

# <SQL>



## { SQL

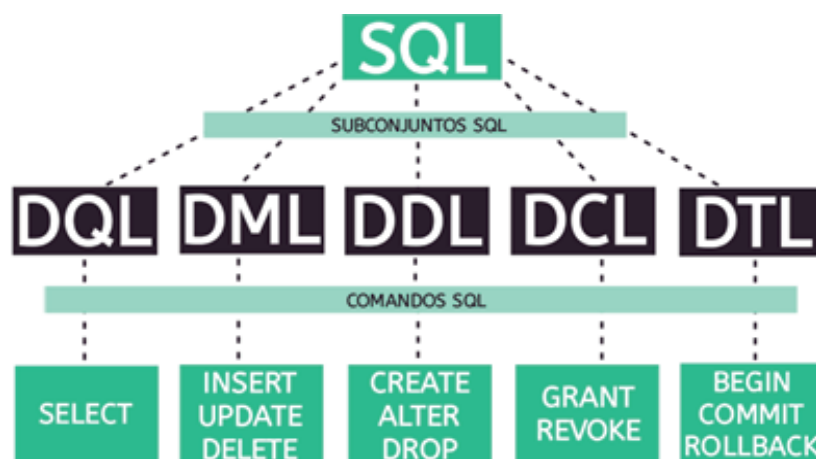
*Structured Query Language* (SQL, em inglês), ou Linguagem de Consulta Estruturada, é a linguagem de pesquisa declarativa padrão para banco/base de dados relacional. A SQL é usada para qualquer tipo de manipulação de registros de um banco de dados, o que permite criar, inserir, atualizar, excluir ou consultar informações armazenadas na base, além de outras funções mais complexas.

Vários Sistemas Gerenciadores de Bancos de Dados (SGBD) amplamente utilizados no mercado hoje são baseados em SQL: bancos do tipo relacional armazenam informações em estruturas semelhantes à tabelas de forma a permitir relações entre elas - a linguagem foi criada justamente para facilitar o gerenciamento de dados armazenados em bancos que seguem esse padrão.

Alguns dos principais sistemas que utilizam essa tecnologia são:

- **MySQL:** criado pela Oracle, oferece serviços gratuitos e pagos;
- **PostgreSQL:** muito usado em aplicações web, é uma opção gratuita de código aberto;
- **Oracle:** conhecido pela segurança, é um dos sistemas mais usados por grandes corporações;
- **SQL Server:** desenvolvido pela Microsoft, oferece tanto serviços pagos quanto versões gratuitas para download.

A linguagem SQL pode ser dividida em cinco subconjuntos: DQL, DML, DDL, DCL e DTL, ilustrados abaixo.



## { Linguagem de Manipulação de Dados (DML)

DML (*Data Manipulation Language*, ou Linguagem de Manipulação de Dados) é o subconjunto da SQL que define comandos usados para manipular dados armazenados em um banco. É um dos conjuntos mais utilizados, já que fornece operadores que permitem inserir, excluir ou alterar registros de uma tabela, por exemplo. Os comandos mais importantes desse subconjunto são *INSERT*, *DELETE* e *UPDATE*.

## { Linguagem de Consulta de Dados (DQL)

DQL (*Data Query Language*, ou Linguagem de Consulta de Dados) é o conjunto SQL que define o comando mais popular da linguagem, o *SELECT*. Esse comando é essencial para consultar dados armazenados em um banco.

## { Linguagem de Definição de Dados (DDL)

DDL (*Data Definition Language*, ou Linguagem de Definição de Dados) é o subconjunto SQL que fornece comandos usados para gerenciar estruturas do banco de dados. Com ele, podemos criar, atualizar e remover objetos da base, como tabelas e índices. Os comandos definidos pelo DDL são *CREATE*, *DROP* e *ALTER*.

## { Linguagem de controle de dados (DCL)

DCL (*Data Control Language*, ou Linguagem de Controle de Dados) é o subconjunto que fornece comandos para controlar o acesso aos dados de base. Com ele, conseguimos estabelecer restrições e permissões para o usuário que acessar o banco por meio dos comandos *GRANT* e *REVOKE*.

## { Linguagem de Transação de Dados (DTL ou TCL)

DTL (*Data Transaction Language*) - também conhecido como TCL (*Transaction Control Language*) - é o subconjunto SQL que define comandos quando se faz necessário gerenciar transações feitas em um banco. Ele permite iniciar, confirmar ou desfazer determinadas alterações. Os comandos estabelecidos pelo conjunto são *COMMIT*, *BEGIN* e *ROLLBACK*.

## { cláusulas de consulta

A sintaxe básica para uma consulta está representada abaixo:

```
SELECT * FROM nome_da_tabela;  
WHERE condição;
```

Os principais comandos utilizados são:

- **SELECT:** comando de consultas utilizado toda vez que o usuário precisa visualizar algum dado. Ele busca a linha dentro de uma tabela de acordo com o critério previamente definido na cláusula *WHERE*;
- **WHERE:** cláusula usada para filtrar registros que especifica critérios que valores do campo precisam cumprir para que seus registros sejam incluídos nos resultados de consulta;
- **SELECT DISTINCT:** instrução para retornar apenas valores distintos;
- **INSERT:** comando usado para inserir novas linhas dentro de uma tabela de acordo com os argumentos transmitidos a ele;
- **DELETE:** comando usado para remover linhas de uma tabela de acordo com critério pré-definido;
- **UPDATE:** comando usado para atualizar linhas de uma tabela de acordo com critério previamente definido na cláusula *WHERE*.

## { Funções de agregação

As funções de agregação reúnem múltiplos valores em um único resultado. As mais populares são:

COUNT	Conta a quantidade de registros ou de valores em uma coluna
SUM	Somatório de valores de um campo numérico
MAX	Valor máximo de uma determinada coluna
MIN	Valor mínimo de uma determinada coluna
AVG	Média dos valores em um campo numérico

## { Operadores relacionais

Operadores relacionais são sinais que realizam comparações entre valores, estruturas ou controles. Os mais populares são:

>	maior
<	menor
>=	maior ou igual
<=	menor ou igual
=	igual
<>	diferente

## { Operadores lógicos

Operadores lógicos validam condições ou testam sua legitimidade retornando os valores VERDADEIRO (**TRUE**), FALSO (**FALSE**) ou DESCONHECIDO (**UNKNOWN**).

Os mais populares são:

Operador	Significado
ALL	TRUE quando tudo em um conjunto de comparações for TRUE
AND	TRUE quando duas expressões booleanas forem TRUE
ANY	TRUE quando qualquer conjunto de comparações for TRUE
BETWEEN	TRUE quando o operando estiver dentro de um intervalo
EXISTS	TRUE se a subconsulta possuir qualquer linha
IN	TRUE se o operador for igual ao de uma lista de expressões
LIKE	TRUE se o operando corresponder a um padrão
NOT	Inverte o valor de qualquer outro operador booleano
OR	TRUE se qualquer expressão booleana for TRUE
SOME	TRUE quando um conjunto de comparações for TRUE

## { União de tabelas via relação entre colunas

SQL *JOIN* permite combinar linhas de várias tabelas com base na relação entre as colunas delas.

Os formatos de combinação se dão conforme abaixo:

- ***INNER JOIN***: retorna registros com seus respectivos valores em ambas as tabelas;
- ***LEFT JOIN***: retorna os registros da tabela esquerda e aqueles com valores correspondentes na tabela direita;
- ***RIGHT JOIN***: retorna todos os registros da tabela da direita e aqueles com valores correspondentes na tabela esquerda;
- ***FULL JOIN***: retorna os registros de ambas as tabelas em caso de correspondência entre elas.



## <Exercícios>

- ➔ 1. Crie a tabela abaixo em seu banco de dados e insira os valores apresentados.

ID_NF	ID_ITEM	COD_PROD	VALOR_UNIT	QUANTIDADE	DESCONTO (%)
1	1	1	25,00	10	5
1	2	2	13,50	3	
1	3	3	15,00	2	
1	4	4	10,00	1	
1	5	5	30,00	1	
2	1	3	15,00	4	
2	2	4	10,00	5	
2	3	5	30,00	7	
3	1	1	25,00	5	
3	2	4	10,00	4	
3	3	5	30,00	5	
3	4	2	13,50	7	
4	1	5	30,00	10	15
4	2	4	10,00	12	5
4	3	1	25,00	13	5
4	4	2	13,50	15	5
5	1	3	15,00	3	
5	2	5	30,00	6	
6	1	1	25,00	22	15
6	2	3	15,00	25	20
7	1	1	25,00	10	3
7	2	2	13,50	10	4
7	3	3	15,00	10	4
7	4	5	30,00	10	1

Em seguida, escreva as consultas abaixo (valores em branco devem ser nulos no banco de dados):

- Pesquise itens vendidos sem desconto. As colunas do resultado são ID\_NF, ID\_ITEM, COD\_PROD e VALOR\_UNIT;
- Pesquise itens vendidos com desconto. As colunas do resultado são ID\_NF, ID\_ITEM, COD\_PROD, VALOR\_UNIT e VALOR VENDIDO (o valor vendido é igual ao VALOR\_UNIT  $(\text{VALOR\_UNIT} * (\text{DESCONTO}/100))$ );
- Altere o valor do desconto para zero em todos os registros onde o campo for nulo;
- Pesquise os itens vendidos. As colunas do resultado são ID\_NF, ID\_ITEM, COD\_PROD, VALOR\_UNIT, VALOR\_TOTAL, DESCONTO e VALOR\_VENDIDO (VALOR\_TOTAL pode ser obtido pela fórmula  $\text{QUANTIDADE} * \text{VALOR\_UNIT}$ . VALOR\_VENDIDO é igual a  $\text{VALOR\_UNIT} - (\text{VALOR\_UNIT} * (\text{DESCONTO}/100))$ );



**e.** Pesquise o valor total das NFs e ordene o resultado do maior valor para o menor. As colunas do resultado são ID\_NF e VALOR\_TOTAL (VALOR\_TOTAL é obtido pela fórmula  $\Sigma \text{QUANTIDADE} * \text{VALOR\_UNIT}$ ). Agrupe o resultado da consulta por ID\_NF;

**f.** Pesquise o valor vendido das NFs e ordene o resultado do maior valor para o menor. As colunas do resultado são ID\_NF e VALOR\_VENDIDO (VALOR\_TOTAL é obtido pela fórmula  $\Sigma \text{QUANTIDADE} * \text{VALOR\_UNIT}$ . VALOR\_VENDIDO é igual a  $\Sigma \text{VALOR\_UNIT} - (\text{VALOR\_UNIT} * (\text{DESCONTO}/100))$ ). Agrupe o resultado por ID\_NF;

**g.** Consulte o produto que mais vendeu. As colunas do resultado são COD\_PROD e QUANTIDADE. Agrupe o resultado por COD\_PROD;

**h.** Consulte NFs com mais de 10 unidades vendidas de pelo menos um produto. As colunas do resultado são ID\_NF e COD\_PROD e QUANTIDADE. Agrupe o resultado da consulta por ID\_NF e COD\_PROD;

**i.** Pesquise o valor total de NFs maiores de 500 e ordene o resultado do maior valor para o menor. As colunas do resultado são ID\_NF e VALOR\_TOT (VALOR\_TOTAL é obtido pela fórmula  $\Sigma \text{QUANTIDADE} * \text{VALOR\_UNIT}$ ). Agrupe o resultado da consulta por ID\_NF;

**j.** Qual o valor médio dos descontos por produto? As colunas do resultado são COD\_PROD e MEDIA. Agrupe o resultado por COD\_PROD;

**k.** Quais os valores médio, menor e maior menor por produto? As colunas do resultado são COD\_PROD, MENOR, MAIOR e MEDIA. Agrupe o resultado da consulta por COD\_PROD;

**l.** Quais são as NFs com mais de três itens vendidos? As colunas do resultado são ID\_NF e QTD\_ITENS (este último não relacionado à quantidade vendida do item, mas sim à quantidade de itens por NF). Agrupe o resultado da consulta por ID\_NF.

## <Referências>

[https://www.w3schools.com/sql/sql\\_syntax.asp](https://www.w3schools.com/sql/sql_syntax.asp)

<https://docs.microsoft.com/pt-br/sql/t-sql/language-elements/logical-operators-transact-sql?view=sql-server-ver16>

<https://blog.betrybe.com/sql/#1>

<https://kenzie.com.br/blog/sql/>

---

