Desenho de personagem de desenho animado

Descrição gerada automaticamente com confiança baixa

**PONTIFÍCIA UNIVERSIDADE CATÓLICA**

**DO RIO GRANDE DO SUL**

**GUSTAVO FILIPI LOPES MACHADO**

**Bacharelado em Ciência da Computação**

**Semestre 2024/1**

**RELATÓRIO DE DESENVOLVIMENTO**

**TRABALHO 1**

**BANCO DE DADOS II**

**Profª Denise Bandeira**

1. **DESCRIÇÃO FORMAL E UNIVERSO DE DISCURSO**

Considere o banco de dados sobre um sistema para registro das viagens de aplicativo realizadas por passageiros. O sistema mantém cadastro dos passageiros e dos carros, com seus respectivos motoristas. E também o registro das corridas realizadas pelos carros e com quais passageiros.

Cada carro é descrito por um código, a placa, marca, modelo e ano do carro, assim como o nome do motorista. No contexto deste trabalho, será armazenado somente o nome do motorista. Sobre o passageiro, serão guardados seu CPF, nome, email, telefone e bairro de sua residência.

Por fim, o sistema registra as informações da viagem, relacionando carros e passageiros, e guardando o preço cobrado pela viagem, distância percorrida e o bairro de destino da viagem.

1. **MODELO CONCEITUAL FORNECIDO – BR MODELO**

**Diagrama

Descrição gerada automaticamente**

1. **ALTERAÇÕES NO MODELO CONCEITUAL – TAREFA I**

Para a realização da tarefa 1 que solicitava a inserção de ao menos um atributo inédito por parte do estudante, fora adicionado ao modelo:

* 1. Entidade “Motorista”, com os seguintes atributos: Id, Email, CPF, Nome e CNH.
  2. Relacionamento entre a entidade “Motorista” com a entidade “Carro”, intitulado como “dirige”, de cardinalidade 1 para 1.
  3. Atributo “CodigoCarro” fora adicionado a entidade “Viagem”, visando um melhor controle e facilidade nas consultas a serem elaboradas.
  4. Atributo “BairroDestino” adicionado na entidade de especialização “Urbana”.
  5. Atributo “CidadePartida” adicionada na entidade de especialização “Interurbana”.

1. **GERANDO OS SCRIPTS PARA ORACLE SQL – TAREFA II**

Após a modelagem do modelo relacional no software de modelagem Astah, cria-se os seguintes scripts:

|  |
| --- |
| CREATE TABLE Motorista (  Id INT PRIMARY KEY, -- Identificador único do motorista  CNH CHAR(11), -- Número da Carteira Nacional de habilitação do motorista  Nome CHAR(80), -- Nome completo do motorista  CPF CHAR(11) UNIQUE, -- CPF do motorista (únicas)  Email CHAR(50) -- Endereço de e-mail do motorista  ); |

|  |
| --- |
| CREATE TABLE Passageiro (  CPF CHAR(11) PRIMARY KEY, -- CPF do passageiro (chave primária)  Nome CHAR(80), -- Nome completo do passageiro  Email CHAR(50), -- Endereço de e-mail do passageiro  Telefone CHAR(11), -- Número de telefone do passageiro  Bairro CHAR(40), -- Bairro de residência do passageiro  MeiodePagamento CHAR(20) -- Meio de pagamento preferido do passageiro  ); |

|  |
| --- |
| CREATE TABLE Carro (  Codigo INT PRIMARY KEY, -- Identificador único do carro (chave primária)  Placa CHAR(7), -- Placa do carro  Marca CHAR(20), -- Marca do carro  Modelo CHAR(40), -- Modelo do carro  Ano INT, -- Ano de fabricação do carro  CPF\_Motorista CHAR(11), -- CPF do motorista associado ao carro  FOREIGN KEY (CPF\_Motorista) REFERENCES Motorista(CPF) -- Chave estrangeira para vincular a um motorista  ); |

|  |
| --- |
| CREATE TABLE Viagem (  Id CHAR(10) PRIMARY KEY, -- Identificador único da viagem (chave primária)  CodigoCarro INT, -- Código do carro utilizado na viagem  CPF\_Passageiro CHAR(11), -- CPF do passageiro que fez a viagem  TipoCorrida CHAR(10), -- Tipo de corrida: ’Interurbana’ ou ’Urbana’  BairroPartida CHAR(30),  -- Bairro de partida da viagem (pode ser nulo para corridas interurbanas)  CidadeDestino CHAR(30), -- Cidade de destino da viagem (pode ser nulo para corridas urbanas)  Kms INT, -- Distância percorrida na viagem  Valor INT, -- Valor da viagem  FOREIGN KEY (CodigoCarro) REFERENCES Carro(Codigo),  -- Chave estrangeira para vincular a um carro  FOREIGN KEY (CPF\_Passageiro) REFERENCES Passageiro(CPF) -- Chave estrangeira para vincular a um passageiro  ); |

1. **INSTANCIANDO OBJETOS NO ORACLE – TAREFA III**

|  |
| --- |
| --Inserção de motoristas:  INSERT INTO Motorista (Id, CNH, Nome, CPF, Email)  VALUES (1, ’12345678901’, ’Dominic Toretto’, ’12345678900’, ’toretto@email.com’);  INSERT INTO Motorista (Id, CNH, Nome, CPF, Email)  VALUES (2, ’98765432109’, ’Relampago McQueen’, ’98765432100’, ’mcqueen@email.com’);  INSERT INTO Motorista (Id, CNH, Nome, CPF, Email)  VALUES (3, ’65432198707’, ’Brian O Conner’, ’65432198700’, ’eobrian@email.com’);  --Inserção de passageiros:  INSERT INTO Passageiro (CPF, Nome, Email, Telefone, Bairro, MeiodePagamento)  VALUES (’11122233344’, ’Brenda Meeks’, ’brendameeks@email.com’, ’11123456789’, ’Centro’  , ’Credito’);  INSERT INTO Passageiro (CPF, Nome, Email, Telefone, Bairro, MeiodePagamento)  VALUES (’22233344455’, ’Fogell McLovin’, ’mclovin@email.com’, ’11234567890’, ’Moinhos’  , ’Dinheiro’);  INSERT INTO Passageiro (CPF, Nome, Email, Telefone, Bairro, MeiodePagamento)  VALUES (’33344455566’, ’Regina George’, ’reginageorge@email.com’, ’11345678901’, ’Floresta’  , ’Debito’);  INSERT INTO Passageiro (CPF, Nome, Email, Telefone, Bairro, MeiodePagamento)  VALUES (’44455566677’, ’Elle Woods’, ’ellewoods@email.com’, ’11456789012’, ’Cascata’  , ’Credito’);  INSERT INTO Passageiro (CPF, Nome, Email, Telefone, Bairro, MeiodePagamento)  VALUES (’55566677788’, ’Jenna Rink’, ’jennar\_30@email.com’, ’11567890123’, ’Partenon’  , ’Dinheiro’);  --Inserção de carros:  INSERT INTO Carro (Codigo, Placa, Marca, Modelo, Ano, CPF\_Motorista)  VALUES (101, ’ABC1234’, ’Toyota’, ’Corolla’, 2020, ’12345678900’);  INSERT INTO Carro (Codigo, Placa, Marca, Modelo, Ano, CPF\_Motorista)  VALUES (102, ’MCQ2006’, ’Chevrolet’, ’Corvete C6’, 2018, ’98765432100’);  INSERT INTO Carro (Codigo, Placa, Marca, Modelo, Ano, CPF\_Motorista)  VALUES (103, ’DEF9012’, ’Volkswagen’, ’Gol’, 2019, ’65432198700’);  --Inserção de viagens:  INSERT INTO Viagem (Id, CodigoCarro, CPF\_Passageiro, TipoCorrida, BairroPartida,  CidadeDestino, Kms, Valor)  VALUES (’000001’, 101, ’11122233344’, ’Urbana’, ’Centro’, NULL, 10, 20);  INSERT INTO Viagem (Id, CodigoCarro, CPF\_Passageiro, TipoCorrida, BairroPartida,  CidadeDestino, Kms, Valor)  VALUES (’000002’, 102, ’22233344455’, ’Urbana’, ’Moinhos’, NULL, 15, 25);  INSERT INTO Viagem (Id, CodigoCarro, CPF\_Passageiro, TipoCorrida, BairroPartida,  CidadeDestino, Kms, Valor)  VALUES (’000003’, 103, ’33344455566’, ’Urbana’, ’Floresta’, NULL, 12, 22);  INSERT INTO Viagem (Id, CodigoCarro, CPF\_Passageiro, TipoCorrida, BairroPartida,  CidadeDestino, Kms, Valor)  VALUES (’000004’, 101, ’44455566677’, ’Urbana’, ’Floresta’, NULL, 8, 18);  INSERT INTO Viagem (Id, CodigoCarro, CPF\_Passageiro, TipoCorrida, BairroPartida,  CidadeDestino, Kms, Valor)  VALUES (’000005’, 102, ’55566677788’, ’Urbana’, ’Cascata’, NULL, 20, 30);  INSERT INTO Viagem (Id, CodigoCarro, CPF\_Passageiro, TipoCorrida, BairroPartida,  CidadeDestino, Kms, Valor)  VALUES (’000006’, 103, ’11122233344’, ’Urbana’, ’Partenon’, NULL, 25, 35);  INSERT INTO Viagem (Id, CodigoCarro, CPF\_Passageiro, TipoCorrida, BairroPartida,  CidadeDestino, Kms, Valor)  VALUES (’000007’, 101, ’22233344455’, ’Interurb’, ’Moinhos’, ’Canoas’, 50, 100);  INSERT INTO Viagem (Id, CodigoCarro, CPF\_Passageiro, TipoCorrida, BairroPartida,  CidadeDestino, Kms, Valor)  VALUES (’000008’, 102, ’33344455566’, ’Interurb’, ’Moinhos’, ’Gravatai’, 70, 120);  INSERT INTO Viagem (Id, CodigoCarro, CPF\_Passageiro, TipoCorrida, BairroPartida,  CidadeDestino, Kms, Valor)  VALUES (’000009’, 103, ’44455566677’, ’Interurb’, ’Floresta’, ’Canoas’, 60, 110);  INSERT INTO Viagem (Id, CodigoCarro, CPF\_Passageiro, TipoCorrida, BairroPartida,  CidadeDestino, Kms, Valor)  VALUES (’000010’, 101, ’55566677788’, ’Interurb’, ’Partenon’, ’Canoas’, 80, 130); |

1. **MODELAGEM NÃO-RELACIONAL DO SISTEMA – TAREFA IV**

**Diagrama

Descrição gerada automaticamente**

O objetivo principal nesta modelagem é tornar o processo de busca mais simples de um ponto de vista computacional, ocupando um espaço maior de armazenamento, mas evitando o uso de operações “$lookup” no MongoDB, equivalente a operação “JOIN” no Oracle SQL.

Foram criados objetos e Collections dentro do Mongo DB com a utilização da Java Script Object Notation (JSON) . A instanciação dos objetos se encontra no arquivo “Viagens.json” na pasta na qual está contido o presente relatório, não sendo incluído diretamente aqui devido a sua extensão.

1. **CONSULTAS: ORACLE VS MONGO ATLAS – TAREFA V**

Foram elaboradas as consultas solicitadas na descrição do trabalho, segue abaixo:

**1.** Listar os Códigos dos Carros que realizaram corridas que partiram do bairro 'Floresta’:

**ORACLE:**

**Interface gráfica do usuário

Descrição gerada automaticamente com confiança média**

**MONGO ATLAS – VIAGENS.AGGREGATE:**

**Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente**

**2.** Listar o nome dos passageiros e o código dos carros que fizeram corridas com esses passageiros, mas somente para as corridas que tem o mesmo bairro de partida do cadastrado para o passageiro:

**ORACLE:**

**Uma imagem contendo Tabela

Descrição gerada automaticamente**

**MONGO ATLAS – VIAGENS.AGGREGATE:**

**Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente**

**Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente**

**3.** Listar o CPF e nome dos passageiros e a quantidade total de quilômetros percorridos por cada um deles em suas corridas de taxi.

**ORACLE:**

**Interface gráfica do usuário

Descrição gerada automaticamente**

**MONGO ATLAS – PASSAGEIROS.AGGREGATE:**

**Texto

Descrição gerada automaticamente**

**Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente**

1. **CONCLUSÃO**

O modelo não-relacional, embora ocupe um espaço ligeiramente maior que o modelo relacional, oferece a capacidade de realizar consultas sem recorrer a operações de junção, como o comando SQL "JOIN". Embora ambas as abordagens permitam consultas eficientes, a ausência de operações de junção torna o modelo não-relacional mais simples e com menor custo computacional, o que é especialmente vantajoso no contexto descrito no trabalho 1.