

Introdução à Linguagem SQL

Características do SQL.

Linguagens SQL abordadas neste curso:

Linguagem de Definição de Dados (DDL):

- A Linguagem de Definição de Dados (DDL) oferece comandos para a definição de esquemas de relação, exclusão de relações e modificações nos esquemas.

Linguagem de Manipulação de Dados (DML):

- A Linguagem de Manipulação de Dados (DML) engloba uma linguagem de consulta, comandos para inserir, excluir e modificar dados no Banco de Dados.

Características do SQL.

Instruções da **DDL**:

- **CREATE**: Instrução que cria um objeto (Banco de dados ou Tabela)
- **DROP**: Instrução que apaga um objeto do banco de dados.
- **ALTER**: Instrução que permite alterar um objeto, por exemplo, adicionar uma nova coluna à uma tabela existente.

Características do SQL.

Instruções da DML:

- **SELECT:** Permite especificar uma consulta com um resultado desejado.
- **INSERT:** Usada para inserir um registro em uma tabela existente.
- **UPDATE:** Usada para alterar os valores de dados em um registro da tabela especificada.
- **DELETE:** Permite remover registros existentes em uma tabela.

Características do SQL.

Alguns tipos de dados no SQL:

- **char(n)**: uma string de caracteres de tamanho fixo, com tamanho n;
- **varchar(n)**: uma string de caracteres de tamanho variável, com tamanho máximo n;
- **int**: um inteiro entre (-2.147.483.647 e 2.147.483.647);
- **smallint**: um inteiro entre (-32.768 e 32.767);

Características do SQL.

Chaves:

- **Chave Primária:** É um campo (ou conjunto de campos) em uma tabela que identifica de forma única cada registro, “tupla”, dessa tabela.
- **Chave Estrangeira:** É um campo (ou conjunto de campos) em uma tabela que cria um vínculo entre essa tabela e a chave primária de outra tabela.

Criar um banco de dados:

Instrução CREATE:

```
CREATE DATABASE nome_do_banco
```

Excluir um Banco de Dado

Instrução DROP:

USE nome_do_banco;

DROP DATABASE nome_do_banco;

Criar uma tabela e Chave Primária:

Instrução CREATE:

USE nome_banco;

CREATE TABLE nome_tabela(
id tipo de dado NOT NULL AUTO_INCREMENT PRIMARY KEY,

nome_campo tipo de dado,

nome_campo tipo de dado,

nome_campo tipo de dado,

);

Excluir uma tabela do Banco de Dado

Instrução DROP:

USE nome_do_banco;

DROP TABLE nome_tabela;

Criar uma tabela e Chave Estrangeira:

Instrução CREATE:

USE nome_banco;

CREATE TABLE nome_tabela(
id tipo de dado **NOT NULL AUTO_INCREMENT PRIMARY KEY**,

nome_chave_estrangeira **tipo de dado not null**,

nome_campo **tipo de dado**,

nome_campo **tipo de dado**,

foreign key(nome_do_campo) **references** nome_tabela(nome_campo)

);

Alterar uma tabela do banco:

Instrução ALTER TABLE:



Adicionar uma nova coluna:

USE nome_banco;

ALTER TABLE nome_tabela ADD COLUMN nome_coluna tipo_de_dado

Adicionar uma nova coluna depois de outra:

USE nome_banco;

ALTER TABLE nome_tabela ADD COLUMN nome_coluna tipo_de_dado **AFTER** nome_coluna

Excluir uma coluna:

USE nome_banco;

ALTER TABLE nome_tabela **DROP COLUMN** nome_coluna

Inserção de dados SQL

Agora que temos o banco pronto, é hora de inserir os dados.

Instrução INSERT INTO:

```
INSERT INTO nome_tabela (campo_string, campo_numerico, campo_data)
VALUES
('dado_string', dado_numerico, 'Y-m-d')
```

Alterar dados de uma tabela do banco:

Instrução UPDATE:

```
UPDATE nome_tabela SET nome_coluna = 'dado atualizado' WHERE  
nome_coluna = 'dado atual'
```

Excluir a linha “tupla” de uma tabela do banco:

Instrução DELETE:

DELETE FROM nome_tabela **WHERE** nome_campo = “valor”

Falar é fácil, quero ver fazer.

Agora que vocês dominam a arte da criação de um banco de dados, é o momento ideal para desenvolver um banco de dados denominado "escola".

A seguir, é necessário criar duas tabelas:

Tabela Curso: (id, nome, descricao, tempo_de_duracao, data_de_inicio)

Tabela Aluno: (id, curso_id, cpf, ra, nome, data_de_nascimento)

Vamos buscar os dados nas tabelas?



Instrução SELECT:

USE nome_banco;

Todos os dados = SELECT * FROM nome_tabela

Dados específicos = SELECT coluna, coluna FROM nome_tabela

Vamos buscar somente os alunos de cada curso?

Instruções **SELECT** e **INNER JOIN**:

USE nome_banco;

Todos os dados:

```
SELECT nome_tabela_1.*, nome_tabela_2.* FROM nome_tabela_1 INNER JOIN nome_tabela_2  
ON nome_tabela_1.chave_primária = nome_tabela_2.chave_estrangeira
```

Dados específicos:

```
SELECT nome_tabela_1.nome_coluna, nome_tabela_2.nome_coluna FROM nome_tabela_1  
INNER JOIN nome_tabela_2 ON nome_tabela_1.chave_primária = nome_tabela_2.chave_estrangeira
```

Alterando o nome das colunas ou tabelas no SELECT

Instrução AS “Alias”:

USE nome_banco;

SELECT nome_coluna **AS** apelido **FROM** nome_tabela

INNER JOIN:

SELECT apelido_1.*, apelido_2.* **FROM** nome_tabela_1 **AS** apelido_1 **INNER JOIN** nome_tabela_2 **AS** apelido_2 **ON** apelido_1.chave_primária = apelido_2.chave_estrangeira