**DATABASE PROJECT**

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

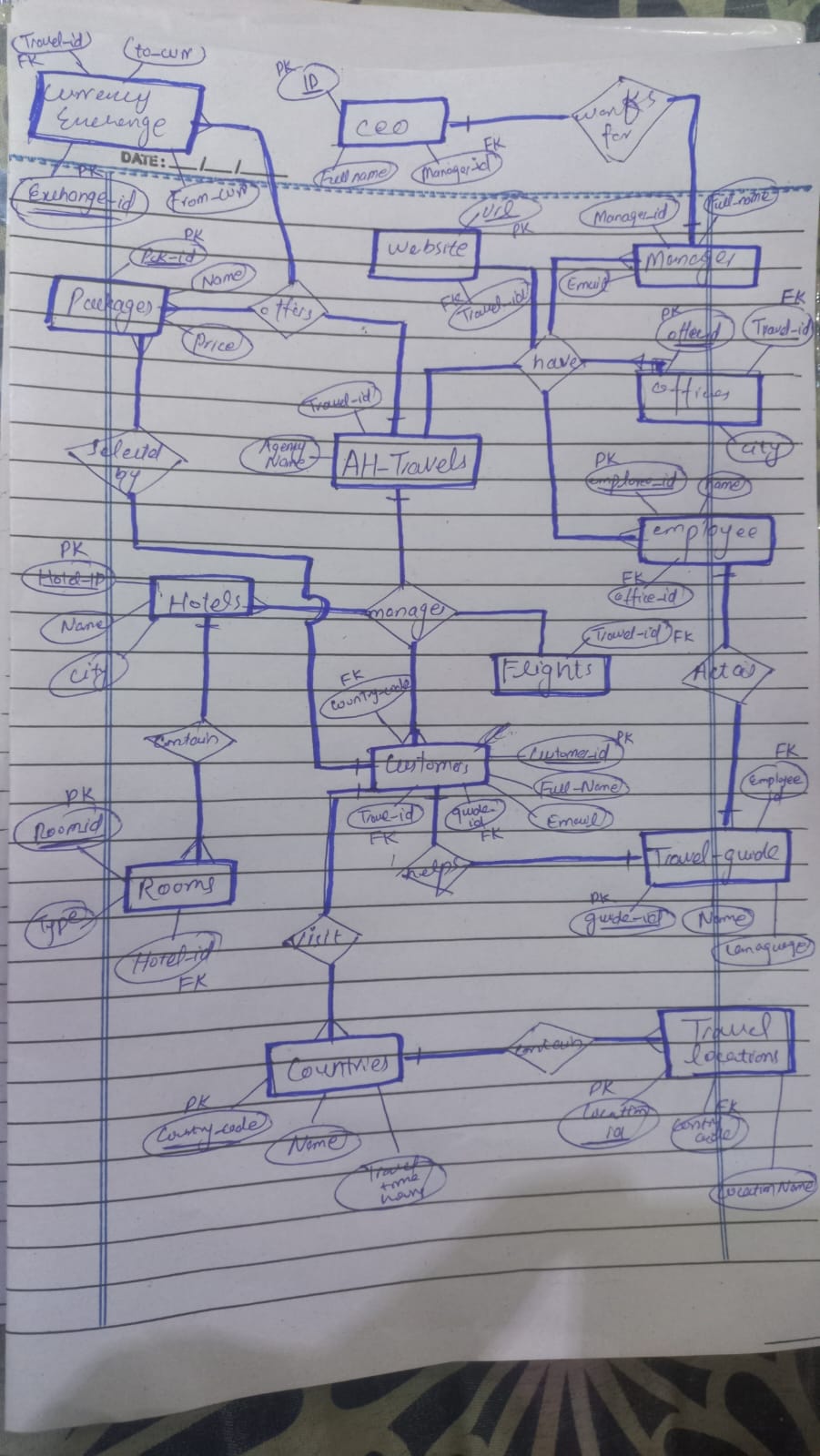
**ROLL-1: 241792**

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

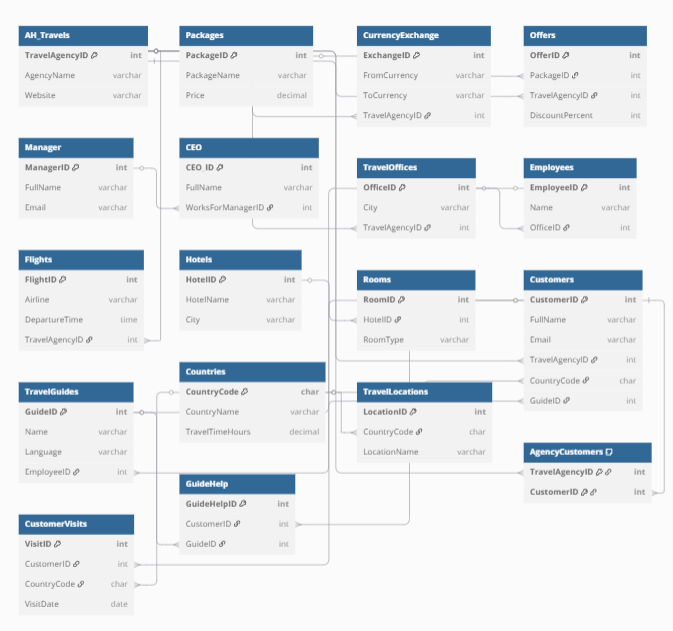
**ROLL-2:241894**

**Project: AH Travels – Travel Agency Database**

**ERD:**

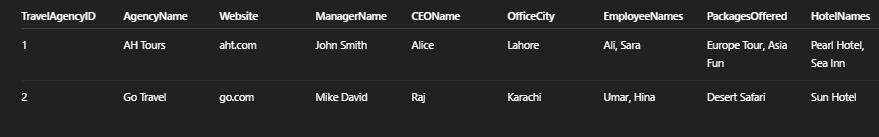


**PHYSICAL SCHEMA:**



Normalization:

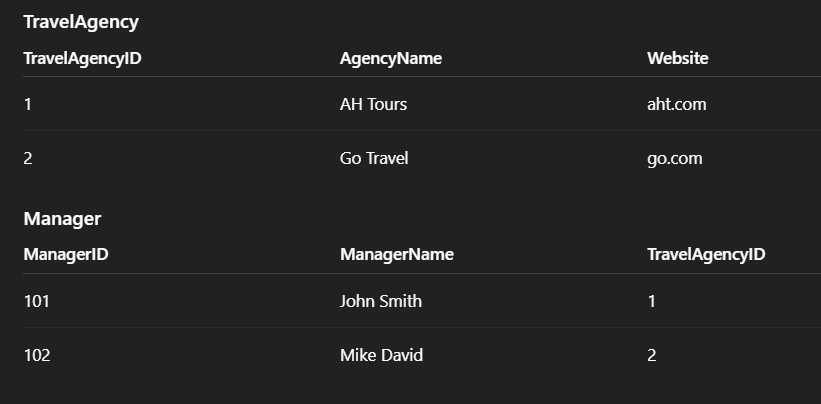
**Unnormalized Table: TravelData\_UNF**



A screenshot of a computer

AI-generated content may be incorrect.

* Repeating groups: EmployeeNames, PackagesOffered, HotelNames, RoomTypes, CustomerNames, GuideNames, LocationsVisited
* **🔹 Step 1: 1NF (Eliminate Repeating Groups)**
* Split repeating groups into separate rows.



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a black screen

AI-generated content may be incorrect.

A screenshot of a black screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a black screen

AI-generated content may be incorrect.

**🔹 Step 2: 2NF (Remove Partial Dependencies)**

All non-key attributes must depend on the whole primary key. Since we’ve already created separate tables for multivalued data, and each table now has a single-column primary key, 2NF is achieved.

✅ Already in 2NF.

**🔹 Step 3: 3NF (Remove Transitive Dependencies)**

Now, we remove transitive dependencies (i.e., non-key → non-key dependencies).

Let’s check:

* In Employees, Name depends only on EmployeeID ✅
* In Rooms, RoomType depends only on RoomID, not HotelName ✅
* In CEO, CEOName depends only on CEO\_ID, not Manager ✅

✅ All tables are in 3NF.

**📘 Schema Design Documentation**

**✅ Objective:**

To build a normalized, relational database schema that captures all business aspects of a travel agency including travel packages, customer data, travel guides, hotels, flights, currency exchange, and staff management.

**🧱 Core Design Principles:**

* **Normalization:** Applied normalization up to **Third Normal Form (3NF)** to eliminate redundancy.
* **Referential Integrity:** Used foreign keys to maintain consistency between related tables.
* **Scalability:** Tables are designed to be easily extendable for future attributes or entities.
* **Modularity:** Separate entities for atomic data and junction tables for many-to-many relationships.

**📦 Tables and Design Decisions:**

**1. AH\_Travels**

* **Purpose:** Represents individual travel agencies.
* **Key Fields:** TravelAgencyID, AgencyName, Website
* **Assumption:** Each travel agency may operate independently with its own website and packages.

**2. Packages**

* **Purpose:** Travel packages offered.
* **Key Fields:** PackageID, PackageName, Price
* **Assumption:** Packages are standalone and may be linked via offers to agencies.

**3. Offers**

* **Purpose:** Connects Packages with Travel Agencies offering them.
* **Design:** Many-to-many resolved using Offers table.
* **Assumption:** An agency can offer multiple packages, and a package can be offered by multiple agencies.

**4. CurrencyExchange**

* **Purpose:** Track exchange rates between currencies.
* **Key Fields:** FromCurrency, ToCurrency, ExchangeID
* **Assumption:** Linked to AH\_Travels to support agencies operating across borders.

**5. Manager and CEO**

* **Purpose:** Represent agency leadership hierarchy.
* **Assumption:** Each CEO reports to a manager; one manager can have multiple CEOs (future extensibility).

**6. TravelOffices**

* **Purpose:** Represents branches in various cities.
* **Assumption:** One agency may have offices in multiple cities.

**7. Employees**

* **Purpose:** Staff working in travel offices.
* **Assumption:** Each employee belongs to a specific office.

**8. Flights**

* **Purpose:** Flights associated with a travel agency.
* **Assumption:** Each agency can manage multiple flights (optional airline partnerships).

**9. Hotels and Rooms**

* **Purpose:** Accommodation database.
* **Assumption:** One-to-many relationship — a hotel has multiple room types.

**10. Customers**

* **Purpose:** End users of travel services.
* **Assumption:** Customers are managed by a travel agency and may be guided by a travel guide.

**11. TravelGuides**

* **Purpose:** Guides who assist customers.
* **Assumption:** Each guide is also an employee and can assist multiple customers.

**12. Countries and TravelLocations**

* **Purpose:** To map countries and tourist destinations.
* **Assumption:** Each location belongs to a specific country.

**🔄 Junction/Bridge Tables**

**13. AgencyCustomers**

* **Purpose:** Link between agencies and customers (if customers book from multiple agencies).
* **Composite Key:** (TravelAgencyID, CustomerID)

**14. CustomerVisits**

* **Purpose:** Track visits made by customers to different countries.
* **Fields:** VisitDate, CountryCode, CustomerID
* **Assumption:** Many-to-many relationship between customers and countries.

**15. GuideHelp**

* **Purpose:** Tracks which guide assisted which customer.
* **Assumption:** A guide can help multiple customers and vice versa.

**🧠 Assumptions Summary**

| **Area** | **Assumption** |
| --- | --- |
| **Currency** | Exchange rates are set per agency, supporting multiple currencies. |
| **Guides** | Each guide is also an employee. |
| **Flights/Hotels** | Operated under AH\_Travels, not external APIs. |
| **CEO-Manager Link** | Each CEO has one manager. |
| **Offers** | Packages are customized per agency. |
| **Customers** | One customer can use services from many guides and agencies. |

**✅ Normalization Summary**

| **Form** | **Description** |
| --- | --- |
| **UNF** | Initial form where customer, package, and location data could repeat. |
| **1NF** | Repeating groups removed, atomic columns ensured. |
| **2NF** | Partial dependencies removed using junction tables like Offers, GuideHelp. |
| **3NF** | Transitive dependencies removed by introducing separate tables for CurrencyExchange, Rooms, Flights. |

**🔗 Entity Relationships Summary**

| **Relationship** | **Type** | **Description** |
| --- | --- | --- |
| TravelAgency ↔ Customers | One-to-Many |  |
| Packages ↔ Offers ↔ TravelAgency | Many-to-Many |  |
| Customers ↔ TravelGuides | Many-to-Many |  |
| Customers ↔ Countries (Visits) | Many-to-Many |  |
| Hotels ↔ Rooms | One-to-Many |  |
| TravelAgency ↔ Offices ↔ Employees | Hierarchical |  |