*A Tour and Travel Management system*

is a comprehensive software solution designed to streamline and automate the operations of travel agencies, tour operators, and travel businesses. It helps manage various aspects of travel planning, booking, and customer service, including:

* **Tour Package Management:** Creating, customizing, and managing tour packages with detailed itineraries, pricing, and inclusions.
* **Booking and Reservation:** Handling online and offline bookings for flights, hotels, transportation, and activities.
* **Customer Relationship Management (CRM):** Storing customer data, managing inquiries, and personalizing travel experiences.
* **Inventory Management:** Tracking availability and managing inventory for flights, hotels, and other travel products.
* **Financial Management:** Processing payments, generating invoices, and managing accounting tasks.
* **Reporting and Analytics:** Generating reports on sales, bookings, and customer behavior to gain insights and improve business decisions.

These systems often include features like online booking portals, mobile apps, and integration with third-party services such as payment gateways and global distribution system (GDS)

*ERDiagram*

CUSTOMER {

int customer\_id PK

string name

string email

string phone

string address

}

TOUR\_PACKAGE {

int package\_id PK

string name

string description

date duration

float price

}

DESTINATION {

int destination\_id PK

string name

string country

string description

}

HOTEL {

int hotel\_id PK

string name

string address

int stars

string contact

}

FLIGHT {

int flight\_id PK

string flight\_number

string airline

date departure\_date

time departure\_time

date arrival\_date

time arrival\_time

}

TRANSPORT {

int transport\_id PK

string type

string vehicle\_number

string driver\_name

}

ACTIVITY {

int activity\_id PK

string name

string description

float price

}

BOOKING {

int booking\_id PK

int customer\_id FK

int package\_id FK

date booking\_date

int number\_of\_travelers

float total\_price

}

PACKAGE\_DESTINATION {

int package\_id FK

int destination\_id FK

}

PACKAGE\_HOTEL {

int package\_id FK

int hotel\_id FK

}

PACKAGE\_FLIGHT {

int package\_id FK

int flight\_id FK

}

PACKAGE\_TRANSPORT {

int package\_id FK

int transport\_id FK

}

PACKAGE\_ACTIVITY {

int package\_id FK

int activity\_id FK

}

CUSTOMER ||--o{ BOOKING : makes

TOUR\_PACKAGE ||--o{ BOOKING : includes

TOUR\_PACKAGE ||--o{ PACKAGE\_DESTINATION : goes to

DESTINATION ||--o{ PACKAGE\_DESTINATION : is in

TOUR\_PACKAGE ||--o{ PACKAGE\_HOTEL : stays at

HOTEL ||--o{ PACKAGE\_HOTEL : is available at

TOUR\_PACKAGE ||--o{ PACKAGE\_FLIGHT : includes flight

FLIGHT ||--o{ PACKAGE\_FLIGHT : is part of

TOUR\_PACKAGE ||--o{ PACKAGE\_TRANSPORT : uses

TRANSPORT ||--o{ PACKAGE\_TRANSPORT : is used for

TOUR\_PACKAGE ||--o{ PACKAGE\_ACTIVITY : includes

ACTIVITY ||--o{ PACKAGE\_ACTIVITY : is part of

**Explanation of Entities and Relationships:**

* **CUSTOMER:** Stores customer information.
* **TOUR\_PACKAGE:** Defines the details of a tour package.
* **DESTINATION:** Represents a place visited on a tour.
* **HOTEL:** Stores information about hotels used in tour packages.
* **FLIGHT:** Contains details about flights included in packages.
* **TRANSPORT:** Represents different modes of transportation (bus, train, etc.).
* **ACTIVITY:** Represents activities included in a tour package (e.g., sightseeing, adventure sports).
* **BOOKING:** Records customer bookings for tour packages.
* **PACKAGE\_DESTINATION:** Links tour packages to destinations. A package can have multiple destinations, and a destination can be part of multiple packages.
* **PACKAGE\_HOTEL:** Links tour packages to hotels. A package can include stays at multiple hotels, and a hotel can be used in multiple packages.
* **PACKAGE\_FLIGHT:** Links tour packages to flights. A package can include multiple flights, and a flight can be part of multiple packages.
* **PACKAGE\_TRANSPORT:** Links tour packages to transportation. A package can use different transport modes, and a transport mode can be used in multiple packages.
* **PACKAGE\_ACTIVITY:** Links tour packages to activities. A package can include multiple activities, and an activity can be part of multiple packages.
* **Data Dictionary for Tour and Travel Management System**
* This data dictionary describes the entities (tables) and their attributes (columns) in a tour and travel management system database.
* **1. CUSTOMER Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| customer\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for the customer. |
| name | VARCHAR(255) | NOT NULL | Full name of the customer. |
| email | VARCHAR(255) | UNIQUE, NOT NULL | Email address of the customer. |
| phone | VARCHAR(20) |  | Phone number of the customer. |
| address | VARCHAR(255) |  | Address of the customer. |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp of customer creation. |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP | Timestamp of last customer update. |

* Export to Sheets
* **2. TOUR\_PACKAGE Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| package\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for the tour package. |
| name | VARCHAR(255) | NOT NULL | Name of the tour package. |
| description | TEXT |  | Detailed description of the tour package. |
| duration | INT |  | Number of days the tour package lasts. |
| price | DECIMAL(10,2) |  | Price of the tour package. |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp of package creation. |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP | Timestamp of last package update. |

* Export to Sheets
* **3. DESTINATION Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| destination\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for the destination. |
| name | VARCHAR(255) | NOT NULL | Name of the destination. |
| country | VARCHAR(255) |  | Country where the destination is located. |
| description | TEXT |  | Description of the destination. |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp of destination creation. |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP | Timestamp of last destination update. |

* Export to Sheets
* **4. HOTEL Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| hotel\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for the hotel. |
| name | VARCHAR(255) | NOT NULL | Name of the hotel. |
| address | VARCHAR(255) |  | Address of the hotel. |
| stars | INT |  | Star rating of the hotel. |
| contact | VARCHAR(255) |  | Contact information for the hotel. |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp of hotel creation. |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP | Timestamp of last hotel update. |

* Export to Sheets
* **5. FLIGHT Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| flight\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for the flight. |
| flight\_number | VARCHAR(20) | NOT NULL | Flight number. |
| airline | VARCHAR(255) |  | Airline name. |
| departure\_date | DATE |  | Departure date. |
| departure\_time | TIME |  | Departure time. |
| arrival\_date | DATE |  | Arrival date. |
| arrival\_time | TIME |  | Arrival time. |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp of flight creation. |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP | Timestamp of last flight update. |

* Export to Sheets
* **6. TRANSPORT Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| transport\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for the transport. |
| type | VARCHAR(255) | NOT NULL | Type of transport (e.g., bus, train, car). |
| vehicle\_number | VARCHAR(20) |  | Vehicle registration number. |
| driver\_name | VARCHAR(255) |  | Name of the driver. |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp of transport creation. |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP | Timestamp of last transport update. |

* Export to Sheets
* **7. ACTIVITY Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| activity\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for the activity. |
| name | VARCHAR(255) | NOT NULL | Name of the activity. |
| description | TEXT |  | Description of the activity. |
| price | DECIMAL(10,2) |  | Price of the activity (if applicable). |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp of activity creation. |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP | Timestamp of last activity update. |

* Export to Sheets
* **8. BOOKING Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| booking\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for the booking. |
| customer\_id | INT | FOREIGN KEY referencing CUSTOMER(customer\_id) | ID of the customer who made the booking. |
| package\_id | INT | FOREIGN KEY referencing TOUR\_PACKAGE(package\_id) | ID of the booked tour package. |
| booking\_date | DATE |  | Date of booking. |
| number\_of\_travelers | INT |  | Number of travelers in the booking. |
| total\_price | DECIMAL(10,2) |  | Total price of the booking. |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp of booking creation. |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP | Timestamp of last booking update. |

* Export to Sheets
* **9. PACKAGE\_DESTINATION Table (Junction Table):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| package\_id | INT | FOREIGN KEY referencing TOUR\_PACKAGE(package\_id) | ID of the tour package. |
| destination\_id | INT | FOREIGN KEY referencing DESTINATION(destination\_id) | ID of the destination. |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp of record creation. |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP | Timestamp of last record update. |

* Export to Sheets
* **10. PACKAGE\_HOTEL Table (Junction Table):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| package\_id | INT | FOREIGN KEY referencing TOUR\_PACKAGE(package\_id) | ID of the tour package. |
| hotel\_id | INT | FOREIGN KEY referencing HOTEL(hotel\_id) | ID of the hotel. |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp of record creation. |
| updated\_at | TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP | Timestamp of last record update. |

* Export to Sheets
* **11. PACKAGE\_FLIGHT Table (Junction Table):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| package\_id | INT | FOREIGN KEY referencing TOUR\_PACKAGE(package\_id) | ID of the tour package. |
| flight\_id | INT | FOREIGN KEY referencing FLIGHT(flight\_id) | ID of the flight. |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP | Timestamp of record creation. |
|  |  |  |  |

* *A logical data model for a tour and travel management system*
* focuses on the entities and their relationships, without getting into the specifics of database implementation. Here's a representation using a common notation (Chen's notation is used here, but others like Crow's Foot are also common):
* **Entities and Attributes:**
* **Customer:** CustomerID, Name, Email, Phone, Address, CreatedAt, UpdatedAt
* **TourPackage:** PackageID, Name, Description, Duration, Price, CreatedAt, UpdatedAt
* **Destination:** DestinationID, Name, Country, Description, CreatedAt, UpdatedAt
* **Hotel:** HotelID, Name, Address, Stars, Contact, CreatedAt, UpdatedAt
* **Flight:** FlightID, FlightNumber, Airline, DepartureDate, DepartureTime, ArrivalDate, ArrivalTime, CreatedAt, UpdatedAt
* **Transport:** TransportID, Type, VehicleNumber, DriverName, CreatedAt, UpdatedAt
* **Activity:** ActivityID, Name, Description, Price, CreatedAt, UpdatedAt
* **Booking:** BookingID, BookingDate, NumberOfTravelers, TotalPrice, CreatedAt, UpdatedAt
* **Relationships:**
* **Customer 1:N Booking:** A customer can make multiple bookings.
* **TourPackage 1:N Booking:** A tour package can have multiple bookings.
* **TourPackage N:M Destination:** A tour package can include multiple destinations, and a destination can be part of multiple tour packages. (This is resolved with a junction table - see below)
* **TourPackage N:M Hotel:** A tour package can include stays at multiple hotels, and a hotel can be part of multiple tour packages. (Junction table)
* **TourPackage N:M Flight:** A tour package can include multiple flights, and a flight can be part of multiple tour packages. (Junction table)
* **TourPackage N:M Transport:** A tour package can use multiple forms of transport, and a transport type can be used in multiple tour packages. (Junction table)
* **TourPackage N:M Activity:** A tour package can include multiple activities, and an activity can be part of multiple tour packages. (Junction table)
* **Junction Tables (to resolve N:M relationships):**
* **PackageDestination:** PackageID, DestinationID, CreatedAt, UpdatedAt
* **PackageHotel:** PackageID, HotelID, CreatedAt, UpdatedAt
* **PackageFlight:** PackageID, FlightID, CreatedAt, UpdatedAt
* **PackageTransport:** PackageID, TransportID, CreatedAt, UpdatedAt
* **PackageActivity:** PackageID, ActivityID, CreatedAt, UpdatedAt
* **Diagrammatic Representation (Simplified):**
* Customer \*--1 Booking
* TourPackage \*--1 Booking
* TourPackage \*--\* Destination (via PackageDestination)
* TourPackage \*--\* Hotel (via PackageHotel)
* TourPackage \*--\* Flight (via PackageFlight)
* TourPackage \*--\* Transport (via PackageTransport)
* TourPackage \*--\* Activity (via PackageActivity)
* **Explanation and Key Improvements over a purely Entity-Relationship Diagram:**
* **Focus on Business Concepts:** The logical model emphasizes the business entities and their relationships in a more conceptual way. It's less concerned with technical database details.
* **N:M Relationships Resolved:** The logical model clearly shows the many-to-many relationships and how they are resolved using junction tables. This is crucial for database design.
* **Data Integrity:** The relationships and attributes defined in the logical model help ensure data integrity. For example, the 1:N relationship between Customer and Booking ensures that a booking is always associated with a customer.
* **Basis for Physical Design:** The logical data model serves as the foundation for the physical database design. It guides the creation of tables, columns, and constraints in the database.
* **Example of a Junction Table (PackageDestination):**
* The PackageDestination table would have PackageID and DestinationID as foreign keys, referencing the TourPackage and Destination tables, respectively. This allows you to represent the fact that a single tour package can visit multiple destinations, and a single destination can be part of multiple tour packages. Each row in the PackageDestination table represents a specific association between a package and a destination.
* This logical data model provides a clear and concise representation of the data requirements for a tour and travel management system. It's a valuable tool for communication between stakeholders and database designers. It's then translated into a physical data model (specific database tables, data types, keys, etc.) for implementation.

phyisical data module for tour and travelling managment

SQL

-- Customer Table

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Email VARCHAR(255) UNIQUE NOT NULL,

Phone VARCHAR(20),

Address VARCHAR(255),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- TourPackage Table

CREATE TABLE TourPackage (

PackageID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Description TEXT,

Duration INT, -- Number of days

Price DECIMAL(10, 2),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Destination Table

CREATE TABLE Destination (

DestinationID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Country VARCHAR(255),

Description TEXT,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Hotel Table

CREATE TABLE Hotel (

HotelID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Address VARCHAR(255),

Stars INT,

Contact VARCHAR(255),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Flight Table

CREATE TABLE Flight (

FlightID INT PRIMARY KEY AUTO\_INCREMENT,

FlightNumber VARCHAR(20) NOT NULL,

Airline VARCHAR(255),

DepartureDate DATE,

DepartureTime TIME,

ArrivalDate DATE,

ArrivalTime TIME,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Transport Table

CREATE TABLE Transport (

TransportID INT PRIMARY KEY AUTO\_INCREMENT,

Type VARCHAR(255) NOT NULL, -- e.g., Bus, Train, Car

VehicleNumber VARCHAR(20),

DriverName VARCHAR(255),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Activity Table

CREATE TABLE Activity (

ActivityID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Description TEXT,

Price DECIMAL(10, 2),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Booking Table

CREATE TABLE Booking (

BookingID INT PRIMARY KEY AUTO\_INCREMENT,

CustomerID INT NOT NULL,

PackageID INT NOT NULL,

BookingDate DATE,

NumberOfTravelers INT,

TotalPrice DECIMAL(10, 2),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID)

);

-- PackageDestination (Junction Table)

CREATE TABLE PackageDestination (

PackageID INT NOT NULL,

DestinationID INT NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (PackageID, DestinationID), -- Composite Key

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID),

FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)

);

-- PackageHotel (Junction Table)

CREATE TABLE PackageHotel (

PackageID INT NOT NULL,

HotelID INT NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (PackageID, HotelID), -- Composite Key

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID),

FOREIGN KEY (HotelID) REFERENCES Hotel(HotelID)

);

-- PackageFlight (Junction Table)

CREATE TABLE PackageFlight (

PackageID INT NOT NULL,

FlightID INT NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (PackageID, FlightID), -- Composite Key

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID),

FOREIGN KEY (FlightID) REFERENCES Flight(FlightID)

);

-- PackageTransport (Junction Table)

CREATE TABLE PackageTransport (

PackageID INT NOT NULL,

TransportID INT NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (PackageID, TransportID), -- Composite Key

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID),

FOREIGN KEY (TransportID) REFERENCES Transport(TransportID)

);

-- PackageActivity (Junction Table)

CREATE TABLE PackageActivity (

PackageID INT NOT NULL,

ActivityID INT NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (PackageID, ActivityID), -- Composite Key

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID),

FOREIGN KEY (ActivityID) REFERENCES Activity(ActivityID)

*Phyisical Data model for Tour and travelling Managment*

SQL

-- Customer Table

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Email VARCHAR(255) UNIQUE NOT NULL,

Phone VARCHAR(20),

Address VARCHAR(255),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- TourPackage Table

CREATE TABLE TourPackage (

PackageID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Description TEXT,

Duration INT, -- Number of days

Price DECIMAL(10, 2),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Destination Table

CREATE TABLE Destination (

DestinationID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Country VARCHAR(255),

Description TEXT,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Hotel Table

CREATE TABLE Hotel (

HotelID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Address VARCHAR(255),

Stars INT,

Contact VARCHAR(255),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Flight Table

CREATE TABLE Flight (

FlightID INT PRIMARY KEY AUTO\_INCREMENT,

FlightNumber VARCHAR(20) NOT NULL,

Airline VARCHAR(255),

DepartureDate DATE,

DepartureTime TIME,

ArrivalDate DATE,

ArrivalTime TIME,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Transport Table

CREATE TABLE Transport (

TransportID INT PRIMARY KEY AUTO\_INCREMENT,

Type VARCHAR(255) NOT NULL, -- e.g., Bus, Train, Car

VehicleNumber VARCHAR(20),

DriverName VARCHAR(255),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Activity Table

CREATE TABLE Activity (

ActivityID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Description TEXT,

Price DECIMAL(10, 2),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

-- Booking Table

CREATE TABLE Booking (

BookingID INT PRIMARY KEY AUTO\_INCREMENT,

CustomerID INT NOT NULL,

PackageID INT NOT NULL,

BookingDate DATE,

NumberOfTravelers INT,

TotalPrice DECIMAL(10, 2),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID)

);

-- PackageDestination (Junction Table)

CREATE TABLE PackageDestination (

PackageID INT NOT NULL,

DestinationID INT NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (PackageID, DestinationID), -- Composite Key

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID),

FOREIGN KEY (DestinationID) REFERENCES Destination(DestinationID)

);

-- PackageHotel (Junction Table)

CREATE TABLE PackageHotel (

PackageID INT NOT NULL,

HotelID INT NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (PackageID, HotelID), -- Composite Key

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID),

FOREIGN KEY (HotelID) REFERENCES Hotel(HotelID)

);

-- PackageFlight (Junction Table)

CREATE TABLE PackageFlight (

PackageID INT NOT NULL,

FlightID INT NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (PackageID, FlightID), -- Composite Key

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID),

FOREIGN KEY (FlightID) REFERENCES Flight(FlightID)

);

-- PackageTransport (Junction Table)

CREATE TABLE PackageTransport (

PackageID INT NOT NULL,

TransportID INT NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (PackageID, TransportID), -- Composite Key

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID),

FOREIGN KEY (TransportID) REFERENCES Transport(TransportID)

);

-- PackageActivity (Junction Table)

CREATE TABLE PackageActivity (

PackageID INT NOT NULL,

ActivityID INT NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (PackageID, ActivityID), -- Composite Key

FOREIGN KEY (PackageID) REFERENCES TourPackage(PackageID),

FOREIGN KEY (ActivityID) REFERENCES Activity(ActivityID)

);