

BANCO DE DADOS

DQL & DTL

PROJETO

INTEGRADOR

Professor

Danilo Soares

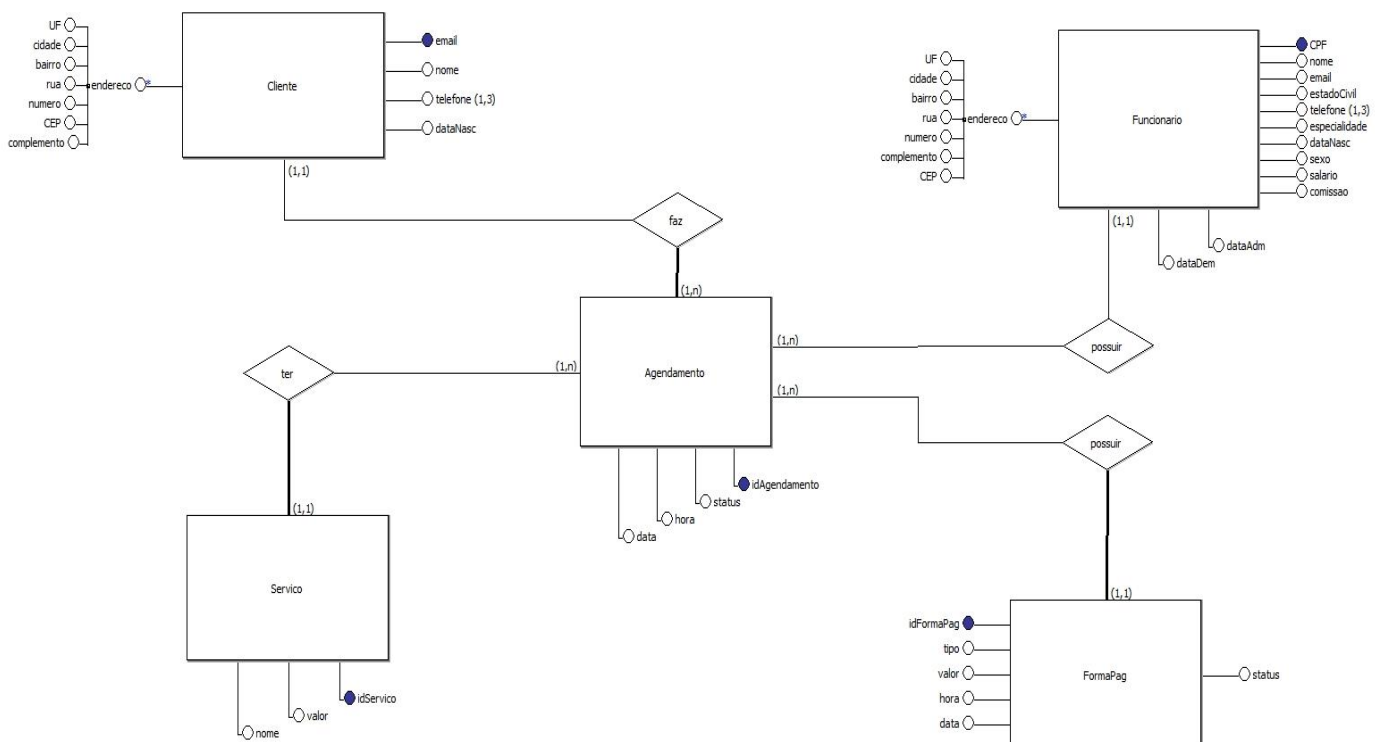
Documentação do Projeto: Sistema de Gestão de Salão de Beleza

Introdução

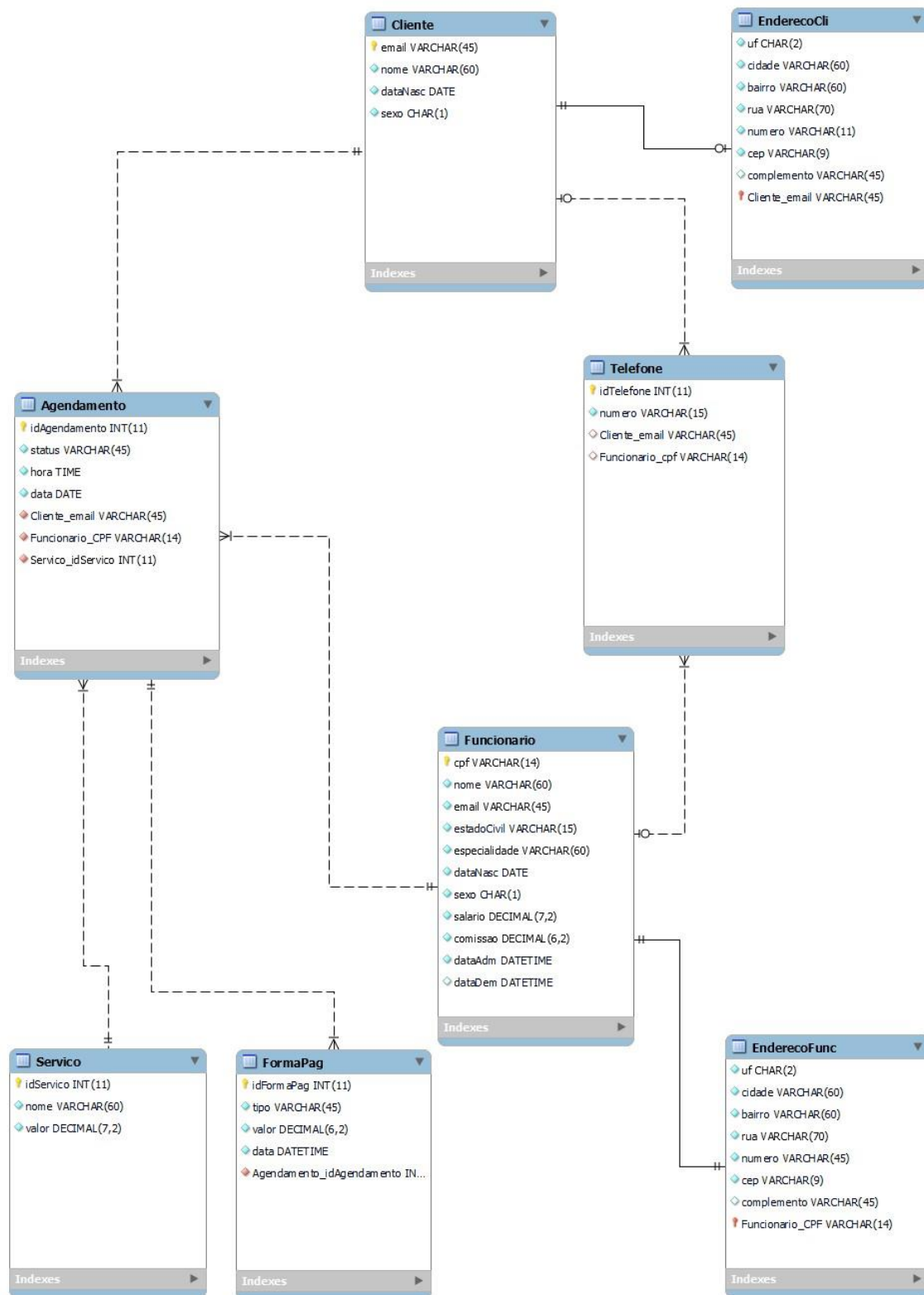
Este projeto visa criar um sistema de gestão para um salão de beleza, permitindo o gerenciamento de clientes, funcionários, agendamentos, serviços, pagamentos e endereços. Abaixo estão as tabelas, procedimentos armazenados, funções e triggers utilizados no banco de dados.

Modelagens de entidade relacionamento (MER)

Salão de beleza



Modelagem relacional (MR)



Estrutura e Criação de Tabelas

1. Tabela Cliente

- **Descrição:** Armazena informações dos clientes.
- **Colunas:**
 - email (VARCHAR 45, PK): E-mail do cliente.
 - nome (VARCHAR 60): Nome do cliente.
 - dataNasc (DATE): Data de nascimento do cliente.
 - sexo (CHAR 1): Sexo do cliente.

2. Tabela Funcionário

- **Descrição:** Armazena informações dos funcionários.
- **Colunas:**
 - cpf (VARCHAR 14, PK): CPF do funcionário.
 - nome (VARCHAR 60): Nome do funcionário.
 - email (VARCHAR 45): E-mail do funcionário.
 - estadoCivil (VARCHAR 15): Estado civil do funcionário.
 - especialidade (VARCHAR 60): Especialidade do funcionário.
 - dataNasc (DATE): Data de nascimento do funcionário.
 - sexo (CHAR 1): Sexo do funcionário.
 - salario (DECIMAL 7,2): Salário do funcionário.
 - comissao (DECIMAL 6,2): Comissão do funcionário.
 - dataAdm (DATETIME): Data de admissão do funcionário.
 - dataDem (DATETIME, NULL): Data de demissão do funcionário (se aplicável).

3. Tabela Telefone

- **Descrição:** Armazena os números de telefone dos clientes e funcionários.
- **Colunas:**
 - idTelefone (INT 11, PK): Identificador do telefone.
 - numero (VARCHAR 15): Número de telefone.
 - Cliente_email (VARCHAR 45, FK): E-mail do cliente associado.
 - Funcionario_cpf (VARCHAR 14, FK): CPF do funcionário associado.

4. Tabela EnderecoCli

- **Descrição:** Armazena os endereços dos clientes.
- **Colunas:**
 - uf (CHAR 2): Unidade federativa.
 - cidade (VARCHAR 60): Cidade.
 - bairro (VARCHAR 60): Bairro.
 - rua (VARCHAR 70): Rua.
 - numero (VARCHAR 11): Número.
 - cep (VARCHAR 9): CEP.
 - complemento (VARCHAR 45, NULL): Complemento.
 - Cliente_email (VARCHAR 45, PK, FK): E-mail do cliente associado.

5. Tabela EnderecoFunc

- **Descrição:** Armazena os endereços dos funcionários.
- **Colunas:**
 - uf (CHAR 2): Unidade federativa.
 - cidade (VARCHAR 60): Cidade.
 - bairro (VARCHAR 60): Bairro.
 - rua (VARCHAR 70): Rua.
 - numero (VARCHAR 45): Número.
 - cep (VARCHAR 9): CEP.
 - complemento (VARCHAR 45, NULL): Complemento.
 - Funcionario_CPF (VARCHAR 14, PK, FK): CPF do funcionário associado.

6. Tabela Servico

- **Descrição:** Armazena os serviços oferecidos pelo salão.
- **Colunas:**
 - idServico (INT 11, PK): Identificador do serviço.
 - nome (VARCHAR 60): Nome do serviço.
 - valor (DECIMAL 7,2): Valor do serviço.

7. Tabela Agendamento

- **Descrição:** Armazena os agendamentos de serviços.
- **Colunas:**
 - idAgendamento (INT 11, PK): Identificador do agendamento.
 - status (VARCHAR 45): Status do agendamento.
 - hora (TIME): Hora do agendamento.
 - data (DATE): Data do agendamento.
 - Cliente_email (VARCHAR 45, FK): E-mail do cliente associado.
 - Funcionario_CPF (VARCHAR 14, FK): CPF do funcionário associado.
 - Servico_idServico (INT 11, FK): Identificador do serviço associado.

8. Tabela FormaPag

- **Descrição:** Armazena as formas de pagamento dos agendamentos.
- **Colunas:**
 - idFormaPag (INT 11, PK): Identificador da forma de pagamento.
 - tipo (VARCHAR 45): Tipo de pagamento.
 - valor (DECIMAL 6,2): Valor do pagamento.
 - data (DATETIME): Data do pagamento.
 - Agendamento_idAgendamento (INT 11, FK): Identificador do agendamento associado.

Scripts

CREAT TABLE

-- 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 salaoSenac
-----
```

```
-----
-- Schema salaoSenac
-----
```

```
CREATE SCHEMA IF NOT EXISTS `salaoSenac` DEFAULT CHARACTER SET utf8
;

USE `salaoSenac` ;
```

```
-----
-- Table `salaoSenac`.`Cliente`
-----
```

```
CREATE TABLE IF NOT EXISTS `salaoSenac`.`Cliente` (
  `email` VARCHAR(45) NOT NULL,
```



```

        `nome` VARCHAR(60) NOT NULL,
        `dataNasc` DATE NOT NULL,
        `sexo` CHAR(1) NOT NULL,
        PRIMARY KEY (`email`),
    UNIQUE INDEX `email_UNIQUE` (`email` ASC) INVISIBLE) ENGINE =
        InnoDB;

```

```

-----
-- Table `salaoSenac`.`Funcionario`
-----

```

```

CREATE TABLE IF NOT EXISTS `salaoSenac`.`Funcionario` (
    `cpf` VARCHAR(14) NOT NULL,
    `nome` VARCHAR(60) NOT NULL,
    `email` VARCHAR(45) NOT NULL,
    `estadoCivil` VARCHAR(15) NOT NULL,
    `especialidade` VARCHAR(60) NOT NULL,
    `dataNasc` DATE NOT NULL,
    `sexo` CHAR(1) NOT NULL,
    `salario` DECIMAL(7,2) NOT NULL,
    `comissao` DECIMAL(6,2) NOT NULL,
    `dataAdm` DATETIME NOT NULL,
    `dataDem` DATETIME NULL,
    PRIMARY KEY (`cpf`),
    UNIQUE INDEX `cpf_UNIQUE` (`cpf` ASC) VISIBLE,
    UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE) ENGINE =
        InnoDB;

```

-- Table `salaoSenac`.`Telefone`

```
CREATE TABLE IF NOT EXISTS `salaoSenac`.`Telefone` (  
    `idTelefone` INT(11) NOT NULL,  
    `numero` VARCHAR(15) NOT NULL,  
    `Cliente_email` VARCHAR(45) NULL,  
    `Funcionario_cpf` VARCHAR(14) NULL, PRIMARY  
        KEY (`idTelefone`),  
    INDEX `fk_Telefone_Cliente1_idx` (`Cliente_email` ASC) VISIBLE,  
    INDEX `fk_Telefone_Funcionario1_idx` (`Funcionario_cpf` ASC) VISIBLE,  
    UNIQUE INDEX `numero_UNIQUE` (`numero` ASC) VISIBLE,  
    CONSTRAINT `fk_Telefone_Cliente1`  
        FOREIGN KEY (`Cliente_email`)  
        REFERENCES `salaoSenac`.`Cliente` (`email`)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT `fk_Telefone_Funcionario1`  
        FOREIGN KEY (`Funcionario_cpf`)  
        REFERENCES `salaoSenac`.`Funcionario` (`cpf`)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE) ENGINE =  
    InnoDB;
```

-- Table `salaoSenac`.`EnderecoCli`

```

-----
CREATE TABLE IF NOT EXISTS `salaoSenac`.`EnderecoCli` (
    `uf` CHAR(2) NOT NULL,
    `cidade` VARCHAR(60) NOT NULL,
    `bairro` VARCHAR(60) NOT NULL,
    `rua` VARCHAR(70) NOT NULL,
    `numero` VARCHAR(11) NOT NULL,
    `cep` VARCHAR(9) NOT NULL,
    `complemento` VARCHAR(45) NULL,
    `Cliente_email` VARCHAR(45) NOT NULL, PRIMARY
    KEY (`Cliente_email`),
    CONSTRAINT `fk_Endereco_Cliente`
    FOREIGN KEY (`Cliente_email`)
    REFERENCES `salaoSenac`.`Cliente` (`email`)
    ON DELETE CASCADE
    ON UPDATE CASCADE) ENGINE =
    InnoDB;

```

```

-----
-- Table `salaoSenac`.`Servico`
-----

```

```

CREATE TABLE IF NOT EXISTS `salaoSenac`.`Servico` (
    `idServico` INT(11) NOT NULL,
    `nome` VARCHAR(60) NOT NULL, `valor`
    DECIMAL(7,2) NOT NULL,
    PRIMARY KEY (`idServico`))
    ENGINE = InnoDB;

```

```

-----
-- Table `salaoSenac`.`Agendamento`
-----

CREATE TABLE IF NOT EXISTS `salaoSenac`.`Agendamento` (
  `idAgendamento` INT(11) NOT NULL, `status`
    VARCHAR(45) NOT NULL,
  `hora` TIME NOT NULL,
  `data` DATE NOT NULL,
  `Cliente_email` VARCHAR(45) NOT NULL,
  `Funcionario_CPF` VARCHAR(14) NOT NULL,
  `Servico_idServico` INT(11) NOT NULL, PRIMARY KEY
    (`idAgendamento`),
  INDEX `fk_Agendamento_Cliente1_idx` (`Cliente_email` ASC) VISIBLE,
  INDEX `fk_Agendamento_Funcionario1_idx` (`Funcionario_CPF` ASC) VISIBLE,
  INDEX `fk_Agendamento_Servico1_idx` (`Servico_idServico` ASC) VISIBLE,
  CONSTRAINT `fk_Agendamento_Cliente1`
    FOREIGN KEY (`Cliente_email`)
      REFERENCES `salaoSenac`.`Cliente` (`email`)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
  CONSTRAINT `fk_Agendamento_Funcionario1`
    FOREIGN KEY (`Funcionario_CPF`)
      REFERENCES `salaoSenac`.`Funcionario` (`cpf`)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
  CONSTRAINT `fk_Agendamento_Servico1`

```

```
FOREIGN KEY (`Servico_idServico`)
REFERENCES `salaoSenac`.`Servico` (`idServico`)

ON DELETE CASCADE

ON UPDATE CASCADE) ENGINE =
InnoDB;
```

```
-----
-- Table `salaoSenac`.`FormaPag`
-----
```

```
CREATE TABLE IF NOT EXISTS `salaoSenac`.`FormaPag` (
  `idFormaPag` INT(11) NOT NULL,
  `tipo` VARCHAR(45) NOT NULL,
  `valor` DECIMAL(6,2) NOT NULL,
  `data` DATETIME NOT NULL,
  `Agendamento_idAgendamento` INT(11) NOT NULL, PRIMARY KEY
  (`idFormaPag`),
INDEX `fk_FormaPag_Agendamento1_idx` (`Agendamento_idAgendamento` ASC)
VISIBLE,
CONSTRAINT `fk_FormaPag_Agendamento1`
FOREIGN KEY (`Agendamento_idAgendamento`)
REFERENCES `salaoSenac`.`Agendamento` (`idAgendamento`)
ON DELETE CASCADE
ON UPDATE CASCADE) ENGINE =
InnoDB;
```

```
-----
-- Table `salaoSenac`.`EnderecoFunc`
-----
```

```

-----
CREATE TABLE IF NOT EXISTS `salaoSenac`.`EnderecoFunc` (
    `uf` CHAR(2) NOT NULL,
    `cidade` VARCHAR(60) NOT NULL,
    `bairro` VARCHAR(60) NOT NULL,
    `rua` VARCHAR(70) NOT NULL,
    `numero` VARCHAR(45) NOT NULL,
    `cep` VARCHAR(9) NOT NULL,
    `complemento` VARCHAR(45) NULL,
    `Funcionario_CPF` VARCHAR(14) NOT NULL,
INDEX `fk_EnderecoFunc_Funcionario1_idx` (`Funcionario_CPF` ASC) VISIBLE,
    PRIMARY KEY (`Funcionario_CPF`),
    CONSTRAINT `fk_EnderecoFunc_Funcionario1`
        FOREIGN KEY (`Funcionario_CPF`)
        REFERENCES `salaoSenac`.`Funcionario` (`cpf`)
        ON DELETE CASCADE
        ON UPDATE CASCADE) ENGINE =
    InnoDB;

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

```

Scripts

INSERT

INSERT CLIENTE

```
INSERT INTO cliente (nome, sexo, email, dataNasc)VALUES
    ("Ana Silva", 'F', "ana.silva@gmail.com", "1985-01-15"),
    ("Carlos Souza", 'M', "carlos.souza@gmail.com", "1978-03-22"),
    ("Mariana Lima", 'F', "mariana.lima@gmail.com", "1992-07-30"),
    ("Pedro Oliveira", 'M', "pedro.oliveira@gmail.com", "1980-11-05"),
    ("Fernanda Costa", 'F', "fernanda.costa@gmail.com", "1995-04-18"),
    ("João Pereira", 'M', "joao.pereira@gmail.com", "1983-09-12"),
    ("Beatriz Santos", 'F', "beatriz.santos@gmail.com", "1998-06-25"),
    ("Ricardo Almeida", 'M', "ricardo.almeida@gmail.com", "1975-12-03"),
    ("Juliana Rocha", 'F', "juliana.rocha@gmail.com", "1990-08-14"),
    ("Gustavo Fernandes", 'M', "gustavo.fernandes@gmail.com", "1987-05-19"),
    ("Larissa Ribeiro", 'F', "larissa.ribeiro@gmail.com", "1993-02-27"),
    ("Felipe Castro", 'M', "felipe.castro@gmail.com", "1982-10-21"),
    ("Camila Mendes", 'F', "camila.mendes@gmail.com", "1996-03-08"),
    ("Lucas Barbosa", 'M', "lucas.barbosa@gmail.com", "1989-07-11"),
    ("Patrícia Gomes", 'F', "patricia.gomes@gmail.com", "1984-11-29"),
    ("Rafael Martins", 'M', "rafael.martins@gmail.com", "1991-01-17"),
    ("Isabela Azevedo", 'F', "isabela.azevedo@gmail.com", "1997-09-04"),
    ("Thiago Teixeira", 'M', "thiago.teixeira@gmail.com", "1986-06-23"),
    ("Renata Carvalho", 'F', "renata.carvalho@gmail.com", "1994-12-15"),
    ("André Lima", 'M', "andre.lima@gmail.com", "1981-04-09"),
    ("Gabriela Souza", 'F', "gabriela.souza@gmail.com", "1999-08-30"),
    ("Marcelo Vieira", 'M', "marcelo.vieira@gmail.com", "1979-02-05"),
```

("Tatiana Silva", 'F', "tatiana.silva@gmail.com", "1988-11-20"),
("Rodrigo Almeida", 'M', "rodrigo.almeida@gmail.com", "1990-05-14"),
("Vanessa Costa", 'F', "vanessa.costa@gmail.com", "1992-03-27"),
("Eduardo Santos", 'M', "eduardo.santos@gmail.com", "1985-07-16"),
("Aline Ferreira", 'F', "aline.ferreira@gmail.com", "1993-10-02"),
("Bruno Martins", 'M', "bruno.martins@gmail.com", "1990-02-28"),
("Sofia Oliveira", 'F', "sofia.oliveira@gmail.com", "1991-06-07"),
("Leonardo Rocha", 'M', "leonardo.rocha@gmail.com", "1984-12-19");

Scripts

INSERT ENDERECOCLI

INSERT INTO enderecoCli (cliente_email, uf, cidade, bairro, rua, numero, complemento, cep) VALUES

("ana.silva@gmail.com", "PE", "Recife", "Boa Viagem", "Rua dos Navegantes", 1234, "Apt 203", "51020-020"),

("carlos.souza@gmail.com", "PE", "Olinda", "Casa Caiada", "Av. Gov. Carlos de Lima Cavalcanti", 567, "Casa", "53130-000"),

("mariana.lima@gmail.com", "PE", "Jaboatão dos Guararapes", "Piedade", "Rua Arão Lins de Andrade", 890, "Apt 101", "54410-000"),

("pedro.oliveira@gmail.com", "PE", "Paulista", "Jardim Paulista", "Rua da Aurora", 345, "Casa", "53401-000"),

("fernanda.costa@gmail.com", "PE", "Camaragibe", "Timbi", "Rua do Sol", 678, "Apt 202", "54756-000"),

("joao.pereira@gmail.com", "PE", "São Lourenço da Mata", "Centro", "Rua da Matriz", 123, "Casa", "54735-000"),

("beatriz.santos@gmail.com", "PE", "Abreu e Lima", "Centro", "Rua da Paz", 456, "Apt 303", "53510-000"),

("ricardo.almeida@gmail.com", "PE", "Igarassu", "Centro", "Rua do Comércio", 789, "Casa", "53610-000"),

("juliana.rocha@gmail.com", "PE", "Recife", "Boa Vista", "Rua do Hospício", 101, "Apt 404", "50050-000"),

("gustavo.fernandes@gmail.com", "PE", "Olinda", "Rio Doce", "Av. Brasil", 234, "Casa", "53170-000"),

("larissa.ribeiro@gmail.com", "PE", "Jaboatão dos Guararapes", "Candeias", "Rua Aníbal Varejão", 567, "Apt 505", "54430-000"),

("felipe.castro@gmail.com", "PE", "Paulista", "Paratibe", "Rua do Futuro", 890, "Casa", "53435-000"),

("camila.mendes@gmail.com", "PE", "Camaragibe", "Viana", "Rua da Alegria", 123, "Apt 606", "54768-000"),

("lucas.barbosa@gmail.com", "PE", "São Lourenço da Mata", "Tiúma", "Rua da Saudade", 456, "Casa", "54725-000"),

("patricia.gomes@gmail.com", "PE", "Abreu e Lima", "Caetés", "Rua da Esperança", 789, "Apt 707", "53520-000"),

("rafael.martins@gmail.com", "PE", "Igarassu", "Sítio Histórico", "Rua do Imperador", 101, "Casa", "53645-000"),

("isabela.azevedo@gmail.com", "PE", "Recife", "Graças", "Rua das Creoulas", 234, "Apt 808", "52011-000"),

("thiago.teixeira@gmail.com", "PE", "Olinda", "Bairro Novo", "Av. Getúlio Vargas", 567, "Casa", "53120-000"),

("renata.carvalho@gmail.com", "PE", "Jaboatão dos Guararapes", "Barra de Jangada", "Rua do Atlântico", 890, "Apt 909", "54460-000"),

("andre.lima@gmail.com", "PE", "Paulista", "Maranguape I", "Rua do Sol", 123, "Casa", "53401-000"),

("gabriela.souza@gmail.com", "PE", "Camaragibe", "Tabatinga", "Rua da União", 456, "Apt 1010", "54756-000"),

("marcelo.vieira@gmail.com", "PE", "São Lourenço da Mata", "Muribara", "Rua da Vitória", 789, "Casa", "54735-000"),

("tatiana.silva@gmail.com", "PE", "Abreu e Lima", "Centro", "Rua da Liberdade", 101, "Apt 1111", "53510-000"),

("rodrigo.almeida@gmail.com", "PE", "Igarassu", "Centro", "Rua da Independência", 234, "Casa", "53610-000"),

("vanessa.costa@gmail.com", "PE", "Recife", "Boa Viagem", "Rua dos Navegantes", 345, "Apt 1212", "51020-020"),

("eduardo.santos@gmail.com", "PE", "Olinda", "Casa Caiada",
"Av. Gov. Carlos de Lima Cavalcanti", 567, "Casa", "53130-000"),
("aline.ferreira@gmail.com", "PE", "Jaboatão dos Guararapes", "Piedade",
"Rua Arão Lins de Andrade", 890, "Apt 1313", "54410-000"),
("bruno.martins@gmail.com", "PE", "Paulista", "Jardim Paulista", "Rua da
Aurora", 123, "Casa", "53401-000"),
("sofia.oliveira@gmail.com", "PE", "Camaragibe", "Timbi", "Rua do Sol", 456,
"Apt 1414", "54756-000"),
("leonardo.rocha@gmail.com", "PE", "São Lourenço da Mata", "Centro", "Rua da
Matriz", 789, "Casa", "54735-000");

Scripts

INSERT TELEFONE

INSERT INTO Telefone (idTelefone, numero, cliente_email) VALUES

(1, "(81) 91234-5678", "ana.silva@gmail.com"),
(2, "(81) 91234-5679", "carlos.souza@gmail.com"),
(3, "(81) 91234-5680", "mariana.lima@gmail.com"),
(4, "(81) 91234-5681", "pedro.oliveira@gmail.com"),
(5, "(81) 91234-5682", "fernanda.costa@gmail.com"),
(6, "(81) 91234-5683", "joao.pereira@gmail.com"),
(7, "(81) 91234-5684", "beatriz.santos@gmail.com"),
(8, "(81) 91234-5685", "ricardo.almeida@gmail.com"),
(9, "(81) 91234-5686", "juliana.rocha@gmail.com"),
(10, "(81) 91234-5687", "gustavo.fernandes@gmail.com"),
(11, "(81) 91234-5688", "larissa.ribeiro@gmail.com"),
(12, "(81) 91234-5689", "felipe.castro@gmail.com"),
(13, "(81) 91234-5690", "camila.mendes@gmail.com"),
(14, "(81) 91234-5691", "lucas.barbosa@gmail.com"),
(15, "(81) 91234-5692", "patricia.gomes@gmail.com"),

(16, "(81) 91234-5693", "rafael.martins@gmail.com"),
(17, "(81) 91234-5694", "isabela.azevedo@gmail.com"),
(18, "(81) 91234-5695", "thiago.teixeira@gmail.com"),
(19, "(81) 91234-5696", "renata.carvalho@gmail.com"),
(20, "(81) 91234-5697", "andre.lima@gmail.com"),
(21, "(81) 91234-5698", "gabriela.souza@gmail.com"),
(22, "(81) 91234-5699", "marcelo.vieira@gmail.com"),
(23, "(81) 91234-5700", "tatiana.silva@gmail.com"),
(24, "(81) 91234-5701", "rodrigo.almeida@gmail.com"),
(25, "(81) 91234-5702", "vanessa.costa@gmail.com"),
(26, "(81) 91234-5703", "eduardo.santos@gmail.com"),
(27, "(81) 91234-5704", "aline.ferreira@gmail.com"),
(28, "(81) 91234-5705", "bruno.martins@gmail.com"),
(29, "(81) 91234-5706", "sofia.oliveira@gmail.com"),
(30, "(81) 91234-5707", "leonardo.rocha@gmail.com");

Scripts

INSERT FUNCIONARIO

```
INSERT INTO funcionario (cpf, nome, email, sexo, estadoCivil, especialidade,
dataNasc, salario, comissao, dataAdm, dataDem) VALUES

('23456789012', 'José Oliveira', 'jose.oliveira@empresa.com', 'M', 'Casado',
'Cabeleireiro', '1990-03-15', 2500.00, 500.00, '2020-03-15', NULL),

('34567890123', 'Maria Souza', 'maria.souza@empresa.com', 'F', 'Solteira',
'Manicure', '1985-08-10', 1800.00, 300.00, '2019-07-25', NULL),

('45678901234', 'Paulo Santos', 'paulo.santos@empresa.com', 'M', 'Divorciado',
'Barbeiro', '1982-11-22', 2700.00, 400.00, '2018-05-20', NULL),

('56789012345', 'Juliana Almeida', 'juliana.almeida@empresa.com', 'F',
'Casada', 'Esteticista', '1992-06-30', 2200.00, 350.00, '2021-02-15', NULL),
```

('67890123456', 'Carlos Pereira', 'carlos.pereira@empresa.com', 'M', 'Solteiro',
'Cabeleireiro', '1994-07-05', 2400.00, 450.00, '2022-01-10', NULL),

('78901234567', 'Luciana Costa', 'luciana.costa@empresa.com', 'F', 'Casada',
'Recepcionista', '1989-10-19', 1500.00, 200.00, '2017-08-25', NULL),

('89012345678', 'Eduardo Gomes', 'eduardo.gomes@empresa.com', 'M',
'Divorciado', 'Cabeleireiro', '1980-04-11', 2900.00, 550.00, '2016-03-05', NULL),
('90123456789', 'Renata Lima', 'renata.lima@empresa.com', 'F', 'Solteira',
'Manicure', '1991-12-10', 2000.00, 300.00, '2019-06-01', NULL),

('12345678901', 'Lucas Martins', 'lucas.martins@empresa.com', 'M', 'Casado',
'Cabeleireiro', '1993-02-12', 2600.00, 450.00, '2021-04-15', NULL),

('23456789014', 'Tatiane Rocha', 'tatiane.rocha@empresa.com', 'F', 'Solteira',
'Pedicure', '1990-01-30', 1900.00, 300.00, '2020-07-20', NULL),

('34567890125', 'Ricardo Alves', 'ricardo.alves@empresa.com', 'M', 'Casado',
'Gerente de Salão', '1984-09-25', 3500.00, 600.00, '2017-11-18', NULL),

('45678901236', 'Fernanda Silva', 'fernanda.silva@empresa.com', 'F', 'Solteira',
'Cabeleireiro', '1995-05-20', 2300.00, 400.00, '2021-03-10', NULL),

('56789012346', 'Marcos Costa', 'marcos.costa@empresa.com', 'M', 'Casado',
'Barbeiro', '1987-08-25', 2700.00, 450.00, '2019-04-10', NULL),

('67890123457', 'Amanda Lima', 'amanda.lima@empresa.com', 'F', 'Divorciada',
'Recepcionista', '1992-11-30', 1600.00, 200.00, '2020-09-10', NULL),

('78901234568', 'Vinícius Rocha', 'vinicius.rocha@empresa.com', 'M', 'Casado',
'Cabeleireiro', '1983-06-15', 2500.00, 500.00, '2018-10-05', NULL),

('23456789015', 'Gustavo Souza', 'gustavo.souza@empresa.com', 'M', 'Casado',
'Cabeleireiro', '1987-01-20', 2400.00, 450.00, '2015-11-01', '2023-05-15'),

('34567890126', 'Isabela Martins', 'isabela.martins@empresa.com', 'F',
'Solteira', 'Manicure', '1994-08-12', 1900.00, 300.00, '2019-06-01', '2023-07-30'),

('45678901237', 'André Pereira', 'andre.pereira@empresa.com', 'M', 'Divorciado',
'Gerente de Salão', '1983-03-09', 3200.00, 550.00, '2016-12-18', '2024-01-10'),

('56789012348', 'Juliana Souza', 'juliana.souza@empresa.com', 'F', 'Casada',
'Esteticista', '1989-04-14', 2100.00, 350.00, '2018-08-22', '2023-11-10'),

('67890123459', 'Leonardo Lima', 'leonardo.lima@empresa.com', 'M', 'Casado',
'Barbeiro', '1991-11-05', 2700.00, 450.00, '2019-04-14', '2024-02-15'),

('78901234570', 'Aline Costa', 'aline.costa@empresa.com', 'F', 'Solteira',
'Cabeleireiro', '1992-05-30', 2300.00, 400.00, '2020-10-20', '2023-08-01'),

('89012345681', 'Thiago Rocha', 'thiago.rocha@empresa.com', 'M', 'Divorciado',
 'Manicure', '1988-03-12', 1800.00, 300.00, '2017-05-05', '2023-06-20'),
 ('90123456782', 'Carolina Almeida', 'carolina.almeida@empresa.com', 'F',
 'Casada', 'Recepcionista', '1986-09-19', 1500.00, 200.00, '2018-06-03', '2023-12-15'),
 ('12345678902', 'Rafael Oliveira', 'rafael.oliveira@empresa.com', 'M', 'Solteiro',
 'Cabeleireiro', '1993-12-25', 2500.00, 450.00, '2020-01-12', '2023-09-10'),
 ('23456789016', 'Marília Gomes', 'marilia.gomes@empresa.com', 'F', 'Solteira',
 'Manicure', '1995-04-05', 1900.00, 300.00, '2021-02-25', '2023-12-01'),
 ('34567890127', 'Ricardo Costa', 'ricardo.costa@empresa.com', 'M',
 'Divorciado', 'Cabeleireiro', '1984-07-10', 2400.00, 400.00, '2016-01-11', '2023-
 1105'),
 ('45678901238', 'Patrícia Lima', 'patricia.lima@empresa.com', 'F', 'Casada',
 'Esteticista', '1990-10-22', 2100.00, 350.00, '2017-11-17', '2023-08-15'),
 ('56789012349', 'José Rocha', 'jose.rocha@empresa.com', 'M',
 'Solteiragendamentoo', 'Barbeiro', '1982-06-30', 2700.00, 450.00, '2016-12-10',
 '2024-02-05');

Scripts

INSERT ENDERECOFUNC

INSERT INTO enderecoFunc (funcionario_cpf, uf, cidade, bairro, rua, numero,
 complemento, cep) VALUES

('23456789012', 'PE', 'Recife', 'Boa Viagem', 'Rua A', '123', 'Apto 101', '51020-001'),
 ('34567890123', 'PE', 'Recife', 'Espinheiro', 'Rua B', '456', 'Bloco 3', '52010-002'),
 ('45678901234', 'PE', 'Olinda', 'Casa Caiada', 'Rua C', '789', 'Apto 202', '53100003'),
 ('56789012345', 'PE', 'Jaboatão dos Guararapes', 'Piedade', 'Rua D', '101', 'Casa',
 '54300-004'),
 ('67890123456', 'PE', 'Recife', 'Centro', 'Rua E', '202', 'Apto 303', '50010-005'),
 ('78901234567', 'PE', 'Olinda', 'Varadouro', 'Rua F', '303', 'Casa', '53120-006'),
 ('89012345678', 'PE', 'Recife', 'Graças', 'Rua G', '404', 'Apto 404', '52011-007'),
 ('90123456789', 'PE', 'Recife', 'Boa Vista', 'Rua H', '505', 'Bloco A', '50060-008'),

('12345678901', 'PE', 'Camaragibe', 'Centro', 'Rua I', '606', 'Apto 505', '54760-009'),
('23456789014', 'PE', 'Jaboatão dos Guararapes', 'Cavaleiro', 'Rua J', '707', 'Casa',
'54320-010'),
('34567890125', 'PE', 'Recife', 'Campo Grande', 'Rua K', '808', 'Bloco B', '50090011'),
('45678901236', 'PE', 'Recife', 'San Martin', 'Rua L', '909', 'Apto 707', '50730-012'),
('56789012346', 'PE', 'Jaboatão dos Guararapes', 'Piedade', 'Rua M', '1010', 'Casa',
'54310-013'),
('67890123457', 'PE', 'Olinda', 'Carmo', 'Rua N', '1111', 'Apto 808', '53140-014'),
('78901234568', 'PE', 'Recife', 'Pina', 'Rua O', '1212', 'Apto 909', '51200-015'),
('23456789015', 'PE', 'Recife', 'Boa Viagem', 'Rua P', '1313', 'Casa', '51020-016'),
('34567890126', 'PE', 'Olinda', 'Peixinhos', 'Rua Q', '1414', 'Bloco C', '53130-017'),
('45678901237', 'PE', 'Camaragibe', 'Centro', 'Rua R', '1515', 'Apto 202',
'54760018'),
('56789012348', 'PE', 'Jaboatão dos Guararapes', 'Prazeres', 'Rua S', '1616', 'Casa',
'54400-019'),
('67890123459', 'PE', 'Recife', 'Aflitos', 'Rua T', '1717', 'Bloco D', '52020-020'),
('78901234570', 'PE', 'Olinda', 'Fazenda Nova', 'Rua U', '1818', 'Apto 303',
'53110021'),
('89012345681', 'PE', 'Recife', 'Jardim São Paulo', 'Rua V', '1919', 'Casa',
'52040022'),
('90123456782', 'PE', 'Olinda', 'Iputinga', 'Rua W', '2020', 'Apto 404', '53150-023'),
('12345678902', 'PE', 'Recife', 'Iputinga', 'Rua X', '2121', 'Bloco E', '50760-024'),
('23456789016', 'PE', 'Jaboatão dos Guararapes', 'Piedade', 'Rua Y', '2222', 'Casa',
'54330-025'),
('34567890127', 'PE', 'Recife', 'Bairro do Recife', 'Rua Z', '2323', 'Apto 505',
'50050026'),
('45678901238', 'PE', 'Olinda', 'Casa Caiada', 'Rua AA', '2424', 'Casa', '53110-027'),
('56789012349', 'PE', 'Recife', 'San Martin', 'Rua AB', '2525', 'Bloco F', '50720-028');

Scripts

INSERT TELEFONE FUNC

INSERT INTO Telefone (idTelefone, numero, funcionario_cpf) VALUES

(31, '81987654321', '23456789012'),
(32, '81998765432', '34567890123'),
(33, '81999876543', '45678901234'),
(34, '81987654322', '56789012345'),
(35, '81976543210', '67890123456'),
(36, '81965432109', '78901234567'),
(37, '81954321098', '89012345678'),
(38, '81943210987', '90123456789'),
(39, '81932109876', '12345678901'),
(40, '81921098765', '23456789014'),
(41, '81910987654', '34567890125'),
(42, '81909876543', '45678901236'),
(43, '81998765431', '56789012346'),
(44, '81987654310', '67890123457'),
(45, '81976543209', '78901234568'),
(46, '81965432108', '23456789015'),
(47, '81954321097', '34567890126'),
(48, '81943210976', '45678901237'),
(49, '81932109865', '56789012348'),
(50, '81921098754', '67890123459'),
(51, '81910987643', '78901234570'),
(52, '81909876532', '89012345681'),
(53, '81998765420', '90123456782'),
(54, '81987654300', '12345678902'),

(55, '81976543201', '23456789016'),
(56, '81965432102', '34567890127'),
(57, '81954321003', '45678901238'),
(58, '81943210910', '56789012349');

Scripts

INSERT SERVICOS

INSERT INTO servico (idServico, nome, valor) VALUES

(1, 'Corte de Cabelo Masculino', 50.00),
(2, 'Corte de Cabelo Feminino', 70.00),
(3, 'Barba', 40.00),
(4, 'Coloração', 150.00),
(5, 'Luzes/Mechas', 200.00),
(6, 'Escova', 60.00),
(7, 'Hidratação Capilar', 80.00),
(8, 'Progressiva', 250.00),
(9, 'Penteado para Eventos', 120.00),
(10, 'Manicure', 30.00),
(11, 'Pedicure', 35.00),
(12, 'Alongamento de Unhas', 100.00),
(13, 'Depilação Facial', 50.00),
(14, 'Depilação Corporal', 100.00),
(15, 'Maquiagem para Eventos', 150.00),
(16, 'Design de Sobrancelhas', 40.00),
(17, 'Massagem Relaxante', 120.00),
(18, 'Tratamento Anti-queda', 90.00),
(19, 'Botox Capilar', 180.00),

(20, 'Cauterização Capilar', 100.00);

Scripts

INSERT AGENDAMNETO

INSERT INTO agendamento (idAgendamento, status, hora, data,
Cliente_email, Funcionario_CPF, Servico_idServico) VALUES

(1, 'Confirmado', '10:00:00', '2024-12-07', 'ana.silva@gmail.com', '23456789012',
1),

(2, 'Confirmado', '11:00:00', '2024-12-07', 'carlos.souza@gmail.com',
'34567890123', 2),

(3, 'Confirmado', '12:00:00', '2024-12-07', 'mariana.lima@gmail.com',
'45678901234', 3),

(4, 'Confirmado', '13:00:00', '2024-12-07', 'pedro.oliveira@gmail.com',
'56789012345', 4),

(5, 'Confirmado', '14:00:00', '2024-12-07', 'fernanda.costa@gmail.com',
'67890123456', 5),

(6, 'Confirmado', '15:00:00', '2024-12-07', 'joao.pereira@gmail.com',
'78901234567', 6),

(7, 'Confirmado', '16:00:00', '2024-12-07', 'beatriz.santos@gmail.com',
'89012345678', 7),

(8, 'Confirmado', '17:00:00', '2024-12-07', 'ricardo.almeida@gmail.com',
'90123456789', 8),

(9, 'Confirmado', '18:00:00', '2024-12-07', 'juliana.rocha@gmail.com',
'12345678901', 9),

(10, 'Confirmado', '10:00:00', '2024-12-08', 'gustavo.fernandes@gmail.com',
'23456789014', 10),

(11, 'Confirmado', '11:00:00', '2024-12-08', 'larissa.ribeiro@gmail.com',
'34567890125', 11),

(12, 'Confirmado', '12:00:00', '2024-12-08', 'felipe.castro@gmail.com',
'45678901236', 12),

(13, 'Confirmado', '13:00:00', '2024-12-08', 'camila.mendes@gmail.com',

'56789012346', 13),
(14, 'Confirmado', '14:00:00', '2024-12-08', 'lucas.barbosa@gmail.com',
'67890123457', 14),
(15, 'Confirmado', '15:00:00', '2024-12-08', 'patricia.gomes@gmail.com',
'78901234568', 15),
(16, 'Confirmado', '16:00:00', '2024-12-08', 'rafael.martins@gmail.com',
'23456789015', 16),
(17, 'Confirmado', '17:00:00', '2024-12-08', 'isabela.azevedo@gmail.com',
'34567890126', 17),
(18, 'Confirmado', '18:00:00', '2024-12-08', 'thiago.teixeira@gmail.com',
'45678901237', 18),
(19, 'Confirmado', '10:00:00', '2024-12-09', 'renata.carvalho@gmail.com',
'56789012348', 19),
(20, 'Confirmado', '11:00:00', '2024-12-09', 'andre.lima@gmail.com',
'67890123459', 20),
(21, 'Confirmado', '12:00:00', '2024-12-09', 'gabriela.souza@gmail.com',
'78901234570', 1),
(22, 'Confirmado', '13:00:00', '2024-12-09', 'marcelo.vieira@gmail.com',
'89012345681', 2),
(23, 'Confirmado', '14:00:00', '2024-12-09', 'tatiana.silva@gmail.com',
'90123456782', 3),
(24, 'Confirmado', '15:00:00', '2024-12-09', 'rodrigo.almeida@gmail.com',
'12345678902', 4),
(25, 'Confirmado', '16:00:00', '2024-12-09', 'vanessa.costa@gmail.com',
'23456789016', 5),
(26, 'Confirmado', '17:00:00', '2024-12-09', 'eduardo.santos@gmail.com',
'34567890127', 6),
(27, 'Confirmado', '18:00:00', '2024-12-09', 'aline.ferreira@gmail.com',
'45678901238', 7),
(28, 'Confirmado', '10:00:00', '2024-12-10', 'bruno.martins@gmail.com',
'56789012349', 8),
(29, 'Confirmado', '11:00:00', '2024-12-10', 'sofia.oliveira@gmail.com',
'67890123456', 9),

**(30, 'Confirmado', '12:00:00', '2024-12-10', 'leonardo.rocha@gmail.com',
'78901234567', 10);**

Scripts

INSERT FORMAPAG

**INSERT INTO formapag (idFormaPag, tipo, valor, data,
Agendamento_idAgendamento) VALUES**

(1, 'Pix', 50.00, '2024-12-07 10:00:00', 1),

(2, 'Cartão de Crédito', 70.00, '2024-12-07 11:00:00', 2), (3,

'Cartão de Débito', 40.00, '2024-12-07 12:00:00', 3),

(4, 'Pix', 150.00, '2024-12-07 13:00:00', 4),

(5, 'Cartão de Crédito', 200.00, '2024-12-07 14:00:00', 5),

(6, 'Cartão de Débito', 60.00, '2024-12-07 15:00:00', 6),

(7, 'Pix', 80.00, '2024-12-07 16:00:00', 7),

(8, 'Cartão de Crédito', 250.00, '2024-12-07 17:00:00', 8), (9,

'Cartão de Débito', 120.00, '2024-12-07 18:00:00', 9),

(10, 'Pix', 30.00, '2024-12-08 10:00:00', 10),

(11, 'Cartão de Crédito', 35.00, '2024-12-08 11:00:00', 11),

(12, 'Cartão de Débito', 100.00, '2024-12-08 12:00:00', 12),

(13, 'Pix', 50.00, '2024-12-08 13:00:00', 13),

(14, 'Cartão de Crédito', 100.00, '2024-12-08 14:00:00', 14), (15,

'Cartão de Débito', 150.00, '2024-12-08 15:00:00', 15),

(16, 'Pix', 40.00, '2024-12-08 16:00:00', 16),

(17, 'Cartão de Crédito', 120.00, '2024-12-08 17:00:00', 17), (18,

'Cartão de Débito', 90.00, '2024-12-08 18:00:00', 18),

(19, 'Pix', 180.00, '2024-12-09 10:00:00', 19),

(20, 'Cartão de Crédito', 100.00, '2024-12-09 11:00:00', 20), (21,

'Cartão de Débito', 50.00, '2024-12-09 12:00:00', 21),

(22, 'Pix', 70.00, '2024-12-09 13:00:00', 22),
 (23, 'Cartão de Crédito', 40.00, '2024-12-09 14:00:00', 23),
 (24, 'Cartão de Débito', 150.00, '2024-12-09 15:00:00', 24),
 (25, 'Pix', 200.00, '2024-12-09 16:00:00', 25),
 (26, 'Cartão de Crédito', 60.00, '2024-12-09 17:00:00', 26), (27,
 'Cartão de Débito', 80.00, '2024-12-09 18:00:00', 27),
 (28, 'Pix', 250.00, '2024-12-10 10:00:00', 28),
 (29, 'Cartão de Crédito', 120.00, '2024-12-10 11:00:00', 29), (30,
 'Cartão de Débito', 30.00, '2024-12-10 12:00:00', 30); Scripts

20 Consultas/Perguntas/Relatórios (DQL)

-- 1.TODOS OS PROFICIONAIS.

```

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
    "Estado Civil",

    date_format(func.dataNasc, '%d/%m/%Y') "Data de
        Nascimento",

    func.email "E-mail", func.especialidade "Especialidade",

    concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",

    concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",

    date_format(func.dataAdm, "%h:%i %d/%m/%Y")
        "Data de Admissão",

    coalesce(date_format(func.dataDem, "%h: %i %d/%m?%Y"), "Ativo") "Data de
        Demissão",

    endF.cidade "Cidade", tel.numero "Telefone"

from funcionario func

    inner join enderecofunc endF on func.cpf =
        endF.Funcionario_cpf
  
```

```
left join telefone tel on tel.Funcionario_cpf = func.cpf  
  
order by nome;
```

-- 2.LISTA DE PROFISSIONAIS ATIVOS NO SALÃO SENAC.

```
select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil  
        "Estado Civil",  
  
        date_format(func.dataNasc, '%d/%m/%Y') "Data de  
        Nascimento",  
  
        func.email "E-mail", func.especialidade "Especialidade",  
  
        concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",  
  
        concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",  
  
        date_format(func.dataAdm, "%h:%i %d/%m/%Y")  
        "Data de Admissão",  
  
        coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de  
        Demissão",  
  
        endF.cidade "Cidade" from  
  
funcionario func  
  
        inner join enderecofunc endF on func.cpf =  
        endF.Funcionario_cpf  
  
        where func.dataDem is null  
  
        order by func.nome;
```

-- 3.RELAÇÃO DE TODOS OS FUNCIONÁRIOS DE SEXO MASCULINO DO SALÃO SENAC.

```
select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil  
        "Estado Civil",
```

```

        date_format(func.dataNasc, '%d/%m/%Y') "Data de
            Nascimento",
        func.email "E-mail", func.especialidade "Especialidade",
        concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",
        concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",
        date_format(func.dataAdm, "%h:%i %d/%m/%Y")
            "Data de Admissão",
        coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de
            Demissão",
        endF.cidade "Cidade"    from
        funcionario func
        inner join enderecofunc
        endF on func.cpf =
        endF.Funcionario_cpf
        where func.sexo = 'M'
        order by func.nome;

```

-- 4.RELAÇÃO DE TODOS OS FUNCIONÁRIOS DE SEXO FEMININO DO SALÃO SENAC.

```

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
    "Estado Civil",
        date_format(func.dataNasc, '%d/%m/%Y') "Data de
            Nascimento",
        func.email "E-mail", func.especialidade "Especialidade",
        concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",
        concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",
        date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
            de Admissão",

```

```

coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de
Demissão",

endF.cidade "Cidade" from

funcionario func

inner join enderecofunc endF on func.cpf =
endF.Funcionario_cpf

where func.sexo = 'F'

order by func.nome;

```

-- 5.RELAÇÃO DE TODOS OS FUNCIONÁRIOS DE OLINDA/PE.

```

select funcionario_cpf from enderecofunc where

cidade like "Olinda";

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
"Estado Civil",

date_format(func.dataNasc, '%d/%m/%Y') "Data de
Nascimento",

func.email "E-mail", func.especialidade "Especialidade",

concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",

concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",

date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
de Admissão",

coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de
Demissão", endF.cidade "Cidade" from funcionario
func

inner join enderecofunc endF on func.cpf =
endF.Funcionario_cpf

```

```
where endF.cidade like "Olinda" order  
by func.nome;
```

-- 6.RELAÇÃO DE TODOS OS FUNCIONÁRIOS DE RECIFE/PE

```
select funcionario_cpf from enderecofunc where  
cidade like "Recife";  
  
select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil  
"Estado Civil",  
date_format(func.dataNasc, '%d/%m/%Y') "Data de  
Nascimento",  
func.email "E-mail", func.especialidade "Especialidade",  
concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",  
concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",  
date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data  
de Admissão",  
coalesce(date_format(func.dataDem, "%h: %i %d/%m%Y"), "Ativo") "Data de  
Demissão", endF.cidade "Cidade" from funcionario  
func  
inner join enderecofunc endF on func.cpf =  
endF.Funcionario_cpf  
where endF.cidade like "Recife" order  
by func.nome;
```

-- 7.LISTA DE RELAÇÃO DE FUNCIONÁRIOS DEMITIDOS ENTRE 2016 A 2024.


```

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
        "Estado Civil",

        date_format(func.dataNasc, '%d/%m/%Y') "Data de
        Nascimento",

        func.email "E-mail", func.especialidade "Especialidade",

        concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",

        concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",

        date_format(func.dataDem, "%h:%i %d/%m/%Y") "Data
        de Admissão",

        date_format(func.dataDem, "%h: %i %d/%m%Y") "Data de Demissão",

        endF.cidade "Cidade"   from funcionario func

        inner join enderecofunc endF on func.cpf =
        endF.Funcionario_cpf

        where func.dataDem between '2016-01-01'
        and '2024-12-31'

        order by func.dataDem;

```

-- 8. LISTA DE FUNCIONÉRIOS ADMITIDOS ENTRE 2015 A 2024.

```

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
        "Estado Civil",

        date_format(func.dataNasc, '%d/%m/%Y') "Data de
        Nascimento",

        func.email "E-mail", func.especialidade "Especialidade",

        concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",

        concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",

        date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
        de Admissão",

        date_format(func.dataDem, "%h: %i %d/%m%Y") "Data de Demissão",

```

```

        endF.cidade "Cidade"    from

funcionario func

        inner join enderecofunc endF on func.cpf =
        endF.Funcionario_cpf

        where func.dataDem between '2015-10-01'
        and '2024-12-31'

        order by func.dataDem;

```

-- 9. LISTA DE FUNCIONÁRIOS QUE GANHAM IGUAL OU MENOR QUÊ R\$ 2.000,00.

```

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
        "Estado Civil",

        date_format(func.dataNasc, '%d/%m/%Y') "Data de
        Nascimento",

        func.email "E-mail", func.especialidade "Especialidade",

        concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",

        concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",

        date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
        de Admissão",

coalesce(date_format(func.dataDem, "%h: %i %d/%m%Y"), "Ativo") "Data de
        Demissão",    endF.cidade "Cidade"    from funcionario
func

        inner join enderecofunc endF on func.cpf =
        endF.Funcionario_cpf

        where salario <= 2000

        order by func.nome;

```

-- 10. LISTA DE FUNCIONÁRIOS QUE GANHAM UM SALÁRIO MAIOR QUÊ R\$

2.000,00.

```
select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
      "Estado Civil",
      date_format(func.dataNasc, '%d/%m/%Y') "Data de
      Nascimento",
      func.email "E-mail", func.especialidade "Especialidade",
      concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",
      concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",
      date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
      de Admissão",
      coalesce(date_format(func.dataDem, "%h: %i %d/%m%Y"), "Ativo") "Data de
      Demissão", endF.cidade "Cidade" from funcionario
      func
      inner join enderecofunc endF on func.cpf =
      endF.Funcionario_cpf
      where salario >= 2001
      order by func.nome;
```

-- 11. TIPO DE PAGAMENTO E VALOR TOTAL PAGO.

```
select tipo as "Tipo de Pagamento",
      SUM(valor) "Total Pago"
      from formapag
      group by tipo
      order by SUM(valor) desc;
```

-- 12. MÉDIA SALARIAL DOS FUNCIONÁRIOS.

```
select avg(salario) from funcionario;

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero",
       func.estadoCivil "Estado Civil",

       date_format(func.dataNasc, '%d/%m/%Y') "Data de
       Nascimento",

       func.email "E-mail", func.especialidade "Especialidade",

       concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",

       concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",

       date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
       de Admissão",

coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de
       Demissão", endF.cidade "Cidade" from funcionario
func

inner join enderecofunc endF on func.cpf =
endF.Funcionario_cpf

where salario > (select avg(salario) from
funcionario)

order by salario desc;
```

-- 13. LISTA DOS FUNCIONÁRIOS DE SEXO 'M' SOLTEIROS.

```
select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
       "Estado Civil",

       date_format(func.dataNasc, '%d/%m/%Y') "Data de
       Nascimento",

       func.email "E-mail", func.especialidade "Especialidade",
```

```

concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",
concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",
date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
de Admissão",
coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de
Demissão",
endF.cidade "Cidade" from
funcionario func
inner join enderecofunc endF on func.cpf =
endF.Funcionario_cpf
where estadoCivil = "Solteiro"
order by nome;

```

-- 14. LISTA DOS FUNCIONÁRIOS DE SEXO 'F' SOLTEIRAS.

```

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
"Estado Civil",
date_format(func.dataNasc, '%d/%m/%Y') "Data de
Nascimento",
func.email "E-mail", func.especialidade "Especialidade",
concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",
concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",
date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
de Admissão",
coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de
Demissão", endF.cidade "Cidade" from funcionario
func
inner join enderecofunc endF on func.cpf =
endF.Funcionario_cpf

```

```
where estadoCivil = "Solteira"

order by nome;
```

-- 15. LISTA DE AGENDAMENTOS.

```
select ag.status, ag.hora, ag.data, cli.nome
      "Nome do Cliente", cli.email "Email do
      Cliente",
      tel.numero "Telefone do Cliente",
      func.nome "Nome do Funcionário", srv.nome
      "Tipo de Serviço",
concat('R$ ', format(srv.valor, 2, 'de_DE')) "Valor do Serviço",
      fp.tipo AS "Forma de Pagamento",
concat('R$ ', format(fp.valor, 2, 'de_DE')) "Valor Pago" from
      agendamento ag
inner join cliente cli on ag.Cliente_email = cli.email inner
      join telefone tel on cli.email =
      tel.Cliente_email
      inner join funcionario func on ag.Funcionario_CPF =
      func.cpf
      inner join servico srv on ag.Servico_idServico =
      srv.idServico
inner join formapag fp on ag.idAgendamento =
      fp.Agendamento_idAgendamento
      order by ag.data, ag.hora;
```

-- 16. LISTA DE CLIENTES SALÃO SENAC

```
select cli.email AS "Email", cli.nome "Nome", cli.sexo "Gênero",  
DATE_FORMAT(cli.dataNasc, '%d/%m/%Y') AS "Data de Nascimento",  
endF.cidade as "Cidade",  
tel.numero as "Telefone"  
  
from cliente cli  
  
inner join endereccli endF on cli.email =  
endF.Cliente_email  
  
left join telefone tel on tel.Cliente_email = cli.email  
  
order by cli.nome;
```

-- 17. LISTA COM OS CLIENTES DE SEXO MASCULINO.

```
select cli.email AS "Email", cli.nome "Nome", cli.sexo "Gênero",  
DATE_FORMAT(cli.dataNasc, '%d/%m/%Y') AS "Data de Nascimento",  
endF.cidade as "Cidade", tel.numero  
as "Telefone"  
  
from cliente cli  
  
inner join endereccli endF on cli.email =  
endF.Cliente_email  
  
left join telefone tel on tel.Cliente_email = cli.email  
  
where cli.sexo = 'M'  
  
order by cli.nome;
```

-- 18. LISTA COM OS CLIENTES DE SEXO FEMININO

```

select cli.email AS "Email", cli.nome "Nome", cli.sexo "Gênero",
DATE_FORMAT(cli.dataNasc, '%d/%m/%Y') AS "Data de Nascimento",
        endF.cidade as "Cidade",    tel.numero
        as "Telefone"
        from cliente cli
        inner join endereccli endF on cli.email =
        endF.Cliente_email
left join telefone tel on tel.Cliente_email = cli.email
where cli.sexo = 'F'
        order by cli.nome;

```

-- 19. VALOR TOTAL ARRECADADO POR FUNCIONÁRIO POR MÊS.

```

select func.nome as "Nome do Funcionário",
        DATE_FORMAT(ag.data, '%Y-%m') "Mês",
        SUM(srv.valor) "Valor Total Arrecadado" from
agendamento ag
        inner join funcionario func on ag.Funcionario_CPF =
        func.cpf
        inner join servico srv on ag.Servico_idServico =
        srv.idServico
        group by func.nome, DATE_FORMAT(ag.data,
        '%Y-%m');

```

-- 20. VALOR TOTAL ARRECADADO POR CLIENTE.


```

select cli.nome AS "Nome do Cliente", cli.email "Email do Cliente",
        SUM(fp.valor) "Valor Total Pago" from agendamento
        ag
        inner join cliente cli on ag.Cliente_email = cli.email
        inner join formapag fp on ag.idAgendamento =
        fp.Agendamento_idAgendamento
        group by cli.nome, cli.email
        order by "Valor Total Pago" desc;

```

Scripts

10 VIEWS

-- 1. funcSexoMascAtivo

```

create view funcSexoMascAtivo as

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
        "Estado Civil",
        date_format(func.dataNasc, '%d/%m/%Y') "Data de
        Nascimento",
        func.email "E-mail", func.especialidade "Especialidade",
        concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",
        concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",
        date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
        de Admissão",
        coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de
        Demissão",
        endF.cidade "Cidade" from
        funcionario func
        inner join enderecofunc endF on func.cpf =

```

endF.Funcionario_cpf

**where func.sexo = 'M' and func.dataDem is
null**

order by func.nome;

-- 2. view funcSexoFemAtivo

create view funcSexoFemAtivo as

**select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
"Estado Civil",**

**date_format(func.dataNasc, '%d/%m/%Y') "Data de
Nascimento",**

func.email "E-mail", func.especialidade "Especialidade",

concat("R\$ ", format(func.salario, 2, 'de_DE')) "Salário",

concat("R\$ ", format(func.comissao, 2, 'de_DE')) "Comissão",

**date_format(func.dataAdm, "%h:%i %d/%m/%Y")
"Data de Admissão",**

**coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de
Demissão",**

endF.cidade "Cidade" from

funcionario func

**inner join enderecofunc endF on func.cpf =
endF.Funcionario_cpf**

**where func.sexo = 'F' and func.dataDem is
null**

order by func.nome;

-- 3. view FuncAtivoOlinda

create view FuncAtivoOlinda as

```
select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil  
      "Estado Civil",  
      date_format(func.dataNasc, '%d/%m/%Y') "Data de  
      Nascimento",  
      func.email "E-mail", func.especialidade "Especialidade",  
      concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",  
      concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",  
      date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data  
      de Admissão",  
coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de  
      Demissão", endF.cidade "Cidade" from funcionario  
func  
      inner join enderecofunc endF on func.cpf =  
      endF.Funcionario_cpf  
      where endF.cidade like "Olinda" and  
      func.dataDem is null  
      order by func.nome;
```

-- 4. view funcAtivoRecife

create view funcAtivoRecife as

```
select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil  
      "Estado Civil",  
      date_format(func.dataNasc, '%d/%m/%Y') "Data de
```

```

        Nascimento",
        func.email "E-mail", func.especialidade "Especialidade",
        concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",
        concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",
        date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
        de Admissão",
        coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de
        Demissão",
        endF.cidade "Cidade" from
        funcionario func
        inner join enderecofunc endF on func.cpf =
        endF.Funcionario_cpf
        where endF.cidade like "Recife" and
        func.dataDem is null

        order by func.nome;

```

-- 5. view funcSalarioMenor2000ativo

```

create view funcSalarioMenor2000ativo as

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
        "Estado Civil",
        date_format(func.dataNasc, '%d/%m/%Y') "Data de
        Nascimento",
        func.email "E-mail", func.especialidade "Especialidade",
        concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",
        concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",
        date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
        de Admissão",

```

```

coalesce(date_format(func.dataDem, "%h: %i %d/%m%Y"), "Ativo") "Data de
Demissão", endF.cidade "Cidade" from funcionario
func
inner join enderecofunc endF on func.cpf =
endF.Funcionario_cpf
where salario <= 2000 and func.dataDem is null
order by func.nome;

```

-- 6. view funcSalariaioMaior2000ativo

```

create view funcSalariaioMaior2000ativo as

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
"Estado Civil",
date_format(func.dataNasc, '%d/%m/%Y') "Data de
Nascimento",
func.email "E-mail", func.especialidade "Especialidade",
concat("R$ ", format(func.salario, 2, 'de_DE')) "Salário",
concat("R$ ", format(func.comissao, 2, 'de_DE')) "Comissão",
date_format(func.dataAdm, "%h:%i %d/%m/%Y")
"Data de Admissão",
coalesce(date_format(func.dataDem, "%h: %i %d/%m?%Y"), "Ativo") "Data de
Demissão",
endF.cidade "Cidade", tel.numero "Telefone"
from funcionario func
inner join enderecofunc endF on func.cpf =
endF.Funcionario_cpf
left join telefone tel on tel.Funcionario_cpf = func.cpf

```

where salario >= 2001 and func.dataDem is null
order by func.nome;

-- 7. view MediaSalarioFuncAtivo

create view MediaSalarioFuncAtivo as

select func.cpf as "CPF", func.nome "Nome", func.sexo "Gênero", func.estadoCivil
"Estado Civil",

date_format(func.dataNasc, '%d/%m/%Y') "Data de
Nascimento",

func.email "E-mail", func.especialidade "Especialidade",

concat("R\$ ", format(func.salario, 2, 'de_DE')) "Salário",

concat("R\$ ", format(func.comissao, 2, 'de_DE')) "Comissão",

date_format(func.dataAdm, "%h:%i %d/%m/%Y") "Data
de Admissão",

coalesce(date_format(func.dataDem, "%h: %i %d/%m/%Y"), "Ativo") "Data de
Demissão", endF.cidade "Cidade" from funcionario
func

inner join enderecofunc endF on func.cpf =
endF.Funcionario_cpf

where salario > (select avg(salario) from
funcionario) and func.dataDem is null

order by salario desc;

-- 8. view exibirAgendamPorData

create view exibirAgendamPorData as

```

select ag.status, ag.hora, ag.data,
cli.nome "Nome do Cliente", cli.email "Email do Cliente",
tel.numero "Telefone do Cliente",
func.nome "Nome do Funcionario", srv.nome "Tipo de Serviço", srv.valor AS
"Valor do Serviço"
from agendamento ag
inner join cliente cli on ag.Cliente_email = cli.email inner
join telefone tel on cli.email =
tel.Cliente_email
inner join funcionario func on ag.Funcionario_CPF =
func.cpf
inner join servico srv on ag.Servico_idServico =
srv.idServico
where ag.data = '2024-12-07'
order by ag.data;

```

-- 9. view listaClientesSalao

```

create view listaClientesSalao as

select cli.email AS "Email", cli.nome "Nome", cli.sexo "Gênero",
DATE_FORMAT(cli.dataNasc, '%d/%m/%Y') AS "Data de Nascimento",
endF.cidade as "Cidade", tel.numero
as "Telefone"
from cliente cli
inner join endereccli endF on cli.email =
endF.Cliente_email

```

```
left join telefone tel on tel.Cliente_email = cli.email  
order by cli.nome;
```

-- 10. view ValorTotalMesFunc

create view ValorTotalMesFunc as

```
select func.nome as "Nome do Funcionário",  
       DATE_FORMAT(ag.data, '%Y-%m') "Mês",  
       SUM(srv.valor) "Valor Total Arrecadado" from  
agendamento ag  
  
       inner join funcionario func on ag.Funcionario_CPF =  
       func.cpf  
  
       inner join servico srv on ag.Servico_idServico =  
       srv.idServico  
  
       group by func.nome, DATE_FORMAT(ag.data,  
       '%Y-%m');
```

Scripts

5 PROCEDURES

-- 1. CADASTRAR NOVO CLIENTE.

delimiter \$\$


```

create procedure cadCliente(
                                in pNome
                                varchar(60),
                                in pSexo char(1),
                                in pEmail
                                varchar(45),
                                in pDataNasc date,
                                in pCidade
                                varchar(60),
                                in pTelefone
                                varchar(15)) begin
insert into cliente (nome, sexo, email, dataNasc)
values (pNome, pSexo, pEmail, pDataNasc);

insert into enderecofunc (Cliente_email, cidade)
values (pEmail, pCidade);

insert into telefone (numero, Cliente_email) values
(pTelefone, pEmail);

end $$

delimiter ;

```

-- 2. ATUALIZAR SALÁRIO DO FUNCIONÁRIO.

```

delimiter $$

create procedure atualizaSalarioFuncionario(
                                in
                                pCPF varchar(14),

```

```
in pNovoSalario decimal(7,2))  
begin  
update funcionario set  
salario = pNovoSalario where  
cpf = pCPF; end $$  
delimiter ;
```

-- 3. DELETAR CLIENTE.

```
delimiter $$  
  
create procedure deletaCliente(  
in pEmail  
varchar(45)) begin  
delete from telefone where Cliente_email = pEmail;  
delete from enderecofunc where Cliente_email = pEmail; delete  
from cliente where email = pEmail;  
end $$  
delimiter ;
```

-- 4. REGISTRAR NOVO AGENDAMENTO.

```
delimiter $$  
  
create procedure regAgendamento(  
in pStatus
```

```

        varchar(45),
                                in pHora
        time,
                                in pData
        date,
                                in
        pClienteEmail varchar(45),
                                in
        pFuncionarioCPF varchar(14),
                                in
        pServicoID int) begin
    insert into agendamento (status, hora, data, Cliente_email,
        Funcionario_CPF, Servico_idServico)
values (pStatus, pHora, pData, pClienteEmail, pFuncionarioCPF,
        pServicoID); end $$
    delimiter ;

```

-- 5. ATUALIZAR STATUS DO AGENDAMENTO.

```

    delimiter $$

create procedure atualizaStatusAgendamento(

    in pIDAgendamento int,

    in pNovoStatus varchar(45))
begin
    update agendamento
    set status = pNovoStatus where
    idAgendamento = pIDAgendamento;

```

end \$\$

delimiter ;

5 FUNÇÕES

-- 1. CALCULAR DESCONTO

delimiter \$\$

create function calcDesconto(preco decimal(7,2), percentual decimal(5,2))

returns decimal(7,2) deterministic

begin

return preco - (preco * percentual / 100);

end \$\$

delimiter ;

-- 2. CALCULAR TOTAL DE SERVIÇOS POR CLIENTE

delimiter \$\$

create function totalServicosCliente(pEmail varchar(45))

returns int deterministic

begin

```
        declare total int;

select COUNT(*) into total  from

        agendamento

where Cliente_email = pEmail;

        return total; end

        $$

delimiter ;
```

-- 3. CALCULAR TOTAL DE SERVIÇOS POR FUNCIONÁRIO.

```
        delimiter $$

create function totalServicosFuncionario(pCPF varchar(14))

        returns int deterministic begin

        declare total int;

select COUNT(*) into total  from

        agendamento

where Funcionario_CPF = pCPF;

        return total; end

        $$

delimiter ;
```

-- 4. CALCULAR RECEITA TOTAL POR FUNCIONÁRIO. delimiter \$\$

```

create function receitaTotalFuncionario(pCPF varchar(14))
    returns decimal(10,2) deterministic
    begin
        declare total decimal(10,2);
        select SUM(srv.valor) into total    from
            agendamento ag
inner join servico srv on ag.Servico_idServico = srv.idServico
        where ag.Funcionario_CPF = pCPF;
        return total; end
    $$

delimiter ;

```

-- 5. CALCULAR RECIETA TOTAL POR CLIENTE.

```

delimiter $$

create function receitaTotalCliente(pEmail varchar(45)) returns
    decimal(10,2) deterministic
    begin
        declare total decimal(10,2);
        select SUM(srv.valor) into total    from
            agendamento ag
inner join servico srv on ag.Servico_idServico = srv.idServico
        where ag.Cliente_email = pEmail;

        return total; end
    $$

```

delimiter ;

Scripts

6 TRIGGERS

**-- 1. TRIGGER PARA ATUALIZAR O VALOR TOTAL DE UM AGENDAMENTO APÓS A
INSERÇÃO DE UM AGENDAMENTO.**

delimiter \$\$

```
create trigger trg_aft_insert_formapag after insert  
on FormaPag  
for each row begin  
update Agendamento  
set valorTotal = valorTotal + new.valor  
where idAgendamento = new.Agendamento_idAgendamento;  
end $$
```

delimiter ;

**-- 2. TRIGGER PARA ATUALIZAR O ESTOQUE DE UM SERVIÇO APÓS A INSERÇÃO
DE UM AGENDAMENTO.**

delimiter \$\$

```
create trigger trg_aft_insert_agendamento after insert
```

```
on Agendamento
for each row begin
    update Servico
    set quantidade = quantidade - 1 where
    idServico = new.Servico_idServico;
end $$

delimiter ;
```

-- 3. TRIGGER PARA IMPEDIR A EXCLUSÃO DE UM CLIENTE COM AGENDAMENTO ASSOCIADOS.

```
delimiter $$

create trigger trg_bef_delete_cliente before delete
on Cliente
for each row begin
if exists (select 1 from Agendamento where Cliente_email = old.email) then
    signal sqlstate '45000' set message_text = 'Não é possível excluir cliente com
    agendamentos associados';
end if; end
$$ delimiter
;
```

-- 4. TRIGGER PARA ATUALIZAR O STATUS DE UM AGENDAMENTO APÓS A CONCLUSÃO.alter

delimiter \$\$

```
create trigger trg_aft_update_agendamento after update  
on Agendamento  
for each row begin  
if new.status = 'Concluído' then  
update Agendamento  
set dataConclusao = now() where  
idAgendamento = new.idAgendamento;  
end if; end  
$$
```

delimiter ;

**-- 5. TRIGGER PARA IMPEDIR A INSERÇÃO DE UM AGENDAMENTO EM UM HORÁRIO
JÁ OCUPADO PELO MESMO FUNCIONÁRIO.**

delimiter \$\$

```
create trigger trg_bef_insert_agendamento before insert  
on Agendamento  
for each row begin  
if exists (  
select 1 from Agendamento  
where Funcionario_CPF = new.Funcionario_CPF
```

```

        and data = new.data
        and hora = new.hora    )
    then
        signal sqlstate '45000' set message_text = 'O funcionário já tem um
        agendamento neste horário';
        end if; end
    $$

```

```

delimiter ;

```

-- 6. TRIGGER PRARA IMPEDIR A INSERÇÃO DE UM CLIENTE COM UM E-MAIL JÁ EXISTENTE.

```

delimiter $$

create trigger trg_bef_insert_cliente before insert
    on Cliente
    for each row begin
        if exists (select 1 from Cliente where email = new.email) then    signal
sqlstate '45000' set message_text = 'O e-mail já está cadastrado para outro
cliente';
        end if;
    end $$

delimiter ;

```