

UNIVERSIDADE FEDERAL DE LAVRAS



TRABALHO

Banco de Dados I - GCC114

Parte III

Alunos: Augusto Soares Pereira - 201210320

Álvaro dos Reis Cozadi - 201211012

Eugênio Costa Portella - 201020887

Emerson Luiz Antonio - 201211216

Lavras - MG

2014

Sumário

1. Descrição do Minimundo.....	pág. 3
2. Descrição da Aplicação.....	pág. 3
3. Descrição dos tipos de Dados.....	pág. 4
3.1 Entidades.....	pág. 5
3.2 Relacionamentos.....	pág. 6
4. Diagrama ER.....	pág. 7
5. Modelo Relacional.....	pág. 8
6. Restrições de Exclusão.....	pág. 9
7. Dicionário de Dados (Modelo Relacional).....	pág. 10
8. Comandos SQL.....	pág. 14
8.1 Tabelas.....	pág. 14
8.2 Inserção de dados.....	pág. 16
8.3 Exemplos de ALTER TABLE e DROP TABLE.....	pág. 19
8.4 Exemplos de DELETE.....	pág. 20
8.5 Exemplos de UPDATE.....	pág. 22
8.6 Exemplos de VIEWS.....	pág. 23
8.7 Exemplos de Consultas.....	pág. 24
8.8 Exemplos de STORED PROCEDURE.....	pág. 29

1. Descrição do Minimundo

O nosso trabalho aborda um contexto de uma oficina de manutenção de veículos e venda de produtos do gênero.

A oficina foi organizada na seguinte forma. Existe a possibilidade de cadastramento de clientes com os seguintes componentes, nome do cliente, CPF do cliente como forma de identificação do mesmo no banco de dados (único), o seu endereço e um contato.

A oficina também conta com um cadastro de veículos. Para cada veículo cadastrado é necessário informar o número de chassi (único), modelo do veículo, sua marca, seu ano de fabricação e também o valor da sua placa (único). E um cliente pode cadastrar vários veículos diferentes.

O controle de pedidos é feito através de Ordens de Serviço. Para cada ordem é necessário informar o seu número de identificação (único), o seu valor total e se for preciso, uma informação complementar. Um único cliente pode solicitar mais de uma ordem de serviço e um único veículo pode possuir mais de uma ordem de serviço.

Cada ordem de serviço, pode conter um serviço de manutenção com os seguintes dados, número de identificação de cada serviço (único), seu valor e se for preciso, uma breve descrição. E ou um produto do gênero com os seguintes dados, valor do produto, sua quantidade, seu custo para a oficina, seu número de identificação (único) e se for preciso, uma breve descrição.

A oficina ainda conta com um cadastro de fornecedores. Para cada fornecedor cadastrado é necessário informar o seu número de identificação (único) e o seu nome. Um mesmo fornecedor, pode fornecer mais de um produto para a oficina.

2. Descrição da Aplicação

Temos por objetivo a produção de um sistema que ajude no gerenciamento operacional de uma oficina de manutenção de veículos. Nele serão oferecidos serviços como:

- Cadastramento de Clientes, veículos e fornecedores;
- Controle de cada ordem de serviço;
- Gerenciamento dos serviços prestados e dos produtos vendidos.

3. Descrição do Tipos de Dados

3.1 - Entidades

Tipo de Entidade	VEÍCULO		
Descrição	Todos os veículos dos clientes cadastrados		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
Numero_Chassi	Forma de registro universal dos veículos automotivos produzidos	String[17]	Não
Modelo	Nome de identificação de uma carro	String[20]	Não
Ano_Fab	Ano de fabricação	Data	Sim
Placa	Identificação de veículos	String[8]	Não
Marca	Nome da fábrica do veículo	String[10]	Sim

Tipo de Entidade	O.S		
Descrição	Pedidos realizados pelos clientes da oficina		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
ID_O.S	Número de identificação	Numérico	Não
Data	Data da ordem de serviço	Data	Não
Valor_Total	Quantia do serviço ou produto prestado pela oficina	Numérico	Não
Info_complementar	Informação complementar	String[45]	Sim

Tipo de Entidade	CLIENTE		
Descrição	Todos os clientes da oficina cadastrados		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
CPF	Número de identificação de pessoas	String[14]	Não

Nome_Cliente	Nome do cliente	String[30]	Não
Endereço	Endereço de cada cliente	String[45]	Não
Contato	Meio de contato com o cliente	String[13]	Não

Tipo de Entidade	SERVIÇO		
Descrição	Todos os diferentes tipos de serviços solicitados pelos clientes		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
ID_Serviço	Número de identificação e cada serviço	Numérico	Não
Valor_Serviço	Valor do serviço prestado	Numérico	Não
Descrição_Serviço	Número de identificação único	String[45]	Sim

Tipo de Entidade	PRODUTO		
Descrição	Todos os diferentes produtos vendidos pela oficina		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
ID_Produto	Número de identificação único	Numérico	Não
Custo	Custo de cada produto	Numérico	Não
Quantidade	Quantidade de cada produto no estoque da oficina	Numérico	Não
Descrição_Porduto	Descrição dos produtos oferecidos pela oficina	String[45]	Sim

Tipo de Entidade	FORNECEDOR		
Descrição	Todos os fornecedores da oficina		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
<u>ID_Fornecedor</u>	Número de identificação único	Numérico	Não
Nome_Fornecedor	Nome de cada fornecedor da oficina	String[45]	Não

3.2 – Relacionamentos

Tipo de Relacionamento	Pertence
Descrição	Veículos pertencentes aos clientes

Tipo de Relacionamento	Possui
Descrição	Ordem de serviço do veículo

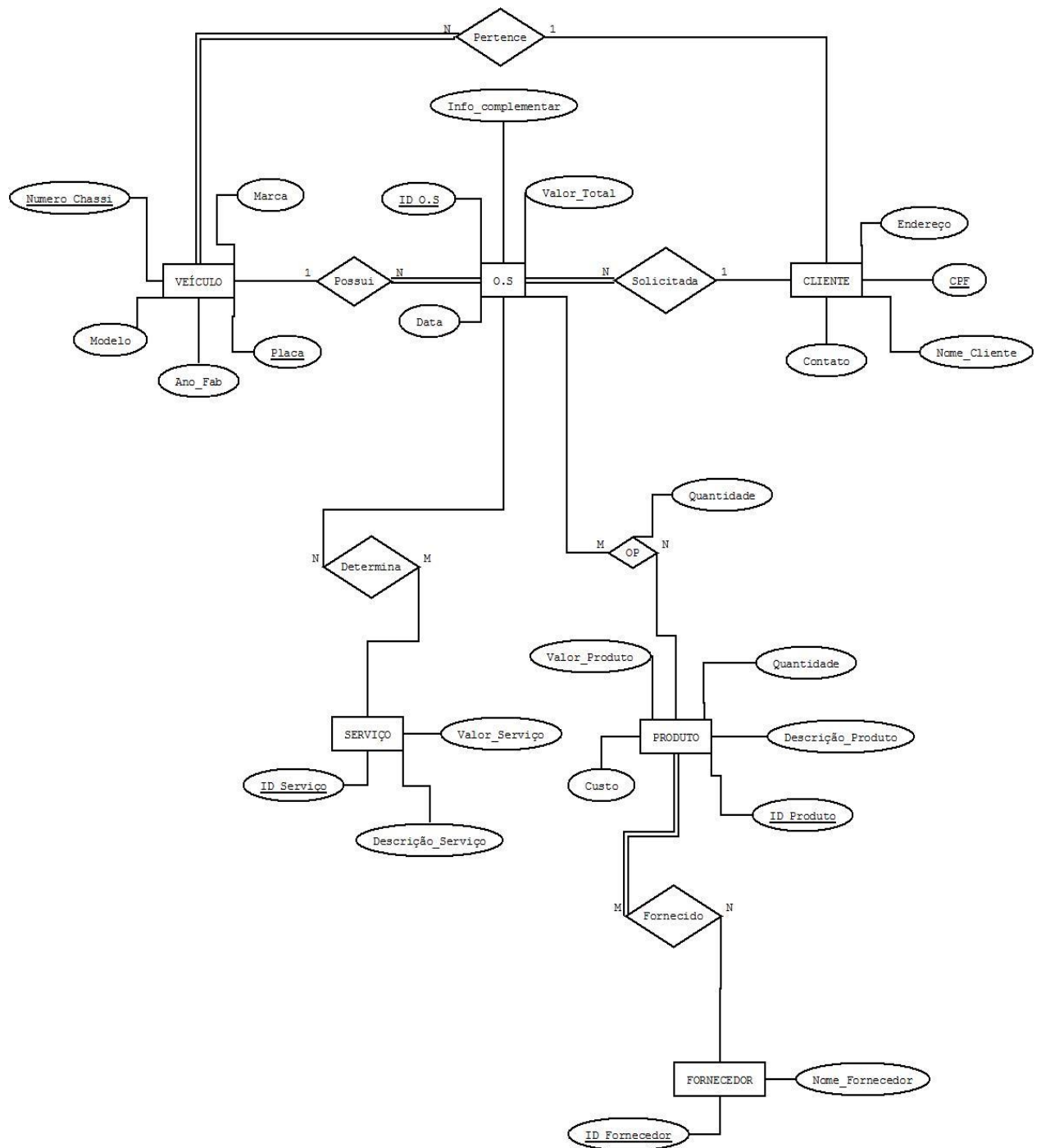
Tipo de Relacionamento	Solicitada
Descrição	Ordem de serviço solicitada pelo cliente

Tipo de Relacionamento	Determina
Descrição	Serviço determinado pela ordem de serviço

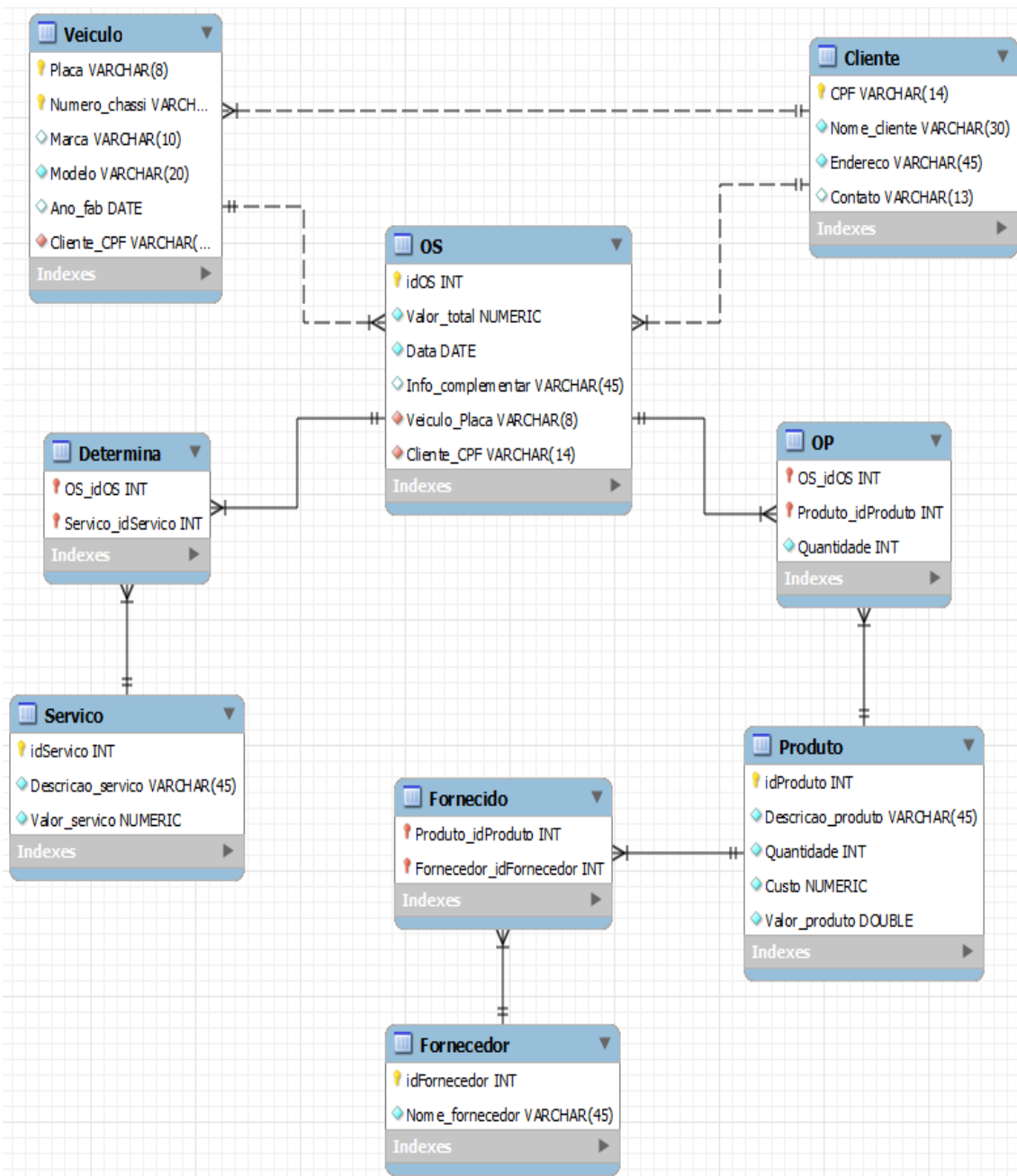
Tipo de Relacionamento	OP
Descrição	Produtos solicitados pela ordem de serviço

Tipo de Relacionamento	Fornecido
Descrição	Produtos fornecidos pelos fornecedores

4. Diagrama Entidade-Relacional



5. Modelo Relacional



6. Restrições de Exclusão

Cliente (CPF(PK), Nome_Cliente, Endereco, Contato)

Veiculo (Placa, Numero_Chassi, Marca, Modelo, Ano_Fab, CPF_Cliente(FK))

Veiculo[CPF_Cliente] → (b) Cliente[CPF]

O.S (ID_O.S(PK), ValorTotal, Data, CPF_Cliente(FK), Placa(FK), Info_Complementar)

O.S[CPF_Cliente] → (b) Cliente[CPF]

O.S[Placa] → (b) Veiculo[Placa]

Servico (ID_Servico(PK), Valor_Servico, Descricao_Servico)

Determina (ID_O.S(PK)(FK), ID_Servico(PK)(FK))

Determina[ID_O.S] → (p) O.S[ID_O.S]

Determina[ID_Servico] → (p) Servico[ID_Servico]

Produto (ID_Produto(PK), Valor_Produto, Custo, Quantidade, Descricao_Produto)

OP (ID_O.S(PK)(FK), ID_Produto(PK)(FK), Quantidade)

OP[ID_O.S] → (p) O.S[ID_O.S]

OP[ID_Produto] → (p) Produto[ID_Produto]

Fornecedor (ID_Fornecedor(PK), Nome_Fornecedor)

Fornecido (ID_Fornecedor(PK)(FK), ID_Produto(PK)(FK))

Fornecido[ID_Fornecedor] → (p) Fornecedor[ID_Fornecedor]

Fornecido[ID_Produto] → (p) Fornecedor[ID_Produto]

7. Dicionário de Dados (Modelo Relacional)

Relação	Cliente		
Descrição	Todos os clientes da oficina cadastrados		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
CPF	Número de identificação de pessoas	String[14]	Não
Nome_Cliente	Nome do cliente	String[30]	Não
Endereço	Endereço de cada cliente	String[45]	Não
Contato	Meio de contato com o cliente	String[13]	Não

Relação	Veiculo		
Descrição	Todos dos clientes cadastrados		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
Numero_Chassi	Forma de registro universal dos veículos automotivos produzidos	String[17]	Não
Modelo	Nome de identificação de uma carro	String[20]	Não
Ano_Fab	Ano de fabricação	Data	Sim
Placa	Identificação de veículos	String[8]	Não
Marca	Nome da fábrica do veículo	String[10]	Sim
CPF_Cliente	CPF dos clientes donos dos veículos	String[14]	Não

Relação	O.S		
Descrição	Pedidos realizados pelos clientes da oficina		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
ID_O.S	Número de identificação	Numérico	Não

Data	Data da ordem de serviço	Data	Não
Valor_Total	Quantia do serviço ou produto prestado pela oficina	Numérico	Não
Info_complementar	Informação complementar	String[45]	Sim
CPF_Cliente	CPF dos clientes donos dos veículos	String[14]	Não
Placa	Placa do veículo	String[8]	Não

Relação	Serviço		
Descrição	Todos os diferentes tipos de serviços solicitados pelos clientes		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
ID_Serviço	Número de identificação e cada serviço	Numérico	Não
Valor_Serviço	Valor do serviço prestado	Numérico	Não
Descrição_Serviço	Número de identificação único	String[45]	Sim

Relação	Determina		
Descrição	Registro de todas as ordens de serviços emitidas		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
ID_Servico	Número de identificação de cada serviço	Numérico	Não
ID_O.S	Número de identificação de cada ordem de serviço	Numérico	Não

Relação	Produto		
Descrição	Todos os diferentes produtos vendidos pela oficina		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
ID_Produto	Número de identificação único	Numérico	Não
Custo	Custo de cada produto	Numérico	Não
Quantidade	Quantidade de cada produto no estoque da oficina	Numérico	Não
Descrição_Porduto	Descrição dos produtos oferecidos pela oficina	String[45]	Sim

Relação	OP		
Descrição	Registro de todos os pedidos de produtos realizados		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
ID_Produto	Número de identificação de cada produto	Numérico	Não
ID_O.S	Número de identificação de cada ordem de serviço	Numérico	Não
Quantidade	Quantidade de produtos	Numérico	Não

Relação	Fornecedor		
Descrição	Todos os fornecedores da oficina		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
<u>ID_Fornecedor</u>	Número de identificação único	Numérico	Não
Nome_Fornecedor	Nome de cada fornecedor da oficina	String[41]	Não

Relação	Fornecido		
Descrição	Todos os produtos fornecidos para a oficina		
Atributos			
Nome	Descrição	Domínio	Permite Nulo(S/N)
<u>ID_Fornecedor</u>	Número de identificação único	Numérico	Não
ID_Produto	Nome de cada produto	Numérico	Não

8. Comandos SQL

8.1 – Tabelas

```
CREATE TABLE Cliente (  
    CPF                VARCHAR(14)        NOT NULL,  
    Nome_Cliente       VARCHAR(30)        NOT NULL,  
    Endereco           VARCHAR(30)        NOT NULL,  
    Contato            VARCHAR(13)        NOT NULL,  
    CONSTRAINT pk_CPF PRIMARY KEY (CPF),  
    CONSTRAINT uk_nome UNIQUE (Nome_Cliente)  
);  
  
CREATE TABLE Veiculo (  
    Placa              VARCHAR(8)         NOT NULL,  
    Numero_Chassi      VARCHAR(17)       NOT NULL,  
    Marca              VARCHAR(20)        NULL,  
    Modelo             VARCHAR(20)       NOT NULL,  
    Ano_Fab            DATE               NULL,  
    CPF_Cliente        VARCHAR(14)       NOT NULL,  
    CONSTRAINT pk_placa PRIMARY KEY (Placa),  
    CONSTRAINT uk_chassi UNIQUE (Numero_Chassi),  
    CONSTRAINT fk_cpf FOREIGN KEY (CPF_Cliente) REFERENCES  
    Cliente (CPF) ON UPDATE RESTRICT  
);  
  
CREATE TABLE OS (  
    id_OS              int               NOT NULL,  
    Valor_Total        DECIMAL(8,2)     NOT NULL,  
    Data_              DATE              NOT NULL,  
    CPF_Cliente        VARCHAR(14)      NOT NULL,  
    Placa              VARCHAR(8)        NOT NULL,  
    Info_Complementar  VARCHAR(20)      NULL,  
    CONSTRAINT pk_idOS PRIMARY KEY (id_OS),  
    CONSTRAINT fk_cpfCli FOREIGN KEY (CPF_Cliente)  
    REFERENCES Cliente (CPF) ON UPDATE RESTRICT,
```

```

        CONSTRAINT fk_placa FOREIGN KEY (Placa) REFERENCES
Veiculo (Placa) ON UPDATE RESTRICT
);

```

```

CREATE TABLE Servico (
    id_Servico          INT          NOT NULL,
    Valor_Servico       DECIMAL(8,2) NOT NULL,
    Descricao_Servico   VARCHAR(20)  NULL,
    CONSTRAINT pk_id_serv PRIMARY KEY (id_Servico)
);

```

```

CREATE TABLE Determina (
    id_OS              INT          NOT NULL,
    id_Servico         INT          NOT NULL,
    CONSTRAINT pk_determina PRIMARY KEY (id_OS,
id_Servico),
    CONSTRAINT fk_determina_os FOREIGN KEY (id_OS)
REFERENCES OS (id_OS) ON UPDATE CASCADE,
    CONSTRAINT fk_determina_serv FOREIGN KEY (id_Servico)
REFERENCES Servico (id_Servico) ON UPDATE CASCADE
);

```

```

CREATE TABLE Produto (
    id_Produto         INT          NOT NULL,
    Valor_Produto      DECIMAL(8,2) NOT NULL,
    Custo              DECIMAL(8,2) NOT NULL,
    Quantidade         INT          NOT NULL,
    Descricao_Produto  VARCHAR(45)  NULL,
    CONSTRAINT pk_idprod PRIMARY KEY (id_Produto)
);

```

```

CREATE TABLE OP (
    id_OS              INT          NOT NULL,
    id_Produto         INT          NOT NULL,
    Quantidade         INT          NOT NULL,
    CONSTRAINT pk_op PRIMARY KEY (id_OS, id_Produto),

```

```

        CONSTRAINT fk_op_idos FOREIGN KEY (id_OS) REFERENCES OS
(id_OS) ON UPDATE CASCADE,

        CONSTRAINT fk_op_idprod FOREIGN KEY (id_Produto)
REFERENCES Produto (id_Produto) ON UPDATE CASCADE

);

```

```

CREATE TABLE Fornecedor (

    id_Fornecedor          INT          NOT NULL,

    Nome_Fornecedor        VARCHAR(45)  NOT NULL,

    CONSTRAINT pk_idforn PRIMARY KEY (id_Fornecedor)

);

```

```

CREATE TABLE Fornecido (

    id_Fornecedor          INT          NOT NULL,

    id_Produto              INT          NOT NULL,

    CONSTRAINT pk_fornecido PRIMARY KEY (id_Fornecedor,
id_Produto),

    CONSTRAINT fk_fornecido_for FOREIGN KEY (id_Fornecedor)
REFERENCES Fornecedor (id_Fornecedor) ON UPDATE CASCADE,

    CONSTRAINT fk_fornecido_pro FOREIGN KEY (id_Produto)
REFERENCES Produto (id_Produto) ON UPDATE CASCADE

);

```

8.2 – Inserção de Dados

Cadastro de clientes:

```

INSERT INTO Cliente
VALUES ('101.111.121-01','João B. Silva','R. Guaicui,
175','(35)3821-0921');
INSERT INTO Cliente
VALUES ('102.222.121-02','Frank T. Santos','R. Gentios,
22','(35)3822-0671');
INSERT INTO Cliente
VALUES ('103.333.121-03','Alice N. Pereira','R. Curitiba,
11','(35)3821-0515');
INSERT INTO Cliente
VALUES ('104.444.121-04','Júnia B. Mendes','R. E. Santos,
123','(35)3822-0706');
INSERT INTO Cliente
VALUES ('105.555.121-05','José S. Tavares','R. Iraí,
153','(35)3821-1012');
INSERT INTO Cliente
VALUES ('106.666.121-06','Luciana S. Santos','R. Iraí,
175','(35)3822-6010');

```



```

INSERT INTO Cliente
VALUES ('107.777.121-07','Maria P. Ramos','R. C. Linhares,
10','(35)3821-1105');
INSERT INTO Cliente
VALUES ('108.888.121-08','Jaime A. Mendes','R. Bahia,
111','(35)3821-1125');

```

Cadastro de veículos:

```

INSERT INTO Veiculo
VALUES ('HKP-3880','RD08X0.4123423567','Ford','Focus
Sedan','2005/02/01','101.111.121-01');
INSERT INTO Veiculo
VALUES ('GHP-
2301','OU87X8.4562523546','Volkswagen','Polo','2008/06/03',
'102.222.121-02');
INSERT INTO Veiculo
VALUES ('CQC-
1232','AU32X8.7747538538','FIAT','Strada','2010/04/12','103
.333.121-03');
INSERT INTO Veiculo
VALUES ('PEF-
0337','AU08X8.8248124817','FIAT','Ideia','2013/12/09','104.
444.121-04');
INSERT INTO Veiculo
VALUES ('NIY-
0045','ER20X8.3875668138','Volkswagen','Gol','2010/11/20','
105.555.121-05');
INSERT INTO Veiculo
VALUES ('PER-2012','BN12X8.1835672357','Citroen','C4
Lounge','2014/02/20','106.666.121-06');
INSERT INTO Veiculo
VALUES ('DUQ-
6219','CV09X8.1763652673','Honda','Civic','2012/08/20','107
.777.121-07');
INSERT INTO Veiculo
VALUES ('HHF-
7713','AK47X8.9876341362','Volkswagen','Fusca','1967/02/20'
,'108.888.121-08');

```

Cadastro de ordem de serviço:

```

INSERT INTO OS
VALUES (1,550,'2014/07/04','101.111.121-01','HKP-
3880','Pneu + Oleo.');
```

```

INSERT INTO OS
VALUES (2,100,'2014/01/04','108.888.121-08','HHF-
7713','Rev. eletrica');
```

```

INSERT INTO OS
VALUES (3,450,'2013/07/04','105.555.121-05','NIY-
0045','Pneus');
```

```

INSERT INTO OS
VALUES (4,500,'2014/07/04','103.333.121-03','CQC-
1232','Troca Freio');
```

```

INSERT INTO OS
VALUES (5,300,'2014/07/04','102.222.121-02','GHP-
2301','Rev. Ar Bag');
```

```

INSERT INTO OS

```

```
VALUES (6,1000,'2014/07/08','104.444.121-04','PEF-0337','Rev. Completa');
INSERT INTO OS
VALUES (7,50,'2013/07/04','106.666.121-06','PER-2012','Filtro ar');
INSERT INTO OS
VALUES (8,100,'2014/01/04','107.777.121-07','DUQ-6219','Troca oleo');
```

Cadastro dos tipos de serviço:

```
INSERT INTO Servico
VALUES (10,450,'Troca de pneus');
INSERT INTO Servico
VALUES (20,100,'Troca de oleo');
INSERT INTO Servico
VALUES (30,1000,'Revisão completa');
INSERT INTO Servico
VALUES (40,500,'Pastilhas de freio');
INSERT INTO Servico
VALUES (50,300,'Revisão Ar Bag');
INSERT INTO Servico
VALUES (60,100,'Revisão eletrica');
INSERT INTO Servico
VALUES (70,300,'Manut. Ar-cond.');
```

Tabela Determina:

```
INSERT INTO Determina
VALUES (1,10);
INSERT INTO Determina
VALUES (1,20);
INSERT INTO Determina
VALUES (2,60);
INSERT INTO Determina
VALUES (3,10);
INSERT INTO Determina
VALUES (4,40);
INSERT INTO Determina
VALUES (5,50);
INSERT INTO Determina
VALUES (6,30);
INSERT INTO Determina
VALUES (7,80);
INSERT INTO Determina
VALUES (8,20);
```

Cadastro dos tipos de produtos:

```
INSERT INTO Produto
VALUES (100,100,50,300,'Pneu novo');
INSERT INTO Produto
VALUES (200,50,30,80,'Filtro de Oleo');
INSERT INTO Produto
VALUES (300,400,80,25,'Pastilhas freio');
INSERT INTO Produto
VALUES (400,30,15,10,'Filtro de ar');
```

Tabela OP:

```
INSERT INTO OP
VALUES (1,100,4);
INSERT INTO OP
VALUES (1,200,1);
INSERT INTO OP
VALUES (3,100,4);
INSERT INTO OP
VALUES (7,400,1);
INSERT INTO OP
VALUES (8,200,2);
```

Cadastro de fornecedores:

```
INSERT INTO Fornecedor
VALUES (1,'ABC Produtos');
INSERT INTO Fornecedor
VALUES (15,'SK Automotive');
INSERT INTO Fornecedor
VALUES (5,'Lubrax Oil');
INSERT INTO Fornecedor
VALUES (20,'STP Oil');
```

Tabela Fornecido:

```
INSERT INTO Fornecido
VALUES (1,100);
INSERT INTO Fornecido
VALUES (1,300);
INSERT INTO Fornecido
VALUES (15,100);
INSERT INTO Fornecido
VALUES (15,400);
INSERT INTO Fornecido
VALUES (5,200);
```

8.3 – Exemplos de ALTER TABLE e DROP TABLE**1. Exclusão do nome de todos os clientes da tabela Cliente.**

```
ALTER TABLE Cliente DROP
COLUMN Nome_Cliente CASCADE;
```

2. Inserção de um novo atributo na tabela Fornecedor.

```
ALTER TABLE Fornecedor
ADD COLUMN Data_Fundação DATE;
```

3. Cria um atributo que mantém a idade de cada cliente.

```
ALTER TABLE Cliente ADD
COLUMN Idade INT;
```

4. Remover o atributo contato da tabela cliente.

```
ALTER TABLE Cliente DROP  
COLUMN Contato CASCADE;
```

5. Remover a restrição "fk_placa", referente a chave estrangeira Placa, da relação OS.

```
ALTER TABLE OS DROP foreign key fk_placa; -- Padrão SQL:  
ALTER TABLE OS DROP CONSTRAINT fk_placa;
```

6. Remover a restrição "fk_fornecido_for", referente a chave estrangeira id_fornecedor da relação Fornecido.

```
ALTER TABLE Fornecido DROP foreign key fk_fornecido_por;
```

7. Remover a restrição "fk_determina_os", referente a chave estrangeira id_OS na relação OS.

```
ALTER TABLE Determina DROP FOREIGN KEY fk_determina_os;
```

8. Remover a restrição "fk_op_idos", referente a chave estrangeira id_OS na relação OS.

```
ALTER TABLE OP DROP FOREIGN KEY fk_op_idos;
```

9. Remover a restrição "fk_cpf", referente a chave estrangeira CPF_Cliente na relação Veículo.

```
ALTER TABLE Veiculo DROP FOREIGN KEY fk_cpf;
```

10. Remover a restrição "fk_cpfCli", referente a chave estrangeira CPF_Cliente na relação Veículo.

```
ALTER TABLE OS DROP FOREIGN KEY fk_cpfCli;
```

11. Remover o esquema de banco de dados Oficina e todas as suas tabelas, domínios e outros elementos.

```
DROP SCHEMA Oficina;
```

8.4 - Exemplos de DELETE:

1. Remover a tupla de Cliente com CPF igual a "101.111.121-01".

```
DELETE FROM Cliente  
WHERE CPF = '101.11.121-01';
```

2. Remover a tupla de cliente com o nome igual a "João".

```
DELETE FROM Cliente
WHERE Nome_Cliente LIKE 'João';
```

3. Remover a tupla de veículo com placa igual a "PEF-0337".

```
DELETE FROM Veiculo
WHERE Placa = 'PEF-0337';
```

4. Remover todos os Fornecedores do banco de dados.

```
DELETE FROM Fornecedor;
```

5. Remover o serviço com o maior custo da oficina.

```
DELETE FROM Servico
WHERE Valor_Servico = (SELECT Valor_Servico
                        FROM Servico
                        WHERE MAX(Valor_Servico));
```

6. Exclua todos as ordens de serviço com valor total entre R\$500,00 e R\$800,00.

```
DELETE FROM OS
WHERE Valor_Total BETWEEN 500 AND 800;
```

7. Exclua todos os fornecedores que não forneceram nenhum produto para a oficina.

```
DELETE FROM Fornecedor
WHERE id_Fornecedor NOT IN (SELECT id_Fornecedor
                             FROM Fornecido);
```

8. Exclua todas as tuplas CPF de cada cliente dono de carro de marca Volkswagen.

```
DELETE FROM Cliente
WHERE CPF IN (SELECT CPF_Cliente
              FROM Veiculo
              WHERE Marca = 'Volkswagen');
```

9. Exclua todas as ordens de serviço com o valor total abaixo da média de todas as ordens de serviço.

```
DELETE FROM OS  
  
WHERE Valor_Total < (SELECT AVG(Valor_Total)  
                     FROM OS);
```

8.5 - Exemplos de UPDATE:

1. Atualizar os valores do tipo de serviço com ID igual a "10".

```
UPDATE Servico SET Valor_Servico = 1200, Descricao_Servico  
= 'Troca de Radiador'  
  
WHERE id_Servico = 10;
```

2. Atualizar os valores do cliente com o nome igual a "Jaime".

```
UPDATE Cliente SET Endereco = 'R. Enriquez, 333', Contato =  
'(35) 3821-0845'  
  
WHERE Nome_Cliente LIKE 'Jaime';
```

3 Atualizar os valores do fornecedor com o ID igual a "5".

```
UPDATE Fornecedor SET Nome_Fornecedor = 'STP Oil'  
  
WHERE id_Fornecedor = 5;
```

4. Atualizar os valores do produto com ID igual a "100".

```
UPDATE Produto SET Valor_Produto = 35, Custo = 80,  
Quantidade = 200, Descricao_Produto = 'Palheta de  
Parabrisa'  
  
WHERE id_Produto = 100;
```

5. Atualizar os valores do veículo com placa igual a "PEF-0337".

```
UPDATE Veiculo SET Numero_Chassi = 'AU08X8.8248124817',  
Marca = 'Honda', Modelo = 'Fit', Ano_Fab = '2010/04/12',  
CPF_Cliente = '104.444.121-04'  
  
WHERE Placa = 'PEF-0337';
```

6. Realizar um aumento de 10% no(s) produto(s) com o menor custo.

```
UPDATE Produto
SET Custo = Custo * 1.10
WHERE Custo IN (SELECT MIN(Custo) FROM Produto);

/* Comando não executado pelo MySQL por causa da restrição
de não
poder utilizar o comando UPDATE em uma tabela com o mesmo
nome.*/
```

7. Realize um aumento nos preços em 10% de todos os produtos com um custo abaixo de R\$200,00, senão um aumento de 5%.

```
UPDATE Produto
SET Custo = CASE
    WHEN Custo <= 200 THEN Custo * 1.10
    ELSE Custo * 1.05
END;
```

8. Atualize em 5% os preços dos produtos que tenham sido fornecidos por algum fornecedor.

```
UPDATE Produto
SET Custo = Custo * 1.05
WHERE id_Produto IN (SELECT id_Produto
    FROM Fornecedor);
```

8.6 - Exemplos de VIEWS:

1. Informações sobre os fornecedores e os produtos fornecidos.

```
CREATE VIEW Fornecedores_Info(id_Fornecedor,
Nome_Fornecedor, id_Produto, Valor_Produto)
AS SELECT F.id_Fornecedor, F.Nome_Fornecedor, P.id_Produto,
P.Valor_Produto
FROM Fornecedor AS F, Fornecedor AS G, Produto AS P
WHERE F.id_Fornecedor = G.id_Fornecedor and G.id_Produto =
P.id_Produto;
```

Listar o nome dos fornecedores, o id do produto fornecido e o seu valor total.

```
SELECT Nome_Fornecedor, id_Produto, Valor_Produto
```

```
FROM Fornecedores_Info  
ORDER BY Nome_Fornecedor;
```

2. Informações contendo o nome e CPF dos clientes e também a placa e modelo dos seus veículos.

```
CREATE VIEW Cliente_Veiculos  
AS SELECT Nome_Cliente, CPF, Placa, Modelo  
FROM Cliente, Veiculo  
WHERE CPF = CPF_Cliente  
GROUP BY Nome_Cliente;
```

Listar o nome dos clientes com o modelos dos seus carros.

```
SELECT Nome_Cliente, Modelo  
FROM Cliente_Veiculos;
```

3. Informações de todas as ordem de serviço solicitadas pelos clientes.

```
CREATE VIEW Cliente_OS  
AS SELECT Nome_Cliente, CPF, Data, Valor_Total  
FROM Cliente, OS  
WHERE CPF = CPF_Cliente;
```

Selecionar o nome dos clientes e o valor total das suas ordens de serviços criadas antes de 2014.

```
SELECT Nome_Cliente, Valor_Total, Data  
FROM Cliente_OS  
WHERE DATA < '2014/01/01';
```

8.7 – Exemplos de Consultas

1. Listar todos os atributos da tabela Cliente.

```
SELECT *  
FROM Cliente;
```

2. Listar todos os atributos da tabela Veiculo.

```
SELECT *  
FROM Veiculo;
```


3. Listar todos os atributos da tabela Ordem de Serviço(OS).

```
SELECT *  
FROM OS;
```

4. Listar todos os atributos da tabela referente aos tipos de serviços.

```
SELECT *  
FROM Servico;
```

5. Listar todos os atributos da tabela referente aos tipos de produtos.

```
SELECT *  
FROM Produto;
```

6. Listar todos os atributos da tabela dos fornecedores.

```
SELECT *  
FROM Fornecedor;
```

7. Selecionar a marca e modelo de veiculos, cujo o numero de chassi seja igual a NULL.

```
SELECT Marca, Modelo  
FROM Veiculo  
WHERE Numero_Chassi IS NULL;
```

8. Selecionar a marca e modelo de veiculos, cujo o numero de chassi seja diferente de NULL.

```
SELECT Marca, Modelo, Numero_Chassi  
FROM Veiculo  
WHERE Numero_Chassi IS NOT NULL;
```

9. Selecionar o nome e o CPF do cliente com a marca e modelo dos seus veiculos, ordenados por modelo.

```
SELECT Nome_Cliente, CPF, Marca, Modelo  
FROM Cliente, Veiculo  
WHERE CPF = CPF_Cliente  
ORDER BY Modelo;
```

10. Buscar o nome do serviço que tenha o maior custo para o cliente.

```
SELECT Descricao_Servico, Valor_Servico
FROM Servico
WHERE Valor_Servico = (SELECT MAX(Valor_Servico) FROM
Servico);
```

11. Buscar o nome de todos os serviços e a quantidade de solicitações.

```
SELECT Descricao_Servico, id_Servico, COUNT(*) AS
Solicitações
FROM Determina NATURAL JOIN Servico
WHERE id_Servico = id_Servico
GROUP BY id_Servico;
```

12. Listar o id da ordem de serviço que foi solicitada antes de 01/01/2013 ou com um valor total entre 500 a 800 reais.

```
(SELECT id_OS
FROM OS
WHERE Data < '2014')
UNION
(SELECT id_OS
FROM OS
WHERE (Valor_Total BETWEEN 500 AND 800));
```

13. Liste o nome dos clientes com carros cadastrados da marca "Honda".

```
SELECT Nome_Cliente
FROM Cliente
WHERE EXISTS (SELECT *
FROM Veiculo
WHERE CPF = CPF_Cliente and Marca =
'Honda');
```

14. Liste o nome dos clientes que começam com a letra "J", o modelo do seu veículo, ano de sua fabricação e a descrição da ordem de serviço emitida para o veículo.

```
SELECT Nome_Cliente, Modelo, Ano_Fab, Info_Complementar
```

```
FROM Cliente AS C, OS AS O, Veiculo AS V
WHERE Nome_Cliente LIKE 'J%' AND C.CPF = V.CPF_Cliente AND
C.CPF = O.CPF_Cliente;
```

15 Listar os 3 primeiros nomes dos clientes da oficina.

```
SELECT Nome_Cliente
FROM Cliente
ORDER BY Nome_Cliente ASC LIMIT 3 OFFSET 0;
```

16. Liste o modelo dos veiculos fabricados antes de 2010 e marca diferente de Honda.

```
SELECT Modelo, Marca, Ano_Fab
FROM Veiculo
WHERE Ano_Fab < '2010' AND Marca != 'Honda';
```

17. Liste o nome do cliente que solicitou a ordem de serviço de número 3 e seu valor.

```
SELECT Nome_Cliente, Valor_Total
FROM Cliente, OS
where id_OS = 3 AND CPF = CPF_Cliente;
```

18. Liste o nome do fornecedor das peças do veículo de placa "DUQ-6219".

```
SELECT Nome_Fornecedor, Descricao_Produto
FROM Fornecedor AS F1, Fornecido AS F2, Produto AS P, OS AS O, OP AS OP
WHERE Placa = 'DUQ-6219' AND F1.id_Fornecedor =
F2.id_Fornecedor AND F2.id_Produto = P.id_Produto AND
P.id_Produto = OP.id_Produto AND OP.id_OS = O.id_OS;
```

19. Qual o valor total dos produtos instalados no veículo de placa "NIY-0045"? E quem é o proprietário? E seu contato?

```
SELECT Valor_Total, Nome_Cliente, Contato
FROM Cliente AS C, OS AS O
WHERE O.Placa = 'NIY-0045' AND O.CPF_Cliente = C.CPF;
```

20. Liste o nome dos clientes e modelo do veiculo, de donos de veiculos fabricados entre 2000 e 2010 da marca Ford.

```

SELECT Nome_Cliente, Modelo
FROM Cliente, Veiculo
WHERE NOT (Ano_Fab < '2000' OR Ano_Fab > '2010') AND Marca
= 'Ford' AND CPF = CPF_Cliente;

```

21. Liste os dados do(s) fornecedor(s) que tenham a letra "u" na segunda posição da palavra.

```

SELECT *
FROM Fornecedor
WHERE Nome_Fornecedor LIKE '_u%';

```

22. Liste o CPF dos clientes que estão cadastrados, mas não tem nenhum veículo cadastrado.

```

/*(SELECT CPF
FROM Cliente)
EXCEPT ou MINUS
(SELECT CPF_Cliente
FROM Veiculo);*/
SELECT CPF
FROM Cliente
WHERE (CPF) NOT IN (SELECT CPF_Cliente FROM Veiculo);

```

23. Liste o custo médio em cada ordem de serviço

```

SELECT id_OS, AVG(Valor_Total) AS Custo_Medio
FROM OS
GROUP BY id_OS;

```

24. Liste a placa do veículo, no qual o seu seu dono tem a letra "M" como primeira letra do seu nome.

```

SELECT Placa
FROM Veiculo
WHERE CPF_Cliente IN (SELECT CPF
                        FROM Cliente
                        WHERE Nome_Cliente LIKE 'M%');

```

25. Encontre todos os clientes que possuem no máximo um veículo de marca Honda.

```
SELECT Nome_Cliente
FROM Cliente
WHERE Nome_Cliente IN (SELECT Nome_Cliente
                        FROM Cliente, Veiculo
                        WHERE CPF = CPF_Cliente AND Marca = 'Honda');
```

26. Liste o número de ordens de serviço emitidas entre 2014/01/04 e 2014/04/04.

```
SELECT COUNT(id_OS) AS Número_de_OS
FROM OS
WHERE Data BETWEEN '2014/01/04' AND '2014/04/04';
```

27. Liste todos os valores dos fornecedores, inclusive aqueles que não forneceram produtos para a oficina.

```
SELECT *
FROM Fornecedor AS F1 LEFT OUTER JOIN Fornecido AS F2 ON
F1.id_Fornecedor = F2.id_Fornecedor;
```

28. Liste o id da ordem de serviço e a placa do veículo que apresentou um valor total o maior na data '2014/07/04'.

```
SELECT id_OS, Placa
FROM OS
WHERE Valor_Total > ALL (SELECT Valor_Total
                        FROM OS
                        WHERE Data = '2014/07/04');
```

29. Liste o id do fornecedor que forneceu mais de 2 produtos para a oficina.

```
SELECT id_Fornecedor, COUNT(*) AS Qtde_Produtos_Fornecidos
FROM Fornecido
GROUP BY id_Fornecedor
HAVING COUNT(*) >= 2;
```

8.8 - Exemplos de STORED PROCEDURE:

1. Obter todos os clientes cadastrados.

```

DELIMITER //
CREATE PROCEDURE ObtemTodosCli()
begin
    SELECT * FROM Cliente;
END //
DELIMITER ;

CALL ObtemTodosCli();

```

2. Obter todos os veículos cadastrados.

```

DELIMITER //
CREATE PROCEDURE ObtemTodosVei()
BEGIN
    SELECT * FROM Veiculo;
END //
DELIMITER ;

CALL ObtemTodosVei();

```

3. Obter todos os fornecedores cadastrados.

```

DELIMITER //
CREATE PROCEDURE ObtemTodosFor()
BEGIN
    SELECT * FROM Fornecedor;
END //
DELIMITER ;

CALL ObtemTodosFor();

```

4. Obter todos os veículos cadastrados e o seu dono.

```

DELIMITER //
CREATE PROCEDURE VeiCli()
BEGIN
    SELECT Nome_Cliente, Placa, Marca, Modelo

```

```

FROM Cliente, Veiculo
WHERE CPF = CPF_Cliente;
END //
DELIMITER ;

```

```
CALL VeiCli();
```

5. Obter todos os serviços de uma determinada ordem serviço passada por parâmetro.

```

DELIMITER //

CREATE PROCEDURE ServOS(IN p_id_OS DECIMAL(2))
BEGIN
    SELECT Descricao_Servico
    FROM OS AS O, Servico AS S, Determina AS D
    WHERE p_id_OS = O.id_OS AND O.id_OS = D.id_OS AND
    D.id_Servico = S.id_Servico;
END //

DELIMITER ;

CALL ServOS(1);

```

6. Obter o nome do fornecedor que forneceu um produto com o valor do custo passado por parâmetro.

```

DELIMITER //

CREATE PROCEDURE NomeFor(IN p_Custo DECIMAL(8), OUT
p_NomeFornecedor VARCHAR(45))
BEGIN
    SELECT Nome_Fornecedor INTO p_NomeFornecedor
    FROM Fornecedor AS F1, Fornecido AS F2, Produto AS P
    WHERE p_Custo = P.Custo AND P.id_Produto = F2.id_Produto
    AND F2.id_Fornecedor = F1.id_Fornecedor;
END //

DELIMITER ;

CALL NomeFor(80, @NomeFornecedor);

SELECT @NomeFornecedor AS NomeFornecedor;

```

7. Liste o mês de fabricação de cada veículo cadastrado.

```
DELIMITER //
```

```
CREATE PROCEDURE MesFab()
```

```
BEGIN
```

```
    SELECT CASE EXTRACT(MONTH FROM Ano_Fab)
```

```
        WHEN 01 THEN 'JAN'
```

```
        WHEN 02 THEN 'FEV'
```

```
        WHEN 03 THEN 'MAR'
```

```
        WHEN 04 THEN 'ABR'
```

```
        WHEN 05 THEN 'MAI'
```

```
        WHEN 06 THEN 'JUN'
```

```
        WHEN 07 THEN 'JUL'
```

```
        WHEN 08 THEN 'AGO'
```

```
        WHEN 09 THEN 'SET'
```

```
        WHEN 10 THEN 'OUT'
```

```
        WHEN 11 THEN 'NOV'
```

```
        WHEN 12 THEN 'DEZ'
```

```
    END AS Mes, GROUP_CONCAT(Modelo) AS Modelo
```

```
FROM Veiculo
```

```
GROUP BY mes;
```

```
END //
```

```
DELIMITER ;
```

```
CALL MesFab();
```

8. Procedimento que retorna 0 se dois nomes passados por parametro fazem parte do banco de dados, -1 se apenas um nome faz parte e -2 se nenhum nome faz parte do banco de dados.

```
DELIMITER //
```

```
CREATE PROCEDURE Checando (in pNome_cliente varchar(30), in
```

```
pNome_cliente2 varchar(30), out resultado varchar(100))
```

```
begin
```



```

        if exists (select * from cliente where Nome_cliente =
pNome_cliente) and exists (select * from cliente where
Nome_cliente = pNome_cliente2) then

            set resultado = 'Os dois nomes existem no banco da
dados';

            elseif exists (select * from cliente where
Nome_cliente = pNome_cliente) or exists (select * from
cliente where Nome_cliente = pNome_cliente2) then

                set resultado = 'Apenas um nome existe no banco da
dados';

            else

                set resultado = 'Nenhum nome existe no banco de
dados';

            end if;
        end//
DELIMITER ;

CALL Checando('Emerson Luiz', 'Augusto Soares',@num);
CALL Checando('Emerson Luiz', 'João B. Silva', @num);
CALL Checando('José S. Tavares', 'Jaime A. Mendes', @num);
SELECT @num AS num;

```

9. Se entrar com um número de id_Produto e se ele estiver na base de dados retorna produto 1, produto 2 e etc e se não estiver retorna, que não está na base de dados.

```

DELIMITER //

CREATE PROCEDURE Produto (in  p_id_Produto int(11), out
p_Descricao varchar(50))
BEGIN
    DECLARE into_var varchar(45);

    SELECT Descricao_Produto INTO into_var
    FROM Produto
    WHERE id_Produto = p_id_Produto;

    CASE into_var

```

```

        WHEN 'Pneu novo' THEN
            SET p_Descricao = 'Produto 1';

        WHEN 'Filtro de Oleo' THEN
            SET p_Descricao = 'Produto 2';

        WHEN 'Pastilhas freio' THEN
            SET p_Descricao = 'Produto 3';

        WHEN 'Filtro de ar' THEN
            SET p_Descricao = 'Produto 4';

        ELSE
            SET p_Descricao = 'Não está na base de dados';
    END CASE;

END//

DELIMITER ;

CALL Produto(100, @resultado);
CALL Produto(200, @resultado);
CALL Produto(1000, @resultado);
SELECT @resultado as Resposta;

```

10. Retorna a descrição do produto e a quantidade.

```

DELIMITER $$

CREATE PROCEDURE WhileLoopProc()
BEGIN
    Declare var int;
    set var = 0;
    WHILE (var <> 3) DO
        if (var = 0) then
            SELECT contato, COUNT(*) as Qtde
            FROM cliente

```

```

        GROUP BY contato;

elseif (var = 1) then
    SELECT Descricao_Produto, COUNT(*) as Qtde
    FROM produto
    GROUP BY Descricao_Produto;

elseif (var = 2) then
    SELECT Nome_Fornecedor, COUNT(*) as Qtde
    FROM fornecedor
    GROUP BY Nome_Fornecedor;

else select var;

end if;
set var = var + 1;
END WHILE;
END$$
DELIMITER ;

CALL WhileLoopProc();

```