BANCO DE DADOS

Prof. Igor Moreira Félix
prof.igorfelix@usjt.br

SQL - Structured Query Language

Conjunto de declarações que são utilizadas para acessar os dados utilizando gerenciadores de banco de dados.

Pode ser utilizada para todas as atividades relativas a um banco de dados, podendo ser utilizada pelo administrador de sistemas, pelo DBA, por programadores, sistemas de suporte à tomada de decisões e outros usuários finais.

SQL - Structured Query Language

Pode ser dividida em:

- DDL (Data Definition Language - Linguagem de Definição de Dados)

  São os comandos que interagem com o **esquema** do banco.

  São comandos DDL: CREATE, ALTER e DROP

SQL - Structured Query Language

DDL:

- **Criação de banco de dados:**

  CREATE DATABASE < nome_db >;

- **Criação de tabela:**

  CREATE TABLE < nome_tabela > (
  nome_atributo1 < tipo > [NOT NULL],
  nome_atributo2 < tipo > [NOT NULL],
  .....
  nome_atributoN < tipo > [NOT NULL],
  PRIMARY KEY(nome_atributo)
) ;

SQL - Structured Query Language

DDL:

- **Exclusão de banco de dados:**

  DROP DATABASE <nome_banco_de_dados>;

- **Exclusão de tabela:**

  DROP TABLE <nome_tabela>;

- **Inserir/eliminar atributos nas tabelas:**

  ALTER TABLE < nome_tabela > ADD / DROP (
  nome_atributo1< tipo >[NOT NULL],
  nome_atributoN< tipo >[NOT NULL]
) ;

SQL - Structured Query Language

Principais tipos de dados:

	Tipo	Tamanho	Descrição
Número	INTEGER ou INT	16 bits	Inteiro
	SMALLINT	32 bits	
	BIGINT	64 bits	
	FLOAT ou REAL	32 bits	Ponto flutuante
	DOUBLE PRECISION	64 bits	
	DECIMAL(i, j) ou NUMERIC(i, j)	Variável	Precisão variável
	SERIAL	32 bits	Número sequencial (auto incremento)
Caractere	CHAR(n)	Variável	Cadeia de caracteres de tamanho fixo (coloca ' ' se necessário)
	VARCHAR(n)	Variável	Cadeia de caracteres de tamanho variável
	TEXT	Variável	Tamanho ilimitado
Data	TIMESTAMP	64 bits	Data e hora (microsegundo)
	DATE	32 bits	Data
	TIME	64 bits	Hora (microsegundo)
Binário	BLOB (BYTEA)	Variável	Binário

SQL - Structured Query Language

Pode ser dividida em:

- DML (Data Manipulation Language - Linguagem de Manipulação de Dados)

  São os comandos que **interagem com os dados** dentro das tabelas.

  São comandos DML: INSERT, DELETE e UPDATE

SQL - Structured Query Language

DML:

- **Inserir registros na tabela:**

```
INSERT INTO destino [(campo1[, campo2[, ...]])]
VALUES (valor1[, valor2[, ...]);
```

- **Atualizar registros da tabela:**

```
UPDATE tabela
SET campo1 = valornovo, ...
WHERE critério;
```

- **Excluir registros da tabela:**

```
DELETE FROM tabela
WHERE critério;
```

SQL - Structured Query Language

Pode ser dividida em:

- DQL - Data Query Language - Linguagem de Consulta de dados.
  São os comandos de consulta.
  São comandos DQL : SELECT (comando de consulta)

SQL - Structured Query Language

DQL:

SELECT *coluna1, coluna2, ...* FROM *nome_tabela* WHERE condição;

Obrigatórios:

- quais colunas devem ser obtidas
- quais as tabelas

Opcional:

- condição

SQL - Structured Query Language

DQL:

Condição:

- Expressão booleana que identifica quais linhas devem ser retornadas.

Operadores:

- Igual → =
- Menor → < ou <=
- Maior → > ou >=
- Diferente → <> ou !=
- Comparar com nulo → IS NULL ou IS NOT NULL

SQL - Structured Query Language

DQL:

BETWEEN:

- Consultas de valores entre um determinado intervalo (inclusive):

  SELECT * FROM funcionario WHERE salario BETWEEN 1000 AND 1500;

DISTINCT:

- Para evitar valores repetidos:

  SELECT DISTINCT cpf_responsavel FROM dependente

SQL - Structured Query Language

DQL:

- **Consulta em duas ou mais tabelas**

Importante definir condições de junção, as quais permitem combinar as linhas das tabelas.

```
SELECT cpf, nome_departamento
FROM funcionario, departamento
WHERE codigo_departamento = codigo;
```

SQL - Structured Query Language

DQL:

- **Apelidos para tabelas**

  Permite simplificar o uso do nome completo (caso ele seja longo) ou caso haja algum conflito de nome.

  ```
  SELECT f.cpf, d.nome_departamento
  FROM funcionario AS f, departamento AS d
  WHERE f.codigo_departamento = d.codigo;
  ```

SQL - Structured Query Language

DQL:

- **Apelidos para colunas**

  Também é possível usar apelidos para renomear uma coluna na saída.

```
SELECT cpf AS cpf_funcionario, nome_departamento AS departamento
FROM funcionario, departamento
WHERE codigo_departamento = codigo;
```

SQL - Structured Query Language

DQL:

- **Procura por subcadeias**

  No caso de colunas que são cadeias de caracteres, é possível comparar partes da cadeia usando o operador LIKE.

```
SELECT cpf
FROM funcionario
WHERE nome LIKE '%a%';
```

SQL - Structured Query Language

DQL:

- **Ordenação dos resultados**

  É possível ordenar os resultados da consulta usando o comando ORDER BY seguido pela coluna.

```
SELECT cpf
FROM funcionario
ORDER BY salario DESC;
```

DESC → para descendente, ou seja, decrescente

ASC → ascendente, ou seja, crescente