	Ano Letivo: 20251	Turno: Noturno	Local: A definir
	Curso: ENGENHARIA DA COMPUTAÇÃO		Turma: 3ª Fase
Disciplina: BANCO DE DADOS II		Professor(a): JORGE LUIZ DA SILVA	

TRABALHO FINAL COM BASE EM METODOLOGIAS ATIVAS DE APRENDIZAGEM

Projeto de banco de dados para um sistema de Sistema de Manutenção de Veículos

Bruno Pagani Rampinelli - [@NaoSouOBruno](#)

Gabriel Reis Klein - [@Gabriel-dos-Reis-Klein](#)

Gabriel Tramontin Aguiar + [@GabrielTramontin](#)

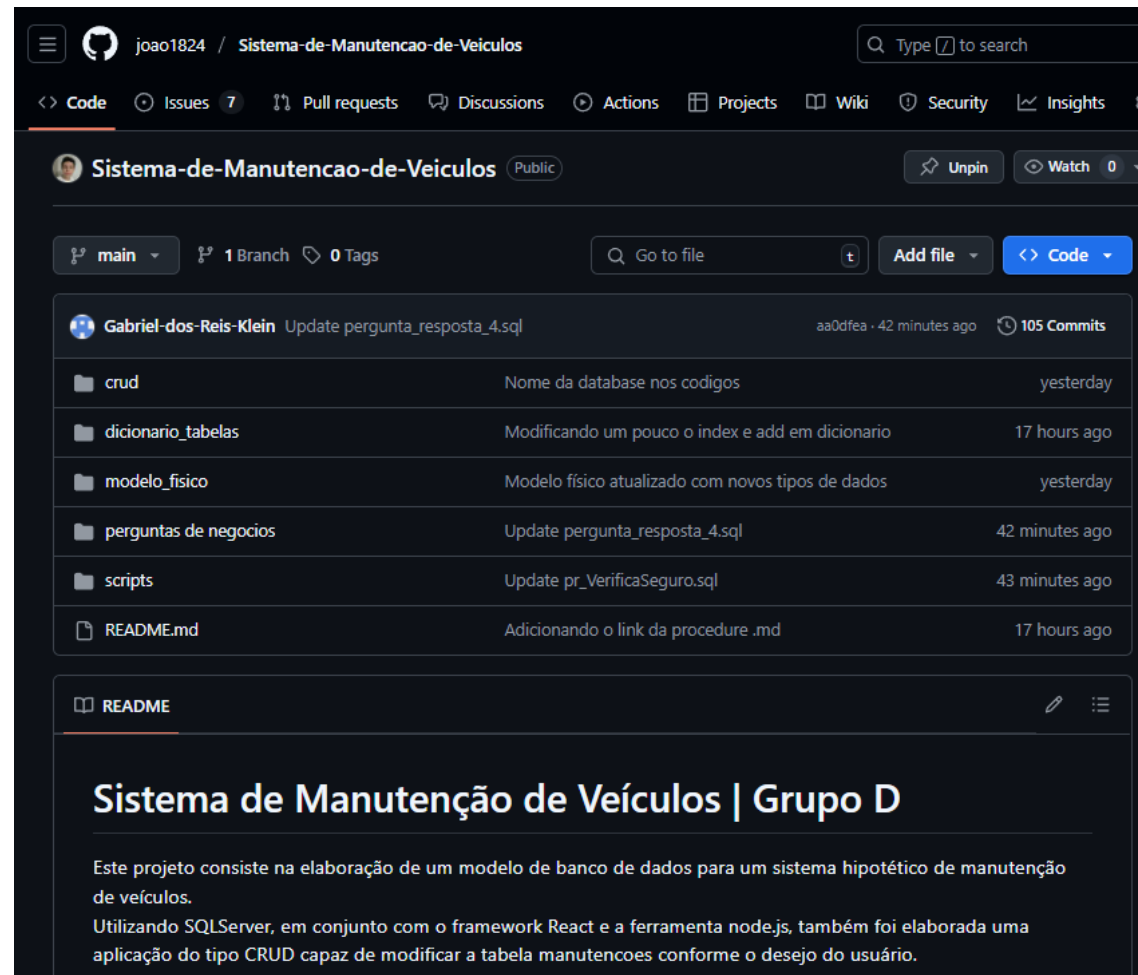
João Henrique Camilo Fogaça + [@joao1824](#)

Yuri Cardoso Maciel + [@DoppelsoldnerHRE](#)

Criciúma, 23/06/2025

URL do projeto no GitHub

<https://github.com/joao1824/Sistema-de-Manutencao-de-Veiculos>



The screenshot displays the GitHub interface for the repository 'Sistema-de-Manutencao-de-Veiculos' by user 'joao1824'. The repository is public and has 7 issues, 1 branch, and 0 tags. The main branch is 'main'. The file list shows the following files and their commit history:

File	Commit Message	Commit Time
crud	Nome da database nos codigos	yesterday
dicionario_tabelas	Modificando um pouco o index e add em dicionario	17 hours ago
modelo_fisico	Modelo físico atualizado com novos tipos de dados	yesterday
perguntas de negocios	Update pergunta_resposta_4.sql	42 minutes ago
scripts	Update pr_VerificaSeguro.sql	43 minutes ago
README.md	Adicionando o link da procedure .md	17 hours ago

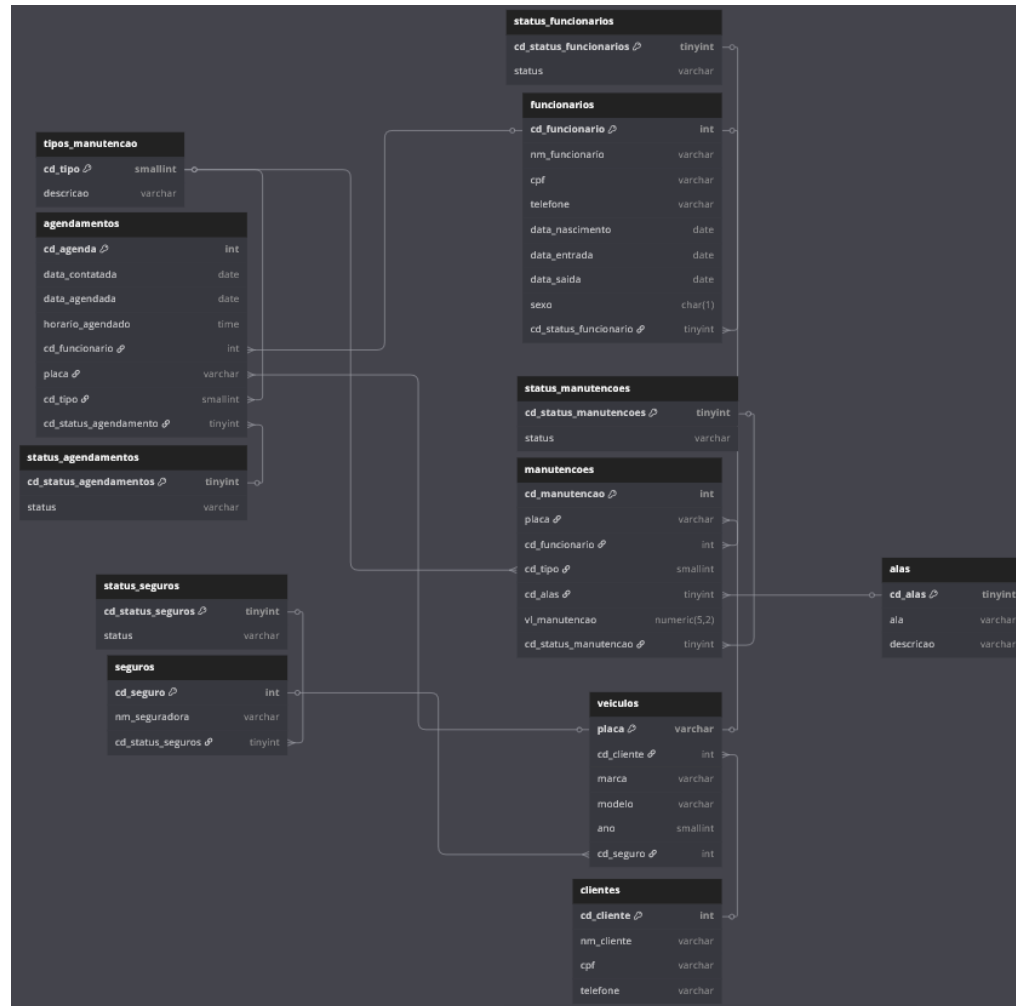
The README file is selected, showing the following content:

Sistema de Manutenção de Veículos | Grupo D

Este projeto consiste na elaboração de um modelo de banco de dados para um sistema hipotético de manutenção de veículos.

Utilizando SQLServer, em conjunto com o framework React e a ferramenta node.js, também foi elaborada uma aplicação do tipo CRUD capaz de modificar a tabela manutencoes conforme o desejo do usuário.

Modelo ER Físico



Dicionário de Dados

Tabela	clientes					
Descrição	Tabela responsável por armazenar os dados dos clientes cadastrados no sistema.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
cd_cliente	int	1 – 2.147.483.647	NOT NULL	X		Código de identificador do cliente
nm_cliente	varchar(50)	1 – 50	NOT NULL			Nome do cliente
cpf	varchar(14)	0 – 14	NULL			CPF do cliente
telefone	varchar(20)	1 – 20	NOT NULL			Telefone de contato do cliente
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_clientes_cd_cliente	X		X	cd_cliente		
idx_clientes_nm_cliente		X		nm_cliente		

Tabela	seguros					
Descrição	Tabela responsável por armazenar os dados dos seguros.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
cd_seguro	int	1 – 2.147.483.647	NOT NULL	X		Código de identificador do seguro
nm_seguradora	varchar(50)	1 – 50	NOT NULL			Nome da seguradora
cd_status_seguros	tinyint	1 – 255	NOT NULL		X	Estado atual do seguro
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_seguros_cd_seguro	X		X	cd_seguro		
idx_seguros_nm_seguradora		x		nm_seguradora		
idx_seguros_cd_status		x	X	cd_status_seguros		

Tabela	status_seguros					
Descrição	Tabela responsável por armazenar os possíveis status dos seguros.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
cd_status_seguros	tinyint	1 – 255	NOT NULL	X		Código do Status do Seguro
status	varchar(20)	1 – 20	NOT NULL			"Ativo", "Inativo"
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_status_seguros_cd_status_seguros	X		X	cd_status_seguros		
idx_status_seguros_status		X		status		

Tabela	funcionarios					
Descrição	Tabela responsável por armazenar os dados dos funcionários cadastrados no sistema.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
cd_funcionario	int	1 – 2.147.483.647	NOT NULL	X		Código de identificador do funcionário
nm_funcionario	varchar(200)	1 – 200	NOT NULL			Nome do funcionário
cpf	varchar(14)	1 – 14	NULL			CPF do funcionário
telefone	varchar(20)	1 – 20	NOT NULL			Telefone do funcionário
data_nascimento	date	0001-01-01 a 9999-12-31	NULL			Data de nascimento do funcionário
data_entrada	date	0001-01-01 a 9999-12-31	NOT NULL			Data de quando entrou na empresa
data_saida	date	0001-01-01 a 9999-12-31	NULL			Data de quando foi desativado da empresa
sexo	char(1)	1 – 1	NOT NULL			Só pode ser "F" (Feminino) ou "M" (Masculino)
cd_status_funcionario	tinyint	1 – 255	NOT NULL		X	Código do status do funcionario
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_funcionarios_cd_funcionario	X		X	cd_funcionario		
idx_funcionarios_cd_status		X		cd_status_funcionario		

idx_func_fn_porcentagem_mediana		X		cd_funcionario, data_entrada
idx_func_relatorio_top3		X		cd_funcionario, cd_status_funcionario, nm_funcionario
IX_funcionarios_sexo_status		X		cd_funcionario, nm_funcionario, data_saida, data_entrada
IX_funcionarios_data_entrada		X		cd_funcionario

Tabela	status_funcionarios					
Descrição	Status do funcionário: "Ativo", "Suspendo", "Demitido" ou "Férias".					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
cd_status_funcionarios	tinyint	1 – 255	NOT NULL	X		Código do status do funcionário
status	varchar(10)	1 – 10	NOT NULL			"Ativo", "Suspendo", "Demitido" ou "Férias"
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_status_funcionarios_cd_status_funcionarios	X		X	cd_status_funcionarios		
idx_status_funcionarios_status		X		status		

Tabela	agendamentos					
Descrição	Tabela responsável por armazenar os dados dos agendamentos.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
cd_agenda	int	1 – 2.147.483.647	NOT NULL	X		Código de identificador do agendamento
data_contatada	date	0001-01-01 a 9999-12-31	NOT NULL			Data que foi contatada para escolher a data do agendamento
data_agendada	date	0001-01-01 a 9999-12-31	NOT NULL			Data que seria realizado o serviço com base no que foi marcado
horario_agendado	time	00:00:00.0000000 a 23:59:59.9999999	NOT NULL			Horario que foi agendado
cd_funcionario	int	1 – 2.147.483.647	NOT NULL		X	Código de funcionario responsavel
placa	varchar(7)	1 – 7	NOT NULL		X	Placa do veículo que será submetido à manutenção.
cd_tipo	smallint	1 – 32.767	NOT NULL		X	Codigo do tipo de manutenção
cd_status_agendamento	tinyint	1 – 255	NOT NULL		X	Código do status do agendamento
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_agendamentos_cd_agenda	X		X	cd_agenda		

Tabela	status_agendamentos					
Descrição	Status do agendamento: "Ativo", "Cancelado", "Retificada" .					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
cd_status_agendamento	tinyint	1 – 255	NOT NULL	X		Código do status do agendamento
status	varchar(20)	1 – 20	NOT NULL			"Ativo", "Cancelado", "Retificada"
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_status_agendamentos_cd_status_agendamento	X		X	cd_status_agendamento		

Tabela	alas					
Descrição	Tabela responsável por armazenar os dados da alas cadastradas no sistema					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
cd_alas	tinyint	1 – 255	NOT NULL	X		Código de identificador da ala
ala	varchar(50)	1 – 50	NOT NULL			Nome da ala
descricao	varchar(200)	0 – 200	NULL			Descrição da ala
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_alas_cd_alas	X		X	cd_alas		

Tabela	manutencoes					
Descrição	Tabela responsável por armazenar os dados das manutenção cadastradas no sistema					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
cd_manutencao	int	1 – 2.147.483.647	NOT NULL	X		Código de identificador da manutenção
placa	varchar(7)	1 – 7	NOT NULL		X	Placa do veículo que será submetido à manutenção.
cd_funcionario	int	1 – 2.147.483.647	NOT NULL		X	Código de identificador do funcionário
cd_tipo	smallint	1 – 32.767	NOT NULL		X	Código do tipo de manutenção
cd_alas	tinyint	1 – 255	NOT NULL		X	Código da ala
vl_manutencao	numeric(5,2)	-99999.99 a 99999.99	NULL			Valor cobrado pela manutenção
cd_status_manutencao	tinyint	1 – 255	NOT NULL		X	Status da manutenção
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_manutencoes_cd_manutencao	X		X	cd_manutencao		
idx_manutencoes_placa		X		placa		
idx_manutencoes_cd_funcionario		X		cd_funcionario		
IX_manutencoes_funcionario_valor		X		vl_manutencao		
idx_manutencoes_vl_manutencao		X		vl_manutencao		
idx_manut_relatorio_top3		X		cd_funcionario , vl_manutencao		

Tabela	status_manutencoes					
Descrição	Tabela responsável por armazenar os possíveis status de manutenção.					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
cd_status_manutencoes	tinyint	1 – 255	NOT NULL	X		Código do Status de Manutenção
status	varchar(40)	1 – 40	NOT NULL			"Pronto", "Em Andamento", "Atrasado"...
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_status_manutencoes_cd_status_manutencoes	X		X	cd_status_manutencoes		

Tabela	tipos_manutencao					
Descrição	Tabela responsável por armazenar os dados dos tipos de manutenção cadastradas no sistema					
Atributos						
Nome da Coluna	Tipo do Dado	Valor min e max	Nulidade	PK	FK	Descrição
cd_tipo	smallint	1 – 32.767	NOT NULL	X		Código de identificador do tipo de manutenção
descricao	varchar(50)	1 – 50	NOT NULL			Descrição do tipo de manutenção
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_tipos_manutencao_cd_tipo	X		X	cd_tipo		
idx_tipos_manutencao_descricao		X		descricao		

Tabela	veiculos					
Descrição	Tabela responsável por armazenar os dados dos veículos cadastrados no sistema					
Atributos						
Nome da coluna	Tipo de dado	valor min e max	Nulidade	PK	FK	Descrição
placa	varchar(7)	1 – 7	NOT NULL	X		Número da Placa
cd_cliente	int	1 – 2.147.483.647	NOT NULL		X	Código do Cliente
marca	varchar(20)	1 – 20	NOT NULL			Marca do Automóvel
modelo	varchar(20)	1 – 20	NOT NULL			Modelo do Automóvel
ano	smallint	1 – 32.767	NOT NULL			Ano do Automóvel
cd_seguro	int	1 – 2.147.483.647	NOT NULL		X	Código do Seguro
Índice						
Nome do índice	Clustered	NonClustered	Unique	Colunas		
idx_veiculos_placa	X		X	placa		
idx_veiculos_cd_cliente		X		cd_cliente		
idx_veiculos_cd_seguro		X		cd_seguro		

Script dos comandos DDL para criação do Banco de dados (mínimo 8 tabelas)

```
-- Tabela clientes
create table clientes(
    cd_cliente int not null primary key,
    nm_cliente varchar(50) not null,
    cpf varchar(14) null,
    telefone varchar(20) not null
)
go

-- Tabela status_seguros
create table status_seguros(
    cd_status_seguros tinyint not null primary key,
    status varchar(20) not null check(status in ('Ativo', 'Inativo'))
)
go

-- Tabela seguros
create table seguros(
    cd_seguro int not null primary key,
    nm_seguradora varchar(50) not null,
    cd_status_seguros tinyint not null,
    constraint fk_status_seguros foreign key (cd_status_seguros)
    references status_seguros(cd_status_seguros)
)
go
```

-- Tabela veiculos

```
create table veiculos(  
    placa varchar(7) not null primary key,  
    cd_cliente int not null,  
    marca varchar(20) not null,  
    modelo varchar(20) not null,  
    ano smallint not null,  
    cd_seguro int not null,  
    constraint fk_cliente foreign key (cd_cliente)  
    references clientes(cd_cliente),  
    constraint fk_seguro foreign key (cd_seguro)  
    references seguros(cd_seguro)  
)  
go
```

-- Tabela status_manutencoes

```
create table status_manutencoes(  
    cd_status_manutencoes tinyint not null primary key,  
    status varchar(40) not null check(status in ('Pronto', 'Em Andamento', 'Atrasado', 'Em espera de peças', 'Em espera de  
pagamento', 'Esperando o Seguro', 'Carro não veio', 'Em segundo Plano', 'Emergência', 'Sem espaço para armazenar'))  
)  
go
```

-- Tabela alas

```
create table alas(  
    cd_alas tinyint not null primary key,  
    ala varchar(50) not null,  
    descricao varchar(200) null  
)  
go
```

```

-- Tabela status_agendamentos
create table status_agendamentos(
    cd_status_agendamento tinyint not null primary key,
    status varchar(20) not null check(status in ('Ativo', 'Cancelado', 'Retificada'))
)
go

-- Tabela status_funcionarios
create table status_funcionarios(
    cd_status_funcionarios tinyint not null primary key,
    status varchar(10) not null check(status in ('Ativo', 'Suspendido', 'Demitido', 'Férias'))
)
go

-- Tabela funcionarios
create table funcionarios(
    cd_funcionario int primary key not null,
    nm_funcionario varchar(200) not null,
    cpf varchar(14) null,
    telefone varchar(20) not null,
    data_nascimento date null,
    data_entrada date not null,
    data_saida date null,
    sexo char(1) not null check(sexo in ('F', 'M')),
    cd_status_funcionario tinyint not null,
    constraint fk_status_funcionario foreign key (cd_status_funcionario)
    references status_funcionarios(cd_status_funcionarios)
)
go

```

```
-- Tabela tipos_manutencao
create table tipos_manutencao(
    cd_tipo smallint not null primary key,
    descricao varchar(50) not null
)
go
```

```
-- Tabela agendamentos
create table agendamentos(
    cd_agenda int not null primary key,
    data_contatada date not null,
    data_agendada date not null,
    horario_agendado time not null,
    cd_funcionario int not null,
    placa varchar(7) not null,
    cd_tipo smallint not null,
    cd_status_agendamento tinyint not null,
    constraint fk_status_agendamento foreign key (cd_status_agendamento)
    references status_agendamentos(cd_status_agendamento),
    constraint fk_tipo_agendamento foreign key (cd_tipo)
    references tipos_manutencao(cd_tipo),
    constraint fk_placa_agendamento foreign key (placa)
    references veiculos(placa),
    constraint fk_funcionario_agendamento foreign key (cd_funcionario)
    references funcionarios(cd_funcionario)
)
go
```

```
-- Tabela manutencoes
create table manutencoes(
    cd_manutencao int identity(1,1) not null primary key,
    placa varchar(7) not null,
    cd_funcionario int not null,
    cd_tipo smallint not null,
    cd_alas tinyint not null,
    vl_manutencao numeric(7,2) null,
    cd_status_manutencoes tinyint not null,
    constraint fk_status_manutencao foreign key (cd_status_manutencoes)
    references status_manutencoes(cd_status_manutencoes),
    constraint fk_alas foreign key (cd_alas)
    references alas(cd_alas),
    constraint fk_tipo_manutencao foreign key (cd_tipo)
    references tipos_manutencao(cd_tipo),
    constraint fk_funcionario_manutencao foreign key (cd_funcionario)
    references funcionarios(cd_funcionario),
    constraint fk_placa_manutencao foreign key (placa)
    references veiculos(placa)
)
go
```


Script que popula as tabelas do Banco de dados (mínimo 10 itens principais)

```
-- Tabela status_agendamentos
insert into status_agendamentos (cd_status_agendamento, status)
values (1, 'Ativo'),
       (2, 'Cancelado'),
       (3, 'Retificada');

go
```

```
-- Tabela status_funcionarios
insert into status_funcionarios (cd_status_funcionarios, status)
values (1, 'Ativo'),
       (2, 'Suspendido'),
       (3, 'Demitido'),
       (4, 'Férias');

go
```

```
-- Tabela status_manutencoes
insert into status_manutencoes (cd_status_manutencoes, status)
values (1, 'Pronto'),
       (2, 'Em Andamento'),
       (3, 'Atrasado'),
       (4, 'Em espera de peças'),
       (5, 'Em espera de pagamento'),
       (6, 'Esperando o Seguro'),
       (7, 'Carro não veio'),
       (8, 'Em segundo Plano'),
       (9, 'Emergência'),
       (10, 'Sem espaço para armazenar');

go
```

```
-- Tabela status_seguros
insert into status_seguros (cd_status_seguros, status)
values (1, 'Ativo'),
       (2, 'Inativo');

go

/* Tabelas Gerais */
-- Tabela tipos_manutencao
insert into tipos_manutencao (cd_tipo, descricao)
values (1, 'Troca de óleo'),
       (2, 'Troca de pneu'),
       (3, 'Troca de bateria'),
       (4, 'Verificação de freios'),
       (5, 'Troca de pastilhas de freio'),
       (6, 'Troca de amortecedores'),
       (7, 'Troca de velas de ignição'),
       (8, 'Verificação do sistema de arrefecimento'),
       (9, 'Troca de embreagem'),
       (10, 'Verificação do sistema de escapamento'),
       (11, 'Revisão completa');

go
```

-- Tabela funcionarios

```
insert into funcionarios (cd_funcionario, nm_funcionario, cpf, telefone, data_nascimento, data_entrada, data_saida, sexo,
cd_status_funcionario)
```

```
values (1, 'Ana Souza', '123.456.789-00', '11987654321', '1990-05-10', '2020-01-15', NULL, 'F', 1),
      (2, 'Bruno Lima', '987.654.321-00', '21912345678', '1985-11-22', '2018-03-10', NULL, 'M', 1),
      (3, 'Carlos Mendes', '111.222.333-44', '31987654321', '1992-07-08', '2021-06-01', NULL, 'M', 2),
      (4, 'Daniela Castro', '222.333.444-55', '11999998888', '1995-09-15', '2022-04-20', NULL, 'F', 4),
      (5, 'Eduardo Silva', '333.444.555-66', '31912344321', '1980-01-30', '2015-09-12', '2023-11-30', 'M', 3),
      (6, 'Fernanda Dias', NULL, '21988887777', '1993-12-25', '2019-02-01', NULL, 'F', 1),
      (7, 'Gabriel Torres', '444.555.166-77', '21955554444', '1988-06-10', '2017-08-05', NULL, 'M', 4),
      (8, 'Helena Ribeiro', '555.626.777-88', '11944443333', '1991-03-18', '2016-11-21', '2022-12-15', 'F', 3),
      (9, 'Igor Almeida', '662.777.888-99', '31999990000', '1983-04-14', '2010-07-09', NULL, 'M', 1),
      (10, 'Juliana Martins', NULL, '11933332222', '1990-08-05', '2023-01-02', NULL, 'F', 2),
      (11, 'Pedro Martins', '123.222.444-23', '11463532222', '1994-08-05', '2023-01-02', NULL, 'M', 1),
      (12, 'João Gabriel', '412.244.543-12', '11146332222', '2000-08-05', '2023-03-02', NULL, 'M', 1),
      (13, 'Marcos Fernandez', NULL, '11933311462', '1999-05-23', '2020-12-23', NULL, 'M', 1),
      (14, 'Lara Camilo', '433.256.368-25', '11933132222', '1997-04-02', '2017-02-02', NULL, 'F', 1),
      (15, 'Bianca Da Silva', '245.333.351-57', '11933332222', '1996-08-05', '2025-01-02', NULL, 'F', 1),
      (16, 'Juliano Mendes', NULL, '11933331222', '1996-08-05', '2023-11-02', NULL, 'M', 1);
```

go

-- Tabela alas

```
insert into alas (cd_alas, ala, descricao)
```

```
values (1, 'Norte', 'Ala da Equipe 1'),
      (2, 'Sul', 'Ala de revisao'),
      (3, 'Leste', 'Ala da Equipe 2'),
      (4, 'Oeste', 'Ala da Equipe 3'),
      (5, 'Centro', 'Ala principal');
```

go

-- Tabela seguros

```
insert into seguros (cd_seguro, nm_seguradora, cd_status_seguros) values (1, 'Neo Seguradora', 1),
    (2, 'Seguro Auto', 2),
    (3, 'PROAUTO Santa Catarina', 2),
    (4, 'Porto Seguro', 2),
    (5, 'Seguro Auto Youse', 1),
    (6, 'Alianca Segura', 1),
    (7, 'Suhai Seguros', 1),
    (8, 'Azul Seguros', 2),
    (9, 'Ituran', 1),
    (10, 'HDI Seguros', 2),
    (11, 'Minuto Seguradora', 1),
    (12, 'Youse', 2),
    (13, 'Seguradora Catarinense', 2),
    (14, 'Pier Seguradora', 1),
    (15, 'Forte Proteção Veicular', 1),
    (16, 'Liderança Proteção Veicular', 1),
    (17, 'Seguradora Tokio Marine', 1),
    (18, 'Allianz Brasil', 1);
```

go

-- Tabela clientes

```
insert into clientes (cd_cliente, nm_cliente, cpf, telefone)
values (1, 'Melissa Ferreira Rocha', '81866913085', '4838565857'),
    (2, 'Victor Cardoso Melo', '52926093071', '1925341664'),
    (3, 'Kaua Barros Barbosa', '28081666052', '8121087022'),
    (4, 'Júlia Sousa Carvalho', '51192887077', '4827155874'),
    (5, 'Nicole Gomes Alves', '72345121017', '4829375332'),
    (6, 'Leonor Castro Barros', '53646858028', '4729228642'),
    (7, 'Rafael Rocha Melo', '75916462069', '4933671675'),
    (8, 'Raissa Pereira Cavalcanti', '52392248036', '5134278273'),
    (9, 'Vitoria Pereira Dias', '83681483086', '4439490267'),
    (10, 'José Pereira Martins', '59231115006', '4525104608');
```

go

-- Tabela veiculos

insert into veiculos (placa, cd_cliente, marca, modelo, ano, cd_seguro)

```
values ('HSU3792', 1, 'Jeep', 'Grand Cherokee', 2019, 1),
      ('IZC1490', 2, 'Citroen', 'C4 Tendance', 2023, 2),
      ('IDL5690', 3, 'Hyundai', 'Elantra', 2014, 3),
      ('EKW3904', 4, 'Chevrolet', 'Corsa Wind Piquet', 2019, 4),
      ('KSX3812', 5, 'Chevrolet', 'Omega CD', 2023, 5),
      ('GES2653', 6, 'Volvo', 'S80', 2013, 6),
      ('NGM2579', 7, 'BMW', 'i3', 2019, 7),
      ('HZO6425', 8, 'Fiat', 'UNO', 2012, 8),
      ('FSZ7576', 9, 'Nissan', 'XTerra', 2023, 9),
      ('ADH2381', 10, 'Honda', 'Accord', 2018, 10);
```

go

-- Tabela agendamentos

insert into agendamentos (cd_agenda, data_contatada, data_agendada, horario_agendado,
cd_funcionario, placa, cd_tipo, cd_status_agendamento)

```
values (1, '2025-06-01', '2025-06-05', '08:00:00', 1, 'HSU3792', 3, 1),
      (2, '2025-06-02', '2025-06-06', '09:30:00', 2, 'IZC1490', 7, 1),
      (3, '2025-06-03', '2025-06-07', '10:00:00', 13, 'IDL5690', 1, 2),
      (4, '2025-06-04', '2025-06-08', '08:00:00', 15, 'EKW3904', 11, 1),
      (5, '2025-06-04', '2025-06-08', '10:30:00', 2, 'KSX3812', 5, 3),
      (6, '2025-06-04', '2025-06-08', '15:00:00', 6, 'GES2653', 9, 1),
      (7, '2025-06-07', '2025-06-11', '15:45:00', 11, 'NGM2579', 2, 1),
      (8, '2025-06-08', '2025-06-12', '16:30:00', 16, 'HZO6425', 10, 2),
      (9, '2025-06-09', '2025-06-13', '08:45:00', 9, 'FSZ7576', 4, 1),
      (10, '2025-06-10', '2025-06-14', '10:30:00', 14, 'ADH2381', 8, 1),
      (11, '2025-07-01', '2025-07-14', '14:30:00', 1, 'ADH2381', 8, 1),
      (12, '2025-07-01', '2025-07-10', '10:30:00', 6, 'IZC1490', 8, 1),
      (13, '2025-07-10', '2025-07-12', '08:30:00', 15, 'NGM2579', 8, 2),
      (14, '2025-07-16', '2025-08-07', '17:00:00', 1, 'EKW3904', 8, 1),
      (15, '2025-07-12', '2025-07-11', '15:30:00', 12, 'ADH2381', 8, 2),
      (16, '2025-07-23', '2025-07-23', '07:30:00', 1, 'NGM2579', 8, 3),
      (17, '2025-07-25', '2025-08-10', '10:30:00', 9, 'FSZ7576', 8, 1);
```

go

-- Tabela manutencoes

```
insert into manutencoes (placa, cd_funcionario, cd_tipo, cd_alas, vl_manutencao, cd_status_manutencoes)
values ('HSU3792', '2', '11', '2', '250.00', '1'),
      ('HSU3792', '6', '3', '1', '350.00', '2'),
      ('EKW3904', '12', '11', '2', '250.00', '1'),
      ('NGM2579', '12', '5', '4', '450.00', '5'),
      ('EKW3904', '1', '9', '5', '500.00', '3'),
      ('IZC1490', '9', '11', '2', '250.00', '2'),
      ('IDL5690', '15', '11', '2', '250.00', '1'),
      ('KSX3812', '15', '8', '1', '300.00', '7'),
      ('GES2653', '9', '10', '4', '450.00', '6'),
      ('FSZ7576', '1', '11', '2', '250.00', '8'),
      ('ADH2381', '11', '11', '2', '250.00', '9'),
      ('HZ06425', '15', '11', '2', '250.00', '8'),
      ('ADH2381', '8', '3', '2', '450.00', '9'),
      ('ADH2381', '14', '6', '2', '450.00', '9'),
      ('IDL5690', '13', '2', '5', '375.00', '10');
```

go

Principais consultas mapeadas baseadas em regras de negócio (mínimo 6)

-- Pergunta 1

-- Quais os três clientes com o maior custo total de manutenções que foram realizadas nos últimos 12 meses,
-- por funcionários ativos, e em veículos com seguro ativo?
-- Na tabela resultado, para cada um desses três clientes, mostrar o nome completo, total gasto,
-- o número total de manutenções realizadas, e a média de valor dessas manutenções.
-- Além disso, atribuir a cada cliente um ranking de fidelidade (Ouro, Prata ou Bronze) baseando-se na soma
-- total de seus gastos com manutenções.

-- Resposta

-- function p/ ranking de fidelidade do cliente

```
drop function if exists dbo.fn_ranking_fidelidade
go
```

```
create function dbo.fn_ranking_fidelidade(@total_gasto numeric(10,2))
returns varchar(10)
as
begin
    declare @fidelidade varchar(10)

    if @total_gasto >= 1000
        set @fidelidade = 'Ouro'
    else if @total_gasto >= 500
        set @fidelidade = 'Prata'
    else
        set @fidelidade = 'Bronze'

    return @fidelidade
end
go
```

```

--== CONSULTA PRINCIPAL ==--
-- cte: filtra manutencoes
with cte_manutencoes as (
- cte
    select m.*, v.cd_cliente
    from manutencoes m
    join veiculos v on m.placa = v.placa
    join funcionarios f on m.cd_funcionario = f.cd_funcionario
    join seguros s on v.cd_seguro = s.cd_seguro
    where f.cd_status_funcionario in (
        select cd_status_funcionarios from status_funcionarios where status = 'Ativo') -- algumas subconsultas
    and s.cd_status_seguros in (
        select cd_status_seguros from status_seguros where status = 'Ativo')
    and m.vl_manutencao is not null
    and m.cd_manutencao in (
        select cd_manutencao from manutencoes
        where getdate() >= dateadd(year, -1, getdate()))
),
-- cte: consulta/calcula valores (qtde., soma, media)
valores_por_cliente as (
    select c.cd_cliente as 'Cód. Cliente',
           c.nm_cliente as 'Nome Cliente',
           count(*) as qtd_manutencoes,
           sum(m.vl_manutencao) as total_gasto,
           avg(m.vl_manutencao) as media_valor
    from cte_manutencoes m
    join clientes c on m.cd_cliente = c.cd_cliente
    group by c.cd_cliente, c.nm_cliente
),

-- cte: rankeando clientes
rank_clientes as (
    select t.*,
           rank() over (order by t.total_gasto desc) as ranking,
           dbo.fn_ranking_fidelidade(t.total_gasto) as fidelidade
    from valores_por_cliente t
)

select top 3 *
from rank_clientes
order by ranking;

```

--Pergunta 2

--Solicita-se a identificação dos três funcionários que mais geraram lucro líquido para a empresa, considerando como custo operacional um desconto de 20% sobre o valor de cada manutenção.

--Devem ser levadas em conta apenas as manutenções com valor igual ou superior a R\$ 200,00, realizadas por funcionários que estejam atualmente ativos na empresa.

--Para cada um desses três funcionários, devem ser apresentados o nome, a quantidade total de manutenções realizadas,

--o valor total das manutenções considerando o desconto de 20% (ou seja, o lucro líquido gerado), a média do lucro líquido por manutenção e o valor de uma comissão.

--Essa comissão deve ser calculada como uma porcentagem da média do lucro líquido, sendo essa porcentagem equivalente ao tempo de empresa do funcionário (em anos completos, se for menos que 1, considera 1 ano) multiplicado por 10.

--Por exemplo, um funcionário com 3 anos de empresa receberá 30% de comissão sobre a média do lucro líquido.

--Aplica um desconto de 20% sobre o valor informado.

```
create or alter function fn_20 (@valor numeric(7,2)) returns numeric(7,2) as
begin
    declare @descontado numeric(7,2);
    set @descontado = @valor * 0.8;
    return @descontado;
end
```

--Função para conseguir a media de uma valor já somado previamente e dividir pela quantidade

```
create or alter function fn_media (@valor numeric(7,2),@quantidade int) returns numeric(7,2) as
begin
    declare @media numeric(7,2);
    set @media = @valor / @quantidade;
    return @media;
end
```

-- Função usada para conseguir a % de comissão do funcionario com base em quanto tempo eles esta na empresa

```
create or alter function fn_porcentagem_media (@valor numeric(7,2),@cd_funcionario int) returns numeric(7,2) as
begin
    declare @resposta numeric(7,2);
    declare @tempo_entrada date;
    declare @tempo_total int;
    set @tempo_entrada = (select data_entrada from funcionarios where cd_funcionario = @cd_funcionario);
    set @tempo_total = floor(datediff(year,@tempo_entrada,getdate()));
    if @tempo_total <= 0
        begin
            set @tempo_total = 1
        end

    set @resposta = (@valor * (@tempo_total*10))/100;

    return @resposta

end
```

create or alter procedure pr_relatorio_top3_manutencoes as

```
begin
    with cte_vl_manutencao_cd as(
        select f.nm_funcionario, f.cd_funcionario ,m.vl_manutencao,dbo.fn_20(m.vl_manutencao) as sem_custos from manutencoes
        m
        inner join funcionarios f
        on m.cd_funcionario = f.cd_funcionario
        where m.vl_manutencao >= 200 and f.cd_status_funcionario = 1
    ),
    cte_soma_manutencao as (
        select nm_funcionario, cd_funcionario, SUM(sem_custos) as soma_manutencoes ,
        COUNT(sem_custos) as quantidade_de_manutencao,(DENSE_RANK() over(order by (sum(sem_custos)) desc)) as ranke
        from cte_vl_manutencao_cd
        group by cd_funcionario,nm_funcionario),
```

```

        cte_media_manutencao as (
            select nm_funcionario as
nome,quantidade_de_manutencao,soma_manutencoes,dbo.fn_media(soma_manutencoes,quantidade_de_manutencao) as media,
            dbo.fn_porcentagem_media(dbo.fn_media(soma_manutencoes,quantidade_de_manutencao),cd_funcionario) as comissao from
cte_soma_manutencao
            where ranke <= 3
        )

        select * from cte_media_manutencao

    end

```

```
exec pr_relatorio_top3_manutencoes
```

```

--Pergunta 3
--Ranquear as alas que fizeram mais manutenções para a empresa e quanto que
--geraram de renda em média com um desconto de 15% sobre o valor como custo de
--peças, mostrando: o nome dessa ala, a quantidade de manutenções que fizeram,
--a renda média (sem desconto) e o lucro líquido

```

```

create or alter procedure pr_AlaLucro as
BEGIN
    select alas.ala AS NomeAla,
COUNT(manutencoes.cd_alas) AS Quant,
    AVG(manutencoes.vl_manutencao) AS Media,
    AVG(manutencoes.vl_manutencao * 0.85) AS LucroLiquido
from
    manutencoes
Inner Join
    alas on manutencoes.cd_alas = alas.cd_alas
group by
    alas.ala
order by
    Quant DESC
END

```

```
exec pr_AlaLucro
```

--Pergunta 4
--Identificar quais carros estão a espera do seguros e verificar se esses seguros estão ativos ou não,
--caso não estiverem aumentar o valor da manutenção por 15% por demora de pagamento, mostrar no resultado:
--placa do carro, nome da seguradora, status da seguradora, valor da manutenção

```
create or alter procedure pr_VerificaSeguro as
BEGIN
    select
        manutencoes.placa,
        seguros.nm_seguradora AS Nome,
        CASE
            WHEN seguros.cd_status_seguros = 1 THEN 'Ativa'
            WHEN seguros.cd_status_seguros = 2 THEN 'Inativa'
            ELSE 'Desconhecido'
        END AS StatusSeguradora,
        CASE
            WHEN seguros.cd_status_seguros = 2 THEN CAST(manutencoes.vl_manutencao * 1.15 as numeric(5,2))
            ELSE manutencoes.vl_manutencao
        END AS ValorManutencao
    FROM
        manutencoes
    INNER JOIN
        veiculos ON manutencoes.placa = veiculos.placa
    INNER JOIN
        seguros ON veiculos.cd_seguro = seguros.cd_seguro
    WHERE
        manutencoes.cd_status_manutencoes = 6
end

exec pr_VerificaSeguro
```

--Pergunta 5
--Solicita-se listar todos os tipos de manutenção cadastrados na empresa, apresentando para cada tipo o código (cd_tipo) e a descrição (descricao). Os resultados devem ser ordenados alfabeticamente pela descrição.”

-- 1. Função para contar quantos tipos de manutenção existem

```
create or alter function fn_contar_tipos_manutencao()  
returns int as  
begin  
    declare @total int;  
    select @total = count(*) from tipos_manutencao;  
    return @total;  
end;  
GO
```

-- 2. Procedure para listar tipos de manutenção ordenados por descrição

```
create or alter procedure pr_listar_tipos_manutencao as  
begin  
    select cd_tipo, descricao  
    from tipos_manutencao  
    order by descricao;  
end;  
GO
```

-- 3. Executar a procedure

```
exec pr_listar_tipos_manutencao;
```

--Pergunta 6
--Solicita-se a identificação das funcionárias da empresa e a concessão de dias de folga e comissão de acordo com o tempo de vínculo empregatício.
--As funcionárias que possuírem dois anos completos ou mais de empresa terão direito a dois dias de folga adicionais.
--As funcionárias com menos de um ano de empresa receberão um dia de folga.
--Além disso, deverá ser calculada uma comissão proporcional ao tempo de casa:
--Para cada funcionária, a comissão será calculada como um percentual sobre a soma do valor total das manutenções realizadas por ela, sendo esse percentual equivalente ao número de anos completos de empresa.
--Por exemplo: uma funcionária com 2 anos de empresa receberá 2% sobre o valor total das manutenções que executou.
--Funcionárias com menos de 1 ano de empresa receberão 1%.

```
-- Função para calcular tempo de empresa em anos completos
CREATE OR ALTER FUNCTION fn_calcular_tempo_empresa (@data_entrada DATE)
RETURNS INT AS
BEGIN
    DECLARE @anos INT
    SET @anos = DATEDIFF(YEAR, @data_entrada, GETDATE())

    -- Ajuste se ainda não completou o aniversário este ano
    IF MONTH(@data_entrada) > MONTH(GETDATE()) OR
        (MONTH(@data_entrada) = MONTH(GETDATE()) AND DAY(@data_entrada) > DAY(GETDATE()))
    BEGIN
        SET @anos = @anos - 1
    END

    RETURN @anos
END
GO
```

```

-- Procedimento para calcular benefícios para mulheres (completo e testado)
CREATE OR ALTER PROCEDURE sp_beneficios_mulheres AS
BEGIN
    -- Tabela temporária para armazenar os resultados
    CREATE TABLE #Resultados (
        cd_funcionario INT,
        nm_funcionario VARCHAR(200),
        tempo_empresa_anos INT,
        dias_folga INT,
        total_manutencoes DECIMAL(10,2),
        comissao DECIMAL(10,2),
        valor_comissao DECIMAL(10,2)
    )

    -- Inserir dados das funcionárias com número correto de colunas
    INSERT INTO #Resultados
    (cd_funcionario, nm_funcionario, tempo_empresa_anos, dias_folga, total_manutencoes, comissao, valor_comissao)
    SELECT
        f.cd_funcionario,
        f.nm_funcionario,
        dbo.fn_calcular_tempo_empresa(f.data_entrada),
        CASE
            WHEN dbo.fn_calcular_tempo_empresa(f.data_entrada) < 1 THEN 1
            ELSE dbo.fn_calcular_tempo_empresa(f.data_entrada)
        END,
        0, -- total_manutencoes inicial
        0, -- comissao inicial
        0 -- valor_comissao inicial
    FROM
        funcionarios f
    WHERE
        f.sexo = 'F'
        AND f.cd_status_funcionario = 1
        AND f.data_saida IS NULL

    -- Atualizar totais de manutenções
    UPDATE r
    SET total_manutencoes = ISNULL((
        SELECT SUM(m.vl_manutencao)
        FROM manutencoes m
        WHERE m.cd_funcionario = r.cd_funcionario
    ), 0)
    FROM #Resultados r

```

```

-- Calcular comissões
UPDATE r
SET
    comissao = CASE
        WHEN tempo_empresa_anos < 1 THEN 0
        ELSE tempo_empresa_anos * 2
    END,
    valor_comissao = (total_manutencoes *
        CASE
            WHEN tempo_empresa_anos < 1 THEN 0
            ELSE tempo_empresa_anos * 2
        END) / 100
FROM #Resultados r

-- Resultado final
SELECT
    cd_funcionario AS 'Código',
    nm_funcionario AS 'Nome',
    tempo_empresa_anos AS 'Anos de Empresa',
    dias_folga AS 'Dias de Folga',
    FORMAT(total_manutencoes, 'C', 'pt-BR') AS 'Total Manutenções',
    CONCAT(comissao, '%') AS '% Comissão',
    FORMAT(valor_comissao, 'C', 'pt-BR') AS 'Valor Comissão'
FROM #Resultados
ORDER BY nm_funcionario

DROP TABLE #Resultados
END
GO

-- Executar o procedimento
EXEC sp_beneficios_mulheres

```