Travel Agency

Obraz zawierający niebo, plaża, na wolnym powietrzu, akcesoria

Opis wygenerowany automatycznie

1 Title photo

Uczelnia: Uniwersytet Łódzki

Wydział: Wydział ekonomiczno-socjologiczny

Kierunek: Informatyka Ekonomiczna

Rok studiów: II Rok

Semestr: 4 Semestr

Rok akademicki: 2023/2024

Autor:

* Maryia Shyliankova

Spis treści

[Opis tła projektowanego i implementowanego zagadnienia: 3](#_Toc166229197)

[Opis wymagań funkcjonalnych i niefunkcjonalnych 3](#_Toc166229198)

[Wymagania funkcjonalne: 3](#_Toc166229199)

[Wymagania niefunkcjonalne: 4](#_Toc166229200)

[Model fizyczny bazy danych w postaci graficznej 6](#_Toc166229201)

[Skrypt tworzenia bazy danych 7](#_Toc166229202)

[Wykaz utworzonych tabel fizycznych 7](#_Toc166229203)

[Skrypt tworzenia tabel i referencji między nimi 7](#_Toc166229204)

[Wykaz zaimplementowanych indeksów 12](#_Toc166229205)

[Skrypt tworzenia indeksów 13](#_Toc166229206)

[Skrypt ładujący przykładowe dane do tabel 14](#_Toc166229207)

[Wykaz utworzonych widoków 61](#_Toc166229208)

[Skrypt tworzenia widoków 61](#_Toc166229209)

[Wykaz utworzonych procedur składowanych 76](#_Toc166229210)

[Skrypt tworzenia procedur składowanych 78](#_Toc166229211)

[UPDATE 78](#_Toc166229212)

[DELETE 98](#_Toc166229213)

[ADD 106](#_Toc166229214)

[GET 121](#_Toc166229215)

[CALCULATE FOR TRIGGERS 125](#_Toc166229216)

[MASS DATA MODIFICATION 128](#_Toc166229217)

[REPORTS 130](#_Toc166229218)

[Wykaz utworzonych funkcji użytkownika 134](#_Toc166229219)

[Skrypt tworzenia funkcji definiowanych przez użytkownika 135](#_Toc166229220)

[Wykaz utworzonych wyzwalaczy 143](#_Toc166229221)

[Skrypt tworzenia wyzwalaczy 144](#_Toc166229222)

[Wykaz obiektów wartości domyślnej 156](#_Toc166229223)

[Skrypt tworzenia obiektów wartości domyślnej 156](#_Toc166229224)

[Schemat zaprojektowanego systemu 157](#_Toc166229225)

[Wnioski i podsumowanie 163](#_Toc166229226)

[Wykaz/spis ilustracji 164](#_Toc166229227)

[Wykaz/spis tabel 165](#_Toc166229228)

# Opis tła projektowanego i implementowanego zagadnienia:

Baza danych "Travel Agency" jest kluczowym narzędziem w organizacji podróży klientów przez agencję turystyczną. Zapewnia ona kompleksowe zarządzanie danymi niezbędnymi do planowania, rezerwacji i monitorowania podróży.

W bazie znajdują się informacje o klientach, obejmujące dane osobowe, kontakty oraz adresy, co ułatwia identyfikację i komunikację. Dodatkowo, dane dotyczące kontynentów, krajów i miast umożliwiają wybór odpowiedniej destynacji podróży.

Informacje o hotelach, lotach, przewodnikach, kierowcach i autobusach są kluczowe dla składania kompleksowych pakietów podróży. Pozwalają one na dobranie najlepszych opcji zakwaterowania, tras przelotów oraz transportu między miejscami docelowymi.

Rezerwacje podróży są śledzone w osobnej tabeli, co ułatwia zarządzanie nimi i monitorowanie historii podróży klientów. To pozwala agencji dostosować ofertę do preferencji klientów, zapewniając wysoką jakość usług i zadowolenie klientów, co w konsekwencji przekłada się na wzrost dochodów.

# Opis wymagań funkcjonalnych i niefunkcjonalnych

## Wymagania funkcjonalne:

Zarządzanie danymi podróży:

* System umożliwia dodawanie, edycję i usuwanie danych dotyczących kontynentów, krajów, miast, tras autobusowych, lotów, hoteli oraz pakietów podróży i td.
* Użytkownicy mogą przeglądać informacje o dostępnych trasach, lotach, hotelach i pakietach podróży.
* Możliwość wyszukiwania pakietów podróży według różnych kryteriów, takich jak kraj, kontynent, miasto, cena itp.

Obliczanie szczegółów pakietu:

* System powinien obliczać i aktualizować szczegóły pakietu, takie jak data rozpoczęcia, data zakończenia, długość trwania i cena, na podstawie danych związanego z pakietem hotelu, lotów i tras autobusowych.

Rezerwacje:

* System umożliwia klientom dokonywanie rezerwacji pakietów podróży.
* Klienci mogą przeglądać swoje dokonane rezerwacje oraz historię transakcji.
* Możliwość zarządzania rezerwacjami przez pracowników biura podróży, takimi jak dodawanie, edycja i usuwanie rezerwacji.

Aktualizacja cen rezerwacji:

* System powinien obliczać i aktualizować ceny dla rezerwacji związanych z określonym pakietem na podstawie udzielonego procentowego rabatu.

Zarządzanie użytkownikami:

* Administratorzy systemu mogą zarządzać użytkownikami, nadawać im role i uprawnienia oraz blokować dostęp w przypadku naruszenia zasad.

Generowanie raportów:

* System powinien umożliwiać generowanie raportów dotyczących przychodów, popularności pakietów podróżnych oraz raportów rezerwacji.
* Raporty powinny być dostępne dla określonego zakresu dat lub określonego miesiąca i roku.
* Raporty przychodów powinny obejmować łączną kwotę przychodów wygenerowaną z rezerwacji pakietów podróżnych.
* Raporty popularności powinny prezentować ilość rezerwacji dla każdego pakietu podróżnego w kolejności malejącej popularności.

## Wymagania niefunkcjonalne:

Wydajność:

* System powinien być responsywny i umożliwiać szybkie przeglądanie oraz operowanie dużą ilością danych.
* Czas odpowiedzi systemu na żądania użytkowników powinien być minimalny.

Bezpieczeństwo:

* Wszystkie dane osobowe klientów powinny być przechowywane zgodnie z przepisami o ochronie danych osobowych.
* System powinien zapewniać mechanizmy autoryzacji i uwierzytelniania użytkowników oraz kontrolować dostęp do danych.

Dostępność:

* System powinien być dostępny przez większość czasu, minimalizując ewentualne przerwy serwisowe.
* Konieczność regularnych kopii zapasowych danych w celu zapewnienia możliwości ich odtworzenia w przypadku awarii systemu.

Skalowalność:

* System powinien być skalowalny, umożliwiając dostosowanie do rosnącej liczby użytkowników oraz danych.

Interfejs użytkownika:

* Interfejs użytkownika powinien być intuicyjny i łatwy w obsłudze, zapewniając pozytywne doświadczenia użytkownika.
* System powinien być responsywny i dostosowany do różnych urządzeń, w tym komputerów stacjonarnych, tabletów i smartfonów.

Dokumentacja:

* Konieczność utrzymania aktualnej dokumentacji systemu, zawierającej opisy funkcjonalności, instrukcje obsługi oraz procedury konserwacji i administracji.

# Model fizyczny bazy danych w postaci graficznej

Obraz zawierający tekst, zrzut ekranu, numer, Równolegle

Opis wygenerowany automatycznie

2 The physical model of the database in graphical form

# Skrypt tworzenia bazy danych

/\*

A script for creating an empty database structure. This script is a basic step in the database creation process.

\*/

CREATE DATABASE [TRAVEL\_AGENCY]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'TRAVEL\_AGENCY', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\TRAVEL\_AGENCY.mdf', SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'CTRAVEL\_AGENCY\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\TRAVEL\_AGENCY\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

WITH CATALOG\_COLLATION = DATABASE\_DEFAULT

# Wykaz utworzonych tabel fizycznych

--This query returns a list of all tables in the "dbo" schema.

USE TRAVEL\_AGENCY;

SELECT TABLE\_NAME

FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_TYPE = 'BASE TABLE';

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

3 List of physical tables created

# Skrypt tworzenia tabel i referencji między nimi

-- This script creates tbl\_continent tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_continent (

contId INT PRIMARY KEY IDENTITY(1,1),

contName VARCHAR(25) NOT NULL UNIQUE

);

GO

-- This script creates tbl\_country tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_country (

ctryId INT PRIMARY KEY IDENTITY(1,1),

ctryName VARCHAR(45) UNIQUE,

contId INT NOT NULL,

FOREIGN KEY (contId) REFERENCES tbl\_continent(contId)

);

GO

-- This script creates tbl\_city tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_city (

cityId INT PRIMARY KEY IDENTITY(1,1),

cityName VARCHAR(60),

ctryId INT NOT NULL,

FOREIGN KEY (ctryId) REFERENCES tbl\_country(ctryId)

);

GO

-- This script creates tbl\_bus tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_bus (

busId INT PRIMARY KEY IDENTITY(1,1),

busNo VARCHAR(20) NOT NULL UNIQUE,

busModel VARCHAR(50),

busColor VARCHAR(20)

);

GO

-- This script creates tbl\_driver tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_driver (

driverId INT PRIMARY KEY IDENTITY(1,1),

driverName VARCHAR(50) NOT NULL,

driverSurName VARCHAR(50) NOT NULL,

driverPhone VARCHAR(20) NOT NULL

);

GO

-- This script creates tbl\_guide tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_guide (

guideId INT PRIMARY KEY IDENTITY(1,1),

guideName VARCHAR(50) NOT NULL,

guideSurname VARCHAR(50) NOT NULL,

guideCityId INT NOT NULL,

guidePhone VARCHAR(20) NOT NULL,

FOREIGN KEY (guideCityId) REFERENCES tbl\_city(cityId)

);

GO

-- This script creates tbl\_address tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_address (

addrId INT PRIMARY KEY IDENTITY(1,1),

addrCityId INT NOT NULL,

addrPostalCode VARCHAR(20),

addrRegion VARCHAR(100),

addrStreet VARCHAR(255) NOT NULL,

addrHouseNo VARCHAR(10) NOT NULL,

FOREIGN KEY (addrCityId) REFERENCES tbl\_city(cityId)

);

GO

-- This script creates tbl\_customer tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_customer (

custId INT PRIMARY KEY IDENTITY(1,1),

custName VARCHAR(50) NOT NULL,

custSurname VARCHAR(50) NOT NULL,

custPhone VARCHAR(20) NOT NULL,

custEmailAddress VARCHAR(255),

custAddrId INT NOT NULL,

FOREIGN KEY (custAddrId) REFERENCES tbl\_address(addrId)

);

GO

-- This script creates tbl\_hotel tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_hotel (

hotId INT PRIMARY KEY IDENTITY(1,1),

hotAddrId INT NOT NULL,

hotName VARCHAR(50) NOT NULL,

hotPricePerNight DECIMAL(10, 2) NOT NULL,

hotTelephoneNo VARCHAR(20),

hotContactEmail VARCHAR(45),

hotStars INT CHECK (hotStars IS NULL OR (hotStars >= 0 AND hotStars <= 5)),

FOREIGN KEY (hotAddrId) REFERENCES tbl\_address(addrId)

);

GO

-- This script creates tbl\_flight tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_flight (

fliId INT PRIMARY KEY IDENTITY(1,1),

fliStartCityId INT NOT NULL,

fliEndCityId INT NOT NULL,

fliStartTime DATETIME NOT NULL,

fliEndTime DATETIME NOT NULL,

fliClass VARCHAR(10) NOT NULL CHECK (fliClass IN ('First', 'Business', 'Economy')),

fliPrice DECIMAL(10, 2) NOT NULL,

CONSTRAINT chk\_flight\_time\_difference CHECK (DATEDIFF(HOUR, fliStartTime, fliEndTime) BETWEEN 1 AND 19),

FOREIGN KEY (fliStartCityId) REFERENCES tbl\_city(cityId),

FOREIGN KEY (fliEndCityId) REFERENCES tbl\_city(cityId)

);

GO

-- This script creates tbl\_busroute tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_busroute (

busRouteId INT PRIMARY KEY IDENTITY(1,1),

driverId INT NOT NULL,

busRouteStartCityId INT NOT NULL,

busRouteEndCityId INT NOT NULL,

busRouteStartTime DATETIME NOT NULL,

busRouteEndTime DATETIME NOT NULL,

busRouteBusID INT NOT NULL,

busRoutePrice DECIMAL(10, 2) NOT NULL,

CONSTRAINT chk\_busRoute\_time\_difference CHECK (DATEDIFF(HOUR, busRouteStartTime, busRouteEndTime) BETWEEN 1 AND 72),

FOREIGN KEY (driverId) REFERENCES tbl\_driver(driverId),

FOREIGN KEY (busRouteStartCityId) REFERENCES tbl\_city(cityId),

FOREIGN KEY (busRouteEndCityId) REFERENCES tbl\_city(cityId),

FOREIGN KEY (busRouteBusID) REFERENCES tbl\_bus(busId)

);

GO

-- This script creates tbl\_package tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_package (

packId INT PRIMARY KEY IDENTITY(1,1),

packTitle VARCHAR(50) NOT NULL,

packDescription TEXT,

packCityId INT NOT NULL,

packHotId INT NOT NULL,

packDuration TINYINT,

packPrice DECIMAL(10, 2),

packStartDate DATE,

packEndDate DATE,

packOutboundFlightId INT,

packReturnFlightId INT,

packOutboundBusRouteId INT,

packReturnBusRouteId INT,

packGuideId INT NOT NULL,

CONSTRAINT chk\_flight\_or\_busRoute

CHECK (

(packOutboundFlightId IS NOT NULL AND packReturnFlightId IS NOT NULL) OR

(packOutboundBusRouteId IS NOT NULL AND packReturnBusRouteId IS NOT NULL)

),

CONSTRAINT chk\_packDuration\_range

CHECK (packDuration BETWEEN 3 AND 30),

FOREIGN KEY (packCityId) REFERENCES tbl\_city(cityId),

FOREIGN KEY (packHotId) REFERENCES tbl\_hotel(hotId),

FOREIGN KEY (packOutboundFlightId) REFERENCES tbl\_flight(fliId),

FOREIGN KEY (packReturnFlightId) REFERENCES tbl\_flight(fliId),

FOREIGN KEY (packOutboundBusRouteId) REFERENCES tbl\_busroute(busRouteId),

FOREIGN KEY (packReturnBusRouteId) REFERENCES tbl\_busroute(busRouteId),

FOREIGN KEY (packGuideId) REFERENCES tbl\_guide(guideId)

);

GO

-- This script creates tbl\_booking tables

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_booking (

bookId INT PRIMARY KEY IDENTITY(1,1),

bookCustId INT NOT NULL,

bookPackageId INT NOT NULL,

bookDiscountPercent DECIMAL(10, 2) NOT NULL DEFAULT 0,

bookDiscountAmnt DECIMAL(10, 2) NOT NULL DEFAULT 0,

bookPrice DECIMAL(10, 2),

bookCreatedAt DATETIME NOT NULL DEFAULT GETDATE(),

FOREIGN KEY (bookCustId) REFERENCES tbl\_customer(custId),

FOREIGN KEY (bookPackageId) REFERENCES tbl\_package(packId)

);

GO

-- This script adds constraints to tables

ALTER TABLE tbl\_hotel

ADD CONSTRAINT chk\_hotel\_hotPricePerNight CHECK (hotPricePerNight >= 0);

ALTER TABLE tbl\_customer

ADD CONSTRAINT chk\_customer\_custEmailAddress CHECK (custEmailAddress LIKE '%@%.%');

ALTER TABLE tbl\_hotel

ADD CONSTRAINT chk\_hotel\_hotContactEmail CHECK (hotContactEmail LIKE '%@%.%');

ALTER TABLE tbl\_flight

ADD CONSTRAINT chk\_flight\_fliPrice CHECK (fliPrice > 0);

ALTER TABLE tbl\_busroute

ADD CONSTRAINT chk\_busroute\_busRoutePrice CHECK (busRoutePrice > 0);

ALTER TABLE tbl\_package

ADD CONSTRAINT chk\_package\_packPrice CHECK (packPrice > 0);

ALTER TABLE tbl\_booking

ADD CONSTRAINT chk\_booking\_bookDiscountPercent CHECK (bookDiscountPercent >= 0);

ALTER TABLE tbl\_booking

ADD CONSTRAINT chk\_booking\_bookDiscountAmnt CHECK (bookDiscountAmnt >= 0);

ALTER TABLE tbl\_booking

ADD CONSTRAINT chk\_booking\_bookPrice CHECK (bookPrice > 0);

# Wykaz zaimplementowanych indeksów

/\*

This script returns a list of indexes with their type

and the name of the table for which it was created

\*/

SELECT

t.name AS TableName,

i.name AS IndexName,

i.type\_desc AS IndexType

FROM

sys.indexes i

INNER JOIN

sys.tables t ON t.object\_id = i.object\_id

WHERE

i.index\_id > 0

ORDER BY

TableName,

IndexName;

Obraz zawierający tekst, Czcionka, dokument

Opis wygenerowany automatycznie

4 List of implemented indexes

# Skrypt tworzenia indeksów

--This script creates indexes on various columns in the "Travel Agency" database tables.

CREATE INDEX idx\_continent\_contName ON tbl\_continent (contName);

CREATE INDEX idx\_country\_contId ON tbl\_country (contId);

CREATE INDEX idx\_country\_ctryName ON tbl\_country (ctryName);

CREATE INDEX idx\_city\_ctryId ON tbl\_city (ctryId);

CREATE INDEX idx\_address\_addrCityId ON tbl\_address (addrCityId);

CREATE INDEX idx\_customer\_custAddrId ON tbl\_customer (custAddrId);

CREATE INDEX idx\_guide\_guideCityId ON tbl\_guide (guideCityId);

CREATE INDEX idx\_hotel\_hotAddrId ON tbl\_hotel (hotAddrId);

CREATE INDEX idx\_flight\_fliStartCityId ON tbl\_flight (fliStartCityId);

CREATE INDEX idx\_flight\_fliEndCityId ON tbl\_flight (fliEndCityId);

CREATE INDEX idx\_flight\_fliStartTime ON tbl\_flight (fliStartTime);

CREATE INDEX idx\_flight\_fliEndTime ON tbl\_flight (fliEndTime);

CREATE INDEX idx\_busRoute\_busRouteStartCity ON tbl\_busroute (busRouteStartCityId);

CREATE INDEX idx\_busRoute\_busRouteEndCity ON tbl\_busroute (busRouteEndCityId);

CREATE INDEX idx\_busRoute\_busRouteStartTime ON tbl\_busroute (busRouteStartTime);

CREATE INDEX idx\_busRoute\_busRouteEndTime ON tbl\_busroute (busRouteEndTime);

CREATE INDEX idx\_package\_packOutboundFlightId ON tbl\_package (packOutboundFlightId);

CREATE INDEX idx\_package\_packReturnFlightId ON tbl\_package (packReturnFlightId);

CREATE INDEX idx\_package\_packOutboundBusRouteId ON tbl\_package (packOutboundBusRouteId);

CREATE INDEX idx\_package\_packReturnBusRouteId ON tbl\_package (packReturnBusRouteId);

CREATE INDEX idx\_booking\_bookCustId ON tbl\_booking (bookCustId);

CREATE INDEX idx\_booking\_bookPackageId ON tbl\_booking (bookPackageId);

CREATE INDEX idx\_booking\_bookDiscountPercent ON tbl\_booking (bookDiscountPercent);

CREATE INDEX idx\_booking\_bookDiscountAmnt ON tbl\_booking (bookDiscountAmnt);

CREATE INDEX idx\_booking\_bookPrice ON tbl\_booking (bookPrice);

# Skrypt ładujący przykładowe dane do tabel

#### tbl\_continent

USE TRAVEL\_AGENCY

GO

INSERT INTO tbl\_continent (contName) VALUES ('Europe');

INSERT INTO tbl\_continent (contName) VALUES ('Asia');

INSERT INTO tbl\_continent (contName) VALUES ('North America');

INSERT INTO tbl\_continent (contName) VALUES ('South America');

INSERT INTO tbl\_continent (contName) VALUES ('Africa');

INSERT INTO tbl\_continent (contName) VALUES ('Australia');

#### tbl\_country

USE TRAVEL\_AGENCY

GO

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Albania', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Andorra', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Austria', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Belarus', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Belgium', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Bosnia and Herzegovina', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Bulgaria', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Croatia', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Cyprus', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Czech Republic', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Denmark', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Estonia', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Finland', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('France', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Germany', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Greece', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Hungary', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Iceland', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Ireland', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Italy', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Kosovo', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Latvia', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Liechtenstein', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Lithuania', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Luxembourg', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Malta', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Moldova', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Monaco', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Montenegro', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Netherlands', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('North Macedonia', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Norway', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Poland', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Portugal', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Romania', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Russia', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('San Marino', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Serbia', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Slovakia', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Slovenia', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Spain', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Sweden', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Switzerland', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Turkey', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Ukraine', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('United Kingdom', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Vatican City', 1);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Afghanistan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Armenia', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Azerbaijan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Bahrain', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Bangladesh', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Bhutan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Brunei', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Cambodia', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('China', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Georgia', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('India', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Indonesia', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Iran', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Iraq', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Israel', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Japan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Jordan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Kazakhstan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Kuwait', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Kyrgyzstan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Laos', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Lebanon', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Malaysia', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Maldives', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Mongolia', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Myanmar (Burma)', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Nepal', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('North Korea', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Oman', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Pakistan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Palestine', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Philippines', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Qatar', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Saudi Arabia', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Singapore', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('South Korea', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Sri Lanka', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Syria', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Taiwan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Tajikistan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Thailand', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Timor-Leste', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Turkey', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Turkmenistan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('United Arab Emirates', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Uzbekistan', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Vietnam', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Yemen', 2);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Antigua and Barbuda', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Bahamas', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Barbados', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Belize', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Canada', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Costa Rica', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Cuba', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Dominica', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Dominican Republic', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('El Salvador', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Grenada', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Guatemala', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Haiti', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Honduras', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Jamaica', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Mexico', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Nicaragua', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Panama', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Saint Kitts and Nevis', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Saint Lucia', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Saint Vincent and the Grenadines', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Trinidad and Tobago', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('United States', 3);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Argentina', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Bolivia', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Brazil', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Chile', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Colombia', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Ecuador', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Guyana', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Paraguay', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Peru', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Suriname', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Uruguay', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Venezuela', 4);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Algeria', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Angola', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Benin', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Botswana', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Burkina Faso', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Burundi', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Cabo Verde', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Cameroon', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Central African Republic', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Chad', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Comoros', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Congo', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Democratic Republic of the Congo', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Djibouti', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Egypt', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Equatorial Guinea', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Eritrea', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Eswatini', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Ethiopia', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Gabon', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Gambia', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Ghana', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Guinea', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Guinea-Bissau', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Ivory Coast', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Kenya', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Lesotho', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Liberia', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Libya', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Madagascar', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Malawi', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Mali', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Mauritania', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Mauritius', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Morocco', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Mozambique', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Namibia', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Niger', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Nigeria', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Rwanda', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('São Tomé and Príncipe', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Senegal', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Seychelles', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Sierra Leone', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Somalia', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('South Africa', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('South Sudan', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Sudan', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Tanzania', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Togo', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Tunisia', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Uganda', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Zambia', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Zimbabwe', 5);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Australia', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Fiji', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Kiribati', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Marshall Islands', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Micronesia', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Nauru', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('New Zealand', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Palau', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Papua New Guinea', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Samoa', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Solomon Islands', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Tonga', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Tuvalu', 6);

INSERT INTO tbl\_country (ctryName, contId) VALUES ('Vanuatu', 6);

#### tbl\_city

USE TRAVEL\_AGENCY

GO

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tirana', 1);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Durrës', 1);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Vlorë', 1);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Shkodër', 1);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Fier', 1);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Andorra la Vella', 2);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Vienna', 3);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Graz', 3);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Linz', 3);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Salzburg', 3);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Innsbruck', 3);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Minsk', 4);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Gomel', 4);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mogilev', 4);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Vitebsk', 4);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Hrodna', 4);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Brussels', 5);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Antwerp', 5);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ghent', 5);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Charleroi', 5);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Liège', 5);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Sarajevo', 6);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Banja Luka', 6);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tuzla', 6);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Zenica', 6);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mostar', 6);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Sofia', 7);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Plovdiv', 7);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Varna', 7);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Burgas', 7);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ruse', 7);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Zagreb', 8);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Split', 8);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Rijeka', 8);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Osijek', 8);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Zadar', 8);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Nicosia', 9);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Limassol', 9);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Prague', 10), ('Brno', 10), ('Ostrava', 10);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Copenhagen', 11), ('Aarhus', 11), ('Odense', 11);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tallinn', 12), ('Tartu', 12), ('Narva', 12);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Helsinki', 13), ('Tampere', 13), ('Turku', 13);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Paris', 14), ('Marseille', 14), ('Lyon', 14);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Berlin', 15), ('Hamburg', 15), ('Munich', 15);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Athens', 16), ('Thessaloniki', 16), ('Patras', 16);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Budapest', 17), ('Debrecen', 17), ('Szeged', 17);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Reykjavik', 18), ('Kopavogur', 18), ('Hafnarfjordur', 18);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Dublin', 19), ('Cork', 19), ('Galway', 19);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Rome', 20), ('Milan', 20), ('Naples', 20);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Pristina', 21), ('Prizren', 21), ('Gjakova', 21);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Riga', 22), ('Daugavpils', 22), ('Liepaja', 22);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Vaduz', 23), ('Schellenberg', 23), ('Triesenberg', 23);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Vilnius', 24), ('Kaunas', 24), ('Klaipeda', 24);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Luxembourg City', 25), ('Esch-sur-Alzette', 25), ('Dudelange', 25);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Valletta', 26), ('Birkirkara', 26), ('Mosta', 26);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Chisinau', 27), ('Tiraspol', 27), ('Balti', 27);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Monaco', 28), ('Monte Carlo', 28), ('La Condamine', 28);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Podgorica', 29), ('Niksic', 29), ('Herceg Novi', 29);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Amsterdam', 30), ('Rotterdam', 30), ('The Hague', 30);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Skopje', 31), ('Bitola', 31), ('Kumanovo', 31);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Oslo', 32), ('Bergen', 32), ('Trondheim', 32);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Warsaw', 33), ('Krakow', 33), ('Lodz', 33);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Lisbon', 34), ('Porto', 34), ('Vila Nova de Gaia', 34);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bucharest', 35), ('Cluj-Napoca', 35), ('Timisoara', 35);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Moscow', 36), ('Saint Petersburg', 36), ('Novosibirsk', 36);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('San Marino', 37), ('Serravalle', 37), ('Borgo Maggiore', 37);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Belgrade', 38), ('Novi Sad', 38), ('Nis', 38);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bratislava', 39), ('Kosice', 39), ('Presov', 39);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ljubljana', 40), ('Maribor', 40), ('Celje', 40);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Madrid', 41), ('Barcelona', 41), ('Valencia', 41);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Stockholm', 42), ('Gothenburg', 42), ('Malmö', 42);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Zurich', 43), ('Geneva', 43), ('Basel', 43);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Istanbul', 44), ('Ankara', 44), ('Izmir', 44);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kyiv', 45), ('Kharkiv', 45), ('Odessa', 45);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('London', 46), ('Manchester', 46), ('Birmingham', 46);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Vatican City', 47), ('Prati', 47), ('Borgo', 47);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kabul', 48);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kandahar', 48);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Yerevan', 49);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Baku', 50);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ganja', 50);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Manama', 51);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Dhaka', 52);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Chittagong', 52);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Thimphu', 53);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bandar Seri Begawan', 54);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Phnom Penh', 55);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Siem Reap', 55);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Beijing', 56);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Shanghai', 56);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tbilisi', 58);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('New Delhi', 59);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mumbai', 59);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Jakarta', 60);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Surabaya', 60);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tehran', 61);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mashhad', 61);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Baghdad', 62);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Jerusalem', 63);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tel Aviv', 63);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tokyo', 64);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Osaka', 64);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Amman', 65);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Nur-Sultan', 66);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Almaty', 66);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kuwait City', 67);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bishkek', 68);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Vientiane', 69);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Beirut', 70);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kuala Lumpur', 71);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('George Town', 71);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Malé', 72);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ulaanbaatar', 73);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Yangon', 74);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kathmandu', 75);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Pyongyang', 76);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Muscat', 77);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Islamabad', 78);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Karachi', 78);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ramallah', 79);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Manila', 80);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Cebu City', 80);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Doha', 81);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Riyadh', 82);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Jeddah', 82);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Singapore', 83);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Seoul', 84);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Busan', 84);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Colombo', 85);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Damascus', 86);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Taipei', 87);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Dushanbe', 88);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bangkok', 89);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Chiang Mai', 89);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Dili', 90);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ashgabat', 92);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Abu Dhabi', 93);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Dubai', 93);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tashkent', 94);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Hanoi', 95);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ho Chi Minh City', 95);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Sana''a', 96);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('St. John''s', 97);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Nassau', 98);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bridgetown', 99);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Belize City', 100);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ottawa', 101);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Toronto', 101);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('San Jose', 102);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Havana', 103);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Roseau', 104);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Santo Domingo', 105);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('San Salvador', 106);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('St. George''s', 107);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Guatemala City', 108);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Port-au-Prince', 109);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tegucigalpa', 110);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kingston', 111);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mexico City', 112);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Managua', 113);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Panama City', 114);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Basseterre', 115);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Castries', 116);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kingstown', 117);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Port of Spain', 118);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Washington, D.C.', 119);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('New York City', 119);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Buenos Aires', 120);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('La Paz', 121);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Sucre', 121);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Santa Cruz de la Sierra', 121);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Brasília', 122);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Rio de Janeiro', 122);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('São Paulo', 122);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Santiago', 123);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Valparaíso', 123);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Concepción', 123);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bogotá', 124);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Medellín', 124);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Cali', 124);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Quito', 125);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Guayaquil', 125);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Cuenca', 125);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Georgetown', 126);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Asunción', 127);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ciudad del Este', 127);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Encarnación', 127);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Lima', 128);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Arequipa', 128);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Cusco', 128);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Paramaribo', 129);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Montevideo', 130);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Caracas', 131);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Maracaibo', 131);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Valencia', 131);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Algiers', 132);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Luanda', 133);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Huambo', 133);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Lobito', 133);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Porto-Novo', 134);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Cotonou', 134);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Parakou', 134);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Gaborone', 135);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Francistown', 135);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Molepolole', 135);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ouagadougou', 136);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bobo-Dioulasso', 136);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Koudougou', 136);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bujumbura', 137);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Praia', 138);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mindelo', 138);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Santa Maria', 138);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Yaoundé', 139);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Douala', 139);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bamenda', 139);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bangui', 140);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bimbo', 140);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Berbérati', 140);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('N''Djamena', 141);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Moundou', 141);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Sarh', 141);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Moroni', 142);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Brazzaville', 143);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Pointe-Noire', 143);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Dolisie', 143);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kinshasa', 144);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Lubumbashi', 144);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mbuji-Mayi', 144);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Djibouti City', 145);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Cairo', 146);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Alexandria', 146);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Giza', 146);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Malabo', 147);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bata', 147);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ebebiyin', 147);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Asmara', 148);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Keren', 148);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Massawa', 148);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mbabane', 149);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Manzini', 149);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Addis Ababa', 150);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Dire Dawa', 150);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mekelle', 150);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Libreville', 151);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Port-Gentil', 151);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Franceville', 151);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Banjul', 152);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Serekunda', 152);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Accra', 153);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kumasi', 153);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tamale', 153);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Conakry', 154);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Nzérékoré', 154);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kankan', 154);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bissau', 155);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Abidjan', 156);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bouaké', 156);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Yamoussoukro', 156);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Nairobi', 157);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mombasa', 157);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kisumu', 157);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Maseru', 158);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Monrovia', 159);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tripoli', 160);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Benghazi', 160);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Misrata', 160);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Antananarivo', 161);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Toamasina', 161);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Antsirabe', 161);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Lilongwe', 162);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Blantyre', 162);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mzuzu', 162);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bamako', 163);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Sikasso', 163);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mopti', 163);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Nouakchott', 164);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Nouadhibou', 164);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Port Louis', 165);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Rabat', 166);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Casablanca', 166);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Fes', 166);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Maputo', 167);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Beira', 167);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Nampula', 167);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Windhoek', 168);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Walvis Bay', 168);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Niamey', 169);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Zinder', 169);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tahoua', 169);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Lagos', 170);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kano', 170);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ibadan', 170);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kigali', 171);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('São Tomé', 172);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Dakar', 173);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Thiès', 173);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kaolack', 173);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Victoria', 174);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Freetown', 175);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mogadishu', 176);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Hargeisa', 176);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bosaso', 176);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Johannesburg', 177);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Cape Town', 177);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Durban', 177);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Juba', 178);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Khartoum', 179);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Omdurman', 179);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Port Sudan', 179);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Dar es Salaam', 180);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Dodoma', 180);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mwanza', 180);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Lomé', 181);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Sokodé', 181);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kara', 181);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Tunis', 182);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Sfax', 182);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Sousse', 182);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kampala', 183);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Gulu', 183);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Lira', 183);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Lusaka', 184);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kitwe', 184);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ndola', 184);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Harare', 185);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bulawayo', 185);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Chitungwiza', 185);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Sydney', 186);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Melbourne', 186);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Brisbane', 186);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Suva', 187);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Nadi', 187);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Lautoka', 187);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('South Tarawa', 188);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Betio', 188);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Bairiki', 188);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Majuro', 189);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ebeye', 189);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Arno', 189);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Palikir', 190);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Weno', 190);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Kolonia', 190);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Yaren', 191);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Auckland', 192);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Wellington', 192);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Christchurch', 192);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Ngerulmud', 193);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Koror', 193);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Port Moresby', 194);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Lae', 194);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Mount Hagen', 194);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Apia', 195);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Vaitele', 195);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Faleolo', 195);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Honiara', 196);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Gizo', 196);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Auki', 196);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Nukuʻalofa', 197);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Vaini', 197);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Haveluloto', 197);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Funafuti', 198);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Fongafale', 198);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Port Vila', 199);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Luganville', 199);

INSERT INTO tbl\_city (cityName, ctryId) VALUES ('Norsup', 199);

#### tbl\_bus

USE TRAVEL\_AGENCY

GO

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('ABC123', 'Mercedes-Benz Sprinter', 'White');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('DEF456', 'Volvo 9700', 'Blue');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('GHI789', 'MAN Lion''s Coach', 'Red');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('JKL012', 'Scania Touring', 'Green');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('MNO345', 'Setra S 431 DT', 'Yellow');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('PQR678', 'Van Hool TX', 'Silver');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('STU901', 'Irizar i8', 'Black');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('VWX234', 'Temsa Prestij', 'Orange');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('YZA567', 'Neoplan Skyliner', 'Brown');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('BCD890', 'MCI J4500', 'Gray');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('EFG123', 'Setra S 417 TC', 'White');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('HIJ456', 'Volvo B11R', 'Blue');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('KLM789', 'MAN Lion''s Intercity', 'Red');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('NOP012', 'Scania Higer Touring', 'Green');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('PQR345', 'Irizar i6', 'Yellow');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('STU678', 'Mercedes-Benz Tourismo', 'Silver');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('VWX901', 'Van Hool TX Altano', 'Black');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('YZA234', 'Temsa MD9', 'Orange');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('BCD567', 'Neoplan Starliner', 'Brown');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('EFG890', 'Irizar i4', 'Gray');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('HIJ123', 'Setra S 511 HD', 'White');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('KLM456', 'Volvo B8R', 'Blue');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('NOP789', 'MAN Lion''s Regio', 'Red');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('PQR012', 'Scania Interlink HD', 'Green');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('STU345', 'Mercedes-Benz OC 500 LE', 'Yellow');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('VWX678', 'Temsa Avenue LF', 'Silver');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('YZA901', 'Solaris InterUrbino', 'Black');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('BCD234', 'Iveco Crossway LE', 'Orange');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('EFG567', 'MAN Lion''s City', 'Brown');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('HIJ890', 'Volvo 7900 Hybrid', 'Gray');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('KLM123', 'Mercedes-Benz Citaro', 'White');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('NOP456', 'Solaris Urbino', 'Blue');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('PQR789', 'VDL Citea', 'Red');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('STU012', 'Iveco Urbanway', 'Green');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('VWX345', 'MAN Lion''s City A23', 'Yellow');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('YZA678', 'Volvo 7900 Electric', 'Silver');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('BCD901', 'Scania Citywide LF', 'Black');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('EFG234', 'Mercedes-Benz Conecto', 'Orange');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('HIJ567', 'MAN Lion''s City G', 'Brown');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('KLM890', 'Volvo 9900', 'Gray');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('NOP123', 'Iveco Bus Crossway', 'White');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('PQR456', 'MAN Lion''s Coach', 'Blue');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('STU789', 'Mercedes-Benz Intouro', 'Red');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('VWX012', 'Iveco Bus Magelys', 'Green');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('YZA345', 'Van Hool EX', 'Yellow');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('BCD678', 'VDL Futura', 'Silver');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('EFG901', 'Scania Touring HD', 'Black');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('HIJ234', 'Volvo 9700 DD', 'Orange');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('KLM567', 'Setra S 531 DT', 'Brown');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('NOP890', 'Neoplan Cityliner', 'Gray');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('PQR123', 'Irizar i8 SC5', 'White');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('STU456', 'MAN Lion''s Coach L', 'Blue');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('VWX789', 'Mercedes-Benz Tourismo M', 'Red');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('YZA012', 'Van Hool TX Altano', 'Green');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('BCD345', 'Temsa Prestij SX', 'Yellow');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('EFG678', 'Neoplan Tourliner', 'Silver');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('HIJ901', 'Scania Higer Touring', 'Black');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('KLM234', 'Irizar i8 HV', 'Orange');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('NOP567', 'Setra S 511 HD', 'Brown');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('PQR890', 'MAN Lion''s Coach C', 'Gray');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('STU123', 'Mercedes-Benz OC 500 LE', 'White');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('VWX456', 'Temsa MD9 LE', 'Blue');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('YZA789', 'Solaris InterUrbino', 'Red');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('BCD012', 'Iveco Crossway LE', 'Green');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('EFG345', 'MAN Lion''s City', 'Yellow');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('HIJ678', 'Volvo 7900 Hybrid', 'Silver');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('KLM901', 'Mercedes-Benz Citaro', 'Black');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('NOP234', 'Solaris Urbino', 'Orange');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('PQR567', 'VDL Citea Electric', 'Brown');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('STU890', 'Iveco Urbanway Hybrid', 'Gray');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('VWX123', 'MAN Lion''s City A37', 'White');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('YZA456', 'Volvo 7900 Electric', 'Blue');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('BCD789', 'Scania Citywide LF', 'Red');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('EFG012', 'Mercedes-Benz Conecto LF', 'Green');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('HIJ345', 'MAN Lion''s City G LNG', 'Yellow');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('KLM678', 'Volvo 9900 DD', 'Silver');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('NOP901', 'Iveco Bus Crossway NP', 'Black');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('PQR234', 'MAN Lion''s Regio LE', 'Orange');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('STU567', 'Mercedes-Benz Intouro M', 'Brown');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('VWX890', 'Iveco Bus Magelys Pro', 'Gray');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('YZA123', 'Van Hool EX LNG', 'White');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('BCD456', 'VDL Futura FDD2', 'Blue');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('EFG789', 'Scania Touring HD', 'Red');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('HIJ012', 'Volvo 9700 DD Plus', 'Green');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('KLM345', 'Setra S 531 DT', 'Yellow');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('NOP678', 'Neoplan Cityliner N', 'Silver');

INSERT INTO tbl\_bus (busNo, busModel, busColor) VALUES ('PQR901', 'Irizar i8 SC5', 'Black');

#### tbl\_driver

USE TRAVEL\_AGENCY

GO

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('John', 'Smith', '+1234567890');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Michael', 'Johnson', '+1987654321');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('James', 'Williams', '+1555666777');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('David', 'Jones', '+1777888999');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Robert', 'Brown', '+1888777666');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('William', 'Davis', '+1666555444');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Joseph', 'Miller', '+1999444333');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Charles', 'Wilson', '+1222333444');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Thomas', 'Taylor', '+1333444555');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Christopher', 'Anderson', '+1444555666');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Daniel', 'White', '+1555666777');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Matthew', 'Harris', '+1666777888');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Anthony', 'Martin', '+1777888999');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Donald', 'Thompson', '+1888999000');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Mark', 'Garcia', '+1999000111');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Paul', 'Martinez', '+1222111222');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Steven', 'Robinson', '+1333222333');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Andrew', 'Clark', '+1444333444');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Kenneth', 'Rodriguez', '+1555444555');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Joshua', 'Lewis', '+1666555666');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Kevin', 'Lee', '+1777666777');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Brian', 'Walker', '+1888777888');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('George', 'Hall', '+1999888999');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Edward', 'Allen', '+1222999000');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Ronald', 'Young', '+1333101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Timothy', 'Hernandez', '+1444101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Jason', 'King', '+1555101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Jeffrey', 'Wright', '+1666101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Ryan', 'Lopez', '+1777101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Jacob', 'Hill', '+1888101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Gary', 'Scott', '+1999101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Nicholas', 'Green', '+1220101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Eric', 'Adams', '+1331101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Stephen', 'Baker', '+1442101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Jonathan', 'Gonzalez', '+1553101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Larry', 'Nelson', '+1664101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Justin', 'Carter', '+1775101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Scott', 'Mitchell', '+1886101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Brandon', 'Perez', '+1997101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Frank', 'Roberts', '+1228101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Benjamin', 'Turner', '+1339101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Gregory', 'Phillips', '+1449101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Raymond', 'Campbell', '+1559101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Samuel', 'Parker', '+1660101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Patrick', 'Evans', '+1771101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Alexander', 'Edwards', '+1882101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Jack', 'Collins', '+1993101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Dennis', 'Stewart', '+1224101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Jerry', 'Flores', '+1335101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Tyler', 'Morris', '+1446101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Aaron', 'Nguyen', '+1557101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Jose', 'Murphy', '+1668101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Henry', 'Rivera', '+1779101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Adam', 'Cook', '+1880101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Zachary', 'Rogers', '+1991101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Nathan', 'Morgan', '+1222101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Walter', 'Peterson', '+1333101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Harold', 'Cooper', '+1444101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Douglas', 'Reed', '+1555101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Peter', 'Bailey', '+1666101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Kyle', 'Sanchez', '+1777101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Roger', 'Morales', '+1888101010');

INSERT INTO tbl\_driver (driverName, driverSurName, driverPhone) VALUES ('Keith', 'Sanders', '+1999101010');

#### tbl\_guide

**Part of the script** was presented, over 450 records have been added

USE TRAVEL\_AGENCY

GO

-- Kabul

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Ahmad', 'Khan', 1, '+93 000-0051');

-- Kandahar

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Mohammad', 'Ahmadi', 2, '+93 000-0052');

-- Herat

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Fatima', 'Nouri', 3, '+93 000-0053');

-- Yerevan

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Arman', 'Sargsyan', 4, '+374 000-0054');

-- Baku

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Elmira', 'Aliyeva', 5, '+994 000-0055');

-- Ganja

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Rashad', 'Mammadov', 6, '+994 000-0056');

-- Manama

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Fatima', 'Al Khalifa', 7, '+973 000-0057');

-- Dhaka

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Mohammad', 'Rahman', 8, '+880 000-0058');

-- Chittagong

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Nadia', 'Akter', 9, '+880 000-0059');

-- Thimphu

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Tenzin', 'Dorji', 10, '+975 000-0060');

-- Bandar Seri Begawan

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Mohammad', 'Abdullah', 11, '+673 000-0061');

-- Phnom Penh

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Sokha', 'Heng', 12, '+855 000-0062');

-- Siem Reap

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Sopheak', 'Chea', 13, '+855 000-0063');

-- Beijing

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Wei', 'Li', 14, '+86 000-0064');

-- Shanghai

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Yan', 'Wang', 15, '+86 000-0065');

-- Nicosia

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Nikos', 'Christou', 16, '+357 000-0066');

-- Limassol

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Maria', 'Papadopoulou', 17, '+357 000-0067');

-- Tbilisi

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Giorgi', 'Gogoladze', 18, '+995 000-0068');

-- New Delhi

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Raj', 'Kumar', 19, '+91 000-0069');

-- Mumbai

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Priya', 'Sharma', 20, '+91 000-0070');

-- Jakarta

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Agus', 'Susanto', 21, '+62 000-0071');

-- Surabaya

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Siti', 'Wahyuni', 22, '+62 000-0072');

-- Tehran

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Ali', 'Najafi', 23, '+98 000-0073');

-- Mashhad

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Zahra', 'Rahmani', 24, '+98 000-0074');

-- Baghdad

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Hassan', 'Abdul', 25, '+964 000-0075');

-- Jerusalem

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('David', 'Cohen', 26, '+972 000-0076');

-- Tel Aviv

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Tamar', 'Levi', 27, '+972 000-0077');

-- Tokyo

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Yuki', 'Sato', 28, '+81 000-0078');

-- Osaka

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Ryota', 'Takahashi', 29, '+81 000-0079');

-- Amman

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Fatima', 'Mustafa', 30, '+962 000-0080');

-- Nur-Sultan

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Nurzhan', 'Akhmetov', 31, '+7 000-0081');

-- Almaty

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Aigerim', 'Kazakhbayeva', 32, '+7 000-0082');

-- Kuwait City

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Ahmed', 'Al Fahad', 33, '+965 000-0083');

-- Bishkek

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Asel', 'Toktobekova', 34, '+996 000-0084');

-- Vientiane

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Phouvieng', 'Xayasone', 35, '+856 000-0085');

-- Beirut

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Nabil', 'Fahed', 36, '+961 000-0086');

-- Kuala Lumpur

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Hafiz', 'Ismail', 37, '+60 000-0087');

-- George Town

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Siew', 'Chong', 38, '+60 000-0088');

-- Malé

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Ahmed', 'Mohamed', 39, '+960 000-0089');

-- Ulaanbaatar

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Bat-Erdene', 'Battulga', 40, '+976 000-0090');

-- Yangon

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Aung', 'Win', 41, '+95 000-0091');

-- Kathmandu

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Nisha', 'Shrestha', 42, '+977 000-0092');

-- Pyongyang

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Jin', 'Kim', 43, '+850 000-0093');

-- Muscat

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Youssef', 'Al-Balushi', 44, '+968 000-0094');

-- Islamabad

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Faisal', 'Malik', 45, '+92 000-0095');

-- Karachi

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Sana', 'Khan', 46, '+92 000-0096');

-- Ramallah

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Ahmad', 'Salam', 47, '+970 000-0097');

-- Manila

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Maria', 'Gonzalez', 48, '+63 000-0098');

-- Cebu City

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('John', 'Santos', 49, '+63 000-0099');

-- Doha

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Ali', 'Al Thani', 50, '+974 000-0100');

-- Riyadh

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Ahmed', 'Al-Farsi', 51, '+966 000-0124');

-- Jeddah

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Fatima', 'Al-Mansouri', 52, '+966 000-0125');

-- Singapore

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Wei', 'Tan', 53, '+65 000-0126');

-- Seoul

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Ji-hoon', 'Kim', 54, '+82 000-0127');

-- Busan

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Min-ji', 'Lee', 55, '+82 000-0128');

-- tbl\_guide

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Rohan', 'Silva', 56, '+94 000-0129');

-- Damascus

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Fatima', 'Abdul', 57, '+963 000-0130');

-- Taipei

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Wei', 'Chen', 58, '+886 000-0131');

-- Dushanbe

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Said', 'Nazarov', 59, '+992 000-0132');

-- Bangkok

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Somchai', 'Sukhumvit', 60, '+66 000-0133');

-- Chiang Mai

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Naree', 'Boonchai', 61, '+66 000-0134');

-- Dili

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Jose', 'Gomes', 62, '+670 000-0135');

-- Ankara

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Mustafa', 'Yilmaz', 63, '+90 000-0136');

-- Istanbul

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Ayse', 'Kaya', 64, '+90 000-0137');

-- Ashgabat

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Serdar', 'Akhmedov', 65, '+993 000-0138');

-- Abu Dhabi

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Fatima', 'Al-Mansouri', 66, '+971 000-0139');

-- Dubai

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Ahmed', 'Al-Maktoum', 67, '+971 000-0140');

-- Tashkent

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Akmal', 'Usmanov', 68, '+998 000-0141');

-- Hanoi

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Pham', 'Nguyen', 69, '+84 000-0142');

-- Ho Chi Minh City

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Minh', 'Tran', 70, '+84 000-0143');

-- Sana'a

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Youssef', 'Al-Habib', 71, '+967 000-0144');

-- St. John's

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Jason', 'Anderson', 72, '+1 000-0145');

-- St. John's

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Michael', 'Smith', 72, '+1 000-0101');

-- Nassau

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Jennifer', 'Brown', 73, '+1 000-0102');

-- Bridgetown

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Marcus', 'Jones', 74, '+1 000-0103');

-- Belize City

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Karen', 'Garcia', 75, '+501 000-0104');

-- Ottawa

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('David', 'Martin', 76, '+1 000-0105');

-- Toronto

INSERT INTO tbl\_guide (guideName, guideSurname, guideCityId, guidePhone)

VALUES ('Emily', 'Taylor', 77, '+1 000-0106');

#### tbl\_address

**Part of the script** was presented, over 550 records have been added

USE TRAVEL\_AGENCY

GO

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (1, '1001', 'Kabul', 'Main Street', '123');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (2, '2001', 'Kandahar', 'First Avenue', '456');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (3, '3001', 'Herat', 'Oak Street', '789');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (4, '375010', 'Yerevan', 'Sunset Boulevard', '456');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (5, 'AZ1000', 'Baku', 'Moonlight Avenue', '789');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (6, 'AZ2000', 'Ganja', 'Rainbow Street', '1011');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (7, '333', 'Manama', 'Palm Street', '1213');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (8, '1212', 'Dhaka', 'River Road', '1415');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (9, '4000', 'Chittagong', 'Ocean Avenue', '1617');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (10, '11001', 'Thimphu', 'Mountain Street', '1819');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (11, 'BA1919', 'Bandar Seri Begawan', 'Golden Avenue', '2021');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (12, '12000', 'Phnom Penh', 'Royal Street', '2223');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (13, '17000', 'Siem Reap', 'Angkor Street', '2425');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (14, '100000', 'Beijing', 'Forbidden City Road', '2627');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (15, '200000', 'Shanghai', 'Pearl Tower Street', '2829');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (16, '1500', 'Nicosia', 'Castle Avenue', '3031');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (17, '3105', 'Limassol', 'Beach Street', '3233');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (18, '0100', 'Tbilisi', 'Rustaveli Avenue', '3435');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (19, '110001', 'New Delhi', 'Gandhi Street', '3637');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (20, '400001', 'Mumbai', 'Bollywood Avenue', '3839');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (21, '10110', 'Jakarta', 'Jalan Thamrin', '4041');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (22, '60175', 'Surabaya', 'Tunjungan Street', '4243');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (23, '19395', 'Tehran', 'Valiasr Avenue', '4445');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (24, '91911', 'Mashhad', 'Imam Reza Street', '4647');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (25, '10001', 'Baghdad', 'Revolution Street', '4849');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (26, '91140', 'Jerusalem', 'Old City Street', '5051');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (27, '6357345', 'Tel Aviv', 'Beach Boulevard', '5253');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (28, '100-0001', 'Tokyo', 'Shibuya Street', '5455');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (29, '540-0002', 'Osaka', 'Dotonbori Avenue', '5657');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (30, '11190', 'Amman', 'Rainbow Street', '5859');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (31, '010000', 'Nur-Sultan', 'Kazakh Street', '6061');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (32, '050000', 'Almaty', 'Mountainside Avenue', '6263');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (33, '15453', 'Kuwait City', 'Desert Street', '6465');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (34, '720000', 'Bishkek', 'Kyrgyz Street', '6667');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (35, '01000', 'Vientiane', 'Mekong Avenue', '6869');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (36, '2073 0808', 'Beirut', 'Corniche Street', '7071');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (37, '50100', 'Kuala Lumpur', 'Petronas Avenue', '7273');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (38, '10200', 'George Town', 'Penang Street', '7475');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (39, '20055', 'Malé', 'Lagoon Street', '7677');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (40, '160725', 'Ulaanbaatar', 'Khan-Uul Street', '7879');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (41, '11191', 'Yangon', 'Shwedagon Avenue', '8081');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (42, '44600', 'Kathmandu', 'Durbar Marg Street', '8283');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (43, '100100', 'Pyongyang', 'Juche Avenue', '8485');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (44, '100', 'Muscat', 'Corniche Street', '8687');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (45, '44000', 'Islamabad', 'Margalla Avenue', '8889');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (46, '74000', 'Karachi', 'Clifton Road', '9091');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (47, '19991', 'Ramallah', 'Main Street', '9293');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (48, '1006', 'Manila', 'Roxas Boulevard', '9495');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (49, '6000', 'Cebu City', 'Fuente Osmeña Avenue', '9697');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (50, 'NA', 'Doha', 'Corniche Street', '9899');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (51, '00265', 'Basseterre', 'Bay Road', '101');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (52, 'LC01 101', 'Castries', 'John Compton Highway', '103');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (53, 'VC0100', 'Kingstown', 'Bay Street', '105');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (54, '11102', 'Port of Spain', 'Independence Square', '107');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (55, '20001', 'Washington, D.C.', 'Pennsylvania Avenue', '109');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (56, '10001', 'New York City', 'Broadway', '111');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (57, '00109', 'Roseau', 'Kennedy Avenue', '123');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (58, '10104', 'Santo Domingo', 'Calle El Conde', '125');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (59, 'CP 1101', 'San Salvador', 'Avenida El Espino', '127');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (60, '01', 'St. George''s', 'Market Square', '129');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (61, '01001', 'Guatemala City', '6th Avenue', '131');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (62, '6110', 'Port-au-Prince', 'Rue Capois', '133');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (63, '10101', 'Tegucigalpa', 'Bulevar Suyapa', '135');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (64, '10', 'Kingston', 'King Street', '137');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (65, '01000', 'Mexico City', 'Paseo de la Reforma', '139');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (66, '10101', 'Managua', 'Avenida Bolívar', '141');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (67, '0831-0184', 'Panama City', 'Avenida Balboa', '143');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (73, 'N-3718', 'Nassau', 'Bay Street', '145');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (74, 'BB14027', 'Bridgetown', 'Broad Street', '147');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (75, '501', 'Belize City', 'Albert Street', '149');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (75, '501', 'Belize City', 'Albert Street', '149');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (76, 'K1P 1A4', 'Ottawa', 'Wellington Street', '151');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (77, 'M5H 2N2', 'Toronto', 'Yonge Street', '153');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (78, '10101', 'San Jose', 'Avenida Central', '155');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (79, '10200', 'Havana', 'Paseo del Prado', '157');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (80, '00109', 'Roseau', 'Kennedy Avenue', '159');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (81, '10104', 'Santo Domingo', 'Calle El Conde', '161');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (82, 'CP 1101', 'San Salvador', 'Avenida El Espino', '163');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (83, '01', 'St. George''s', 'Market Square', '165');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (84, '01001', 'Guatemala City', '6th Avenue', '167');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (85, '6110', 'Port-au-Prince', 'Rue Capois', '169');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (86, '10101', 'Tegucigalpa', 'Bulevar Suyapa', '171');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (87, '10', 'Kingston', 'King Street', '173');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (88, '01000', 'Mexico City', 'Paseo de la Reforma', '175');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (89, '10101', 'Managua', 'Avenida Bolívar', '177');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (90, '0831-0184', 'Panama City', 'Avenida Balboa', '179');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (91, 'KN0200', 'Basseterre', 'Bank Street', '181');

INSERT INTO tbl\_address (addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (92, 'LC01 101', 'Castries', 'Micoud Street', '183');

#### tbl\_customer

USE TRAVEL\_AGENCY

GO

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('John', 'Doe', '+93 123456789', 'johndoe@example.com', 1);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Jane', 'Smith', '+93 987654321', 'janesmith@example.com', 2);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Alice', 'Johnson', '+93 555555555', 'alicejohnson@example.com', 3);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Bob', 'Brown', '+374 11223344', 'bobbrown@example.com', 4);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Emily', 'Davis', '+994 555555555', 'emilydavis@example.com', 5);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Michael', 'Martinez', '+994 987654321', 'michaelmartinez@example.com', 6);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Sophia', 'Garcia', '+973 12345678', 'sophiagarcia@example.com', 7);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('James', 'Wilson', '+880 11223344', 'jameswilson@example.com', 8);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Olivia', 'Taylor', '+880 555555555', 'oliviataylor@example.com', 9);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('William', 'Anderson', '+975 987654321', 'williamanderson@example.com', 10);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Ava', 'Thomas', '+673 12345678', 'avathomas@example.com', 11);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Noah', 'Hernandez', '+855 11223344', 'noahhernandez@example.com', 12);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Isabella', 'Young', '+855 555555555', 'isabellayoung@example.com', 13);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Liam', 'Lee', '+86 123456789', 'liamlee@example.com', 14);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Mia', 'Lewis', '+86 987654321', 'mialewis@example.com', 15);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Lucas', 'Hall', '+357 11223344', 'lucashall@example.com', 16);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Amelia', 'Allen', '+357 555555555', 'ameliaallen@example.com', 17);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Henry', 'Green', '+995 12345678', 'henrygreen@example.com', 18);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Ella', 'Baker', '+91 987654321', 'ellabaker@example.com', 19);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Oliver', 'Rossi', '+91 1122334455', 'oliverrossi@example.com', 20);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Avery', 'Romero', '+62 555555555', 'averyromero@example.com', 21);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Lily', 'Lee', '+62 123456789', 'lilylee@example.com', 22);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Jack', 'Yang', '+98 987654321', 'jackyang@example.com', 23);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Charlotte', 'Wang', '+98 11223344', 'charlottewang@example.com', 24);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Alexander', 'Chen', '+964 555555555', 'alexanderchen@example.com', 25);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Amelia', 'Zhang', '+972 123456789', 'ameliazhang@example.com', 26);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Michael', 'Liu', '+972 987654321', 'michaelliu@example.com', 27);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Sophia', 'Lopez', '+81 1122334455', 'sophialopez@example.com', 28);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Daniel', 'Gonzalez', '+81 555555555', 'danielgonzalez@example.com', 29);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Olivia', 'Hernandez', '+962 12345678', 'oliviahernandez@example.com', 30);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Logan', 'Martin', '+7 987654321', 'loganmartin@example.com', 31);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Amelia', 'Thompson', '+7 11223344', 'amelia.thompson@example.com', 32);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Ethan', 'White', '+965 555555555', 'ethanwhite@example.com', 33);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Evelyn', 'Lopez', '+996 123456789', 'evelynlopez@example.com', 34);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Aiden', 'Harris', '+856 987654321', 'aidenharris@example.com', 35);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Mia', 'Young', '+961 555555555', 'miayoung@example.com', 36);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('James', 'Lee', '+60 1122334455', 'james.lee@example.com', 37);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Emily', 'Garcia', '+60 987654321', 'emily.garcia@example.com', 38);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Noah', 'Martinez', '+960 12345678', 'noahmartinez@example.com', 39);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Sophia', 'Rodriguez', '+976 555555555', 'sophiarodriguez@example.com', 40);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Ethan', 'Hernandez', '+95 11223344', 'ethanhernandez@example.com', 41);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Ava', 'Thompson', '+977 987654321', 'avathompson@example.com', 42);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Aiden', 'White', '+850 555555555', 'aidenwhite@example.com', 43);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Olivia', 'Harris', '+968 12345678', 'oliviaharris@example.com', 44);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Logan', 'Gonzalez', '+92 987654321', 'logangonzalez@example.com', 45);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Evelyn', 'Wang', '+92 11223344', 'evelynwang@example.com', 46);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Daniel', 'Zhang', '+970 555555555', 'danielzhang@example.com', 47);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Ava', 'Liu', '+63 123456789', 'avaliu@example.com', 48);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Michael', 'Garcia', '+63 987654321', 'michaelgarcia@example.com', 49);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Olivia', 'Rodriguez', '+974 11223344', 'oliviarodriguez@example.com', 50);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Sophia', 'Lopez', '+1 869-000-0001', 'sophialopez@example.com', 51);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Daniel', 'Gonzalez', '+1 758-000-0002', 'danielgonzalez@example.com', 52);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Olivia', 'Hernandez', '+1 784-000-0003', 'oliviahernandez@example.com', 53);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Logan', 'Martin', '+1 868-000-0004', 'loganmartin@example.com', 54);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Amelia', 'Thompson', '+1 202-000-0005', 'amelia.thompson@example.com', 55);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Ethan', 'White', '+1 212-000-0006', 'ethanwhite@example.com', 56);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Liam', 'Harris', '+1 767-000-0012', 'liamharris@example.com', 57);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Aria', 'Martinez', '+1 809-000-0013', 'ariamartinez@example.com', 58);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Lucas', 'Gonzalez', '+503 00-000-0014', 'lucasgonzalez@example.com', 59);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Harper', 'Lopez', '+1 473-000-0015', 'harperlopez@example.com', 60);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Evelyn', 'Rodriguez', '+502 0000-0016', 'evelynrodriguez@example.com', 61);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Oliver', 'Martinez', '+509 0000-0017', 'olivermartinez@example.com', 62);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Amelia', 'Garcia', '+504 0000-0018', 'ameliagarcia@example.com', 63);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('William', 'Young', '+1 876-000-0019', 'williamyoung@example.com', 64);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Abigail', 'Hernandez', '+52 55 0000-0020', 'abigailhernandez@example.com', 65);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Mason', 'Gonzalez', '+505 0000-0021', 'masongonzalez@example.com', 66);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Emily', 'Rodriguez', '+507 0000-0022', 'emilyrodriguez@example.com', 67);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Michael', 'Harris', '+1 242-000-0023', 'michaelharris@example.com', 68);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Sophia', 'Martinez', '+1 246-000-0024', 'sophiamartinez@example.com', 69);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Alexander', 'Young', '+501 000-0025', 'alexanderyoung@example.com', 70);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Alexander', 'Young', '+501 000-0025', 'alexanderyoung@example.com', 71);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Sofia', 'Gonzalez', '+1 613-000-0026', 'sofiagonzalez@example.com', 72);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Benjamin', 'Rodriguez', '+1 416-000-0027', 'benjaminrodriguez@example.com', 73);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Charlotte', 'Hernandez', '+506 0000-0028', 'charlottehernandez@example.com', 74);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Henry', 'Young', '+53 0000-0029', 'henryyoung@example.com', 75);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Madison', 'Martinez', '+1 767-000-0030', 'madisonmartinez@example.com', 76);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Ethan', 'Rodriguez', '+1 809-000-0031', 'ethanrodriguez@example.com', 77);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Scarlett', 'Gonzalez', '+503 00-000-0032', 'scarlettgonzalez@example.com', 78);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Lucas', 'Young', '+1 473-000-0033', 'lucasyoung@example.com', 79);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Chloe', 'Hernandez', '+502 0000-0034', 'chloehernandez@example.com', 80);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Jackson', 'Martinez', '+509 0000-0035', 'jacksonmartinez@example.com', 81);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Madison', 'Gonzalez', '+504 0000-0036', 'madisongonzalez@example.com', 82);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Sebastian', 'Young', '+1 876-000-0037', 'sebastianyoung@example.com', 83);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Sophia', 'Rodriguez', '+52 55 0000-0038', 'sophiarodriguez@example.com', 84);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Henry', 'Garcia', '+505 0000-0039', 'henrygarcia@example.com', 85);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Emily', 'Young', '+507 0000-0040', 'emilyyoung@example.com', 86);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Mason', 'Martinez', '+1 869-000-0041', 'masonmartinez@example.com', 87);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Oliver', 'Rodriguez', '+1 758-000-0042', 'oliverrodriguez@example.com', 88);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Charlotte', 'Martinez', '+1 784-000-0043', 'charlottemartinez@example.com', 89);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Jackson', 'Gonzalez', '+1 868-000-0044', 'jacksongonzalez@example.com', 90);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Madison', 'Young', '+1 202-000-0045', 'madisonyoung@example.com', 91);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Ethan', 'Rodriguez', '+1 212-000-0046', 'ethanrodriguez@example.com', 92);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Ava', 'Martinez', '+420 000-000-007', 'avamartinez@example.com', 93);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Noah', 'Young', '+420 000-000-008', 'noahyoung@example.com', 94);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Mia', 'Garcia', '+420 000-000-009', 'miagarcia@example.com', 95);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('James', 'Rodriguez', '+45 00-000-0010', 'jamesrodriguez@example.com', 96);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES ('Charlotte', 'Lopez', '+45 00-000-0011', 'charlottelopez@example.com', 97);

INSERT INTO tbl\_customer (custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES

('John', 'Doe', '+1234567890', 'john.doe@example.com', 198),

('Alice', 'Smith', '+1987654321', 'alice.smith@example.com', 199),

('Michael', 'Johnson', '+1654321879', 'michael.johnson@example.com', 200),

('Emily', 'Brown', '+1765432987', 'emily.brown@example.com', 201),

('Christopher', 'Taylor', '+1897654321', 'christopher.taylor@example.com', 202),

('Jessica', 'Martinez', '+1238765432', 'jessica.martinez@example.com', 203),

('Matthew', 'Anderson', '+1564321098', 'matthew.anderson@example.com', 204),

('Emma', 'Wilson', '+1675432109', 'emma.wilson@example.com', 205),

('Daniel', 'Thomas', '+1898765432', 'daniel.thomas@example.com', 206),

('Olivia', 'Lopez', '+1298765432', 'olivia.lopez@example.com', 207),

('William', 'Harris', '+1432109876', 'william.harris@example.com', 208),

('Sophia', 'Clark', '+1543210987', 'sophia.clark@example.com', 209),

('James', 'Lewis', '+1765432109', 'james.lewis@example.com', 210),

('Ava', 'Lee', '+1987654321', 'ava.lee@example.com', 211),

('Alexander', 'Walker', '+1298765432', 'alexander.walker@example.com', 212),

('Mia', 'Perez', '+1654321098', 'mia.perez@example.com', 213),

('Benjamin', 'Moore', '+1765432109', 'benjamin.moore@example.com', 214),

('Charlotte', 'Gonzalez', '+1897654321', 'charlotte.gonzalez@example.com', 215),

('Ethan', 'Rodriguez', '+1432109876', 'ethan.rodriguez@example.com', 216),

('Amelia', 'Hill', '+1543210987', 'amelia.hill@example.com', 217),

('Jacob', 'Scott', '+1765432109', 'jacob.scott@example.com', 218),

('Harper', 'Green', '+1987654321', 'harper.green@example.com', 219),

('Daniel', 'Adams', '+1298765432', 'daniel.adams@example.com', 220),

('Evelyn', 'Baker', '+1654321098', 'evelyn.baker@example.com', 221),

('Michael', 'Nelson', '+1765432109', 'michael.nelson@example.com', 222),

('Ella', 'King', '+1897654321', 'ella.king@example.com', 223),

('Logan', 'Rivera', '+1432109876', 'logan.rivera@example.com', 224),

('Abigail', 'Campbell', '+1543210987', 'abigail.campbell@example.com', 225),

('Henry', 'Parker', '+1765432109', 'henry.parker@example.com', 226),

('Sofia', 'Collins', '+1987654321', 'sofia.collins@example.com', 227),

('Jackson', 'Stewart', '+1298765432', 'jackson.stewart@example.com', 228),

('Liam', 'Turner', '+1234567890', 'liam.turner@example.com', 229),

('Avery', 'Cooper', '+1987654321', 'avery.cooper@example.com', 230),

('Lucas', 'Hill', '+1654321879', 'lucas.hill@example.com', 231),

('Scarlett', 'Garcia', '+1765432987', 'scarlett.garcia@example.com', 232),

('Mason', 'Evans', '+1897654321', 'mason.evans@example.com', 233),

('Aria', 'Phillips', '+1238765432', 'aria.phillips@example.com', 234),

('Sebastian', 'Reed', '+1564321098', 'sebastian.reed@example.com', 235),

('Lily', 'Cook', '+1675432109', 'lily.cook@example.com', 236),

('Grayson', 'Morris', '+1898765432', 'grayson.morris@example.com', 237),

('Zoey', 'Bell', '+1298765432', 'zoey.bell@example.com', 238),

('Carter', 'Howard', '+1432109876', 'carter.howard@example.com', 239),

('Luna', 'Long', '+1543210987', 'luna.long@example.com', 240),

('Mateo', 'Diaz', '+1765432109', 'mateo.diaz@example.com', 241),

('Madison', 'Bailey', '+1987654321', 'madison.bailey@example.com', 242),

('Elijah', 'Price', '+1298765432', 'elijah.price@example.com', 243),

('Penelope', 'Reyes', '+1654321098', 'penelope.reyes@example.com', 244),

('Ryan', 'Ward', '+1765432109', 'ryan.ward@example.com', 245),

('Nora', 'Foster', '+1897654321', 'nora.foster@example.com', 246),

('Gavin', 'Bryant', '+1432109876', 'gavin.bryant@example.com', 247),

('Aubrey', 'Russell', '+1543210987', 'aubrey.russell@example.com', 248),

('Hudson', 'Griffin', '+1765432109', 'hudson.griffin@example.com', 249),

('Chloe', 'Dunn', '+1987654321', 'chloe.dunn@example.com', 250),

('Eli', 'Flores', '+1298765432', 'eli.flores@example.com', 251),

('Emma', 'Gordon', '+1654321098', 'emma.gordon@example.com', 252),

('Lincoln', 'Wallace', '+1765432109', 'lincoln.wallace@example.com', 253),

('Stella', 'Reynolds', '+1897654321', 'stella.reynolds@example.com', 254),

('Nathan', 'Hamilton', '+1432109876', 'nathan.hamilton@example.com', 255),

('Violet', 'Cole', '+1543210987', 'violet.cole@example.com', 256),

('Hazel', 'Murray', '+1765432109', 'hazel.murray@example.com', 257),

('Connor', 'Mills', '+1987654321', 'connor.mills@example.com', 258),

('Levi', 'Cox', '+1298765432', 'levi.cox@example.com', 259),

('Olivia', 'Sullivan', '+1432109876', 'olivia.sullivan@example.com', 260),

('Jackson', 'Coleman', '+1543210987', 'jackson.coleman@example.com', 261),

('Aurora', 'Perry', '+1765432109', 'aurora.perry@example.com', 262),

('Ethan', 'Barnes', '+1987654321', 'ethan.barnes@example.com', 263),

('Charlotte', 'Morrison', '+1298765432', 'charlotte.morrison@example.com', 264),

('Hunter', 'Knight', '+1654321098', 'hunter.knight@example.com', 265),

('Eva', 'Kennedy', '+1765432109', 'eva.kennedy@example.com', 266),

('Daniel', 'Warren', '+1897654321', 'daniel.warren@example.com', 267),

('Mila', 'Sanders', '+1432109876', 'mila.sanders@example.com', 268),

('Logan', 'Harrison', '+1543210987', 'logan.harrison@example.com', 269),

('Liam', 'Simpson', '+1765432109', 'liam.simpson@example.com', 270),

('Alice', 'Griffiths', '+1987654321', 'alice.griffiths@example.com', 271),

('Evan', 'Fletcher', '+1298765432', 'evan.fletcher@example.com', 272),

('Harper', 'Walsh', '+1654321098', 'harper.walsh@example.com', 273),

('Isaac', 'Willis', '+1765432109', 'isaac.willis@example.com', 274),

('Avery', 'Dixon', '+1897654321', 'avery.dixon@example.com', 275),

('Grace', 'Hunt', '+1432109876', 'grace.hunt@example.com', 276),

('Mason', 'Johnston', '+1543210987', 'mason.johnston@example.com', 277),

('Nova', 'Porter', '+1765432109', 'nova.porter@example.com', 278),

('Elena', 'Ryan', '+1987654321', 'elena.ryan@example.com', 279),

('Nolan', 'Parker', '+1298765432', 'nolan.parker@example.com', 280),

('Emilia', 'Gardner', '+1654321098', 'emilia.gardner@example.com', 281),

('Caleb', 'Bates', '+1765432109', 'caleb.bates@example.com', 282),

('Hannah', 'Sims', '+1897654321', 'hannah.sims@example.com', 283),

('Wyatt', 'Ford', '+1432109876', 'wyatt.ford@example.com', 284),

('Ariana', 'Woods', '+1543210987', 'ariana.woods@example.com', 285),

('Jack', 'Harper', '+1765432109', 'jack.harper@example.com', 286),

('Aiden', 'Mason', '+1987654321', 'aiden.mason@example.com', 287);

#### tbl\_hotel

**Part of the script** was presented, over 500 records have been added

USE TRAVEL\_AGENCY

GO

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (98, 'Kabul Grand Hotel', 100, '+93 123456789', 'info@kabulgrandhotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (98, 'Kabul Plaza Hotel', 80, '+93 987654321', 'info@kabulplazahotel.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (98, 'Kabul Inn', 60, '+93 555555555', 'info@kabulinn.com', 2);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (99, 'Kandahar Grand Hotel', 90, '+93 111111111', 'info@kandahargrandhotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (99, 'Kandahar Palace Hotel', 70, '+93 222222222', 'info@kandaharpalacehotel.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (99, 'Kandahar Inn', 50, '+93 333333333', 'info@kandaharinn.com', 2);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (100, 'Herat Grand Hotel', 110, '+93 444444444', 'info@heratgrandhotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (100, 'Herat Palace Hotel', 90, '+93 555555555', 'info@heratpalacehotel.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (100, 'Herat Inn', 70, '+93 666666666', 'info@heratinn.com', 2);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (101, 'Yerevan Marriott Hotel', 120, '+374 10 599000', 'info@yerevanmarriott.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (101, 'Hyatt Place Yerevan', 100, '+374 11 556000', 'info@hyattplace.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (101, 'Ibis Yerevan Center Hotel', 80, '+374 60 503000', 'info@ibis.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (102, 'Fairmont Baku, Flame Towers', 150, '+994 12 565 48 48', 'info@fairmont.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (102, 'JW Marriott Absheron Baku', 120, '+994 12 499 88 88', 'info@jwmarriott.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (102, 'Holiday Inn Baku', 90, '+994 12 610 90 00', 'info@holidayinn.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (103, 'Ganja Hotel', 80, '+994 22 222 22 22', 'info@ganjahotel.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (103, 'Hotel Ganja', 70, '+994 22 333 33 33', 'info@hotelganja.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (103, 'Ganja Inn', 60, '+994 22 444 44 44', 'info@ganjainn.com', 2);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (104, 'Four Seasons Hotel Bahrain Bay', 200, '+973 17 116000', 'info@fourseasons.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (104, 'The Ritz-Carlton, Bahrain', 180, '+973 17 586500', 'info@ritzcarlton.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (104, 'InterContinental Regency Bahrain', 150, '+973 17 227777', 'info@intercontinental.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (105, 'Pan Pacific Sonargaon Hotel Dhaka', 150, '+880 2 5502 8008', 'info@panpacific.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (105, 'InterContinental Dhaka', 120, '+880 2 5566 3030', 'info@intercontinental.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (105, 'Radisson Blu Dhaka Water Garden', 100, '+880 2 8754555', 'info@radissonblu.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (106, 'Hotel Agrabad', 120, '+880 31 713311', 'info@hotelagrabad.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (106, 'The Peninsula Chittagong', 100, '+880 31 2850860', 'info@thepeninsula.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (106, 'Hotel Saint Martin Ltd.', 80, '+880 31 651441', 'info@hotelsaintmartin.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (107, 'Taj Tashi', 200, '+975 2 336699', 'info@tajtashi.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (107, 'Le Méridien Thimphu', 180, '+975 2 337788', 'info@lemeridien.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (107, 'Hotel Druk', 150, '+975 2 322903', 'info@hoteldruk.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (108, 'The Empire Hotel & Country Club', 300, '+673 241 8888', 'info@theempire.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (108, 'Radisson Hotel Brunei Darussalam', 200, '+673 224 4272', 'info@radisson.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (108, 'The Brunei Hotel', 150, '+673 222 4235', 'info@bruneihotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (109, 'Rosewood Phnom Penh', 250, '+855 23 936 888', 'info@rosewoodhotels.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (109, 'NagaWorld Hotel & Entertainment Complex', 200, '+855 23 228 822', 'info@nagaworld.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (109, 'Sokha Phnom Penh Hotel', 150, '+855 23 685 8888', 'info@sokhahotels.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (110, 'Park Hyatt Siem Reap', 300, '+855 63 211 234', 'info@hyatt.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (110, 'Shinta Mani Angkor', 250, '+855 63 761 998', 'info@shintamani.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (110, 'Sokha Angkor Resort', 200, '+855 63 969 999', 'info@sokhahotels.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (111, 'The Peninsula Beijing', 350, '+86 10 8516 2888', 'info@peninsula.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (111, 'Waldorf Astoria Beijing', 300, '+86 10 8520 8989', 'info@waldorfastoria.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (111, 'Hilton Beijing Wangfujing', 250, '+86 10 5812 8888', 'info@hilton.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (112, 'Four Seasons Hotel Shanghai', 400, '+86 21 6256 8888', 'info@fourseasons.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (112, 'JW Marriott Hotel Shanghai at Tomorrow Square', 350, '+86 21 5359 4969', 'info@jwmarriott.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (112, 'The Ritz-Carlton Shanghai, Pudong', 300, '+86 21 2020 1111', 'info@ritzcarlton.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (113, 'Hilton Nicosia', 200, '+357 22 696300', 'info@hilton.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (113, 'Merit Lefkoşa Hotel & Casino', 150, '+357 22 616100', 'info@meritlefkosa.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (113, 'Semeli Hotel', 100, '+357 22 456000', 'info@semelihotel.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (114, 'Four Seasons Hotel Limassol', 250, '+357 25 858000', 'info@fourseasons.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (114, 'Amathus Beach Hotel Limassol', 200, '+357 25 832000', 'info@amathus.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (114, 'Londa Hotel', 150, '+357 25 865540', 'info@londahotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (115, 'Radisson Blu Iveria Hotel, Tbilisi', 200, '+995 32 240 22 00', 'info@radissonblu.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (115, 'Tbilisi Marriott Hotel', 150, '+995 32 277 91 00', 'info@marriott.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (115, 'Rooms Hotel Tbilisi', 100, '+995 322 02 00 20', 'info@roomshotel.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (116, 'The Leela Palace New Delhi', 300, '+91 11 3933 1234', 'info@theleela.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (116, 'Taj Palace, New Delhi', 250, '+91 11 2611 0202', 'info@tajhotels.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (116, 'ITC Maurya, New Delhi', 200, '+91 11 2611 2233', 'info@itchotels.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (117, 'The Oberoi Mumbai', 350, '+91 22 6632 5757', 'info@oberoihotels.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (117, 'Trident, Nariman Point', 300, '+91 22 6632 4343', 'info@tridenthotels.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (117, 'The Taj Mahal Palace', 250, '+91 22 6665 3366', 'info@tajhotels.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (118, 'Hotel Indonesia Kempinski Jakarta', 300, '+62 21 2358 3800', 'info@kempinski.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (118, 'The Ritz-Carlton Jakarta, Pacific Place', 250, '+62 21 2550 1888', 'info@ritzcarlton.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (118, 'Fairmont Jakarta', 200, '+62 21 2970 3333', 'info@fairmont.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (119, 'Hotel Majapahit Surabaya', 200, '+62 31 545 4333', 'info@majapahit-hotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (119, 'Sheraton Surabaya Hotel & Towers', 150, '+62 31 546 8000', 'info@sheraton.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (119, 'Hotel Bumi Surabaya', 100, '+62 31 531 1234', 'info@hotelbumi.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (120, 'Espinas Palace Hotel', 300, '+98 21 75411', 'info@espinashotel.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (120, 'Parsian Azadi Hotel Tehran', 250, '+98 21 2265 9999', 'info@parsianhotels.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (120, 'Tehran Grand Hotel', 200, '+98 21 2282 1000', 'info@tehrangrandhotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (121, 'Pars International Hotel', 250, '+98 51 376 0000', 'info@parsinternationalhotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (121, 'Homa Hotel Mashhad', 200, '+98 51 371 3000', 'info@homahotelmashhad.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (121, 'Parsian Kowsar Mashhad Hotel', 150, '+98 51 387 0000', 'info@parsianhotels.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (122, 'Al-Rasheed Hotel Baghdad', 250, '+964 780 491 3200', 'info@alrasheedhotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (122, 'Babylon Warwick Hotel', 200, '+964 750 345 6985', 'info@warwickhotels.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (122, 'Royal Tulip Al Rasheed Hotel', 150, '+964 783 158 2345', 'info@royaltulipalrasheed.com', 3);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (123, 'Waldorf Astoria Jerusalem', 400, '+972 2 542 3333', 'info@waldorfastoria.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (123, 'The King David Jerusalem', 350, '+972 2 620 8888', 'info@danhotels.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (123, 'The American Colony Hotel', 300, '+972 2 627 9777', 'info@americancolony.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (124, 'Dan Tel Aviv Hotel', 300, '+972 3 520 2525', 'info@dantelaviv.co.il', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (124, 'The Norman Tel Aviv', 250, '+972 3 543 5555', 'info@thenorman.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (124, 'Hilton Tel Aviv', 200, '+972 3 520 2222', 'info@hilton.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (125, 'The Tokyo Station Hotel', 500, '+81 3-5220-1111', 'info@tokyostationhotel.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (125, 'The Prince Park Tower Tokyo', 450, '+81 3-5400-1111', 'info@princehotels.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (125, 'Conrad Tokyo', 400, '+81 3-6388-8000', 'info@conradhotels.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (126, 'InterContinental Osaka', 400, '+81 6-6374-5700', 'info@intercontinental.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (126, 'The Ritz-Carlton Osaka', 350, '+81 6-6343-7000', 'info@ritzcarlton.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (126, 'Swissotel Nankai Osaka', 300, '+81 6-6646-1111', 'info@swissotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (127, 'Four Seasons Hotel Amman', 350, '+962 6 550 5555', 'info@fourseasons.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (127, 'InterContinental Jordan', 300, '+962 6 4641361', 'info@intercontinental.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (127, 'Sheraton Amman Al Nabil Hotel', 250, '+962 6 593 4111', 'info@sheraton.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (128, 'The Ritz-Carlton, Astana', 400, '+7 7172 734000', 'info@ritzcarlton.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (128, 'Hilton Astana', 350, '+7 7172 242000', 'info@hilton.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (128, 'Radisson Hotel, Astana', 300, '+7 7172 622000', 'info@radisson.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (129, 'The Ritz-Carlton, Almaty', 400, '+7 727 332 8888', 'info@ritzcarlton.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (129, 'InterContinental Almaty', 350, '+7 727 258 5000', 'info@intercontinental.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (129, 'The Ritz Palace Hotel', 300, '+7 727 397 7777', 'info@ritzpalacehotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (130, 'Jumeirah Messilah Beach Hotel & Spa', 450, '+965 2226 9600', 'info@jumeirah.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (130, 'Four Seasons Hotel Kuwait at Burj Alshaya', 400, '+965 2200 0000', 'info@fourseasons.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (130, 'Sheraton Kuwait, a Luxury Collection Hotel', 350, '+965 2242 2055', 'info@sheraton.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (131, 'Hyatt Regency Bishkek', 250, '+996 312 66 1234', 'info@hyatt.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (131, 'Golden Dragon Hotel', 200, '+996 312 31 80 00', 'info@goldendragon.kg', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (131, 'Ambassador Hotel', 150, '+996 312 31 03 33', 'info@ambassador.kg', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (132, 'Settha Palace Hotel', 200, '+856 21 217 581', 'info@setthapalace.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (132, 'Salana Boutique Hotel', 150, '+856 21 254 254', 'info@salanaboutique.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (132, 'Landmark Mekong Riverside Hotel', 100, '+856 21 215 699', 'info@landmarkmekonghotel.com', 4);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (133, 'Four Seasons Hotel Beirut', 300, '+961 1 761 000', 'info@fourseasons.com', 5);

INSERT INTO tbl\_hotel (hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars) VALUES (133, 'Le Gray Beirut', 250, '+961 1 971 111', 'info@legray.com', 5);

#### tbl\_flight

Almost **8,000,000** **records** were **generated using this procedure**.

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_insert\_flight\_records

--------------------------------------------------------------------------------------

-- This procedure is used to generate flight records between all possible city combinations.

-- It inserts flight records into the tbl\_flight table with random flight start and end times,

-- and prices for different classes (Economy, Business, First).

--

--

-- Example Usage:

-- EXEC up\_insert\_flight\_records

--

-- Result of the action

-- Procedure adds flights

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_insert\_flight\_records

AS

BEGIN

DECLARE @StartCityId INT

DECLARE @EndCityId INT

DECLARE @EconomyPrice FLOAT

DECLARE @BusinessPrice FLOAT

DECLARE @FirstClassPrice FLOAT

DECLARE @RandomHours INT

DECLARE @RandomDiff INT

DECLARE @StartTime DATETIME

DECLARE @EndTime DATETIME

CREATE TABLE #tbl\_city\_combinations (

StartCityId INT,

EndCityId INT

)

INSERT INTO #tbl\_city\_combinations (StartCityId, EndCityId)

SELECT c1.cityId AS StartCityId, c2.cityId AS EndCityId

FROM tbl\_city c1

CROSS JOIN tbl\_city c2

WHERE c1.cityId <> c2.cityId

DECLARE crs\_city\_list CURSOR FOR

SELECT StartCityId, EndCityId FROM #tbl\_city\_combinations

OPEN crs\_city\_list

FETCH NEXT FROM crs\_city\_list INTO @StartCityId, @EndCityId

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @EconomyPrice = ROUND(RAND() \* (300 - 70) + 70, 2)

SET @BusinessPrice = ROUND(@EconomyPrice + 100, 2)

SET @FirstClassPrice =ROUND(@EconomyPrice + 250, 2)

SET @RandomHours = ROUND(RAND() \* (4760 - 168) + 168, 0)

SET @StartTime = DATEADD(HOUR, @RandomHours, GETDATE())

SET @RandomDiff = ROUND(RAND() \* (19 - 1) + 1, 0)

SET @EndTime = DATEADD(HOUR, @RandomDiff, @StartTime)

INSERT INTO tbl\_flight(fliStartCityId, fliEndCityId, fliStartTime, fliEndTime, fliClass, fliPrice)

VALUES (@StartCityId, @EndCityId, @StartTime, @EndTime, 'Economy', @EconomyPrice),

(@StartCityId, @EndCityId, @StartTime, @EndTime, 'Business', @BusinessPrice),

(@StartCityId, @EndCityId, @StartTime, @EndTime, 'First', @FirstClassPrice)

FETCH NEXT FROM crs\_city\_list INTO @StartCityId, @EndCityId

END

CLOSE crs\_city\_list

DEALLOCATE crs\_city\_list

DROP TABLE #tbl\_city\_combinations

END

/\* ---------------------------------------------------------------------------------\*/

DECLARE @Counter INT = 0

WHILE @Counter < 5

BEGIN

EXEC up\_insert\_flight\_records

SET @Counter = @Counter + 1

END

#### tbl\_busroute

Over **2,500,000 records** were **generated using this procedure**.

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_insert\_busRoute\_records

--------------------------------------------------------------------------------------

-- This procedure is used to generate bus route records between all possible city combinations.

-- It inserts bus route records into the tbl\_busroute table with random start and end times,

-- drivers, buses, and prices.

--

--

-- Example Usage:

-- EXEC up\_insert\_busRoute\_records

--

-- Result of the action

-- Procedure adds bus routes

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_insert\_busRoute\_records

AS

BEGIN

DECLARE @StartCityId INT;

DECLARE @EndCityId INT;

DECLARE @DriverId INT;

DECLARE @BusId INT;

DECLARE @StartTime DATETIME;

DECLARE @EndTime DATETIME;

DECLARE @HoursToAdd INT;

DECLARE @Price FLOAT;

CREATE TABLE #tbl\_city\_combinations (

StartCityId INT,

EndCityId INT

);

INSERT INTO #tbl\_city\_combinations (StartCityId, EndCityId)

SELECT c1.cityId AS StartCityId, c2.cityId AS EndCityId

FROM tbl\_city c1

CROSS JOIN tbl\_city c2

WHERE c1.cityId <> c2.cityId;

DECLARE crs\_city\_list CURSOR FOR

SELECT StartCityId, EndCityId FROM #tbl\_city\_combinations;

OPEN crs\_city\_list;

FETCH NEXT FROM crs\_city\_list INTO @StartCityId, @EndCityId;

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @DriverId = ROUND(RAND() \* (63 - 1) + 1, 0);

SET @BusId = ROUND(RAND() \* (87 - 1) + 1, 0);

SET @StartTime = DATEADD(HOUR, ROUND(RAND() \* (4760 - 168) + 168, 0), GETDATE());

SET @HoursToAdd = ROUND(RAND() \* (72 - 1) + 1, 0);

SET @EndTime = DATEADD(HOUR, @HoursToAdd, @StartTime);

SET @Price = ROUND(RAND() \* (250 - 50) + 50, 2);

INSERT INTO tbl\_busroute (driverId, busRouteStartCityId, busRouteEndCityId, busRouteStartTime, busRouteEndTime, busRouteBusID, busRoutePrice)

VALUES (@DriverId, @StartCityId, @EndCityId, @StartTime, @EndTime, @BusId, @Price);

FETCH NEXT FROM crs\_city\_list INTO @StartCityId, @EndCityId;

END

CLOSE crs\_city\_list;

DEALLOCATE crs\_city\_list;

DROP TABLE #tbl\_city\_combinations;

END;

/\* -------------------------------------------------------------------------------------------------------\*/

DECLARE @Counter INT = 0

WHILE @Counter < 5

BEGIN

EXEC up\_insert\_busRoute\_records

SET @Counter = @Counter + 1

END

#### tbl\_package

Over **12,000 records** were **generated using these procedures.**

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_insert\_package\_flight\_records

--------------------------------------------------------------------------------------

-- This procedure is used to generate package records with flight details.

-- It randomly selects a city and fetches outbound and return flight IDs for that city.

-- It also fetches a guide ID and hotel ID for the selected city.

-- Then it inserts a package record into the tbl\_package table with the retrieved details.

--

-- Example Usage:

-- EXEC up\_insert\_package\_flight\_records

--

-- Result of the action

-- Procedure adds package records with flight details

--------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_insert\_package\_bus\_records

--------------------------------------------------------------------------------------

-- This procedure is used to generate package records with bus route details.

-- It randomly selects a city and fetches outbound and return bus route IDs for that city.

-- It also fetches a guide ID and hotel ID for the selected city.

-- Then it inserts a package record into the tbl\_package table with the retrieved details.

--

-- Example Usage:

-- EXEC up\_insert\_package\_bus\_records

--

-- Result of the action

-- Procedure adds package records with bus route details

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_insert\_package\_flight\_records

AS

BEGIN

DECLARE @CityId INT;

DECLARE @CityName VARCHAR(50);

DECLARE @OutboundFlightId INT;

DECLARE @ReturnFlightId INT;

DECLARE @GuideId INT;

DECLARE @HotelId INT;

DECLARE @RandomIndex INT;

SET @RandomIndex = ROUND(RAND() \* (441 - 1) + 1, 0);

SELECT @CityId = cityId, @CityName = cityName

FROM tbl\_city

WHERE cityId = @RandomIndex;

SELECT TOP 1 @OutboundFlightId = fliId FROM tbl\_flight WHERE fliEndCityId = @CityId ORDER BY NEWID();

SELECT TOP 1 @ReturnFlightId = fliId

FROM tbl\_flight WHERE fliStartCityId = (SELECT fliEndCityId FROM tbl\_flight WHERE fliId = @OutboundFlightId) AND

fliEndCityId = (SELECT fliStartCityId FROM tbl\_flight WHERE fliId = @OutboundFlightId) AND

fliStartTime > (SELECT fliEndTime FROM tbl\_flight WHERE fliId = @OutboundFlightId) AND

DATEDIFF(DAY, (SELECT fliEndTime FROM tbl\_flight WHERE fliId = @OutboundFlightId), fliStartTime) < 30 AND

DATEDIFF(DAY, (SELECT fliEndTime FROM tbl\_flight WHERE fliId = @OutboundFlightId), fliStartTime) > 3

ORDER BY NEWID();

SELECT TOP 1 @GuideId = guideId FROM tbl\_guide WHERE guideCityId = @CityId ORDER BY NEWID();

SELECT TOP 1 @HotelId = hotId FROM tbl\_hotel WHERE hotAddrId IN (SELECT addrId FROM tbl\_address WHERE addrCityId = @CityId) ORDER BY NEWID();

IF @OutboundFlightId IS NULL OR @GuideId IS NULL OR @HotelId IS NULL OR @ReturnFlightId IS NULL

BEGIN

RETURN;

END

ELSE

BEGIN

INSERT INTO tbl\_package(packTitle, packDescription, packCityId, packHotId, packOutboundFlightId, packReturnFlightId, packGuideId)

VALUES ('Unforgettable trip to ' + @CityName, NULL, @CityId, @HotelId, @OutboundFlightId, @ReturnFlightId, @GuideId);

END

END;

/\*----------------------------------------------------------------------------------\*/

CREATE OR ALTER PROCEDURE up\_insert\_package\_bus\_records

AS

BEGIN

DECLARE @CityId INT;

DECLARE @CityName VARCHAR(50);

DECLARE @OutboundBusRouteId INT;

DECLARE @ReturnBusRouteId INT;

DECLARE @StartDate DATE;

DECLARE @EndDate DATE;

DECLARE @GuideId INT;

DECLARE @HotelId INT;

DECLARE @Duration TINYINT;

DECLARE @RandomIndex INT;

DECLARE @Price FLOAT;

SET @RandomIndex = ROUND(RAND() \* (441 - 1) + 1, 0);

SELECT @CityId = cityId, @CityName = cityName

FROM tbl\_city

WHERE cityId = @RandomIndex;

SELECT TOP 1 @OutboundBusRouteId = busRouteId FROM tbl\_busroute WHERE busRouteEndCityId = @CityId ORDER BY NEWID();

SELECT TOP 1 @ReturnBusRouteId = busRouteId, @EndDate = busRouteEndTime

FROM tbl\_busroute WHERE busRouteStartCityId = (SELECT busRouteEndCityId FROM tbl\_busroute WHERE busRouteId = @OutboundBusRouteId) AND

busRouteEndCityId = (SELECT busRouteStartCityId FROM tbl\_busroute WHERE busRouteId = @OutboundBusRouteId) AND

busRouteStartTime > (SELECT busRouteEndTime FROM tbl\_busroute WHERE busRouteId = @OutboundBusRouteId) AND

DATEDIFF(DAY, (SELECT busRouteEndTime FROM tbl\_busroute WHERE busRouteId = @OutboundBusRouteId), busRouteStartTime) < 30 AND

DATEDIFF(DAY, (SELECT busRouteEndTime FROM tbl\_busroute WHERE busRouteId = @OutboundBusRouteId), busRouteStartTime) > 3

ORDER BY NEWID();

SELECT TOP 1 @GuideId = guideId FROM tbl\_guide WHERE guideCityId = @CityId ORDER BY NEWID();

SELECT TOP 1 @HotelId = hotId FROM tbl\_hotel WHERE hotAddrId IN (SELECT addrId FROM tbl\_address WHERE addrCityId = @CityId) ORDER BY NEWID();

IF @OutboundBusRouteId IS NULL OR @GuideId IS NULL OR @HotelId IS NULL OR @ReturnBusRouteId IS NULL

BEGIN

RETURN;

END

ELSE

BEGIN

INSERT INTO tbl\_package (packTitle, packDescription, packCityId, packHotId, packOutboundBusRouteId, packReturnBusRouteId, packGuideId)

VALUES ('Unforgettable trip to ' + @CityName, NULL, @CityId, @HotelId, @OutboundBusRouteId, @ReturnBusRouteId, @GuideId);

END

END;

/\*----------------------------------------------------------------------------------\*/

DECLARE @Counter INT = 0

WHILE @Counter < 500

BEGIN

EXEC up\_insert\_package\_bus\_records

EXEC up\_insert\_package\_flight\_records

SET @Counter = @Counter + 1

END

#### tbl\_booking

Almost **400 records** were **generated using this procedure.**

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_insert\_booking\_records

--------------------------------------------------------------------------------------

-- This procedure is used to generate booking records.

-- It randomly selects a customer and a package, and assigns a random discount percentage.

-- Then it inserts a booking record into the tbl\_booking table with the retrieved details.

--

-- Example Usage:

-- EXEC up\_insert\_booking\_records

--

-- Result of the action

-- Procedure adds booking records

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_insert\_booking\_records

AS

BEGIN

DECLARE @Discount TABLE (Discount FLOAT);

INSERT INTO @Discount (Discount) VALUES (0.0), (0.0), (0.0), (0.0), (0.0), (0.0), (0.0), (0.0), (0.0), (0.0), (0.0), (0.0), (0.02), (0.03), (0.05), (0.07), (0.1), (0.12), (0.15), (0.18);

DECLARE @CustId INT;

DECLARE @PackageId INT;

DECLARE @DiscountPercent FLOAT;

DECLARE @CustomerCount INT;

SELECT @CustomerCount = COUNT(\*) FROM tbl\_customer;

DECLARE @Counter INT = 1;

WHILE @Counter <= @CustomerCount

BEGIN

SELECT @CustId = custId FROM tbl\_customer ORDER BY custId OFFSET (@Counter - 1) ROWS FETCH NEXT 1 ROW ONLY;

SELECT TOP 1 @DiscountPercent = Discount

FROM @Discount

ORDER BY NEWID();

SET @PackageId = ROUND(RAND() \* (2854 - 1) + 1, 0);

IF @CustId IS NOT NULL AND @PackageId IS NOT NULL AND EXISTS (SELECT 1 FROM tbl\_package WHERE packId = @PackageId)

BEGIN

INSERT INTO tbl\_booking(bookCustId, bookPackageId, bookDiscountPercent, bookPrice)

VALUES (@CustId, @PackageId, @DiscountPercent, 1);

SET @Counter = @Counter + 1;

END

END

END;

--------------------------------------------------------------------------------------

EXEC up\_insert\_booking\_records

# Wykaz utworzonych widoków

--This SQL query returns the names of all views contained in the 'dbo' schema in the database.

SELECT TABLE\_NAME AS ViewName

FROM INFORMATION\_SCHEMA.VIEWS

WHERE TABLE\_SCHEMA = 'dbo';

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

5 List of created views

# Skrypt tworzenia widoków

#### vw\_package\_details

/\*------------------------------------------------------------------------------------

View Name: vw\_package\_details

--------------------------------------------------------------------------------------

Output Description:

- Id: The unique identifier of the package.

- Title: The title of the package.

- Description: The description of the package.

- Destination: The destination city of the package.

- HotelName: The name of the hotel included in the package.

- Duration: The duration of the package.

- Price: The price of the package.

- StartDate: The start date of the package.

- EndDate: The end date of the package.

- OutboundFlightId: The outbound flight ID included in the package.

- ReturnFlightId: The return flight ID included in the package.

- OutboundBusRouteId: The outbound bus route ID included in the package.

- ReturnBusRouteId: The return bus route ID included in the package.

- GuideId: The guide ID included in the package.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_package\_details;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_package\_details AS

SELECT p.packId AS Id, p.packTitle AS Title, p.packDescription AS Description, c.cityName AS Destination, h.hotName AS HotelName, p.packDuration AS Duration, p.packPrice AS Price, p.packStartDate AS StartDate, p.packEndDate AS EndDate, ouf.fliId AS OutboundFlightId, rf.fliId AS ReturnFlightId, ob.busRouteId AS OutboundBusRouteId, rb.busRouteId AS ReturnBusRouteId, g.guideId AS GuideId

FROM tbl\_package p

INNER JOIN tbl\_city c ON p.packCityId = c.cityId

INNER JOIN tbl\_hotel h ON p.packHotId = h.hotId

LEFT JOIN tbl\_flight ouf ON p.packOutboundFlightId = ouf.fliId

LEFT JOIN tbl\_flight rf ON p.packReturnFlightId = rf.fliId

LEFT JOIN tbl\_busroute ob ON p.packOutboundBusRouteId = ob.busRouteId

LEFT JOIN tbl\_busroute rb ON p.packReturnBusRouteId = rb.busRouteId

INNER JOIN tbl\_guide g ON p.packGuideId = g.guideId;

Obraz zawierający tekst, zrzut ekranu, menu

Opis wygenerowany automatycznie

6 Sample output of the vw\_package\_details view

#### vw\_package\_by\_country

/\*------------------------------------------------------------------------------------

View Name: vw\_package\_by\_country

--------------------------------------------------------------------------------------

Output Description:

- CountryName: The name of the country.

- PackageCount: The count of packages available in the country.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_package\_by\_country;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_package\_by\_country AS

SELECT co.ctryName, COUNT(p.packId) AS PackageCount

FROM tbl\_package p

INNER JOIN tbl\_city c ON p.packCityId = c.cityId

INNER JOIN tbl\_country co ON c.ctryId = co.ctryId

GROUP BY co.ctryName;

Obraz zawierający tekst, numer, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

7 Sample output of the vw\_package\_by\_country view

#### vw\_hotel\_contact\_details

/\*------------------------------------------------------------------------------------

View Name: vw\_hotel\_contact\_details

--------------------------------------------------------------------------------------

Output Description:

- Id: The unique identifier of the hotel.

- Name: The name of the hotel.

- TelephoneNo: The telephone number of the hotel.

- Email: The contact email of the hotel.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_hotel\_contact\_details;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_hotel\_contact\_details AS

SELECT h.hotId AS Id, h.hotName AS Name, h.hotTelephoneNo AS TelephoneNo, h.hotContactEmail AS Email

FROM tbl\_hotel h

INNER JOIN tbl\_address a ON h.hotAddrId = a.addrId

INNER JOIN tbl\_city ci ON a.addrCityId = ci.cityId

INNER JOIN tbl\_country co ON ci.ctryId = co.ctryId;

Obraz zawierający tekst, zrzut ekranu, menu, Czcionka

Opis wygenerowany automatycznie

8 Sample output of the vw\_hotel\_contact\_details view

#### vw\_hotel\_by\_country

/\*------------------------------------------------------------------------------------

View Name: vw\_hotel\_by\_country

--------------------------------------------------------------------------------------

Output Description:

- CountryId: The unique identifier of the country.

- CountryName: The name of the country.

- HotelCount: The count of hotels located in the country.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_hotel\_by\_country;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_hotel\_by\_country AS

SELECT co.ctryId AS CountryId ,co.ctryName AS CountryName, COUNT(h.hotId) AS HotelCount

FROM tbl\_hotel h

INNER JOIN tbl\_address a ON h.hotAddrId = a.addrId

INNER JOIN tbl\_city ci ON a.addrCityId = ci.cityId

INNER JOIN tbl\_country co ON ci.ctryId = co.ctryId

GROUP BY co.ctryId, co.ctryName;

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

9 Sample output of the vw\_hotel\_by\_country view

#### vw\_hotel\_by\_city

/\*------------------------------------------------------------------------------------

View Name: vw\_hotel\_by\_city

--------------------------------------------------------------------------------------

Output Description:

- CityId: The unique identifier of the city.

- CityName: The name of the city.

- HotelCount: The count of hotels located in the city.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_hotel\_by\_city;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_hotel\_by\_city AS

SELECT ci.cityId AS CityId, ci.cityName AS CityName, COUNT(h.hotId) AS HotelCount

FROM tbl\_hotel h

INNER JOIN tbl\_address a ON h.hotAddrId = a.addrId

INNER JOIN tbl\_city ci ON a.addrCityId = ci.cityId

GROUP BY ci.cityId, ci.cityName;

Obraz zawierający tekst, numer, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

10 Sample Output of the vw\_hotel\_by\_city view

#### vw\_hotel\_address\_details

/\*------------------------------------------------------------------------------------

View Name: vw\_hotel\_address\_details

--------------------------------------------------------------------------------------

Output Description:

- Id: The unique identifier of the hotel.

- Name: The name of the hotel.

- Country: The name of the country where the hotel is located.

- City: The name of the city where the hotel is located.

- Region: The region of the address.

- Street: The street of the address.

- PostalCode: The postal code of the address.

- HouseNo: The house number of the address.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_hotel\_address\_details;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_hotel\_address\_details AS

SELECT h.hotId AS Id, h.hotName AS Name, co.ctryName AS Country, ci.cityName AS City, a.addrRegion AS Region, a.addrStreet AS Street, a.addrPostalCode AS PostalCode, a.addrHouseNo AS HouseNo

FROM tbl\_hotel h

INNER JOIN tbl\_address a ON h.hotAddrId = a.addrId

INNER JOIN tbl\_city ci ON a.addrCityId = ci.cityId

INNER JOIN tbl\_country co ON ci.ctryId = co.ctryId;

Obraz zawierający tekst, numer, zrzut ekranu

Opis wygenerowany automatycznie

11 Sample Output of the vw\_hotel\_address\_details view

#### vw\_guide\_city

/\*------------------------------------------------------------------------------------

View Name: vw\_guide\_city

--------------------------------------------------------------------------------------

Output Description:

- Id: The unique identifier of the guide.

- Name: The first name of the guide.

- Surname: The last name of the guide.

- City: The name of the city where the guide operates.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_guide\_city;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_guide\_city AS

SELECT g.guideId AS Id, g.guideName AS Name, g.guideSurname AS Surname, c.cityName AS City

FROM tbl\_guide g

INNER JOIN tbl\_city c ON g.guideCityId = c.cityId;

Obraz zawierający tekst, numer, menu, Czcionka

Opis wygenerowany automatycznie

12 Sample Output of the vw\_guide\_city view

#### vw\_flight\_details

/\*------------------------------------------------------------------------------------

View Name: vw\_flight\_details

--------------------------------------------------------------------------------------

Output Description:

- Id: The unique identifier of the flight.

- StartCity: The name of the city where the flight starts.

- EndCity: The name of the city where the flight ends.

- DepartureDate: The date of departure.

- DepartureTime: The time of departure.

- ArrivalDate: The date of arrival.

- ArrivalTime: The time of arrival.

- Class: The class of the flight.

- Price: The price of the flight.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_flight\_details;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_flight\_details AS

SELECT f.fliId AS Id, s.cityName AS StartCity, e.cityName AS EndCity, CONVERT(DATE, f.fliStartTime) AS DepartureDate,FORMAT(f.fliStartTime, 'HH:mm') AS DepartureTime, CONVERT(DATE, f.fliEndTime) AS ArrivalDate,FORMAT(f.fliEndTime, 'HH:mm') AS ArrivalTime, f.fliClass AS Class, f.fliPrice AS Price

FROM tbl\_flight f

INNER JOIN tbl\_city s ON f.fliStartCityId = s.cityId

INNER JOIN tbl\_city e ON f.fliEndCityId = e.cityId;

Obraz zawierający tekst, zrzut ekranu, numer

Opis wygenerowany automatycznie

13 Sample Output of the vw\_flight\_details view

#### vw\_customers\_by\_country

/\*------------------------------------------------------------------------------------

View Name: vw\_customers\_by\_country

--------------------------------------------------------------------------------------

Output Description:

- CustomerCount: The count of customers residing in each country.

- Country: The name of the country.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_customers\_by\_country;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_customers\_by\_country AS

SELECT COUNT(cu.custId) AS CustomerCount, co.ctryName AS Country

FROM tbl\_customer cu

INNER JOIN tbl\_address a ON cu.custAddrId = a.addrId

INNER JOIN tbl\_city c ON a.addrCityId = c.cityId

INNER JOIN tbl\_country co ON c.ctryId = co.ctryId

GROUP BY co.ctryName;

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

14 Sample Output of the vw\_customers\_by\_country view

#### vw\_customer\_address

/\*------------------------------------------------------------------------------------

View Name: vw\_customer\_address

--------------------------------------------------------------------------------------

Output Description:

- Id: The unique identifier of the customer.

- Name: The first name of the customer.

- Surname: The last name of the customer.

- City: The name of the city where the customer resides.

- Region: The region of the address.

- Street: The street of the address.

- PostalCode: The postal code of the address.

- HouseNo: The house number of the address.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_customer\_address;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_customer\_address AS

SELECT cu.custId AS Id, cu.custName AS Name, cu.custSurname AS Surname, c.cityName AS City,

a.addrRegion AS Region, a.addrStreet AS Street, a.addrPostalCode AS PostalCode, a.addrHouseNo AS HouseNo

FROM tbl\_customer cu

INNER JOIN tbl\_address a ON cu.custAddrId = a.addrId

INNER JOIN tbl\_city c ON a.addrCityId = c.cityId;

Obraz zawierający tekst, zrzut ekranu, numer

Opis wygenerowany automatycznie

15 Sample Output of the vw\_customer\_address view

#### vw\_city\_country

/\*------------------------------------------------------------------------------------

View Name: vw\_city\_country

--------------------------------------------------------------------------------------

Output Description:

- Id: The unique identifier of the city.

- City: The name of the city.

- Country: The name of the country to which the city belongs.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_city\_country;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_city\_country AS

SELECT c.cityId AS Id, c.cityName AS City, co.ctryName AS Country

FROM tbl\_city c

INNER JOIN tbl\_country co ON c.ctryId = co.ctryId;

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

16 Sample Output of the vw\_city\_country view

#### vw\_busroute\_details

/\*------------------------------------------------------------------------------------

View Name: vw\_busroute\_details

--------------------------------------------------------------------------------------

Output Description:

- Id: The unique identifier of the bus route.

- DriverName: The name of the driver.

- DriverSurname: The surname of the driver.

- DriverPhone: The phone number of the driver.

- BusNo: The number of the bus.

- BusModel: The model of the bus.

- BusColor: The color of the bus.

- DepartureDate: The date of departure.

- DepartureTime: The time of departure.

- ArrivalDate: The date of arrival.

- ArrivalTime: The time of arrival.

- StartCity: The name of the city where the bus route starts.

- EndCity: The name of the city where the bus route ends.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_busroute\_details;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_busroute\_details AS

SELECT br.busRouteId AS Id, d.driverName AS DriverName, d.driverSurName AS DriverSurname, d.driverPhone AS DriverPhone, b.busNo AS BusNo, b.busModel AS BusModel, b.busColor AS BusColor,CONVERT(DATE, br.busRouteStartTime) AS DepartureDate,FORMAT(br.busRouteStartTime, 'HH:mm') AS DepartureTime, CONVERT(DATE, br.busRouteEndTime) AS ArrivalDate, FORMAT(br.busRouteEndTime, 'HH:mm') AS ArrivalTime, s.cityName AS StartCity, e.cityName AS EndCity

FROM tbl\_busroute br

INNER JOIN tbl\_driver d ON br.driverId = d.driverId

INNER JOIN tbl\_bus b ON br.busRouteBusID = b.busId

INNER JOIN tbl\_city s ON br.busRouteStartCityId = s.cityId

INNER JOIN tbl\_city e ON br.busRouteEndCityId = e.cityId;

Obraz zawierający tekst, zrzut ekranu, numer, dokument

Opis wygenerowany automatycznie

17 Sample Output of the vw\_busroute\_details view

#### vw\_busdriver\_details

/\*------------------------------------------------------------------------------------

View Name: vw\_busdriver\_details

--------------------------------------------------------------------------------------

Output Description:

- Id: The unique identifier of the bus driver.

- Name: The name of the bus driver.

- Surname: The surname of the bus driver.

- Phone: The phone number of the bus driver.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_busdriver\_details;

------------------------------------------------------------------------------------\*/

Use TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_busdriver\_details AS

SELECT d.driverId AS Id, d.driverName AS Name, d.driverSurName AS Surname, d.driverPhone AS Phone

FROM tbl\_driver d;

Obraz zawierający tekst, menu, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

18 Sample Output of the vw\_busdriver\_details view

#### vw\_booking\_details

/\*------------------------------------------------------------------------------------

View Name: vw\_booking\_details

--------------------------------------------------------------------------------------

Output Description:

- Id: The unique identifier of the booking.

- CustomerName: The name of the customer who made the booking.

- CustomerSurname: The surname of the customer who made the booking.

- Title: The title of the package booked.

- Discount: The discount percentage applied to the booking.

- DiscountAmount: The amount of discount applied to the booking.

- TotalPrice: The total price of the booking after discount.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_booking\_details;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_booking\_details AS

SELECT b.bookId AS Id, c.custName AS CustomerName, c.custSurname AS CustomerSurame, p.packTitle AS Title, b.bookDiscountPercent AS Discount, b.bookDiscountAmnt AS DiscountAmount, b.bookPrice AS TotalPrice

FROM tbl\_booking b

INNER JOIN tbl\_customer c ON b.bookCustId = c.custId

INNER JOIN tbl\_package p ON b.bookPackageId = p.packId;

Obraz zawierający tekst, zrzut ekranu, numer

Opis wygenerowany automatycznie

19 Sample Output of the vw\_booking\_details view

#### vw\_annual\_income

/\*------------------------------------------------------------------------------------

View Name: vw\_annual\_income

--------------------------------------------------------------------------------------

Output Description:

- BookingYear: The year of the booking.

- BookingMonth: The month of the booking.

- TotalPrice: The total price of bookings made in the specified year and month.

- TotalDiscountAmount: The total amount of discount applied to bookings made in the specified year and month.

- Income: The income generated from bookings after applying taxes.

--------------------------------------------------------------------------------------

Example Invocation:

USE TRAVEL\_AGENCY

GO

SELECT \* FROM vw\_annual\_income;

------------------------------------------------------------------------------------\*/

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER VIEW vw\_annual\_income AS

SELECT YEAR(b.bookCreatedAt) AS BookingYear,MONTH(b.bookCreatedAt) AS BookingMonth,

SUM(b.bookPrice) AS TotalPrice, SUM(b.bookDiscountAmnt) AS TotalDiscountAmount,

CAST(SUM((((b.bookPrice + b.bookDiscountAmnt) / 115) \* 15) - b.bookDiscountAmnt) AS DECIMAL(10, 2)) AS Income

FROM tbl\_booking b

GROUP BY YEAR(b.bookCreatedAt),MONTH(b.bookCreatedAt);

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

20 Sample output of the vw\_annual\_income view

# Wykaz utworzonych procedur składowanych

/\*

This SQL query returns the names of all stored procedures (and their details)

contained in the 'dbo' schema in the database.

\*/

SELECT name, object\_id, schema\_id, type, create\_date

FROM sys.procedures WHERE type\_desc = 'SQL\_STORED\_PROCEDURE';

Obraz zawierający tekst, menu, zrzut ekranu, dokument

Opis wygenerowany automatycznie

21 1 List of stored procedures created

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

21 2 List of stored procedures created

# Skrypt tworzenia procedur składowanych

## UPDATE

#### up\_update\_package

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_package

--------------------------------------------------------------------------------------

-- This procedure is used to update the details of a package in the database.

-- Additionally, it ensures that the specified flight or bus route matches the package city.

-- It also ensures that either flights or bus routes are specified for the package, not both.

-- If the input parameters pass all validations, the package details are updated in the database.

--

-- Input Parameters:

-- @packId: The ID of the package to be updated (required).

-- @packTitle: The updated title of the package (optional).

-- @packDescription: The updated description of the package (optional).

-- @packCityId: The updated city ID associated with the package (optional).

-- @packHotId: The updated hotel ID associated with the package (optional).

-- @packOutboundFlightId: The updated outbound flight ID associated with the package (optional).

-- @packReturnFlightId: The updated return flight ID associated with the package (optional).

-- @packOutboundBusRouteId: The updated outbound bus route ID associated with the package (optional).

-- @packReturnBusRouteId: The updated return bus route ID associated with the package (optional).

-- @packGuideId: The updated guide ID associated with the package (optional).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_package @packId = 1, @packTitle = 'Updated Package Title';

--

-- Result of the action:

-- Package data has been successfully modified.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_package

@packId INT,

@packTitle VARCHAR(30) = NULL,

@packDescription TEXT = NULL,

@packCityId INT = NULL,

@packHotId INT = NULL,

@packOutboundFlightId INT = NULL,

@packReturnFlightId INT = NULL,

@packOutboundBusRouteId INT = NULL,

@packReturnBusRouteId INT = NULL,

@packGuideId INT = NULL

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_package WHERE packId = @packId)

BEGIN

RAISERROR('The package with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @packTitle IS NOT NULL AND LEN(@packTitle) = 0

BEGIN

RAISERROR('The package title cannot be empty.', 16, 1);

RETURN;

END;

SELECT @packCityId= COALESCE(@packCityId, packCityId) FROM tbl\_package WHERE packId=@packId;

IF @packCityId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @packCityId)

BEGIN

RAISERROR('The specified city ID does not exist.', 16, 1);

RETURN;

END;

SELECT @packHotId=COALESCE(@packHotId, packHotId) FROM tbl\_package WHERE packId=@packId;

IF @packHotId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_hotel WHERE hotId = @packHotId)

BEGIN

RAISERROR('The specified hotel ID does not exist.', 16, 1);

RETURN;

END;

IF EXISTS (

SELECT 1

FROM tbl\_hotel AS H

INNER JOIN tbl\_address AS A ON H.hotAddrId = A.addrId

WHERE H.hotId = @packHotId AND A.addrCityId <> @packCityId

)

BEGIN

RAISERROR('The city of the specified hotel does not match the package city.', 16, 1);

RETURN;

END;

SELECT @packGuideId=COALESCE(@packGuideId, packGuideId) FROM tbl\_package WHERE packId=@packId;

IF @packGuideId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_guide WHERE guideId = @packGuideId)

BEGIN

RAISERROR('The specified guide ID does not exist.', 16, 1);

RETURN;

END;

IF @packGuideId IS NOT NULL AND @packCityId IS NOT NULL

BEGIN

SELECT @packCityId= COALESCE(@packCityId, packCityId) FROM tbl\_package WHERE packId=@packId;

IF EXISTS (

SELECT 1

FROM tbl\_guide AS G

INNER JOIN tbl\_city AS C ON G.guideCityId = C.cityId

WHERE G.guideId = @packGuideId AND C.cityId <> @packCityId

)

BEGIN

RAISERROR('The city of the specified guide does not match the package city.', 16, 1);

RETURN;

END;

END;

SELECT @packOutboundFlightId = COALESCE(@packOutboundFlightId, packOutboundFlightId) FROM tbl\_package WHERE packId=@packId;

SELECT @packReturnFlightId = COALESCE(@packReturnFlightId, packReturnFlightId) FROM tbl\_package WHERE packId=@packId;

SELECT @packOutboundBusRouteId = COALESCE(@packOutboundBusRouteId, packOutboundBusRouteId) FROM tbl\_package WHERE packId=@packId;

SELECT @packReturnBusRouteId = COALESCE(@packReturnBusRouteId, packReturnBusRouteId) FROM tbl\_package WHERE packId=@packId;

IF @packOutboundFlightId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_flight WHERE fliId = @packOutboundFlightId)

BEGIN

RAISERROR('The specified outbound flight ID does not exist.', 16, 1);

RETURN;

END;

IF @packReturnFlightId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_flight WHERE fliId = @packReturnFlightId)

BEGIN

RAISERROR('The specified return flight ID does not exist.', 16, 1);

RETURN;

END;

IF @packOutboundBusRouteId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_busroute WHERE busRouteId = @packOutboundBusRouteId)

BEGIN

RAISERROR('The specified outbound bus route ID does not exist.', 16, 1);

RETURN;

END;

IF @packReturnBusRouteId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_busroute WHERE busRouteId = @packReturnBusRouteId)

BEGIN

RAISERROR('The specified return bus route ID does not exist.', 16, 1);

RETURN;

END;

IF (SELECT packOutboundFlightId FROM tbl\_package WHERE packId=@packId) IS NULL AND (@packOutboundFlightId IS NOT NULL OR @packReturnFlightId IS NOT NULL)

BEGIN

RAISERROR('Bus routes are specified for this package. You cannot specify both flights and bus routes for the same package.', 16, 1);

RETURN;

END

ELSE IF (SELECT packOutboundBusRouteId FROM tbl\_package WHERE packId=@packId) IS NULL AND (@packOutboundBusRouteId IS NOT NULL OR @packReturnBusRouteId IS NOT NULL)

BEGIN

RAISERROR('Flights are specified for this package. You cannot specify both flights and bus routes for the same package.', 16, 1);

RETURN;

END;

IF @packOutboundFlightId IS NOT NULL

BEGIN

IF EXISTS (

SELECT 1

FROM tbl\_flight AS F

INNER JOIN tbl\_city AS C ON F.fliEndCityId = C.cityId

WHERE F.fliId = @packOutboundFlightId AND C.cityId <> @packCityId

)

BEGIN

RAISERROR('The city of the specified outbound flight does not match the package city.', 16, 1);

RETURN;

END;

END;

IF @packReturnFlightId IS NOT NULL

BEGIN

IF EXISTS (

SELECT 1

FROM tbl\_flight AS F

INNER JOIN tbl\_city AS C ON F.fliStartCityId = C.cityId

WHERE F.fliId = @packReturnFlightId AND C.cityId <> @packCityId

)

BEGIN

RAISERROR('The city of the specified return flight does not match the package city.', 16, 1);

RETURN;

END;

END;

IF @packOutboundBusRouteId IS NOT NULL

BEGIN

IF EXISTS (

SELECT 1

FROM tbl\_busroute AS B

INNER JOIN tbl\_city AS C ON B.busRouteEndCityId = C.cityId

WHERE B.busRouteId = @packOutboundBusRouteId AND C.cityId <> @packCityId

)

BEGIN

RAISERROR('The city of the specified outbound bus route does not match the package city.', 16, 1);

RETURN;

END;

END;

IF @packReturnBusRouteId IS NOT NULL

BEGIN

IF EXISTS (

SELECT 1

FROM tbl\_busroute AS B

INNER JOIN tbl\_city AS C ON B.busRouteStartCityId = C.cityId

WHERE B.busRouteId = @packReturnBusRouteId AND C.cityId <> @packCityId

)

BEGIN

RAISERROR('The city of the specified return bus route does not match the package city.', 16, 1);

RETURN;

END;

END;

IF (@packOutboundFlightId IS NOT NULL OR @packReturnFlightId IS NOT NULL) AND (@packOutboundBusRouteId IS NOT NULL OR @packReturnBusRouteId IS NOT NULL)

BEGIN

RAISERROR('You cannot specify both flights and bus routes for the same package.', 16, 1);

RETURN;

END;

IF (@packOutboundFlightId IS NULL AND @packReturnFlightId IS NULL) AND (@packOutboundBusRouteId IS NULL AND @packReturnBusRouteId IS NULL)

BEGIN

RAISERROR('You need to add either a flight or a bus route, outbound and return.', 16, 1);

RETURN;

END;

UPDATE tbl\_package

SET

packTitle = COALESCE(@packTitle, packTitle),

packDescription = COALESCE(@packDescription, packDescription),

packCityId = @packCityId,

packHotId = @packHotId,

packOutboundFlightId = @packOutboundFlightId,

packReturnFlightId = @packReturnFlightId,

packOutboundBusRouteId = @packOutboundBusRouteId,

packReturnBusRouteId = @packReturnBusRouteId,

packGuideId = @packGuideId

WHERE packId = @packId;

PRINT 'Package data has been successfully modified.'

END;

GO

#### up\_update\_hotel\_and\_address

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_hotel\_and\_address

--------------------------------------------------------------------------------------

-- This procedure is used to update the details of a hotel and its address in the database.

--

-- Input Parameters:

-- @hotId: The ID of the hotel to be updated.

-- @hotName: The new name of the hotel. (optional)

-- @hotPricePerNight: The new price per night of the hotel. (optional)

-- @hotTelephoneNo: The new telephone number of the hotel. (optional)

-- @hotContactEmail: The new contact email of the hotel. (optional)

-- @hotStars: The new star rating of the hotel. (optional)

-- @addrCityId: The new city ID of the hotel's address. (optional)

-- @addrPostalCode: The new postal code of the hotel's address. (optional)

-- @addrRegion: The new region of the hotel's address. (optional)

-- @addrStreet: The new street of the hotel's address. (optional)

-- @addrHouseNo: The new house number of the hotel's address. (optional)

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_hotel\_and\_address @hotId = 123, @hotName = 'New Hotel Name', @hotPricePerNight = 150.00

--

-- Result of the action:

-- Hotel details and address have been successfully changed.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_hotel\_and\_address

@hotId INT,

@hotName VARCHAR(30) = NULL,

@hotPricePerNight FLOAT = NULL,

@hotTelephoneNo VARCHAR(20) = NULL,

@hotContactEmail VARCHAR(45) = NULL,

@hotStars INT = NULL,

@addrCityId INT = NULL,

@addrPostalCode VARCHAR(20) = NULL,

@addrRegion VARCHAR(100) = NULL,

@addrStreet VARCHAR(255) = NULL,

@addrHouseNo VARCHAR(10) = NULL

AS

BEGIN

SET NOCOUNT ON;

DECLARE @addrId INT;

IF NOT EXISTS (SELECT 1 FROM tbl\_hotel WHERE hotId = @hotId)

BEGIN

RAISERROR('Hotel with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @hotPricePerNight IS NOT NULL AND @hotPricePerNight <= 0

BEGIN

RAISERROR('Price per night must be greater than zero.', 16, 1);

RETURN;

END;

IF @hotStars IS NOT NULL AND @hotStars<= 0 AND @hotStars > 5

BEGIN

RAISERROR('Hotel stars must be greater than zero and less than or equal to five.', 16, 1);

RETURN;

END;

SELECT @addrId = hotAddrId FROM tbl\_hotel WHERE hotId = @hotId;

IF @addrCityId IS NOT NULL OR @addrPostalCode IS NOT NULL OR @addrRegion IS NOT NULL OR @addrStreet IS NOT NULL OR @addrHouseNo IS NOT NULL

BEGIN

IF @addrId IS NOT NULL

BEGIN

IF @addrCityId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @addrCityId)

BEGIN

RAISERROR('City with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @addrCityId IS NOT NULL AND EXISTS (SELECT 1 FROM tbl\_package WHERE packHotId = @hotId)

BEGIN

RAISERROR('The hotel is associated with one or more packages and the city of its address cannot be changed.', 16, 1);

RETURN;

END;

UPDATE tbl\_address

SET

addrCityId = COALESCE(@addrCityId, addrCityId),

addrPostalCode = COALESCE(@addrPostalCode, addrPostalCode),

addrRegion = COALESCE(@addrRegion, addrRegion),

addrStreet = COALESCE(@addrStreet, addrStreet),

addrHouseNo = COALESCE(@addrHouseNo, addrHouseNo)

WHERE addrId = @addrId;

PRINT 'Hotel address have been successfully changed.';

END

ELSE

BEGIN

RAISERROR('Address with the specified ID does not exist.', 16, 1);

RETURN;

END;

END;

IF @hotName IS NOT NULL OR @hotPricePerNight IS NOT NULL OR @hotTelephoneNo IS NOT NULL OR @hotContactEmail IS NOT NULL OR @hotStars IS NOT NULL

BEGIN

UPDATE tbl\_hotel

SET

hotName = COALESCE(@hotName, hotName),

hotPricePerNight = COALESCE(@hotPricePerNight, hotPricePerNight),

hotTelephoneNo = COALESCE(@hotTelephoneNo, hotTelephoneNo),

hotContactEmail = COALESCE(@hotContactEmail, hotContactEmail),

hotStars = COALESCE(@hotStars, hotStars)

WHERE hotId = @hotId;

PRINT 'Hotel details have been successfully changed.';

END;

END;

GO

#### up\_update\_hotel\_address\_Id

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_hotel\_address\_Id

--------------------------------------------------------------------------------------

-- This procedure is used to update the address ID of a hotel in the database.

--

-- Input Parameters:

-- @hotId: The ID of the hotel to update its address ID.

-- @addrId: The new address ID to be associated with the hotel.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_hotel\_address\_Id @hotId = 123, @addrId = 456

--

-- Result of the action:

-- The hotel address ID has been successfully changed.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_hotel\_address\_Id

@hotId INT,

@addrId INT

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_hotel WHERE hotId = @hotId)

BEGIN

RAISERROR('Hotel with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_address WHERE addrId = @addrId)

BEGIN

RAISERROR('Address with the specified ID does not exist.', 16, 1);

RETURN;

END;

UPDATE tbl\_hotel

SET hotAddrId = @addrId

WHERE hotId = @hotId;

PRINT 'The hotel address ID has been successfully changed';

END;

#### up\_update\_guide

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_guide

--------------------------------------------------------------------------------------

-- This procedure is used to update the details of a guide in the database.

--

-- Input Parameters:

-- @guideId: The ID of the guide to be updated.

-- @guideName: (Optional) The new name of the guide.

-- @guideSurname: (Optional) The new surname of the guide.

-- @guideCityId: (Optional) The new city ID of the guide.

-- @guidePhone: (Optional) The new phone number of the guide.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_guide @guideId = 123, @guideName = 'John', @guideSurname = 'Doe', @guideCityId = 1, @guidePhone = '123456789'

--

-- Result of the action:

-- Guide data was changed successfully.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_guide

@guideId INT,

@guideName VARCHAR(50) = NULL,

@guideSurname VARCHAR(50) = NULL,

@guideCityId INT = NULL,

@guidePhone VARCHAR(20) = NULL

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_guide WHERE guideId = @guideId)

BEGIN

RAISERROR('The guide with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @guideCityId IS NOT NULL

BEGIN

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @guideCityId)

BEGIN

RAISERROR('The specified city ID does not exist.', 16, 1);

RETURN;

END;

IF EXISTS (SELECT 1 FROM tbl\_package WHERE packGuideId = @guideId)

BEGIN

RAISERROR('This guide is associated with one or more packages. The city cannot be changed because one or more packages are related to this city and this city''s guide.', 16, 1);

RETURN;

END;

END;

UPDATE tbl\_guide

SET

guideName = COALESCE(@guideName, guideName),

guideSurname = COALESCE(@guideSurname, guideSurname),

guideCityId = COALESCE(@guideCityId, guideCityId),

guidePhone = COALESCE(@guidePhone, guidePhone)

WHERE guideId = @guideId;

PRINT 'Guide data was changed successfully';

END;

GO

#### up\_update\_flight

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_flight

--------------------------------------------------------------------------------------

-- This procedure is used to update flight information in the database.

-- The procedure checks if the flight ID exists and if the provided data is valid before performing the update.

-- If the flight is associated with packages, certain fields cannot be updated to maintain data consistency.

--

-- Input Parameters:

-- @fliId: The ID of the flight to be updated.

-- @fliStartCityId: (Optional) The ID of the start city for the flight.

-- @fliEndCityId: (Optional) The ID of the end city for the flight.

-- @fliStartTime: (Optional) The departure time of the flight.

-- @fliEndTime: (Optional) The arrival time of the flight.

-- @fliClass: (Optional) The class of the flight (First, Business, Economy).

-- @fliPrice: (Optional) The price of the flight.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_flight @fliId = 1, @fliPrice = 500.00

--

-- Result of the action:

-- Flight data was changed successfully.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_flight

@fliId INT,

@fliStartCityId INT = NULL,

@fliEndCityId INT = NULL,

@fliStartTime DATETIME = NULL,

@fliEndTime DATETIME = NULL,

@fliClass VARCHAR(10) = NULL,

@fliPrice FLOAT = NULL

AS

BEGIN

SET NOCOUNT ON;

DECLARE @packageOutboundCount INT;

DECLARE @packageReturnCount INT;

IF NOT EXISTS (SELECT 1 FROM tbl\_flight WHERE fliId = @fliId)

BEGIN

RAISERROR('The flight with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @fliClass IS NOT NULL AND @fliClass NOT IN ('First', 'Business', 'Economy')

BEGIN

RAISERROR('Invalid flight class. Class must be First, Business, or Economy.', 16, 1);

RETURN;

END;

IF @fliPrice IS NOT NULL AND @fliPrice <= 0

BEGIN

RAISERROR('Flight price must be greater than zero.', 16, 1);

RETURN;

END;

DECLARE @timeDifference INT;

IF @fliStartTime IS NOT NULL OR @fliEndTime IS NOT NULL

BEGIN

DECLARE @currentStartTime DATETIME;

DECLARE @currentEndTime DATETIME;

SELECT @currentStartTime = fliStartTime, @currentEndTime = fliEndTime FROM tbl\_flight WHERE fliId = @fliId;

SET @fliStartTime = COALESCE(@fliStartTime, @currentStartTime);

SET @fliEndTime = COALESCE(@fliEndTime, @currentEndTime);

SET @timeDifference = DATEDIFF(HOUR, @fliStartTime, @fliEndTime);

IF @timeDifference > 19 OR @timeDifference < 1

BEGIN

RAISERROR('Time difference between start and end cannot exceed 19 hours and be less than 1 hour.', 16, 1);

RETURN;

END;

END;

IF @fliStartCityId IS NULL AND @fliEndCityId IS NULL

BEGIN

UPDATE tbl\_flight

SET

fliStartTime = COALESCE(@fliStartTime, fliStartTime),

fliEndTime = COALESCE(@fliEndTime, fliEndTime),

fliClass = COALESCE(@fliClass, fliClass),

fliPrice = COALESCE(@fliPrice, fliPrice)

WHERE fliId = @fliId;

END

ELSE

BEGIN

SELECT @packageOutboundCount = COUNT(\*)

FROM tbl\_package

WHERE packOutboundFlightId = @fliId;

SELECT @packageReturnCount = COUNT(\*)

FROM tbl\_package

WHERE packReturnFlightId = @fliId;

IF @packageOutboundCount > 0 AND @fliEndCityId IS NOT NULL

BEGIN

RAISERROR('The flight is associated with one or more packages and cannot have its end city changed.', 16, 1);

RETURN;

END

ELSE IF @packageReturnCount > 0 AND @fliStartCityId IS NOT NULL

BEGIN

RAISERROR('The flight is associated with one or more packages and cannot have its start city changed.', 16, 1);

RETURN;

END

ELSE

BEGIN

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @fliStartCityId) OR NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @fliEndCityId)

BEGIN

RAISERROR('One or both of the provided city IDs do not exist.', 16, 1);

RETURN;

END;

UPDATE tbl\_flight

SET

fliStartCityId = COALESCE(@fliStartCityId, fliStartCityId),

fliEndCityId = COALESCE(@fliEndCityId, fliEndCityId),

fliStartTime = COALESCE(@fliStartTime, fliStartTime),

fliEndTime = COALESCE(@fliEndTime, fliEndTime),

fliClass = COALESCE(@fliClass, fliClass),

fliPrice = COALESCE(@fliPrice, fliPrice)

WHERE fliId = @fliId;

END;

END;

END;

GO

#### up\_update\_driver

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_driver

--------------------------------------------------------------------------------------

-- This procedure is used to update the details of a driver in the database.

--

-- Input Parameters:

-- @driverId: The ID of the driver to be updated.

-- @driverName: (Optional) The new name of the driver.

-- @driverSurname: (Optional) The new surname of the driver.

-- @driverPhone: (Optional) The new phone number of the driver.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_driver @driverId = 123, @driverName = 'John', @driverSurname = 'Doe', @driverPhone = '123456789'

--

-- Result of the action:

-- Driver details have been successfully changed.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_driver

@driverId INT,

@driverName VARCHAR(50) = NULL,

@driverSurname VARCHAR(50) = NULL,

@driverPhone VARCHAR(20) = NULL

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_driver WHERE driverId = @driverId)

BEGIN

RAISERROR('The driver with the specified ID does not exist.', 16, 1);

RETURN;

END;

UPDATE tbl\_driver

SET

driverName = COALESCE(@driverName, driverName),

driverSurname = COALESCE(@driverSurname, driverSurname),

driverPhone = COALESCE(@driverPhone, driverPhone)

WHERE driverId = @driverId;

PRINT 'Driver details have been successfully changed.';

END;

GO

#### up\_update\_customer\_and\_address

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_customer\_and\_address

--------------------------------------------------------------------------------------

-- This procedure is used to update customer and address details in the database.

-- It allows updating various attributes of a customer and their associated address.

--

-- Input Parameters:

-- @custId: The ID of the customer to be updated.

-- @custName: (Optional) The new name of the customer.

-- @custSurname: (Optional) The new surname of the customer.

-- @custPhone: (Optional) The new phone number of the customer.

-- @custEmailAddress: (Optional) The new email address of the customer.

-- @addrCityId: (Optional) The new city ID for the address.

-- @addrPostalCode: (Optional) The new postal code of the address.

-- @addrRegion: (Optional) The new region of the address.

-- @addrStreet: (Optional) The new street of the address.

-- @addrHouseNo: (Optional) The new house number of the address.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_customer\_and\_address

-- @custId = 123,

-- @custName = 'John',

-- @custSurname = 'Doe',

-- @custPhone = '987654321',

-- @custEmailAddress = 'john.doe@example.com',

-- @addrCityId = 456,

-- @addrPostalCode = '54321',

-- @addrRegion = 'New Region',

-- @addrStreet = 'New Street',

-- @addrHouseNo = '20'

--

-- Result of the action:

-- Customer details or customer address have been successfully changed

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_customer\_and\_address

@custId INT,

@custName VARCHAR(50) = NULL,

@custSurname VARCHAR(50) = NULL,

@custPhone VARCHAR(20) = NULL,

@custEmailAddress VARCHAR(255) = NULL,

@addrCityId INT = NULL,

@addrPostalCode VARCHAR(20) = NULL,

@addrRegion VARCHAR(100) = NULL,

@addrStreet VARCHAR(255) = NULL,

@addrHouseNo VARCHAR(10) = NULL

AS

BEGIN

SET NOCOUNT ON;

DECLARE @addrId INT;

IF NOT EXISTS (SELECT 1 FROM tbl\_customer WHERE custId = @custId)

BEGIN

RAISERROR('Customer with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @addrCityId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @addrCityId)

BEGIN

RAISERROR('The specified city ID does not exist.', 16, 1);

RETURN;

END;

IF @custEmailAddress IS NOT NULL AND

@custEmailAddress NOT LIKE '%@%.%' AND

PATINDEX('%[^a-zA-Z0-9.@]%', @custEmailAddress) <> 0

BEGIN

RAISERROR('Invalid email address format.', 16, 1);

RETURN;

END;

SELECT @addrId = custAddrId FROM tbl\_customer WHERE custId = @custId;

IF @addrId IS NOT NULL

BEGIN

IF @addrCityId IS NOT NULL

UPDATE tbl\_address SET addrCityId = @addrCityId WHERE addrId = @addrId;

IF @addrPostalCode IS NOT NULL

UPDATE tbl\_address SET addrPostalCode = @addrPostalCode WHERE addrId = @addrId;

IF @addrRegion IS NOT NULL

UPDATE tbl\_address SET addrRegion = @addrRegion WHERE addrId = @addrId;

IF @addrStreet IS NOT NULL

UPDATE tbl\_address SET addrStreet = @addrStreet WHERE addrId = @addrId;

IF @addrHouseNo IS NOT NULL

UPDATE tbl\_address SET addrHouseNo = @addrHouseNo WHERE addrId = @addrId;

END;

IF @custName IS NOT NULL

UPDATE tbl\_customer SET custName = @custName WHERE custId = @custId;

IF @custSurname IS NOT NULL

UPDATE tbl\_customer SET custSurname = @custSurname WHERE custId = @custId;

IF @custPhone IS NOT NULL

UPDATE tbl\_customer SET custPhone = @custPhone WHERE custId = @custId;

IF @custEmailAddress IS NOT NULL

UPDATE tbl\_customer SET custEmailAddress = @custEmailAddress WHERE custId = @custId;

PRINT 'Customer details or customer address have been successfully changed';

END;

#### up\_update\_customer\_address\_Id

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_customer\_address\_Id

--------------------------------------------------------------------------------------

-- This procedure is used to update the address ID of a customer in the database.

-- It allows associating a different address with an existing customer.

--

-- Input Parameters:

-- @custId: The ID of the customer whose address ID is to be updated.

-- @addrId: The new address ID to be associated with the customer.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_customer\_address\_Id @custId = 123, @addrId = 456

--

-- Result of the action:

-- The customer address ID has been successfully changed

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_customer\_address\_Id

@custId INT,

@addrId INT

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_customer WHERE custId = @custId)

BEGIN

RAISERROR('Customer with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_address WHERE addrId = @addrId)

BEGIN

RAISERROR('Address with the specified ID does not exist.', 16, 1);

RETURN;

END;

UPDATE tbl\_customer

SET custAddrId = @addrId

WHERE custId = @custId;

PRINT 'The customer address ID has been successfully changed';

END;

#### up\_update\_busRoute

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_busRoute

--------------------------------------------------------------------------------------

-- This procedure is used to update the details of a bus route in the database.

-- If the bus route is associated with one or more packages and the start city or end city is being updated, an error is raised.

--

-- Input Parameters:

-- @busRouteId: The ID of the bus route to be updated (required).

-- @driverId: The ID of the driver assigned to the bus route (optional).

-- @busRouteStartCity: The ID of the start city for the bus route (optional).

-- @busRouteEndCity: The ID of the end city for the bus route (optional).

-- @busRouteStartTime: The start time of the bus route (optional).

-- @busRouteEndTime: The end time of the bus route (optional).

-- @busRouteBusID: The ID of the bus assigned to the bus route (optional).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_busRoute @busRouteId = 1, @driverId = 2, @busRouteStartTime = '2024-04-01 08:00:00', @busRouteEndTime = '2024-04-01 16:00:00';

--

-- Result of the action:

-- The details of the bus route have been successfully updated.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_busRoute

@busRouteId INT,

@driverId INT = NULL,

@busRouteStartCity INT = NULL,

@busRouteEndCity INT = NULL,

@busRouteStartTime DATETIME = NULL,

@busRouteEndTime DATETIME = NULL,

@busRouteBusID INT = NULL

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_busroute WHERE busRouteId = @busRouteId)

BEGIN

RAISERROR('The bus route with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @driverId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_driver WHERE driverId = @driverId)

BEGIN

RAISERROR('The driver with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @busRouteBusID IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_bus WHERE busId = @busRouteBusID)

BEGIN

RAISERROR('The bus with the specified ID does not exist.', 16, 1);

RETURN;

END;

DECLARE @timeDifference INT;

IF @busRouteStartTime IS NOT NULL OR @busRouteEndTime IS NOT NULL

BEGIN

DECLARE @currentStartTime DATETIME;

DECLARE @currentEndTime DATETIME;

SELECT @currentStartTime = busRouteStartTime, @currentEndTime = busRouteEndTime FROM tbl\_busroute WHERE busRouteId = @busRouteId;

SET @busRouteStartTime = COALESCE(@busRouteStartTime, @currentStartTime);

SET @busRouteEndTime = COALESCE(@busRouteEndTime, @currentEndTime);

SET @timeDifference = DATEDIFF(DAY, @busRouteStartTime, @busRouteEndTime);

IF @timeDifference > 3

BEGIN

RAISERROR('The difference between start time and end time cannot exceed 3 days.', 16, 1);

RETURN;

END;

END;

IF @busRouteStartCity IS NULL AND @busRouteEndCity IS NULL

BEGIN

UPDATE tbl\_busroute

SET

driverId = COALESCE(@driverId, driverId),

busRouteStartTime = COALESCE(@busRouteStartTime, busRouteStartTime),

busRouteEndTime = COALESCE(@busRouteEndTime, busRouteEndTime),

busRouteBusID = COALESCE(@busRouteBusID, busRouteBusID)

WHERE busRouteId = @busRouteId;

END

ELSE

BEGIN

DECLARE @packageOutboundCount INT;

SELECT @packageOutboundCount = COUNT(\*)

FROM tbl\_package

WHERE packOutboundBusRouteId = @busRouteId;

DECLARE @packageReturnCount INT;

SELECT @packageReturnCount = COUNT(\*)

FROM tbl\_package

WHERE packOutboundBusRouteId = @busRouteId;

IF @packageOutboundCount > 0 AND @busRouteEndCity IS NOT NULL

BEGIN

RAISERROR('The bus route is associated with one or more packages and cannot have its end city changed.', 16, 1);

RETURN;

END

ELSE IF @packageReturnCount > 0 AND @busRouteStartCity IS NOT NULL

BEGIN

RAISERROR('The bus route is associated with one or more packages and cannot have its start city changed.', 16, 1);

RETURN;

END

ELSE

BEGIN

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @busRouteStartCity) OR NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @busRouteEndCity)

BEGIN

RAISERROR('One or both of the provided city IDs do not exist.', 16, 1);

RETURN;

END;

UPDATE tbl\_busroute

SET

driverId = COALESCE(@driverId, driverId),

busRouteStartCityId = COALESCE(@busRouteStartCity, busRouteStartCityId),

busRouteEndCityId = COALESCE(@busRouteEndCity, busRouteEndCityId),

busRouteStartTime = COALESCE(@busRouteStartTime, busRouteStartTime),

busRouteEndTime = COALESCE(@busRouteEndTime, busRouteEndTime),

busRouteBusID = COALESCE(@busRouteBusID, busRouteBusID)

WHERE busRouteId = @busRouteId;

END;

END;

END;

GO

#### up\_update\_bus

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_bus

--------------------------------------------------------------------------------------

-- This procedure is used to update the details of a bus in the database.

-- It requires providing the ID of the bus to be updated and optionally the new values for its attributes.

-- If the bus with the specified ID does not exist, an error is raised.

-- If a new bus number is provided and it already exists for another bus, an error is raised.

--

-- Input Parameters:

-- @busId: The ID of the bus to be updated (required).

-- @busNo: The new number of the bus (optional).

-- @busModel: The new model of the bus (optional).

-- @busColor: The new color of the bus (optional).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_bus @busId = 123, @busNo = 'ABC1234', @busModel = 'Model X'

--

-- Result of the action:

-- The details of the bus have been successfully updated.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_bus

@busId INT,

@busNo VARCHAR(20) = NULL,

@busModel VARCHAR(50) = NULL,

@busColor VARCHAR(20) = NULL

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_bus WHERE busId = @busId)

BEGIN

RAISERROR('The bus with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @busNo IS NOT NULL AND EXISTS (SELECT 1 FROM tbl\_bus WHERE busNo = @busNo AND busId != @busId)

BEGIN

RAISERROR('The bus number specified already exists.', 16, 1);

RETURN;

END;

UPDATE tbl\_bus

SET

busNo = CASE WHEN @busNo IS NOT NULL THEN @busNo ELSE busNo END,

busModel = CASE WHEN @busModel IS NOT NULL THEN @busModel ELSE busModel END,

busColor = CASE WHEN @busColor IS NOT NULL THEN @busColor ELSE busColor END

WHERE busId = @busId;

END;

GO

#### up\_update\_booking

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_booking

--------------------------------------------------------------------------------------

-- This procedure is used to update the details of a booking in the database based on the booking ID.

--

-- Input Parameters:

-- @bookId: The ID of the booking to be updated (required).

-- @bookCustId: The ID of the customer associated with the booking (optional).

-- @bookPackageId: The ID of the package associated with the booking (optional).

-- @bookDiscountPercent: The discount percentage applied to the booking (optional).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_booking @bookId = 1, @bookPackageId = 5, @bookDiscountPercent = 0.1;

--

-- Result of the action:

-- Booking data has been successfully modified.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_booking

@bookId INT,

@bookCustId INT = NULL,

@bookPackageId INT = NULL,

@bookDiscountPercent FLOAT = NULL

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_booking WHERE bookId = @bookId)

BEGIN

RAISERROR('The specified booking ID does not exist.', 16, 1);

RETURN;

END;

IF @bookCustId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_customer WHERE custId = @bookCustId)

BEGIN

RAISERROR('The specified customer ID does not exist.', 16, 1);

RETURN;

END;

IF @bookPackageId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_package WHERE packId = @bookPackageId)

BEGIN

RAISERROR('The specified package ID does not exist.', 16, 1);

RETURN;

END;

SELECT @bookPackageId = COALESCE(@bookPackageId, bookPackageId) FROM tbl\_booking WHERE bookId = @bookId;

SELECT @bookDiscountPercent = COALESCE(@bookDiscountPercent, bookDiscountPercent) FROM tbl\_booking WHERE bookId = @bookId;

UPDATE tbl\_booking

SET

bookCustId = COALESCE(@bookCustId, bookCustId),

bookPackageId = @bookPackageId,

bookDiscountPercent = @bookDiscountPercent

WHERE bookId = @bookId;

PRINT 'Booking data has been successfully modified.'

END;

GO

#### up\_update\_address

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_update\_address

--------------------------------------------------------------------------------------

-- This procedure is used to update the details of an existing address in the database.

--

-- Input Parameters:

-- @addrId: The ID of the address to be updated.

-- @addrCityId: The ID of the city associated with the address (optional).

-- @addrPostalCode: The postal code of the address (optional).

-- @addrRegion: The region of the address (optional).

-- @addrStreet: The street of the address (optional).

-- @addrHouseNo: The house number of the address (optional).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_update\_address @addrId = 123, @addrCityId = 456, @addrPostalCode = '12345', @addrRegion = 'Region', @addrStreet = 'Street', @addrHouseNo = '10'

--

-- Result of the action:

-- Address details have been successfully modified.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_update\_address

@addrId INT,

@addrCityId INT = NULL,

@addrPostalCode VARCHAR(20) = NULL,

@addrRegion VARCHAR(100) = NULL,

@addrStreet VARCHAR(255) = NULL,

@addrHouseNo VARCHAR(10) = NULL

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_address WHERE addrId = @addrId)

BEGIN

RAISERROR('Address with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @addrCityId IS NOT NULL

BEGIN

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @addrCityId)

BEGIN

RAISERROR('The specified city ID does not exist.', 16, 1);

RETURN;

END;

IF EXISTS (SELECT 1 FROM tbl\_package WHERE packHotId IN (SELECT hotId FROM tbl\_hotel WHERE hotAddrId = @addrId))

BEGIN

RAISERROR('This address is associated with one or more hotels, which are associated with one or more packages. At this moment, it is not possible to change the city of this address.', 16, 1);

RETURN;

END;

END;

UPDATE tbl\_address

SET

addrCityId = COALESCE(@addrCityId, addrCityId),

addrPostalCode = COALESCE(@addrPostalCode, addrPostalCode),

addrRegion = COALESCE(@addrRegion, addrRegion),

addrStreet = COALESCE(@addrStreet, addrStreet),

addrHouseNo = COALESCE(@addrHouseNo, addrHouseNo)

WHERE addrId = @addrId;

PRINT 'Address details have been successfully modified.';

END;

GO

## DELETE

#### up\_delete\_package

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_delete\_package

--------------------------------------------------------------------------------------

-- This procedure is used to delete a package from the database.

-- The procedure first checks if the package ID exists and whether the package is associated with any bookings.

-- If the package is associated with bookings, it cannot be deleted, and an error is raised.

-- Otherwise, the package is deleted from the database.

--

-- Input Parameters:

-- @packId: The ID of the package to be deleted (required).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delete\_package @packId = 1;

--

-- Result of the action:

-- The package has been successfully deleted.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delete\_package

@packId INT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @confirmation CHAR(1);

IF NOT EXISTS (SELECT 1 FROM tbl\_package WHERE packId = @packId)

BEGIN

RAISERROR('The specified package ID does not exist.', 16, 1);

RETURN;

END;

IF EXISTS (SELECT 1 FROM tbl\_booking WHERE bookPackageId = @packId)

BEGIN

RAISERROR('The package is associated with one or more bookings and cannot be deleted.', 16, 1);

RETURN;

END;

DELETE FROM tbl\_package WHERE packId = @packId;

PRINT 'The package has been successfully deleted.';

END;

GO

#### up\_delete\_hotel\_and\_address

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_delete\_hotel\_and\_address

--------------------------------------------------------------------------------------

-- This procedure is used to delete a hotel along with its address from the database.

--

-- Input Parameters:

-- @hotId: The ID of the hotel to be deleted.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delete\_hotel\_and\_address @hotId = 123

--

-- Result of the action:

-- The hotel and its address have been successfully deleted.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delete\_hotel\_and\_address

@hotId INT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @addrId INT;

IF NOT EXISTS (SELECT 1 FROM tbl\_hotel WHERE hotId = @hotId)

BEGIN

RAISERROR('Hotel with the specified ID does not exist.', 16, 1);

RETURN;

END;

SELECT @addrId = hotAddrId FROM tbl\_hotel WHERE hotId = @hotId;

IF EXISTS (SELECT 1 FROM tbl\_package WHERE packHotId = @hotId)

BEGIN

RAISERROR('The hotel is associated with one or more packages and cannot be deleted.', 16, 1);

RETURN;

END;

DELETE FROM tbl\_hotel WHERE hotId = @hotId;

DELETE FROM tbl\_address WHERE addrId = @addrId;

PRINT 'The hotel and its address have been successfully deleted.';

END;

GO

#### up\_delete\_guide

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_delete\_guide

--------------------------------------------------------------------------------------

-- This procedure is used to delete a guide from the database.

--

-- Input Parameters:

-- @guideId: The ID of the guide to be deleted.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delete\_guide @guideId = 123

--

-- Result of the action:

-- The guide has been successfully deleted.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delete\_guide

@guideId INT

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_guide WHERE guideId = @guideId)

BEGIN

RAISERROR('The guide with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF EXISTS (SELECT 1 FROM tbl\_package WHERE packGuideId = @guideId)

BEGIN

RAISERROR('The guide is associated with one or more packages and cannot be deleted.', 16, 1);

RETURN;

END;

DELETE FROM tbl\_guide WHERE guideId = @guideId;

PRINT 'The guide has been successfully deleted.';

END;

GO

#### up\_delete\_flight

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_delete\_flight

--------------------------------------------------------------------------------------

-- This procedure is used to delete a flight from the database.

--

-- Input Parameters:

-- @fliId: The ID of the flight to be deleted.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delete\_flight @fliId = 1

--

-- Result of the action:

-- The flight has been successfully deleted.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delete\_flight

@fliId INT

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_flight WHERE fliId = @fliId)

BEGIN

RAISERROR('Flight with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF EXISTS (SELECT 1 FROM tbl\_package WHERE packOutboundFlightId = @fliId OR packReturnFlightId = @fliId)

BEGIN

RAISERROR('The flight is associated with one or more packages and cannot be deleted.', 16, 1);

RETURN;

END;

DELETE FROM tbl\_flight WHERE fliId = @fliId;

PRINT 'The flight has been successfully deleted.';

END;

GO

#### up\_delete\_driver

----------------------------------------------------------------------------------------- PROCEDURE DEFINITION

--- up\_delete\_driver

--------------------------------------------------------------------------------------

-- This procedure is used to delete a driver from the database.

--

-- Input Parameters:

-- @driverId: The ID of the driver to be deleted.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delete\_driver @driverId = 123

--

-- Result of the action:

-- The driver has been successfully deleted.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delete\_driver

@driverId INT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @isReferenced BIT;

IF NOT EXISTS (SELECT 1 FROM tbl\_driver WHERE driverId = @driverId)

BEGIN

RAISERROR('The driver with the specified ID does not exist.', 16, 1);

RETURN;

END;

SELECT @isReferenced = CASE

WHEN EXISTS (SELECT 1 FROM tbl\_busroute WHERE driverId = @driverId) THEN 1

ELSE 0

END;

IF @isReferenced = 1

BEGIN

RAISERROR('The driver is associated with one or more bus routes and cannot be deleted.', 16, 1);

RETURN;

END;

DELETE FROM tbl\_driver WHERE driverId = @driverId;

PRINT 'The driver has been successfully deleted.';

END;

GO

#### up\_delete\_customer\_and\_address

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_delete\_customer\_and\_address

--------------------------------------------------------------------------------------

-- This procedure deletes a customer along with their address from the database.

--

-- Input Parameters:

-- @custId: INT - The ID of the customer to be deleted.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delete\_customer\_and\_address @custId = 123

--

-- Sample Output:

-- The client has been successfully deleted.

-- The customer's address has been successfully deleted.

--

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delete\_customer\_and\_address

@custId INT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @addrId INT;

IF NOT EXISTS (SELECT 1 FROM tbl\_customer WHERE custId = @custId)

BEGIN

RAISERROR('Customer with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF EXISTS (SELECT 1 FROM tbl\_booking WHERE bookCustId = @custId)

BEGIN

RAISERROR('The customer is associated with one or more bookings and cannot be deleted.', 16, 1);

RETURN;

END;

SELECT @addrId = custAddrId FROM tbl\_customer WHERE custId = @custId;

DELETE FROM tbl\_customer WHERE custId = @custId;

PRINT 'The client has been successfully deleted.';

DELETE FROM tbl\_address WHERE addrId = @addrId;

PRINT 'The customer''s address has been successfully deleted.';

END;

GO

#### up\_delete\_busRoute

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_delete\_busRoute

--------------------------------------------------------------------------------------

-- This procedure is used to delete a bus route from the database.

-- It requires providing the bus route ID as input.

-- If the specified bus route ID does not exist or if the bus route is associated with one or more packages, an error is raised.

-- Otherwise, the bus route is deleted from the database.

--

-- Input Parameters:

-- @busRouteId: The ID of the bus route to be deleted (required).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delete\_busRoute @busRouteId = 1;

--

-- Result of the action:

-- The bus route has been successfully deleted.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delete\_busRoute

@busRouteId INT

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_busroute WHERE busRouteId = @busRouteId)

BEGIN

RAISERROR('The bus route with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF EXISTS (SELECT 1 FROM tbl\_package WHERE packOutboundBusRouteId = @busRouteId OR packReturnBusRouteId = @busRouteId)

BEGIN

RAISERROR('The bus route is associated with one or more packages and cannot be deleted.', 16, 1);

RETURN;

END;

DELETE FROM tbl\_busroute WHERE busRouteId = @busRouteId;

PRINT 'The bus route has been successfully deleted.';

END;

GO

#### up\_delete\_bus

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_delete\_bus

--------------------------------------------------------------------------------------

-- This procedure is used to delete a bus from the database.

-- It requires providing the ID of the bus to be deleted.

-- If the bus with the specified ID does not exist or is associated with a route, an error is raised.

--

-- Input Parameters:

-- @busId: The ID of the bus to be deleted (required).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delete\_bus @busId = 123

--

-- Result of the action:

-- The bus has been successfully deleted.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delete\_bus

@busId INT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @isAssociatedWithRoute BIT;

IF NOT EXISTS (SELECT 1 FROM tbl\_bus WHERE busId = @busId)

BEGIN

RAISERROR('The bus with the specified ID does not exist.', 16, 1);

RETURN;

END;

SELECT @isAssociatedWithRoute = CASE

WHEN EXISTS (SELECT 1 FROM tbl\_busroute WHERE busRouteBusID = @busId) THEN 1

ELSE 0

END;

IF @isAssociatedWithRoute = 1

BEGIN

RAISERROR('The bus is associated with one or more routes and cannot be deleted.', 16, 1);

RETURN;

END;

DELETE FROM tbl\_bus WHERE busId = @busId;

PRINT 'The bus has been successfully deleted.';

END;

GO

#### up\_delete\_booking

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_delete\_booking

--------------------------------------------------------------------------------------

-- This procedure is used to delete a booking from the database based on the booking ID.

--

-- Input Parameters:

-- @bookId: The ID of the booking to be deleted (required).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delete\_booking @bookId = 1;

--

-- Result of the action:

-- The booking has been successfully deleted.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delete\_booking

@bookId INT

AS

BEGIN

SET NOCOUNT ON;

DECLARE @confirmation CHAR(1);

IF NOT EXISTS (SELECT 1 FROM tbl\_booking WHERE bookId = @bookId)

BEGIN

RAISERROR('The specified booking ID does not exist.', 16, 1);

RETURN;

END;

DELETE FROM tbl\_booking WHERE bookId = @bookId;

PRINT 'The booking has been successfully deleted.';

END;

GO

#### up\_delete\_address

----------------------------------------------------------------------------------------- PROCEDURE DEFINITION

--- up\_delete\_address

--------------------------------------------------------------------------------------

-- This procedure is used to delete an address from the database.

--

-- Input Parameters:

-- @addrId: The ID of the address to be deleted.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delete\_address @addrId = 123

--

-- Result of the action:

-- The address was successfully deleted.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delete\_address

@addrId INT

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_address WHERE addrId = @addrId)

BEGIN

RAISERROR('Address with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF EXISTS (SELECT 1 FROM tbl\_customer WHERE custAddrId = @addrId)

BEGIN

RAISERROR('The address is still associated with one or more customers and cannot be deleted.', 16, 1);

RETURN;

END;

IF EXISTS (SELECT 1 FROM tbl\_hotel WHERE hotAddrId = @addrId)

BEGIN

RAISERROR('The address is still associated with one or more hotels and cannot be deleted.', 16, 1);

RETURN;

END;

DELETE FROM tbl\_address WHERE addrId = @addrId;

PRINT 'The address was successfully deleted.';

END;

GO

## ADD

#### up\_add\_package

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_add\_package

--------------------------------------------------------------------------------------

-- This procedure is used to add a new package to the database.

-- It ensures consistency between the package details and the associated entities, such as verifying that the hotel and guide are in the same city as the package.

-- The procedure also checks for the existence of outbound and return flights or bus routes, requiring both to be specified.

-- Additionally, it calculates the start and end dates of the package based on the provided flights or bus routes and verifies that the duration is between 3 and 30 days.

--

-- Input Parameters:

-- @packTitle: The title of the package (required).

-- @packDescription: The description of the package.

-- @packCityId: The ID of the city where the package is offered (required).

-- @packHotId: The ID of the hotel associated with the package (required).

-- @packOutboundFlightId: The ID of the outbound flight associated with the package (optional).

-- @packReturnFlightId: The ID of the return flight associated with the package (optional).

-- @packOutboundBusRouteId: The ID of the outbound bus route associated with the package (optional).

-- @packReturnBusRouteId: The ID of the return bus route associated with the package (optional).

-- @packGuideId: The ID of the guide associated with the package (required).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_add\_package @packTitle = 'Summer Vacation', @packDescription = 'Enjoy a relaxing summer vacation by the beach.',

-- @packCityId = 1, @packHotId = 101, @packOutboundFlightId = 201, @packReturnFlightId = 202, @packGuideId = 301;

--

-- Result of the action:

-- The package has been successfully added.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_add\_package

@packTitle VARCHAR(30),

@packDescription TEXT,

@packCityId INT,

@packHotId INT,

@packOutboundFlightId INT = NULL,

@packReturnFlightId INT = NULL,

@packOutboundBusRouteId INT = NULL,

@packReturnBusRouteId INT = NULL,

@packGuideId INT

AS

BEGIN

SET NOCOUNT ON;

IF LEN(@packTitle) = 0

BEGIN

RAISERROR('Package title cannot be empty.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @packCityId)

BEGIN

RAISERROR('The specified city ID does not exist.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_hotel WHERE hotId = @packHotId)

BEGIN

RAISERROR('The specified hotel ID does not exist.', 16, 1);

RETURN;

END;

IF EXISTS (

SELECT 1

FROM tbl\_hotel AS H

INNER JOIN tbl\_address AS A ON H.hotAddrId = A.addrId

WHERE H.hotId = @packHotId AND A.addrCityId <> @packCityId

)

BEGIN

RAISERROR('The city of the specified hotel does not match the package city.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_guide WHERE guideId = @packGuideId)

BEGIN

RAISERROR('The specified guide ID does not exist.', 16, 1);

RETURN;

END;

IF EXISTS (

SELECT 1

FROM tbl\_guide AS G

INNER JOIN tbl\_city AS C ON G.guideCityId = C.cityId

WHERE G.guideId = @packGuideId AND C.cityId <> @packCityId

)

BEGIN

RAISERROR('The city of the specified guide does not match the package city.', 16, 1);

RETURN;

END;

IF (@packOutboundFlightId IS NULL AND @packReturnFlightId IS NOT NULL) OR (@packOutboundBusRouteId IS NOT NULL AND @packReturnBusRouteId IS NULL)

BEGIN

RAISERROR('You need to add either a flight or a bus route.', 16, 1);

RETURN;

END;

IF (@packOutboundFlightId IS NOT NULL AND @packReturnFlightId IS NULL) OR (@packOutboundBusRouteId IS NULL AND @packReturnBusRouteId IS NOT NULL)

BEGIN

RAISERROR('You need to add either a flight or a bus route.', 16, 1);

RETURN;

END;

IF (@packOutboundFlightId IS NOT NULL AND @packReturnFlightId IS NULL) OR (@packOutboundFlightId IS NULL AND @packReturnFlightId IS NOT NULL)

BEGIN

RAISERROR('You need to specify both outbound and return flights.', 16, 1);

RETURN;

END;

IF (@packOutboundBusRouteId IS NOT NULL AND @packReturnBusRouteId IS NULL) OR (@packOutboundBusRouteId IS NULL AND @packReturnBusRouteId IS NOT NULL)

BEGIN

RAISERROR('You need to specify both outbound and return bus routes.', 16, 1);

RETURN;

END;

IF @packOutboundFlightId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_flight WHERE fliId = @packOutboundFlightId)

BEGIN

RAISERROR('The specified outbound flight ID does not exist.', 16, 1);

RETURN;

END;

IF @packReturnFlightId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_flight WHERE fliId = @packReturnFlightId)

BEGIN

RAISERROR('The specified return flight ID does not exist.', 16, 1);

RETURN;

END;

IF @packOutboundBusRouteId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_busroute WHERE busRouteId = @packOutboundBusRouteId)

BEGIN

RAISERROR('The specified outbound bus route ID does not exist.', 16, 1);

RETURN;

END;

IF @packReturnBusRouteId IS NOT NULL AND NOT EXISTS (SELECT 1 FROM tbl\_busroute WHERE busRouteId = @packReturnBusRouteId)

BEGIN

RAISERROR('The specified return bus route ID does not exist.', 16, 1);

RETURN;

END;

DECLARE @packStartDate DATE;

IF @packOutboundFlightId IS NOT NULL

SELECT @packStartDate = fliEndTime FROM tbl\_flight WHERE fliId = @packOutboundFlightId

ELSE IF @packOutboundBusRouteId IS NOT NULL

SELECT @packStartDate = busRouteEndTime FROM tbl\_busroute WHERE busRouteId = @packOutboundBusRouteId

DECLARE @packEndDate DATE;

IF @packReturnFlightId IS NOT NULL

SELECT @packEndDate = fliStartTime FROM tbl\_flight WHERE fliId = @packReturnFlightId

ELSE IF @packReturnBusRouteId IS NOT NULL

SELECT @packEndDate = busRouteStartTime FROM tbl\_busroute WHERE busRouteId = @packReturnBusRouteId

IF (DATEDIFF(DAY, @packStartDate, @packEndDate) + 1) < 3 OR (DATEDIFF(DAY, @packStartDate, @packEndDate) + 1) > 30

BEGIN

RAISERROR('The difference in days between the start date and end date must be greater than 3 and less than 30.', 16, 1);

RETURN;

END;

INSERT INTO tbl\_package(packTitle, packDescription, packCityId, packHotId, packOutboundFlightId, packReturnFlightId, packOutboundBusRouteId, packReturnBusRouteId, packGuideId)

VALUES (@packTitle, @packDescription, @packCityId, @packHotId, @packOutboundFlightId, @packReturnFlightId, @packOutboundBusRouteId, @packReturnBusRouteId, @packGuideId);

PRINT 'The package has been successfully added.';

END;

GO

#### up\_add\_hotel\_and\_address

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_add\_hotel\_and\_address

--------------------------------------------------------------------------------------

-- This procedure is used to add a new hotel along with its address to the database.

--

-- Input Parameters:

-- @hotName: The name of the hotel.

-- @hotPricePerNight: The price per night for staying at the hotel.

-- @hotTelephoneNo: The telephone number of the hotel.

-- @hotContactEmail: The contact email of the hotel.

-- @hotStars: The star rating of the hotel.

-- @cityId: The ID of the city where the hotel is located.

-- @addrPostalCode: The postal code of the hotel's address.

-- @addrRegion: The region of the hotel's address.

-- @addrStreet: The street of the hotel's address.

-- @addrHouseNo: The house number of the hotel's address.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_add\_hotel\_and\_address 'Example Hotel', 150.00, '123456789', 'example@example.com', 4, 1, '12345', 'Region', 'Main Street', '10'

--

-- Result of the action:

-- The hotel and its address have been successfully added.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_add\_hotel\_and\_address

@hotName VARCHAR(30),

@hotPricePerNight FLOAT,

@hotTelephoneNo VARCHAR(20),

@hotContactEmail VARCHAR(45),

@hotStars INT,

@cityId INT = NULL,

@addrPostalCode VARCHAR(20),

@addrRegion VARCHAR(100),

@addrStreet VARCHAR(255),

@addrHouseNo VARCHAR(10)

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRY

IF @hotName IS NULL OR @hotPricePerNight IS NULL

BEGIN

RAISERROR('Hotel name and price per night must be provided.', 16, 1);

RETURN;

END;

IF @hotPricePerNight <= 0

BEGIN

RAISERROR('Price per night must be greater than zero.', 16, 1);

RETURN

END;

IF @hotStars IS NOT NULL AND @hotStars<= 0 AND @hotStars > 5

BEGIN

RAISERROR('Hotel stars must be greater than zero and less than or equal to five.', 16, 1);

RETURN;

END;

IF @cityId IS NULL

BEGIN

RAISERROR('CityId must be provided.', 16, 1);

RETURN;;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @cityId)

BEGIN

RAISERROR('The specified city ID does not exist.', 16, 1);

RETURN;

END;

IF @addrStreet IS NULL OR @addrHouseNo IS NULL

BEGIN

RAISERROR('Address details are incomplete. Please provide street, and house number.', 16, 1);

RETURN;

END;

DECLARE @addrId INT;

INSERT INTO tbl\_address(addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (@cityId, @addrPostalCode, @addrRegion, @addrStreet, @addrHouseNo);

SET @addrId = SCOPE\_IDENTITY();

INSERT INTO tbl\_hotel(hotAddrId, hotName, hotPricePerNight, hotTelephoneNo, hotContactEmail, hotStars)

VALUES (@addrId, @hotName, @hotPricePerNight, @hotTelephoneNo, @hotContactEmail, @hotStars);

PRINT 'The hotel and its address have been successfully added.';

END TRY

BEGIN CATCH

SELECT ERROR\_MESSAGE() AS ErrorMessage;

END CATCH;

END;

GO

#### up\_add\_guide

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_add\_guide

--------------------------------------------------------------------------------------

-- This procedure is used to add a new guide to the database.

--

-- Input Parameters:

-- @guideName: The name of the guide.

-- @guideSurname: The surname of the guide.

-- @guideCityId: The city ID where the guide operates.

-- @guidePhone: The phone number of the guide.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_add\_guide @guideName = 'John', @guideSurname = 'Doe', @guideCityId = 123, @guidePhone = '123456789'

--

-- Result of the action:

-- The guide has been successfully added.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_add\_guide

@guideName VARCHAR(50),

@guideSurname VARCHAR(50),

@guideCityId INT,

@guidePhone VARCHAR(20)

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRY

IF @guideName IS NULL OR @guideSurname IS NULL OR @guidePhone IS NULL

BEGIN

RAISERROR('Name, surname, and phone must be provided.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @guideCityId)

BEGIN

RAISERROR('The specified city ID does not exist.', 16, 1);

RETURN;

END;

INSERT INTO tbl\_guide(guideName, guideSurname, guideCityId, guidePhone)

VALUES (@guideName, @guideSurname, @guideCityId, @guidePhone);

PRINT 'The guide has been successfully added.';

END TRY

BEGIN CATCH

DECLARE @ErrorMessage NVARCHAR(4000);

SET @ErrorMessage = ERROR\_MESSAGE();

RAISERROR(@ErrorMessage, 16, 1);

END CATCH;

END;

#### up\_add\_flight

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_add\_flight

--------------------------------------------------------------------------------------

-- This procedure is used to add a new flight to the database.

--

-- Input Parameters:

-- @fliStartCityId: The ID of the city where the flight starts.

-- @fliEndCityId: The ID of the city where the flight ends.

-- @fliStartTime: The start time of the flight.

-- @fliEndTime: The end time of the flight.

-- @fliClass: The class of the flight (First, Business, or Economy).

-- @fliPrice: The price of the flight.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_add\_flight @fliStartCityId = 1, @fliEndCityId = 2, @fliStartTime = '2024-03-25 08:00:00', @fliEndTime = '2024-03-25 10:00:00', @fliClass = 'Economy', @fliPrice = 500.00

--

-- Result of the action:

-- The flight was successfully added.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_add\_flight

@fliStartCityId INT,

@fliEndCityId INT,

@fliStartTime DATETIME,

@fliEndTime DATETIME,

@fliClass VARCHAR(10),

@fliPrice FLOAT

AS

BEGIN

SET NOCOUNT ON;

IF @fliStartTime IS NULL OR @fliEndTime IS NULL OR @fliClass = '' OR @fliPrice IS NULL

BEGIN

RAISERROR('One or more required fields are empty.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @fliStartCityId)

BEGIN

RAISERROR('Start city with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @fliEndCityId)

BEGIN

RAISERROR('End city with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @fliClass NOT IN ('First', 'Business', 'Economy')

BEGIN

RAISERROR('Invalid flight class. Class must be First, Business, or Economy.', 16, 1);

RETURN;

END;

IF @fliPrice <= 0

BEGIN

RAISERROR('Flight price must be greater than zero.', 16, 1);

RETURN;

END;

DECLARE @timeDifference INT;

SET @timeDifference = DATEDIFF(HOUR, @fliStartTime, @fliEndTime);

IF @timeDifference > 19 OR @timeDifference < 1

BEGIN

RAISERROR('Time difference between start and end cannot exceed 19 hours and be less than 1 hour.', 16, 1);

RETURN;

END;

INSERT INTO tbl\_flight(fliStartCityId, fliEndCityId, fliStartTime, fliEndTime, fliClass, fliPrice)

VALUES (@fliStartCityId, @fliEndCityId, @fliStartTime, @fliEndTime, @fliClass, @fliPrice);

PRINT 'The flight was successfully added.';

END;

GO

#### up\_add\_driver

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_add\_driver

--------------------------------------------------------------------------------------

-- This procedure is used to add a new driver to the database.

--

-- Input Parameters:

-- @driverName: The name of the driver.

-- @driverSurname: The surname of the driver.

-- @driverPhone: The phone number of the driver.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_add\_driver @driverName = 'John', @driverSurname = 'Doe', @driverPhone = '123456789'

--

-- Result of the action:

-- The driver has been successfully added.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_add\_driver

@driverName VARCHAR(50),

@driverSurname VARCHAR(50),

@driverPhone VARCHAR(20)

AS

BEGIN

SET NOCOUNT ON;

IF @driverName IS NULL OR @driverSurname IS NULL OR @driverPhone IS NULL

BEGIN

RAISERROR('Name, surname, and phone must be provided.', 16, 1);

RETURN;

END;

INSERT INTO tbl\_driver(driverName, driverSurName, driverPhone)

VALUES (@driverName, @driverSurname, @driverPhone);

PRINT 'The driver has been successfully added.';

END;

GO

#### up\_add\_customer\_and\_address

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_add\_customer\_and\_address

--------------------------------------------------------------------------------------

-- This procedure adds a new customer along with their address to the database.

--

-- Input Parameters:

-- @custName: VARCHAR(50) - The first name of the customer.

-- @custSurname: VARCHAR(50) - The last name of the customer.

-- @custPhone: VARCHAR(20) - The phone number of the customer.

-- @custEmailAddress: VARCHAR(255) - The email address of the customer.

-- @cityId: INT - The ID of the city where the customer resides.

-- @addrPostalCode: VARCHAR(20) - The postal code of the customer's address.

-- @addrRegion: VARCHAR(100) - The region of the customer's address.

-- @addrStreet: VARCHAR(255) - The street name of the customer's address.

-- @addrHouseNo: VARCHAR(10) - The house number of the customer's address.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_add\_customer\_and\_address 'John', 'Doe', '123456789', 'john@example.com', 1, '12345', 'Region', 'Main Street', '10'

--

-- Result of the action

-- The customer and his address have been successfully added.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_add\_customer\_and\_address

@custName VARCHAR(50),

@custSurname VARCHAR(50),

@custPhone VARCHAR(20),

@custEmailAddress VARCHAR(255),

@cityId INT = NULL,

@addrPostalCode VARCHAR(20),

@addrRegion VARCHAR(100),

@addrStreet VARCHAR(255),

@addrHouseNo VARCHAR(10)

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRY

IF @custName = '' OR @custSurname = '' OR @custPhone = ''

BEGIN

RAISERROR('Name, surname, and phone must be provided.', 16, 1);

RETURN;

END;

IF @cityId IS NULL

BEGIN

RAISERROR('CityId must be provided.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @cityId)

BEGIN

RAISERROR('The specified city ID does not exist.', 16, 1);

RETURN;

END;

IF @custEmailAddress IS NOT NULL AND

@custEmailAddress NOT LIKE '%@%.%' AND

PATINDEX('%[^a-zA-Z0-9.@]%', @custEmailAddress) <> 0

BEGIN

RAISERROR('Invalid email address format.', 16, 1);

RETURN;

END;

IF @addrStreet IS NULL OR @addrHouseNo IS NULL

BEGIN

RAISERROR('Address details are incomplete. Please provide street, and house number.', 16, 1);

RETURN;

END;

INSERT INTO tbl\_address(addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (@cityId, @addrPostalCode, @addrRegion, @addrStreet, @addrHouseNo);

DECLARE @addrId INT = SCOPE\_IDENTITY();

INSERT INTO tbl\_customer(custName, custSurname, custPhone, custEmailAddress, custAddrId)

VALUES (@custName, @custSurname, @custPhone, @custEmailAddress, @addrId);

PRINT 'The customer and his address have been successfully added.';

END TRY

BEGIN CATCH

SELECT ERROR\_MESSAGE() AS ErrorMessage;

END CATCH;

END

#### up\_add\_busRoute

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_add\_busRoute

--------------------------------------------------------------------------------------

-- This procedure is used to add a new bus route to the database.

--

-- Input Parameters:

-- @driverId: The ID of the driver for the bus route (required).

-- @busRouteStartCity: The ID of the start city for the bus route (required).

-- @busRouteEndCity: The ID of the end city for the bus route (required).

-- @busRouteStartTime: The start time of the bus route (required).

-- @busRouteEndTime: The end time of the bus route (required).

-- @busRouteBusID: The ID of the bus for the route (required).

-- @busRoutePrice: The price of the bus route (required).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_add\_busRoute @driverId = 1, @busRouteStartCity = 2, @busRouteEndCity = 3, @busRouteStartTime = '2024-03-25 08:00:00', @busRouteEndTime = '2024-03-25 18:00:00', @busRouteBusID = 4, @busRoutePrice = 50.00

--

-- Result of the action:

-- The bus route has been successfully added.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_add\_busRoute

@driverId INT,

@busRouteStartCity INT,

@busRouteEndCity INT,

@busRouteStartTime DATETIME,

@busRouteEndTime DATETIME,

@busRouteBusID INT,

@busRoutePrice FLOAT

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_driver WHERE driverId = @driverId)

BEGIN

RAISERROR('The driver with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @busRouteStartCity)

BEGIN

RAISERROR('The start city with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @busRouteEndCity)

BEGIN

RAISERROR('The end city with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_bus WHERE busId = @busRouteBusID)

BEGIN

RAISERROR('The bus with the specified ID does not exist.', 16, 1);

RETURN;

END;

IF @busRoutePrice <= 0

BEGIN

RAISERROR('The route price must be greater than zero.', 16, 1);

RETURN;

END;

IF DATEDIFF(DAY, @busRouteStartTime, @busRouteEndTime) > 3

BEGIN

RAISERROR('The difference between start time and end time cannot exceed 3 days.', 16, 1);

RETURN;

END;

IF @busRouteStartTime IS NULL OR @busRouteEndTime IS NULL

BEGIN

RAISERROR('All parameters must be provided.', 16, 1);

RETURN;

END;

INSERT INTO tbl\_busroute(driverId, busRouteStartCityId, busRouteEndCityId, busRouteStartTime, busRouteEndTime, busRouteBusID, busRoutePrice)

VALUES (@driverId, @busRouteStartCity, @busRouteEndCity, @busRouteStartTime, @busRouteEndTime, @busRouteBusID, @busRoutePrice);

PRINT 'The bus route has been successfully added.';

END;

GO

#### up\_add\_bus

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_add\_bus

--------------------------------------------------------------------------------------

-- This procedure is used to add a new bus to the database.

--

-- Input Parameters:

-- @busNo: The number of the bus to be added (required).

-- @busModel: (Optional) The model of the bus.

-- @busColor: (Optional) The color of the bus.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_add\_bus @busNo = '1234', @busModel = 'Mercedes', @busColor = 'Blue'

--

-- Result of the action:

-- The bus was successfully added.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_add\_bus

@busNo VARCHAR(20),

@busModel VARCHAR(50),

@busColor VARCHAR(20)

AS

BEGIN

SET NOCOUNT ON;

IF @busNo IS NULL

BEGIN

RAISERROR('Bus number cannot be null.', 16, 1);

RETURN;

END;

IF EXISTS (SELECT 1 FROM tbl\_bus WHERE busNo = @busNo)

BEGIN

RAISERROR('Bus with the specified number already exists.', 16, 1);

RETURN;

END;

INSERT INTO tbl\_bus (busNo, busModel, busColor)

VALUES (@busNo, @busModel, @busColor);

PRINT 'The bus was successfully added.'

END;

GO

#### up\_add\_booking

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_add\_booking

--------------------------------------------------------------------------------------

-- This procedure is used to add a booking for a customer and a package in the database.

--

-- Input Parameters:

-- @bookCustId: The ID of the customer making the booking (required).

-- @bookPackageId: The ID of the package being booked (required).

-- @bookDiscountPercent: The discount percentage for the booking (required).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_add\_booking @bookCustId = 1, @bookPackageId = 2, @bookDiscountPercent = 0.2;

--

-- Result of the action:

-- The booking has been successfully added.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_add\_booking

@bookCustId INT,

@bookPackageId INT,

@bookDiscountPercent FLOAT

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM tbl\_customer WHERE custId = @bookCustId)

BEGIN

RAISERROR('The specified customer ID does not exist.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_package WHERE packId = @bookPackageId)

BEGIN

RAISERROR('The specified package ID does not exist.', 16, 1);

RETURN;

END;

IF @bookDiscountPercent < 0 OR @bookDiscountPercent > 1

BEGIN

RAISERROR('The discount percentage must be between 0 and 1.', 16, 1);

RETURN;

END;

INSERT INTO tbl\_booking(bookCustId, bookPackageId, bookDiscountPercent,bookPrice)

VALUES (@bookCustId, @bookPackageId, @bookDiscountPercent,1);

PRINT 'The booking has been successfully added.';

END;

GO

#### up\_add\_address

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_add\_address

--------------------------------------------------------------------------------------

-- This procedure is used to add a new address to the database.

--

-- Input Parameters:

-- @addrCityId: The ID of the city associated with the address.

-- @addrPostalCode: The postal code of the address.

-- @addrRegion: The region of the address.

-- @addrStreet: The street of the address.

-- @addrHouseNo: The house number of the address.

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_add\_address @addrCityId = 123, @addrPostalCode = '12345', @addrRegion = 'Region', @addrStreet = 'Street', @addrHouseNo = '10'

--

-- Result of the action:

-- The address has been successfully added.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_add\_address

@addrCityId INT,

@addrPostalCode VARCHAR(20),

@addrRegion VARCHAR(100),

@addrStreet VARCHAR(255),

@addrHouseNo VARCHAR(10)

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRY

IF @addrCityId IS NULL

BEGIN

RAISERROR('CityId must be provided.', 16, 1);

RETURN;

END;

IF NOT EXISTS (SELECT 1 FROM tbl\_city WHERE cityId = @addrCityId)

BEGIN

RAISERROR('The specified city ID does not exist.', 16, 1);

RETURN;

END;

IF @addrStreet IS NULL OR @addrHouseNo IS NULL

BEGIN

RAISERROR('Address details are incomplete. Please provide street and house number.', 16, 1);

RETURN;

END;

INSERT INTO tbl\_address(addrCityId, addrPostalCode, addrRegion, addrStreet, addrHouseNo)

VALUES (@addrCityId, @addrPostalCode, @addrRegion, @addrStreet, @addrHouseNo);

PRINT 'The address has been successfully added.';

END TRY

BEGIN CATCH

SELECT ERROR\_MESSAGE() AS ErrorMessage;

END CATCH;

END;

## GET

#### up\_search\_travel\_packages

----------------------------------------------------------------------------------------- PROCEDURE DEFINITION

--- up\_search\_travel\_packages

--------------------------------------------------------------------------------------

-- This procedure searches for travel packages based on various criteria such as destination city,

-- start and end dates, maximum price, and preferred mode of transport. It returns packages that match

-- the specified criteria.

--

-- Input Parameters:

-- @destinationCityId: The ID of the destination city for the travel packages (optional).

-- @startDate: The start date for the travel packages (optional).

-- @endDate: The end date for the travel packages (optional).

-- @maxPrice: The maximum price for the travel packages (optional).

-- @preferredTransport: The preferred mode of transport for the travel packages ('flight', 'bus', or NULL for any) (optional).

--

-- Output Parameters:

-- @packId: The ID of the travel package.

-- @packTitle: The title of the travel package.

-- @packDescription: The description of the travel package.

-- @packPrice: The price of the travel package.

-- @packStartDate: The start date of the travel package.

-- @packEndDate: The end date of the travel package.

-- @destinationCity: The destination city of the travel package.

-- @hotelName: The name of the hotel associated with the travel package.

-- @hotelStars: The star rating of the hotel associated with the travel package.

--

-- Example Usage:

-- EXEC up\_search\_travel\_packages @destinationCityId = 123, @startDate = '2024-04-01', @endDate = '2024-04-10', @maxPrice = 2000.00, @preferredTransport = 'flight';

--

-- Result of the action:

-- Returns travel packages that match the specified criteria.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_search\_travel\_packages

@destinationCityId INT = NULL,

@startDate DATE = NULL,

@endDate DATE = NULL,

@maxPrice DECIMAL(10, 2) = NULL,

@preferredTransport VARCHAR(10) = NULL

AS

BEGIN

SET NOCOUNT ON;

SELECT

packId,

packTitle,

packDescription,

packPrice,

packStartDate,

packEndDate,

(SELECT cityName FROM tbl\_city WHERE cityId = packCityId) AS destinationCity,

(SELECT hotName FROM tbl\_hotel WHERE hotId = packHotId) AS hotelName,

(SELECT hotStars FROM tbl\_hotel WHERE hotId = packHotId) AS hotelStars

FROM

tbl\_package

WHERE

(packCityId = @destinationCityId OR @destinationCityId IS NULL)

AND (packStartDate >= @startDate OR @startDate IS NULL)

AND (packEndDate <= @endDate OR @endDate IS NULL)

AND (packPrice <= @maxPrice OR @maxPrice IS NULL)

AND

(

(@preferredTransport = 'flight' AND packOutboundFlightId IS NOT NULL AND packReturnFlightId IS NOT NULL) OR

(@preferredTransport = 'bus' AND packOutboundBusRouteId IS NOT NULL AND packReturnBusRouteId IS NOT NULL) OR

(@preferredTransport IS NULL OR @preferredTransport NOT IN ('flight', 'bus'))

);

END;

#### up\_get\_hotel\_bookings

----------------------------------------------------------------------------------------- PROCEDURE DEFINITION

--- up\_get\_hotel\_bookings

--------------------------------------------------------------------------------------

-- This procedure retrieves bookings made for a specific hotel identified by its hotel ID.

--

-- Input Parameters:

-- @hotelId: The ID of the hotel for which bookings are to be retrieved.

--

-- Output Columns:

-- @hotId: The ID of the hotel.

-- @hotName: The name of the hotel.

-- @bookId: The ID of the booking.

-- @packTitle: The title of the travel package booked.

-- @packStartDate: The start date of the travel package booked.

-- @packEndDate: The end date of the travel package booked.

-- @bookPrice: The price of the booking.

-- @bookDiscountAmnt: The discount amount applied to the booking.

-- @hotPricePerNight: The price per night of the hotel.

-- @hotelTotalPrice: The total price of the hotel stay for the duration of the package.

--

-- Example Usage:

-- EXEC up\_get\_hotel\_bookings 189;

--

-- Result of the action:

-- Returns the bookings made for the specified hotel.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_get\_hotel\_bookings

@hotelId INT

AS

BEGIN

SET NOCOUNT ON;

SELECT

h.hotId,

h.hotName,

b.bookId,

p.packTitle,

p.packStartDate,

p.packEndDate,

b.bookPrice,

b.bookDiscountAmnt,

h.hotPricePerNight,

h.hotPricePerNight \* p.packDuration AS hotelTotalPrice

FROM

tbl\_booking b

INNER JOIN

tbl\_package p ON b.bookPackageId = p.packId

INNER JOIN

tbl\_hotel h ON p.packHotId = h.hotId

WHERE

h.hotId = @hotelId;

END;

#### up\_get\_customer\_bookings

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_get\_customer\_bookings

--------------------------------------------------------------------------------------

-- This procedure retrieves bookings made by a specific customer identified by their customer ID.

--

-- Input Parameters:

-- @customerId: The ID of the customer whose bookings are to be retrieved.

--

-- Output Columns:

-- @bookId: The ID of the booking.

-- @packTitle: The title of the travel package booked.

-- @packStartDate: The start date of the travel package booked.

-- @packEndDate: The end date of the travel package booked.

-- @bookPrice: The price of the booking.

-- @bookDiscountAmnt: The discount amount applied to the booking.

--

-- Example Usage:

-- EXEC up\_get\_customer\_bookings @customerId=56;

--

-- Result of the action:

-- Returns the bookings made by the specified customer.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_get\_customer\_bookings

@customerId INT

AS

BEGIN

SET NOCOUNT ON;

SELECT

b.bookId,

p.packTitle,

p.packStartDate,

p.packEndDate,

b.bookPrice,

b.bookDiscountAmnt

FROM

tbl\_booking b

INNER JOIN

tbl\_package p ON b.bookPackageId = p.packId

WHERE

b.bookCustId = @customerId;

END;

#### up\_get\_countries\_by\_continent\_with\_packages

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_get\_countries\_by\_continent\_with\_packages

--------------------------------------------------------------------------------------

-- This procedure retrieves distinct countries from a specified continent that have packages

-- associated with them. It returns the country IDs and names based on the provided continent name.

--

-- Input Parameters:

-- @continentName: The name of the continent for which countries with associated packages are

-- retrieved (required).

--

-- Output Parameters:

-- @CountryId: The ID of the country with packages associated with it.

-- @CountryName: The name of the country with packages associated with it.

--

-- Example Usage:

-- EXEC up\_get\_countries\_by\_continent\_with\_packages @continentName = 'Europe';

--

-- Result of the action:

-- Returns the distinct countries from the specified continent that have packages associated with

-- them.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_get\_countries\_by\_continent\_with\_packages

@continentName VARCHAR(25)

AS

BEGIN

SET NOCOUNT ON;

SELECT DISTINCT

c.ctryId,

c.ctryName

FROM

tbl\_country c

INNER JOIN

tbl\_city ci ON c.ctryId = ci.ctryId

INNER JOIN

tbl\_continent co ON c.contId = co.contId

INNER JOIN

tbl\_package p ON p.packCityId = ci.cityId

WHERE

co.contName = @continentName;

END;

#### up\_get\_cities\_by\_country\_with\_packages

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_get\_cities\_by\_country\_with\_packages

--------------------------------------------------------------------------------------

-- This procedure retrieves distinct cities from a specified country that have packages

-- associated with them. It returns the names of the cities based on the provided country name.

--

-- Input Parameters:

-- @countryName: The name of the country for which cities with associated packages are

-- retrieved (required).

--

-- Output Parameters:

-- @CityName: The name of the city with packages associated with it.

--

-- Example Usage:

-- EXEC up\_get\_cities\_by\_country\_with\_packages @countryName = 'France';

--

-- Result of the action:

-- Returns the distinct cities from the specified country that have packages associated with them.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_get\_cities\_by\_country\_with\_packages

@countryName VARCHAR(45)

AS

BEGIN

SET NOCOUNT ON;

SELECT DISTINCT

ci.cityName

FROM

tbl\_city ci

INNER JOIN

tbl\_country co ON ci.ctryId = co.ctryId

INNER JOIN

tbl\_package p ON p.packCityId = ci.cityId

WHERE

co.ctryName = @countryName;

END;

## CALCULATE FOR TRIGGERS

#### up\_calculate\_package\_details

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_calculate\_package\_details

--------------------------------------------------------------------------------------

-- This procedure calculates and updates the details of a package.

-- It calculates the start date, end date, duration, and price of the package and

-- updates the corresponding fields in the tbl\_package table.

--

-- This procedure is triggered when inserting or modifying a record in the tbl\_package table.

--

--

-- Input Parameters:

-- @packId: The ID of the package for which details need to be calculated and updated (required).

-- @packHotId: The ID of the hotel associated with the package (required).

-- @packOutboundFlightId: The ID of the outbound flight associated with the package (optional).

-- @packReturnFlightId: The ID of the return flight associated with the package (optional).

-- @packOutboundBusRouteId: The ID of the outbound bus route associated with the package (optional).

-- @packReturnBusRouteId: The ID of the return bus route associated with the package (optional).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_calculate\_package\_details @packId = 1, @packHotId = 101, @packOutboundFlightId = 201,

-- @packReturnFlightId = 202, @packOutboundBusRouteId = 301, @packReturnBusRouteId = 302;

--

-- Result of the action:

-- The details of the specified package have been successfully calculated and updated.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_calculate\_package\_details

@packId INT,

@packHotId INT,

@packOutboundFlightId INT,

@packReturnFlightId INT,

@packOutboundBusRouteId INT,

@packReturnBusRouteId INT

AS

BEGIN

DECLARE @packStartDate DATE

DECLARE @packEndDate DATE

DECLARE @packDuration TINYINT

DECLARE @packPrice FLOAT

IF @packOutboundFlightId IS NOT NULL

SELECT @packStartDate = fliEndTime FROM tbl\_flight WHERE fliId = @packOutboundFlightId

ELSE IF @packOutboundBusRouteId IS NOT NULL

SELECT @packStartDate = busRouteEndTime FROM tbl\_busroute WHERE busRouteId = @packOutboundBusRouteId

IF @packReturnFlightId IS NOT NULL

SELECT @packEndDate = fliStartTime FROM tbl\_flight WHERE fliId = @packReturnFlightId

ELSE IF @packReturnBusRouteId IS NOT NULL

SELECT @packEndDate = busRouteStartTime FROM tbl\_busroute WHERE busRouteId = @packReturnBusRouteId

SELECT @packDuration = DATEDIFF(DAY, @packStartDate, @packEndDate) + 1

SELECT @packPrice =

(

(SELECT hotPricePerNight FROM tbl\_hotel WHERE hotId = @packHotId) \* @packDuration +

ISNULL((SELECT fliPrice FROM tbl\_flight WHERE fliId = @packOutboundFlightId), 0) +

ISNULL((SELECT fliPrice FROM tbl\_flight WHERE fliId = @packReturnFlightId), 0) +

ISNULL((SELECT busRoutePrice FROM tbl\_busroute WHERE busRouteId = @packOutboundBusRouteId), 0) +

ISNULL((SELECT busRoutePrice FROM tbl\_busroute WHERE busRouteId = @packReturnBusRouteId), 0)

) \* 1.15

UPDATE tbl\_package

SET packStartDate = @packStartDate,

packEndDate = @packEndDate,

packDuration = @packDuration,

packPrice = @packPrice

WHERE packId = @packId

END

#### up\_calculate\_booking\_prices

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_calculate\_booking\_prices

--------------------------------------------------------------------------------------

-- This procedure calculates and updates the prices for bookings associated with a specified package

--

-- This procedure is triggered when inserting or modifying a record in the tbl\_booking table.

--

-- Input Parameters:

-- @bookPackageId: The ID of the package for which booking prices need to be calculated and updated (required).

-- @bookDiscountPercent: The discount percentage applied to the package price (required).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_calculate\_booking\_prices @bookPackageId = 1, @bookDiscountPercent = 0.1;

--

-- Result of the action:

-- The prices for bookings associated with the specified package have been successfully calculated

-- and updated.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

ALTER PROCEDURE up\_calculate\_booking\_prices

@bookPackageId INT,

@bookDiscountPercent DECIMAL(10, 2)

AS

BEGIN

DECLARE @packPrice DECIMAL(10, 2)

DECLARE @bookDiscountAmnt DECIMAL(10, 2)

DECLARE @bookPrice DECIMAL(10, 2)

SELECT @packPrice = packPrice

FROM tbl\_package

WHERE packId = @bookPackageId

SET @bookDiscountAmnt = @packPrice \* @bookDiscountPercent

SET @bookPrice = @packPrice - @bookDiscountAmnt

UPDATE tbl\_booking

SET bookDiscountAmnt = @bookDiscountAmnt,

bookPrice = @bookPrice

WHERE bookPackageId = @bookPackageId

END

## MASS DATA MODIFICATION

#### up\_delay\_busRoute

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_delay\_busRoute

--------------------------------------------------------------------------------------

-- This procedure is used to delay the start and end times of bus routes in the database.

-- The procedure updates the start and end times of bus routes based on the provided parameters.

--

-- Input Parameters:

-- @startCityId: The ID of the start city of bus routes to be delayed (optional).

-- @endCityId: The ID of the end city of bus routes to be delayed (optional).

-- @delayInHours: The number of hours by which the bus routes should be delayed (required).

-- @startDateFilter: The start date filter for bus routes to be delayed (optional).

-- @endDateFilter: The end date filter for bus routes to be delayed (optional).

-- @startTimeOfDayFilter: The start time of day filter for bus routes to be delayed (optional).

-- @endTimeOfDayFilter: The end time of day filter for bus routes to be delayed (optional).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delay\_busRoute @delayInHours = 2;

--

-- Result of the action:

-- The start and end times of bus routes have been successfully delayed.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delay\_busRoute

@startCityId INT = NULL,

@endCityId INT = NULL,

@delayInHours INT,

@startDateFilter DATETIME = NULL,

@endDateFilter DATETIME = NULL,

@startTimeOfDayFilter TIME = NULL,

@endTimeOfDayFilter TIME = NULL

AS

BEGIN

UPDATE tbl\_busroute

SET busRouteStartTime = DATEADD(HOUR, @delayInHours, busRouteStartTime),

busRouteEndTime = DATEADD(HOUR, @delayInHours, busRouteEndTime)

WHERE (@startCityId IS NULL OR busRouteStartCityId = @startCityId)

AND (@endCityId IS NULL OR busRouteEndCityId = @endCityId)

AND (@startDateFilter IS NULL OR CONVERT(DATE, busRouteStartTime) >= @startDateFilter)

AND (@endDateFilter IS NULL OR CONVERT(DATE, busRouteEndTime) <= @endDateFilter)

AND (@startTimeOfDayFilter IS NULL OR CAST(busRouteStartTime AS TIME) >= @startTimeOfDayFilter)

AND (@endTimeOfDayFilter IS NULL OR CAST(busRouteEndTime AS TIME) <= @endTimeOfDayFilter);

END;

#### up\_delay\_flight

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_delay\_flight

--------------------------------------------------------------------------------------

-- This procedure is used to delay the start and end times of flights in the database.

-- The procedure updates the start and end times of flights based on the provided parameters.

--

-- Input Parameters:

-- @startCityId: The ID of the start city of flights to be delayed (optional).

-- @endCityId: The ID of the end city of flights to be delayed (optional).

-- @delayInHours: The number of hours by which the flights should be delayed (required).

-- @startDateFilter: The start date filter for flights to be delayed (optional).

-- @endDateFilter: The end date filter for flights to be delayed (optional).

-- @startTimeOfDayFilter: The start time of day filter for flights to be delayed (optional).

-- @endTimeOfDayFilter: The end time of day filter for flights to be delayed (optional).

--

-- Output Parameters:

-- None

--

-- Example Usage:

-- EXEC up\_delay\_flight @delayInHours = 2;

--

-- Result of the action:

-- The start and end times of flights have been successfully delayed.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_delay\_flight

@startCityId INT = NULL,

@endCityId INT = NULL,

@delayInHours INT,

@startDateFilter DATETIME = NULL,

@endDateFilter DATETIME = NULL,

@startTimeOfDayFilter TIME = NULL,

@endTimeOfDayFilter TIME = NULL

AS

BEGIN

UPDATE tbl\_flight

SET fliStartTime = DATEADD(HOUR, @delayInHours, fliStartTime),

fliEndTime = DATEADD(HOUR, @delayInHours, fliEndTime)

WHERE (@startCityId IS NULL OR fliStartCityId = @startCityId)

AND (@endCityId IS NULL OR fliEndCityId = @endCityId)

AND (@startDateFilter IS NULL OR CONVERT(DATE, fliStartTime) >= @startDateFilter)

AND (@endDateFilter IS NULL OR CONVERT(DATE, fliEndTime) <= @endDateFilter)

AND (@startTimeOfDayFilter IS NULL OR CAST(fliStartTime AS TIME) >= @startTimeOfDayFilter)

AND (@endTimeOfDayFilter IS NULL OR CAST(fliEndTime AS TIME) <= @endTimeOfDayFilter);

END;

## REPORTS

#### up\_generate\_revenue\_report

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_generate\_revenue\_report

--------------------------------------------------------------------------------------

-- This procedure generates a revenue report for a specified month and year. It calculates the total

-- revenue generated in the given month and year based on the bookings made during that period.

--

-- Input Parameters:

-- @month: The month for which the revenue report is generated.

-- @year: The year for which the revenue report is generated.

--

-- Output Parameters:

-- @BookedMonth: The month of the bookings.

-- @BookedYear: The year of the bookings.

-- @TotalRevenue: The total revenue generated in the specified month and year.

--

-- Example Usage:

-- EXEC up\_generate\_revenue\_report @month = 3, @year = 2024;

--

-- Result of the action:

-- Returns a revenue report for the specified month and year.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_generate\_revenue\_report

@month INT,

@year INT

AS

BEGIN

SET NOCOUNT ON;

SELECT

MONTH(b.bookCreatedAt) AS BookedMonth,

YEAR(b.bookCreatedAt) AS BookedYear,

CAST(SUM((((b.bookPrice + b.bookDiscountAmnt) / 115) \* 15) - b.bookDiscountAmnt) AS DECIMAL(10, 2)) AS TotalRevenue

FROM

tbl\_booking b

WHERE

MONTH(b.bookCreatedAt) = @month AND YEAR(b.bookCreatedAt) = @year

GROUP BY

MONTH(b.bookCreatedAt), YEAR(b.bookCreatedAt);

END;



21 Sample Output of the up\_generate\_revenue\_report procedure

#### up\_generate\_popularity\_report\_by\_date\_range

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_generate\_popularity\_report\_by\_date\_range

--------------------------------------------------------------------------------------

-- This procedure generates a popularity report for travel packages within a specified date range.

-- It calculates the total number of bookings made for each package during the specified period

-- and lists them in descending order of popularity.

--

-- Input Parameters:

-- @startDate: The start date of the date range.

-- @endDate: The end date of the date range.

--

-- Output Parameters:

-- @packId: The ID of the travel package.

-- @packTitle: The title of the travel package.

-- @TotalBookings: The total number of bookings made for the travel package within the date range.

--

-- Example Usage:

-- EXEC up\_generate\_popularity\_report\_by\_date\_range '2024-01-01', '2024-12-31';

--

-- Result of the action:

-- Returns a popularity report for travel packages within the specified date range.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_generate\_popularity\_report\_by\_date\_range

@startDate DATE,

@endDate DATE

AS

BEGIN

SET NOCOUNT ON;

SELECT

p.packId,

p.packTitle,

COUNT(b.bookId) AS TotalBookings

FROM

tbl\_package p

LEFT JOIN

tbl\_booking b ON p.packId = b.bookPackageId

WHERE

b.bookCreatedAt BETWEEN @startDate AND @endDate

GROUP BY

p.packId, p.packTitle

ORDER BY

TotalBookings DESC;

END;

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

22Sample Output of the up\_generate\_popularity\_report\_by\_date\_range procedure

#### up\_generate\_popularity\_report

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_generate\_popularity\_report

--------------------------------------------------------------------------------------

-- This procedure generates a popularity report for travel packages. It calculates the total number

-- of bookings made for each package and lists them in descending order of popularity.

--

-- Output Parameters:

-- @packId: The ID of the travel package.

-- @packTitle: The title of the travel package.

-- @TotalBookings: The total number of bookings made for the travel package.

--

-- Example Usage:

-- EXEC up\_generate\_popularity\_report;

--

-- Result of the action:

-- Returns a popularity report listing travel packages based on the total number of bookings made.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_generate\_popularity\_report

AS

BEGIN

SET NOCOUNT ON;

SELECT

p.packId,

p.packTitle,

COUNT(b.bookId) AS TotalBookings

FROM

tbl\_package p

LEFT JOIN

tbl\_booking b ON p.packId = b.bookPackageId

GROUP BY

p.packId, p.packTitle

ORDER BY

TotalBookings DESC;

END;

Obraz zawierający tekst, numer, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

23 Sample Output of the up\_generate\_popularity\_report procedure

#### up\_generate\_booking\_report

--------------------------------------------------------------------------------------

--- PROCEDURE DEFINITION

--- up\_generate\_booking\_report

--------------------------------------------------------------------------------------

-- This procedure generates a booking report for a specified time range. It retrieves booking details

-- including customer information, package details, and pricing for bookings created within the given

-- start and end dates.

--

-- Input Parameters:

-- @startDate: The start date for the booking report.

-- @endDate: The end date for the booking report.

--

-- Output Parameters:

-- @bookId: The ID of the booking.

-- @bookCustId: The ID of the customer who made the booking.

-- @custName: The name of the customer who made the booking.

-- @custSurname: The surname of the customer who made the booking.

-- @bookPackageId: The ID of the package booked.

-- @packTitle: The title of the package booked.

-- @bookPrice: The price of the booking.

-- @bookDiscountAmnt: The amount of discount applied to the booking.

-- @destinationCity: The destination city of the booked package.

-- @bookDiscountPercent: The percentage of discount applied to the booking.

-- @totalPrice: The total price after applying the discount.

--

-- Example Usage:

-- EXEC up\_generate\_booking\_report @startDate = '2024-01-01', @endDate = '2024-12-31';

--

-- Result of the action:

-- Returns a booking report for bookings created within the specified time range.

--------------------------------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER PROCEDURE up\_generate\_booking\_report

@startDate DATE,

@endDate DATE

AS

BEGIN

SET NOCOUNT ON;

SELECT

b.bookId,

b.bookCustId,

c.custName,

c.custSurname,

b.bookPackageId,

p.packTitle,

(SELECT cityName FROM tbl\_city WHERE cityId = p.packCityId) AS destinationCity,

b.bookPrice + b.bookDiscountAmnt AS packagePrice,

b.bookDiscountPercent,

b.bookDiscountAmnt,

b.bookPrice

FROM

tbl\_booking b

INNER JOIN

tbl\_customer c ON b.bookCustId = c.custId

INNER JOIN

tbl\_package p ON b.bookPackageId = p.packId

WHERE

b.bookCreatedAt BETWEEN @startDate AND @endDate;

END;

Obraz zawierający tekst, Czcionka, linia, zrzut ekranu

Opis wygenerowany automatycznie

24 Sample Output of the up\_generate\_booking\_report procedure

# Wykaz utworzonych funkcji użytkownika

-- This SQL query returns a list of user functions (and their details) in the database

SELECT

ROUTINE\_NAME,

ROUTINE\_TYPE,

DATA\_TYPE,

CREATED,

LAST\_ALTERED

FROM INFORMATION\_SCHEMA.ROUTINES

WHERE ROUTINE\_TYPE = 'FUNCTION';

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

25 List of created user functions

# Skrypt tworzenia funkcji definiowanych przez użytkownika

#### uf\_return\_total\_income\_by\_year

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_total\_income\_by\_year

---------------------------------------------------------------

-- This function calculates the total income generated in a specific year by summing up the net income (total booking price after deducting discounts and taxes) of all bookings made in that year.

--

-- Input Parameters:

-- - @Year: The year for which the total income is to be calculated.

--

-- Output Parameters:

-- - @TotalIncome : The total income generated in the specified year

--

-- Example Usage:

-- SELECT dbo.uf\_return\_total\_income\_by\_year(2023);

--

-- Result of the action:

-- Returns the total income generated in the specified year as DECIMAL(10, 2).

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER FUNCTION dbo.uf\_return\_total\_income\_by\_year

(

@Year INT

)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @TotalIncome DECIMAL(10, 2);

SELECT @TotalIncome = SUM((((bookPrice + bookDiscountAmnt) / 115) \* 15) - bookDiscountAmnt)

FROM tbl\_booking

WHERE YEAR(bookCreatedAt) = @Year;

RETURN @TotalIncome;

END;

#### uf\_return\_total\_booking\_price\_by\_customer

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_total\_booking\_price\_by\_customer

---------------------------------------------------------------

-- This function calculates the total booking price for a given customer by summing up the prices of all bookings made by that customer.

--

-- Input Parameters:

-- - @CustomerId: The identifier of the customer for whom the total booking price is to be calculated.

--

-- Output Parameters:

-- - @TotalPrice : The total booking price for the specified customer

--

-- Example Usage:

-- SELECT dbo.uf\_return\_total\_booking\_price\_by\_customer(123);

--

-- Result of the action:

-- Returns the total booking price for the specified customer as DECIMAL(10, 2).

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER FUNCTION dbo.uf\_return\_total\_booking\_price\_by\_customer

(

@CustomerId INT

)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @TotalPrice DECIMAL(10, 2);

SELECT @TotalPrice = SUM(bookPrice)

FROM tbl\_booking

WHERE bookCustId = @CustomerId;

RETURN @TotalPrice;

END;

#### uf\_return\_hotel\_count\_by\_stars\_and\_country

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_hotel\_count\_by\_stars\_and\_country

---------------------------------------------------------------

-- This function calculates the number of hotels with a specified number of stars in a given country.

--

-- Input Parameters:

-- - @Stars: The number of stars of the hotels.

-- - @CountryName: The name of the country.

--

-- Output Parameters:

-- - @HotelCount : The count of hotels with the specified number of stars in the specified country.

--

-- Example Usage:

-- SELECT dbo.uf\_return\_hotel\_count\_by\_stars\_and\_country(5, 'France');

--

-- Result of the action:

-- Returns the count of hotels with the specified number of stars in the specified country.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER FUNCTION dbo.uf\_return\_hotel\_count\_by\_stars\_and\_country

(

@Stars INT,

@CountryName VARCHAR(45)

)

RETURNS INT

AS

BEGIN

DECLARE @HotelCount INT;

SELECT @HotelCount = COUNT(hotId)

FROM tbl\_hotel h

INNER JOIN tbl\_address a ON h.hotAddrId = a.addrId

INNER JOIN tbl\_city c ON a.addrCityId = c.cityId

INNER JOIN tbl\_country co ON c.ctryId = co.ctryId

WHERE hotStars = @Stars AND co.ctryName = @CountryName;

RETURN @HotelCount;

END;

#### uf\_return\_flight\_count\_between\_cities

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_flight\_count\_between\_cities

---------------------------------------------------------------

-- This function calculates the number of flights available between two specified cities.

--

-- Input Parameters:

-- - @StartCity: The name of the starting city.

-- - @EndCity: The name of the destination city.

--

-- Output Parameters:

-- - @FlightCount : The count of flights available between the specified start and end cities.

--

-- Example Usage:

-- SELECT dbo.uf\_return\_flight\_count\_between\_cities('Norsup', 'Port Vila');

--

-- Result of the action:

-- Returns the count of flights available between the specified start and end cities.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER FUNCTION dbo.uf\_return\_flight\_count\_between\_cities

(

@StartCity VARCHAR(60),

@EndCity VARCHAR(60)

)

RETURNS INT

AS

BEGIN

DECLARE @FlightCount INT;

SELECT @FlightCount = COUNT(fliId)

FROM tbl\_flight f

INNER JOIN tbl\_city s ON f.fliStartCityId = s.cityId

INNER JOIN tbl\_city e ON f.fliEndCityId = e.cityId

WHERE s.cityName = @StartCity AND e.cityName = @EndCity;

RETURN @FlightCount;

END;

#### uf\_return\_country\_name

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_country\_name

---------------------------------------------------------------

-- This function retrieves the name of a country based on its identifier.

--

-- Input Parameters:

-- - @CountryId: The identifier of the country.

--

-- Output Parameters:

-- - @CountryName : The name of the country corresponding to the provided identifier

--

-- Example Usage:

-- SELECT dbo.uf\_return\_country\_name(1);

--

-- Result of the action:

-- Returns the name of the country corresponding to the provided identifier.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER FUNCTION dbo.uf\_return\_country\_name

(

@CountryId INT

)

RETURNS VARCHAR(100)

AS

BEGIN

DECLARE @CountryName VARCHAR(100);

SELECT @CountryName = ctryName

FROM tbl\_country

WHERE ctryId = @CountryId;

RETURN @CountryName;

END;

#### uf\_return\_continent\_name

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_continent\_name

---------------------------------------------------------------

-- This function retrieves the name of a continent based on its identifier.

--

-- Input Parameters:

-- - @ContinentId: The identifier of the continent.

--

-- Output Parameters:

-- - @ContinentName : The name of the continent corresponding to the provided identifier.

--

-- Example Usage:

-- SELECT dbo.uf\_return\_continent\_name(1);

--

-- Result of the action:

-- Returns the name of the continent corresponding to the provided identifier.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER FUNCTION dbo.uf\_return\_continent\_name

(

@ContinentId INT

)

RETURNS VARCHAR(100)

AS

BEGIN

DECLARE @ContinentName VARCHAR(100);

SELECT @ContinentName = contName

FROM tbl\_continent

WHERE contId = @ContinentId;

RETURN @ContinentName;

END;

#### uf\_return\_city\_name

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_city\_name

---------------------------------------------------------------

-- This function retrieves the name of a city based on its identifier.

--

-- Input Parameters:

-- - @CityId: The identifier of the city.

--

-- Output Parameters:

-- - @CityName : The name of the city corresponding to the provided identifier.

--

-- Example Usage:

-- SELECT dbo.uf\_return\_city\_name(1);

--

-- Result of the action:

-- Returns the name of the city corresponding to the provided identifier.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER FUNCTION dbo.uf\_return\_city\_name

(

@CityId INT

)

RETURNS VARCHAR(100)

AS

BEGIN

DECLARE @CityName VARCHAR(100);

SELECT @CityName = cityName

FROM tbl\_city

WHERE cityId = @CityId;

RETURN @CityName;

END;

#### uf\_return\_busRoute\_count\_between\_cities

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_busRoute\_count\_between\_cities

---------------------------------------------------------------

-- This function calculates the number of bus routes available between two specified cities.

--

-- Input Parameters:

-- - @StartCity: The name of the starting city.

-- - @EndCity: The name of the destination city.

--

-- Output Parameters:

-- - @BusRouteCount : The count of bus routes available between the specified start and end cities.

--

-- Example Usage:

-- SELECT dbo.uf\_return\_busRoute\_count\_between\_cities('Norsup', 'Port Vila');

--

-- Result of the action:

-- Returns the count of bus routes available between the specified start and end cities.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER FUNCTION dbo.uf\_return\_busRoute\_count\_between\_cities

(

@StartCity VARCHAR(60),

@EndCity VARCHAR(60)

)

RETURNS INT

AS

BEGIN

DECLARE @BusRouteCount INT;

SELECT @BusRouteCount = COUNT(busRouteId)

FROM tbl\_busroute br

INNER JOIN tbl\_city s ON br.busRouteStartCityId = s.cityId

INNER JOIN tbl\_city e ON br.busRouteEndCityId = e.cityId

WHERE s.cityName = @StartCity AND e.cityName = @EndCity;

RETURN @BusRouteCount;

END;

#### uf\_return\_booking\_count\_by\_customer

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_booking\_count\_by\_customer

---------------------------------------------------------------

-- This function calculates the total number of bookings made by a given customer.

--

-- Input Parameters:

-- - @CustomerId: The identifier of the customer for whom the total booking count is to be calculated.

--

-- Output Parameters:

-- - @BookingCount : The total number of bookings made by the specified customer as INT.

--

-- Example Usage:

-- SELECT dbo.uf\_return\_booking\_count\_by\_customer(123);

--

-- Result of the action:

-- Returns the total number of bookings made by the specified customer as INT.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

Go

CREATE OR ALTER FUNCTION dbo.uf\_return\_booking\_count\_by\_customer

(

@CustomerId INT

)

RETURNS INT

AS

BEGIN

DECLARE @BookingCount INT;

SELECT @BookingCount = COUNT(bookId)

FROM tbl\_booking

WHERE bookCustId = @CustomerId;

RETURN @BookingCount;

END;

#### uf\_return\_average\_package\_price\_by\_country

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_average\_package\_price\_by\_country

---------------------------------------------------------------

-- This function calculates the average package price for a given country by computing the average price of all packages offered in that country.

--

-- Input Parameters:

-- - @CountryName: The name of the country for which the average package price is to be calculated.

--

-- Output Parameters:

-- - @AveragePrice : The average package price for the specified country

--

-- Example Usage:

-- SELECT dbo.uf\_return\_average\_package\_price\_by\_country('France');

--

-- Result of the action:

-- Returns the average package price for the specified country as DECIMAL(10, 2).

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER FUNCTION dbo.uf\_return\_average\_package\_price\_by\_country

(

@CountryName VARCHAR(45)

)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @AveragePrice DECIMAL(10, 2);

SELECT @AveragePrice = AVG(p.packPrice)

FROM tbl\_package p

INNER JOIN tbl\_city c ON p.packCityId = c.cityId

INNER JOIN tbl\_country co ON c.ctryId = co.ctryId

WHERE co.ctryName = @CountryName;

RETURN @AveragePrice;

END;

#### uf\_return\_average\_package\_duration\_by\_country

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_average\_package\_duration\_by\_country

---------------------------------------------------------------

-- This function calculates the average package duration for a given country by computing the average duration of all packages offered in that country.

--

-- Input Parameters:

-- - @CountryName: The name of the country for which the average package duration is to be calculated.

--

-- Output Parameters:

-- - @AverageDuration : The average package duration for the specified country

--

-- Example Usage:

-- SELECT dbo.uf\_return\_average\_package\_duration\_by\_country('France');

--

-- Result of the action:

-- Returns the average package duration for the specified country as DECIMAL(10, 2).

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER FUNCTION dbo.uf\_return\_average\_package\_duration\_by\_country

(

@CountryName VARCHAR(45)

)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @AverageDuration DECIMAL(10, 2);

SELECT @AverageDuration = AVG(packDuration)

FROM tbl\_package p

INNER JOIN tbl\_city c ON p.packCityId = c.cityId

INNER JOIN tbl\_country co ON c.ctryId = co.ctryId

WHERE co.ctryName = @CountryName;

RETURN @AverageDuration;

END;

#### uf\_return\_average\_hotel\_price\_by\_city

---------------------------------------------------------------

--- FUNCTION DEFINITION

--- uf\_return\_average\_hotel\_price\_by\_city

---------------------------------------------------------------

-- This function calculates the average price per night of hotels in a specific city.

--

-- Input Parameters:

-- - @CityName: The name of the city for which the average hotel price is to be calculated.

--

-- Output Parameters:

-- - @AveragePrice : The average price per night of hotels in the specified city

--

-- Example Usage:

-- SELECT dbo.uf\_return\_average\_hotel\_price\_by\_city('Paris');

--

-- Result of the action:

-- Returns the average price per night of hotels in the specified city as DECIMAL(10, 2).

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER FUNCTION dbo.uf\_return\_average\_hotel\_price\_by\_city

(

@CityName VARCHAR(60)

)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @AveragePrice DECIMAL(10, 2);

SELECT @AveragePrice = AVG(hotPricePerNight)

FROM tbl\_hotel h

INNER JOIN tbl\_address a ON h.hotAddrId = a.addrId

INNER JOIN tbl\_city c ON a.addrCityId = c.cityId

WHERE c.cityName = @CityName;

RETURN @AveragePrice;

END;

# Wykaz utworzonych wyzwalaczy

/\*

This SQL query returns the names of the triggers and

the names of the tables to which they are assigned

\*/

SELECT

name AS TriggerName,

OBJECT\_NAME(parent\_id) AS TableName

FROM sys.triggers

ORDER BY TableName, TriggerName;

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

26 List of triggers created

# Skrypt tworzenia wyzwalaczy

#### trg\_package\_audit

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_package\_audit

---------------------------------------------------------------

-- This trigger is designed to audit DELETE operations on the tbl\_package table.

--

-- Trigger Events:

-- AFTER DELETE

--

-- Result of the action:

-- Prints 'Action: DELETE' to indicate that a DELETE operation has been performed on the tbl\_package table.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_package\_audit

ON tbl\_package

AFTER DELETE

AS

BEGIN

PRINT 'Action: DELETE';

END;

#### trg\_hotel\_audit

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_hotel\_audit

---------------------------------------------------------------

-- This trigger is designed to audit changes made to the tbl\_hotel table, including INSERT, UPDATE,

-- and DELETE operations.

--

-- Trigger Events:

-- AFTER INSERT, UPDATE, DELETE

--

-- Result of the action:

-- Audits the changes made to hotel records, printing the action performed (INSERT, UPDATE, DELETE)

-- and the affected hotel and address details.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_hotel\_audit

ON tbl\_hotel

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @Action NVARCHAR(10);

IF EXISTS (SELECT \* FROM inserted) AND EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'UPDATE';

END

ELSE IF EXISTS (SELECT \* FROM inserted)

BEGIN

SET @Action = 'INSERT';

END

ELSE IF EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'DELETE';

END

PRINT 'Action: ' + @Action;

IF @Action IN ('INSERT', 'UPDATE')

BEGIN

DECLARE @hotName VARCHAR(50);

DECLARE @hotPricePerNight DECIMAL(10, 2);

DECLARE @hotTelephoneNo VARCHAR(20);

DECLARE @hotContactEmail VARCHAR(45);

DECLARE @hotStars INT;

DECLARE @cityId INT;

DECLARE @addrPostalCode VARCHAR(20);

DECLARE @addrRegion VARCHAR(100);

DECLARE @addrStreet VARCHAR(255);

DECLARE @addrHouseNo VARCHAR(10);

SELECT @hotName = hotName,

@hotPricePerNight = hotPricePerNight,

@hotTelephoneNo = hotTelephoneNo,

@hotContactEmail = hotContactEmail,

@hotStars = hotStars,

@cityId = a.addrCityId,

@addrPostalCode = a.addrPostalCode,

@addrRegion = a.addrRegion,

@addrStreet = a.addrStreet,

@addrHouseNo = a.addrHouseNo

FROM inserted i

INNER JOIN tbl\_address a ON i.hotAddrId = a.addrId;

PRINT '';

PRINT '--------HOTEL--------';

PRINT 'hotName : ' + ISNULL(@hotName, '');

PRINT 'hotPricePerNight : ' + ISNULL(CONVERT(VARCHAR(20), @hotPricePerNight), '');

PRINT 'hotTelephoneNo : ' + ISNULL(@hotTelephoneNo, '');

PRINT 'hotContactEmail : ' + ISNULL(@hotContactEmail, '');

PRINT 'hotStars : ' + ISNULL(CONVERT(VARCHAR(10), @hotStars), '');

PRINT '';

PRINT '--------ADDRESS--------';

PRINT 'cityId : ' + ISNULL(CONVERT(VARCHAR(10), @cityId), '');

PRINT 'addrPostalCode : ' + ISNULL(@addrPostalCode, '');

PRINT 'addrRegion : ' + ISNULL(@addrRegion, '');

PRINT 'addrStreet : ' + ISNULL(@addrStreet, '');

PRINT 'addrHouseNo : ' + ISNULL(@addrHouseNo, '');

PRINT '';

END

END;

#### trg\_guide\_audit

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_guide\_audit

---------------------------------------------------------------

-- This trigger is designed to audit changes made to the tbl\_guide table, including INSERT, UPDATE,

-- and DELETE operations.

--

-- Trigger Events:

-- AFTER INSERT, UPDATE, DELETE

--

-- Result of the action:

-- Audits the changes made to guide records, printing the action performed (INSERT, UPDATE, DELETE)

-- and the affected guide details.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_guide\_audit

ON tbl\_guide

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @Action NVARCHAR(10);

IF EXISTS (SELECT \* FROM inserted) AND EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'UPDATE';

END

ELSE IF EXISTS (SELECT \* FROM inserted)

BEGIN

SET @Action = 'INSERT';

END

ELSE IF EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'DELETE';

END

PRINT 'Action: ' + @Action;

IF @Action IN ('INSERT', 'UPDATE')

BEGIN

DECLARE @guideName VARCHAR(50);

DECLARE @guideSurname VARCHAR(50);

DECLARE @guideCityId INT;

DECLARE @guidePhone VARCHAR(20);

SELECT @guideName = guideName,

@guideSurname = guideSurname,

@guideCityId = guideCityId,

@guidePhone = guidePhone

FROM inserted i;

PRINT '';

PRINT '--------GUIDE--------';

PRINT 'guideName : ' + ISNULL(@guideName, '');

PRINT 'guideSurname : ' + ISNULL(@guideSurname, '');

PRINT 'guideCityId : ' + ISNULL(CONVERT(VARCHAR(10), @guideCityId), '');

PRINT 'guidePhone : ' + ISNULL(@guidePhone, '');

PRINT '';

END

END;

#### trg\_flight\_audit

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_flight\_audit

---------------------------------------------------------------

-- This trigger is designed to audit changes made to the tbl\_flight table, including INSERT, UPDATE,

-- and DELETE operations.

--

-- Trigger Events:

-- AFTER INSERT, UPDATE, DELETE

--

-- Result of the action:

-- Audits the changes made to flight records, printing the action performed (INSERT, UPDATE, DELETE)

-- and the affected flight details.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_flight\_audit

ON tbl\_flight

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @Action NVARCHAR(10);

IF EXISTS (SELECT \* FROM inserted) AND EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'UPDATE';

END

ELSE IF EXISTS (SELECT \* FROM inserted)

BEGIN

SET @Action = 'INSERT';

END

ELSE IF EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'DELETE';

END

PRINT 'Action: ' + @Action;

IF @Action IN ('INSERT', 'UPDATE')

BEGIN

DECLARE @fliStartCityId INT;

DECLARE @fliEndCityId INT;

DECLARE @fliStartTime DATETIME;

DECLARE @fliEndTime DATETIME;

DECLARE @fliClass VARCHAR(10);

DECLARE @fliPrice DECIMAL(10, 2);

SELECT @fliStartCityId = fliStartCityId,

@fliEndCityId = fliEndCityId,

@fliStartTime = fliStartTime,

@fliEndTime = fliEndTime,

@fliClass = fliClass,

@fliPrice = fliPrice

FROM inserted i;

PRINT '';

PRINT '--------FLIGHT--------';

PRINT 'fliStartCityId : ' + ISNULL(CONVERT(VARCHAR(10), @fliStartCityId), '');

PRINT 'fliEndCityId : ' + ISNULL(CONVERT(VARCHAR(10), @fliEndCityId), '');

PRINT 'fliStartTime : ' + ISNULL(CONVERT(VARCHAR(30), @fliStartTime, 120), '');

PRINT 'fliEndTime : ' + ISNULL(CONVERT(VARCHAR(30), @fliEndTime, 120), '');

PRINT 'fliClass : ' + ISNULL(@fliClass, '');

PRINT 'fliPrice : ' + ISNULL(CONVERT(VARCHAR(20), @fliPrice), '');

PRINT '';

END

END;

#### trg\_driver\_audit

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_driver\_audit

---------------------------------------------------------------

-- This trigger is designed to audit changes made to the tbl\_driver table, including INSERT, UPDATE,

-- and DELETE operations.

--

-- Trigger Events:

-- AFTER INSERT, UPDATE, DELETE

--

-- Result of the action:

-- Audits the changes made to driver records, printing the action performed (INSERT, UPDATE, DELETE)

-- and the affected driver details.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_driver\_audit

ON tbl\_driver

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @Action NVARCHAR(10);

IF EXISTS (SELECT \* FROM inserted) AND EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'UPDATE';

END

ELSE IF EXISTS (SELECT \* FROM inserted)

BEGIN

SET @Action = 'INSERT';

END

ELSE IF EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'DELETE';

END

PRINT 'Action: ' + @Action;

IF @Action IN ('INSERT', 'UPDATE')

BEGIN

DECLARE @driverName VARCHAR(50);

DECLARE @driverSurName VARCHAR(50);

DECLARE @driverPhone VARCHAR(20);

SELECT @driverName = driverName,

@driverSurName = driverSurName,

@driverPhone = driverPhone

FROM inserted i;

PRINT '';

PRINT '--------DRIVER--------';

PRINT 'driverName : ' + ISNULL(@driverName, '');

PRINT 'driverSurName : ' + ISNULL(@driverSurName, '');

PRINT 'driverPhone : ' + ISNULL(@driverPhone, '');

PRINT '';

END

END;

#### trg\_customer\_audit

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_customer\_audit

---------------------------------------------------------------

-- This trigger is designed to audit changes made to the tbl\_customer table, including INSERT, UPDATE,

-- and DELETE operations.

--

-- Trigger Events:

-- AFTER INSERT, UPDATE, DELETE

--

-- Result of the action:

-- Audits the changes made to customer records, printing the action performed (INSERT, UPDATE, DELETE)

-- and the affected customer and address details.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_customer\_audit

ON tbl\_customer

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @Action NVARCHAR(10);

IF EXISTS (SELECT \* FROM inserted) AND EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'UPDATE';

END

ELSE IF EXISTS (SELECT \* FROM inserted)

BEGIN

SET @Action = 'INSERT';

END

ELSE IF EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'DELETE';

END

PRINT 'Action: ' + @Action;

IF @Action IN ('INSERT', 'UPDATE')

BEGIN

DECLARE @custName VARCHAR(50);

DECLARE @custSurname VARCHAR(50);

DECLARE @custPhone VARCHAR(20);

DECLARE @custEmailAddress VARCHAR(255);

DECLARE @cityId INT;

DECLARE @addrPostalCode VARCHAR(20);

DECLARE @addrRegion VARCHAR(100);

DECLARE @addrStreet VARCHAR(255);

DECLARE @addrHouseNo VARCHAR(10);

SELECT @custName = custName,

@custSurname = custSurname,

@custPhone = custPhone,

@custEmailAddress = custEmailAddress,

@cityId = a.addrCityId,

@addrPostalCode = a.addrPostalCode,

@addrRegion = a.addrRegion,

@addrStreet = a.addrStreet,

@addrHouseNo = a.addrHouseNo

FROM inserted i

INNER JOIN tbl\_address a ON i.custAddrId = a.addrId;

PRINT '';

PRINT '--------CUSTOMER--------';

PRINT 'custName : ' + ISNULL(@custName, '');

PRINT 'custSurname : ' + ISNULL(@custSurname, '');

PRINT 'custPhone : ' + ISNULL(@custPhone, '');

PRINT 'custEmailAddress : ' + ISNULL(@custEmailAddress, '');

PRINT '';

PRINT '--------ADDRESS--------';

PRINT 'cityId : ' + ISNULL(CONVERT(VARCHAR(10), @cityId), '');

PRINT 'addrPostalCode : ' + ISNULL(@addrPostalCode, '');

PRINT 'addrRegion : ' + ISNULL(@addrRegion, '');

PRINT 'addrStreet : ' + ISNULL(@addrStreet, '');

PRINT 'addrHouseNo : ' + ISNULL(@addrHouseNo, '');

PRINT '';

END

END;

#### trg\_calculate\_package\_details

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_calculate\_package\_details

---------------------------------------------------------------

-- This trigger is designed to calculate package details after INSERT or UPDATE operations on the tbl\_package table.

--

-- Trigger Events:

-- AFTER INSERT, UPDATE

--

-- Result of the action:

-- Calls the stored procedure up\_calculate\_package\_details to update package details based on the inserted or updated data.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_calculate\_package\_details

ON tbl\_package

AFTER INSERT,UPDATE

AS

BEGIN

DECLARE @packId INT

DECLARE @packHotId INT

DECLARE @packOutboundFlightId INT

DECLARE @packReturnFlightId INT

DECLARE @packOutboundBusRouteId INT

DECLARE @packReturnBusRouteId INT

SELECT @packId = packId,

@packHotId = packHotId,

@packOutboundFlightId = packOutboundFlightId,

@packReturnFlightId = packReturnFlightId,

@packOutboundBusRouteId = packOutboundBusRouteId,

@packReturnBusRouteId = packReturnBusRouteId

FROM inserted

EXEC up\_calculate\_package\_details @packId, @packHotId, @packOutboundFlightId, @packReturnFlightId, @packOutboundBusRouteId, @packReturnBusRouteId

END

#### trg\_calculate\_booking\_prices

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_calculate\_booking\_prices

---------------------------------------------------------------

-- This trigger is designed to calculate booking prices after INSERT or UPDATE operations on the tbl\_booking table.

--

-- Trigger Events:

-- AFTER INSERT, UPDATE

--

-- Result of the action:

-- Calls the stored procedure up\_calculate\_booking\_prices to update booking prices based on the inserted or updated data.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_calculate\_booking\_prices

ON tbl\_booking

AFTER INSERT,UPDATE

AS

BEGIN

DECLARE @bookPackageId INT

DECLARE @bookDiscountPercent DECIMAL(10, 2)

SELECT @bookPackageId = bookPackageId,

@bookDiscountPercent= bookDiscountPercent

FROM inserted

EXEC up\_calculate\_booking\_prices @bookPackageId, @bookDiscountPercent

END;

#### trg\_busRoute\_audit

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_busRoute\_audit

---------------------------------------------------------------

-- This trigger is designed to audit changes made to the tbl\_busroute table, including INSERT, UPDATE,

-- and DELETE operations.

--

-- Trigger Events:

-- AFTER INSERT, UPDATE, DELETE

--

-- Result of the action:

-- Audits the changes made to bus route records, printing the action performed (INSERT, UPDATE, DELETE)

-- and the affected bus route details.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_busRoute\_audit

ON tbl\_busroute

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @Action NVARCHAR(10);

IF EXISTS (SELECT \* FROM inserted) AND EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'UPDATE';

END

ELSE IF EXISTS (SELECT \* FROM inserted)

BEGIN

SET @Action = 'INSERT';

END

ELSE IF EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'DELETE';

END

PRINT 'Action: ' + @Action;

IF @Action IN ('INSERT', 'UPDATE')

BEGIN

DECLARE @driverId INT;

DECLARE @busRouteStartCityId INT;

DECLARE @busRouteEndCityId INT;

DECLARE @busRouteStartTime DATETIME;

DECLARE @busRouteEndTime DATETIME;

DECLARE @busRouteBusID INT;

DECLARE @busRoutePrice DECIMAL(10, 2);

SELECT @driverId = driverId,

@busRouteStartCityId = busRouteStartCityId,

@busRouteEndCityId = busRouteEndCityId,

@busRouteStartTime = busRouteStartTime,

@busRouteEndTime = busRouteEndTime,

@busRouteBusID = busRouteBusID,

@busRoutePrice = busRoutePrice

FROM inserted i;

PRINT '';

PRINT '--------BUS ROUTE--------';

PRINT 'driverId : ' + ISNULL(CONVERT(VARCHAR(10), @driverId), '');

PRINT 'busRouteStartCityId : ' + ISNULL(CONVERT(VARCHAR(10), @busRouteStartCityId), '');

PRINT 'busRouteEndCityId : ' + ISNULL(CONVERT(VARCHAR(10), @busRouteEndCityId), '');

PRINT 'busRouteStartTime : ' + ISNULL(CONVERT(VARCHAR(30), @busRouteStartTime, 120), '');

PRINT 'busRouteEndTime : ' + ISNULL(CONVERT(VARCHAR(30), @busRouteEndTime, 120), '');

PRINT 'busRouteBusID : ' + ISNULL(CONVERT(VARCHAR(10), @busRouteBusID), '');

PRINT 'busRoutePrice : ' + ISNULL(CONVERT(VARCHAR(20), @busRoutePrice), '');

PRINT '';

END

END;

#### trg\_bus\_audit

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_bus\_audit

---------------------------------------------------------------

-- This trigger is designed to audit changes made to the tbl\_bus table, including INSERT, UPDATE,

-- and DELETE operations.

--

-- Trigger Events:

-- AFTER INSERT, UPDATE, DELETE

--

-- Result of the action:

-- Audits the changes made to bus records, printing the action performed (INSERT, UPDATE, DELETE)

-- and the affected bus details.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_bus\_audit

ON tbl\_bus

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @Action NVARCHAR(10);

IF EXISTS (SELECT \* FROM inserted) AND EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'UPDATE';

END

ELSE IF EXISTS (SELECT \* FROM inserted)

BEGIN

SET @Action = 'INSERT';

END

ELSE IF EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'DELETE';

END

PRINT 'Action: ' + @Action;

IF @Action IN ('INSERT', 'UPDATE')

BEGIN

DECLARE @busNo VARCHAR(20);

DECLARE @busModel VARCHAR(50);

DECLARE @busColor VARCHAR(20);

SELECT @busNo = busNo,

@busModel = busModel,

@busColor = busColor

FROM inserted i;

PRINT '';

PRINT '--------BUS--------';

PRINT 'busNo : ' + ISNULL(@busNo, '');

PRINT 'busModel : ' + ISNULL(@busModel, '');

PRINT 'busColor : ' + ISNULL(@busColor, '');

PRINT '';

END

END;

#### trg\_booking\_audit

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_booking\_audit

---------------------------------------------------------------

-- This trigger is designed to audit DELETE operations on the tbl\_booking table.

--

-- Trigger Events:

-- AFTER DELETE

--

-- Example Usage:

-- No direct usage. This trigger automatically executes after DELETE operations on the tbl\_booking table.

--

-- Result of the action:

-- Prints 'Action: DELETE' to indicate that a DELETE operation has been performed on the tbl\_booking table.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_booking\_audit

ON tbl\_booking

AFTER DELETE

AS

BEGIN

PRINT 'Action: DELETE';

END;

#### trg\_address\_audit

---------------------------------------------------------------

--- TRIGGER DEFINITION

--- trg\_address\_audit

---------------------------------------------------------------

-- This trigger is designed to audit changes made to the tbl\_address table, including INSERT, UPDATE,

-- and DELETE operations.

--

-- Trigger Events:

-- AFTER INSERT, UPDATE, DELETE

--

-- Result of the action:

-- Audits the changes made to address records, printing the action performed (INSERT, UPDATE, DELETE)

-- and the affected address details.

---------------------------------------------------------------

USE TRAVEL\_AGENCY

GO

CREATE OR ALTER TRIGGER trg\_address\_audit

ON tbl\_address

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

SET NOCOUNT ON;

DECLARE @Action NVARCHAR(10);

IF EXISTS (SELECT \* FROM inserted) AND EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'UPDATE';

END

ELSE IF EXISTS (SELECT \* FROM inserted)

BEGIN

SET @Action = 'INSERT';

END

ELSE IF EXISTS (SELECT \* FROM deleted)

BEGIN

SET @Action = 'DELETE';

END

PRINT 'Action: ' + @Action;

IF @Action IN ('INSERT', 'UPDATE')

BEGIN

DECLARE @addrCityId INT;

DECLARE @addrPostalCode VARCHAR(20);

DECLARE @addrRegion VARCHAR(100);

DECLARE @addrStreet VARCHAR(255);

DECLARE @addrHouseNo VARCHAR(10);

SELECT @addrCityId = addrCityId,

@addrPostalCode = addrPostalCode,

@addrRegion = addrRegion,

@addrStreet = addrStreet,

@addrHouseNo = addrHouseNo

FROM inserted i;

PRINT '';

PRINT '--------ADDRESS--------';

PRINT 'addrCityId : ' + ISNULL(CONVERT(VARCHAR(10), @addrCityId), '');

PRINT 'addrPostalCode : ' + ISNULL(@addrPostalCode, '');

PRINT 'addrRegion : ' + ISNULL(@addrRegion, '');

PRINT 'addrStreet : ' + ISNULL(@addrStreet, '');

PRINT 'addrHouseNo : ' + ISNULL(@addrHouseNo, '');

PRINT '';

END

END;

# Wykaz obiektów wartości domyślnej

/\*

This query returns table names, column names, default constraint

names, and default expressions for columns in the database.

\*/

SELECT

OBJECT\_NAME(dc.parent\_object\_id) AS TableName,

c.name AS ColumnName,

dc.name AS DefaultObjectName,

dc.definition AS DefaultExpression

FROM sys.default\_constraints dc

INNER JOIN sys.columns c ON dc.parent\_object\_id = c.object\_id AND dc.parent\_column\_id = c.column\_id;

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

27 List of default value objects

# Skrypt tworzenia obiektów wartości domyślnej

USE TRAVEL\_AGENCY

GO

CREATE TABLE tbl\_booking (

bookId INT PRIMARY KEY IDENTITY(1,1),

bookCustId INT NOT NULL,

bookPackageId INT NOT NULL,

bookDiscountPercent DECIMAL(10, 2) NOT NULL DEFAULT 0,

bookDiscountAmnt DECIMAL(10, 2) NOT NULL DEFAULT 0,

bookPrice DECIMAL(10, 2),

bookCreatedAt DATETIME NOT NULL DEFAULT GETDATE(),

FOREIGN KEY (bookCustId) REFERENCES tbl\_customer(custId),

FOREIGN KEY (bookPackageId) REFERENCES tbl\_package(packId)

);

GO

# Schemat zaprojektowanego systemu

Obraz zawierający tekst, zrzut ekranu, numer, Równolegle

Opis wygenerowany automatycznie

28 The physical model of the database in graphical form

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, System operacyjny

Opis wygenerowany automatycznie

29.1 Input/output layer

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

29.2 Input/output layer

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, System operacyjny

Opis wygenerowany automatycznie

29.3 Input/output layer

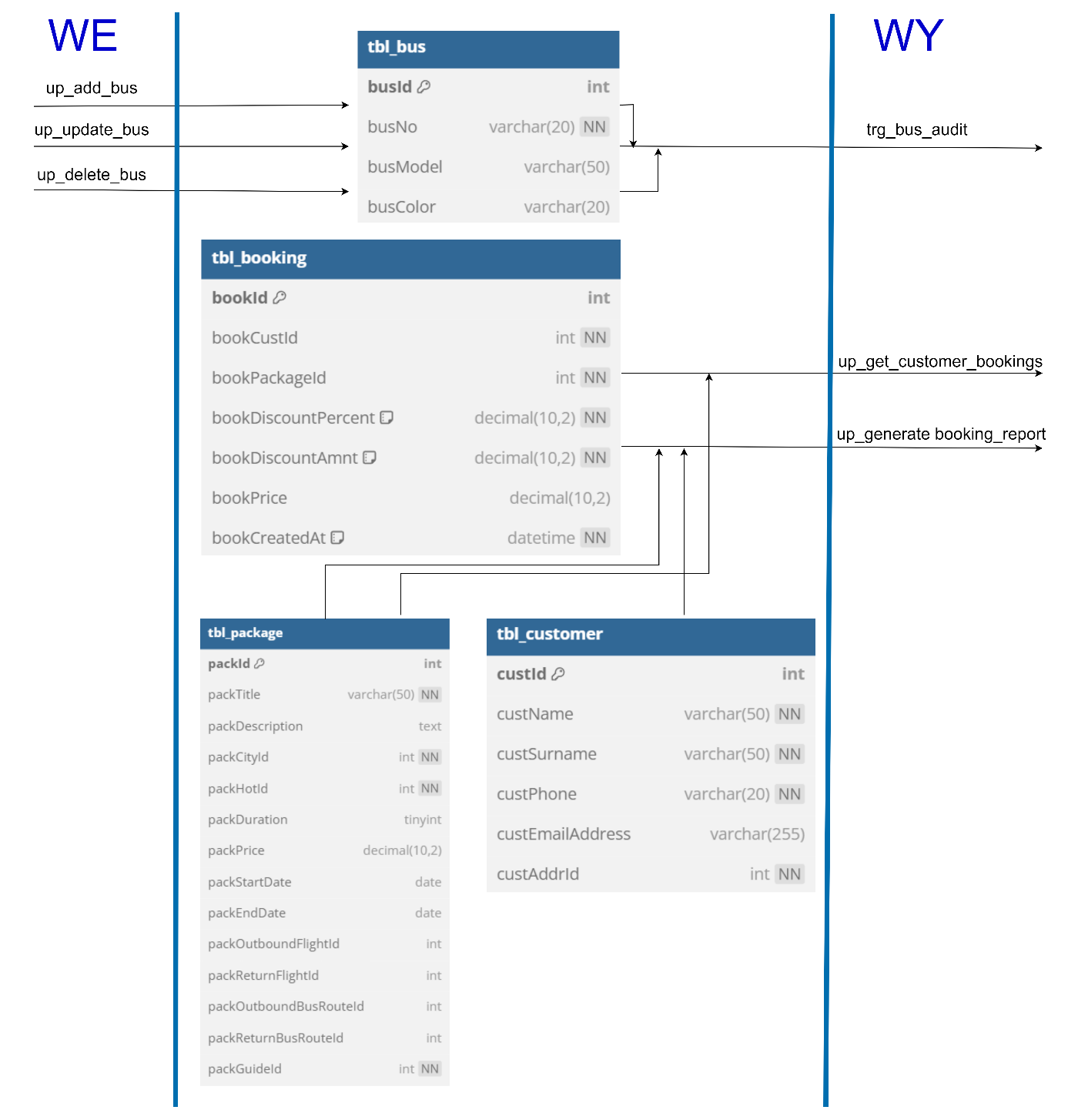
Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Strona internetowa

Opis wygenerowany automatycznie

29.4 Input/output layer



29.5 Input/output layer



29.6 Input/output layer

# Wnioski i podsumowanie

#### Wnioski:

Baza danych jako fundament systemu:

* Baza danych jest kluczowym elementem w architekturze systemu zarządzania podróżami, zapewniając strukturalne przechowywanie danych dotyczących klientów, ofert podróży, rezerwacji i transakcji.
* Wykorzystanie relacyjnej bazy danych umożliwia efektywne zarządzanie złożonymi relacjami między danymi, ułatwiając operacje wyszukiwania, filtrowania i analizy danych.

Programowanie serwera bazy danych:

* Tworzenie procedur składowanych, funkcji i wyzwalaczy w bazie danych pozwala na przeniesienie części logiki biznesowej do samej bazy danych, co może poprawić wydajność i zabezpieczenia systemu.
* Wykorzystanie odpowiednich indeksów i ograniczeń zapewnia optymalizację działania bazy danych oraz zachowanie integralności i spójności danych.

Zarządzanie danymi:

* Baza danych umożliwia skuteczne zarządzanie danymi klientów, ofert podróży, rezerwacji, płatności i innych informacji związanych z działalnością biura podróży.
* Odpowiednie zabezpieczenia i kontrole dostępu do danych są kluczowe dla zapewnienia poufności, integralności i dostępności informacji.

#### Podsumowanie:

Stworzona baza danych stanowi solidny fundament dla systemu zarządzania podróżami, umożliwiając kompleksowe zarządzanie danymi i operacjami biznesowymi. Programowanie serwera bazy danych pozwala na efektywne wykorzystanie mocy obliczeniowej bazy danych oraz wprowadzenie logiki biznesowej na poziomie danych. Dzięki odpowiednim indeksom, ograniczeniom i procedurom składowanym, baza danych zapewnia optymalną wydajność, integralność danych oraz zabezpieczenia systemu. Wdrażając tę bazę danych, biuro podróży może efektywnie zarządzać swoją działalnością, oferując klientom wysokiej jakości usługi podróżne. Jednocześnie, studiowanie i praktyczne zastosowanie technik programowania serwera bazy danych stanowi wartościowe doświadczenie dla osób zajmujących się programowaniem aplikacji bazodanowych oraz systemów informatycznych.

# Wykaz/spis ilustracji

[1 Title photo 1](#_Toc164260503)

[2 The physical model of the database in graphical form 6](#_Toc164260504)

[3 List of physical tables created 7](#_Toc164260505)

[4 List of implemented indexes 13](#_Toc164260506)

[5 List of created views 61](#_Toc164260507)

[6 Sample output of the vw\_package\_details view 62](#_Toc164260508)

[7 Sample output of the vw\_package\_by\_country view 63](#_Toc164260509)

[8 Sample output of the vw\_hotel\_contact\_details view 64](#_Toc164260510)

[9 Sample output of the vw\_hotel\_by\_country view 65](#_Toc164260511)

[10 Sample Output of the vw\_hotel\_by\_city view 66](#_Toc164260512)

[11 Sample Output of the vw\_hotel\_address\_details view 67](#_Toc164260513)

[12 Sample Output of the vw\_guide\_city view 68](#_Toc164260514)

[13 Sample Output of the vw\_flight\_details view 69](#_Toc164260515)

[14 Sample Output of the vw\_customers\_by\_country view 70](#_Toc164260516)

[15 Sample Output of the vw\_customer\_address view 71](#_Toc164260517)

[16 Sample Output of the vw\_city\_country view 72](#_Toc164260518)

[17 Sample Output of the vw\_busroute\_details view 73](#_Toc164260519)

[18 Sample Output of the vw\_busdriver\_details view 74](#_Toc164260520)

[19 Sample Output of the vw\_booking\_details view 75](#_Toc164260521)

[20 Sample output of the vw\_annual\_income view 76](#_Toc164260522)

[21 Sample Output of the up\_generate\_revenue\_report procedure 131](#_Toc164260523)

[22Sample Output of the up\_generate\_popularity\_report\_by\_date\_range procedure 132](#_Toc164260524)

[23 Sample Output of the up\_generate\_popularity\_report procedure 133](#_Toc164260525)

[24 Sample Output of the up\_generate\_booking\_report procedure 134](#_Toc164260526)

[25 List of created user functions 135](#_Toc164260527)

[26 List of triggers created 144](#_Toc164260528)

[27 List of default value objects 156](#_Toc164260529)

[28 The physical model of the database in graphical form 157](#_Toc164260530)

# Wykaz/spis tabel

Brak tabel do wyświetlenia.