

Introdução à linguagem SQL

1 Histórico

A linguagem de consulta estruturada (SQL - Structured Query Language) foi desenvolvida pela empresa IBM, no final dos anos 70. O primeiro banco de dados comercial a utilizar a linguagem SQL foi apresentado em 1979, pela empresa Oracle. No início dos anos 80, o Instituto Americano de Padronização (ANSI – American National Standards Institute) iniciou os trabalhos de desenvolvimento de uma versão padronizada do SQL, que foi publicada nos anos de 1986 e 1987. Como a SQL ela é padronizada pelo ANSI, diversos fabricantes de bancos de dados sentiram-se seguros em adotá-la como padrão para seus produtos.

2 Comandos para a manipulação de bases de dados

2.1 Criação

O comando utilizado para criar um banco de dados é o **CREATE DATABASE**, que pode ser utilizado conforme o exemplo a seguir:

```
CREATE DATABASE biblioteca;
```

Ao executar esse comando, será criado um banco de dados com o nome **biblioteca**.

2.2 Comando para acessar o banco

Após a criação do banco de dados, você pode acessá-lo por meio do comando **USE DATABASE**, conforme a sintaxe:

```
USE biblioteca;
```

2.3 Exclusão do banco

Para excluir o banco de dados, use o comando **DROP DATABASE 'banco'**;

Exemplo:

```
DROP DATABASE 'biblioteca';
```

2.4 Exclusão de tabelas

Para excluir tabelas, utilize o comando **DROP TABLE**.

```
DROP TABLE `autor`, `obra`, `obra_autor`;
```

3 Comandos para manipulação das tabelas

3.1 Criação

A sintaxe básica para a criação de uma tabela é:

```
CREATE TABLE nome_da_tabela(  
    'nome_do_campo1' tipo(tamanho) REGRAS,  
    'nome_do_campo2' tipo(tamanho) REGRAS,  
    'nome_do_campoN' tipo(tamanho) REGRAS,  
    PRIMARY KEY('nome_do_campo_chave')  
)  
ENGINE=MecanismoInternodeGestãoodaTabela  
AUTO_INCREMENT=NúmeroInicialdoAutoincremento DEFAULT  
CHARSET=IdiomaPadrão
```

Exemplo:

```
CREATE TABLE `autor` (  
    `id_autor` int(11) NOT NULL AUTO_INCREMENT,  
    `nome` varchar(45) NOT NULL,  
    PRIMARY KEY (`id_autor`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1
```

3.2 Alteração

Você pode utilizar o comando **ALTER TABLE** para:

- a) adicionar campos;
- b) remover campos;
- c) alterar propriedades de um campo;
- d) alterar o auto-incremento;

3.2.1 Adicionando campos

Sintaxe básica

```
ALTER TABLE `banco`.`tabela` ADD COLUMN `campo` tipo(tamanho)  
REGRAS POSICAO;
```

Exemplo:

```
ALTER TABLE `biblioteca`.`autor` ADD COLUMN `fone` VARCHAR(45)
NULL AFTER `nome` ;
```

3.2.2 Removendo campos

Sintaxe básica

```
ALTER TABLE `banco`.`tabela` DROP COLUMN `campo` ;
```

Exemplo:

```
ALTER TABLE `biblioteca`.`autor` DROP COLUMN `nome` ;
```

3.2.3 Alterando as propriedades de um campo

Sintaxe básica

```
ALTER TABLE `tabela`.`campo` CHANGE COLUMN `campo` `campo`
tipo(tamanho) REGRAS;
```

Exemplo:

```
ALTER TABLE `biblioteca`.`autor` CHANGE COLUMN `nome` `nome`
VARCHAR(100) NOT NULL;
```

3.2.4 Alterando o auto-incremento

Sintaxe básica

```
ALTER TABLE `banco`.`tabela` AUTO_INCREMENT = VALOR;
```

Exemplo:

```
ALTER TABLE `biblioteca`.`autor` AUTO_INCREMENT = 10;
```

3.2.5 Exclusão

Para excluir uma tabela, utilize o comando:

```
DROP TABLE 'nome_da_tabela'.'nome_do_banco';
```

Exemplo:

```
DROP TABLE `biblioteca`.`autor`;
```

3.2.6 Limpeza

Para excluir apenas os dados de uma tabela, sem excluir a tabela em si, utilize o comando:

```
TRUNCATE TABLE 'nome_do_banco'. 'nome_da_tabela';
```

Exemplo:

```
TRUNCATE TABLE `biblioteca`.`autor`;
```

4 Comandos para manipulação dos dados

4.1 Inclusão

Para incluir registros, utilize o comando INSERT, com a seguinte sintaxe:

```
INSERT INTO `banco`.`tabela`(`campo1`,`campo2`,`campoN`) VALUES ('valor1', 'valor2', 'valorN');
```

Exemplo:

```
INSERT INTO `biblioteca`.`tipo_obra` (``,`descricao`) VALUES ('Livro');
```

```
INSERT INTO `biblioteca`.`tipo_obra` (``,`descricao`) VALUES ('DVD');
```

```
INSERT INTO `biblioteca`.`tipo_obra` (``,`descricao`) VALUES ('CD');
```

```
INSERT INTO `biblioteca`.`tipo_obra` (``,`descricao`) VALUES ('VHS');
```

4.2 Seleção

4.2.1 Selecionando registros em uma tabela

Para selecionar registros em uma tabela, utilize o comando SELECT

4.2.1.1 Selecionando todos os campos

Sintaxe: SELECT * FROM `tabela`;

Exemplo: SELECT * FROM `obra`;

4.2.1.2 Escolhendo os campos a serem exibidos

Sintaxe: SELECT tabela.campo1, tabela.campo2 FROM tabela;

Exemplo: SELECT obra.titulo, obra.tipo FROM `obra`;

4.2.1.3 Escolhendo apelidos para os campos

```
SELECT tabela.campo1 AS apelido1, tabela.campo2 AS apelido2  
FROM `tabela`;
```

```
SELECT obra.titulo AS meu_titulo, obra.tipo AS meu_tipo FROM `obra`;
```

4.2.1.4 Ordenando os registros

```
SELECT * FROM tabela ORDER BY campo;
```

```
SELECT * FROM obra ORDER BY obra.titulo;
```

```
SELECT * FROM obra ORDER BY obra.titulo DESC;
```

```
SELECT * FROM obra ORDER BY obra.tipo,obra.titulo;
```

```
SELECT * FROM obra ORDER BY obra.tipo, obra.titulo DESC;
```

```
SELECT * FROM obra ORDER BY obra.tipo DESC, obra.titulo;
```

4.2.1.5 Selecionado registros em duas tabelas

Exemplo com uso da cláusula WHERE

```
SELECT obra.id_obra, obra.titulo, tipo_obra.dsc_tipo FROM obra, tipo_obra WHERE tipo_obra.id_tipo = obra.tipo;
```

Exemplo com uso da cláusula JOIN

```
SELECT obra.id_obra, obra.titulo, tipo_obra.descricao FROM obra INNER JOIN tipo_obra ON tipo_obra.id_tipo = obra.tipo;
```

4.2.1.6 Selecionando registros em três tabelas

Exemplo com uso da cláusula JOIN

```
SELECT obra_autor.id_autor, autor.nome, obra_autor.id_obra, obra.titulo FROM obra_autor INNER JOIN autor ON autor.id_autor = obra_autor.id_autor INNER JOIN obra ON obra.id_obra = obra_autor.id_obra;
```

Exemplo com uso da cláusula WHERE

```
SELECT obra.titulo, autor.nome FROM obra, autor, obra_autor WHERE obra_autor.id_autor = autor.id_autor AND obra_autor.id_obra = obra.id_obra;
```

4.3 Alteração

Para alterar dados em uma tabela, utilize o comando UPDATE, com a seguinte sintaxe:

```
UPDATE `banco`.`tabela` SET `campo` = 'novo valor' WHERE `campo` = 'regra';
```

Exemplos:

```
UPDATE `biblioteca`.`obra` SET `tipo`='5' WHERE `id_obra`='4';
```

```
UPDATE `biblioteca`.`obra` SET `tipo`='2' WHERE `id_obra`='3';
```

4.4 Exclusão

Para excluir dados de uma tabela, utilize o comando DELETE, com a seguinte sintaxe:

```
DELETET FROM `banco`.`tabela` WHERE `campo` = 'regra';
```

Exemplo:

```
DELETE FROM `biblioteca`.`obra` WHERE `id_obra`='4';
```