

SQL – DDL

(Data Language Definition -
linguagem de definição de dados)



LINGUAGEM SQL

- SQL (*Structured Query Language*) é uma linguagem para criação e manipulação de bancos de dados relacionais.
- Tem como base a álgebra relacional e cálculo relacional.



LINGUAGEM SQL

- **DDL** (*Data Definition Language*): linguagem de definição de dados
 - Define esquemas, tabelas, chaves primárias e estrangeiras, exclusão de esquemas, tabelas e colunas, alteração de tabelas.
 - Diz respeito à **estrutura** das tabelas e esquema do BD.
- **DML** (*Data Manipulation Language*): linguagem de manipulação de dados.
 - Consulta, inserção dos dados do BD, exclusão de dados, alteração de dados. Diz respeito aos **dados** das tabelas do BD – CRUD.
- **DCL** (*Data Control Language*): linguagem de controle de dados.
 - Define permissões.

LINGUAGEM SQL

- SQL = DDL + DML + DCL
- Principais comandos:
 - DDL (Linguagem de definição de dados):
 - CREATE, DROP, ALTER
 - DML (Linguagem de manipulação de dados):
 - SELECT, INSERT, UPDATE, DELETE
- Conceitos:
 - Table = Relação (tabela)
 - Row = tupla (linha)
 - Column = atributo (coluna)

DDL: PROPRIEDADES

- A DDL permite não só a especificação de um conjunto de relações (tabelas), como também informações acerca de cada uma das relações, incluindo:
 - O esquema de cada relação (estrutura);
 - O domínio dos valores associados a cada atributo (int, float, varchar, etc);
 - As regras de integridade (**chave, entidade, referencial**);
 - Informações acerca de cada uma das relações:
 - O conjunto de índices para manutenção de cada relação;
 - Informações sobre segurança e autoridade sobre cada relação;
 - A estrutura de armazenamento físico de cada relação no disco.

DDL: CREATE DATABASE/ESQUEMA

- Antes de criar qualquer tabela é necessário criar um esquema (ou base de dados!!!)
- Em SQL, uma base de dados (ou esquema) é identificada através de um **nome**. Os elementos do esquema incluem tabelas, restrições, etc.

- **Sintaxe:**

```
CREATE SCHEMA <nome>;  
ou  
CREATE DATABASE <nome>
```

- Exemplos:
 - CREATE SCHEMA Empresa;
 - CREATE SCHEMA Universidade;
 - CREATE DATABASE Hospital;

DDL: TIPOS DE DOMÍNIOS

- Numéricos:
 - **INTEGER** : é um inteiro, abreviação da palavra, **integer** (em inglês).
 - **NUMERIC (p, d)**: é um numero de ponto fixo cuja precisão é definida pelo usuário. O número consiste de p dígitos (mais o sinal), sendo que d dos p dígitos estão à direita do ponto decimal.
 - Ex.: `numeric(6,2)`: 3000,00
 - **SERIAL**: números inteiros auto incrementáveis.
- Character/String:
 - **CHAR(n)**: é uma cadeia de caracteres de tamanho fixo, com o tamanho n definido pelo DBA. Abreviação de **character** (em inglês).
 - **VARCHAR(N)**: é uma cadeia de caracteres de tamanho variável, com o tamanho n máximo definido pelo DBA. Abreviação de **character varying** (em inglês).

DDL: TIPOS DE DOMÍNIOS

- Booleano:
 - **BOOLEAN:** assume dois valores, falso e verdadeiro (true/ false).
- Data/Tempo:
 - **DATE:** tipo de dado que contém um ano (com 4 dígitos) mês e dia do mês, sendo o formato “**aaaa/mm/dd**” o padrão do MySQL Workbench.
 - **Year (date)** – retorna o ano de uma data
 - **Month (date)** – retorna o mês de um data
 - **Day (date)** – retorna o dia de uma data
 - **TIME:** representa horário, com o seguinte formato: horas, minutos e segundos (00:00:00).
 - OBS: Alguns SGBDs oferecem o domínio **TIMESTAMP** que contém a data (ano, mês e dia) e horário (horas, minutos, segundos e milissegundos).

DDL: TIPOS DE DOMÍNIOS

OBSERVAÇÕES

- Uma **chave estrangeira** deve possuir o **mesmo tipo de domínio** da **chave primária** correspondente.
- O valor *nulo* é um membro de todos os domínios, isto é, **qualquer atributo pode assumir o valor NULL, exceto o atributo chave primária (restrição integridade de entidade)**;
- A SQL permite que a declaração de domínio de qualquer atributo inclua a especificação de **not null**, proibindo, assim, a inserção de um valor nulo para esse tipo de atributo (obrigatório na PK).

DDL: CREATE TABLE, DROP TABLE

- **CREATE TABLE:**

- define a estrutura de uma tabela, suas restrições de integridade e cria a tabela vazia

- Sintaxe: `CREATE TABLE <nome_tabela> (...);`

- Ex.: CREATE TABLE empregado (atributo1 tipo, atributo 2, tipo ...);

- **DROP TABLE:**

- Remove todos os dados e a própria relação (estando vazia ou não).

- Sintaxe: `DROP TABLE <nome_tabela>;`

- Ex.: DROP TABLE empregado;

DDL: CREATE TABLE – SINTAXE

Exemplo

```
CREATE DATABASE EMPRESA;
```

```
CREATE TABLE DEPARTAMENTO (
```

```
  CodDep SERIAL NOT NULL,
```

```
  NomeDep VARCHAR(30),
```

```
  PRIMARY KEY (CodDep)
```

```
);
```

```
CREATE TABLE FUNCIONARIO (
```

```
  Matricula INTEGER NOT NULL,
```

```
  Nome VARCHAR(30) NOT NULL,
```

```
  Salario NUMERIC(8,2), Cargo VARCHAR(15) DEFAULT 'Analista',
```

```
  Estado CHAR(2),
```

```
  Idade INTEGER, CodDeppto INT,
```

```
  PRIMARY KEY (Matricula),
```

```
  FOREIGN KEY (CodDeppto) references DEPARTAMENTO (CodDep) ON DELETE  
    NO ACTION ON UPDATE NO ACTION
```

```
);
```

DDL: CREATE TABLE - SINTAXE

- Criação de um campo que gere **códigos automáticos** não é padrão SQL, mas caso seja necessário, basta colocar o tipo **SERIAL** na criação do campo.
- Muito utilizado em relações que possuem **IDs**.

- Exemplo:

```
CREATE TABLE cidade (  
    id_cidade SERIAL NOT NULL,  
    nome_cidade VARCHAR(100) NOT NULL,  
    PRIMARY KEY (id_cidade)  
);
```

DDL: ALTER TABLE

A SQL oferece instruções para definição do esquema da base de dados:

- **ALTER TABLE:**
- Usado para alterar o esquema da relação.
- permite modificar a definição de uma tabela
 - Operações de inserção, alteração e exclusão → atenção aos atributos e restrições de integridade
 - atributos chave não podem ser removidos;
- Sintaxe: `ALTER TABLE <nome da tabela> ;`

DDL: ALTER TABLE

1. Sintaxe básica para inclusão de uma coluna:

ALTER TABLE nome_tabela **ADD COLUMN** nome_coluna tipo_atributo;

Ex.: ALTER TABLE funcionario ADD COLUMN identidade varchar(10);

2. Sintaxe básica para exclusão de uma coluna:

ALTER TABLE nome_tabela **DROP** nome_coluna;

Ex.: ALTER TABLE empregado DROP estado;

3. Sintaxe básica para alteração do nome de uma coluna:

ALTER TABLE nome_tabela **RENAME COLUMN** nome_coluna_atual TO
novo_nome_coluna;

Ex.: ALTER TABLE empregado RENAME COLUMN Sexo TO Genero;

DDL: ALTER TABLE

- **Observe-se que:**
 - A instrução **adiciona uma nova coluna com o valor vazio** para todas as linhas, isto é, **sem nenhum valor** armazenado.
 - O mesmo acontece quando há a criação de uma tabela...
 - A princípio ela não está “povoada” com dados, está vazia, ausente de valores (em outras palavras: não há linhas/ tuplas na tabela).
 - Os valores para as diversas linhas devem ser adicionadas através de instruções da DML (INSERT INTO...), que será assunto da próxima aula.

DDL: Atributos, Primary Key, Foreign Key

- Chave primária:
 - PRIMARY KEY (PK)
- Restrições de atributos (PostgreSQL):
 - NOT NULL (não nulo) - NN
 - DEFAULT <value>
- Chave estrangeira:
 - FOREIGN KEY (FK)
 - Cláusulas:
 - ON DELETE
 - CASCADE
 - SET NULL
 - SET DEFAULT
 - RESTRICT
 - NO ACTION
 - ON UPDATE

ON DELETE, ON UPDATE

NO ACTION

Produce an error indicating that the deletion or update would create a foreign key constraint violation. If the constraint is deferred, this error will be produced at constraint check time if there still exist any referencing rows. This is the default action.

RESTRICT

Produce an error indicating that the deletion or update would create a foreign key constraint violation. This is the same as NO ACTION except that the check is not deferrable.

CASCADE

Delete any rows referencing the deleted row, or update the values of the referencing column(s) to the new values of the referenced columns, respectively.

SET NULL [(*column_name* [, ...])]

Set all of the referencing columns, or a specified subset of the referencing columns, to null. A subset of columns can only be specified for ON DELETE actions.

SET DEFAULT [(*column_name* [, ...])]

Set all of the referencing columns, or a specified subset of the referencing columns, to their default values. A subset of columns can only be specified for ON DELETE actions. (There must be a row in the referenced table matching the default values, if they are not null, or the operation will fail.)

If the referenced column(s) are changed frequently, it might be wise to add an index to the referencing column(s) so that referential actions associated with the foreign key constraint can be performed more efficiently.

DDL: FK – ON DELETE

- CASCADE

- Pode ser definida a propagação da exclusão da linha para as linhas que a referenciam (em cascata).
- Sintaxe:

... FOREIGN KEY ... REFERENCES ... ON DELETE **CASCADE**

OBS.: Ao excluir uma tabela, a utilização do CASCADE informa se a exclusão deve ser propagada ou não para valores definidos com base na tabela (referências).

DDL: FK – ON DELETE

- RESTRICT

- Especifica que uma tupla (linha) não pode ser excluída caso existam chaves estrangeiras que a referenciem (dará erro ao tentar excluir).

- Sintaxe:

... FOREIGN KEY ... REFERENCES ... ON DELETE **RESTRICT**

DDL: FK – ON DELETE

- NO ACTION (padrão – default)
 - Especifica que ao tentar efetuar a exclusão de uma linha com uma chave primária referenciada por chaves estrangeiras em outras tabelas, um erro é ativado e a instrução DELETE é revertida.
 - Sintaxe:

... FOREIGN KEY ... REFERENCES ... ON DELETE NO ACTION

DDL: FK – ON DELETE

- SET NULL

- Especifica que o valor de atributo da chave estrangeira, caso seja excluído em outra tabela como chave primária, receba o valor NULL.

- Sintaxe:

... FOREIGN KEY ... REFERENCES ... ON DELETE SET NULL

DDL: FK – ON DELETE

- SET DEFAULT

- Especifica que o valor de atributo da chave estrangeira, caso seja excluído em outra tabela como chave primária, receba o valor DEFAULT.

- Sintaxe:

... FOREIGN KEY ... REFERENCES ... ON DELETE SET DEFAULT

DDL: FK – ON UPDATE

- CASCADE

- Alteração da chave primária é propagada para as chaves estrangeiras que a referenciam (em cascata).

- Sintaxe:

FOREIGN KEY ... REFERENCES ... ON UPDATE CASCADE

DDL: FK – ON UPDATE

- RESTRICT

- Uma chave primária não pode ser alterada, caso existam chaves estrangeiras que a referenciem. Neste caso não será alterado automaticamente. Software acusa erro.
- Sintaxe:

FOREIGN KEY ... REFERENCES ... ON UPDATE **RESTRICT**

DDL: FK – ON UPDATE

- SET NULL
 - Especifica que o valor de atributo da chave estrangeira, caso seja atualizado e não exista tal valor chave primária em outra tabela, receberá o valor NULL.
 - Sintaxe:

... FOREIGN KEY ... REFERENCES ... ON DELETE SET NULL

DDL: FK – ON UPDATE

- SET DEFAULT

- Especifica que o valor de atributo da chave estrangeira, caso seja atualizado e não exista tal valor chave primária em outra tabela, receberá o valor DEFAULT.

- Sintaxe:

... FOREIGN KEY ... REFERENCES ... ON DELETE SET
DEFAULT

DDL: FK – ON UPDATE

- NO ACTION (padrão)

- Especifica que se uma tentativa for feita para atualização de uma chave referenciada em uma linha cujas chaves sejam referenciadas por chaves estrangeiras em linhas existentes de outras tabelas, um erro é ativado e a instrução UPDATE é revertida.

- Sintaxe:

FOREIGN KEY ... REFERENCES ... ON UPDATE NO ACTION

DDL: DROP DATABASE

- Para excluir uma base de dados (ou um esquema) existente, é necessário usar o comando DROP.
 - **Sintaxe:** `DROP <nome_esquema>;`
- Exemplos:
 - DROP Empresa;
 - DROP Universidade;
 - DROP Hospital;

Cuidado ao remover uma base de dados. TODAS as tabelas, relacionamentos e dados contidos na base serão permanentemente removidos!!!