**DATABASE PROJECT**

**NAME-1:MUHAMMAD ASAD HAYAT AWAN**

**ROLL-1: 241792**

**NAME-2:M.AFFAN TARAR**

**ROLL-2:241894**

**EXPLANATION:**

**Database Design Documentation for AH\_Travels**

**1. Project Overview**

The database is designed to support a travel agency system called **AH\_Travels**. It includes essential entities such as travel packages, customers, flights, hotels, travel guides, and associated business operations. The design ensures data normalization, consistency, and scalability, and supports operations like booking, customer management, and employee assignments.

**2. Schema Design Decisions**

**A. Core Entity: AH\_Travels**

* Acts as the central organization managing travel-related services.
* Contains TravelAgencyID, AgencyName, and Website to uniquely identify and describe each agency.
* Serves as a foreign key in related tables like Offers, TravelOffices, Flights, and CurrencyExchange.

**B. Travel Packages & Offers**

* Packages: Stores details about different travel packages.
* Offers: Links Packages and AH\_Travels, allowing each agency to provide specific offers with discounts.
* **Assumption:** Each offer is tied to exactly one package and one travel agency.

**C. Currency Exchange**

* Tracks supported currency conversions for customers traveling abroad.
* Includes a link to AH\_Travels to indicate which agency supports that conversion.

**D. Manager and CEO Hierarchy**

* Manager table stores managerial-level staff details.
* CEO table references Manager, assuming a direct reporting relationship (WorksForManagerID).
* **Assumption:** Each CEO works under one Manager; this relationship helps trace the chain of command.

**E. Travel Offices and Employees**

* TravelOffices are branches operated by the agency in various cities.
* Employees work under these offices and are linked via foreign key (OfficeID).
* **Assumption:** Every employee belongs to one office; offices are agency-specific.

**F. Flights**

* Stores airline and flight timing info.
* Linked to AH\_Travels, assuming the agency manages or books these flights.

**G. Hotels and Rooms**

* Hotels stores accommodation providers.
* Rooms linked to Hotels using HotelID, supporting room-type distinctions.
* **Assumption:** Each hotel can have multiple rooms, enabling detailed customer accommodation handling.

**H. Customers**

* Central to the system, customers can:
  + Be managed by an agency.
  + Be assisted by a travel guide.
  + Visit specific countries.
* Linked with foreign keys to:
  + AH\_Travels
  + TravelGuides
  + Countries
* **Assumption:** A customer can be assigned one guide and visit one primary country in a booking.

**I. Travel Guides**

* Guides are multi-lingual and support customers.
* Optionally linked to Employees (if they’re also staff members).
* **Assumption:** Some employees may function as guides, and this relationship is flexible.

**J. Countries and Locations**

* Countries represent destinations.
* Each country contains multiple TravelLocations (e.g., cities, landmarks).
* **Assumption:** Travel packages, visits, and bookings may be aligned to specific locations.

**3. Relationship Overview**

* **One-to-Many**:
  + One agency → many offers, flights, offices, currency exchanges.
  + One office → many employees.
  + One hotel → many rooms.
  + One country → many locations.
* **Many-to-One**:
  + Many customers → one agency, one guide, one country.

**4. Assumptions and Considerations**

* Each travel package is unique and tied to an agency through offers.
* Customers are served by only one travel guide per trip.
* Travel offices operate under a single agency.
* Room type distinctions help enable advanced filtering for bookings.
* A CEO reports to only one manager.

**5. Normalization**

The schema adheres to **3rd Normal Form (3NF)**:

* No repeating groups or redundant data.
* Non-key attributes are dependent only on primary keys.
* Referential integrity enforced via foreign keys.

**6. Future Scalability**

* The schema supports easy extension for:
  + Multi-country visits (via intermediate/junction table).
  + Booking transactions and payment history.
  + Real-time flight tracking and hotel availability.
  + Reviews and ratings for travel services.

**7. Additional Features Implemented**

* **Joins**: Used for reports like customer-location, flight-agency, hotel-room listings.
* **Stored Procedures**: For bulk insertion, customer registration, and reporting.
* **Triggers**: For logging updates/deletes and data validation.
* **CRUD Operations**: Ensured through parameterized queries.
* **Views**: (if used) can be created for simplified reporting.

**SQL CODE:**

**-- 1. AH\_Travels**

**CREATE TABLE AH\_Travels (**

**TravelAgencyID INT PRIMARY KEY,**

**AgencyName VARCHAR(100),**

**Website VARCHAR(100)**

**);**

**-- 2. Packages**

**CREATE TABLE Packages (**

**PackageID INT PRIMARY KEY,**

**PackageName VARCHAR(100),**

**Price DECIMAL(10, 2)**

**);**

**-- 3. CurrencyExchange**

**CREATE TABLE CurrencyExchange (**

**ExchangeID INT PRIMARY KEY,**

**FromCurrency VARCHAR(10),**

**ToCurrency VARCHAR(10)**

**);**

**-- 4. Offers**

**CREATE TABLE Offers (**

**OfferID INT PRIMARY KEY,**

**PackageID INT,**

**TravelAgencyID INT,**

**DiscountPercent INT,**

**FOREIGN KEY (PackageID) REFERENCES Packages(PackageID),**

**FOREIGN KEY (TravelAgencyID) REFERENCES AH\_Travels(TravelAgencyID)**

**);**

**-- 5. Manager**

**CREATE TABLE Manager (**

**ManagerID INT PRIMARY KEY,**

**FullName VARCHAR(100),**

**Email VARCHAR(100)**

**);**

**-- 6. CEO**

**CREATE TABLE CEO (**

**CEO\_ID INT PRIMARY KEY,**

**FullName VARCHAR(100),**

**WorksForManagerID INT,**

**FOREIGN KEY (WorksForManagerID) REFERENCES Manager(ManagerID)**

**);**

**-- 7. TravelOffices**

**CREATE TABLE TravelOffices (**

**OfficeID INT PRIMARY KEY,**

**City VARCHAR(50),**

**TravelAgencyID INT,**

**FOREIGN KEY (TravelAgencyID) REFERENCES AH\_Travels(TravelAgencyID)**

**);**

**-- 8. Employees**

**CREATE TABLE Employees (**

**EmployeeID INT PRIMARY KEY,**

**Name VARCHAR(100),**

**OfficeID INT,**

**FOREIGN KEY (OfficeID) REFERENCES TravelOffices(OfficeID)**

**);**

**-- 9. Flights**

**CREATE TABLE Flights (**

**FlightID INT PRIMARY KEY,**

**Airline VARCHAR(50),**

**DepartureTime TIME**

**);**

**-- 10. Hotels**

**CREATE TABLE Hotels (**

**HotelID INT PRIMARY KEY,**

**HotelName VARCHAR(100),**

**City VARCHAR(50)**

**);**

**-- 11. Rooms**

**CREATE TABLE Rooms (**

**RoomID INT PRIMARY KEY,**

**HotelID INT,**

**RoomType VARCHAR(50),**

**FOREIGN KEY (HotelID) REFERENCES Hotels(HotelID)**

**);**

**-- 12. Customers**

**CREATE TABLE Customers (**

**CustomerID INT PRIMARY KEY,**

**FullName VARCHAR(100),**

**Email VARCHAR(100)**

**);**

**-- 13. TravelGuides**

**CREATE TABLE TravelGuides (**

**GuideID INT PRIMARY KEY,**

**Name VARCHAR(100),**

**Language VARCHAR(50)**

**);**

**-- 14. Countries**

**CREATE TABLE Countries (**

**CountryCode CHAR(2) PRIMARY KEY,**

**CountryName VARCHAR(100),**

**TravelTimeHours DECIMAL(4,1)**

**);**

**-- 15. TravelLocations**

**CREATE TABLE TravelLocations (**

**LocationID INT PRIMARY KEY,**

**CountryCode CHAR(2),**

**LocationName VARCHAR(100),**

**FOREIGN KEY (CountryCode) REFERENCES Countries(CountryCode)**

**);**

**-- Relationships**

**-- AH Travels manages Customers**

**CREATE TABLE AgencyCustomers (**

**TravelAgencyID INT,**

**CustomerID INT,**

**PRIMARY KEY (TravelAgencyID, CustomerID),**

**FOREIGN KEY (TravelAgencyID) REFERENCES AH\_Travels(TravelAgencyID),**

**FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)**

**);**

**-- Customers visit Countries**

**CREATE TABLE CustomerVisits (**

**VisitID INT PRIMARY KEY,**

**CustomerID INT,**

**CountryCode CHAR(2),**

**VisitDate DATE,**

**FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),**

**FOREIGN KEY (CountryCode) REFERENCES Countries(CountryCode)**

**);**

**-- Customers helped by TravelGuides**

**CREATE TABLE GuideHelp (**

**GuideHelpID INT PRIMARY KEY,**

**CustomerID INT,**

**GuideID INT,**

**FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),**

**FOREIGN KEY (GuideID) REFERENCES TravelGuides(GuideID)**

**);**

**-- 1. AH\_Travels manages Customers**

**ALTER TABLE Customers**

**ADD TravelAgencyID INT,**

**ADD FOREIGN KEY (TravelAgencyID) REFERENCES AH\_Travels(TravelAgencyID);**

**-- 2. AH\_Travels has Flights**

**ALTER TABLE Flights**

**ADD TravelAgencyID INT,**

**ADD FOREIGN KEY (TravelAgencyID) REFERENCES AH\_Travels(TravelAgencyID);**

**-- 3. Customers visit Countries**

**ALTER TABLE Customers**

**ADD CountryCode CHAR(2),**

**ADD FOREIGN KEY (CountryCode) REFERENCES Countries(CountryCode);**

**-- 4. Customers helped by TravelGuides**

**ALTER TABLE Customers**

**ADD GuideID INT,**

**ADD FOREIGN KEY (GuideID) REFERENCES TravelGuides(GuideID);**

**-- 5. CurrencyExchange linked to AH\_Travels**

**ALTER TABLE CurrencyExchange**

**ADD TravelAgencyID INT,**

**ADD FOREIGN KEY (TravelAgencyID) REFERENCES AH\_Travels(TravelAgencyID);**

**-- 6. (Optional) TravelGuides contain Employees (if this relationship is valid)**

**ALTER TABLE TravelGuides**

**ADD EmployeeID INT,**

**ADD FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID);**