

INSERT: adicionando registros a uma tabela;

UPDATE: atualizando os registros já inseridos;

DELETE: excluindo registros de uma tabela;

SELECT: retomando registros na tabela;

CREATE: criando novas tabelas em um banco de dados;

ALTER: alterando uma tabela já criada;

DROP: excluindo uma tabela do banco de dados;

GRANT: permitindo acesso a objetos do banco de dados;

REVOKE: removendo o acesso a objetos do banco de dados;

DENY: bloqueando o acesso para objetos e usuários específicos.

DML: consultando e modificando os registros da tabela!

O DML (Data Manipulation Language) é um subconjunto da linguagem SQL que comporta os comandos utilizados para manipular diretamente os dados. Ou seja, por meio deles, é possível realizar ações como inserir informações em uma tabela, realizar consultas, deletar e alterar os registros.

Veja mais detalhes sobre esses comandos abaixo!

INSERT: adicionando registros a uma tabela

O comando INSERT é usado para inserir dados em uma tabela do banco. Para realizar essa ação, podemos usar a seguinte sintaxe:

```
INSERT into estudantes (id, nome, curso) values (23, 'Rafael', 'Desenvolvimento de Software');
```

Perceba que, na sintaxe usada, a palavra “estudantes” se refere ao nome da tabela e os termos “id”, “nome” e “curso” são os campos dessa tabela que vão receber novas informações.

UPDATE: atualizando os registros já inseridos

O comando UPDATE é o responsável por fazer edições em registros que já constam no banco. Essa instrução é muito importante já que ela permite corrigir ou complementar os dados, garantindo que o banco tenha sempre informações atualizadas.

A sintaxe usada nesse comando é:

```
UPDATE estudantes SET nome = 'Rafael Rodrigues Maia' WHERE id = 23;
```

Observe que, com a instrução acima, a tabela “estudantes” sofreu uma atualização, na qual o registro com id = 23 recebeu no campo “nome” o valor “Rafael Rodrigues Maia”.

DELETE: excluindo registros de uma tabela

Por sua vez, o comando DELETE é utilizado para excluir informações da nossa base de dados. A sintaxe para realizar essa ação é bastante simples:

```
DELETE FROM estudantes WHERE id = 23;
```

Entenda que, após rodar esse comando, todas as informações referentes ao registro que possui id = 23 serão deletadas da tabela “estudantes”. Já, para apagar todos os registros de uma tabela de uma vez, usamos a seguinte instrução:

```
DELETE FROM estudantes;
```

É importante explicar que o comando acima apaga apenas as informações armazenadas na tabela, ou seja, a estrutura, os índices e atributos do objeto continuam intactos no banco.

SELECT: retomando registros na tabela

Finalmente, o SELECT é um dos comandos SQL mais importantes, pois com ele podemos elaborar diversas consultas aos registros da nossa base de dados. É possível, por exemplo, fazer uma pesquisa que retornará todos os campos de uma tabela dessa forma:

```
SELECT * FROM estudantes;
```

Perceba que, nessa consulta, o caractere * é usado para indicar que os dados de todas as colunas da tabela “estudantes” devem ser retornados na pesquisa. Por isso, para fazer uma consulta mais específica, os campos desejados devem ser informados da seguinte maneira:

```
SELECT nome, curso FROM estudantes;
```

Além disso, utilizando a cláusula WHERE, podemos fazer uma consulta ainda mais refinada. Veja:

```
SELECT nome FROM estudantes WHERE curso = 'Desenvolvimento de Software';
```

Como você deve ter notado, com o comando acima serão retornados apenas os nomes de estudantes que frequentam o curso de Desenvolvimento de Software. Ademais, caso queira uma lista ordenada, é possível utilizar a cláusula ORDER BY para que os dados sejam exibidos em ordem ascendente. Observe:

```
SELECT nome FROM estudantes ORDER BY nome;
```

DDL: criando, alterando e excluindo tabelas de registros!

Os comandos do tipo DDL (Data Definition Language) são utilizados para gerenciar a estrutura do banco de dados. Sendo assim, eles nos permitem alterar a estrutura de um objeto do banco, além de serem usados para criar e excluir tabelas.

Nos tópicos seguintes, conheça os principais comandos SQL que se encaixam nessa categoria!

CREATE: criando novas tabelas em um banco de dados

O comando CREATE é usado sempre que precisamos criar novos objetos na base de dados. Primeiramente, essa instrução pode ser utilizada para criar um banco completamente vazio, por exemplo:

```
CREATE DATABASE escola;
```

Já utilizando o comando CREATE TABLE, podemos criar novas tabelas dentro de um banco de dados. Observe:

Comando SQL CREATE TABLE para criar novas tabelas em um banco de dados.

Repare que, ao criar a tabela, também especificamos quais colunas ela terá e os tipos de dados que cada coluna pode receber. Perceba ainda que, nas colunas de tipo VARCHAR, também foi especificado a quantidade máxima de caracteres que cada campo vai comportar.

Outra observação importante é a utilização da especificação PRIMARY KEY ao definirmos o campo id. Isso indica que essa coluna é uma chave-primária, ou seja, o seu conteúdo é único e representa a identificação do registro. A chave-primária é utilizada para facilitar a manipulação dos dados e para criar o relacionamento entre as tabelas.

ALTER: alterando uma tabela já criada

Por sua vez, o comando ALTER é utilizado quando precisamos alterar a estrutura de um objeto que já existe na base de dados. Um exemplo da aplicação dessa instrução é:

```
ALTER TABLE estudantes ADD idade INT;
```

No exemplo mostrado, estamos adicionando a coluna “idade” na tabela “estudantes” e definindo que ela receberá dados do tipo INT. Além disso, o comando também pode ser usado da seguinte forma:

Comando sql ALTER TABLE com MODIFY para alterar o tipo de dados que uma coluna poderá receber

Nesse caso, estamos usando o comando ALTER TABLE junto com a instrução MODIFY para alterar o tipo de dados que a coluna “email” poderá receber.

DROP: excluindo uma tabela do banco de dados

Por fim, o comando DROP é o responsável por deletar objetos da nossa base de dados. Utilizamos a sintaxe abaixo para realizar essa ação:

DROP TABLE estudantes;

É interessante ressaltar que ele também pode ser usado para excluir o banco de dados completo. Para isso, inserimos a instrução dessa forma:

DROP DATABASE escola;

Em ambos os casos, é possível notar que as sintaxes são bem simples, bastando acrescentar após o comando o nome da tabela ou do banco que deve ser excluído.

DCL: definindo permissões aos objetos do banco de dados!

Entre os comandos SQL, também existem os que fazem parte do subconjunto DCL (Data Control Language). Essa categoria é muito importante, pois é por meio dessas instruções que a pessoa programadora controla os privilégios e o nível de acesso dos objetos do banco.

Na sequência, veja quais são esses comandos e como eles funcionam!

GRANT: permitindo acesso a objetos do banco de dados

O comando GRANT é utilizado para conceder privilégios dentro do banco de dados. Isso significa que, com ele, podemos autorizar que determinadas ações sejam executadas apenas por pessoas específicas. Por exemplo:

GRANT SELECT, INSERT, UPDATE ON estudantes TO Maria;

No exemplo acima, observe que estamos autorizando a pessoa cadastrada com usuário “Maria” a fazer consultas, inserções e atualizações na tabela “estudantes”. Outro exemplo que podemos dar para o uso dessa instrução é:

```
GRANT SELECT ON estudantes TO public;
```

Com o comando mostrado, repare que as consultas na tabela “estudantes” serão definidas como públicas.

REVOKE: removendo o acesso a objetos do banco de dados

Como você deve imaginar, com o comando REVOKE podemos revogar os privilégios que foram concedidos anteriormente. A sintaxe usada nesse caso é:

```
REVOKE SELECT ON estudantes FROM Maria;
```

Após rodar esse comando, a usuária “Maria” não terá mais permissão para realizar consultas na tabela “estudantes”. No entanto, os outros privilégios recebidos serão mantidos.

DENY: bloqueando o acesso para objetos e usuários específicos

Finalmente, utilizando o comando DENY podemos negar explicitamente uma ou mais permissões a um objeto do banco. Um exemplo simples de aplicação desse código que podemos mostrar é:

```
DENY SELECT ON estudantes TO João;
```

Você já deve imaginar que aqui estamos negando especificamente ao usuário “João” o acesso para fazer consultas na tabela “estudantes”, certo? No mais, é interessante lembrar que as permissões negadas com DENY também podem ser removidas com o comando REVOKE.

Como foi possível perceber, os comandos SQL são simples e intuitivos, o que facilita bastante a aprendizagem. No entanto, lembre-se que esse é apenas o início da jornada e, para ter sucesso na área de desenvolvimento, é preciso buscar cada vez mais conhecimento para ampliar suas hard skills!