

Database Schema and Functionality Summary

This database schema is designed for a **Travel Agency Management System**. It supports managing employees, travel packages, destinations, customers, and bookings while ensuring smooth operations through structured relationships, procedures, and triggers.

Part A: Database Setup and Creation

1. Database Creation

- The database is named **TravelAgencyDB**.
- It ensures that any existing database with the same name is dropped before creating a new one.

2. Tables and Relationships

Employees Table

- Stores employee details.
- Role can be **Guide, Admin, or Analyst**.
- Department is limited to **SALES or ACCOUNTING**.

GuidesLanguages Table

- Manages **languages spoken by guides**.
- Each guide can speak multiple languages.
- Establishes a **foreign key relationship** with Employees.

Packages Table

- Represents **travel packages**.
- Each package has a **start and end date, cost per person, transport method, guide assignment, and status**.
- The Status can be:
 - **Πλήρες (Full)**
 - **Ανοικτό (Open)**
 - **Κλειστό (Closed)**
 - **Ακυρωμένο (Canceled)**
- The Realized field indicates if the package was completed.

Destinations Table

- Stores information about **destinations**, including a **description and language**.

PackageDestinations Table (Many-to-Many Relationship)

- Links **Packages** with **Destinations** (each package can include multiple destinations).

Customers Table

- Stores **customer details**, including contact information.

Bookings Table

- Links **Customers** with **Packages**.
- Each booking includes a **seat number**.

PackageCategories Table (Many-to-Many Relationship)

- Categorizes packages as **Romantic, Winter, or Summer**.

SalesTracking Table

- Tracks **total bookings, revenue, and available seats** per package.

PackageReviews Table

- Stores **customer reviews** for travel packages.
- Ensures **each booking can have only one review**.

Part B: Sample Data Inserts

Sample data is inserted for:

- **Employees** (e.g., Guides, Admins, Analysts).

- **GuidesLanguages** (e.g., A guide speaks both English and Greek).
- **Packages** (e.g., Summer trips to Santorini, Crete).
- **Destinations** (e.g., Santorini, Athens, Crete).
- **Package Destinations** (Mapping destinations to packages).
- **Customers** (Customer names and contact details).
- **Bookings** (Which customer booked which package).
- **Package Categories** (Assigning categories to travel packages).
- **Package Reviews** (Customer ratings and comments).

Part C: SQL Queries for Data Analysis

1. Find the Most Popular Destination

- Counts bookings per **destination** to determine the most booked destination.

2. Retrieve Detailed Package Information

- Lists **all destinations, descriptions, languages, transport methods, guides, and categories** for each package.

3. Calculate Total Revenue per Package

- Computes **total revenue** for each package based on bookings.

4. List Packages Managed by Each Guide

- Displays which **travel packages** each guide is responsible for.

5. Identify the Customer with the Most Bookings

- Finds the **most frequent traveler**.

Part D: Stored Procedures for Automation

1. Add a Package Review

- Ensures that a review can only be added for a **valid booking**.
- If an invalid BookingID is provided, an **error is raised**.

Usage Example

CALL AddPackageReview(401, 4, 'Wonderful guide and locations!');

2. Generate Passenger List for a Package

- Returns a **list of all customers** booked on a specific package.

Usage Example

CALL GeneratePassengerList(101);

Part E: Triggers for Automated Sales Tracking

1. Trigger: After Insert on Bookings

- Updates **SalesTracking** when a new booking is made.
- Ensures **Booking Count, Total Cost, and Available Seats** are updated.

2. Trigger: After Update on Bookings

- Automatically updates **SalesTracking** when booking details are changed.

3. Trigger: After Delete on Bookings

- Updates **SalesTracking** when a booking is canceled.

Testing the Trigger

```
INSERT INTO Bookings (BookingID, PackageID, CustomerID, SeatNumber)
VALUES (404, 101, 302, 2);
```

- This triggers an **update** in SalesTracking.

Viewing the Updated SalesTracking Table

```
SELECT * FROM SalesTracking;
```

Conclusion

This **Travel Agency Management System** is structured to efficiently handle:

- **Employee roles and language skills.**
- **Travel packages, destinations, and bookings.**
- **Customer reviews and feedback management.**
- **Sales tracking through triggers and stored procedures.**

The system ensures **data integrity, automation, and efficiency** in travel agency operations.