

## SEGUNDA ENTREGA DO PROJETO FINAL

MARÍLIA TRIVILIN MENDES

### CONTEÚDO DA PRIMEIRA ENTREGA ATUALIZADO

#### TEMÁTICA DO BANCO DE DADOS

O tema do meu banco de dados é **Restaurante**, que consiste num banco de dados que armazena as informações de várias franquias/lojas de restaurantes que atuam em diferentes estados do Brasil.

#### TABELAS QUE COMPREENDEM O BANCO DE DADOS

**lojas\_franquia** – listagem de todas as lojas que compõem a franquias

Column	Type
id_loja	int
municipio	text
nome_unidade	varchar(20)
UF	text

**Primary Key (PK):** id\_loja

**Foreign Key (FK):** nenhuma.

**pedidos** – listagem de todos os pedidos feitos desde o início das lojas.

Column	Type
◇ cpf_cliente	varchar(20)
◇ data_pedido	datetime
◇ forma_pagamento	varchar(30)
◇ id_colaborador	int
◇ id_loja	int
◇ id_pedido	int
◇ nome_cliente	varchar(30)
◇ valor_pedido	decimal(5,2)

Primary Key (PK): id\_pedido

Foreign Key (FK): id\_loja.

**funcionarios** - listagem dos funcionários ativos e/ou inativos que trabalharam nas lojas

Column	Type
◇ cargo	varchar(30)
◇ cpf_funcionario	varchar(20)
◇ dt_admissao	datetime
◇ dt_desligamento	datetime
◇ id_colaborador	int
◇ id_loja	int
◇ salario	decimal(7,2)

Primary Key (PK):

Foreign Key (FK): id\_loja.

**fornecedores** - listagem dos fornecedores que abastecem as lojas

Column	Type
◇ cnpj_f	varchar(20)
◇ id_item	int
◇ id_loja	int
◇ nome_item	varchar(50)

Primary Key (PK): cnpj\_f

Foreign Key (FK): id\_loja.

**cardapio** - listagem com a composição do menu

Column	Type
◇ cnpj_f	varchar(20)
◇ id_cardapio	int
◇ id_item	int
◇ nome_item	varchar(50)
◇ valor_item	decimal(7,2)

**Primary Key (PK):** id\_cardapio

**Foreign Key (FK):** cnpj\_f

## CONTEÚDO DA SEGUNDA ENTREGA

### VIEWS

1. **Faturamento:** view que retorna a soma dos valores dos pedidos se colocar o id\_loja como input (minha base de dados)

```
CREATE DEFINER='root'@'localhost' FUNCTION `faturamento`(IDLOJA INT) RETURNS  
decimal(5,2) CHARSET utf8mb4  
  
    READS SQL DATA  
  
    BEGIN  
  
        DECLARE faturamento1 DECIMAL(5,2);  
  
        SET faturamento1 = (select sum(valor_pedido) from pedidos where id_loja = IDLOJA) ;  
  
        RETURN faturamento1;  
  
    END
```

### LISTAGEM DE FUNÇÕES

1. **Maior Faturamento:** Retorna a loja de maior faturamento no 1º Semestre/ 2022.

```
select p.id_loja, l.nome_unidade, sum(valor_pedido) as faturamento
FROM pedidos as p
INNER JOIN lojas_franquia as l ON p.id_loja = l.id_loja
WHERE data_pedido > '2022-01-01 00:00:00' AND data_pedido < '2022-06-30 00:00:00'
GROUP BY l.nome_unidade
ORDER BY faturamento DESC;
```

**2. Maior Faturamento:** Retorna o Meio de pagamento mais utilizado pelos clientes da loja de Moema.

```
select forma_pagamento, count(forma_pagamento) as contagem from pedidos
WHERE id_loja = 1
GROUP BY forma_pagamento;
```

**3. Cliente com maior gasto:** Retorna o cliente de maior gasto.

```
select id_loja, nome_cliente, cpf_cliente, valor_pedido
from pedidos
GROUP BY nome_cliente, cpf_cliente
ORDER BY valor_pedido DESC
LIMIT 5;
```

**4. Quantidade Lojas:** Retorna quantidade de lojas por estado

```
select count(id_loja) as qtd_lojas, uf
FROM lojas_franquia
GROUP BY UF;
```

**5. Quantidade de fornecedores por loja:** Retorna quantidade de fornecedores por loja.

```
select id_loja, COUNT(cnpj_f) as contagem_fornecedores from fornecedores
GROUP BY id_loja;
```

## LISTAGEM DE STORED PROCEDURES

### 1. Ordena campos da tabela pedido;

```
CREATE PROCEDURE `ordenacao` (IN field CHAR(20) )
BEGIN
    IF field <> '' THEN
        SET @ordem = concat( 'ORDER BY', field );
    ELSE
        SET @ordem = '' ;
    END IF;
    SET @clausula = concat('SELECT * FROM rede_restaurantes.pedidos', @ordem );
    prepare runSql from @clausula;
    execute runSql;
END
```

### 2. Traz os pedidos específicos de uma loja;

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `pedidos_loja`(IN field INT )
BEGIN
    IF field BETWEEN 1 AND 8 THEN
        SET @pedidos = concat( 'WHERE id_loja =', field );
    ELSE
        SET @pedidos = '' ;
    END IF;
    SET @pedidos = concat('SELECT * FROM rede_restaurantes.pedidos', @pedidos );
    prepare runSql from @clausula;
    execute runSql;
END
```