

Estudo de caso: operadora de turismo

OBJETIVO: Criar a modelagem para o banco de dados (OPERTUR) utilizando a ferramenta **MySQL Workbench**¹, com as respectivas tabelas populadas e queries para atender às perguntas do item 3.

O sistema gerencia serviços que são ofertados por uma operadora de turismo de modo que a pessoa possa comprar ou obter informações sobre os mesmos, no qual podem ser quaisquer tipos de viagens, estadias em hotéis e excursões, podendo escolher um só serviço ou vários.

1. Entidades

1.1 Pessoa

Pessoa é uma entidade que pode ser Atendente, Guia Turístico ou Cliente que adquire os pacotes que a operadora oferece.

Apresenta os seguintes atributos:

- Atributos simples: CPF, Nome, Data de Nascimento, Nacionalidade, RG.
- Atributos compostos: Endereço (constituído por Logradouro², número, CEP, Complemento, Bairro, Cidade, UF).
- Atributos multivalorados: Telefone.

1.2 Atendente

Atendente é quem vai oferecer os serviços da operadora. Possui mesmos atributos de pessoa além de:

- Atributos simples: Código de funcionário, Salário, Turno.

1.3 Guia Turístico

O guia é quem vai realizar a excursão. Ele possui mesmos atributos de pessoa e também:

- Atributos simples: Código de funcionário, Salário.

1.4 Pacote de viagem

Os pacotes de viagem turísticos são um grupo de reservas montados pela agência de viagem que agregam desde a reserva de passagens de avião, quartos de hotéis/pousadas, locação de carros e outros serviços como traslados, passeios turísticos, guias turísticos e outros passeios pelos locais desejados.

Aqui, referido apenas pelo termo “pacote”, é oferecido pela operadora à pessoa que pode vir a comprá-lo. Contempla: passagem, hotel e/ou passeio e nada impede a pessoa de comprar vários pacotes para uma viagem só.

Apresenta os seguintes atributos:

- Atributos simples: Código, Preço.

1.5 Passagem

Trata-se de um tipo de pacote que a operadora fornece.

Apresenta os seguintes atributos:

- Atributos simples: Código, Preço, Companhia, Cidade de Destino, UF.
- Atributos compostos: Saída e Chegada.

1.6 Hotel

Trata-se de outro tipo de pacote que a operadora fornece.

Apresenta os seguintes atributos:

- Atributos simples: Código, Preço, Nome.
- Atributos compostos: Endereço (constituído por Logradouro, número, CEP, Bairro, Cidade, UF).

1.7 Passeio

Trata-se de um tipo de pacote que a operadora fornece. Neste tipo de pacote se contrata um guia para acompanhar a pessoa no passeio.

Apresenta os seguintes atributos:

- Atributos simples: Código, Preço, Data, Hora, Destino.

1.8 Credencial

Cada Guia tem uma credencial que o identifica como tal (sendo uma entidade fraca).

Apresenta os seguintes atributos:

- Atributos simples: Registro, Validade.

¹ <https://www.devmedia.com.br/introducao-ao-novo-mysql-workbench/25939>

² Logradouro significa aquilo que pode ser logrado, usufruído ou desfrutado por alguém. Em termos gerais, logradouro é uma rua; um endereço de espaço público ou privado. Em Urbanismo, logradouro é um espaço público reconhecido oficialmente pela administração de cada município. São os espaços livres como as ruas, avenidas, praças, jardins, etc., destinados ao uso comum dos cidadãos e à circulação de veículos.

2. Relacionamentos

2.1 Oferece

Relacionamento triplo entre Cliente, Atendente e Pacotes.

Um atendente pode atender várias pessoas para vários pacotes e uma pessoa pode escolher vários pacotes com um atendente.

Relacionamento Obrigatório: O pacote tem que ser oferecido pelo atendente para a pessoa.

2.2 Supervisiona

Auto Relacionamento de Guia, onde um guia pode supervisionar vários guias.

2.3 Guia

Relacionamento entre Passeio e Guia Turístico, no qual o um guia pode guiar vários passeios, mas um passeio é somente guiado por um guia.

Relacionamento obrigatório total.

2.4 Tem

Relacionamento fraco entre Guia Turístico e Credencial.

Um Guia tem uma credencial.

Relacionamento Obrigatório total.

2.5 Compra

Relacionamento entre Cliente e Passagem. Deve haver um atributo temporal para registrar o histórico da venda de passagens.

Atributo Temporal: Data.

Uma pessoa pode comprar várias passagens. Uma passagem é comprada por uma pessoa.

Relacionamento obrigatório: uma pessoa pode comprar uma passagem, mas uma passagem tem que ser comprada por uma pessoa.

2.6 Reserva

Relacionamento entre Cliente e Hotel.

Um hotel pode ser reservado por vários clientes, mas um cliente só faz reserva em um hotel.

Relacionamento obrigatório: um cliente pode reservar um hotel, mas um hotel tem que ser reservado por um cliente.

2.7 Agenda

Relacionamento entre Cliente e Passeio.

Um cliente pode agendar vários passeios e um passeio é agendado por vários clientes.

Relacionamento obrigatório: um cliente pode agendar um passeio, mas um passeio tem que ser agendado por um cliente.

2.8 Pacote Passagem

Relacionamento entre Pacote e Passagem.

Um pacote pode possuir várias passagens e uma passagem pode estar contida em vários pacotes.

2.9 Pacote Hotel

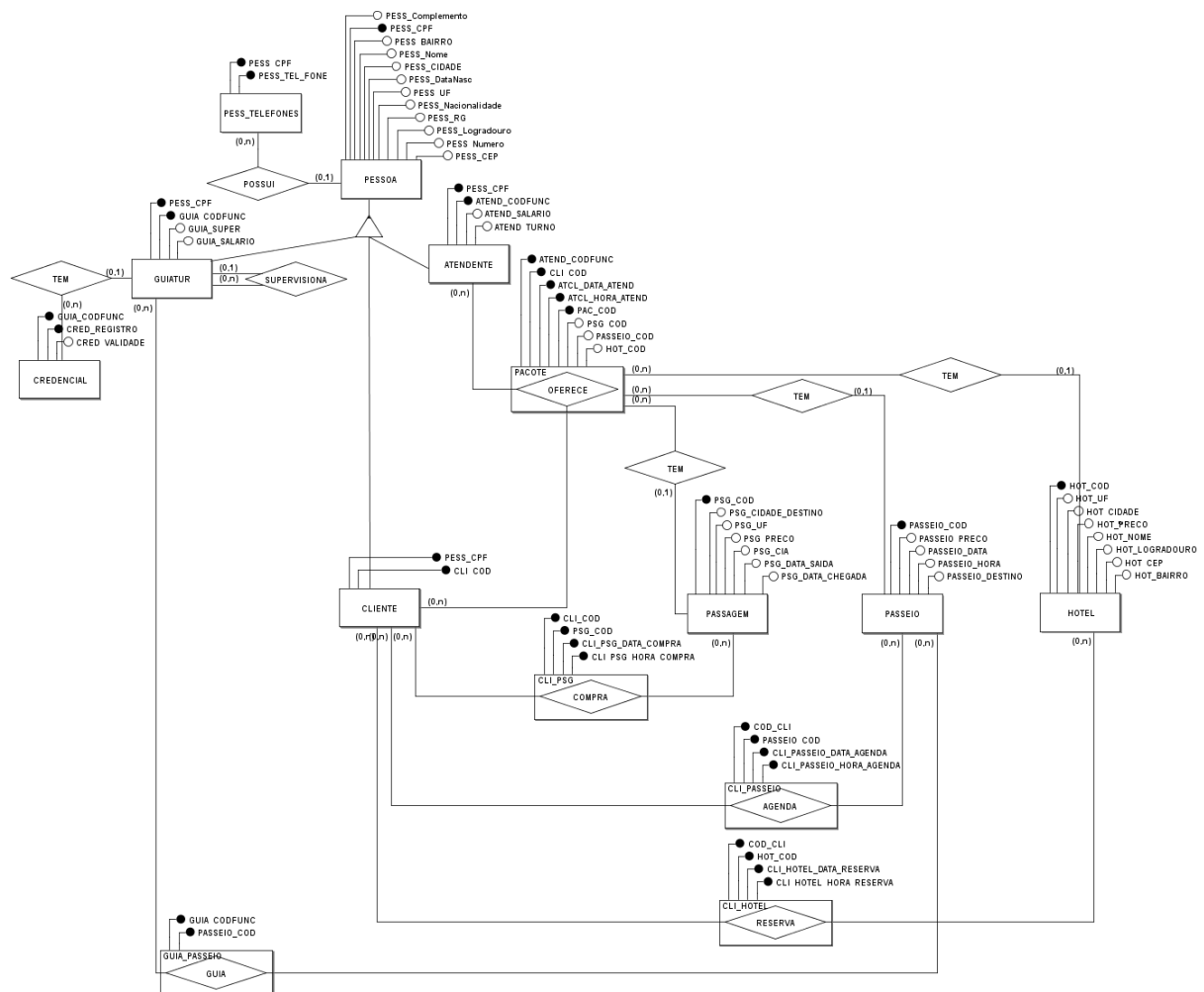
Relacionamento entre Pacote e Hotel.

Um pacote pode possuir vários hotéis e um hotel pode estar contido em vários pacotes.

2.10 Pacote Passeio

Relacionamento entre Pacote e Passeio.

Um pacote pode possuir vários passeios e um passeio pode estar contido em vários pacotes.



[illegible]

```

/* BANCO DE DADOS DBOPERTUR */
-----
-- CRIAÇÃO DO BANCO DE DADOS DBOPERTUR
-----

CREATE DATABASE DBOPERTUR2021;

-----

-- ABRE O BANCO DE DADOS DBOPERTUR
-----

USE DBOPERTUR2021;

-----

-- CRIAÇÃO DAS TABELAS DO BANCO DE DADOS DBOPERTUR
-----

CREATE TABLE PESSOA (
    PESS_CPF                VARCHAR(11) NOT NULL PRIMARY KEY,
    PESS_Nome                VARCHAR(50),
    PESS_DataNasc            DATE,
    PESS_Nacionalidade       VARCHAR(20),
    PESS_RG                  VARCHAR(11),
    PESS_Logradouro           VARCHAR(20),
    PESS_Numero              VARCHAR(11),
    PESS_CEP                 VARCHAR(11),
    PESS_Complemento         VARCHAR(11),
    PESS_BAIRRO              VARCHAR(20),
    PESS_CIDADE              VARCHAR(50),
    PESS_UF                  VARCHAR(2)
);

CREATE TABLE PESS_TELEFONES (
    PESS_TEL_FONE            VARCHAR(11) NOT NULL, -- PRIMARY KEY,
    FK_PESSOA_PESS_CPF       VARCHAR(11),
    PRIMARY KEY (PESS_TEL_FONE, FK_PESSOA_PESS_CPF)
);

```

```

CREATE TABLE ATENDENTE (
    ATEND_CODFUNC          INTEGER NOT NULL,
    ATEND_SALARIO          FLOAT,
    ATEND_TURNO            VARCHAR(11),
    FK_PESSOA_PESS_CPF     VARCHAR(11),
    PRIMARY KEY (ATEND_CODFUNC, FK_PESSOA_PESS_CPF)
);

CREATE TABLE GUIATUR (
    GUIA_CODFUNC          INTEGER NOT NULL, -- PRIMARY KEY,
    GUIA_SALARIO          FLOAT,
    GUIA_SUPER            INTEGER,
    FK_PESSOA_PESS_CPF     VARCHAR(11),
    PRIMARY KEY (GUIA_CODFUNC, FK_PESSOA_PESS_CPF)
);

CREATE TABLE PASSAGEM (
    PSG_COD               INTEGER NOT NULL PRIMARY KEY,
    PSG_PRECO             FLOAT,
    PSG_CIA               VARCHAR(50),
    PSG_DATA_SAIDA        DATE,
    PSG_DATA_CHEGADA      DATE,
    PSG_CIDADE_DESTINO    VARCHAR(50),
    PSG_UF                VARCHAR(2)
);

CREATE TABLE HOTEL (
    HOT_COD               INTEGER NOT NULL PRIMARY KEY,
    HOT_PRECO             FLOAT,
    HOT_NOME              VARCHAR(50),
    HOT_LOGRADOURO        VARCHAR(11),
    HOT_CEP               VARCHAR(11),
    HOT_BAIRRO            VARCHAR(20),
    HOT_UF                VARCHAR(2),
    HOT_CIDADE            VARCHAR(50)
);

CREATE TABLE PASSEIO (
    PASSEIO_COD           INTEGER NOT NULL PRIMARY KEY,
    PASSEIO_PRECO         FLOAT,
    PASSEIO_DATA          DATE,
    PASSEIO_HORA          TIME,
    PASSEIO_DESTINO       VARCHAR(50)
);

CREATE TABLE CREDENCIAL (
    CRED_REGISTRO         INTEGER NOT NULL,
    CRED_VALIDADE         DATE,
    FK_GUIATUR_FK_PESSOA_PESS_CPF VARCHAR(11),
    FK_GUIATUR_GUIA_CODFUNC INTEGER,
    PRIMARY KEY (CRED_REGISTRO, FK_GUIATUR_FK_PESSOA_PESS_CPF, FK_GUIATUR_GUIA_CODFUNC)
);

CREATE TABLE CLIENTE (
    CLI_COD               INTEGER NOT NULL,
    FK_PESSOA_PESS_CPF     VARCHAR(11),
    PRIMARY KEY (CLI_COD, FK_PESSOA_PESS_CPF)
);

CREATE TABLE PACOTE_OFERECE (
    PAC_COD               INTEGER NOT NULL,
    ATCL_DATA_ATEND       DATE NOT NULL,
    ATCL_HORA_ATEND       TIME NOT NULL,
    FK_PASSAGEM_PSG_COD   INTEGER,
    FK_PASSEIO_PASSEIO_COD INTEGER,
    FK_HOTEL_HOT_COD      INTEGER,
    FK_CLIENTE_CLI_COD    INTEGER,
    FK_CLIENTE_FK_PESSOA_PESS_CPF VARCHAR(11),
    FK_ATENDENTE_ATEND_CODFUNC INTEGER,
    FK_ATENDENTE_FK_PESSOA_PESS_CPF VARCHAR(11),
    PRIMARY KEY (FK_ATENDENTE_ATEND_CODFUNC, FK_CLIENTE_CLI_COD, ATCL_DATA_ATEND, ATCL_HORA_ATEND, PAC_COD )
);

CREATE TABLE CLI_PSG_COMPRA (
    CLI_PSG_DATA_COMPRA   DATE NOT NULL,
    CLI_PSG_HORA_COMPRA   TIME NOT NULL,
    FK_CLIENTE_CLI_COD    INTEGER,
    FK_CLIENTE_FK_PESSOA_PESS_CPF VARCHAR(11),
    FK_PASSAGEM_PSG_COD   INTEGER,
    PRIMARY KEY (CLI_PSG_DATA_COMPRA, CLI_PSG_HORA_COMPRA, FK_CLIENTE_CLI_COD, FK_PASSAGEM_PSG_COD)
);

CREATE TABLE CLI_HOTEL_RESERVA (
    CLI_HOTEL_DATA_RESERVA DATE NOT NULL,
    CLI_HOTEL_HORA_RESERVA TIME NOT NULL,
    FK_HOTEL_HOT_COD      INTEGER,
    FK_CLIENTE_CLI_COD    INTEGER,
    FK_CLIENTE_FK_PESSOA_PESS_CPF VARCHAR(11),

```

```

        PRIMARY KEY (FK_CLIENTE_CLI_COD, FK_HOTEL_HOT_COD, CLI_HOTEL_DATA_RESERVA, CLI_HOTEL_HORA_RESERVA)
    );

CREATE TABLE CLI_PASSEIO_AGENDA (
    CLI_PASSEIO_DATA_AGENDA          DATE NOT NULL,
    CLI_PASSEIO_HORA_AGENDA          TIME NOT NULL,
    FK_PASSEIO_PASSEIO_COD          INTEGER,
    FK_CLIENTE_CLI_COD              INTEGER,
    FK_CLIENTE_FK_PESSOA_PESS_CPF    VARCHAR(11),
    PRIMARY KEY (CLI_PASSEIO_DATA_AGENDA, CLI_PASSEIO_HORA_AGENDA, FK_CLIENTE_CLI_COD, FK_PASSEIO_PASSEIO_COD)
);

CREATE TABLE GUIA_PASSEIO_GUIA (
    FK_PASSEIO_PASSEIO_COD          INTEGER,
    FK_GUIATUR_GUIA_CODFUNC          INTEGER,
    PRIMARY KEY (FK_PASSEIO_PASSEIO_COD, FK_GUIATUR_GUIA_CODFUNC)
);

-- -----
-- MOSTRA AS TABELAS CRIADAS
-- -----
SHOW TABLES;

```

ABASTECENDO AS TABELAS PARA TESTES

```

-- -----
-- INCLUINDO DADOS NA TABELA PESSOA
-- -----
INSERT INTO PESSOA (PESS_CPF, PESS_RG, PESS_NOME)
VALUES ('10148976529', '181201832', 'MARIA LUIZA SANTOS');

INSERT INTO PESSOA (PESS_CPF, PESS_RG, PESS_NOME, PESS_DataNasc, PESS_Nacionalidade, PESS_Logradouro,
PESS_BAIRRO, PESS_Numero, PESS_Complemento, PESS_CEP)
VALUES ('11122233344', '123456789', 'TERESINHA QUEIROZ', '1960-09-30', 'BRASILEIRA', 'RUA
BOTUCATU', 'ACLIÇÃO', 2098, 'A2-52', '01531000' );

INSERT INTO PESSOA (PESS_CPF, PESS_RG, PESS_NOME, PESS_DataNasc, PESS_Nacionalidade, PESS_Logradouro,
PESS_BAIRRO, PESS_Numero, PESS_Complemento, PESS_CEP)
VALUES ('11111111111', '111111111', 'JUVENAL SANTOS', '1974-07-04', 'ARGENTINA', 'RUA APA',
'CAMPOS ELÍSEOS', 195, 'AP 12', '01201030' );

INSERT INTO PESSOA (PESS_CPF, PESS_RG, PESS_NOME, PESS_DataNasc, PESS_Nacionalidade, PESS_Logradouro,
PESS_BAIRRO, PESS_Numero, PESS_Complemento, PESS_CEP)
VALUES ('22222222222', '222222222', 'ANA PAULA SILVA', '2000-11-23', 'BRASILEIRA', 'AVENIDA JACÚ
PÊSSEGO', 'ITAQUERA', 3007, NULL, '08260001' );

INSERT INTO PESSOA (PESS_CPF, PESS_RG, PESS_NOME, PESS_DataNasc, PESS_Nacionalidade, PESS_Logradouro,
PESS_BAIRRO, PESS_Numero, PESS_CEP)
VALUES ('33333333333', '333333333', 'CÍCERO MATEUS', '1998-10-10', 'BRASILEIRA', 'RUA DOUTOR
CÉSAR', 'SANTANA', 1378, '02013004' );

INSERT INTO PESSOA (PESS_CPF, PESS_RG, PESS_NOME, PESS_DataNasc, PESS_Nacionalidade, PESS_Logradouro,
PESS_BAIRRO, PESS_Numero, PESS_CEP)
VALUES ('44444444444', '444444444', 'IVAN MARTINEZ', '2000-10-10', 'BRASILEIRA', 'RUA PEDRO
VICENTE', 'CANINDE', 695, '01109010' );

-- -----
-- EXIBINDO O CONTEÚDO DE PESSOA
-- -----
SELECT * FROM PESSOA;

-- -----
-- ALTERANDO OS DADOS DA TABELA PESSOA
-- -----
UPDATE PESSOA SET PESS_DataNasc = '1957-09-21' WHERE PESS_CPF = '10148976529';

UPDATE PESSOA SET PESS_Nacionalidade = 'BRASILEIRA', PESS_Logradouro = 'RUA PEDRO VICENTE', PESS_BAIRRO =
'CANINDE', PESS_Numero = 695 WHERE PESS_CPF = '10148976529';

UPDATE PESSOA SET PESS_CEP = '01109010' WHERE PESS_CPF = '10148976529';
UPDATE PESSOA SET PESS_DataNasc = '1950-09-30' WHERE PESS_CPF = '11122233344';
UPDATE PESSOA SET PESS_DataNasc = '1988-10-10' WHERE PESS_CPF = '33333333333';
UPDATE PESSOA SET PESS_DataNasc = '1979-11-23' WHERE PESS_CPF = '22222222222';

-- -----
-- INCLUINDO OS TELEFONES DAS PESSOAS NA TABELA PESS_TELEFONES
-- -----

INSERT INTO PESS_TELEFONES (FK_PESSOA_PESS_CPF, PESS_TEL_FONE)
VALUES ('10148976529', '027637500');
INSERT INTO PESS_TELEFONES (FK_PESSOA_PESS_CPF, PESS_TEL_FONE)
VALUES ('10148976529', '027637525');

```

```

-- -----
-- EXIBINDO O CONTEÚDO DE PESS_TELEFONES
-- -----
SELECT * FROM PESS_TELEFONES;

-- -----
-- DEFININDO QUEM SERÃO OS ATENDENTES
-- -----
INSERT INTO ATENDENTE (ATEND_CODFUNC, ATEND_SALARIO, ATEND_TURNO, FK_PESSOA_PESS_CPF)
VALUES (1000, 954.50, 'VESPERTINO', '11122233344');

INSERT INTO ATENDENTE (ATEND_CODFUNC, ATEND_SALARIO, ATEND_TURNO, FK_PESSOA_PESS_CPF)
VALUES (2000, 954.50, 'MATUTINO', '10148976529');

-- -----
-- EXIBINDO O CONTEÚDO DE ATENDENTE
-- -----
SELECT * FROM ATENDENTE;

-- -----
-- DEFININDO QUEM SERÃO OS GUIAS
-- -----
INSERT INTO GUIATUR (GUIA_CODFUNC, FK_PESSOA_PESS_CPF, GUIA_SALARIO, GUIA_SUPER )
VALUES (1000, '1111111111', 2400.00, NULL);

INSERT INTO GUIATUR (GUIA_CODFUNC, FK_PESSOA_PESS_CPF, GUIA_SALARIO, GUIA_SUPER )
VALUES (1001, '2222222222', 1800.00, 1000);

INSERT INTO GUIATUR (GUIA_CODFUNC, FK_PESSOA_PESS_CPF, GUIA_SALARIO, GUIA_SUPER )
VALUES (1002, '3333333333', 1800.00, 1000);

INSERT INTO GUIATUR (GUIA_CODFUNC, FK_PESSOA_PESS_CPF, GUIA_SALARIO, GUIA_SUPER )
VALUES (1003, '4444444444', 1800.00, 1000);

-- -----
-- EXIBINDO O CONTEÚDO DE GUIATUR
-- -----
SELECT * FROM GUIATUR;

-- -----
-- POSSÍVEIS RESPOSTAS
-- -----

-- 3.1 Quais os nomes dos atendentes e respectivos turnos de trabalho?
SELECT PESS_NOME, ATEND_TURNO
FROM PESSOA, ATENDENTE
WHERE PESSOA.PESS_CPF = ATENDENTE.FK_PESSOA_PESS_CPF;

-- 3.2 Quais os nomes dos guias e respectivas nacionalidades?
SELECT PESS_NOME, PESS_Nacionalidade
FROM PESSOA, GUIATUR
WHERE PESSOA.PESS_CPF = GUIATUR.FK_PESSOA_PESS_CPF;

-- 3.3 Qual o salário do guia mais antigo? Qual o nome dele e quando começou a trabalhar para a operadora de turismo?

-- ALTERANDO AS TABELAS GUIATUR E ATENDENTES PARA CONTEMPLAR O FUNCIONÁRIO MAIS ANTIGO
ALTER TABLE GUIATUR ADD GUIA_DTADMISS DATE;
ALTER TABLE ATENDENTE ADD ATEND_DTADMISS DATE;

-- DEFININDO AS DATAS DE ADMISSÃO DOS ATENDENTES: TODOS FORAM CONTRATADOS EM 01/01/1997
-- SELECIONANDO OS ATENDENTES
SELECT * FROM ATENDENTE;
UPDATE ATENDENTE SET ATEND_DTADMISS = '1997-01-01' WHERE ATEND_CODFUNC = 1000;
UPDATE ATENDENTE SET ATEND_DTADMISS = '1997-01-01' WHERE ATEND_CODFUNC = 2000;

-- DEFININDO AS DATAS DE ADMISSÃO DOS GUIAS: '1998-07-01' '1999-02-02' '2007-09-03' '2018-12-24'
-- SELECIONANDO OS GUIAS
SELECT * FROM GUIATUR;
UPDATE GUIATUR SET GUIA_DTADMISS = '1998-07-01' WHERE GUIA_CODFUNC = 1000;
UPDATE GUIATUR SET GUIA_DTADMISS = '1999-02-02' WHERE GUIA_CODFUNC = 1001;
UPDATE GUIATUR SET GUIA_DTADMISS = '2007-09-03' WHERE GUIA_CODFUNC = 1002;
UPDATE GUIATUR SET GUIA_DTADMISS = '2018-12-24' WHERE GUIA_CODFUNC = 1003;

-- RESPONDENDO O ITEM 3.3 Qual o salário do guia mais antigo? Qual o nome dele e quando começou a trabalhar para a operadora de turismo?
SELECT GUIA_SALARIO, PESS_CPF, PESS_NOME, MIN(GUIA_DTADMISS) AS DTADMISS_GUIA_ANTIGO
FROM PESSOA, GUIATUR
WHERE PESSOA.PESS_CPF = GUIATUR.FK_PESSOA_PESS_CPF;

-- 3.4 Identifique o nome e há quantos anos atua na operadora de turismo a pessoa que trabalha há mais tempo? (Critério de desempate: maior idade)
SELECT YEAR(GUIA_DTADMISS) AS ANO_INICIO_TRABALHO FROM GUIATUR;
SELECT YEAR(ATEND_DTADMISS) AS ANO_INICIO_TRABALHO FROM ATENDENTE;

SELECT MIN(GUIA_DTADMISS) AS DATA MAIS ANTIGA GUIA FROM GUIATUR;
SELECT MIN(ATEND_DTADMISS) AS DATA MAIS ANTIGA ATEND FROM ATENDENTE;
-- ANOS TRABALHADOS DOS GUIAS

```

```

select PESS_CPF, PESS_NOME, GUIA_DTADMISS, YEAR(CURDATE()) - YEAR(GUIA_DTADMISS) as ANOS_TRABALHADOS_GUIA,
PESS_DataNasc,
YEAR(CURDATE()) - YEAR(PESS_DataNasc) as IDADE
from PESSOA, GUIATUR
WHERE PESSOA.PESS_CPF = GUIATUR.FK_PESSOA_PESS_CPF;

-- ANOS TRABALHADOS DOS ATENDENTES
select PESS_CPF, PESS_NOME, ATEND_DTADMISS, YEAR(CURDATE()) - YEAR(ATEND_DTADMISS) as
ANOS_TRABALHADOS_ATENDENTE, PESS_DataNasc,
YEAR(CURDATE()) - YEAR(PESS_DataNasc) as IDADE
from PESSOA, ATENDENTE
WHERE PESSOA.PESS_CPF = ATENDENTE.FK_PESSOA_PESS_CPF;

-- DADOS DO GUIA MAIS ANTIGO
select PESS_CPF, PESS_NOME, MIN(GUIA_DTADMISS) AS DTADMISS MAIS ANTIGO_GUIA, YEAR(CURDATE()) -
YEAR(GUIA_DTADMISS) as ANOS TRABALHADOS_GUIA,
PESS_DataNasc, YEAR(CURDATE()) - YEAR(PESS_DataNasc) as IDADE_GUIA
from PESSOA, GUIATUR
WHERE PESSOA.PESS_CPF = GUIATUR.FK_PESSOA_PESS_CPF;

-- DADOS DO ATENDENTE MAIS ANTIGO
select PESS_CPF, PESS_NOME, MIN(ATEND_DTADMISS) AS DTADMISS MAIS ANTIGO_ATEND, YEAR(CURDATE()) -
YEAR(ATEND_DTADMISS) as ANOS TRABALHADOS_ATEND,
PESS_DataNasc, YEAR(CURDATE()) - YEAR(PESS_DataNasc) as IDADE_ATEND
from PESSOA, ATENDENTE
WHERE PESSOA.PESS_CPF = ATENDENTE.FK_PESSOA_PESS_CPF;

-- MÁXIMO DE ANOS TRABALHADOS DO GUIA
select PESS_NOME, MAX(YEAR(CURDATE()) - YEAR(GUIA_DTADMISS)) as MAXIMO_ANOS_TRABALHADOS
from PESSOA, GUIATUR
WHERE PESSOA.PESS_CPF = GUIATUR.FK_PESSOA_PESS_CPF;

-- MÁXIMO DE ANOS TRABALHADOS DO ATENDENTE
select PESS_NOME, MAX(YEAR(CURDATE()) - YEAR(ATEND_DTADMISS)) as MAXIMO_ANOS_TRABALHADOS
from PESSOA, ATENDENTE
WHERE PESSOA.PESS_CPF = ATENDENTE.FK_PESSOA_PESS_CPF;

-- UNINDO OS DOIS RESULTADOS
WITH TB AS
(
select PESS_NOME, MAX(YEAR(CURDATE()) - YEAR(GUIA_DTADMISS)) as MAXIMO_ANOS_TRABALHADOS
from PESSOA, GUIATUR
WHERE PESSOA.PESS_CPF = GUIATUR.FK_PESSOA_PESS_CPF

UNION

select PESS_NOME, MAX(YEAR(CURDATE()) - YEAR(ATEND_DTADMISS)) as MAXIMO_ANOS_TRABALHADOS
from PESSOA, ATENDENTE
WHERE PESSOA.PESS_CPF = ATENDENTE.FK_PESSOA_PESS_CPF
)
SELECT * FROM TB ORDER BY MAXIMO_ANOS_TRABALHADOS DESC LIMIT 1;

-- 3.5 Identifique o nome e há quantos anos atua na operadora de turismo a pessoa que trabalha há menos
tempo? (Critério de desempate: menor idade)
-- DADOS DO GUIA MAIS NOVO
select DISTINCT PESS_NOME, GUIA_DTADMISS,
timestampdiff( YEAR, GUIA_DTADMISS, CURDATE() ) as ANOS TRABALHADOS, PESS_DataNasc,
timestampdiff( YEAR, PESS_DataNasc, CURDATE() ) as IDADE_EM_ANOS,
DATEDIFF(CURDATE(),GUIA_DTADMISS) AS DIAS TRABALHADOS,
DATEDIFF(CURDATE(),PESS_DataNasc) AS idade_EM_DIAS
from PESSOA, GUIATUR
WHERE PESSOA.PESS_CPF = GUIATUR.FK_PESSOA_PESS_CPF
ORDER BY idade_EM_DIAS ASC LIMIT 1;

-- DADOS DO ATENDENTE MAIS NOVO
select DISTINCT PESS_NOME, ATEND_DTADMISS,
timestampdiff( YEAR, ATEND_DTADMISS, CURDATE() ) as ANOS TRABALHADOS, PESS_DataNasc,
timestampdiff( YEAR, PESS_DataNasc, CURDATE() ) as IDADE_EM_ANOS,
DATEDIFF(CURDATE(),ATEND_DTADMISS) AS DIAS TRABALHADOS,
DATEDIFF(CURDATE(),PESS_DataNasc) AS idade_EM_DIAS
from PESSOA, ATENDENTE
WHERE PESSOA.PESS_CPF = ATENDENTE.FK_PESSOA_PESS_CPF
ORDER BY idade_EM_DIAS ASC LIMIT 1;

WITH TB AS
(
(select DISTINCT PESS_NOME, GUIA_DTADMISS,
timestampdiff( YEAR, GUIA_DTADMISS, CURDATE() ) as ANOS TRABALHADOS, PESS_DataNasc,
timestampdiff( YEAR, PESS_DataNasc, CURDATE() ) as IDADE_EM_ANOS,
DATEDIFF(CURDATE(),GUIA_DTADMISS) AS DIAS TRABALHADOS,
DATEDIFF(CURDATE(),PESS_DataNasc) AS idade_EM_DIAS
from PESSOA, GUIATUR
WHERE PESSOA.PESS_CPF = GUIATUR.FK_PESSOA_PESS_CPF
ORDER BY idade_EM_DIAS ASC LIMIT 1
)
UNION
(select DISTINCT PESS_NOME, ATEND_DTADMISS,
timestampdiff( YEAR, ATEND_DTADMISS, CURDATE() ) as ANOS TRABALHADOS, PESS_DataNasc,
timestampdiff( YEAR, PESS_DataNasc, CURDATE() ) as IDADE_EM_ANOS,

```



```

DATEDIFF(CURDATE(),ATEND_DTADMISS) AS DIAS_TRABALHADOS,
DATEDIFF(CURDATE(),PESS_DataNasc) AS idade_EM_DIAS
from    PESSOA, ATENDENTE
WHERE   PESSOA.PESS_CPF = ATENDENTE.FK_PESSOA_PESS_CPF
ORDER BY idade_EM_DIAS ASC LIMIT 1
)
) SELECT PESS_NOME, MIN(ANOS_TRABALHADOS) FROM TB LIMIT 1;

-- 3.6 Quantos pessoas trabalham na operadora de turismo?
WITH TB AS
( SELECT DISTINCT PESS_CPF, PESS_NOME
  FROM    PESSOA, GUIATUR, ATENDENTE
  WHERE   PESSOA.PESS_CPF = GUIATUR.FK_PESSOA_PESS_CPF OR PESSOA.PESS_CPF = ATENDENTE.FK_PESSOA_PESS_CPF
)
SELECT COUNT(*) AS TOTAL_FUNCIONARIOS
FROM    TB;

```