



Instituto de Tecnologia Ciência da Computação

Banco de Dados

Linguagem – SQL: DDL



INTRODUÇÃO

- ❑ **SQL** (Structured Query Language), ou Linguagem de Consulta Estruturada é a linguagem de consulta padrão para banco de dados relacional.
- ❑ usada para definir tabelas, operações (consulta, inserção, atualização, remoção de dados) e definições de regras de integridade no bancos de dados
- ❑ SQL foi desenvolvida pela IBM
 - ❑ A SQL foi inicialmente foi chamada de SEQUEL
 - ❑ Padrões propostos: ANSI-SQL(SQL-89), SQL-92 e padrões mais recentes: SQL:1999 e SQL:2003 (características relacionadas ao XML).





INTRODUÇÃO

- ❑ SQL possui os seguintes comandos:
 - ❑ Linguagem de Definição de Dados (DDL): fornece comandos para definições de esquemas de relação, criação/remoção de tabelas, criação de índices e modificação de esquemas.
 - ❑ Os comandos SQL para definição de dados são:
 - ❑ CREATE
 - ❑ DROP
 - ❑ ALTER
 - ❑ SQL ainda possui comandos DML(manipulação), DQL (consulta), DCL(controle) e DTL(transação)



SQL – comandos DDL

❑ CREATE TABLE

- ✓ especifica *uma nova tabela (relação)*, dando o seu nome e definindo colunas(atributos), cada uma com seu nome, tipo de dados e restrições

Sintaxe:

```
CREATE TABLE nome_da_tabela (
    coluna + tipo_de_dados + constraints(opcional)
    coluna + tipo_de_dados + constraints(opcional)
    ...
)
```

❑ Sintaxe conforme SGBD - Postgresql

- ❑ <https://www.postgresql.org/docs/12/sql-createtable.html>





SQL – comandos DDL

- ❑ As definições das colunas têm o seguinte formato:
 - ❑ *coluna; tipo de dados; [constraints]*
 - ❑ *[NOT NULL][UNIQUE][DEFAULT valor]*

Onde:

- ❑ **Coluna:** nome do atributo que está sendo definido
- ❑ **Tipo:** domínio do atributo (tipo de dados)
- ❑ **NOT NULL:** expressa que o atributo não pode ser nulo
- ❑ **UNIQUE:** indica que o atributo tem valor único na tabela.
Qualquer tentativa de se introduzir uma linha na tabela contendo um valor igual ao do atributo será rejeitada. Serve para indicar chaves secundárias
- ❑ **DEFAULT:** indica um valor default para a coluna



SQL – comandos DDL

Constraints (Restrições de Integridade e de domínio):

- ❑ **Integridade de Chave:**
 - ❑ PRIMARY KEY(atributos_chave – chave primária)
- ❑ **Integridade Referencial** (chave estrangeira):
 - ❑ FOREIGN KEY (atributos) REFERENCES tabela_base(atributos)
- ❑ **Restrição de Integridade** - semântica:
 - ❑ CHECK(condição)



Exemplo de CREATE TABLE:

```
CREATE TABLE empregado (
    matricula      INT          NOT NULL,
    nome           VARCHAR(15)   NOT NULL,
    dataNasc       DATE,
    sexo            CHAR(1),
    salario         NUMERIC(6,2) CHECK (salario >= 0),
    depto          INT          NOT NULL,
    PRIMARY KEY(matricula),
    FOREIGN KEY (depto) REFERENCES departamento(codDep)
)
```

- Chave primária, pode ser declarada na criação do atributo
- Tipo de dados:
 - <https://www.postgresql.org/docs/12/datatype.html>





- Declaração de chaves Estrangeiras:
 - Como criar tabelas que as chaves dependem umas das outras ?

Exemplo:

```
CREATE TABLE Galinha (
    galinha_id     INT      PRIMARY KEY,
    ovo_id         INT      REFERENCES ovo(ovo_id)
);

CREATE TABLE Ovo(
    ovo_id        INT      PRIMARY KEY,
    galinha_id    INT      REFERENCES
                           Galinha(galinha_id)
);
```





Solução para tabelas relacionadas

- Criar tabelas sem referência... são criadas depois

```
CREATE TABLE Galinha(  
    galinha_id      INT      PRIMARY KEY,  
    ovo_id          INT  
)
```

```
CREATE TABLE Ovo(  
    ovo_id          INT      PRIMARY KEY,  
    galinha_id      INT  
)
```

```
ALTER TABLE Galinha ADD CONSTRAINT fk_Galinha  
    FOREIGN KEY (ovo_id) REFERENCES Ovo(ovo_id);
```

```
ALTER TABLE Ovo ADD CONSTRAINT fk_ovo  
    FOREIGN KEY (galinha_id) REFERENCES Galinha(galinha_id);
```



SQL - ALTER TABLE

- Comando SQL usado para alterar a estrutura de uma tabela, para: acrescentar, alterar, retirar campos; e alterar nomes, tipo de dados das colunas e a integridade referencial definidas em uma determinada tabela

Opções:

- DROP** - realiza a retirada da coluna ou constraint(*not null*) especificada na estrutura da tabela.
- ADD** - realiza a inclusão de nova coluna ou constraint
- Atualização** – usado para alterar nome ou campo

- Pesquisa do comando ALTER no PostgreSQL:

- Documentação: <https://www.postgresql.org/docs/12/sql-altertable.html>





Exemplo: ALTER TABLE - ADD

Sintaxe:

`ALTER TABLE nome_da_tabela
ADD atributo tipo_de_dados;`

Exemplo 1:

`ALTER TABLE Peca ADD espessura INT;`

Exemplo 2:

`ALTER TABLE empregado ADD funcao varchar(12);`

Pesquisa de tipos de dados do PostgreSQL:

Documentação: <https://www.postgresql.org/docs/12/datatype.html>



Exemplo: ALTER TABLE - DROP

Sintaxe:

`ALTER TABLE nome_da_tabela
DROP atributo [CASCADE|RESTRICT]`

Opções:

- CASCADE: remove todas as restrições relativas ao atributo em outras tabelas
- RESTRICT: não permite a remoção do atributo se este é usado como chave estrangeira em uma outra tabela

Exemplo:

`ALTER TABLE empregado DROP codigo CASCADE;`





SQL – ALTER

Outras opções de alteração:

- ❑ Alterar o nome de um campo/coluna
 - ❑ ALTER TABLE ALTER TABLE <nome_da_tabela>
RENAME COLUMN <nome_da_coluna_a_ser_alterada> TO
<nome_da_nova_coluna>;
 - ❑ Alterar o tipo de dados de um campo/coluna
 - ❑ ALTER TABLE <nome_da_tabela>
ALTER COLUMN nome_da_nova_coluna TYPE tipo_de_dados;



SQL - DROP TABLE

- ❑ O comando DROP TABLE remove tabelas do banco de dados. Somente o dono pode remover a tabela.
- ❑ O comando DROP TABLE sempre remove todos os índices, regras, gatilhos e restrições existentes na tabela de destino.
 - ❑ Entretanto, para remover uma tabela referenciada por uma visão ou por uma restrição de chave estrangeira de outra tabela, deve ser especificado CASCADE (restrição de Integridade referencial)





DROP TABLE

- ❑ Parâmetros
 - ❑ **Nome**: nome da tabela a ser removida
 - ❑ **CASCADE**: remove automaticamente os objetos que dependem da tabela
 - ❑ **RESTRICT**: recusa remover a tabela se existirem objetos que dependem da mesma. Este é o padrão

- ❑ Sintaxe: `DROP TABLE "nome_tabela";`
ou `DROP TABLE "nome_tabela" cascade constraints;`

