

CONTEÚDO

- 01 Definição
- 02 História
- 03 SGBD's
- 04 Subconjuntos SQL
- 05 Exemplos
- 06 Cláusulas
- 07 Funções de Agregação
- 08 Relacionamento entre tabelas
- 09 Join's
- 10 Prática

DEFINIÇÃO

Structured Query Language | SQL

Linguagem de consulta estruturada

É uma linguagem padrão para operação em sistemas de gerenciamento de bancos de dados relacionais.

Com SQL é possível:

- Inserir, alterar, excluir, selecionar, ordenar, agrupar e mesclar dados.
- Criar, alterar, remover tabelas.
- Alterar estrutura de dados, estabelecer relações entre tabelas, gerenciar usuários.
- Importar, exportar dados.
- Criar chaves e restrições.



UM POUCO DE HISTÓRIA

- Surgiu a partir de um estudo realizado no laboratório da IBM na década de 70.
- Criado pelos cientistas Donald Chamberlim e Raymond Boyce.
- Criar um modelo capaz de gerenciar informações dentro do modelo relacional proposto por Edgar Codd.
- Padronização da linguagem em 1986 (Ansi) e 1987 (ISO).



SGBD'S

Sistema Gerenciador de Banco de Dados

Controle de acessos

Controle de concorrência

Backup e restore

Restrições de integridade



SUBCONJUNTOS SQL

DML

Data Manipulation Language

Manipulação de dados: inclusões, consultas, alterações e exclusões de dados

INSERT

UPDATE

DELETE

DDL

Data Definition Language

Criação e alteração da estrutura; tabelas, views, índices

CREATE

ALTER

DROP

DQL

Data Query Language

Consulta aos dados armazenados no banco de dados

SELECT

EXEMPLOS

DDL | SINTAXE

CREATE

CREATE DATABASE nome_database;

CREATE TABLE nome_tabela (
 coluna1 int NOT NULL,
 coluna2 varchar(30),
 coluna3 datetime,
 coluna4 decimal(3, 2));

ALTER

ALTER TABLE nome_tabela
ADD nome_coluna varchar(10)
AFTER nome_coluna;

ALTER TABLE nome_tabela
DROP COLUMN nome_coluna;

DROP

DROP DATABASE nome_database;
DROP TABLE nome_tabela

EXEMPLOS

DML | SINTAXE

INSERT

```
INSERT INTO nome_tabela  
(coluna1, coluna2)  
VALUES ('valor1', 'valor2');
```

UPDATE

```
UPDATE nome_tabela  
SET nome_coluna = 'valor'  
WHERE id = 1;
```

DELETE

```
DELETE FROM nome_tabela  
WHERE id = 1;
```


EXEMPLOS

DQL | SINTAXE

SELECT

```
SELECT t1.coluna1, t2.coluna2  
FROM nome_tabela1 T1  
INNER JOIN nome_tabela2 T2 ON t1.codigo = T2.codigo  
WHERE id = 1  
ORDER BY coluna1
```

```
SELECT coluna1, coluna2  
FROM nome_tabela  
WHERE id = 1  
ORDER BY coluna1
```

CLÁUSULAS

Condições utilizadas para definir os dados que devem ser selecionados ou modificados na consulta.

FROM: Especifica a(s) tabela(s) de onde serão selecionados os registros.

WHERE: Especifica a(s) condição(ões) que devem ser retornadas na consulta.

ORDER BY: Utilizado para retornar os registros em uma ordem especificada.

GROUP BY: Separa os registros em grupos específicos.

DISTINCT: Obtém os dados sem repetição.

FUNÇÕES DE AGREGAÇÃO

Funções que devolvem um único valor a um grupo de registro

COUNT: Obtém a quantidade de registros em um determinado campo.

SUM: Obtém a soma de todos os valores de um campo determinado.

AVG: Calcula a média dos valores de um campo especificado.

MAX: Obtém o maior valor de um campo especificado.

MIN: Obtém o menor valor de um campo especificado.

RELACIONAMENTO ENTRE TABELAS

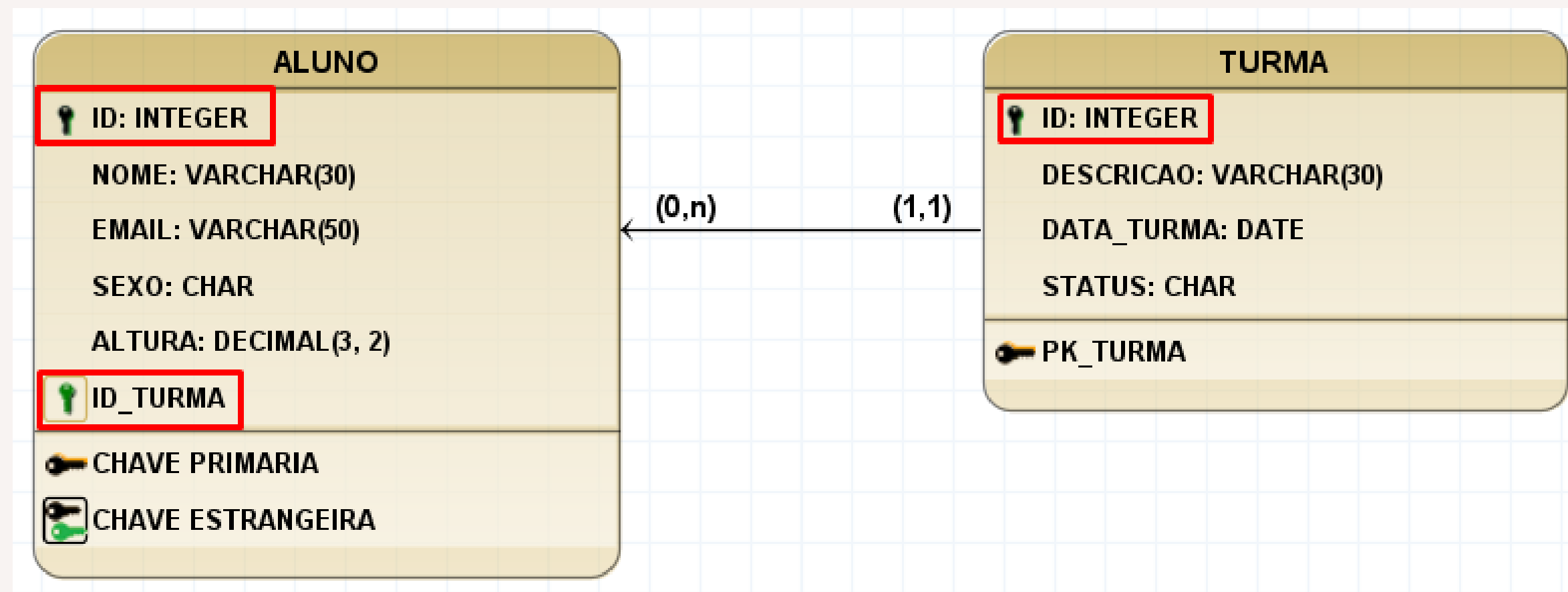
As tabelas se relacionam por meio das chaves que identificam os registros de forma única (chave primária e chave estrangeira)

CHAVE PRIMÁRIA (PK)

Campo que identifica os registro de forma única dentro da tabela

CHAVE ESTRANGEIRA (FK)

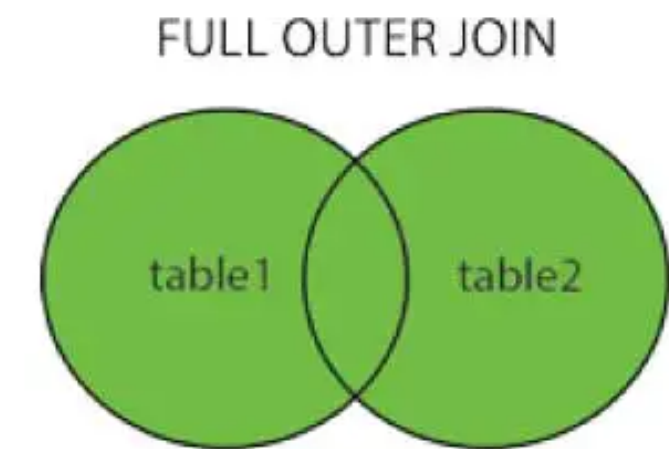
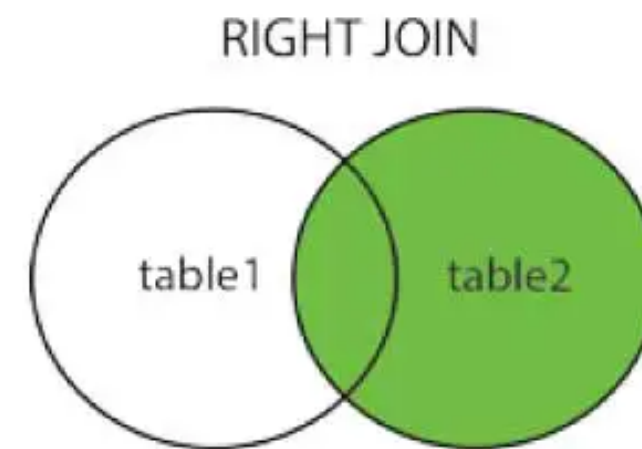
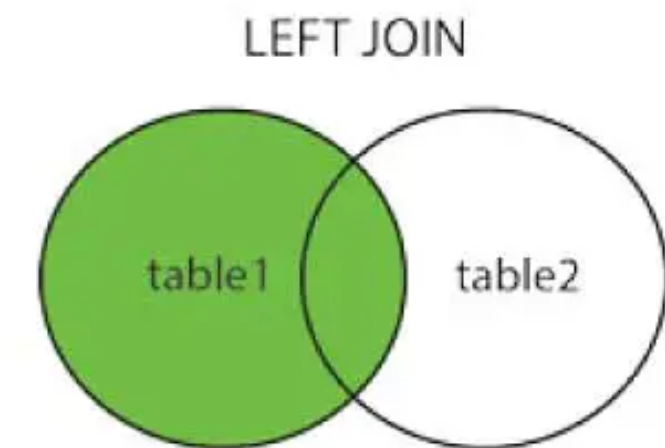
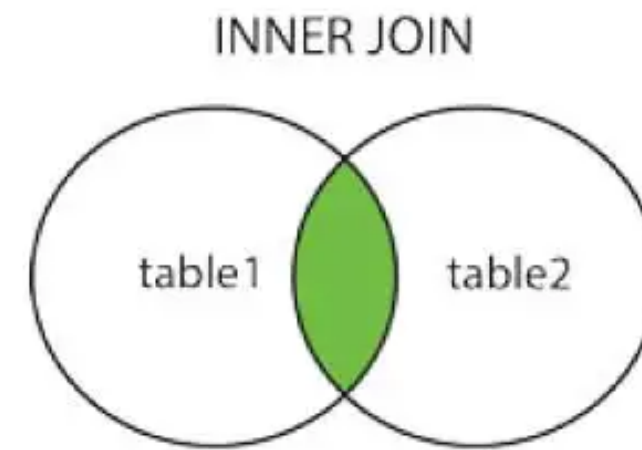
Chave primária da tabela que é acrescentada como um campo dentro da tabela relacionada.




RELACIONAMENTOS

| JOIN'S

Join's são usados para gerar consultas entre tabelas de acordo os relacionamentos em comum entre ambas.



LET'S PRACTICE!

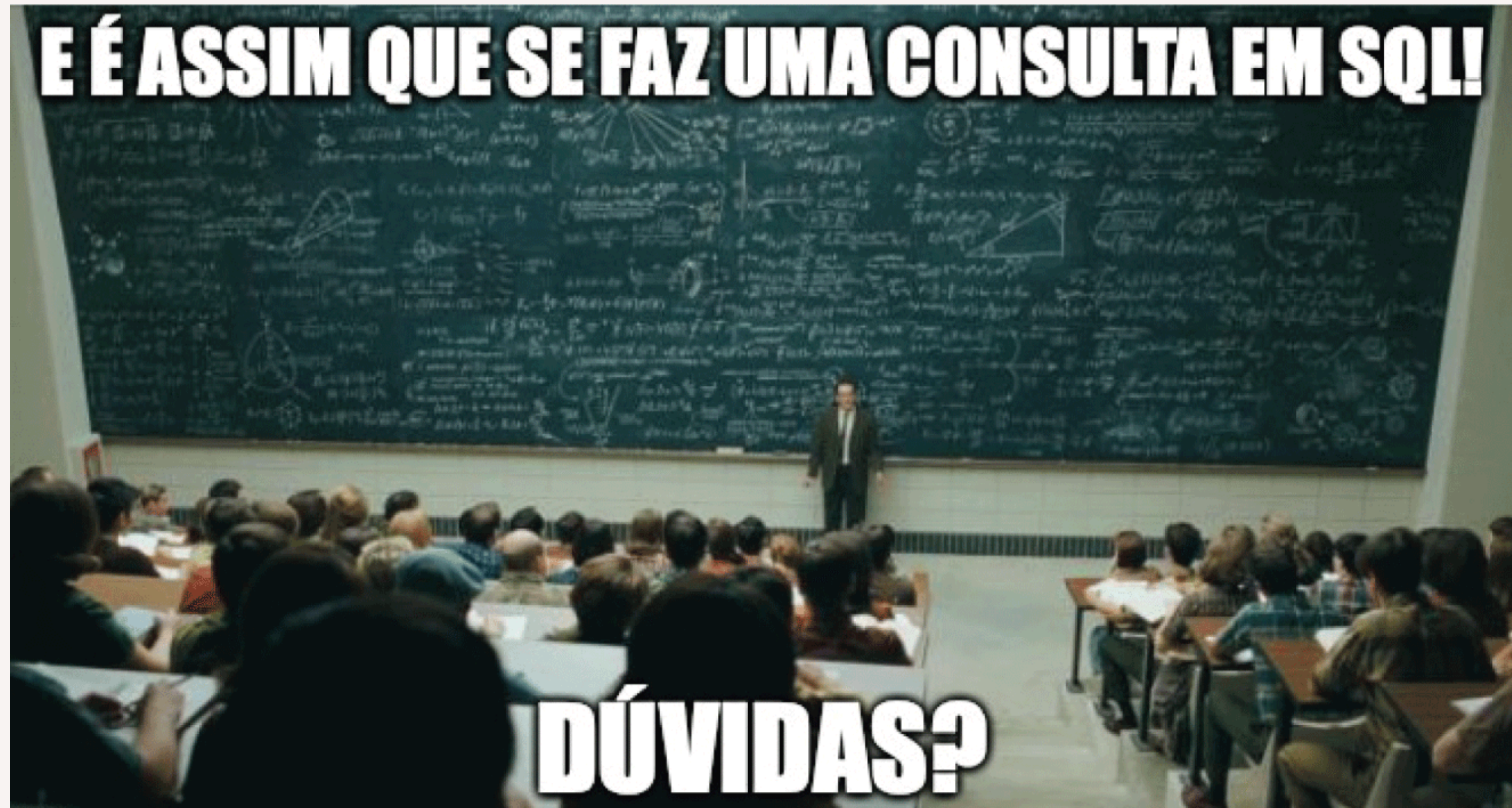


QUE OS JOGOS COMECEM...

VAMOS PRATICAR!

```
SELECT
  OrderH.invoiceNo, OrderH.invoiceDate, OrderH.customerCode, OrderH.netPrice,
  OrderD.itemCode, I.itemName, OrderD.qty, OrderD.netPrice, OrderD.grossPrice
FROM
  OrderHeader AS OrderH
  INNER JOIN Customer AS Cust ON OrderH.customerCode = Cust.customerCode
  INNER JOIN OrderDetail AS OrderD ON OrderH.orderNo = OrderD.orderNo
  INNER JOIN Item AS I ON OrderD.itemCode = I.itemCode
WHERE
  OrderD.netPrice > 1000
ORDER BY
  OrderH.customerCode, OrderD.netPrice
```


OBRIGADO!!!



E É ASSIM QUE SE FAZ UMA CONSULTA EM SQL!

DÚVIDAS?