

MODELAGEM DE DADOS

TRABALHO : DIAGRAMA ENTIDADE RELACIONAL

CURSO: GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

DISCIPLINA: MODELAGEM DE DADOS

UNIDADE FORMATIVA 2

ALUNO: LUIZ NUNES DE ALMEIDA JÚNIOR



Introdução

Este relatório técnico descreve o modelo Entidade-Relacionamento (ER) para o modelo que representa um o banco de dados de um sistema de aluguel de ciclomotores, denominado **db_move_rent**, que inclui entidades para pessoas (clientes), ciclomotores, locações, e trajetos realizados. O script SQL foi gerado por **MySQL Workbench** e implementa um banco de dados relacional com chaves primárias, chaves estrangeiras, índices e restrições de integridade referencial.

1. Entidades do Modelo Relacional

O modelo relacional é composto pelas seguintes entidades:

- **tbl_pessoa**: Armazena informações de clientes (pessoas).
- **tbl_ciclomotor**: Armazena dados sobre os ciclomotores disponíveis para locação.
- **tbl_locacao**: Registra as locações dos ciclomotores por parte dos clientes.
- **tbl_trajetos**: Registra informações sobre trajetos realizados em cada locação.

1.1. Entidade **tbl_pessoa**

A tabela **tbl_pessoa** (figura 1) armazena os dados dos clientes que podem alugar ciclomotores. Cada pessoa tem um identificador único, o **idPessoa**, que é a chave primária da tabela.

Campo	Tipo	Descrição
idPessoa	INT	Identificador único (chave primária).
cpf	VARCHAR(14)	CPF do cliente (único).
nome	VARCHAR(50)	Nome completo do cliente.
email	VARCHAR(100)	Email do cliente.
telefone	VARCHAR(14)	Telefone de contato.

Figura 1

- **Restrições:**

- **Chave primária:** `idPessoa`.
- **Índice único:** `cpf_UNIQUE` para garantir que não existam CPFs duplicados.

1.2. Entidade `tbl_ciclomotor`

A tabela `tbl_ciclomotor` (figura 2) contém os dados dos ciclomotores disponíveis para locação.

Campo	Tipo	Descrição
<code>cm_id</code>	INT	Identificador único do ciclomotor (chave primária).
<code>cm_nome</code>	VARCHAR(50)	Nome ou modelo do ciclomotor.
<code>cm_ano</code>	INT	Ano de fabricação (padrão: 2000).
<code>cm_cor</code>	VARCHAR(15)	Cor do ciclomotor (opcional).

Figura 2

- **Restrições:**
 - **Chave primária:** `cm_id`.

1.3. Entidade `tbl_locacao`

A tabela `tbl_locacao` (figura 3) registra as locações dos ciclomotores. Cada locação é associada a um cliente (na tabela `tbl_pessoa`) e a um ciclomotor (na tabela `tbl_ciclomotor`).

Campo	Tipo	Descrição
<code>lc_id</code>	INT	Identificador único da locação (chave primária).
<code>lc_data_locacao</code>	DATE	Data da locação (padrão: data atual).
<code>lc_hora_locacao</code>	VARCHAR(45)	Hora da locação (padrão: hora atual).
<code>lc_local_locacao</code>	VARCHAR(50)	Local onde ocorreu a locação.
<code>lc_id_pessoa</code>	INT	Chave estrangeira referenciando <code>tbl_pessoa</code> .
<code>lc_id_ciclomotor</code>	INT	Chave estrangeira referenciando <code>tbl_ciclomotor</code> .

Figura 3

- **Restrições:**
 - **Chave primária:** `lc_id`.
 - **Chave estrangeira** `fk_tbl_locacao_id_pessoa` referenciando a coluna `idPessoa` da tabela `tbl_pessoa`.
 - **Chave estrangeira** `fk_tbl_locacao_id_ciclomotor` referenciando a coluna `cm_id` da tabela `tbl_ciclomotor`.
- **Relacionamentos:**
 - **Um cliente** (pessoa) pode realizar **várias locações**.
 - **Um ciclomotor** pode ser locado em **várias locações**.

2. Relacionamentos entre as Entidades

- `tbl_locacao` → `tbl_pessoa` (**1:n**): Um cliente pode realizar várias locações, mas uma locação pertence a apenas um cliente.
- `tbl_locacao` → `tbl_ciclomotor` (**1:n**): Um ciclomotor pode ser locado várias vezes, mas cada locação envolve apenas um ciclomotor.
- `tbl_trajetos` → `tbl_locacao` (**1:n**): Cada locação pode ter vários trajetos associados, mas um trajeto pertence a apenas uma locação.

3. Restrições de Integridade Referencial

- **ON DELETE CASCADE:** Aplica-se quando um registro relacionado é excluído.

- Exemplo: Se um cliente (pessoa) for excluído, todas as locações associadas a ele serão automaticamente excluídas (chave estrangeira **lc_id_pessoa** em **tbl_locacao**).
- **ON UPDATE CASCADE**: Quando um registro relacionado é atualizado, as chaves estrangeiras que dependem desse registro também são atualizadas.

4. Resumo do Modelo ER

- Tabelas principais: **tbl_pessoa** (clientes), **tbl_ciclomotor** (ciclomotores), **tbl_locacao** (locações), **tbl_trajetos** (trajetos).
- Relacionamentos principais:
 - Uma pessoa pode realizar várias locações.
 - Um ciclomotor pode ser alugado várias vezes.
 - Uma locação pode conter vários trajetos.

O modelo implementa um sistema relacional para gerenciar aluguéis de ciclomotores, garantindo integridade referencial e restrições para evitar dados duplicados, como no caso do CPF dos clientes.

5. Representação visual do DER do Banco de dados Move Rent

Abaixo vemos na imagem (figura 5) a representação visual, construída usando o MySQL Workbench, do diagrama entidade relacional do banco de dados da Move Rent, conforme proposto no exercício:

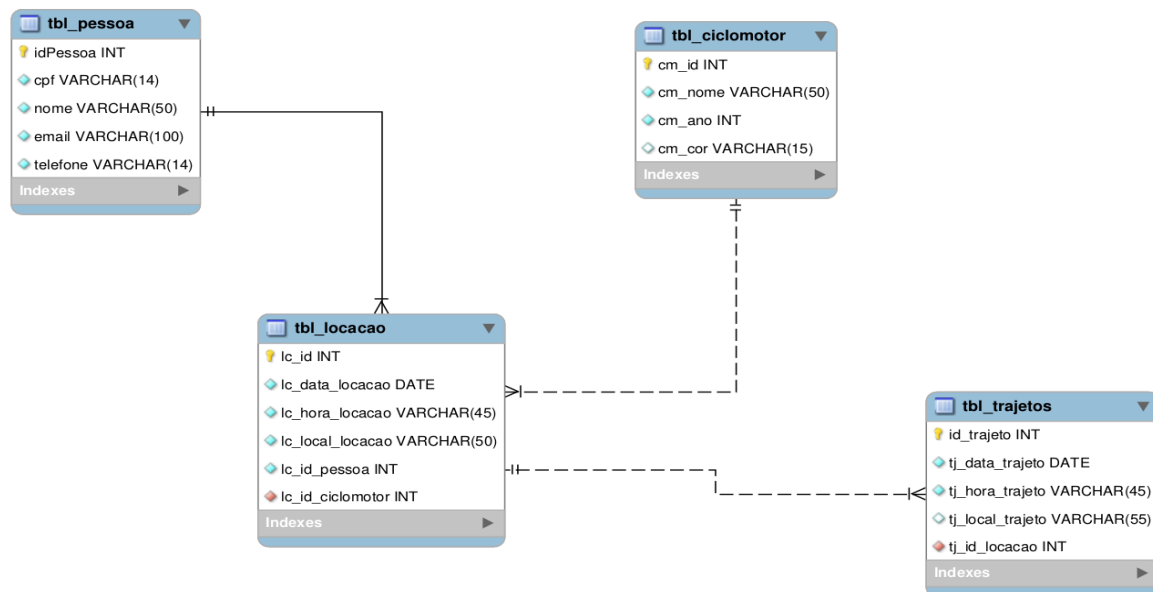


Figura 5

6. Scripts SQL para definição de entidades do Banco de Dados Move Rent

Abaixo, seguem os scripts SQL de criação das entidades do banco de dados Move Rent, criados a partir do DER do mesmo banco de dados, usando a ferramenta de geração de scripts do MYSQL WorkBench:


```
-- MySQL Script generated by MySQL Workbench
-- Sun Sep 15 23:13:12 2024
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,
NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```

-- -----
-- Schema db_move_rent
-- -----

-- Schema db_move_rent
-- -----

CREATE SCHEMA IF NOT EXISTS `db_move_rent` ;
USE `db_move_rent` ;

-- -----
-- Table `db_move_rent`.`tbl_pessoa`
-- -----

CREATE TABLE IF NOT EXISTS `db_move_rent`.`tbl_pessoa` (
  `idPessoa` INT NOT NULL AUTO_INCREMENT,
  `cpf` VARCHAR(14) NOT NULL,
  `nome` VARCHAR(50) NOT NULL,
  `email` VARCHAR(100) NOT NULL,
  `telefone` VARCHAR(14) NOT NULL,
  PRIMARY KEY (`idPessoa`),
  UNIQUE INDEX `cpf_UNIQUE` (`cpf` ASC) VISIBLE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;

```

```

-- -----
-- Table `db_move_rent`.`tbl_ciclomotor`
-- -----

CREATE TABLE IF NOT EXISTS `db_move_rent`.`tbl_ciclomotor` (
  `cm_id` INT NOT NULL AUTO_INCREMENT,
  `cm_nome` VARCHAR(50) NOT NULL,
  `cm_ano` INT NOT NULL DEFAULT 2000,
  `cm_cor` VARCHAR(15) NULL,
  PRIMARY KEY (`cm_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;

```

```

-- -----
-- Table `db_move_rent`.`tbl_locacao`
-- -----

CREATE TABLE IF NOT EXISTS `db_move_rent`.`tbl_locacao` (
  `lc_id` INT NOT NULL AUTO_INCREMENT,
  `lc_data_locacao` DATE NOT NULL DEFAULT CURRENT_DATE,
  `lc_hora_locacao` VARCHAR(45) NOT NULL DEFAULT 'CURRENT_TIME',
  `lc_local_locacao` VARCHAR(50) NOT NULL,
  `lc_id_pessoa` INT NOT NULL,
  `lc_id_ciclomotor` INT NOT NULL,
  PRIMARY KEY (`lc_id`),
  INDEX `fk_tbl_locacao_id_ciclomotor_idx` (`lc_id_ciclomotor` ASC) VISIBLE,
  CONSTRAINT `fk_tbl_locacao_id_pessoa`
    FOREIGN KEY (`lc_id_pessoa`)
      REFERENCES `db_move_rent`.`tbl_pessoa` (`idPessoa`)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
  CONSTRAINT `fk_tbl_locacao_id_ciclomotor`
    FOREIGN KEY (`lc_id_ciclomotor`)
      REFERENCES `db_move_rent`.`tbl_ciclomotor` (`cm_id`)
      ON DELETE NO ACTION
      ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;

```

```

-- -----
-- Table `db_move_rent`.`tbl_trajetos`
-- -----

CREATE TABLE IF NOT EXISTS `db_move_rent`.`tbl_trajetos` (
  `id_trajeto` INT NOT NULL AUTO_INCREMENT,
  `tj_data_trajeto` DATE NOT NULL DEFAULT CURRENT_DATE,
  `tj_hora_trajeto` VARCHAR(45) NOT NULL DEFAULT 'CURRENT_TIME',
  `tj_local_trajeto` VARCHAR(55) NULL,
  `tj_id_locacao` INT NOT NULL,
  PRIMARY KEY (`id_trajeto`),
  INDEX `fk_tbl_trajetos_id_locacao_idx` (`tj_id_locacao` ASC) VISIBLE,
  CONSTRAINT `fk_tbl_trajetos_id_locacao`
    FOREIGN KEY (`tj_id_locacao`)
      REFERENCES `db_move_rent`.`tbl_locacao` (`lc_id`)
      ON DELETE CASCADE
      ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4;

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

```