especificando restrições em SQL
- Instruções de alteração de esquema em SQL

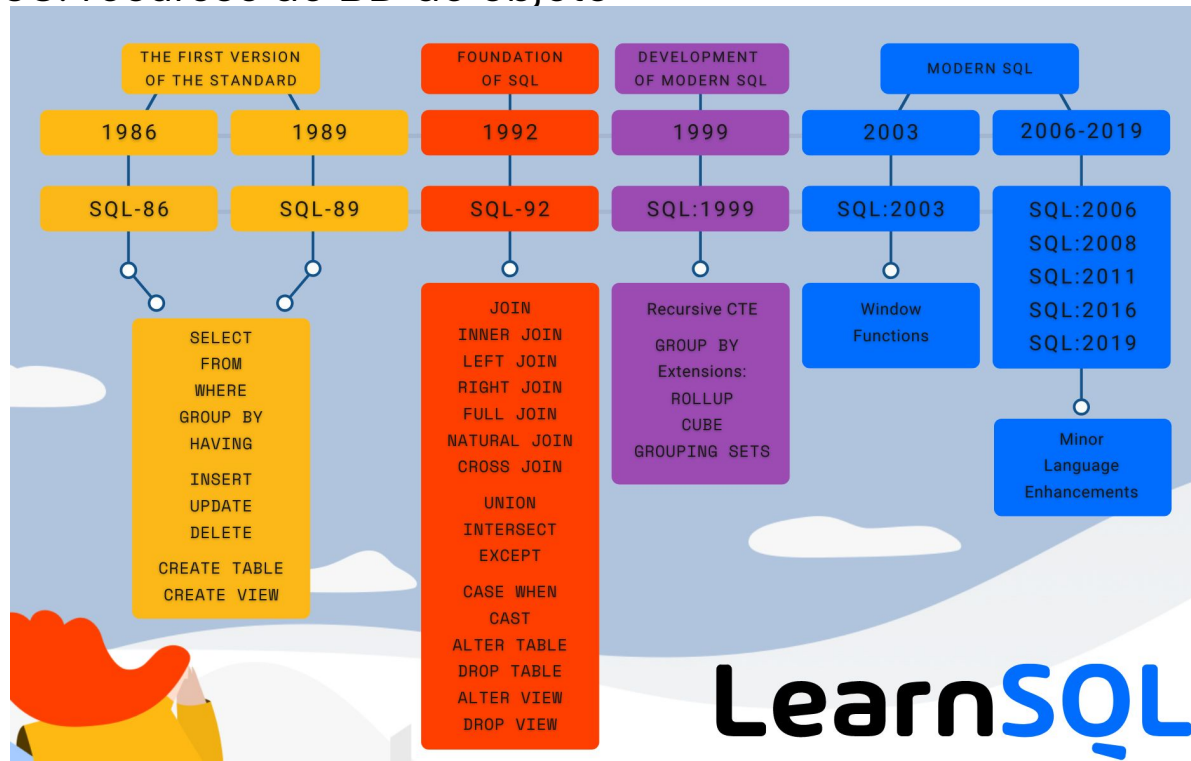
Introdução

- Originalmente, SQL era chamada de SEQUEL e foi criada e implementada na IBM Research para o SYSTEM R.
- Considerada um dos principais motivos para o sucesso dos bancos de dados relacionais comerciais



Introdução

- Utiliza os padrões ANSI e ISO
 - SQL-86 ou SQL1
 - SQL-92 ou SQL2
 - SQL-99 ou SQL3
 - SQL-2003 e SQL-2006: recursos de XML
 - SQL-2008: recursos de BD de objeto



SQL básica



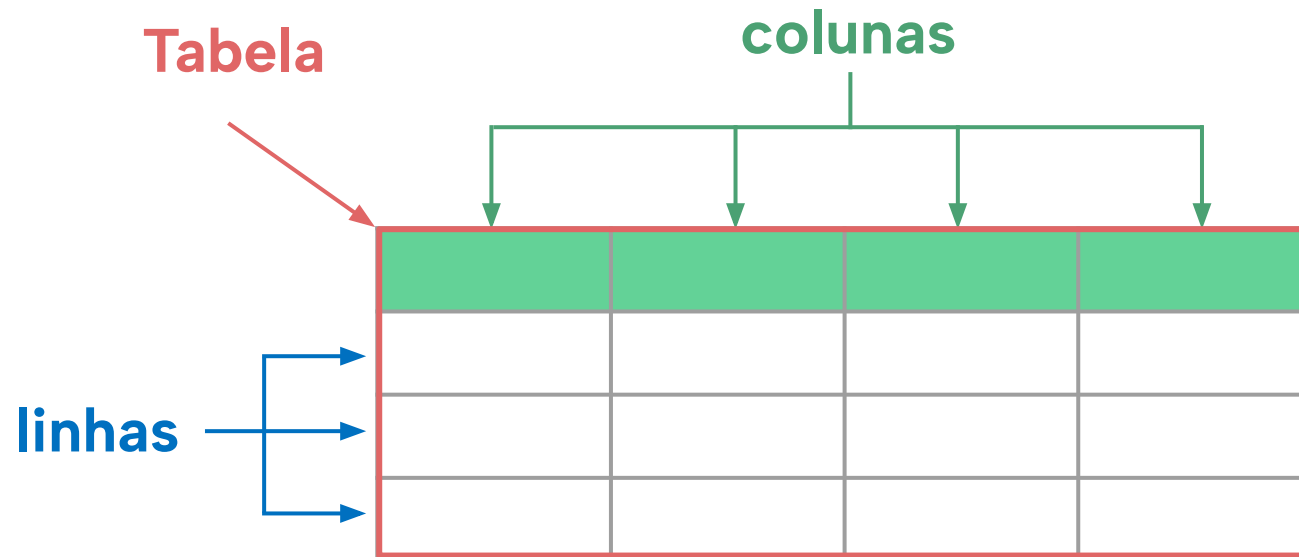
- **SQL – Structured Query Language**
 - **Data Definition Language (DDL)**
 - Instruções para definição de dados
 - Comandos para a definição, a modificação e a remoção de relações, além da criação e da remoção de índices.
 - **Data Manipulation Language (DML)**
 - Instruções para consultas e atualizações
 - Comandos para a consulta, a inserção, a remoção e a modificação de tuplas no banco de dados.
 - **Data Control Language (DCL)**
 - Instruções para controle de dados
 - comandos para controlar o acesso aos dados em um banco de dados.

DDL

Data Definition Language

Definições em SQL

- Terminologia:
 - Tabela, linha e coluna** usados para os termos do modelo relacional *relação*, *tupla* e *atributo*



- Instruções **CREATE**
 - Principal comando SQL para a definição de dados

Definições em SQL

- Terminologia:
 - **Tabela, linha e coluna** usados para os termos do modelo relacional *relação*, *tupla* e *atributo*

No MR	Em SQL
relação	tabela
tupla	linha
atributo	coluna

- Instruções **CREATE**
 - Principal comando SQL para a definição de dados

Conceitos de esquema e catálogo em SQL

Schema no MySQL:

Um schema ou database é um espaço onde se armazenam tabelas e outros objetos

■ Esquema SQL

- Identificado por um **nome de esquema**
- Inclui um **identificador de autorização e descritores** para cada elemento

Um **ID de autorização** é utilizado para duas finalidades: identificação e verificação de autorização.

■ Instrução CREATE SCHEMA

- CREATE SCHEMA EMPRESA;

DDL

- DDL trabalha com três comandos básicos para definição de dados:
 - **CREATE** – Cria uma definição de dados.
 - Ex.: **CREATE TABLE** tab ...
 - **ALTER** – Altera uma definição.
 - Ex.: **ALTER TABLE** tab ADD ...
 - **DROP** – Exclui uma definição.
 - Ex.: **DROP TABLE** tab.

Identificadores

Iniciam com letras (a–z) ou underscore/traço baixo (_).

- Caracteres subsequentes: letras, dígitos (0–9), _

Identificadores e palavras-chave **não são case-sensitive**. Os comandos a seguir são equivalentes:

```
UPDATE MY_TABLE SET A = 5;
```

```
uPDaTE my_TabLE SeT a = 5;
```

Convenção adotada:

```
UPDATE my_table SET a = 5;
```

- Palavras-chave em maiúsculas;
- Identificadores em minúsculas.

O comando CREATE TABLE em SQL

- Especificar uma nova relação
 - Dar um nome
 - Especificar atributos e restrições iniciais, depois **restrições** de chave e **integridade referencial**.

Restrições são regras que limitam os valores que podem ser inseridos, excluídos ou atualizados em uma tabela

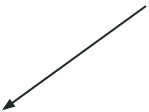
conceito relacionado à chaves estrangeiras onde o valor que é chave estrangeira em uma tabela destino, deve ser chave primária de algum registro na tabela origem.

- Pode especificar o esquema:
 - **CREATE TABLE** funcionario ...

O comando CREATE TABLE em SQL (cont.)

```
CREATE TABLE <nome_da_tabela>(
     $C_1$   $D_1$  [(restrição atributo)],
     $C_2$   $D_2$  [(restrição atributo)],
     $C_n$   $D_n$  [(restrição atributo)],
PRIMARY KEY <lista_de_Colunas>,
FOREIGN KEY <nome_da_coluna> REFERENCES
    <nome_tab_ref>(<nome_da_coluna_ref>));
```

restrições iniciais



cada C_i é uma coluna no esquema da tabela
 D_i é o tipo de dado no domínio da coluna C_i

O comando CREATE TABLE em SQL (cont.)

- Algumas chaves estrangeiras podem causar erros
 - Especificadas por:
 - Referências circulares
 - Ou porque dizem respeito a uma tabela que ainda não foi criada

O comando CREATE TABLE em SQL (cont.)

```
CREATE TABLE FUNCIONARIO
```

```
(Pnome          VARCHAR(15)          NOT NULL,  
  Minicial      CHAR,  
  Unome         VARCHAR(15)          NOT NULL,  
  Cpf           CHAR(11),             NOT NULL,  
  Datanasc      DATE,  
  Endereço      VARCHAR(30),  
  Sexo          CHAR,  
  Salario       DECIMAL(10,2),  
  Cpf_supervisor CHAR(11),             NOT NULL,  
  Dnr           INT
```

```
  PRIMARY KEY (Cpf),
```

```
  FOREIGN KEY (Cpf_supervisor) REFERENCES FUNCIONARIO(Cpf),
```

```
  FOREIGN KEY (Dnr) REFERENCES DEPARTAMENTO(Dnumero) );
```

```
CREATE TABLE DEPARTAMENTO
```

```
( Dnome          VARCHAR(15)          NOT NULL,  
  Dnumero        INT                  NOT NULL,  
  Cpf_gerente     CHAR(11),             NOT NULL,  
  Data_inicio_gerente DATE,
```

```
  PRIMARY KEY (Dnumero),
```

```
  UNIQUE (Dnome),
```

```
  FOREIGN KEY (Cpf_gerente) REFERENCES FUNCIONARIO(Cpf) );
```

O comando ALTER TABLE em SQL

```
ALTER TABLE nome_tabela
ADD [COLUMN] nome_atributo_1 tipo_1 [{RIs}]
    [{, nome_atributo_n tipo_n [{RIs}]]]
|
MODIFY [COLUMN] nome_atributo_1 tipo_1 [{RIs}]
    [{, nome_atributo_n tipo_n [{RIs}]]]
|
DROP COLUMN nome_atributo_1
    [{, nome_atributo_n }]
|
ADD CONSTRAINT nome_RI_1 def_RI_1
    [{, nome_RI_n def_RI_n}]
|
DROP CONSTRAINT nome_RI_1
    [{, nome_RI_n}]
|
[ADD|DROP] [PRIMARY KEY ...|FOREIGN KEY ...]
```


O comando DROP TABLE em SQL

- Exclui uma relação
- Pode excluir a relação:
 - DROP TABLE EMPRESA.FUNCIONARIO ...
ou
 - DROP TABLE FUNCIONARIO ...

Tipos de dados de atributo em SQL

- **Tipos de dados básicos**

- Tipos de dados **numérico**

- Incluem números inteiros: BIGINT, INTEGER ou INT e SMALLINT
 - Números de ponto flutuante (reais): FLOAT ou REAL e DOUBLE PRECISION

- Tipos de dados de **cadeia de caracteres**

- Tamanho fixo: CHAR(*n*) ou CHARACTER(*n*)
 - Tamanho variável: VARCHAR(*n*)

Tipos de dados de atributo em SQL (cont.)

- Tipos de dados de **cadeia de bits**
 - Tamanho fixo: BIT(n)
 - Tamanho variável: BIT VARYING(n)
- Tipo de dado **booleano**
 - Valores TRUE ou FALSE ou NULL
- Tipo de dados **DATE**
 - Componentes são DAY, MONTH e YEAR na forma DD-MM-YYYY.
 - No MySQL a forma é YYYY-MM-DD.
- Tipo de dados **TIME**
 - Componentes são HOUR, MINUTE e SECOND na forma HH:MM:SS

Tipos de dados de atributo em SQL (cont.)

- Tipos de dados adicionais
 - Tipo de dado **Timestamp** (TIMESTAMP)
 - Inclui os campos DATE e TIME
 - Mais um mínimo de seis posições para frações decimais de segundos
 - Qualificador opcional WITH TIME ZONE
 - Tipo de dado INTERVAL
 - Especifica valor relativo que pode ser usado para incrementar ou decrementar um valor absoluto de uma data, hora ou timestamp

Domínios em SQL

- Domínio
 - Nome usado para especificar o tipo de dado de um atributo que é usado várias vezes num esquema.
 - É mais fácil mudar o tipo de dado de um domínio do que diversos atributos.
 - Melhora a legibilidade do esquema.
 - Exemplo:
 - **CREATE DOMAIN TIPO_CPF AS CHAR(11);**

Especificando restrições em SQL

- Restrições básicas:
 - Restrições de chave e integridade referencial
 - Restrições sobre domínios de atributo e NULLs
 - Restrições sobre tuplas individuais dentro de uma relação

Especificando restrições de atributo e defaults de atributo

- NOT NULL
 - Valor NULL não é permitido para um determinado atributo.
 - Ex: ProjNome VARCHAR(15) **NOT NULL**.
- Valor padrão
 - **DEFAULT** <valor>
 - Usado para inicializar o valor de um atributo.
 - Ex: Estado CHAR(2) **DEFAULT** 'MS'.
- cláusula **CHECK**
- Dnumero INT NOT NULL CHECK (Dnumero > 0 AND Dnumero < 21);

Especificando restrições de chave e integridade referencial

- Cláusula PRIMARY KEY
 - Especifica um ou mais atributos que compõem a chave primária de uma relação
 - Dnumero INT PRIMARY KEY;
- Cláusula **UNIQUE**
 - Especifica chaves alternativas (secundárias)
 - Dnome VARCHAR(15) UNIQUE;

```
CREATE TABLE DEPARTAMENTO
    (Dnome VARCHAR(15) NOT NULL,
    Dnumero INT,
    PRIMARY KEY (Dnumero),
    UNIQUE (Dnome)
);
```


Especificando restrições de chave e integridade referencial (cont.)

- Cláusula **FOREIGN KEY**
 - Opção **RESTRICT**: rejeita atualização sobre violação
 - Conectado à cláusula de **ação de disparo referencial**
 - Opções incluem:
 - **SET NULL**
 - **SET DEFAULT**
 - **CASCADE**

CASCADE propagará a alteração quando o pai for alterado.

SET NULL define o valor da coluna como NULL quando uma linha pai desaparece.

RESTRICT faz com que a tentativa DELETE de uma linha pai falhe.

Especificando restrições de atributo e defaults de atributo

```
CREATE TABLE FUNCIONARIO
( ...,
  Dnr          INT          NOT NULL   DEFAULT 1,
  CONSTRAINT CHPFUNC
    PRIMARY KEY (Cpf),
  CONSTRAINT CHESUPERFUNC
    FOREIGN KEY (Cpf_supervisor) REFERENCES FUNCIONARIO(Cpf)
    ON DELETE SET NULL   ON UPDATE CASCADE,
  CONSTRAINT CHEDEPFUNC
    FOREIGN KEY(Dnr) REFERENCES DEPARTAMENTO(Dnumero)
    ON DELETE SET DEFAULT ON UPDATE CASCADE);

CREATE TABLE DEPARTAMENTO
( ...,
  Cpf_gerente  CHAR(11)     NOT NULL   DEFAULT '88866555576',
  ...,
  CONSTRAINT CHPDEP
    PRIMARY KEY(Dnumero),
  CONSTRAINT CHSDEP
    UNIQUE (Dnome),
  CONSTRAINT CHEGERDEP
    FOREIGN KEY (Cpf_gerente) REFERENCES FUNCIONARIO(Cpf)
    ON DELETE SET DEFAULT ON UPDATE CASCADE);

CREATE TABLE LOCALIZACAO_DEP
( ...,
  PRIMARY KEY (Dnumero, Dlocal),
  FOREIGN KEY (Dnumero) REFERENCES DEPARTAMENTO(Dnumero)
    ON DELETE CASCADE   ON UPDATE CASCADE);
```

Restrição de chave primária (**PRIMARY KEY**) na coluna;

Restrição de chave estrangeira (**FOREIGN KEY**) na coluna;

palavra chave **REFERENCES** é usada.

Restrição de unicidade (**UNIQUE**) na coluna.

Figura 4.2

Exemplo ilustrando como os valores de atributo default e as ações disparadas por integridade referencial são especificadas em SQL.

Dando nomes a restrições

- Palavra-chave **CONSTRAINT**
 - Nome de restrição
 - Útil para alterações posteriores

São regras aplicadas ao banco de dados com que permitem que você configure as características de uma determinada coluna:

Exemplos:

- Se uma coluna aceita ou não valores nulos
- Se uma coluna é ou não uma chave primária, ou chave estrangeira

Especificando restrições sobre tuplas usando CHECK

- Cláusula CHECK ao final de uma instrução CREATE TABLE
 - Aplicam a cada tupla individualmente
 - CHECK (Dep_data_criacao <= Data_inicio_gerente);

Resumo

- Comandos para definição de esquemas
 - **create table**
 - define a estrutura da tabela, suas restrições de integridade e cria uma tabela vazia
 - **alter table**
 - modifica a definição de uma tabela (I / E / A atributos; I / E RIs)
 - RIs básicas:
 - atributos chave não podem ser removidos de uma tabela
 - atributos NOT NULL não podem ser inseridos em uma tabela
 - **drop table**
 - remove uma tabela com todas as suas tuplas