

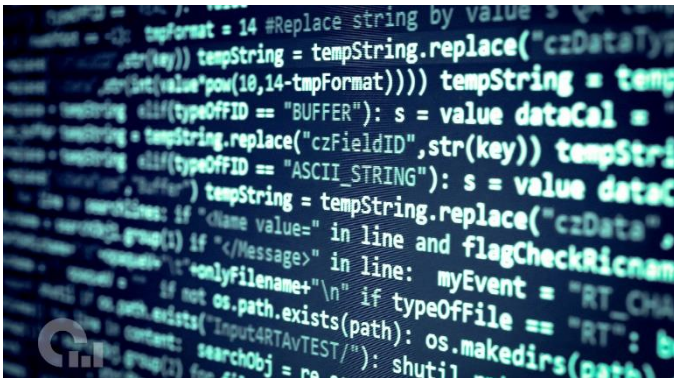
INSTITUTO MAUÁ DE TECNOLOGIA



18

Banco de dados

Definições



Conceitos Básicos

Dados

São fatos conhecidos que podem ser armazenados e possuem um conhecimento implícito;

Banco de dados

É uma coleção de dados relacionados;

Sistema Gerenciador de Banco de Dados (SGBD) ou *Database Management System (DBMS)*:

Software/pacote que facilita a criação/manutenção de um banco de dados computadorizado;

Sistema de banco de dados

É o *software* SGDB junto aos dados e aplicações.

Definir o banco de dados

Definir tipos de dados, estruturas e restrições;

Construir o banco de dados

Armazenar o banco de dados em um meio de armazenamento secundário;

Manipular o banco de dados

Realizar consultas, gerar relatórios, inserir, eliminar e modificar seu conteúdo;

Concorrência

Processamento concorrente e compartilhamento de recursos por um conjunto de usuários e programas, mantendo todos os dados válidos e consistentes.

Outras funcionalidades

Proteção ou segurança contra acessos não autorizados;

Apresentação e visualização dos dados;

Processamento ativo gerando ações internas sobre os dados.

Sistema de informação de uma universidade

Elementos de dados que podem ser armazenados:

- Estudantes;
- Cursos;
- Departamentos;
- Disciplinas;
- Professores.

Elementos de ligação (relacionamentos) que devem ser levados em conta:

- Um **Departamento** pode oferecer vários **Cursos** e um **Curso** deve estar ligado a um único **Departamento**;
- Um **Professor** pode lecionar diversas **Disciplinas** e uma **Disciplina** pode ser ministrada por vários **Professores**.

Quem usa um SGBD?

Administrador de banco de dados

Responsável por **autorizar acessos** ao banco de dados, **coordenar** e **monitorar** seu **uso**, **adquirir recursos** de *software* e *hardware*, **controlar** o **uso** e **monitorar** a eficiência de suas operações;

Projetista de banco de dados

Responsável pela **definição** do **conteúdo**, a **estrutura**, **restrições**, **funções** ou **transações** sobre o banco de dados. **Comunicam-se** com **usuários finais** para entender suas **necessidades**;

Usuários finais

Executam consultas e relatórios, bem como podem realizar operações atualizações no BD.

Definição

É um conjunto de conceitos que descreve a **estrutura** do banco de dados e certas **restrições** que o banco de dados deve obedecer;

Operações de um modelo de dados

São as operações para especificar **como recuperar e atualizar dados** no banco de dados, utilizando conceitos do modelo de dados. Incluem operações básicas e definidas pelos usuários.

Modelo relacional

É o modelo de banco de dados mais utilizado atualmente;

Princípios

Os dados são organizados em tabelas: cada tabela representa uma entidade de dados do “negócio”;

Cada coluna de uma tabela representa um atributo da entidade;

Cada linha representa uma instância de uma entidade – um registro daquela entidade;

Operações entre tabelas resultam em tabelas – conceitos de álgebra relacional e conjuntos;

A linguagem que se consagrou para manipular banco de dados relacionais é SQL – *Structured Query Language*.

DDL (*Data Definition Language*)

É uma linguagem ou notação utilizada para a criação de esquemas de BD, permitindo a geração de tabelas que são armazenadas em um dicionário de dados.

DML (*Data Manipulation Language*)

É a linguagem utilizada para acessar e manipular os dados organizados segundo algum modelo de dados. Podem ser classificadas em:

- **Procedimentais:** o usuário especifica os dados e como recuperar esses dados (funções e procedimentos + dados);
- **Não procedimentais:** o usuário especifica os dados sem especificar como recuperar esses dados (linguagens declarativas). É o caso de SQL (*Structured Query Language*).

Objetivos

Identificar, no domínio da aplicação, as entidades envolvidas, seus atributos, bem como relacionamentos existentes entre si, de modo independente de tecnologia;

É normalmente conduzida como parte das tarefas de engenharia de requisitos e de análise do sistema, em um processo de desenvolvimento de *software*;

Execução

Realizada com ferramentas gráficas/textuais:

- Diagramas de entidade-relacionamento – **DER** (tradicional e mais utilizado);
- Diagramas de classe **UML** (sendo gradativamente adotado nas empresas);

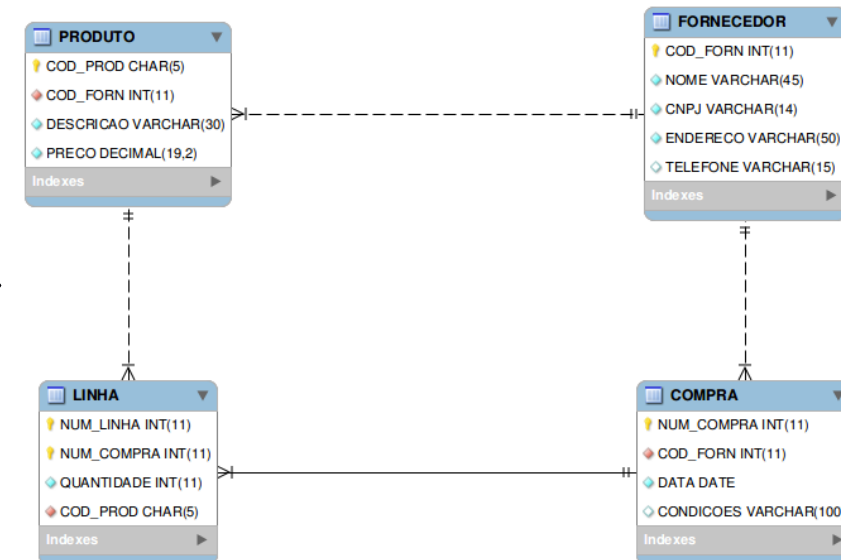
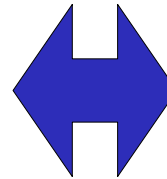
Criação do banco de dados

Projeto conceitual

O projeto conceitual é normalmente realizado com o auxílio de um diagrama de entidade-relacionamento (DER);

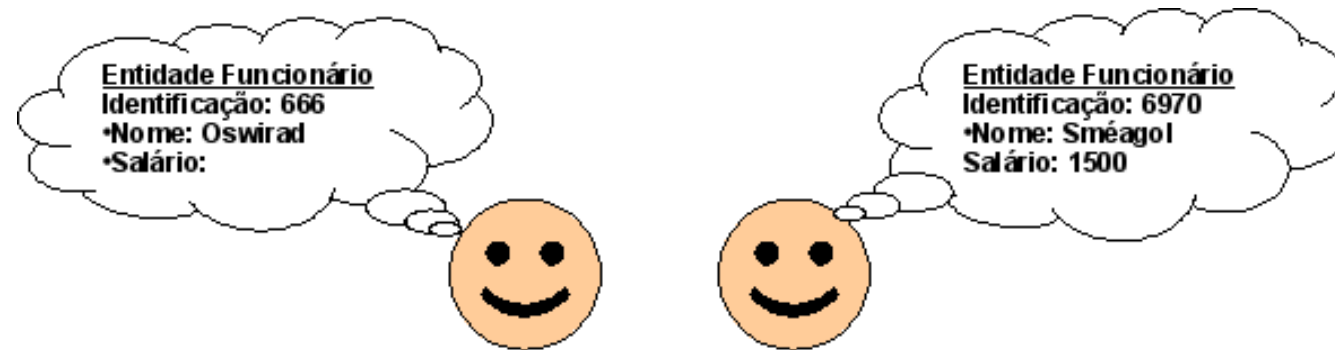
Exemplo: sistema simplificado de ordens de compra.

Ordem de Compra					
Numero:	1				
Data:	17/5/2012				
Condições:	45 dias				
Cod.Fornecedor:	F2				
#	Cod.Produto	Preço	Qtde	Subtotal	
1	P8	R\$ 7,00	10	R\$ 70,00	
2	P11	R\$ 2,00	5	R\$ 10,00	
3	P8	R\$ 7,00	3	R\$ 21,00	
4					
5					
			Total:	R\$ 101,00	

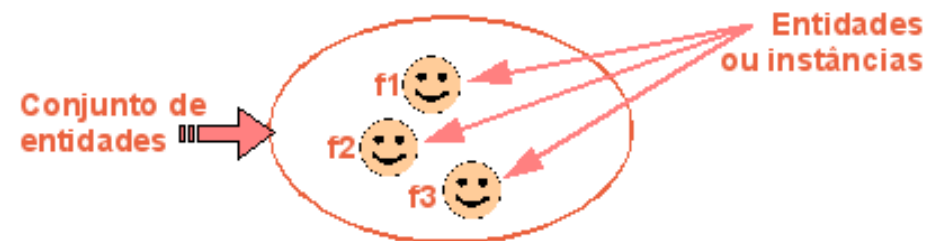


Definições

Entidade: Algum “objeto” real que foi abstraído no sistema.



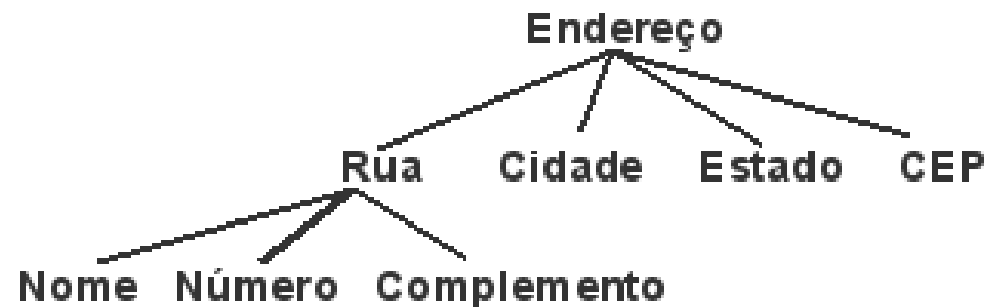
Conjunto de entidades: Categoria ou agrupamento de entidades com características similares e são descritas por um tipo de entidade – imaginar como sendo um conjunto matemático contendo entidades particulares.



Definições

Atributo: Representa uma propriedades de uma entidade;. Por exemplo, os funcionários do slide anterior possuem os atributos: Identificação, Nome e Salário;
Podem ser simples ou atômicos (do exemplo acima) ou ainda compostos ou multi-valorados.

Exemplo de atributo multi-valorado (endereço de uma empresa):

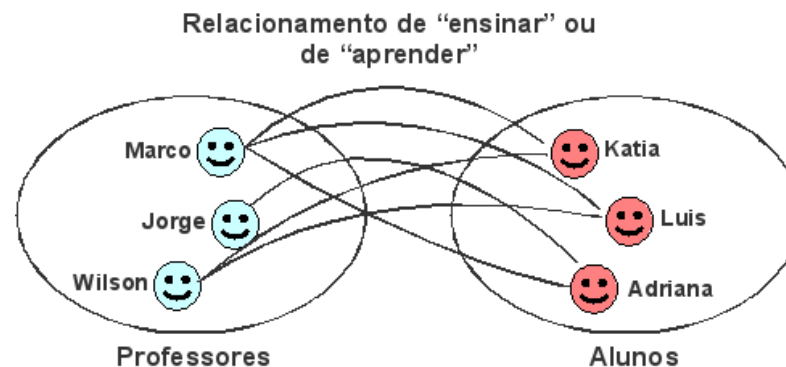


Definições

Atributo chave: É o atributo que permite diferenciar as entidades de um tipo entidade entre si. Por exemplo, o CPF de uma pessoa;

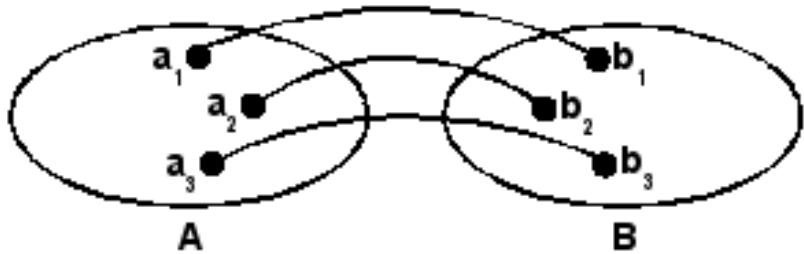
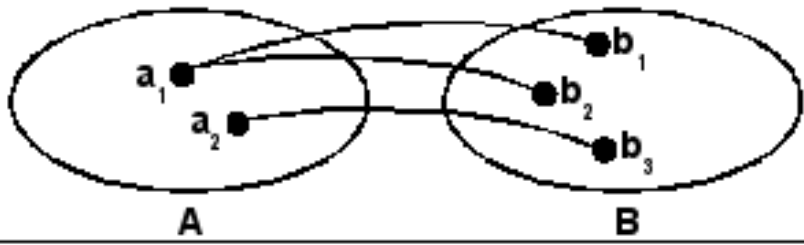

Relacionamento: Representa uma associação existente (de acordo com o domínio em estudo) entre conjuntos de entidades.

Relacionamentos do mesmo tipo são agrupados em um tipo de relacionamento, que mapeia entidades de um conjunto com entidades de um outro conjunto:



Definições

Cardinalidade do relacionamento: Indica o grau de associação entre duas ou mais tipos de entidades. Alguns exemplos:

Cardinalidade	Exemplo
1:1 - uma instância de A se associa exatamente com uma instância de B e vice-versa.	
1:N ou N:1 - uma instância de A se associa com um número qualquer de instâncias de B, mas uma instância de B se associa com somente uma de A.	
M:N - uma instância de A pode se associar com qualquer número de instâncias de B e vice-versa.	

Instruções para Manipulação de Banco de dados Linguagem SQL

Structured Query Language

- Linguagem de manipulação de banco de dados;
- Não procedural: pois baseia-se na descrição “do que obter” ao invés de “como obter”.
- As operações são realizadas sobre tabelas e as repostas obtidas são tabelas → álgebra relacional.
- Padrão internacional
- SQL2 ou SQL-92 (ISO 9075): maior revisão do padrão, amplamente suportado pelos bancos de dados;
- SQL3 ou SQL-99: adições de expressões regulares, consultas recursivas e características orientadas a objetos;
- Recentemente: SQL-2003 e SQL-2006 – extensões XML.

Manipulação de dados (DML)

SELECT: recupera dados de um banco de dados;

INSERT: adiciona novas linhas ao banco de dados;

DELETE: remove linhas de dados de um banco de dados;

UPDATE: modifica dados existentes em um banco de dados.

Definição de dados (DDL)

CREATE TABLE: adiciona nova tabela ao banco de dados;

DROP TABLE: remove uma tabela do banco de dados;

ALTER TABLE: altera a estrutura de uma tabela;

CREATE VIEW: adiciona uma *view* ao banco de dados;

DROP VIEW: remove uma *view* do banco de dados;

Definição de dados (DDL) [continuação]

CREATE SCHEMA: adiciona um *schema* ao banco de dados;

DROP SCHEMA: remove um *schema* do banco de dados;

CREATE DOMAIN: cria um novo domínio no banco de dados;

DROP DOMAIN: remove um domínio do banco de dados;

ALTER DOMAIN: altera um domínio do banco de dados.

Controle de acesso

GRANT: Concede privilégios de uso ao um usuário;

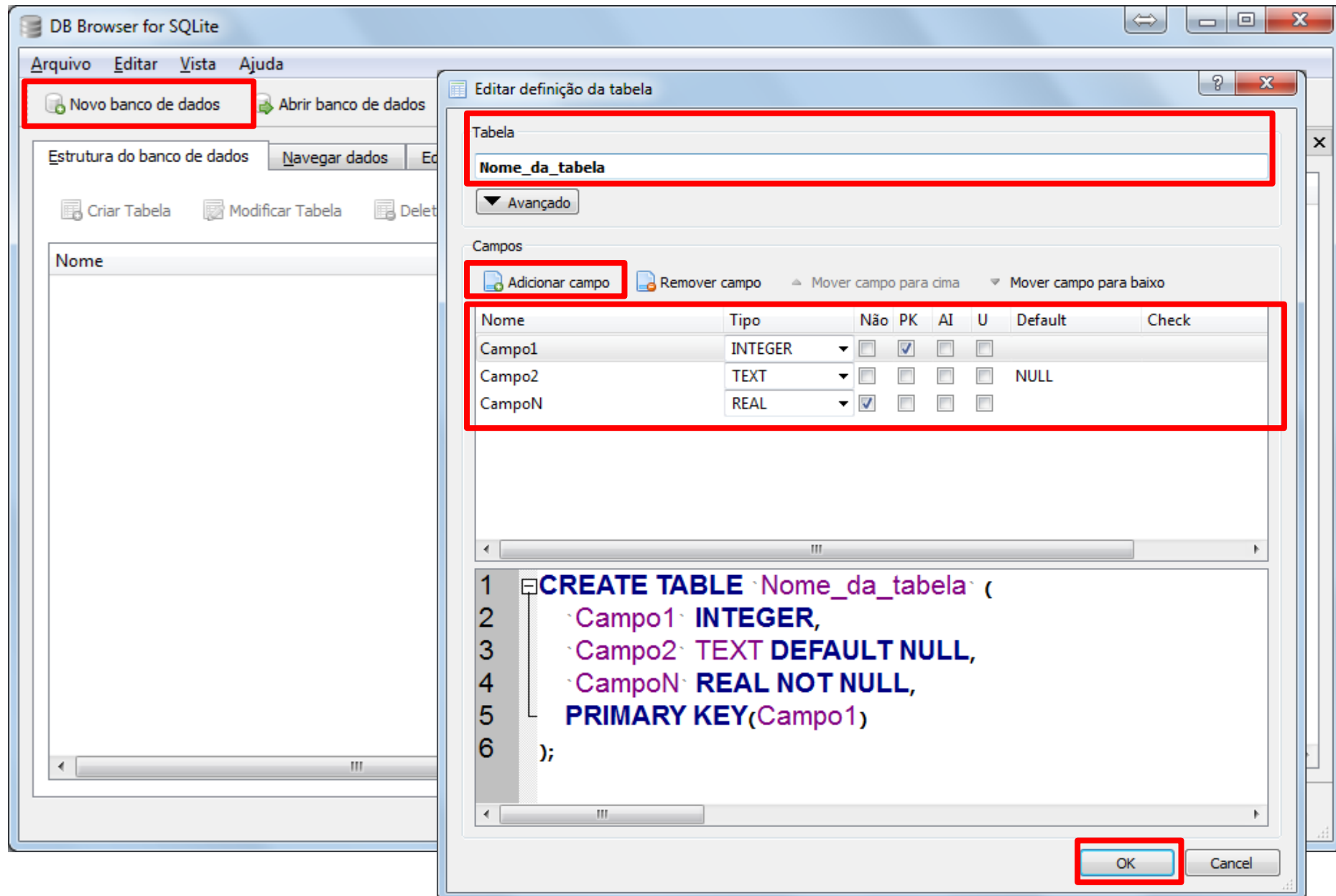
REVOKE: Remove privilégios de uso ao um usuário.

Controle de transações

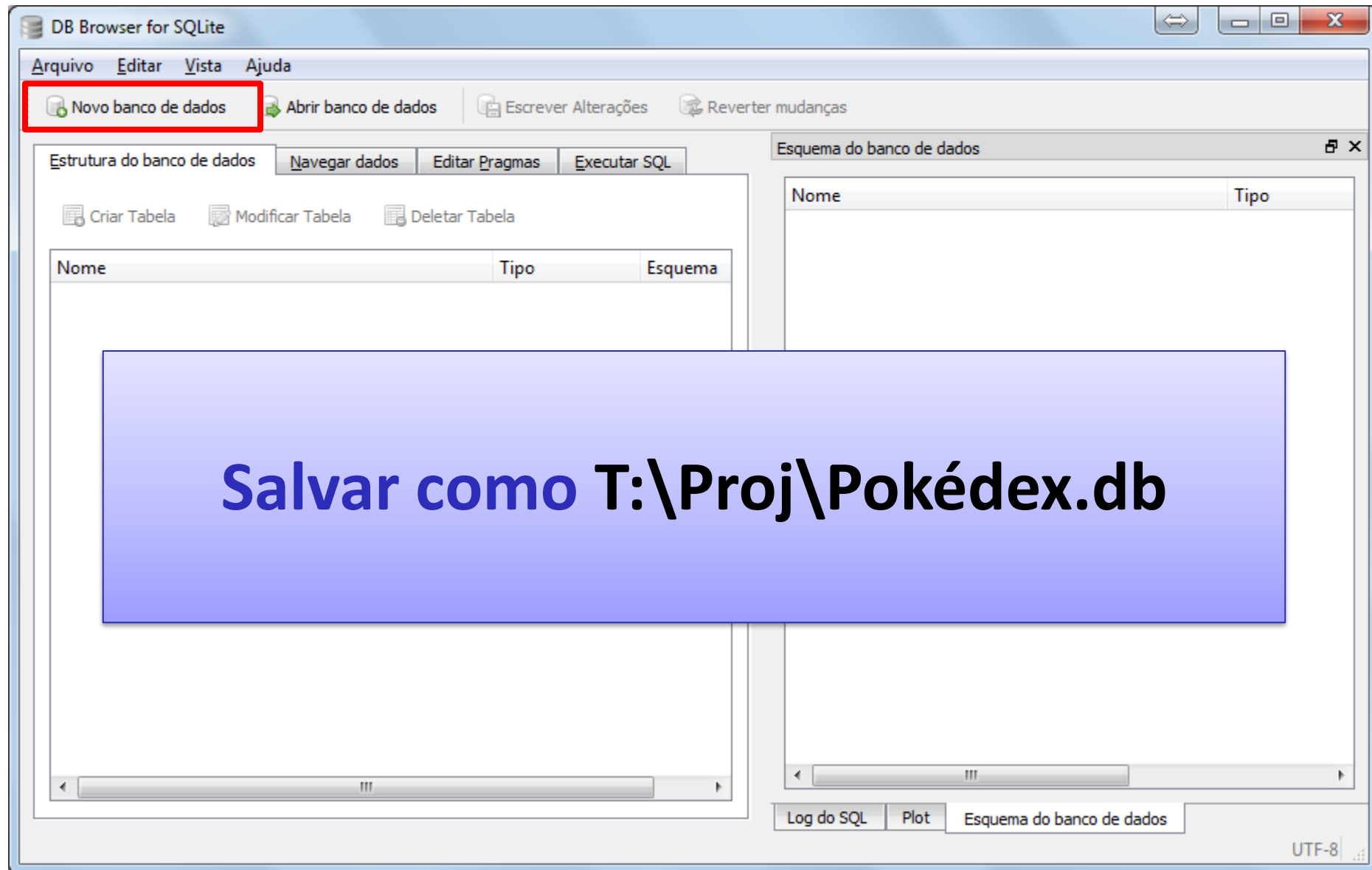
COMMIT: termina a transação atual;

ROLLBACK: aborta a transação atual.

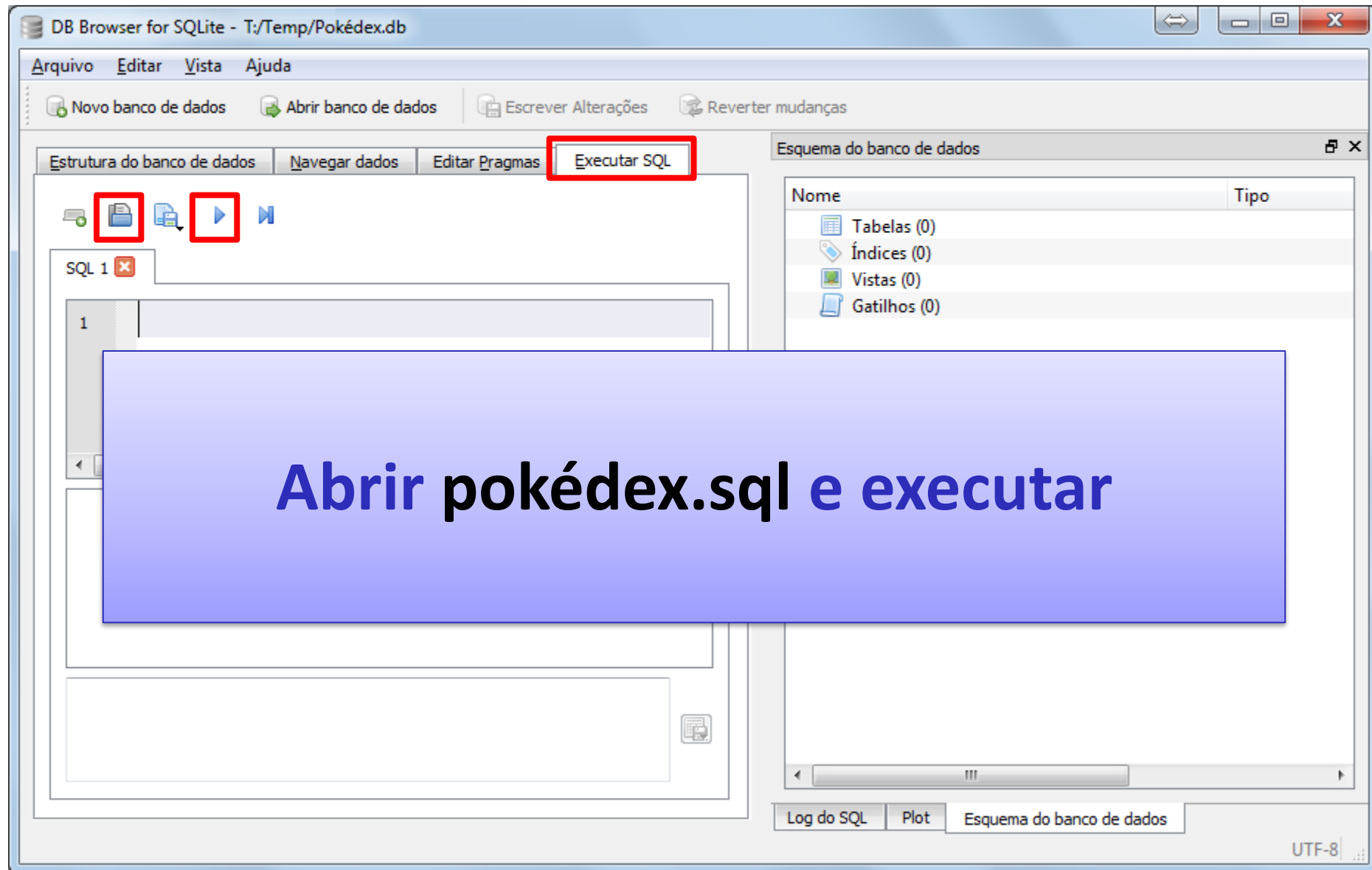
DB Browser for SQLite



DB Browser for SQLite



DB Browser for SQLite



Inserção de dados

A operação de inserção de dados é realizada em SQL pelo comando **INSERT INTO**;

Sintaxe:

```
INSERT INTO tabela  
  (campok, campoj,..., campom)  
VALUES  
  (valork, valorj, ... ,valorm);
```

```
INSERT INTO tabela  
VALUES (valor1, valor2, ... ,valorn);
```

Na **segunda forma**, **preenche-se obrigatoriamente os valores de todos os campos na ordem** que foram criados na tabela!

Na criação da tabela, em campos que se deseja um valor padrão utiliza-se a palavra **DEFAULT** como valor.

Exemplos de INSERT INTO

Exemplo 1: **INSERT INTO.** *Inserir dados de novos Pokémons.*

```
INSERT INTO pokédex  
VALUES (1, 'Bulbasaur', 'Grass', 'Poison', 45, 49, 49, 45);
```

```
INSERT INTO pokédex  
(idPokémon, Nome, Tipo1, Tipo2, HP, Ataque, Defesa, Velocidade)  
VALUES  
(2, 'Ivysaur', 'Grass', 'Poison', '60', '62', '63', '60');
```

Verificar mensagem de erro!!

Comando SELECT

Retorna dados de uma tabela, por meio de uma consulta (“*queries*”);

Permite executar **consultas** que **resultam** em seleção de: **campos, registros, agrupamento de tabelas;**

Opcionalmente podem envolver **comandos aritméticos e agregação** para campos calculados;

Sintaxe:

```
SELECT [ DISTINCT ] colunas
FROM tabelas
  WHERE < condições de busca >
    [ GROUP BY coluna [ HAVING < condição_busca >]]
    [ ORDER BY < lista_colunas >];
```

Exemplos de SELECT

Exemplo 02: Seleção sem critérios. “*Selecionar todos os registros da Pokédex*”:

```
SELECT * FROM pokédex;
```

O **símbolo** '*' indica no comando que se deseja recuperar todas as colunas da tabela original na tabela resultado.

Exemplos de SELECT

Exemplo 03: Seleção de um conjunto específico de colunas. “*Selecionar o id e o nome de todos os Pokémons*”:

```
SELECT idPokémon, Nome FROM pokédex;
```

Os nomes das colunas são **separados** por **vírgulas**;
O **resultado** será uma **tabela** com os dados referentes apenas às **colunas selecionadas**, na **ordem** em que foram **especificadas**.

Exemplos de SELECT

Exemplo 04: Seleção de um conjunto específico de colunas, renomeando-as. “Selecionar o número, nome e tipos de todos os Pokémons”:

```
SELECT  
  idPokémon AS Número,  
  Nome,  
  Tipo1,  
  Tipo2  
FROM pokédex;
```

O operador de renomeação de campo é o operador **AS**.

Exemplos de SELECT

Exemplo 05: Seleção de um conjunto distinto (sem repetição) de valores. “*Selecionar sem repetir os tipos de Pokémons*”:

```
SELECT DISTINCT Tipo1 FROM pokédex;
```

Mesmo que existam vários Pokémons associados a um mesmo tipo, o resultado apresentará apenas tipos distintos de Pokémons.

Exemplos de SELECT

Exemplo 06: Seleção com restrição. “*Listar o nome e o HP dos Pokémons com HP maior que 100*”:

```
SELECT Nome, HP  
FROM pokédex  
WHERE HP > 100;
```

A cláusula **WHERE** introduz uma restrição na consulta: apenas as linhas que atendem a condição de **WHERE** serão selecionadas;

Exemplos de SELECT

Exemplo 07: Seleção com restrição. “*Listar o nome e o tipo dos Pokémons onde o Tipo1 é igual a Grass*”:

```
SELECT Nome, Tipo1  
FROM pokédex  
WHERE Tipo1 = 'Grass';
```

Constantes de cadeias de caracteres são delimitadas por aspas simples.

Exemplos de SELECT

Exemplo 08: Seleção com restrições. “*Listar o nome e os tipos dos Pokémons onde o Tipo1 ou Tipo2 é igual a Fire*”:

```
SELECT Nome, Tipo1, Tipo2
FROM pokédex
WHERE (Tipo1 = 'Fire') OR (Tipo2 = 'Fire');
```

Quando se tem expressões complexas é interessante agrupá-las com **parênteses**.

Resumo dos operadores condicionais e operadores lógicos

Símbolo	Descrição
=	Igual
!=	Diferente de
<>	Diferente de
<	Menor que
>	Maior que
<=	Menor ou igual a
>=	Maior ou igual a
AND	E lógico
OR	Ou lógico
NOT	Negação lógica

Exemplos de SELECT

Exemplo 09: Cláusula BETWEEN. “*Listar o nome e o ataque dos Pokémons com Ataque entre 50 e 75*”:

```
SELECT Nome, Ataque  
FROM pokédex  
WHERE Ataque BETWEEN 50 AND 75;
```

A cláusula **BETWEEN** permite indicar um intervalo sobre o qual se deseja testar a pertinência de um valor; O intervalo é indicado por um valor menor, seguido de **AND** e pelo valor maior.

Exemplos de SELECT

Exemplo 10: Testando valores nulos. “*Listar o nome e o tipos de todos os Pokémon sem Tipo2 cadastrado*”:

```
SELECT Nome, Tipo1, Tipo2  
FROM pokédex  
WHERE Tipo2 IS NULL;
```

A constante **NULL** representa um valor não existente, isto é um campo não preenchido;
O teste de **nulidade** é realizado com o operador **IS**.

Exemplos de SELECT

Exemplo 11: Testando valores não nulos. “*Listar o nome e o tipos de todos os Pokémon com Tipo2 cadastrado*”:

```
SELECT Nome, Tipo1, Tipo2  
FROM pokédex  
WHERE Tipo2 IS NOT NULL;
```

A constante **NULL** representa um valor não existente, isto é um campo não preenchido;

O teste de **não nulidade** é realizado com o operador **IS NOT**.

Exemplos de SELECT

Exemplo 12: Testando a pertinência de um conjunto. “*Listar o nome e Tipo1 dos Pokémons cujo Tipo1 seja Electric ou Ground*”:

```
SELECT Nome, Tipo1  
FROM pokédex  
WHERE Tipo1 IN ('Electric','Ground');
```

O operador **IN** testa os valores do campo com o conjunto apresentado entre parênteses à sua direita; Serão selecionadas as linhas cujos valores de campo estejam dentro do conjunto testado.

Exemplos de SELECT

Exemplo 13: Busca aproximada de cadeia de caracteres. “*Listar o nome dos Pokémons cujo nome se inicia por ‘a’, independente da caixa (‘A’)*”:

```
SELECT Nome  
FROM pokédex  
WHERE Nome LIKE 'a%';
```

O operador **LIKE** executa um “casamento de padrão”, procurando determinar se uma cadeia atende ao padrão fornecido;

O símbolo de padrão '%' é substituído por zero ou mais caracteres – **ANSI SQL**.

Exemplos de SELECT

Exemplo 14: Busca aproximada de cadeia de caracteres. “*Listar o nome e o HP de todos os Pokémons cujos HP estejam na faixa de 40 a 49*”:

```
SELECT Nome, HP  
FROM pokédex  
WHERE HP LIKE '4_';
```

O símbolo de padrão '_' é substituído por um e único caractere em seu lugar - **ANSI SQL**.

Exemplos de SELECT

Exemplo 15: Campos calculados. *“Listar o nome, ataque e defesa dos Pokémons que possuem a média entre o ataque e defesa maior que 80”:*

```
SELECT Nome, Ataque, Defesa, (Ataque+Defesa)/2 AS Média  
FROM pokédex  
WHERE (Ataque+Defesa)/2 > 80;
```

Pode-se especificar como uma coluna de resultado uma expressão envolvendo constantes e/ou outros nomes de colunas.

Algo estranho acontece... verifique os resultados.

Resumo dos operadores aritméticos

Símbolo	Descrição
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da divisão

Concatenar strings:

- ❖ Em geral, no **SQL** utilize a função **CONCAT**;
- ❖ No **SQLite** utilize o operador **||**.

```
SELECT (idPokémon || ' - ' || Nome) AS Nome from pokédex;
```

Exemplos de SELECT

Exemplo 16: Subconsultas com **SOME**. “Listar o nome e o HP dos Pokémons do tipo Electric cujo HP é maior que o HP de algum Pokémon do tipo Grass”:

```
SELECT Nome, HP
FROM pokédex
WHERE Tipo1 = 'Electric' AND
HP > SOME(SELECT HP FROM pokédex WHERE Tipo1 = 'Grass');
```

O operador **SOME** é um operador **lógico** que retorna verdadeiro se o critério for verdadeiro para pelo menos um elemento testado.

Exemplos de SELECT

Exemplo 17: Subconsultas com **ALL** “Listar o nome e o HP dos Pokémons do tipo Normal cujo HP é maior que o HP de todos os Pokémons do tipo Grass”:

```
SELECT Nome, HP
FROM pokédex
WHERE Tipo1 = 'Normal' AND
HP > ALL(SELECT HP FROM pokédex WHERE Tipo1 = 'Grass');
```

O operador **ALL** é um operador **lógico** que retorna verdadeiro se o critério for verdadeiro para todos os elementos testados.

Exemplos de SELECT

Exemplo 18: Operações com conjuntos. “*Listar os nome e o Tipo1 dos Pokémons cujo nome tenha 4 letras, iniciado por ‘A’, ou o Tipo1 seja Water*”:

```
SELECT Nome, Tipo1
FROM pokédex
WHERE Nome LIKE 'A____'
UNION
SELECT Nome, Tipo1
FROM pokédex
WHERE Tipo1 = 'Water';
```

UNION é um operador **de conjunto** que retorna a união do resultado de duas consultas ou mais.

Exemplos de SELECT

Exemplo 19: Agrupamento de resultados. *“Listar a velocidade média dos Pokémons, agrupando-os por Tipo1 ordenando-os de forma decrescente”:*

```
SELECT Tipo1, AVG(Velocidade) AS 'Velocidade Média'  
FROM pokédex  
GROUP BY Tipo1  
ORDER BY AVG(Velocidade) DESC;
```

A cláusula **GROUP BY**, que agrupa as linhas resultantes de acordo com valores de uma determinada coluna, permite a criação de linhas de sumários de dados;

▪ **AVG** calcula a média em uma coluna.

▪ **DESC** = Decrescente e **ASC** = Crescente

Exemplos de SELECT

Exemplo 20: Agrupamento de resultados. *“Listar o maior e o menor HP dos Pokémons, por Tipo1”:*

```
SELECT  
  Tipo1,  
  MIN(HP) AS 'HP Mínimo',  
  MAX(HP) AS 'HP Máximo'  
FROM pokédex  
GROUP BY Tipo1;
```

MIN retorna o menor valor de um atributo;

MAX retorna o maior valor de um atributo;

SUM retorna a soma de um atributo;

AVG retorna a média de um atributo..

Exemplos de SELECT

Exemplo 21: Agrupamento de resultados. “*Listar o ataque médio dos Pokémons, por tipo, para aqueles tipos que possuem um ataque médio superior a 70*”:

```
SELECT Tipo1, AVG(Ataque) AS 'Ataque Médio'  
FROM pokédex  
GROUP BY Tipo1  
HAVING AVG(Ataque) > 70;
```

A cláusula **HAVING** é utilizada para **filtrar grupos calculados** a partir de **funções específicas** (funções agregadas) ou da condição de busca com a cláusula **GROUP BY**.

Exemplos de SELECT

Exemplo 22: Funções agregadas. “*Quantos Pokémons existem de cada tipo?*”:

```
SELECT Tipo1, COUNT(*) AS Quantidade  
FROM pokédex  
GROUP BY Tipo1;
```

COUNT(*) calcula e retorna a quantidade de *tuplas* (linhas) em uma tabela.

Exemplos de SELECT

Exemplo 23: Funções agregadas. “Qual são os Pokémons com a maior defesa?”:

```
SELECT Nome, Defesa  
FROM pokédex  
WHERE Defesa = (SELECT MAX(Defesa) FROM pokédex);
```

Pode-se comparar imediatamente o valor resultante de uma subconsulta que calcula como resultado um único valor.

Exemplos de INSERT INTO

Exemplo 24: INSERT INTO. “*Criar uma nova tabela e copiar os nomes e tipos dos Pokémons com um comando **SELECT***”.

```
CREATE TABLE pokédex_simplificada
(
    nome VARCHAR(40) NOT NULL,
    Tipo_principal VARCHAR(20) NOT NULL,
    Tipo_secundário VARCHAR(20),
    HP INT
);
```

```
INSERT INTO pokédex_simplificada
SELECT Nome, Tipo1, Tipo2, HP FROM pokédex;
```

```
SELECT * FROM pokédex_simplificada;
```

Eliminação de dados

A operação de eliminação de dados é realizada em SQL pelo comando **DELETE FROM**;

Sintaxe:

```
DELETE [FROM] {nome_tabela | nome_view}  
[WHERE condições]
```

Nota: Se a cláusula **WHERE** for omitida, todos os dados da tabela serão apagados!

Exemplos de DELETE

Exemplo 25: DELETE. *“Eliminar todos os Pokémons do Tipo Bug”:*

```
DELETE FROM pokédex_simplificada  
WHERE (Tipo_principal = 'Bug') OR (Tipo_secundário = 'Bug');
```

Alteração de dados

A operação de alteração de dados é realizada em SQL pelo comando **UPDATE**;

Sintaxe:

```
UPDATE {nome_tabela | nome_view}  
SET nome_coluna1 = {expressão1 | NULL | (comando_select)}  
    [, nome_coluna2 = ...]  
[WHERE condições_busca]
```

Exemplos de UPDATE

Exemplo 26: UPDATE. “Blastoise ataca Squirtle”:

```
SELECT * FROM pokédex_simplificada  
WHERE Nome IN ('Squirtle', 'Blastoise');
```

```
UPDATE pokédex_simplificada  
SET HP = HP - (  
  (SELECT Ataque FROM pokédex WHERE Nome = 'Blastoise')  
  -  
  (SELECT Defesa FROM pokédex WHERE Nome = 'Squirtle')  
)  
WHERE nome = 'Squirtle';
```

```
SELECT * FROM pokédex_simplificada  
WHERE Nome IN ('Squirtle', 'Blastoise');
```

Exemplos de UPDATE

Exemplo 27: UPDATE em várias colunas. *“Atualizar o Tipo1 do Pokémon de “Fire” para “Fogo” e adicionar o código “(E)” ao atributo Nome para indicar que a entidade foi alterada”:*

```
UPDATE pokédex_simplificada
SET
    Tipo_principal = 'Fogo',
    Nome = (Nome || ' (E)')
WHERE Tipo_principal = 'Fire';

SELECT * FROM pokédex_simplificada;
```


Exemplo 28: DROP TABLE. “Apagar a Pokedéx simplificada”.

```
DROP TABLE pokédex_simplificada;
```

