

Projeto Lógico de Banco de Dados Oficina Mecânica

Neste desafio, vamos criar o esquema lógico de um banco de dados para o contexto de uma oficina mecânica, com base no esquema conceitual desenvolvido anteriormente. O banco de dados deve permitir o gerenciamento de clientes, veículos, ordens de serviço, mecânicos, serviços e peças utilizadas.

1. Esquema Lógico do Banco de Dados

1.1 Tabelas e Relacionamentos

1. Cliente: Armazena os dados dos clientes que levam seus veículos para a oficina.
2. Veículo: Armazena informações dos veículos pertencentes aos clientes.
3. Ordem de Serviço (OS): Registra as ordens de serviço emitidas para os veículos.
4. Serviço: Lista de serviços oferecidos pela oficina.
5. Peça: Registra as peças usadas nos serviços.
6. Mecânico: Informações dos mecânicos que realizam os serviços.
7. Equipe: Cada ordem de serviço é realizada por uma equipe de mecânicos.
8. Item_OS: Associação entre as ordens de serviço e os serviços realizados, com valores e quantidades.
9. Peça_OS: Associação entre as ordens de serviço e as peças usadas.

2. Script SQL de Criação do Banco de Dados

```
```sql
```

Tabela Cliente

```
CREATE TABLE Cliente (
 ClienteID INT PRIMARY KEY AUTO_INCREMENT,
 Nome VARCHAR(100) NOT NULL,
 Telefone VARCHAR(15),
 Endereco VARCHAR(255)
);
```

Tabela Veículo

```
CREATE TABLE Veiculo (
 VeiculoID INT PRIMARY KEY AUTO_INCREMENT,
 Placa VARCHAR(10) UNIQUE NOT NULL,
 Modelo VARCHAR(100),
 Marca VARCHAR(50),
 Ano INT,
 ClienteID INT,
 FOREIGN KEY (ClienteID) REFERENCES Cliente(ClienteID)
);
```

Tabela Ordem de Serviço (OS)

```
CREATE TABLE OS (
 OSID INT PRIMARY KEY AUTO_INCREMENT,
 VeiculoID INT,
 FOREIGN KEY (VeiculoID) REFERENCES Veiculo(VeiculoID),
 Data DATE,
 Status VARCHAR(20),
 MecanicoID INT,
 FOREIGN KEY (MecanicoID) REFERENCES Mecanico(MecanicoID),
 Equipe VARCHAR(50),
 ItemOSID INT,
 FOREIGN KEY (ItemOSID) REFERENCES ItemOS(ItemOSID),
 Quantidade INT,
 Valor DECIMAL(10,2),
 ValorTotal DECIMAL(10,2)
);
```

```

 OSID INT PRIMARY KEY AUTO_INCREMENT,
 VeiculoID INT,
 DataEmissao DATE NOT NULL,
 DataConclusao DATE,
 ValorTotal DECIMAL(10,2),
 Status ENUM('Aberta', 'Em Andamento', 'Concluída', 'Cancelada') NOT NULL,
 FOREIGN KEY (VeiculoID) REFERENCES Veiculo(VeiculoID)
);

```

Tabela Serviço

```

CREATE TABLE Servico (
 ServicoID INT PRIMARY KEY AUTO_INCREMENT,
 Descricao VARCHAR(255) NOT NULL,
 PrecoMaoDeObra DECIMAL(10,2) NOT NULL
);

```

Tabela Peça

```

CREATE TABLE Peca (
 PecaID INT PRIMARY KEY AUTO_INCREMENT,
 Descricao VARCHAR(255),
 Preco DECIMAL(10,2) NOT NULL
);

```

Tabela Mecânico

```

CREATE TABLE Mecanico (
 MecanicoID INT PRIMARY KEY AUTO_INCREMENT,
 Nome VARCHAR(100) NOT NULL,
 Endereco VARCHAR(255),
 Especialidade VARCHAR(100)
);

```

Tabela Equipe

```

CREATE TABLE Equipe (
 EquipeID INT PRIMARY KEY AUTO_INCREMENT,
 NomeEquipe VARCHAR(100)
);

```

Tabela Equipe\_Mecanico (Associação N:N entre Equipe e Mecânico)

```

CREATE TABLE Equipe_Mecanico (
 EquipeID INT,
 MecanicoID INT,
 PRIMARY KEY (EquipeID, MecanicoID),
 FOREIGN KEY (EquipeID) REFERENCES Equipe(EquipeID),
 FOREIGN KEY (MecanicoID) REFERENCES Mecanico(MecanicoID)
);

```

Tabela Item\_OS (Associação N:N entre OS e Serviço)

```

CREATE TABLE Item_OS (

```

```

ItemOSID INT PRIMARY KEY AUTO_INCREMENT,
OSID INT,
ServicoID INT,
Quantidade INT,
ValorServico DECIMAL(10,2),
FOREIGN KEY (OSID) REFERENCES OS(OSID),
FOREIGN KEY (ServicoID) REFERENCES Servico(ServicoID)
);

```

```

Tabela Peca_OS (Associação N:N entre OS e Peça)
CREATE TABLE Peca_OS (
 PecaOSID INT PRIMARY KEY AUTO_INCREMENT,
 OSID INT,
 PecaID INT,
 Quantidade INT,
 ValorPeca DECIMAL(10,2),
 FOREIGN KEY (OSID) REFERENCES OS(OSID),
 FOREIGN KEY (PecaID) REFERENCES Peca(PecaID)
);
...

```

### 3. Inserção de Dados para Testes

```

```sql

```

Inserindo Clientes

```

INSERT INTO Cliente (Nome, Telefone, Endereco)
VALUES ('João da Silva', '11999999999', 'Rua A, 123'),
('Maria Souza', '21988888888', 'Avenida B, 456');

```

Inserindo Veículos

```

INSERT INTO Veiculo (Placa, Modelo, Marca, Ano, ClienteID)
VALUES ('ABC1234', 'Civic', 'Honda', 2020, 1),
('XYZ5678', 'Corolla', 'Toyota', 2019, 2);

```

Inserindo Serviços

```

INSERT INTO Servico (Descricao, PrecoMaoDeObra)
VALUES ('Troca de óleo', 100.00),
('Revisão completa', 500.00);

```

Inserindo Peças

```

INSERT INTO Peca (Descricao, Preco)
VALUES ('Óleo 5W30', 30.00),
('Filtro de Ar', 50.00);

```

Inserindo Mecânicos

```

INSERT INTO Mecanico (Nome, Endereco, Especialidade)
VALUES ('Carlos Mendes', 'Rua C, 789', 'Eletricista'),
('Fernanda Lima', 'Rua D, 101', 'Mecânica Geral');

```

Inserindo Equipes

```
INSERT INTO Equipe (NomeEquipe)
VALUES ('Equipe Alfa'),
       ('Equipe Beta');
```

Associando Mecânicos a Equipes

```
INSERT INTO Equipe_Mecanico (EquipeID, MecanicoID)
VALUES (1, 1), (1, 2),
       (2, 1);
```

Inserindo Ordens de Serviço

```
INSERT INTO OS (VeiculoID, DataEmissao, ValorTotal, Status)
VALUES (1, '20240901', 130.00, 'Aberta'),
       (2, '20240902', 550.00, 'Aberta');
```

Associando Serviços a Ordens de Serviço

```
INSERT INTO Item_OS (OSID, ServicoID, Quantidade, ValorServico)
VALUES (1, 1, 1, 100.00),
       (2, 2, 1, 500.00);
```

Associando Peças a Ordens de Serviço

```
INSERT INTO Peca_OS (OSID, PecaID, Quantidade, ValorPeca)
VALUES (1, 1, 1, 30.00),
       (2, 2, 1, 50.00);
...
```

4. Consultas SQL

1. Recuperações Simples com SELECT

```
```sql
Listar todos os clientes
SELECT FROM Cliente;
```
```

2. Filtros com WHERE

```
```sql
Buscar ordens de serviço que estão abertas
SELECT FROM OS WHERE Status = 'Aberta';
```
```

3. Expressões para Atributos Derivados

```
```sql
Calcular o valor total gasto por cada cliente
SELECT Cliente.Nome, SUM(OS.ValorTotal) AS TotalGasto
FROM Cliente
```

```
JOIN Veiculo ON Cliente.ClienteID = Veiculo.ClienteID
JOIN OS ON Veiculo.VeiculoID = OS.VeiculoID
GROUP BY Cliente.Nome;
...
```

#### 4. Ordenação com ORDER BY

```
```sql
Listar serviços ordenados pelo preço da mão de obra
SELECT Descricao, PrecoMaoDeObra FROM Servico ORDER BY PrecoMaoDeObra
DESC;
...
```

5. Condições de Filtros com HAVING

```
```sql
Mostrar clientes que gastaram mais de R$ 1000 em ordens de serviço
SELECT Cliente.Nome, SUM(OS.ValorTotal) AS TotalGasto
FROM Cliente
JOIN Veiculo ON Cliente.ClienteID = Veiculo.ClienteID
JOIN OS ON Veiculo.VeiculoID = OS.VeiculoID
GROUP BY Cliente.Nome
HAVING TotalGasto > 1000;
...
```

#### 6. Junções entre Tabelas

```
```sql
Relação de veículos e seus proprietários
SELECT Veiculo.Placa, Veiculo.Modelo, Cliente.Nome AS Proprietario
FROM Veiculo
JOIN Cliente ON Veiculo.ClienteID = Cliente.ClienteID;
...
```

7. Quantidade de Ordens de Serviço por Cliente

```
```sql
Quantas ordens de serviço foram feitas por cada cliente
SELECT Cliente.Nome, COUNT(OS.OSID) AS TotalOS
FROM Cliente
JOIN Veiculo ON Cliente.ClienteID = Veiculo.ClienteID
JOIN OS ON Veiculo.VeiculoID = OS.VeiculoID
GROUP BY Cliente.Nome;
...
```

#### Conclusão

Neste projeto, criamos um esquema lógico completo para uma oficina mecânica com suporte a múltiplas entidades e relacionamentos. O modelo inclui consultas complexas,

como junções, agregações e filtros avançados, permitindo uma visão abrangente das operações realizadas no sistema da oficina.

O script SQL acima pode ser usado para criar o banco de dados, inserir dados de teste e executar consultas que atendam aos requisitos do cenário.