

Seguindo ali no exercicio da cafeteria bom gosto, refaça todos os passos, agora incluindo passos para criação no ubuntu usando o dbeaver



Cafeteria BomGosto - Guia Completo Ubuntu + DBeaver



Pré-requisitos

1. Instalar MySQL no Ubuntu

```
bash
```

```
# Atualizar sistema
```

```
sudo apt update && sudo apt upgrade -y
```

```
# Instalar MySQL Server
```

```
sudo apt install mysql-server -y
```

```
# Configurar segurança do MySQL
```

```
sudo mysql_secure_installation
```

```
# Acessar MySQL
```

```
sudo mysql -u root -p
```

2. Instalar DBeaver

```
bash
```

```
# Adicionar repositório do DBeaver
```

```
wget -O - https://dbeaver.io/debs/dbeaver.gpg.key | sudo apt-key add -
```

```
echo "deb https://dbeaver.io/debs/dbeaver-ce /" | sudo tee /etc/apt/sources.list.d/dbeaver.list
```

```
# Instalar DBeaver
```

```
sudo apt update
```

```
sudo apt install dbeaver-ce
```



Configuração no DBeaver

1. Criar Conexão com MySQL

1. Abra o DBeaver
2. Clique em "Nova Conexão de Banco de Dados"
3. Selecione MySQL
4. Preencha os dados:
 - Host: localhost
 - Port: 3306
 - Database: bomgosto (vamos criar)
 - Username: root
 - Password: sua_senha

2. Criar Banco de Dados

```
sql
```

```
-- Executar no DBeaver  
CREATE DATABASE bomgosto;  
USE bomgosto;
```

Scripts SQL Completos

1. Criação das Tabelas

```
sql
```

```
-- Tabela Cardapio  
CREATE TABLE Cardapio (  
    codigo INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(100) NOT NULL UNIQUE,  
    descricao TEXT,  
    preco_unitario DECIMAL(10,2) NOT NULL  
);  
  
-- Tabela Comanda  
CREATE TABLE Comanda (  
    codigo INT PRIMARY KEY AUTO_INCREMENT,  
    data DATE NOT NULL,  
    mesa INT NOT NULL,
```

```
    nome_cliente VARCHAR(100) NOT NULL
);

-- Tabela ItemComanda
CREATE TABLE ItemComanda (
    codigo_comanda INT,
    codigo_cardapio INT,
    quantidade INT NOT NULL CHECK (quantidade > 0),
    PRIMARY KEY (codigo_comanda, codigo_cardapio),
    FOREIGN KEY (codigo_comanda) REFERENCES Comanda(codigo) ON DELETE CASCADE,
    FOREIGN KEY (codigo_cardapio) REFERENCES Cardapio(codigo) ON DELETE CASCADE
);
```

2. Inserir Dados de Exemplo

sql

```
-- Inserindo cafés no cardápio
INSERT INTO Cardapio (nome, descricao, preco_unitario) VALUES
('Expresso', 'Café puro e forte, preparado sob pressão', 5.00),
('Cappuccino', 'Café expresso com leite vaporizado e espuma', 8.50),
('Latte', 'Café expresso com leite vaporizado', 7.50),
('Mocha', 'Café com chocolate e leite vaporizado', 9.00),
('Americano', 'Café expresso diluído em água quente', 6.00),
('Macchiato', 'Café expresso com uma pequena quantidade de leite', 7.00),
('Affogato', 'Sorvete com café expresso', 12.00);

-- Inserindo comandas
INSERT INTO Comanda (data, mesa, nome_cliente) VALUES
('2024-01-15', 1, 'João Silva'),
('2024-01-15', 2, 'Maria Santos'),
('2024-01-15', 3, 'Carlos Oliveira'),
('2024-01-16', 1, 'Ana Costa'),
('2024-01-16', 2, 'Pedro Martins'),
('2024-01-17', 4, 'Fernanda Lima');

-- Inserindo itens das comandas
INSERT INTO ItemComanda (codigo_comanda, codigo_cardapio, quantidade) VALUES
(1, 1, 2), -- João: 2 Expressos
(1, 2, 1), -- João: 1 Cappuccino
(2, 3, 1), -- Maria: 1 Latte
(2, 4, 1), -- Maria: 1 Mocha
(3, 1, 3), -- Carlos: 3 Expressos
(4, 2, 2), -- Ana: 2 Cappuccinos
(4, 5, 1), -- Ana: 1 Americano
(5, 6, 1), -- Pedro: 1 Macchiato
```

```
(5, 7, 1), -- Pedro: 1 Affogato  
(6, 3, 2), -- Fernanda: 2 Lattes  
(6, 4, 1); -- Fernanda: 1 Mocha
```



Consultas Solicitadas

1

Listagem do cardápio ordenada por nome

sql

```
SELECT  
    codigo,  
    nome,  
    descricao,  
    CONCAT('R$ ', FORMAT(preco_unitario, 2)) AS preco  
FROM Cardapio  
ORDER BY nome;
```

2

Comandas com itens detalhados

sql

```
SELECT  
    c.codigo AS codigo_comanda,  
    DATE_FORMAT(c.data, '%d/%m/%Y') AS data,  
    c.mesa,  
    c.nome_cliente,  
    card.nome AS nome_cafe,  
    card.descricao,  
    ic.quantidade,  
    CONCAT('R$ ', FORMAT(card.preco_unitario, 2)) AS preco_unitario,  
    CONCAT('R$ ', FORMAT(ic.quantidade * card.preco_unitario, 2)) AS preco_total  
FROM Comanda c  
INNER JOIN ItemComanda ic ON c.codigo = ic.codigo_comanda  
INNER JOIN Cardapio card ON ic.codigo_cardapio = card.codigo  
ORDER BY c.data, c.codigo, card.nome;
```

3

Comandas com valor total

sql

```
SELECT
    c.codigo,
    DATE_FORMAT(c.data, '%d/%m/%Y') AS data,
    c.mesa,
    c.nome_cliente,
    CONCAT('R$ ', FORMAT(SUM(ic.quantidade * card.preco_unitario), 2)) AS valor_total
FROM Comanda c
INNER JOIN ItemComanda ic ON c.codigo = ic.codigo_comanda
INNER JOIN Cardapio card ON ic.codigo_cardapio = card.codigo
GROUP BY c.codigo, c.data, c.mesa, c.nome_cliente
ORDER BY c.data;
```

4 Comandas com mais de um tipo de café

sql

```
SELECT
    c.codigo,
    DATE_FORMAT(c.data, '%d/%m/%Y') AS data,
    c.mesa,
    c.nome_cliente,
    COUNT(ic.codigo_cardapio) AS tipos_cafe,
    CONCAT('R$ ', FORMAT(SUM(ic.quantidade * card.preco_unitario), 2)) AS valor_total
FROM Comanda c
INNER JOIN ItemComanda ic ON c.codigo = ic.codigo_comanda
INNER JOIN Cardapio card ON ic.codigo_cardapio = card.codigo
GROUP BY c.codigo, c.data, c.mesa, c.nome_cliente
HAVING COUNT(ic.codigo_cardapio) > 1
ORDER BY c.data;
```

5 Faturamento total por data

sql

```
SELECT
    DATE_FORMAT(c.data, '%d/%m/%Y') AS data,
    CONCAT('R$ ', FORMAT(SUM(ic.quantidade * card.preco_unitario), 2)) AS faturamento_d
ia
FROM Comanda c
INNER JOIN ItemComanda ic ON c.codigo = ic.codigo_comanda
INNER JOIN Cardapio card ON ic.codigo_cardapio = card.codigo
GROUP BY c.data
ORDER BY c.data;
```

Configurações Avançadas no DBeaver

1. Criar Índices para Performance

sql

```
-- Executar no DBeaver para melhorar performance
CREATE INDEX idx_comanda_data ON Comanda(data);
CREATE INDEX idx_item_comanda_cardapio ON ItemComanda(codigo_cardapio);
CREATE INDEX idx_cardapio_nome ON Cardapio(nome);
```

2. Views para Consultas Frequentes

sql

```
-- View para relatório diário
CREATE VIEW VW_Relatorio_Diario AS
SELECT
    c.data,
    COUNT(DISTINCT c.codigo) AS total_comandas,
    SUM(ic.quantidade) AS total_cafes_vendidos,
    SUM(ic.quantidade * card.preco_unitario) AS faturamento_total
FROM Comanda c
INNER JOIN ItemComanda ic ON c.codigo = ic.codigo_comanda
INNER JOIN Cardapio card ON ic.codigo_cardapio = card.codigo
GROUP BY c.data;

-- View para top produtos
CREATE VIEW VW_Top_Produtos AS
SELECT
    card.nome,
    SUM(ic.quantidade) AS total_vendido,
    SUM(ic.quantidade * card.preco_unitario) AS faturamento
FROM ItemComanda ic
INNER JOIN Cardapio card ON ic.codigo_cardapio = card.codigo
GROUP BY card.codigo, card.nome
ORDER BY total_vendido DESC;
```

3. Stored Procedure para Nova Comanda

sql

```
DELIMITER //
```

```
CREATE PROCEDURE sp_nova_comanda(  
    IN p_data DATE,  
    IN p_mesa INT,  
    IN p_cliente VARCHAR(100)  
)  
BEGIN  
    INSERT INTO Comanda (data, mesa, nome_cliente)  
    VALUES (p_data, p_mesa, p_cliente);  
    SELECT LAST_INSERT_ID() AS nova_comanda_id;  
END //
```

```
DELIMITER ;
```



Como Executar no DBeaver

Passo a Passo:

1. Abra o DBeaver
2. Conecte ao MySQL (bomgosto database)
3. Abra um novo script SQL (Ctrl+N)
4. Copie e execute cada seção:
 - Primeiro: Criação das tabelas
 - Segundo: Inserção de dados
 - Terceiro: Consultas individuais

Executar Procedure:

```
sql
```

```
-- Testar a stored procedure  
CALL sp_nova_comanda('2024-01-18', 5, 'Teste Cliente');
```



Consultas de Análise (Extras)

Média de Valor por Comanda

```
sql
```

```
SELECT
    DATE_FORMAT(data, '%Y-%m') AS mes,
    COUNT(*) AS total_comandas,
    CONCAT('R$ ', FORMAT(AVG(valor_total), 2)) AS media_comanda
FROM (
    SELECT
        c.data,
        c.codigo,
        SUM(ic.quantidade * card.preco_unitario) AS valor_total
    FROM Comanda c
    INNER JOIN ItemComanda ic ON c.codigo = ic.codigo_comanda
    INNER JOIN Cardapio card ON ic.codigo_cardapio = card.codigo
    GROUP BY c.data, c.codigo
) AS comandas
GROUP BY DATE_FORMAT(data, '%Y-%m')
ORDER BY mes;
```

Clientes Frequentes

sql

```
SELECT
    nome_cliente,
    COUNT(*) AS total_comandas,
    CONCAT('R$ ', FORMAT(SUM(valor_total), 2)) AS total_gasto
FROM (
    SELECT
        c.nome_cliente,
        c.codigo,
        SUM(ic.quantidade * card.preco_unitario) AS valor_total
    FROM Comanda c
    INNER JOIN ItemComanda ic ON c.codigo = ic.codigo_comanda
    INNER JOIN Cardapio card ON ic.codigo_cardapio = card.codigo
    GROUP BY c.nome_cliente, c.codigo
) AS cliente_comandas
GROUP BY nome_cliente
HAVING total_comandas > 1
ORDER BY total_gasto DESC;
```



Verificação Final

Testar se tudo está funcionando:

sql

```
-- Verificar contagem de registros
```

```
SELECT
```

```
    (SELECT COUNT(*) FROM Cardapio) AS total_cafes,  
    (SELECT COUNT(*) FROM Comanda) AS total_comandas,  
    (SELECT COUNT(*) FROM ItemComanda) AS total_itens;
```

Resultado esperado:

text

total_cafes: 7

total_comandas: 6