CARLOS MAGNO DO NASCIMENTO JUNIOR
IORRANA MARIA DO NASCIMENTO
LUIZ CARLOS BESSA DE LIMA
MATEUS AUGUSTO DA SILVEIRA PINTO
TAIANE RODRIGUES DE SOUSA

SISTEMA DE CORRIDAS MOTOTÁXI

LAVRAS - MG
2021

CARLOS MAGNO DO NASCIMENTO JUNIOR
IORRANA MARIA DO NASCIMENTO
LUIZ CARLOS BESSA DE LIMA
MATEUS AUGUSTO DA SILVEIRA PINTO
TAIANE RODRIGUES DE SOUSA

SISTEMA DE CORRIDAS MOTOTÁXI

Trabalho apresentado à disciplina GCC214 - Introdução a Sistemas de Banco de Dados, como parte das exigências apresentadas no plano de curso.

LAVRAS – MG
2021

Sumário

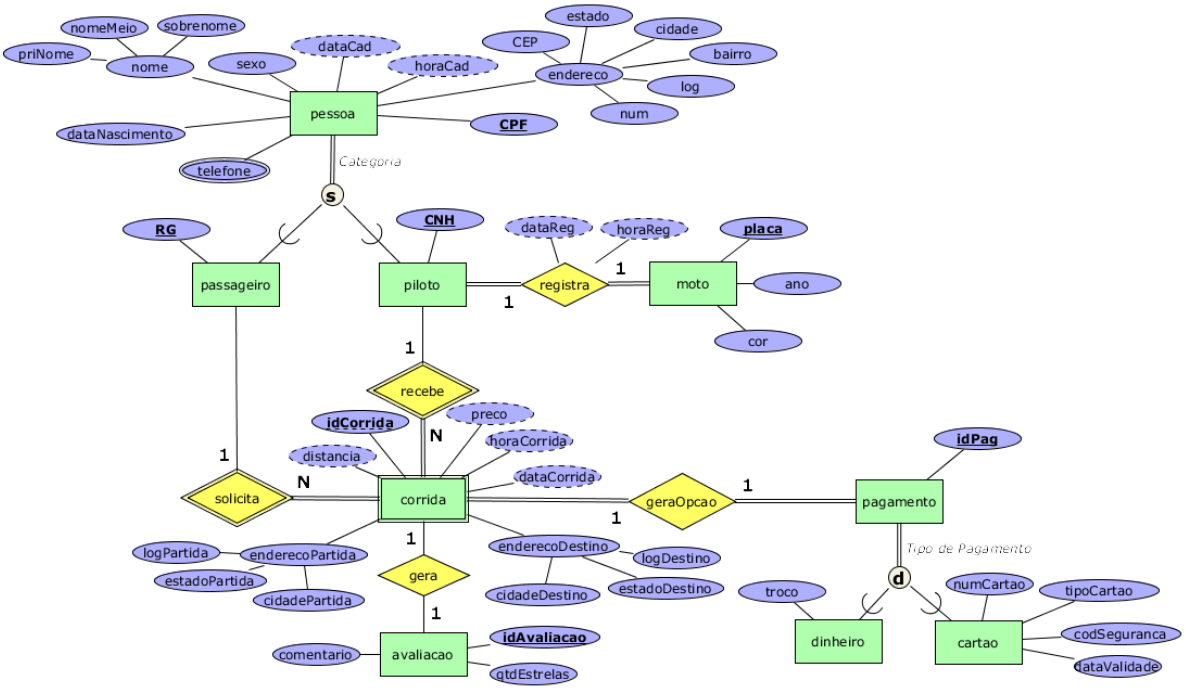
1. DESCRIÇÃO

Nosso modelo E.R consiste em um sistema de corridas mototáxi. O sistema possui um cadastro para passageiros e para pilotos, onde cada um deve cadastrar em comum as seguintes informações: CPF, nome completo, data de nascimeto, sexo (Masculino ou Feminino), endereço (CEP, estado, cidade, bairro, logradouro, número residencial e telefones, tendo em incomum o cadastro do RG respectivo somente ao passageiro, para o piloto somente o cadastro da CNH e registro da moto com os seguintes dados: placa, ano e cor. Ao realizar um cadastro ou registrar uma nova moto, sempre haverá um registro de data e hora gerado pelo próprio servidor.

Após cadastrado no sistema, é possível realizar a solicitação de uma corrida para um piloto que a recebe (cada corrida conterá um identificador único, gerado pelo sistema para consulta), sendo os seguintes dados da mesma visível para ambos: endereço de partida (estado, cidade, logradouro), endereço de destino (estado, cidade, logradouro), distância (calculada a partir de uma api, por exemplo, do google maps) e preço gerados de acordo a corrida tendo sempre registrada data e hora a partir do acontecimento da mesma.

Caso tenha ocorrido a corrida, será gerada a opção de pagamento podendo ser escolhida entre pagamento em dinheiro (opção de inserir valor para troco) ou pagamento no cartão tendo os dados do número do mesmo, código de segurança e data de validade (com opção para débito ou crédito), sendo gerado pelo sistema um protocolo de identificação do respectivo pagamento para possíveis consultas. E além disso pode ser efetuada a critério do piloto uma avaliação de comportamento e a critério do passageiro uma avaliação da qualidade do serviço prestado (cada avaliação conterá um identificador único, gerado pelo sistema para consulta), contendo quantidade de estrelas (1 - 5) e opção para comentário.

2. DIAGRAMA ER



3. DICIONÁRIO

3.1 ENTIDADES

Tipo Entidade	pessoa		
Descrição	Cadastra informações de uma pessoa		
Atributos			
Nome	**Descrição**	**Domínio**	**Permite nulo**
CPF	Cadastro de Pessoa Física	Texto(11) Formato: ddd.ddd.ddd-dd	N
priNome	Primeiro nome da pessoa	Texto(15)	N
nomeMeio	Nome do meio da pessoa	Texto(15)	N
sobrenome	Sobrenome da pessoa	Texto(40)	N
dataNascimento	Data de nascimento da pessoa	Texto(8) Formato: dd/dd/dddd	N
telefone	Número de telefone	Texto(11) Formato: (dd) ddddd-dddd	N
sexo	Sexo (masculino, feminino)	Texto(1) M - Masculino F - Feminino	N
CEP	Código de Endereçamento Postal	Texto(8) Formato: dd.ddd-ddd	N
estado	Sigla do estado	Texto(2)	N
cidade	Nome da cidade	Texto(30)	N
bairro	Nome do bairro	Texto(30)	N
log	Nome do logradouro (rua, avenida, alameda, etc)	Texto(40)	N
num	Número da residência	Inteiro positivo	N
dataCad	Data de cadastro da pessoa	DATE Formato: “aaaa-mm-dd”	N
horaCad	Hora de cadastro da pessoa	TIME Formato: “hh;mm:ss”	N

Tipo Entidade	passageiro		
Descrição	Pessoa que solicita uma corrida		
Atributos			
Nome	**Descrição**	**Domínio**	**Permite nulo**
RG	Carteira de identidade	Texto(9)	N

Tipo Entidade	piloto		
Descrição	Pessoa que recebe uma corrida		
Atributos			
Nome	**Descrição**	**Domínio**	**Permite nulo**
CNH	Carteira de motorista	Texto(11)	N

Tipo Entidade	moto		
Descrição	Veículo usado pelo piloto para transporte de passageiro.		
Atributos			
Nome	**Descrição**	**Domínio**	**Permite nulo**
placa	Placa da moto	Texto(7) Formato: AAA-1111	N
ano	Ano de fabricação/modelo da moto	Inteiro(4) Positivo	N
cor	Cor da moto	Texto(10)	N

Tipo Entidade	corrida		
Descrição	Guarda informações da corrida		
Atributos			
Nome	**Descrição**	**Domínio**	**Permite nulo**
distancia	Distância da corrida. Calcula a distância entre o endereço de partida e o endereço de destino	FLOAT Positivo	N
idCorrida	Identifica corrida	Inteiro positivo	N
preco	Preço da corrida	Real(5,2) Positivo	N
horaCorrida	Horário da corrida	TIME Formato: “hh;mm:ss”	N
dataCorrida	Data da corrida	DATE Formato: “aaaa-mm-dd”	N
logPartida	Nome do logradouro de partida	Texto(40)	N
estadoPartida	Sigla do estado de partida	Texto(2)	N
cidadePartida	Nome da cidade de partida	Texto(30)	N
logDestino	Nome do logradouro destino	Texto(40)	N
estadoDestino	Sigla do estado destino	Texto(2)	N
cidadeDestino	Nome da cidade destino	Texto(30)	N

Tipo Entidade	avaliacao		
Descrição	Passageiro avalia a qualidade do serviço prestado pelo piloto e piloto avalia comportamento do passageiro.		
Atributos			
Nome	**Descrição**	**Domínio**	**Permite nulo**
comentario	Comentário do passageiro e do piloto sobre a viagem	Texto(50)	S
idAvaliacao	Identificador do comentário	Inteiro Positivo	N
qtdEstrelas	Quantidade de estrelas para avaliação	Inteiro(5) Positivo	N

Tipo Entidade	pagamento		
Descrição	Pagamento da corrida		
Atributos			
Nome	**Descrição**	**Domínio**	**Permite nulo**
idPag	Número de protocolo	Texto(11)	N

Tipo Entidade	cartao		
Descrição	Cartão para pagamento		
Atributos			
Nome	**Descrição**	**Domínio**	**Permite nulo**
numCartao	Número do cartão.	Texto(16)	N
tipoCartao	Tipo do cartão (débito ou crédito).	Texto(2)	N
codSeguranca	Código de segurança do cartão.	Inteiro(3) positivo	N
dataValidade	Data de validade do cartão.	Texto(4) Formato: dd/dd	N

Tipo Entidade	dinheiro		
Descrição	Dinheiro para pagamento.		
Atributos			
Nome	**Descrição**	**Domínio**	**Permite nulo**
troco	Troco do pagamento.	Real(3, 2) positivo	S

3.2 RELACIONAMENTOS

Tipo Relacionamento	registra		
Descrição	Piloto registra sua moto; transporte que será utilizado para as corridas.		
Atributos			
Nome	**Descrição**	**Domínio**	**Permite nulo**
dataReg	Registra a data que a moto foi cadastrada.	DATE Formato: “aaaa-mm-dd”	N
horaReg	Registra a hora que a moto foi cadastrada.	TIME Formato: “hh;mm:ss”	N

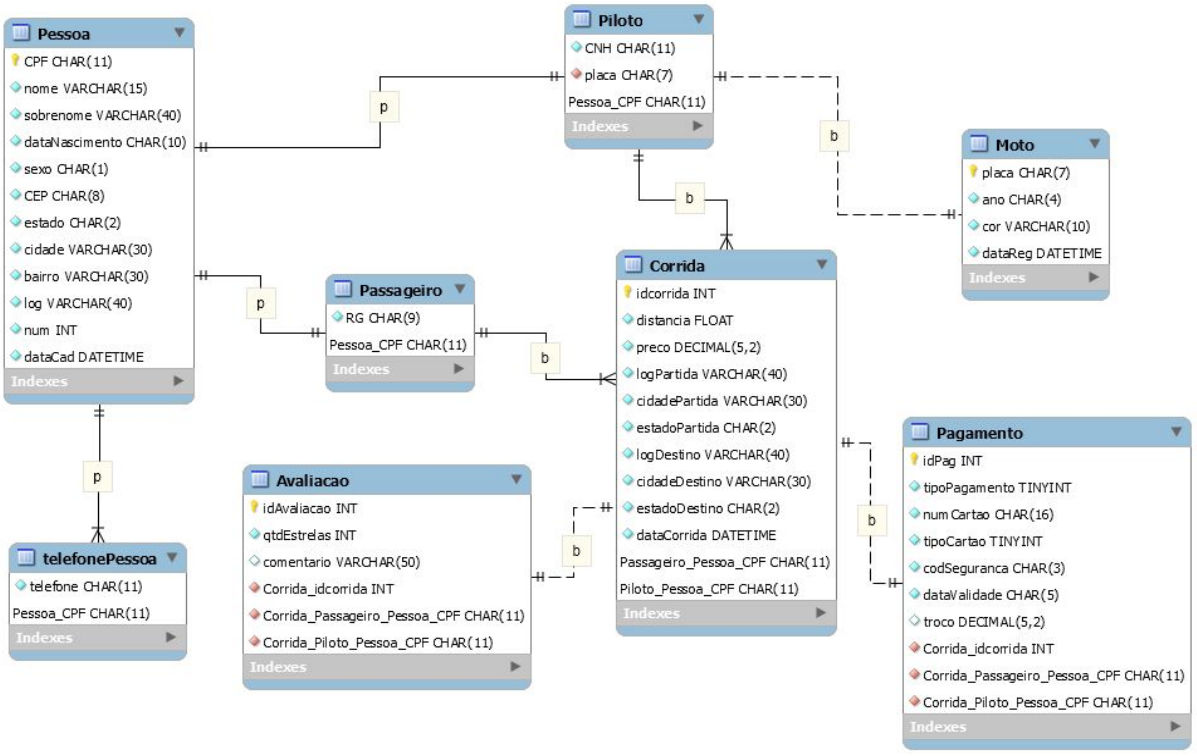
Tipo Relacionamento	solicita
Descrição	Passageiro que deseja se deslocar solicita uma corrida.

Tipo Relacionamento	recebe
Descrição	Piloto recebe uma corrida, solicitada pelo passageiro.

Tipo Relacionamento	gera
Descrição	A corrida gera uma avaliação.

Tipo Relacionamento	geraOpcao
Descrição	Após a corrida gera opção de pagamento.

4. DIAGRAMA RELACIONAL



5. DICIONÁRIO DE DADOS

Tabela	Pessoa
Descrição	Guarda as informações de uma pessoa
Atributos	
Nome	**Descrição**
CPF	Cadastro de Pessoa Física
nome	Primeiro nome da pessoa
sobrenome	Sobrenome da pessoa
dataNascimento	Data de nascimento da pessoa
sexo	Sexo (masculino, feminino)
CEP	Código de Endereçamento Postal que a pessoa se localiza
estado	Estado que a pessoa se localiza
cidade	Cidade que a pessoa se localiza
bairro	Bairro que a pessoa se localiza
log	Logradouro que a pessoa se localiza
num	Número da casa que a pessoa se localiza
dataCad	Data e hora do cadastro da pessoa

Tabela	telefonePessoa
Descrição	Guarda os números de telefones de uma pessoa
Atributos	
Nome	**Descrição**
telefone	Número de telefone da pessoa
Pessoa_CPF	Cadastro de Pessoa Física que identifica uma pessoa

Tabela	Passageiro
Descrição	Pessoa que solicita uma corrida
Atributos	
Nome	**Descrição**
RG	Número da carteira de identidade do passageiro
Pessoa_CPF	Cadastro de Pessoa Física que identifica uma pessoa

Tabela	Avaliacao
Descrição	Após a corrida temos o pagamento
Atributos	
Nome	**Descrição**
idAvaliacao	Número de identificação do avaliação
qtdEstrelas	Quantidade de estrelas
comentario	Comentário do passageiro e do piloto sobre a viagem
Corrida_idcorrida	Identificação da corrida
Corrida_Passageiro_Pessoa_CPF	Número do CPF do passageiro
Corrida_Piloto_Pessoa_CPF	Número do CPF do piloto

Tabela	Corrida
Descrição	Guarda informações da corrida
Atributos	
Nome	**Descrição**
idcorrida	Número identificação da corrida
distancia	Distância da corrida. Calcula a distância entre o endereço de partida e o endereço de destino
preco	Preço da corrida
logPartida	Nome do logradouro de partida
cidadePartida	Nome da cidade de partida
estadoPartida	Sigla do estado de partida
logDestino	Nome do logradouro destino
cidadeDestino	Nome da cidade destino
estadoDestino	Sigla do estado destino
dataCorrida	Data e hora da corrida
idPag	Número de protocolo de pagamento
Passageiro_Pessoa_CPF	Número do CPF do passageiro
Piloto_Pessoa_CPF	Número do CPF do piloto

Tabela	Piloto
Descrição	Pessoa que faz uma corrida
Atributos	
Nome	**Descrição**
CNH	Número da carteira de motorista do piloto
placa	Placa de seu veículo
Pessoa_CPF	Número do CPF do piloto

Tabela	Moto
Descrição	Veículo usado na corrida (pertence a um piloto)
Atributos	
Nome	**Descrição**
Placa	Placa do veículo
ano	ano de fabricação/modelo veículo
cor	Cor do veículo
dataReg	Data e hora do registro do veículo

Tabela	Pagamento
Descrição	Após a corrida gera a opção do pagamento
Atributos	
Nome	**Descrição**
idPag	Protocolo de identificação do pagamento
tipoPagamento	Identifica o tipo do pagamento, cartão ou dinheiro (1-cartão / 2-dinheiro)
numCartao	Número do cartão
tipoCartao	identifica o tipo do cartão, débito ou crédito (1-crédito / 2-débito / 3-pré pago)
codSeguranca	código de segurança do cartão
dataValidade	data de validade do cartão
troco	diferença entre o valor dado pelo usuário e o valor da corrida.
Corrida_idcorrida	Identificação da corrida
Corrida_Passageiro_Pessoa_CPF	Número do CPF do passageiro
Corrida_Piloto_Pessoa_CPF	Número do CPF do piloto

6. IMPLEMENTAÇÃO EM SQL

a) CRIAÇÃO DE TODAS AS TABELAS

Criação tabela Pessoa

```
CREATE TABLE IF NOT EXISTS `TrabalhoISBD`.`Pessoa` (
  `CPF` CHAR(11) NOT NULL,
  `nome` VARCHAR(15) NOT NULL,
  `sobrenome` VARCHAR(40) NOT NULL,
  `dataNascimento` CHAR(10) NOT NULL,
  `sexo` CHAR(1) NOT NULL,
  `CEP` CHAR(8) NOT NULL,
  `estado` CHAR(2) NOT NULL,
  `cidade` VARCHAR(30) NOT NULL,
  `bairro` VARCHAR(30) NOT NULL,
  `log` VARCHAR(40) NOT NULL,
  `num` INT NOT NULL,
  `dataCad` DATETIME NOT NULL,
  PRIMARY KEY (`CPF`))
ENGINE = InnoDB;
```

Criação tabela telefonePessoa

```
CREATE TABLE IF NOT EXISTS `TrabalhoISBD`.`telefonePessoa` (
  `telefone` CHAR(11) NOT NULL,
  `Pessoa_CPF` CHAR(11) NOT NULL,
  INDEX `fk_enderecoPessoa_Pessoa1_idx` (`Pessoa_CPF` ASC),
  UNIQUE INDEX `telefone_UNIQUE` (`telefone` ASC),
  PRIMARY KEY (`Pessoa_CPF`),
  CONSTRAINT `fk_enderecoPessoa_Pessoa1`
    FOREIGN KEY (`Pessoa_CPF`)
    REFERENCES `TrabalhoISBD`.`Pessoa` (`CPF`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

Criação tabela Passageiro

```
CREATE TABLE IF NOT EXISTS `TrabalhoISBD`.`Passageiro` (
  `RG` CHAR(9) NOT NULL,
  `Pessoa_CPF` CHAR(11) NOT NULL,
  PRIMARY KEY (`Pessoa_CPF`),
  UNIQUE INDEX `RG_UNIQUE` (`RG` ASC),
  CONSTRAINT `fk_Passageiro_Pessoa1`
    FOREIGN KEY (`Pessoa_CPF`)
    REFERENCES `TrabalhoISBD`.`Pessoa` (`CPF`)
```

```
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

Criação tabela Moto

```
CREATE TABLE IF NOT EXISTS `TrabalhoISBD`.`Moto` (
  `placa` CHAR(7) NOT NULL,
  `ano` CHAR(4) NOT NULL,
  `cor` VARCHAR(10) NOT NULL,
  `dataReg` DATETIME NOT NULL,
  PRIMARY KEY (`placa`))
ENGINE = InnoDB;
```

Criação tabela Piloto

```
CREATE TABLE IF NOT EXISTS `TrabalhoISBD`.`Piloto` (
  `CNH` CHAR(11) NOT NULL,
  `placa` CHAR(7) NOT NULL,
  `Pessoa_CPF` CHAR(11) NOT NULL,
  UNIQUE INDEX `CNH_UNIQUE` (`CNH` ASC),
  PRIMARY KEY (`Pessoa_CPF`),
  UNIQUE INDEX `placa_UNIQUE` (`placa` ASC),
  CONSTRAINT `fk_Piloto_Moto`
    FOREIGN KEY (`placa`)
    REFERENCES `TrabalhoISBD`.`Moto` (`placa`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT,
  CONSTRAINT `fk_Piloto_Pessoa1`
    FOREIGN KEY (`Pessoa_CPF`)
    REFERENCES `TrabalhoISBD`.`Pessoa` (`CPF`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

Criação tabela Corrida

```
CREATE TABLE IF NOT EXISTS `TrabalhoISBD`.`Corrida` (
  `idcorrida` INT NOT NULL AUTO_INCREMENT,
  `distancia` FLOAT NOT NULL,
  `preco` DECIMAL(5,2) NOT NULL,
  `logPartida` VARCHAR(40) NOT NULL,
  `cidadePartida` VARCHAR(30) NOT NULL,
  `estadoPartida` CHAR(2) NOT NULL,
  `logDestino` VARCHAR(40) NOT NULL,
  `cidadeDestino` VARCHAR(30) NOT NULL,
  `estadoDestino` CHAR(2) NOT NULL,
  `dataCorrida` DATETIME NOT NULL,
```

```
  `Passageiro_Pessoa_CPF` CHAR(11) NOT NULL,
  `Piloto_Pessoa_CPF` CHAR(11) NOT NULL,
  PRIMARY KEY (`idcorrida`, `Passageiro_Pessoa_CPF`, `Piloto_Pessoa_CPF`),
  INDEX `fk_Corrida_Passageiro1_idx` (`Passageiro_Pessoa_CPF` ASC),
  INDEX `fk_Corrida_Piloto1_idx` (`Piloto_Pessoa_CPF` ASC),
  CONSTRAINT `fk_Corrida_Passageiro1`
    FOREIGN KEY (`Passageiro_Pessoa_CPF`)
    REFERENCES `TrabalhoISBD`.`Passageiro` (`Pessoa_CPF`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT,
  CONSTRAINT `fk_Corrida_Piloto1`
    FOREIGN KEY (`Piloto_Pessoa_CPF`)
    REFERENCES `TrabalhoISBD`.`Piloto` (`Pessoa_CPF`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT)
ENGINE = InnoDB;
```

Criação tabela Pagamento

```
CREATE TABLE IF NOT EXISTS `TrabalhoISBD`.`Pagamento` (
  `idPag` INT NOT NULL AUTO_INCREMENT,
  `tipoPagamento` TINYINT NOT NULL,
  `numCartao` CHAR(16) NOT NULL,
  `tipoCartao` TINYINT NOT NULL,
  `codSeguranca` CHAR(3) NOT NULL,
  `dataValidade` CHAR(5) NOT NULL,
  `troco` DECIMAL(5,2) NULL,
  `Corrida_idcorrida` INT NOT NULL,
  `Corrida_Passageiro_Pessoa_CPF` CHAR(11) NOT NULL,
  `Corrida_Piloto_Pessoa_CPF` CHAR(11) NOT NULL,
  PRIMARY KEY (`idPag`),
  INDEX `fk_Pagamento_Corrida1_idx` (`Corrida_idcorrida` ASC,
`Corrida_Passageiro_Pessoa_CPF` ASC, `Corrida_Piloto_Pessoa_CPF` ASC),
  CONSTRAINT `fk_Pagamento_Corrida1`
    FOREIGN KEY (`Corrida_idcorrida` , `Corrida_Passageiro_Pessoa_CPF` ,
`Corrida_Piloto_Pessoa_CPF`)
    REFERENCES `TrabalhoISBD`.`Corrida` (`idcorrida` , `Passageiro_Pessoa_CPF` ,
`Piloto_Pessoa_CPF`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Criação tabela Avaliação

```
CREATE TABLE IF NOT EXISTS `TrabalhoISBD`.`Avaliacao` (
  `idAvaliacao` INT NOT NULL AUTO_INCREMENT,
  `qtdEstrelas` INT NOT NULL,
  `comentario` VARCHAR(50) NULL,
```

```
  `Corrida_idcorrida` INT NOT NULL,
  `Corrida_Passageiro_Pessoa_CPF` CHAR(11) NOT NULL,
  `Corrida_Piloto_Pessoa_CPF` CHAR(11) NOT NULL,
  PRIMARY KEY (`idAvaliacao`),
  INDEX `fk_Avaliacao_Corrida1_idx` (`Corrida_idcorrida` ASC,
`Corrida_Passageiro_Pessoa_CPF` ASC, `Corrida_Piloto_Pessoa_CPF` ASC),
  CONSTRAINT `fk_Avaliacao_Corrida1`
    FOREIGN KEY (`Corrida_idcorrida` , `Corrida_Passageiro_Pessoa_CPF` ,
`Corrida_Piloto_Pessoa_CPF`)
    REFERENCES `TrabalhoISBD`.`Corrida` (`idcorrida` , `Passageiro_Pessoa_CPF` ,
`Piloto_Pessoa_CPF`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

b) EXEMPLOS USANDO ALTER TABLE E DROP TABLE

Alterando tabela Pessoa incluindo a coluna idade

```
ALTER TABLE Pessoa ADD COLUMN idade CHAR(2) NOT NULL;
```

Alterando tabela Pessoa excluindo a coluna idade

```
ALTER TABLE Pessoa DROP COLUMN idade;
```

Alterando tabela Pessoa renomeando a coluna nome para primeiroNome

```
ALTER TABLE Pessoa RENAME COLUMN nome TO primeiroNome;
```

Criando tabela satisfacao do usuário

```
CREATE TABLE SatisfacaoUsuario (
    idSatisfacao INT NOT NULL AUTO_INCREMENT,
    avaliacaoApp VARCHAR(45) NOT NULL,
    PRIMARY KEY (idSatisfacao)
);
```

Excluindo tabela satisfacao do usuário

```
DROP TABLE SatisfacaoUsuario;
```

Exemplo de exclusão das tabelas

```
DROP TABLE  Pessoa;
DROP TABLE  telefonePessoa;
DROP TABLE Passageiro;
DROP TABLE Moto;
DROP TABLE Piloto;
DROP TABLE Corrida;
DROP TABLE Avaliacao;
DROP TABLE Pagamento;
```

c) INSERÇÕES EM TODAS AS TABELAS

Inserindo na tabela Pessoa

```
INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('12345678900', 'Denilson', 'Alves Pereira', '15/08/1972', 'M', '37200000', 'MG',
'Lavras', 'UFLA', 'Aquenta Sol', 209, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('21435465769', 'Joaquim', 'Quinteiro Uchôa', '12/07/1962', 'M', '38056677', 'MG',
'Uberaba', 'Alfredo Freire II', 'Rua Lulo Abrão Felício', 483, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('13243546578', 'Juliana', 'Galvani Gregh', '14/08/1988', 'F', '37205970', 'MG', 'Ijaci',
'Centro', 'Praça Prefeito Elias A. Filho 145', 221, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('30779797566', 'Antônio Maria', 'Pereira de Resende', '23/09/1947', 'M', '36046640',
'MG', 'Juiz de Fora', 'Quintas das Avenidas', 'Rua Cortes Vilela', 622, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('71145175643', 'Marluce', 'Rodrigues Pereira', '15/03/1988', 'F', '36081370', 'MG',
'Juiz de Fora', 'Industrial', 'Rua João Gualberto', 256, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('94289028454', 'Fabiana', 'Isabelle Barros', '05/08/1996', 'F', '79321554', 'MS',
'Corumbá', 'Vila Guarani', 'Alameda Espanha', 312, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('15440525025', 'Breno Eduardo', 'Otávio Almada', '20/05/1979', 'M', '37200970',
'MG', 'Lavras', 'Centro', 'Rua Raul Soares 159', 702, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('60098656465', 'Aline', 'Teixeira Alves', '01/09/1947', 'F', '96095270', 'MG', 'Lavras',
'Ouro Preto', 'Rua Professor Alberto Carvalho 750', 703, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('69636891990', 'Thiago', 'Barros Vieira', '06/12/1997', 'M', '36370970', 'MG',
'Nazareno', 'Centro', 'Avenida Padre Francisco Andrade 241', 886, NOW());
```

```
INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('49813853905', 'Osvaldo', 'Calebe da Paz', '17/04/1963', 'M', '11380225', 'SP', 'São
Vicente', 'Vila São Jorge', 'Praça da Divisa', 566, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('64942285348', 'Mário Thomas', 'Severino Monteiro', '26/05/1989', 'M', '37901300',
'MG', 'Passos', 'Jardim Vila Rica', 'Rodovia MG-050', 546, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('49914061885', 'Nicolas', 'Severino Assunção', '18/04/1994', 'M', '37200000', 'MG',
'Lavras', 'UFLA', 'Aquenta Sol', 342, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('62663964670', 'Sophia', 'Carossela Assunção', '26/06/1960', 'F', '37260970', 'MG',
'Perdões', 'Centro', 'Avenida Régis Bitencourt 342', 450, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('74226338910', 'Jorge', 'Henrique Novaes', '26/04/1971', 'M', '37200000', 'MG',
'Lavras', 'UFLA', 'Aquenta Sol', 125, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('80415810809', 'Fabiana', 'Teixeira Junqueira', '26/12/1971', 'F', '37260972', 'MG',
'Perdões', 'Centro', 'Rua Romão Fagundes, s/n', 602, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('28493614106', 'Antônio', 'Rosa Monteiro', '03/12/1966', 'M', '13611764', 'SP',
'Leme', 'Conjunto Francisco Coelho', 'Rua Henrique Coelho', 223, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('00613146336', 'Breno', 'Benício Pinto', '06/09/1970', 'M', '14060664', 'SP', 'Ribeirão
Preto', 'Jardim Javari', 'Rua Geraldo Magela', 912, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('02855606896', 'Márcio', 'Falcon da Luz', '17/05/1970', 'M', '13611764', 'SP', 'Leme',
'Conjunto Francisco Coelho', 'Rua Henrique Coelho', 139, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
```

```
VALUES('53258466637', 'Fabiana', 'Aparecida Assunção', '07/06/1988', 'F', '37880970', 'MG',
'Cabo Verde', 'Centro', 'Praça Capitão Luis Romão Siqueira 380', 494, NOW());

INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado,
cidade, bairro, log, num, dataCad)
VALUES('04797247681', 'Augusto', 'Oliveira Mendes', '03/04/1948', 'M', '30642587', 'MG',
'Belo Horizonte', 'Miramar (Barreiro)', 'Rua Esperança', 181, NOW());
```

Inserindo na tabela Pessoa e telefonePessoa

```
INSERT INTO Passageiro(Pessoa_CPF, RG) VALUES('12345678900','413062545');
INSERT INTO telefonepessoa(Pessoa_CPF, telefone)
VALUES('12345678900','35995080238');

INSERT INTO Passageiro(Pessoa_CPF, RG) VALUES('13243546578','395579557');
INSERT INTO telefonepessoa(Pessoa_CPF, telefone)
VALUES('13243546578','35999415361');

INSERT INTO Passageiro(Pessoa_CPF, RG) VALUES('71145175643','237312657');
INSERT INTO telefonepessoa(Pessoa_CPF, telefone)
VALUES('71145175643','37993765966');

INSERT INTO Passageiro(Pessoa_CPF, RG) VALUES('15440525025','185189544');
INSERT INTO telefonepessoa(Pessoa_CPF, telefone)
VALUES('15440525025','38989719941');

INSERT INTO Passageiro(Pessoa_CPF, RG) VALUES('69636891990','336565537');
INSERT INTO telefonepessoa(Pessoa_CPF, telefone)
VALUES('69636891990','18998422828');

INSERT INTO Passageiro(Pessoa_CPF, RG) VALUES('64942285348','425744942');
INSERT INTO telefonepessoa(Pessoa_CPF, telefone)
VALUES('64942285348','35986472850');

INSERT INTO Passageiro(Pessoa_CPF, RG) VALUES('62663964670','411327574');
INSERT INTO telefonepessoa(Pessoa_CPF, telefone)
VALUES('62663964670','35982330971');

INSERT INTO Passageiro(Pessoa_CPF, RG) VALUES('80415810809','493314969');
INSERT INTO telefonepessoa(Pessoa_CPF, telefone)
VALUES('80415810809','19992680584');

INSERT INTO Passageiro(Pessoa_CPF, RG) VALUES('00613146336','185662079');
INSERT INTO telefonepessoa(Pessoa_CPF, telefone)
VALUES('00613146336','21982086270');

INSERT INTO Passageiro(Pessoa_CPF, RG) VALUES('53258466637','127467531');
INSERT INTO telefonepessoa(Pessoa_CPF, telefone)
VALUES('53258466637','11999401667');
```

Inserindo na tabela Moto e Piloto

```
INSERT INTO moto(placa,ano,cor,dataReg) values('MXB9519', '2012', 'Vermelha', now());
INSERT INTO Piloto(Pessoa_CPF, CNH, placa) VALUES('21435465769','44839904554',
'MXB9519');

INSERT INTO moto(placa,ano,cor,dataReg) values('JVE9775', '2018', 'Prata', now());
INSERT INTO Piloto(Pessoa_CPF, CNH, placa) VALUES('30779797566','45383534000',
'JVE9775');

INSERT INTO moto(placa,ano,cor,dataReg) values('NFB9054', '2019', 'Prata', now());
INSERT INTO Piloto(Pessoa_CPF, CNH, placa) VALUES('94289028454','01030728878',
'NFB9054');

INSERT INTO moto(placa,ano,cor,dataReg) values('LVZ2012', '2013', 'Vermelha', now());
INSERT INTO Piloto(Pessoa_CPF, CNH, placa) VALUES('60098656465','64850114220',
'LVZ2012');

INSERT INTO moto(placa,ano,cor,dataReg) values('LNJ2326', '2020', 'Preta', now());
INSERT INTO Piloto(Pessoa_CPF, CNH, placa) VALUES('49813853905','66264662420',
'LNJ2326');

INSERT INTO moto(placa,ano,cor,dataReg) values('NAV6805', '2019', 'Prata', now());
INSERT INTO Piloto(Pessoa_CPF, CNH, placa) VALUES('49914061885','39574375084',
'NAV6805');

INSERT INTO moto(placa,ano,cor,dataReg) values('JLY6793', '2019', 'Prata', now());
INSERT INTO Piloto(Pessoa_CPF, CNH, placa) VALUES('74226338910','61684074143',
'JLY6793');

INSERT INTO moto(placa,ano,cor,dataReg) values('MRZ7354', '2021', 'Preta', now());
INSERT INTO Piloto(Pessoa_CPF, CNH, placa) VALUES('28493614106','32274964710',
'MRZ7354');

INSERT INTO moto(placa,ano,cor,dataReg) values('MTE4423', '2013', 'Preta', now());
INSERT INTO Piloto(Pessoa_CPF, CNH, placa) VALUES('02855606896','94878174648',
'MTE4423');

INSERT INTO moto(placa,ano,cor,dataReg) values('IAC5850', '2019', 'Branca', now());
INSERT INTO Piloto(Pessoa_CPF, CNH, placa) VALUES('04797247681','72938639649',
'IAC5850');
```

Inserindo na tabela Corrida, Pagamento e Avaliação

```
INSERT INTO Corrida(idcorrida, distancia, preco, logPartida, cidadePartida, estadoPartida,
cidadeDestino, logDestino,
estadoDestino, dataCorrida, Passageiro_Pessoa_CPF, Piloto_Pessoa_CPF)
values(1, 234, 120.00, 'Aquenta Sol', 'Lavras', 'MG', 'Perdões','Santa luz','MG',
now(),'12345678900', '74226338910');
```

```
INSERT INTO
Pagamento(idPag,tipoPagamento,numCartao,tipoCartao,codSeguranca,dataValidade,troco,
Corrida_idcorrida,Corrida_Passageiro_Pessoa_CPF,Corrida_Piloto_Pessoa_CPF)
values(1, '1', '1234567890123456','1','123','11/23', null, 1, '12345678900', '74226338910');

INSERT INTO
Avaliacao(idAvaliacao,qtdEstrelas,comentario,Corrida_idcorrida,Corrida_Passageiro_Pessoa
_CPF,Corrida_Piloto_Pessoa_CPF)
values(1, 5, 'Serviço ótimo, recomendo!', 1, '12345678900', '74226338910');
--
-----------------------------------------------------------------------------------------------------------------------
INSERT INTO Corrida(idcorrida, distancia, preco, logPartida, cidadePartida, estadoPartida,
cidadeDestino, logDestino,
estadoDestino, dataCorrida, Passageiro_Pessoa_CPF, Piloto_Pessoa_CPF)
values(2, 28.5, 15.31, 'Rua Esperança', 'Belo Horizonte', 'MG', 'Lavras','Rua Raul Soares
159','MG', now(),'15440525025', '04797247681');

INSERT INTO
Pagamento(idPag,tipoPagamento,numCartao,tipoCartao,codSeguranca,dataValidade,troco,
Corrida_idcorrida,Corrida_Passageiro_Pessoa_CPF,Corrida_Piloto_Pessoa_CPF)
values(2, '1', '4716770493297423','2','118','10/22', null, 2, '15440525025', '04797247681');

INSERT INTO
Avaliacao(idAvaliacao,qtdEstrelas,comentario,Corrida_idcorrida,Corrida_Passageiro_Pessoa
_CPF,Corrida_Piloto_Pessoa_CPF)
values(2, 0, 'Péssimo serviço, o piloto chegou atrasado!', 2, '15440525025', '04797247681');
--
-----------------------------------------------------------------------------------------------------------------------
INSERT INTO Corrida(idcorrida, distancia, preco, logPartida, cidadePartida, estadoPartida,
cidadeDestino, logDestino,
estadoDestino, dataCorrida, Passageiro_Pessoa_CPF, Piloto_Pessoa_CPF)
values(3, 300, 150.28, 'Rua João Gualberto', 'Juiz de Fora', 'MG', 'Lavras','Rua Professor
Alberto Carvalho','MG', now(),'71145175643', '30779797566');

INSERT INTO
Pagamento(idPag,tipoPagamento,numCartao,tipoCartao,codSeguranca,dataValidade,troco,
Corrida_idcorrida,Corrida_Passageiro_Pessoa_CPF,Corrida_Piloto_Pessoa_CPF)
values(3, '1', '5191401132511781','1','260','05/24', null, 3, '71145175643', '30779797566');

INSERT INTO
Avaliacao(idAvaliacao,qtdEstrelas,comentario,Corrida_idcorrida,Corrida_Passageiro_Pessoa
_CPF,Corrida_Piloto_Pessoa_CPF)
values(3, 4, null, 3, '71145175643', '30779797566');
--
-----------------------------------------------------------------------------------------------------------------------
INSERT INTO Corrida(idcorrida, distancia, preco, logPartida, cidadePartida, estadoPartida,
cidadeDestino, logDestino,
```

```
estadoDestino, dataCorrida, Passageiro_Pessoa_CPF, Piloto_Pessoa_CPF)
values(4, 54, 21.95, 'Praça da Divisa', 'São Vicente', 'SP', 'São Paulo','Rua Geraldo
Magela','SP', now(),'00613146336', '49813853905');

INSERT INTO
Pagamento(idPag,tipoPagamento,numCartao,tipoCartao,codSeguranca,dataValidade,troco,
Corrida_idcorrida,Corrida_Passageiro_Pessoa_CPF,Corrida_Piloto_Pessoa_CPF)
values(4, '1', '30284551386695','3','318','12/29', null, 4, '00613146336', '49813853905');

INSERT INTO
Avaliacao(idAvaliacao,qtdEstrelas,comentario,Corrida_idcorrida,Corrida_Passageiro_Pessoa
_CPF,Corrida_Piloto_Pessoa_CPF)
values(4, 4, 'Atrasou um pouco, mas o serviço é bom.', 4, '00613146336', '49813853905');
--
-----------------------------------------------------------------------------------------------------------------------
INSERT INTO Corrida(idcorrida, distancia, preco, logPartida, cidadePartida, estadoPartida,
cidadeDestino, logDestino,
estadoDestino, dataCorrida, Passageiro_Pessoa_CPF, Piloto_Pessoa_CPF)
values(5, 1300, 250.56, 'Rua São Miguel', 'São Paulo', 'SP', 'Ijaci','Centro','MG',
now(),'13243546578', '49813853905');

INSERT INTO
Pagamento(idPag,tipoPagamento,numCartao,tipoCartao,codSeguranca,dataValidade,troco,
Corrida_idcorrida,Corrida_Passageiro_Pessoa_CPF,Corrida_Piloto_Pessoa_CPF)
values(5, '1', '4430155695964829','2','460','01/22', null, 5, '13243546578', '49813853905');

INSERT INTO
Avaliacao(idAvaliacao,qtdEstrelas,comentario,Corrida_idcorrida,Corrida_Passageiro_Pessoa
_CPF,Corrida_Piloto_Pessoa_CPF)
values(5, 2, null, 5, '13243546578', '49813853905');
--
-----------------------------------------------------------------------------------------------------------------------
INSERT INTO Corrida(idcorrida, distancia, preco, logPartida, cidadePartida, estadoPartida,
cidadeDestino, logDestino,
estadoDestino, dataCorrida, Passageiro_Pessoa_CPF, Piloto_Pessoa_CPF)
values(6, 37.4, 24.78, 'Rua Nossa Senhora Aparecida', 'Piracicaba', 'SP', 'Rua Helio
Mendoça','João Pinheiro','MG', now(),'64942285348', '94289028454');

INSERT INTO
Pagamento(idPag,tipoPagamento,numCartao,tipoCartao,codSeguranca,dataValidade,troco,
Corrida_idcorrida,Corrida_Passageiro_Pessoa_CPF,Corrida_Piloto_Pessoa_CPF)
values(6, '1', '30023154819060','3','927','03/22', null, 6, '64942285348', '94289028454');

INSERT INTO
Avaliacao(idAvaliacao,qtdEstrelas,comentario,Corrida_idcorrida,Corrida_Passageiro_Pessoa
_CPF,Corrida_Piloto_Pessoa_CPF)
values(6, 1, 'Péssima qualidade', 6, '64942285348', '94289028454');
```

```
--
-----------------------------------------------------------------------------------------------------------------------
INSERT INTO Corrida(idcorrida, distancia, preco, logPartida, cidadePartida, estadoPartida,
cidadeDestino, logDestino,
estadoDestino, dataCorrida, Passageiro_Pessoa_CPF, Piloto_Pessoa_CPF)
values(7, 350, 400.53, 'Rua Rei Pelé', 'Santos', 'SP', 'Rua das flores','Paracatu','MG',
now(),'12345678900', '28493614106');

INSERT INTO
Pagamento(idPag,tipoPagamento,numCartao,tipoCartao,codSeguranca,dataValidade,troco,
Corrida_idcorrida,Corrida_Passageiro_Pessoa_CPF,Corrida_Piloto_Pessoa_CPF)
values(7, '1', '5527140333005913','1','635','02/23', null, 7, '12345678900', '28493614106');

INSERT INTO
Avaliacao(idAvaliacao,qtdEstrelas,comentario,Corrida_idcorrida,Corrida_Passageiro_Pessoa
_CPF,Corrida_Piloto_Pessoa_CPF)
values(7, 5, 'Serviço demorado, porém preço acessível', 7, '12345678900', '28493614106');
--
-----------------------------------------------------------------------------------------------------------------------
INSERT INTO Corrida(idcorrida, distancia, preco, logPartida, cidadePartida, estadoPartida,
cidadeDestino, logDestino,
estadoDestino, dataCorrida, Passageiro_Pessoa_CPF, Piloto_Pessoa_CPF)
values(8, 95, 183.92, 'Rua Rei Pelé', 'Santos', 'SP', 'Rua São Miguel','São Paulo','SP',
now(),'00613146336', '49813853905');

INSERT INTO
Pagamento(idPag,tipoPagamento,numCartao,tipoCartao,codSeguranca,dataValidade,troco,
Corrida_idcorrida,Corrida_Passageiro_Pessoa_CPF,Corrida_Piloto_Pessoa_CPF)
values(8, '1', '4539441168402459','1','565','08/29', null, 8, '00613146336', '49813853905');

INSERT INTO
Avaliacao(idAvaliacao,qtdEstrelas,comentario,Corrida_idcorrida,Corrida_Passageiro_Pessoa
_CPF,Corrida_Piloto_Pessoa_CPF)
values(8, 5, null, 8, '00613146336', '49813853905');
--
-----------------------------------------------------------------------------------------------------------------------
INSERT INTO Corrida(idcorrida, distancia, preco, logPartida, cidadePartida, estadoPartida,
cidadeDestino, logDestino,
estadoDestino, dataCorrida, Passageiro_Pessoa_CPF, Piloto_Pessoa_CPF)
values(9, 288, 234.90, 'Rua são João', 'Pirapora', 'MG', 'Rua Hortência','Patos','MG',
now(),'80415810809', '94289028454');

INSERT INTO
Pagamento(idPag,tipoPagamento,numCartao,tipoCartao,codSeguranca,dataValidade,troco,
Corrida_idcorrida,Corrida_Passageiro_Pessoa_CPF,Corrida_Piloto_Pessoa_CPF)
values(9, '1', '30015715131282','3','362','08/29', null, 9, '80415810809', '94289028454');
```

```
INSERT INTO
Avaliacao(idAvaliacao,qtdEstrelas,comentario,Corrida_idcorrida,Corrida_Passageiro_Pessoa
_CPF,Corrida_Piloto_Pessoa_CPF)
values(9, 0, null, 9, '80415810809', '94289028454');

--
-----------------------------------------------------------------------------------------------------------------
INSERT INTO Corrida(idcorrida, distancia, preco, logPartida, cidadePartida, estadoPartida,
cidadeDestino, logDestino,
estadoDestino, dataCorrida, Passageiro_Pessoa_CPF, Piloto_Pessoa_CPF)
values(10, 438, 576.34, 'Rua do Ouro', 'Ouro Preto', 'MG', 'Rua Pão de Açucar','Volta
Redonda','RJ', now(),'15440525025', '30779797566');

INSERT INTO
Pagamento(idPag,tipoPagamento,numCartao,tipoCartao,codSeguranca,dataValidade,troco,
Corrida_idcorrida,Corrida_Passageiro_Pessoa_CPF,Corrida_Piloto_Pessoa_CPF)
values(10, '1', '4532098577426979','2','982','11/21', null, 10, '15440525025', '30779797566');

INSERT INTO
Avaliacao(idAvaliacao,qtdEstrelas,comentario,Corrida_idcorrida,Corrida_Passageiro_Pessoa
_CPF,Corrida_Piloto_Pessoa_CPF)
values(10, 0, null, 10, '15440525025', '30779797566');
```

d) EXEMPLOS USANDO UPDATE

Alterando o endereço de uma pessoa

```
UPDATE Pessoa SET CEP = '12335390', estado = 'SP', cidade = 'Jacareí', bairro = 'Parque
Meia Lua', log = 'Rua José Aristeu da Cunha', num = '638'
WHERE CPF = '94289028454';
```

Alterando o RG de um passageiro

```
UPDATE Passageiro SET RG = '464096947' WHERE Pessoa_CPF = '15440525025';
```

Alterando a CNH de um piloto

```
UPDATE Piloto SET CNH = '55619473899' WHERE Pessoa_CPF = '60098656465';
```

Alterando o cor de uma moto

```
UPDATE Moto SET cor = 'Vermelha' WHERE placa = 'NAV6805';
```

Alterando o telefone de uma pessoa usando aninhamento

```
UPDATE telefonePessoa SET telefone = '12987482224' WHERE Pessoa_CPF IN (SELECT
CPF FROM Pessoa WHERE (nome = 'Juliana' AND sobrenome = 'Galvani Gregh'));
```

e) EXEMPLOS USANDO DELETE

Deletando um piloto pelo CPF
DELETE FROM piloto WHERE Pessoa_CPF = '02855606896';

Deletando um passageiro pelo RG
DELETE FROM passageiro WHERE RG = '127467531';

Deletando a avaliação de identificação 9
DELETE FROM avaliacao WHERE idAvaliacao = 9;
Deletando o pagamento do passageiro de cpf 12345678900 e piloto de cpf 28493614106
DELETE FROM pagamento WHERE Corrida_Passageiro_Pessoa_CPF = '12345678900' AND Corrida_Piloto_Pessoa_CPF = '28493614106';

Deletando por aninhamento a piloto de nome Aline Teixeira Alves
DELETE FROM Piloto WHERE Pessoa_CPF IN (SELECT CPF FROM Pessoa WHERE (nome = 'Aline' AND sobrenome = 'Teixeira Alves'));

f) EXEMPLOS USANDO CONSULTAS (SELECT, FROM, WHERE, ORDER BY, GROUP BY, HAVING) E OPERADORES (JOIN, OUTER JOIN, UNION, AND, OR, NOT, BETWEEN, IN, LIKE, IS NULL, ANY/SOME, ALL, EXISTS)

Recupera o nome ordenado e cpf de todas as mulheres cadastradas
SELECT nome, cpf
FROM Pessoa
WHERE sexo = 'F'
ORDER BY nome;

Recupera nome e bairro dos passageiros do sexo feminino e que não moram na UFLA
SELECT p.nome, p.bairro
FROM pessoa p JOIN passageiro pa ON p.CPF = pa.Pessoa_CPF
WHERE NOT p.bairro = "UFLA" AND p.sexo = "F";

Recupera os estados em que temos pilotos cadastrados
SELECT DISTINCT estado AS 'Temos pilotos cadastrados em'
FROM pessoa, piloto
WHERE pessoa.CPF = piloto.Pessoa_CPF;

Recupera nome de pessoas que começam com a letra A, seu sobrenome e data de Nascimento
SELECT nome, sobrenome, dataNascimento
FROM Pessoa
WHERE nome LIKE 'A%';

Recupera o nome, cpf da pessoa, ano e cor da moto de placa 'NFB9054'

```
SELECT nome, cpf, ano, cor
FROM Pessoa JOIN Piloto ON Pessoa.CPF=Piloto.Pessoa_CPF NATURAL JOIN Moto
WHERE placa = 'NFB9054';
```

Recupera o CPF do piloto cuja corrida obteu-se 5 estrelas e o preco entre 100 a 200 reais

```
SELECT Piloto_Pessoa_CPF, preco
FROM Corrida C JOIN Avaliacao A ON C.idCorrida=A.Corrida_idCorrida
WHERE qtdEstrelas = '5' AND preco BETWEEN 100.00 AND 200.00;
```

Recupera todos os nome dos Pilotos que tenha, percorrido a distância maior que 100km ou a moto seja da cor PRATA

```
SELECT ALL  P.nome
FROM  Pessoa P, Corrida C
WHERE C.distancia>100 AND C.Piloto_Pessoa_CPF=P.CPF
UNION
SELECT P.nome
FROM Pessoa P, Moto M, Piloto Pl
WHERE M.cor = 'Prata' AND M.placa=Pl.placa AND Pl.Pessoa_CPF=P.CPF;
```

Agrupa por tipo de cartão e retorna o tipo do Cartão e a quantidade de cada, quando o tipo do cartão é maior que 1

```
SELECT tipoCartao, count(*) AS 'Quantidade'
FROM Corrida C JOIN Pagamento P ON P.Corrida_idCorrida=C.idCorrida
GROUP BY tipoCartao
HAVING tipoCartao>1;
```

Recupera o nome, sexo e estado dos pilotos que possuem placa 'JVE9775' OU que a cor da sua moto seja vermelha

```
SELECT nome, sexo, estado
FROM Piloto LEFT OUTER JOIN Pessoa ON Pessoa.CPF=Piloto.Pessoa_CPF
WHERE placa = 'JVE9775' OR
        placa IN (SELECT placa
             FROM Moto M
             WHERE M.cor = 'Vermelha');
```

Recupera nome, sexo, estado da placa = 'JVE9775' OU quando o troco do pagamento for nulo.

```
SELECT nome, sexo, estado
FROM Piloto LEFT OUTER JOIN Pessoa ON Pessoa.CPF=Piloto.Pessoa_CPF
WHERE placa = 'JVE9775' OR
                Pessoa_CPF IN (SELECT Pessoa_CPF
```

```
                                FROM Pagamento RIGHT OUTER JOIN Corrida ON
Corrida.idCorrida=Pagamento.Corrida_idCorrida JOIN Piloto ON
Piloto.Pessoa_CPF=Corrida.Piloto_Pessoa_CPF
                                WHERE troco IS NULL);
```

Seleciona os precos de corridas de Minas que sao maiores do que o maior valor de uma corrida em Sao Paulo

```
SELECT estadoPartida, preco
FROM Corrida C
WHERE estadoPartida='MG'AND preco >SOME (SELECT MAX(preco)
                                                FROM Corrida
                                                WHERE estadoPartida = 'SP');
```

Recupera o nome, cidade de origem e cidade de destino dos passageiros que viajaram com motorista Osvaldo

```
SELECT p.nome, c.cidadePartida AS Origem, c.logDestino AS Destino
FROM corrida c JOIN pessoa p ON c.Passageiro_Pessoa_CPF = p.CPF
WHERE EXISTS (SELECT *
                FROM corrida c JOIN piloto p ON c.Piloto_Pessoa_CPF = p.Pessoa_CPF
                JOIN pessoa pe ON p.Pessoa_CPF = pe.CPF
                WHERE nome = "Osvaldo" AND p.CPF = c.Passageiro_Pessoa_CPF);
```

Recupera o nome e sobrenome dos pilotos ordenados por melhores avaliações

```
SELECT DISTINCT p.nome, p.sobrenome, a.qtdEstrelas AS Avaliação, a.comentario
FROM avaliacao a NATURAL JOIN corrida c
        JOIN pessoa p ON c.Piloto_Pessoa_CPF = p.CPF
        ORDER BY a.qtdEstrelas DESC;
```

g) EXEMPLOS DE CRIAÇÃO DE VIEWS

Cria uma view para recuperar dados de viagens feitas por Passageiros

```
CREATE VIEW pilotoPassageiro AS
        SELECT pe.nome AS Passageiro, ps.nome AS Piloto, c.cidadePartida, c.logDestino
AS cidadeDestino, c.distancia, c.preco
                FROM corrida c JOIN passageiro p ON c.Passageiro_Pessoa_CPF =
p.Pessoa_CPF
                JOIN pessoa pe ON p.Pessoa_CPF = pe.CPF
                JOIN piloto pi ON c.Piloto_Pessoa_CPF = pi.Pessoa_CPF
                JOIN pessoa ps ON pi.Pessoa_CPF = ps.CPF;
```

Usando a view pilotoPassageiro para recuperar maior viagem registrada

```
SELECT passageiro, piloto, MAX(distancia) AS 'Distância máxima'
FROM pilotoPassageiro;
```

Usando a view pilotoPassageiro para recuperar os pilotos com quem o passageiro Denilson já viajou

```
SELECT piloto
FROM pilotoPassageiro
WHERE passageiro = 'Denilson';
```

Cria uma view com número de telefone e cpf de pessoas cadastradas

```
SELECT nome, telefone
FROM pessoatelefone JOIN passageiro p ON cpf = p.Pessoa_CPF;
```

Usa a view pessoatelefone para recuperar telefone de pessoas que são passageiros

```
SELECT nome, telefone
FROM pessoatelefone JOIN passageiro p ON cpf = p.Pessoa_CPF;
```

Usa a view pessoatelefone para recuperar telefones com DDD 35

```
SELECT nome, telefone
FROM pessoatelefone
WHERE telefone LIKE '35%';
```

Cria uma view com pilotos e suas motos cadastradas

```
CREATE VIEW pilotoMoto AS
        SELECT p.nome, m.placa, m.ano, m.cor
        FROM piloto pi NATURAL JOIN moto m
        JOIN pessoa p ON pi.Pessoa_CPF = p.CPF;
```

Usa a view pilotomoto para recuperar pilotos com motos de ano 2019 ou mais recentes

```
SELECT nome, placa, ano
FROM pilotomoto
WHERE ano >= '2019';
```

Usa a view pilotomoto para recuperar pilotos com motos de cor prata

```
SELECT nome, placa, cor
FROM pilotomoto
WHERE cor = 'Prata';
```

h) EXEMPLO DE CRIAÇÃO DE USUÁRIO, CONCESSÃO (GRANT) E REVOCAÇÃO (REVOKE) DE PERMISSÃO DE ACESSO

Cria o usuário Jussara

```
CREATE USER 'Jussara'@'Localhost' IDENTIFIED BY '1234';
```

Permite acesso total a tabela pessoa

```
GRANT ALL ON trabalhoisbd.pessoa TO 'Jussara'@'Localhost';
```

Permite acesso a selecão na tabela corrida a apenas alguns atributos

```
GRANT SELECT (cidadePartida, cidadeDestino, distancia, preco) ON trabalhoisbd.corrida
TO 'Jussara'@'Localhost';
```

Revoga o acesso aos atributos da tabela corrida
REVOKE SELECT (cidadePartida, cidadeDestino, distancia, preco) ON trabalhoisbd.corrida FROM 'Jussara'@'Localhost';

Cria o usuário Joao
CREATE USER 'Joao'@'Localhost' IDENTIFIED BY '4321';

Permite acesso total a tabela pessoa
GRANT ALL ON trabalhoisbd.moto TO 'Joao'@'Localhost';

Permite acesso a selecão na tabela corrida a apenas alguns atributos
GRANT SELECT ON trabalhoisbd.pagamento TO 'Joao'@'Localhost';

Revoga o acesso aos atributos da tabela corrida
REVOKE SELECT ON trabalhoisbd.pagamento FROM 'Joao'@'Localhost';

i) EXEMPLO DE PROCEDIMENTOS/FUNÇÕES, COM E SEM PARÂMETROS, DE ENTRADA DE SAÍDA (IF, CASE, WHEN, WHILE)

Cria um procedimento para cadastrar dados na tabela pessoa

```
DELIMITER //
CREATE PROCEDURE CadastraPessoa(
  pCPF CHAR(11),
  pNome VARCHAR(15),
  pSobrenome VARCHAR(40),
  pDataNasc char(10),
  pSexo CHAR(1),
  pCEP CHAR(8),
  pEstado CHAR(2),
  pCidade VARCHAR(30),
  PBairro VARCHAR(30),
  pLog VARCHAR (40),
  pNum INT(11),
  pDataCad DATETIME
)
BEGIN
  INSERT INTO pessoa
  VALUES (pCPF, pNome, pSobrenome, pDataNasc, pSexo, pCEP, pEstado, pCidade,
pBairro, pLog, pNum, pDataCad);
END //
$$
DELIMITER ;
```

Chama cadastraPessoa passando dados por parâmetro

```
CALL cadastraPessoa ('12345678988', 'Carlos', 'Alberto', '21/03/1998', 'M', '37210000', 'MG',
'Itumirim', 'Centro', 'Rua das laranjeiras', '521', now());
```

Cria um procedimento para mostrar corridas feitas em cidade passada por parâmetro

```
DELIMITER //
CREATE PROCEDURE corridasCidades (IN pCidade VARCHAR (30))
BEGIN
        SELECT c.cidadePartida, c.cidadeDestino, c.dataCorrida
   from corrida c
   where c.cidadePartida = pCidade;
END//
DELIMITER ;
```

Chama o procedimento corridaCidades passando por parâmetro uma cidade

```
CALL corridasCidades ('Lavras');
```

Cria um procedimento para retornar telefone de uma pessoa passada por parâmetro

```
DELIMITER //
CREATE PROCEDURE retornaTelefone (IN pNome VARCHAR (15), OUT Telefone
VARCHAR (11))
BEGIN
   SELECT t.telefone INTO telefone
   FROM pessoa p JOIN telefonepessoa t on p.CPF = t.Pessoa_CPF
   WHERE p.nome = pNome;
END //
DELIMITER ;
```

Chama o procedimento retornaTelefone passando por parâmetro um nome, e recebendo um telefone como retorno

```
CALL retornaTelefone ('Breno', @tel);
```

Mostrando o resultado na tela

```
SELECT @tel AS Telefone;
```

j) EXEMPLOS DE TRIGGERS (INSERÇÃO, ALTERAÇÃO, EXCLUSÃO)

Cria uma trigger para controle do atributo sexo

```
DELIMITER $$
CREATE TRIGGER before_pessoa_insert
BEFORE INSERT ON pessoa FOR EACH ROW
BEGIN
        IF (new.sexo = '1') THEN
                SET new.sexo = 'M';
        ELSEIF (new.sexo = '2') THEN
                SET new.sexo = 'F';
        ELSEIF (new.sexo NOT IN ('M','F')) THEN
```

```
            SIGNAL SQLSTATE '45000' SET message_text = 'Caracter inválido para o atributo sexo, informe M, F, 1 para M e 2 para F ';
        END IF;
END;
$$
DELIMITER ;
```

Exemplo de como disparar a trigger before_pessoa_insert

```
INSERT INTO Pessoa(CPF, nome, sobrenome, dataNascimento, sexo, CEP, estado, cidade, bairro, log, num, dataCad)
VALUES('78946464465', 'Denis', 'Pereira', '1972-08-15', 'G', '37200000', 'MG', 'Lavras', 'UFLA', 'Aquenta Sol', 209, NOW());
```

Cria uma trigger para controlar troca de motocicletas somente para mais novas

```
DELIMITER $$
CREATE TRIGGER after_moto_update
AFTER UPDATE ON moto
FOR EACH ROW
BEGIN
        IF OLD.ano > NEW.ano THEN
            SIGNAL SQLSTATE '45000' SET message_text = 'Você não pode trocar sua moto por um modelo mais antigo, fere os padrões de qualidade do aplicativo';
   END IF;
END;
$$
DELIMITER ;
```

Exemplo de como disparar a trigger after_moto_update

```
UPDATE moto
set ano = '2018'
where placa = 'IAC5850'
```

Cria uma trigger para não deixar excluir corridas do ano atual

```
DELIMITER $$
CREATE TRIGGER before_corrida_delete
BEFORE DELETE ON corrida
FOR EACH ROW
BEGIN
        IF old.dataCorrida > "2021-01-01" THEN
            SIGNAL SQLSTATE '45000' SET message_text = 'Não pode excluir corridas do ano atual, somente dos anos anteriores';
        END IF;
END;
$$
DELIMITER ;
```

Exemplo de como disparar a trigger before_corrida_delete

```
delete from corrida
```

```
where idcorrida = '5';
```