Inatel
Instituto Nacional de Telecomunicações

BANCOS DE DADOS

Cap.4 - SQL (Pt.1)

Prof. MSc. Renzo P. Mesquita
renzo@inatel.br

Objetivos

- Introduzir a linguagem SQL (Structured Query Language) e sua importância no mundo dos Bancos de Dados;
- Entender os principais tipos de dados envolvidos em Bancos de Dados SQL;
- Compreender comandos importantes de duas subfamílias de linguagens do SQL: a DDL (Data Definition Language) e a DCL (Data Control Language);
- Entender como configurar as Chaves Estrangeiras das Tabelas de forma adequada;

Capítulo 4
SQL (Pt.1)

4.1. Introdução;
4.2. Tipos de Dados;
4.3. Comandos DDL;
4.4. Comandos DCL;
4.5. Configurando Chaves Estrangeiras;

4.1. Introdução

O SQL (Structured Query Language) é a linguagem de consulta mais popular entre os Bancos de Dados comerciais.

- É utilizada principalmente para facilitar a criação e uso de Bancos de Dados que se baseiam no Modelo Relacional de dados (que vimos no Cap.3);
- É altamente portável entre diversos Bancos de Dados, possuindo poucas variações entre um Banco e outro;

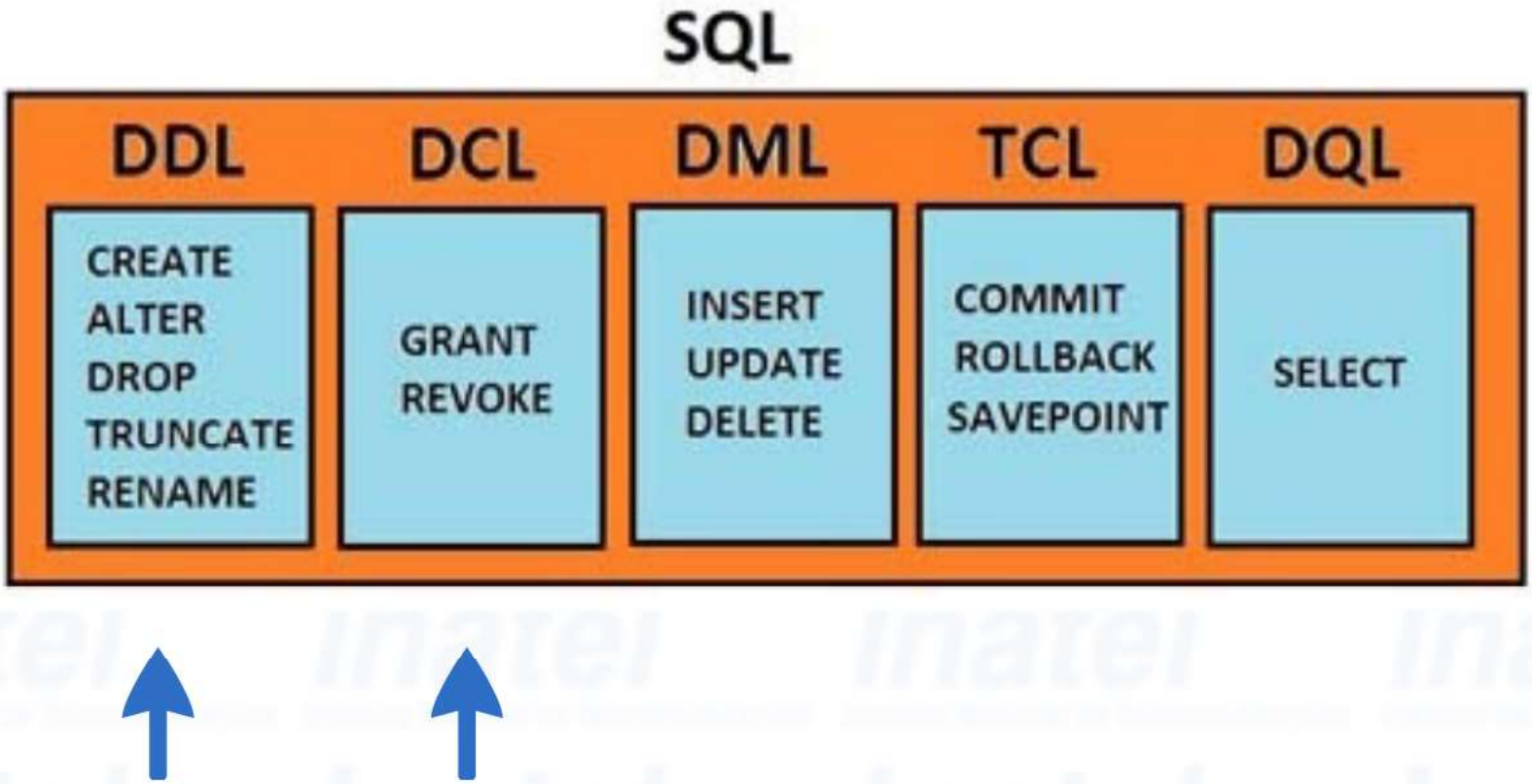
Revisões (upgrades) do SQL com o passar dos anos:

- SQL-86
- SQL-89
- SQL-92
- SQL-1999
- SQL:2003
- SQL:2006
- SQL:2008
- SQL:2011
- SQL:2016
- SQL:2019

Os comandos do SQL são categorizados em subfamílias, cada uma responsável por realizar diferentes perfis de ações no banco. Vamos compreendê-las um pouco melhor?

4.1. Introdução

Subfamílias SQL e seus principais comandos:



Focos de estudo nesta primeira parte do Cap.4.

4.1. Introdução

Modelo Relacional X *SQL (Structured Query Language)*

MODELO RELACIONAL	SQL
Relação	Tabela
Tupla	Linha
Atributo	Coluna

Agora sim! Chegamos em expressões mais cotidianas (Relação é a mesma coisa que Tabela, Tupla a mesma coisa que Linha e Atributos são as Colunas).

4.2. Tipos de Dados

São inúmeros os tipos de dados presentes em Bancos SQL. Porém, destacam-se os mais populares:

1. Numéricos

Inteiros de vários tamanhos:

- TINYINT (0 a 255);
- SMALLINT (-32768 a 32767);
- INTEGER (-(2^32)/2 a (2^32)/2);
- BIGINT (-(2^64)/2 a (2^64)/2);

Números reais:

- FLOAT (-3.4028^38 a 3.4028^38)
- DOUBLE (-1.7976^308 a 1.7976^308)

Números formatados:

- DECIMAL(i,j) ou NUMERIC(i,j) (-10^38 +1 a 10^38 -1) (i = digitos totais e j = digitos depois da vírgula);

2. Cadeia de Caracteres

Tamanho Fixo: • CHAR(n);

Tamanho Variável: • VARCHAR(n);

3. Booleano

- BIT: aceita os valores 1 (para representar TRUE), 0 (para representar FALSE) e NULL (para valor desconhecido) ;

4.2. Tipos de Dados

4. Date e Time

- DATE: Guarda Ano, Mês e Dia no formato YYYY-MM-DD;
- TIME: Guarda a Hora, Minuto e Segundo no formato HH:MI:SS;
- DATETIME: Guarda o Ano, Mês, Dia, Hora, Minuto e Segundo no formato YYYY-MM-DD HH:MI:SS.

ENTRE MUITOS OUTROS..

Observações Importantes:

- Apesar dos tipos apresentados serem populares na maioria dos Bancos SQL, alguns Bancos podem apresentar tipos de dados específicos. Por isso, ao usar um novo Banco, não deixe de consultar uma boa referência para compreender melhor seus tipos suportados;
- Pode acontecer de cada Banco ter um tamanho limite diferente para os tipos de dados apresentados;
- A faixa de valores de um tipo de dado também pode mudar caso o banco permita configurar números sinalizados (positivos ou negativos) ou não sinalizados (somente números positivos);
- Ao utilizar uma Linguagem de Programação para se conectar a um Banco específico, procure saber o equivalente de cada tipo de dado da Linguagem (ou os mais populares) dentro do Banco de Dados (Ex: String == VARCHAR).

4.3. Comandos DDL

Os comandos DDL (Data Definition Language) são utilizados para criação (CREATE) de esquemas, tabelas, usuários, visões e funções, atualização (ALTER) dessas estruturas, assim como a exclusão (DROP) delas.

- Comando *CREATE*

- Principal comando SQL para definição de estruturas;

ESQUEMA do Banco de Dados:

- Identificado por um nome que representa a Base de Dados;
- Agrupa tabelas, visões (views), funções (functions) e outros componentes que pentencem a uma mesma Base de Dados;

Ex: *CREATE SCHEMA* **Empresa**

ou

CREATE DATABASE **Empresa**

4.3. Comandos DDL

TABELAS do Banco de Dados:

- Cada Tabela é identificada por um nome;
- Possui N atributos com tipos e restrições iniciais (NOT NULL por exemplo);
- Restrições ou regras podem ser especificadas inicialmente ou mais tarde (por meio do ALTER TABLE);

Ex:

CREATE TABLE **Empresa.Empregado;**

(Criação da Tabela Empregado anexado ao esquema Empresa)

OU

CREATE TABLE **Empregado;**

(Criação da Tabela Empregado dentro do ambiente do esquema Empresa uma vez que ele já esteja selecionado como esquema padrão)

As tabelas criadas pelo comando *CREATE TABLE* são REAIS, ou seja, são realmente criadas e armazenadas pelo SGBD, diferente das tabelas VIRTUAIS, criadas pelo comando CREATE VIEW (veremos em breve).

4.3. Comandos DDL

Agora, à partir do ESQUEMA Relacional abaixo, vamos buscar compreender um pouco melhor como ficariam seus respectivos códigos DDL.

EMPREGADO

PNome	MInicial	UNome	SSN	DataNasc
Endereco	Sexo	Salario	SSN_Supervisor	DNumero_Departamento

DEPARTAMENTO

DNome	DNumero	SSN_Empregado	DataInicio

DEPTO_LOCALIZACOES

LNumero	DLocalizacao	DNumero_Departamento

PROJETO

PNome	PNumero	PLocalizacao	DNumero_Departamento

TRABALHA_EM

SSN_Empregado	PNumero_Projeto	Horas

DEPENDENTE

SSN_Empregado	Nome_Dependente	Sexo	DataNasc	Parentesco

Como ficaria, por exemplo, um possível código SQL para criação da Tabela Empregado?

4.3. Comandos DDL

Ex:

Tabela EMPREGADO (versão 1)

```
CREATE TABLE EMPREGADO
(
    PNOME   VARCHAR(15)   NOT NULL,
    MINICIAL   CHAR,
    UNOME   VARCHAR(15)   NOT NULL,
    SSN   BIGINT   NOT NULL,
    DATANASC   DATE,
    ENDERECO   VARCHAR(80),
    SEXO   BIT,
    SALARIO   DECIMAL(10,2),
    SSN_SUPERVISOR  BIGINT,
    DNUMERO_DEPARTAMENTO   INTEGER   NOT NULL,

    PRIMARY KEY (SSN),
    CONSTRAINT fk1
    FOREIGN KEY (SSN_SUPERVISOR) REFERENCES EMPREGADO (SSN),
    CONSTRAINT fk2
    FOREIGN KEY (DNUMERO_DEPARTAMENTO) REFERENCES DEPARTAMENTO (DNUMERO)
);
```

4.3. Comandos DDL

Outros exemplos de comandos SQL DDL populares que poderiam ser utilizados internamente ao se criar uma nova Tabela:

UNIQUE (DNome);

(É como se fosse uma chave primária, ou seja, certifica que um campo não receba valores repetidos. Porém, não é indexado pelo banco para prover buscas mais rápidas e pode receber valores nulos)

PRIMARY KEY (Nome_Dependente, SSN_Empregado)

(Quando dois atributos são usados para se formar uma chave primária - Chave Composta)

DNUMERO_DEPARTAMENTO INT NOT NULL DEFAULT 1;

(Atribui um valor DEFAULT (padrão) para um atributo caso ele não seja preenchido)

DNUMERO_DEPARTAMENTO INT NOT NULL CHECK (DNUMERO_DEPARTAMENTO > 0 AND DNUMERO_DEPARTAMENTO < 10);

(Programa o campo para limitar os valores possíveis para um atributo)

LNUMERO INT NOT NULL AUTO_INCREMENT

(Permite programar uma Tabela para gerar valores de chave primária de forma automática e incrementável para um novo registro)

ENTRE MUITOS OUTROS..

4.3. Comandos DDL

- *Comando DROP*

- Deleta uma Tabela ou todo um Esquema de um Banco de Dados.

Ex: *DROP SCHEMA* **Empresa**

(Deleta o Esquema e todas suas Tabelas. Para alguns BD's, usa-se a tag DATABASE no lugar de SCHEMA)

DROP TABLE **Dependente**

(Deleta a Tabela Dependente e todos os seus registros)

- *Comando ALTER*

- Adiciona ou elimina uma coluna, altera a definição de uma coluna, adiciona ou remove restrições (regras).

Ex: *ALTER TABLE* **Empresa.Empregado** *ADD* **Cargo VARCHAR(15)**

(Criação do atributo Cargo na Tabela Empregado. O novo atributo conterá o valor NULL para todas as tuplas existentes na Tabela, pois foi adicionado depois)

4.3. Comandos DDL

- *Comando* *ALTER* (mais exemplos)

ALTER TABLE **Empresa.Empregado** *DROP* **Cargo**

(Remoção do atributo Cargo na tabela Empregado)

ALTER TABLE **Empregado** *CHANGE COLUMN* **SEXO SEX BIT**

(Trocando o nome da coluna Sexo para Sex. Atenção, deve-se colocar o tipo no final)

ALTER TABLE **Empresa.Empregado** *ALTER* **SSN_SUPERVISOR** *SET DEFAULT* **1**

(coloca o supervisor com o id=1 como default para as novas tuplas que serão criadas)

- *Comando* *TRUNCATE*

Ex: *TRUNCATE TABLE* **Dependentes**

(Limpa todos os registros de uma Tabela mas mantém sua estrutura para futuras inserções)

- *Comando* *RENAME*

Ex:

RENAME TABLE **Funcionario** *TO* **Empregado**

(Renomeia o nome de uma Tabela. Em alguns bancos, também pode ser usado para renomear uma coluna)

ENTRE MUITOS OUTROS..

4.4. Comandos DCL

Os comandos DCL (Data Control Language) são utilizados para conceder (GRANT) e revogar (REVOKE) acesso de usuários a um Banco de Dados.

Mas antes de compreendermos melhor estes comandos, precisamos entender como criar um novo usuário para fazer uso de um Banco de Dados.

Ex: *CREATE USER* **'jaspion'** *IDENTIFIED BY* **'daileon123'**

usuário senha

Uma vez que um novo usuário é criado, é necessário que o acesso dele a um Esquema ou Base de Dados seja configurado de forma adequada, ou seja, é necessário definir os PRIVILÉGIOS que o mesmo terá sobre ela.

Os privilégios mais populares são:

ALL PRIVILEGES :o usuário tem total liberdade para manipular o Banco;

CREATE *DROP* :o usuário poderá crair e excluir Tabelas em um Banco;

INSERT *UPDATE* *DELETE* *SELECT* :o usuário poderá inserir, atualizar, excluir e buscar informações no Banco;

4.4. Comandos DCL

- Comando *GRANT*

Ex:

GRANT SELECT, INSERT, UPDATE, DELETE ON **Empresa.Projeto** *TO* **'jaspion'**

(Permitindo que o usuário use de comandos DML dentro da tabela Projeto)

GRANT ALL PRIVILEGES ON **Empresa.*** *TO* **'jaspion'**

(Permitindo que o usuário tenha todos os privilégios sobre o Banco de Dados Empresa)

- Comando *REVOKE*

REVOKE ALTER, CREATE, DROP ON **Empresa.*** *FROM* **'jaspion'**

(Retirando algumas permissões DDL do usuário)

ENTRE MUITOS OUTROS..

4.5. Configurando Chaves Estrangeiras

Uma *INTEGRIDADE REFERENCIAL* pode ser VIOLADA quando linhas são inseridas, deletadas ou quando o valor do atributo da chave estrangeira for modificado.

A ação padrão do SQL é REJEITAR a operação que iria causar a violação.

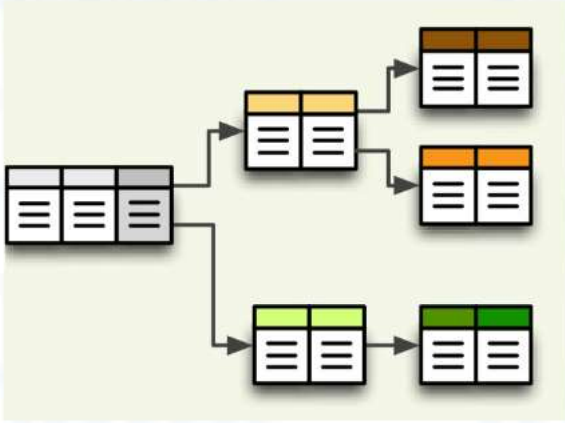
Para evitar este problema, o projetista do BD pode especificar algumas ações chamadas de **AÇÕES REFERENCIAIS ENGATILHADAS.**

Estas ações devem ser configuradas nas CHAVES ESTRANGEIRAS por meio da combinação dos comandos:

ON DELETE ou *ON UPDATE*

com

SET NULL , *CASCADE* ou *SET DEFAULT*



4.5. Configurando Chaves Estrangeiras

Ex:

ON DELETE SET NULL (ao deletar, marque nulo)

(se uma linha referenciada for deletada, todas as linhas que fazem referência a ela receberão valor NULL na chave estrangeira);

ON DELETE CASCADE (ao deletar, propague)

(se uma linha referenciada for deletada, todas as linhas que fazem referência a ela também serão deletadas);

ON UPDATE CASCADE (ao atualizar, propague)

(se uma linha referenciada for atualizada, todas as tuplas que fazem referência a esta linha receberão o novo valor dela na chave estrangeira);

Para pensar: Quais combinações não fariam tanto sentido?

4.5. Configurando Chaves Estrangeiras

Ex:

Tabela EMPREGADO (versão 2)

```
CREATE TABLE EMPREGADO
(
    PNOME   VARCHAR(15)   NOT NULL,
    MINICIAL   CHAR,
    UNOME   VARCHAR(15)   NOT NULL,
    SSN BIGINT   NOT NULL,
    DATANASC   DATE,
    ENDERECO   VARCHAR(80),
    SEXO   BIT,
    SALARIO   DECIMAL(10,2),
    SSN_SUPERVISOR   BIGINT,
    DNUMERO_DEPARTAMENTO   INTEGER   NOT NULL   DEFAULT 1,

    PRIMARY KEY (SSN),
    CONSTRAINT fk1
    FOREIGN KEY (SSN_SUPERVISOR) REFERENCES EMPREGADO(SSN) ON DELETE SET NULL ON
    UPDATE CASCADE,
    CONSTRAINT fk2
    FOREIGN KEY (DNUMERO_DEPARTAMENTO) REFERENCES DEPARTAMENTO(DNUMERO) ON
    DELETE SET DEFAULT ON UPDATE CASCADE;
);
```

4.5. Configurando Chaves Estrangeiras

Exercício Proposto

Para as Tabelas faltantes do Modelo Relacional abaixo, escreva suas declarações DDL SQL. Para as Chaves Estrangeiras, escolha a ação apropriada – cascade, set null, set default.. – para cada restrição de integridade referencial, tanto para exclusão da linha (ON DELETE), quanto para atualização (ON UPDATE).

EMPREGADO

PNome	MInicial	UNome	SSN	DataNasc
Endereco	Sexo	Salario	SSN_Supervisor	DNumero_Departamento

DEPARTAMENTO

DNome	DNumero	SSN_Empregado	DataInicio

DEPTO_LOCALIZACOES

LNumero	DLocalizacao	DNumero_Departamento

PROJETO

PNome	PNumero	PLocalizacao	DNumero_Departamento

TRABALHA_EM

SSN_Empregado	PNumero_Projeto	Horas

DEPENDENTE

SSN_Empregado	Nome_Dependente	Sexo	DataNasc	Parentesco

Observações:

- O DDL da Tabela Empregado não precisa ser feito, pois já foi usada como exemplo;
- Para configuração das Chaves Estrangeiras, utilize as opções que achar mais conveniente;
- Para os tipos das colunas das Tabelas, utilize as opções que achar mais conveniente.

FIM
DO
CAPÍTULO 4
(Parte 1)

EXERCÍCIOS