Aula 22

```
'''
Criar um banco de dados chamado loja

1 - Acesse Cloud SQL
2 - Crie uma instância com MySQL
3 - Pelo card da instância criada acesse o Shell
4 - Pelo shell conecte-se ao Mysql
5 - create database loja;
'''

'''
Agora vamos criar 6 tabelas e suas relações referentes a uma loja de utensílios
domésticos
tabelas: Cliente, Pedido, Produto, ItemPedido, Categoria, ProdutoCategoria


create table Cliente (
  id_cliente INT PRIMARY KEY,
  nome VARCHAR (50),
  endereco VARCHAR (100)
);

CREATE TABLE Pedido (
    id_pedido INT PRIMARY KEY,
    id_cliente INT,
    descricao VARCHAR(100),
    FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente)
);

CREATE TABLE Produto (
    id_produto INT PRIMARY KEY,
    nome VARCHAR(50),
    preco DECIMAL(10, 2)
);

CREATE TABLE ItemPedido (
    id_item INT PRIMARY KEY,
    id_pedido INT,
    id_produto INT,
    quantidade INT,
    FOREIGN KEY (id_pedido) REFERENCES Pedido(id_pedido),
    FOREIGN KEY (id_produto) REFERENCES Produto(id_produto)
);

CREATE TABLE Categoria (
    id_categoria INT PRIMARY KEY,
    nome VARCHAR(50)
);

CREATE TABLE ProdutoCategoria (
    id_produto INT,
    id_categoria INT,
    PRIMARY KEY (id_produto, id_categoria),
    FOREIGN KEY (id_produto) REFERENCES Produto(id_produto),
    FOREIGN KEY (id_categoria) REFERENCES Categoria(id_categoria)
);

Agora vamos popular as 6 tabelas com dados referentes a 20 transações comerciais
dessa loja


INSERT INTO Cliente (id_cliente, nome, endereco)
VALUES
    (1, 'Douglas', 'SP'),
    (2, 'Daniela', 'SP'),
    (3, 'Ester', 'MG'),
    (4, 'Maria', 'BA'),
    (5, 'Douglas', 'PE'),
    (6, 'Raimundo', 'CE'),
    (7, 'Jucimara', 'GO'),
    (8, 'Marlene', 'AC'),
    (9, 'Douglas', 'MA'),
    (10, 'Daniela', 'PI'),
    (11, 'Pedro', 'PB'),
    (12, 'Joaquim', 'RS'),
    (13, 'Julio', 'TO'),
    (14, 'Carla', 'DF'),
```

```
    (15, 'Bianca', 'PA'),
    (16, 'Sandra', 'SP'),
    (17, 'Celso', 'SP'),
    (18, 'Silvana', 'BA'),
    (19, 'Prof Raquel', 'BA'),
    (20, 'Pedro', 'SP'),
    (21, 'Erick', 'BA'),
    (22, 'Denise', 'DF'),
    (23, 'Diana', 'SP'),
    (24, 'Dagmar', 'DF'),
    (25, 'Deiverson', 'SP');
'''
'''
INSERT INTO Pedido (id_pedido, id_cliente, descricao)
VALUES
    (1, '1', 'Caneca'),
    (2, '1', 'Garfo'),
    (3, '2', 'Faca'),
    (4, '5', 'Copo'),
    (5, '19', 'Copo'),
    (6, '19', 'Colher'),
    (7, '15', 'Colher'),
    (8, '8', 'Colher'),
    (9, '2', 'Copo'),
    (10, '19', 'Colher'),
    (11, '6', 'Copo'),
    (12, '11', 'Colher'),
    (13, '10', 'Garfo'),
    (14, '4', 'Garfo'),
    (15, '3', 'Colher'),
    (16, '3', 'Garfo'),
    (17, '2', 'Copo'),
    (18, '17', 'Garfo'),
    (19, '14', 'Caneca'),
    (20, '13', 'Caneca');
'''
'''
INSERT INTO Produto (id_produto, nome, preco)
VALUES
    (1, 'Caneca', 15.99),
    (2, 'Garfo', 5.99),
    (3, 'Faca', 7.99),
    (4, 'Colher', 3.99),
    (5, 'Copo', 10.99);
'''

'''
INSERT INTO ItemPedido (id_item, id_pedido, id_produto, quantidade)
VALUES
    (1, 1, 1, 2),
    (2, 2, 2, 5),
    (3, 3, 3, 7),
    (4, 4, 5, 10),
    (5, 5, 5, 11),
    (6, 6, 4, 30),
    (7, 7, 4, 32),
    (8, 8, 4, 44),
    (9, 9, 5, 13),
    (10, 10, 4, 33),
    (11, 11, 5, 34),
    (12, 12, 4, 32),
    (13, 13, 2, 33),
    (14, 14, 2, 34),
    (15, 15, 4, 27),
    (16, 16, 2, 18),
    (17, 17, 5, 17),
    (18, 18, 2, 33),
    (19, 19, 1, 50),
    (20, 20, 1, 47);
'''
'''
INSERT INTO Categoria (id_categoria, nome)
VALUES
    (1, 'Comer'),
    (2, 'Beber');

INSERT INTO ProdutoCategoria (id_produto, id_categoria)
VALUES
    (1, 2),
    (2, 1),
    (3, 1),
    (4, 1),
    (5, 2);
```

```
    (3, 2);
'''
```

Agora vamos aprender os principais comandos de consulta para o Analista de Dados

```
'''
colocar os selects * from para consulta de cada banco populado
'''
'''
-- seta o banco de dados, indica que usaremos o banco de dados loja
USE loja;

SELECT * FROM Cliente;
-- Lendo o comando: Seleciona todas as colunas da tabela: Cliente

SELECT * FROM Pedido;
-- Lendo o comando: Seleciona todas as colunas da tabela: Pedido

SELECT * FROM Produto;
-- Lendo o comando: Seleciona todas as colunas da tabela: Produto

SELECT * FROM ItemPedido;
-- Lendo o comando: Seleciona todas as colunas da tabela: ItemPedido

SELECT * FROM Categoria;
-- Lendo o comando: Seleciona todas as colunas da tabela: Categoria

SELECT * FROM ProdutoCategoria;
-- Lendo o comando: Seleciona todas as colunas da tabela: ProdutoCategoria

-- Warning: Evite o uso do *, seja específico, adicione apenas as colunas que irá precisar na sua análise.
-- Exemplo:
SELECT id_produto, quantidade FROM ItemPedido;

  Lendo o comando: Seleciona as colunas: id_produto e quantidade da tabela ItemPedido
'''
```

WHERE — São as condições

```
'''
SELECT id_produto, quantidade FROM ItemPedido
WHERE quantidade >= 40;
Lendo o comando: Seleciona as colunas: id_produto e quantidade da tabela: ItemPedido, quando a quantidade for maior ou igual a 10
'''

'''
SELECT * FROM Cliente
WHERE endereco = 'SP';

  Lendo o comando: Seleciona todas as colunas, quando o endereço for igual a SP
'''
```

DISTINCT

```
'''
SELECT distinct id_produto, quantidade FROM ItemPedido
WHERE quantidade >= 10;

  Lendo o comando: Seleciona distintamente as colunas: id_produto e quantidade da tabela: ItemPedido, quando o quando a quantidade for
'''

'''
SELECT distinct descricao FROM Pedido;

  Lendo o comando: Seleciona distintamente as colunas: id_produto e quantidade da tabela: ItemPedido, quando o quando a quantidade for
'''
```

LIMIT — Retornar os n primeiros registros

```
'''
SELECT * FROM Cliente
LIMIT 5;
  Lendo o comando: Seleciona todos os atributos da tabela Cliente e mostra
  apenas os cinco primeiros registros
'''
```

BOOLEANOS (E | OU | NÃO) Bom para combinar com WHERE

```
'''

-- Exemplo de AND
SELECT nome, endereco
FROM Cliente
WHERE endereco = 'BA'
AND id_cliente = 19;
Lendo o comando:
   Seleciona as colunas: nome e endereco da tabela: Clientes,
   apenas os registros com endereco igual a 'BA' e id_cliente igual a 19


-- Exemplo de OR
SELECT nome, endereco
FROM Cliente
WHERE endereco = 'SP'
OR endereco = 'MG';
Lendo o comando:
   Seleciona as colunas: nome e endereco da tabela: Clientes,
   apenas os registros com endereco igual a 'SP' ou endereco a 'MG'

-- Exemplo de NOT
SELECT nome, endereco
FROM Cliente
WHERE NOT endereco = 'SP';
Lendo o comando:
   Seleciona as colunas: nome e endereco da tabela: Clientes,
   apenas os registros com endereco não seja igual a 'SP'
```

Funções de agregação (SUM , AVG , MAX , MIN , COUNT e ROUND)

```
'''
Somatório da quantidade de produtos vendidos

SELECT SUM(quantidade) AS soma_qtde
FROM ItemPedido;
'''

'''
Média da quantidade de produtos vendidos

SELECT AVG(quantidade) AS media_qtde,
round(AVG(quantidade),2) AS media_qtde_arredondado
FROM ItemPedido;
'''


'''
Maior quantidade de produtos vendidos

SELECT MAX(quantidade) AS maior_quantidade_venda
  FROM ItemPedido;
'''
'''
SELECT MIN(quantidade)  menor_quantidade_venda
  FROM ItemPedido;
'''

'''
Contagem de produtos (quantos produtos temos na nossa base?)

SELECT count(id_produto) as qtd_produtos
  FROM Produto;
'''
```

Aula 23

GROUP BY

```
'''
SELECT endereco,
     count(*) AS 'freq'
FROM Cliente
GROUP BY endereco;
'''

'''
Lendo o código: Comando que seleciona o atributo endereço, cria e seleciona a
coluna count renomeada com freq da tabela Cliente e agrupa por endereco
'''
```

HAVING

```
'''
SELECT endereco,
     count(*)
FROM Cliente
GROUP BY endereco
HAVING count(*) > 1;
'''

'''
Lendo o código: Consulta que seleciona endereco e contangem de endereco da
tabela Cliente, Agrupado por endereço tendo apenas (HAVING) os edereços que se
repetem pelo menos duas vezes.
'''
```

UNION

```
'''
Serve para unir duas consultas diferentes e mostá-la na mesma tabela

SELECT id_pedido, id_produto
  FROM ItemPedido
WHERE id_produto = 1
UNION
SELECT id_pedido, id_produto
  FROM ItemPedido
WHERE id_produto = 4;

Lendo o código: Unindo duas consultas, sendo a primeira como a seleção de
id_pedido e id_produto da tabela ItemPedido onde o id_produto é 1, e a segunda
como a seleção de id_pedido e id_produto da tabela ItemPedido onde o
id_produto é 4
'''
```

IN BETWEEN

```
'''
SELECT
  id_pedido,
  id_produto
FROM ItemPedido
WHERE id_produto = 1 OR id_produto = 2;

Alternativa seria usar IN

SELECT
  id_pedido,
  id_produto
FROM ItemPedido
WHERE id_produto IN (1,2);
'''

'''
Essas duas consultas anteriores trazem exatamente a mesma resposta, porém de
formas diferentes
'''
'''
SELECT
  id_pedido,
  id_produto
FROM ItemPedido
WHERE id_produto = 1 OR id_produto = 2;
'''
'''
Lendo a consulta: Selecionar id_pedido e id_produto da tabela ItemPedido
onde id_produto seja igual a 1 ou igual a dois usando where combinado com or
'''
'''
SELECT
  id_pedido,
  id_produto
FROM ItemPedido
WHERE id_produto IN (1,2);

'''

'''
Agora vamos imaginar que em relação ao que tivemos de retorno na consulta
anterior nosso objetivo seja ainda mais restritivo: Agora além de
Selecionar id_pedido e id_produto da tabela ItemPedidoonde id_produto seja
igual a 1 ou igual a dois queremos também que o númerodo pedido esteja
entre (BETWEEN) 1 e 9, pois o nosso objetivos é selcionar

SELECT
  id_pedido,
  id_produto
FROM ItemPedido
WHERE id_produto IN (1,2) AND
  id_pedido BETWEEN 1 AND 9;

Lendo a consulta: Selecionar id_pedido e id_produto da tabela ItemPedido
onde id_produto seja igual a 1 ou igual a dois usando IN E além disso, o número
do pedido esteja entre (BETWEEN) 1 e 9, pois o nosso objetivos é selcionar
somente
os id_pedido de 1 dígito.

'''


```

LIKE

```
'''
Agora vamos para a tabela Produto e no atributo nome vamos realizar uma consulta
que nos retorne todos os produtos que terminem com a letra o e os seus
respectivos ids

SELECT
  id_produto,
  nome
FROM Produto
WHERE nome LIKE '%o';

Agora vamos para a tabela Produto e no atributo nome vamos realizar uma consulta
que nos retorne todos os produtos que comecem com a letra C e os seus
respectivos ids

SELECT
  id_produto,
  nome
FROM Produto
WHERE nome LIKE 'C%';
'''

'''
Agora vamos para a tabela Produto e no atributo nome vamos realizar uma consulta
que nos retorne todos os produtos que contenha a sequencia lh e os seus
respectivos ids


SELECT
  id_produto,
  nome
FROM Produto
WHERE nome LIKE '%lh%';

'''
```

NULL e NOT NULL

```
'''
Verficar a existência de valores nulos para o atributo nome na tabela Produto

SELECT nome
FROM Produto
WHERE nome IS NULL;


Verficar a existência de valores não nulos para o atributo
nome na tabela Produto

SELECT nome
FROM Produto
WHERE nome IS NOT NULL;
'''

```

ORDER BY

```
'''
Vamos selecionar todos os atributos da tabela ItemPedido, ordenando de forma
Crescente por quantidade

SELECT * FROM ItemPedido
ORDER BY quantidade ASC;

Vamos selecionar todos os atributos da tabela ItemPedido, ordenando de forma
Decrescente por quantidade

SELECT id_pedido FROM ItemPedido
ORDER BY quantidade DESC;
'''
```

TRIM - Remove os espaços

```
'''
SELECT nome,
  TRIM(nome) AS nome_no_space
FROM Produto;
'''
```

REPLACE Substitui um caracter ou sequencia de caracteres

```
'''
SELECT nome,
  REPLACE(nome, 'o', 'oo') AS nome_zuado
FROM Produto;
'''
```

LPAD

```
'''
Vamos deixar todos os ids de pedidos na tabela ItemPedido com cinco dígitos,
Assim ele preencherá todos os valores com zeros à esquerda que forem necessários
para o números der dígitos solicitado

SELECT id_pedido,
  LPAD(id_pedido, 5, "0")
FROM ItemPedido;
'''
```

SUBSTRING

```
'''
SELECT nome,
  SUBSTRING(nome, 1, 2) AS subs
FROM Produto;
'''
```

UPPER

```

'''
Deixando todos os valores de um atributo com todas as letras maiúsculas
SELECT nome,
  UPPER(nome)
FROM Produto;
'''
'''
Deixando todos os valores de um atributo com todas as letras minúsculas
SELECT nome,
  LOWER(nome)
FROM Produto;
'''
'''
SELECT nome,
  LENGTH(nome)
FROM Produto;
'''
```

CAST

```
'''
SELECT id_cliente,
       concat('Cliente - ', CAST(id_cliente AS CHAR)) AS id_cliente_texto
  FROM Cliente;
'''
```

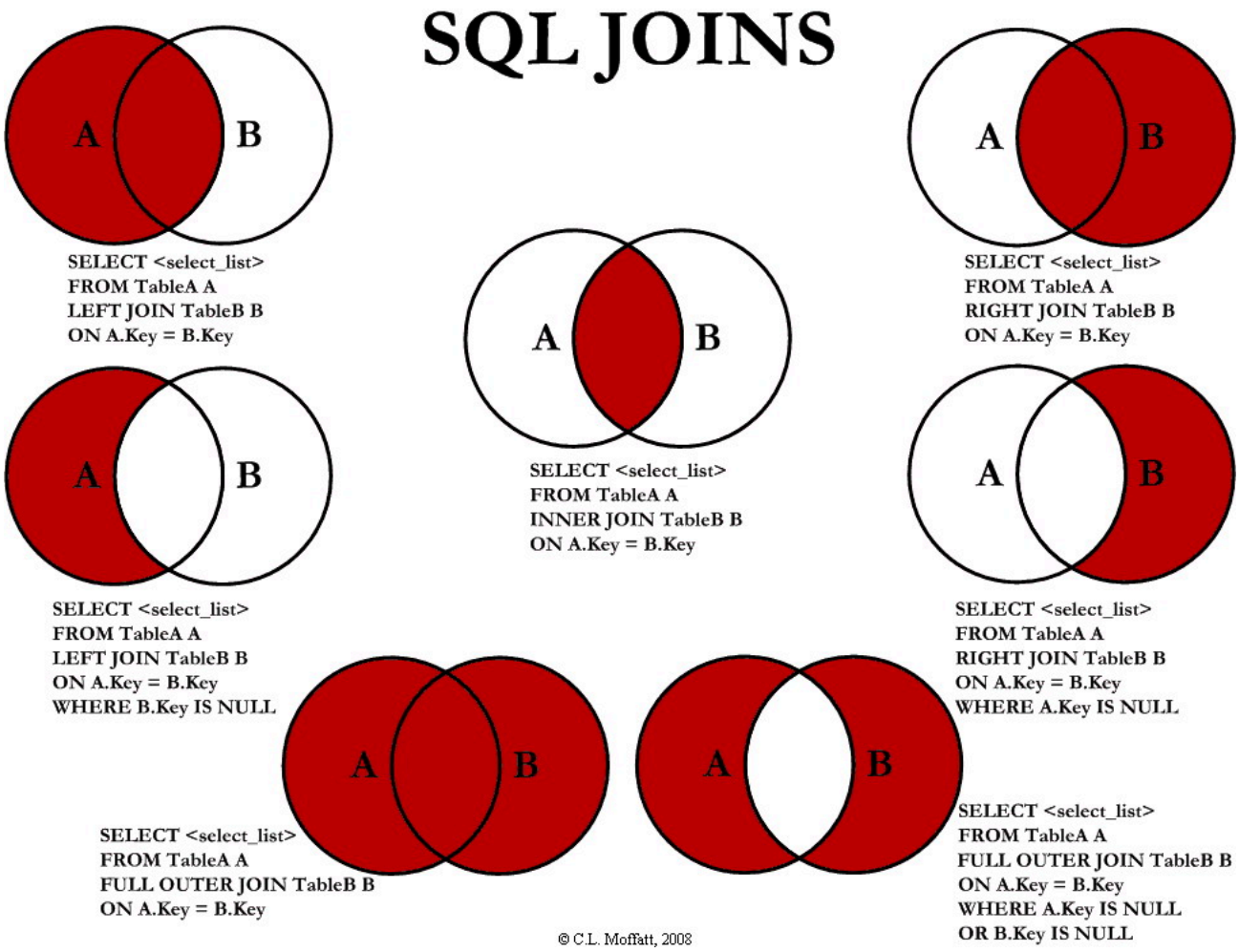
CASE — Semelhante ao if/else

```
'''
SELECT id_pedido, quantidade,
    CASE WHEN quantidade > 30 THEN 'Atacado'
    ELSE 'Varejo' END
FROM ItemPedido;
'''
'''
Lendo o código: Selecione os atributos id_pedido, quantidade
(1ª linha do bloco de código) da tabela ItemPedido
(4ª linha1ª linha do bloco de código) e quando a quantidade for maior do que 30
devolva em uma coluna separada a classificação daquele pedido como atacado
caso seja menor que 30 classifique como varejo e depois encerre (END)
'''
```

VIEW

```
'''
CREATE OR REPLACE VIEW query AS
SELECT id_pedido, quantidade,
    CASE WHEN quantidade > 30 THEN 'Atacado'
    ELSE 'Varejo' END
FROM ItemPedido;

'''
```




```
'''
Inserindo novos clientes na tabela Cliente para melhorar a visualição dos joins
INSERT INTO Cliente (id_cliente, nome, endereco)
VALUES
    (21, 'Erick', 'BA'),
    (22, 'Denise', 'DF'),
    (23, 'Diana', 'SP'),
    (24, 'Dagmar', 'DF'),
    (25, 'Deiverson', 'SP');
'''
```

```
'''
SELECT *
  FROM  ItemPedido v
  INNER JOIN ProdutoCategoria c
  ON  v.id_produto = c.id_produto;

'''


'''
SELECT *
  FROM Cliente c
  LEFT JOIN Pedido p
  ON p.id_cliente = c.id_cliente;
'''
'''
SELECT *
  FROM Cliente c
```