SQL Básica

Herysson R. Figueiredo
herysson.figueiredo@ufn.edu.br

Sumário

- História
- Instruções CREATE, ALTER.

História

O nome **SQL** hoje é expandido como ***S**tructured **Q**uery **L**anguage* (Linguagem de Consulta Estruturada). Originalmente, SQL era chamada de **SEQUEL** (**S**tructured **E**nglish **QUE**ry **L**anguage) e foi criada e implementada na **IBM *Research*** como a interface para um sistema de banco de dados relacional experimental, chamado *SYSTEM R. A SQL* agora é a linguagem padrão para SGBDs relacionais comerciais

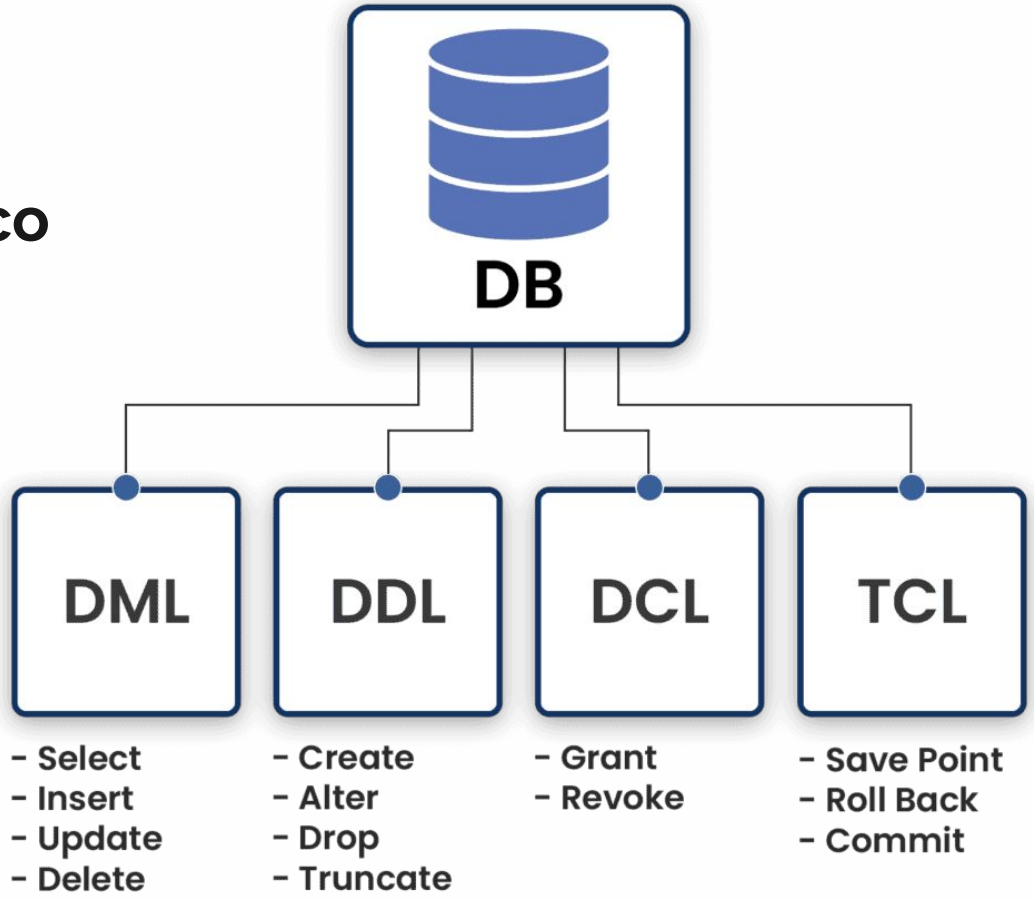
SQL Básico

SQL é uma linguagem de banco de dados abrangente: tem instruções para definição de dados, consultas e atualizações. Logo, ela é uma **DDL - *Data definition language*** (Linguagem de definição de dados) e uma **DML - *Data Manipulation Language*** (Linguagem de manipulação de dados).

DDL - CREATE - DROP - ALTER

DML - SELECT, INSERT, UPDATE E DELETE.

SQL Básico



DCL - Data Control Language | TCL - Transaction Control Language

Revisão

A SQL usa os termos **tabela**, **linha** e **coluna** para os termos do modelo relacional formal ***relação***, ***tupla*** e ***atributo***, respectivamente

Instruções DDL - CREATE, ALTER e DROP

SCHEMA - SQL

Um **esquema SQL** é identificado por um **nome de esquema**, e inclui um **identificador de autorização** para indicar o usuário ou conta proprietário do esquema, bem como **descritores** para cada elemento.

TABLE - SQL

Esses elementos incluem tabelas, restrições, views, domínios e outras construções (como concessões — grants — de autorização) que descrevem o esquema que é criado por meio da instrução:

CREATE SCHEMA ou CREATE DATABASE

CREATE DATABASE DATABASENAME;

CREATE DATABASE EMPRESA;

TABLE - SQL

O comando **CREATE TABLE** é usado para especificar uma nova relação, dando-lhe um nome e especificando seus atributos e restrições iniciais. 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.

TABLE - SQL

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**.

TABLE - SQL

As relações declaradas por meio das instruções CREATE TABLE são chamadas de **tabelas da base** (ou relações da base); isso significa que a relação e suas tuplas são realmente criadas e armazenadas como um arquivo pelo SGBD.

TABLE - SQL

Em SQL, os atributos em uma tabela da base são considerados **ordenados na sequência em que são especificados** no comando CREATE TABLE. No entanto, as linhas (tuplas) não são consideradas ordenadas dentro de uma relação.

TABLE - SQL

```
CREATE TABLE TABLE_NAME (
   Column1 datatype,
   Column2 datatype,
   Column3 datatype,
  ....
);
```

TABLE - SQL

É importante observar que, algumas chaves estrangeiras podem causar erros, pois são especificadas por referências circulares ou porque dizem respeito a uma tabela **que ainda não foi criada.**

TABLE - SQL

Para lidar com esse tipo de problema, essas restrições podem ser omitidas inicialmente do comando **CREATE TABLE**, e depois acrescentadas usando a instrução **ALTER TABLE**.

TABLE - SQL

```
ALTER TABLE FUNCIONARIO
ADD CONSTRAINT Cpf_supervisor
FOREIGN KEY (Cpf_supervisor) REFERENCES FUNCIONARIO(Cpf);
```

Tipos de dados de atributo e domínios em SQL

Os tipos de dados básicos disponíveis para atributos são **numérico**, **cadeia ou sequência de caracteres**, **cadeia ou sequência de bits**, **booleano**, **data** e **hora.**

Numéricos

- números inteiros de vários tamanhos (**INTEGER** ou **INT** e **SMALLINT**)
- números de ponto flutuante (reais) de várias precisões (**FLOAT** ou **REAL** e **DOUBLE PRECISION**).
- números podem ser declarado usando **DECIMAL**(i, j) — ou **DEC**(i, j) ou **NUMERIC**(i, j) — onde i, a precisão, é o número total de dígitos decimais e j, a escala, é o número de dígitos após o ponto decimal.

Cadeia de caracteres

- cadeia de caracteres são de tamanho fixo — **CHAR**(n) ou **CHARACTER**(n), onde n é o número de caracteres
- cadeia de caracteres de tamanho variável — **VARCHAR**(n) ou **CHARVARYING**(n) ou **CHARACTER VARYING**(n)

Cadeia de bits

- cadeia de tamanho fixo n — **BIT**(n)
- cadeia de tamanho variável — **BIT VARYING**(n), onde n é o número máximo de bits.
- cadeia de bits de tamanho variável, chamado **BINARY LARGE OBJECT** ou **BLOB**, também está disponível para especificar colunas que possuem grandes valores binários, como imagens.

Booleano

- os valores tradicionais **TRUE** (verdadeiro) ou **FALSE** (falso).

Em SQL, devido à presença de valores **NULL** (nulos), uma lógica de três valores é utilizada, de modo que um terceiro valor possível para um tipo de dado booleano é **UNKNOWN** (indefinido)

Date

- tipo de dados **DATE** possui dez posições, e seus componentes são **DAY** (dia), **MONTH** (mês) e **YEAR** (ano) na forma **DD-MM-YYYY;**
- O tipo de dado **TIME** (tempo) tem pelo menos oito posições, com os componentes **HOUR** (hora), **MINUTE** (minuto) e **SECOND** (segundo) na forma **HH:MM:SS.**

Timestamp

- tipo de dado timestamp (**TIMESTAMP**) inclui os campos **DATE** e **TIME**, mais um mínimo de seis posições para frações decimais de segundos e um qualificador opcional **WITH TIME ZONE**.
- tipo de dado **INTERVA**L. Este especifica um intervalo — um valor relativo que pode ser usado para incrementar ou decrementar um valor absoluto de uma data, hora ou timestamp.

Especificação de domínio

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 Por exemplo, podemos criar um domínio TIPO_CPF com a seguinte instrução:

```
CREATE DOMAIN TIPO_CPF AS CHAR(11);
```

Podemos usar TIPO_CPF no lugar de CHAR(11) .

Especificando restrições em SQL

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

Nome VARCHAR(15) **NOT NULL**,

Especificando restrições em SQL

Isso sempre é especificado de maneira implícita para os atributos que fazem parte da **chave primária** de cada relação, mas pode ser especificado para quaisquer outros atributos cujos valores **não podem ser NULL.**

Especificando restrições em SQL

Também é possível definir um valor padrão para um atributo anexando a cláusula **DEFAULT** a uma definição de atributo. O valor padrão está incluído em qualquer nova tupla se um valor explícito não for fornecido para esse atributo

Dptnumero INT NOT NULL **DEFAULT** 1,

Especificando restrições em SQL

Outro tipo de restrição pode limitar valores de atributo ou domínio usando a cláusula **CHECK** (verificação) após uma definição de atributo ou domínio.

idade INT NOT NULL CHECK (idade > 0 AND idade < 95),

Especificando restrições de chave e integridade referencial

Como chaves e restrições de integridade referencial são muito importantes, existem cláusulas especiais dentro da instrução CREATE TABLE para especificá-las

Especificando restrições de chave e integridade referencial

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.

```
Id INT PRIMARY KEY,
```

Especificando restrições de chave e integridade referencial

A cláusula **UNIQUE** específica chaves alternativas (secundárias).

Email VARCHAR(50) UNIQUE;

Especificando restrições de chave e integridade referencial

A integridade referencial é especificada por meio da cláusula **FOREIGN KEY** (chave estrangeira). Uma restrição de integridade referencial pode ser violada quando tuplas são inseridas ou excluídas, ou quando um valor de atributo de chave estrangeira ou chave primária é modificado.

Especificando restrições de chave e integridade referencial

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

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

FOREIGN KEY (Cpf_supervisor) REFERENCES FUNCIONARIO(Cpf)

ON DELETE **SET NULL** ON UPDATE **CASCADE**,

Dando nomes a restrições

Uma restrição pode receber um nome de restrição, seguindo a palavra-chave CONSTRAINT.

```
CONSTRAINT CHPFUNC PRIMARY KEY (Cpf),
```

Especificando restrições sobre tuplas usando CHECK

Além das restrições de chave e integridade referencial, que são especificadas por palavras-chave especiais, outras restrições de tabela podem ser especificadas por meio de cláusula adicional **CHECK** ao final de uma instrução CREATE TABLE.

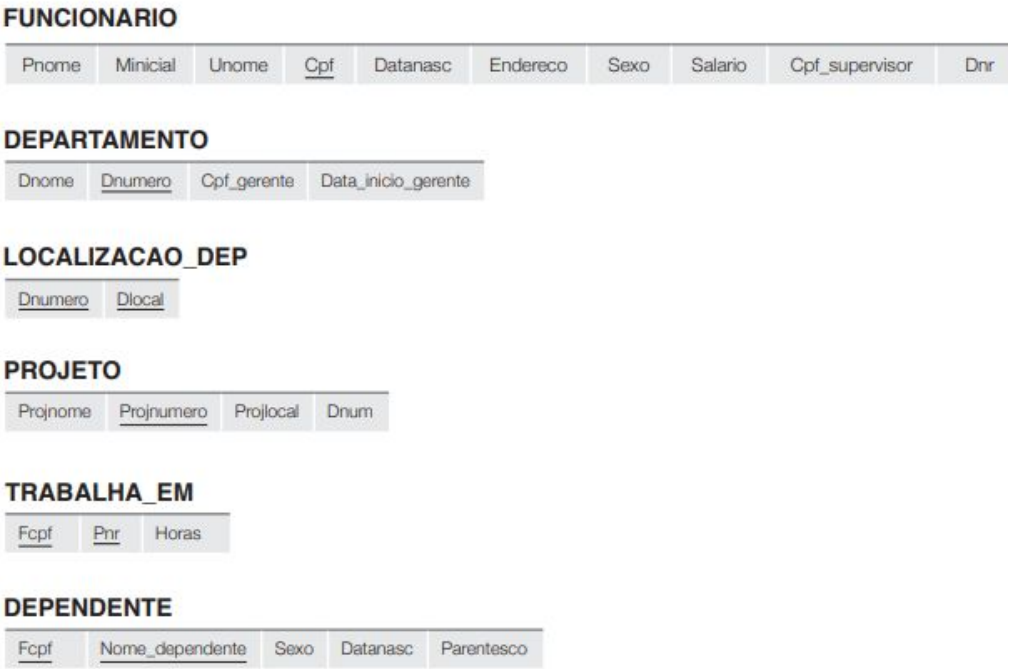
Especificando restrições sobre tuplas usando CHECK

Estas podem ser chamadas de **restrições baseadas em tupla**, pois se aplicam a cada tupla individualmente e são verificadas sempre que uma tupla é inserida ou modificada.

```
CHECK (Dep_data_criacao <= Data_inicio_gerente),
```

Prática:

Crie a data base EMPRESA representada pela imagem ao lado.



Referência Bibliográfica

ELMASRI, Ramez; NAVATHE, Shamkant B.. Sistemas de banco de dados, 7ª ed., 2018

PUGA, Sandra; FRANÇA, Edson e GOYA, Milton. Banco de dados: Implementação em SQL, PL/SQL e Oracle 11g, 2013.

W3SCHOOL, MySQL Database, https://www.w3schools.com/mysql/ acessado em 10/02/2023;