



UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS – PICOS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO
DISCIPLINA: BANCO DE DADOS I – 2022.2
PROFESSOR: FRANCISCO DAS CHAGAS IMPERES FILHO

Trabalho Final

Baseando-se no Modelo Entidade Relacional (M-ER) desenvolver um projeto lógico de banco de dados para um problema do mundo real (mini-mundo). O modelo deve ser capaz de mapear o esquema lógico para esquema conceitual e possibilitar o acesso a informações armazenadas em um Sistema Gerenciador de Banco de Dados (SGBD) como, por exemplo, Firebird, MySQL, PostgreSQL, Oracle, DB2 etc. O M-ER deverá satisfazer às necessidades da empresa/problema de acordo com a(s) entrevista(s) realizada(s) com o(s) gestor(es) da organização.

Fase II – Projetos Lógico e Físico de BD

A Fase II do trabalho final da disciplina é uma continuação do que foi definido nas Fase I. Além das especificações contidas na fase anterior, a Fase II deve apresentar o D-ER do projeto e os scripts para criação do esquema físico de banco de dados (instruções DDL) e demais características pertinentes ao projeto.

Data para postagem no SIGAA:

Início.....: 14/01/2023

Prazo final: 25/01/2023

Apresentações: 26/01/2023

Nota: 4 (quatro) pontos.



UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS – PICOS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO
PROFESSOR: FRANCISCO DAS CHAGAS IMPERES FILHO

PROJETO FINAL DA DISCIPLINA BANCO DE DADOS I

Componentes:

Aluno 1
Aluno 2
Aluno 3

1. Informações sobre o projeto

Temática: Comércio Automotivo

Título:.....: RevAutos - Revenda de Automóveis

1.1 Requisitos e principais características da aplicação:

A presente proposta representa as necessidades de armazenamento de dados da empresa RevAutos, atuante no comércio de revenda de veículos e motocicletas novas e seminovas na cidade Cangaço dos Autos, região do semiárido nordestino. Após interação e levantamento de requisitos juntos aos usuários finais (diretores, gerentes, vendedores), o estudo de caso destaca os requisitos relacionados a seguir:

- Controlar a comercialização de veículos da empresa.
- Armazenar marcas e a categorias dos veículos:
 - Categorias identificadas: passeio, carga, transporte de passageiros e motocicleta.
- Vincular o veículo ao respectivo proprietário (clientes).
- Relacionar vendedores a gerentes.
 - OBS: Um gerente pode gerenciar um ou mais vendedores. Um vendedor só pode ser gerenciado por um gerente.
- Saber em qual cidade e estado o veículo foi devidamente emplacado.

1.2 Principais entidades:

- Estados
- Municípios
- Marcas_dos_Veículos
- Categorias_dos_Veículos
- Seguradoras
- Veículos
- Vendedores
- Proprietários
- Vendas
 - Relacionamento ternário entre as entidades Veículos, Vendedores e Proprietários
- Contatos_Telefônicos
 - Entidade franca – depende da existência da entidade Proprietários.

Os atributos com seus respectivos tipos de dados podem ser visualizados na próxima seção.

2. Diagrama Entidade-Relacionamento (Exemplo)

Ferramenta utilizada para criação do diagrama:

- MySQL Workbench
 - Versão 8.0.15 (64 bits)

O diagrama de banco de dados relacional para o sistema de gerenciamento de vendas de veículos apresenta as seguintes tabelas e atributos:

- Marcas**: id_Marca (INT, PK), ds_Marca (VARCHAR(45)).
- Categorias**: id_Categoria (INT, PK), ds_categoria (VARCHAR(45)).
- Seguradoras**: id_seguradora (INT, PK), nm_seguradora (VARCHAR(100)), nm_responsavel (VARCHAR(100)), cnpj_seguradora (VARCHAR(13)), dt_fundacao (DATE).
- Veiculos**: id_Veiculo (INT, PK), chassi (VARCHAR(17)), placa (VARCHAR(7)), ano_fabricacao (DATE), ano_modelo (DATE), Municipios_id_Municipio (INT, FK), Marcas_id_Marca (INT, FK), Categorias_id_Categoria (INT, FK), Seguradoras_id_seguradora (INT, FK).
- Municipios**: id_Municipio (INT, PK), nm_municipio (VARCHAR(100)), Estados_id_estado (INT, FK).
- Estados**: id_estado (INT, PK), Nm_estado (VARCHAR(100)).
- Proprietarios**: id_Proprietario (INT, PK), nm_proprietario (VARCHAR(100)), cpf_proprietario (VARCHAR(11)), RG (VARCHAR(10)), tp_logradouro (VARCHAR(15)), nm_logradouro (VARCHAR(60)), end_numero (VARCHAR(6)), end_complemento (VARCHAR(45)), dt_nascimento_proprietario (DATE), Municipios_id_Municipio (INT, FK).
- Vendas**: id_Venda (INT, PK), Veiculos_id_Veiculo (INT, FK), Proprietarios_id_Proprietario (INT, FK), Vendedores_id_Vendedor (INT, FK), dt_venda (DATE), qt_vendida (INT), vl_unitario (DECIMAL(10,2)).
- Vendedores**: id_vendedor (INT, PK), nm_vendedor (VARCHAR(100)), cpf_vendedor (VARCHAR(11)), contato_vendedor (VARCHAR(45)), dt_nascimento_vendedor (DATE), Vendedores_id_vendedor (INT, FK).
- Contatos**: id_Contato (INT, PK), Proprietarios_id_Proprietario (INT, FK), nu_contato (VARCHAR(11)).

As relações entre as tabelas são as seguintes:

- Marcas** e **Veiculos**: Relação 1:N.
- Categorias** e **Veiculos**: Relação 1:N.
- Seguradoras** e **Veiculos**: Relação 1:N.
- Municipios** e **Veiculos**: Relação 1:N.
- Municipios** e **Estados**: Relação 1:N.
- Proprietarios** e **Estados**: Relação 1:N.
- Proprietarios** e **Vendas**: Relação 1:N.
- Vendedores** e **Vendas**: Relação 1:N.
- Proprietarios** e **Contatos**: Relação 1:N.

Adicionalmente, há uma relação de generalização entre **Vendedores** e **Proprietarios**, onde **Vendedores** é um tipo específico de **Proprietario**, indicado pelo símbolo "Gerente <---- Vendedor".

-- Schema DER_Veiculos2

CREATE SCHEMA IF NOT EXISTS `DER_Veiculos2` DEFAULT CHARACTER SET utf8 ;
USE `DER_Veiculos2` ;

-- Table `Estados`

DROP TABLE IF EXISTS `Estados` ;

CREATE TABLE IF NOT EXISTS `Estados` (
 `id_estado` INT NOT NULL AUTO_INCREMENT,
 `Nm_estado` VARCHAR(100) NOT NULL,
 PRIMARY KEY (`id_estado`))
ENGINE = InnoDB;

-- Table `Municipios`

DROP TABLE IF EXISTS `Municipios` ;

CREATE TABLE IF NOT EXISTS `Municipios` (
 `id_Municipio` INT NOT NULL AUTO_INCREMENT,
 `nm_municipio` VARCHAR(100) NOT NULL,
 `Estados_id_estado` INT NOT NULL,
 PRIMARY KEY (`id_Municipio`),
 CONSTRAINT `fk_Municipios_Estados`
 FOREIGN KEY (`Estados_id_estado`)
 REFERENCES `Estados` (`id_estado`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- Table `Marcas`

DROP TABLE IF EXISTS `Marcas` ;

CREATE TABLE IF NOT EXISTS `Marcas` (
 `id_Marca` INT NOT NULL AUTO_INCREMENT,
 `ds_Marca` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`id_Marca`))
ENGINE = InnoDB;

-- Table `Categorias`

DROP TABLE IF EXISTS `Categorias` ;

CREATE TABLE IF NOT EXISTS `Categorias` (
 `id_categoria` INT NOT NULL AUTO_INCREMENT,
 `ds_categoria` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`id_categoria`))
ENGINE = InnoDB;

```
`id_Categoria` INT NOT NULL AUTO_INCREMENT,  
`ds_categoria` VARCHAR(45) NOT NULL,  
PRIMARY KEY (`id_Categoria`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `Seguradoras`  
-----
```

```
DROP TABLE IF EXISTS `Seguradoras` ;
```

```
CREATE TABLE IF NOT EXISTS `Seguradoras` (  
  `id_seguradora` INT NOT NULL AUTO_INCREMENT,  
  `nm_seguradora` VARCHAR(100) NOT NULL,  
  `nm_responsavel` VARCHAR(100) NOT NULL,  
  `cnpj_seguradora` VARCHAR(13) NOT NULL,  
  `dt_fundacao` DATE NOT NULL,  
  PRIMARY KEY (`id_seguradora`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `Veiculos`  
-----
```

```
DROP TABLE IF EXISTS `Veiculos` ;
```

```
CREATE TABLE IF NOT EXISTS `Veiculos` (  
  `id_Veiculo` INT NOT NULL AUTO_INCREMENT,  
  `chassi` VARCHAR(17) NOT NULL,  
  `placa` VARCHAR(7) NULL,  
  `ano_fabricação` DATE NOT NULL,  
  `ano_modelo` DATE NOT NULL,  
  `Municipios_id_Municipio` INT NOT NULL,  
  `Marcas_id_Marca` INT NOT NULL,  
  `Categorias_id_Categoria` INT NOT NULL,  
  `Seguradoras_id_seguradora` INT NOT NULL,  
  PRIMARY KEY (`id_Veiculo`),  
  CONSTRAINT `fk_Veiculos_Municipios1`  
    FOREIGN KEY (`Municipios_id_Municipio`)  
    REFERENCES `Municipios` (`id_Municipio`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Veiculos_Marcas1`  
    FOREIGN KEY (`Marcas_id_Marca`)  
    REFERENCES `Marcas` (`id_Marca`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Veiculos_Categorias1`  
    FOREIGN KEY (`Categorias_id_Categoria`)  
    REFERENCES `Categorias` (`id_Categoria`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,
```



```

CONSTRAINT `fk_Veiculos_Seguradoras1`
  FOREIGN KEY (`Seguradoras_id_seguradora`)
  REFERENCES `Seguradoras` (`id_seguradora`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `Proprietarios`
-- -----

```

```

DROP TABLE IF EXISTS `Proprietarios` ;

```

```

CREATE TABLE IF NOT EXISTS `Proprietarios` (
  `id_Proprietario` INT NOT NULL AUTO_INCREMENT,
  `nm_proprietario` VARCHAR(100) NOT NULL,
  `cpf_proprietario` VARCHAR(11) NOT NULL,
  `RG` VARCHAR(10) NOT NULL,
  `tp_logradouro` VARCHAR(15) NOT NULL DEFAULT 'Rua' COMMENT 'Aeroporto\nAlameda\
nÁrea\nAvenida\nCampo\nChácara\nColônia\nCondomínio\nConjunto\nDistrito\nRua\nOutros',
  `nm_logradouro` VARCHAR(60) NOT NULL,
  `end_numero` VARCHAR(6) NOT NULL,
  `end_complemento` VARCHAR(45) NULL,
  `dt_nascimento_proprietario` DATE NOT NULL,
  `Municipios_id_Municipio` INT NOT NULL,
  PRIMARY KEY (`id_Proprietario`),
  CONSTRAINT `fk_Proprietarios_Municipios1`
    FOREIGN KEY (`Municipios_id_Municipio`)
    REFERENCES `Municipios` (`id_Municipio`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `Vendedores`
-- -----

```

```

DROP TABLE IF EXISTS `Vendedores` ;

```

```

CREATE TABLE IF NOT EXISTS `Vendedores` (
  `id_vendedor` INT NOT NULL AUTO_INCREMENT,
  `nm_vendedor` VARCHAR(100) NOT NULL,
  `cpf_vendedor` VARCHAR(11) NOT NULL,
  `contato_vendedor` VARCHAR(45) NOT NULL,
  `dt_nascimento_vendedor` DATE NOT NULL,
  `Vendedores_id_vendedor` INT NOT NULL,
  PRIMARY KEY (`id_vendedor`),
  CONSTRAINT `fk_Vendedores_Vendedores1`
    FOREIGN KEY (`Vendedores_id_vendedor`)
    REFERENCES `Vendedores` (`id_vendedor`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

-- Table `Vendas`

DROP TABLE IF EXISTS `Vendas` ;

CREATE TABLE IF NOT EXISTS `Vendas` (
 `id_Venda` INT NOT NULL AUTO_INCREMENT,
 `Veiculos_id_Veiculo` INT NOT NULL,
 `Proprietarios_id_Proprietario` INT NOT NULL,
 `Vendedores_id_vendedor` INT NOT NULL,
 `dt_venda` DATE NOT NULL,
 `qt_vendida` INT NOT NULL,
 `vl_unitario` DECIMAL(12,2) NOT NULL,
 PRIMARY KEY (`id_Venda`),
 CONSTRAINT `fk_Vendas_Veiculos1`
 FOREIGN KEY (`Veiculos_id_Veiculo`)
 REFERENCES `Veiculos` (`id_Veiculo`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION,
 CONSTRAINT `fk_Vendas_Proprietarios1`
 FOREIGN KEY (`Proprietarios_id_Proprietario`)
 REFERENCES `Proprietarios` (`id_Proprietario`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION,
 CONSTRAINT `fk_Vendas_Vendedores1`
 FOREIGN KEY (`Vendedores_id_vendedor`)
 REFERENCES `Vendedores` (`id_vendedor`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- Table `Contatos`

DROP TABLE IF EXISTS `Contatos` ;

CREATE TABLE IF NOT EXISTS `Contatos` (
 `id_Contato` INT NOT NULL AUTO_INCREMENT,
 `Proprietarios_id_Proprietario` INT NOT NULL,
 `nu_cantato` VARCHAR(11) NOT NULL,
 PRIMARY KEY (`id_Contato`, `Proprietarios_id_Proprietario`),
 CONSTRAINT `fk_Contatos_Proprietarios1`
 FOREIGN KEY (`Proprietarios_id_Proprietario`)
 REFERENCES `Proprietarios` (`id_Proprietario`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

