

# Curso SQL do Básico ao Avançado

```
SELECT coluna1, coluna2, ...  
FROM nome_tabela  
WHERE condição  
GROUP BY colunas_agrupadas  
HAVING condição_agrupamento  
ORDER BY coluna_ordenacao ASC|DESC  
LIMIT quantidade_limite  
OFFSET quantidade_deslocamento;
```



## Módulo 02

# Conceitos Básicos e Consultas



# Relacionamentos: 1:N e N:1

---

**1:N (Um para Muitos):** Um elemento na tabela pai pode se relacionar com vários elementos na tabela filho.  
Exemplo: Uma tabela de Clientes com vários Pedidos.  
Cada pedido está vinculado a um único cliente.

---

**N:1 (Muitos para Um):** Muitos elementos na tabela filho estão relacionados a um único elemento na tabela pai.  
Exemplo invertido: Vários Pedidos associados a um Cliente específico.



# Relacionamentos: N:N

---

**N:N (Muitos para Muitos):** Relacionamento onde várias linhas em uma tabela se relacionam com várias linhas em outra tabela. Requer uma tabela de junção para gerenciar essas relações.

---

**Exemplo:** Alunos e Cursos. Uma tabela de junção mapeia quais alunos estão inscritos em quais cursos.



# Consultas Básicas



- A sintaxe do SELECT

A sintaxe básica da declaração SELECT no SQL é a seguinte:

```
SELECT coluna1, coluna2, ...  
FROM nome_tabela  
WHERE condição  
GROUP BY colunas_agrupadas  
HAVING condição_agrupamento  
ORDER BY coluna_ordenacao ASC|DESC  
LIMIT quantidade_limite  
OFFSET quantidade_deslocamento;
```



# Consultas Básicas

Cada parte da sintaxe serve a um propósito específico:

**SELECT:** Especifica as colunas a serem retornadas na consulta. O uso de asterisco (\*) em vez de nomes de colunas seleciona todas as colunas(deve ser utilizado com cautela).

**FROM:** Especifica a tabela de onde os dados devem ser selecionados ou recuperados.

**WHERE:** A cláusula WHERE em SQL é utilizada para filtrar registros e especificar quais linhas devem ser incluídas no resultado de comandos como SELECT, UPDATE, DELETE, entre outros. Ela permite a utilização de operadores lógicos como =, <> (ou !=), >, <, >=, <=, BETWEEN, LIKE, IN, além de conjunções como AND, OR e negação NOT.

**GROUP BY:** Agrupa as linhas que têm os mesmos valores em colunas especificadas para realizar operações de agregação, como contagens, médias, máximos e mínimos.

**HAVING:** Filtra grupos de acordo com uma condição de agregação. É frequentemente usado em conjunto com GROUP BY.

**ORDER BY:** Ordena o resultado com base nos valores de uma ou mais colunas. Usa-se ASC para ordem ascendente e DESC para ordem descendente. Se não especificado, ASC é o padrão.

**LIMIT:** Limita o número de linhas retornadas pela consulta. É útil quando você só precisa de uma subseção do conjunto de resultados.

**OFFSET:** Especifica o número de linhas a serem ignoradas antes de começar a contar as linhas para o

