**Para saber mais: a combinação avançada dos comandos INSERT e SELECT em SQL**

Duas das instruções mais fundamentais em SQL são INSERT e SELECT. Individualmente, esses comandos são bastante diretos, sendo responsáveis por inserir dados em uma tabela (INSERT) e recuperar dados de uma ou mais tabelas (SELECT). Mas você sabia que é possível combinar estes dois comandos para realizar tarefas mais complexas?

A combinação do INSERT com SELECT permite inserir dados em uma tabela baseados em uma seleção de dados de outra ou da mesma tabela. Esta combinação é bastante útil quando desejamos duplicar dados, migrar informação entre tabelas ou criar backups. Vamos realizar um aprofundamento no uso combinado desses comandos para melhor entendimento.

A sintaxe básica da combinação de INSERT e SELECT é:

**INSERT** **INTO** tabela\_destino (coluna1, coluna2, ... )

**SELECT** coluna1, coluna2, ...

**FROM** tabela\_origem

**WHERE** condição;

Aqui, tabela\_destino é a tabela onde você quer inserir os dados e tabela\_origem é a tabela onde os dados estão originalmente. As coluna1, coluna2,... precisam ser as mesmas tanto na tabela de destino quanto na de origem. A cláusula WHERE é opcional e permite especificar condições para a seleção dos dados.

## Exemplos diversificados e complexos

1 - **Duplicação de dados de uma tabela**

Suponha que temos uma tabela chamada Alunos, com as colunas ID, nome e idade, e queremos copiar todos os dados para uma nova tabela chamada Backup\_Alunos. Para isso, usamos o seguinte comando:

**INSERT** **INTO** Backup\_Alunos (ID, nome, idade)

**SELECT** ID, nome, idade

**FROM** Alunos;

2 - **Transferência de dados entre tabelas**

Supondo agora que temos duas tabelas, Alunos e Ex\_Alunos, ambas com as colunas ID, nome e idade. Queremos mover os alunos com mais de 20 anos para a tabela Ex\_Alunos. Para isso, primeiramente, vamos inserir esses alunos na tabela Ex\_Alunos:

**INSERT** **INTO** Ex\_Alunos (ID, nome, idade)

**SELECT** ID, nome, idade

**FROM** Alunos

**WHERE** idade > 20;

Em seguida, precisaríamos remover esses alunos da tabela Alunos, o que é feito com o comando DELETE:

**DELETE** **FROM** Alunos

**WHERE** idade > 20;

3 - **Inserção condicional de dados**

Imagine que temos uma tabela de Vendas, com as colunas ID\_venda, ID\_produto e quantidade, e uma tabela de Estoque, com as colunas ID\_produto e quantidade. Queremos diminuir a quantidade em estoque dos produtos vendidos. Para isso, poderíamos usar a seguinte combinação de INSERT, SELECT e UPDATE:

**INSERT** **INTO** Estoque (ID\_produto, quantidade)

**SELECT** Vendas.ID\_produto, Estoque.quantidade - Vendas.quantidade

**FROM** Vendas, Estoque

**WHERE** Vendas.ID\_produto = Estoque.ID\_produto;

Esses são apenas alguns exemplos do que você pode fazer com a combinação de INSERT e SELECT. A PostgreSQL, MySQL, SQLite e outros sistemas de gerenciamento de bancos de dados SQL possuem suporte a esta combinação, abrindo um universo de possibilidades em manipulação de dados. Agora é com você: aproveite o que aprendeu para explorar ainda mais o SQL!