Table Customers { CustomerID integer [primary key] Name varchar Email varchar Phone varchar Address text CreatedAt timestamp } Table TourPackages { PackageID integer [primary key] PackageName varchar Source varchar Destination varchar Duration integer [note: 'In days'] Cost decimal Description text CreatedAt timestamp } Table Bookings { BookingID integer [primary key] CustomerID integer [not null, ref: > Customers.CustomerID] PackageID integer [not null, ref: > TourPackages.PackageID] BookingDate timestamp NoOfPersons integer TotalCost decimal Status varchar [note: 'Pending, Confirmed, Cancelled'] } Table Payments { PaymentID integer [primary key] BookingID integer [not null, ref: > Bookings.BookingID] PaymentDate timestamp Amount decimal PaymentMode varchar [note: 'UPI, Card, Netbanking'] PaymentStatus varchar [note: 'Success, Failed, Pending'] } Table Guides { GuideID integer [primary key] Name varchar Contact varchar Language varchar Experience integer [note: 'Years of experience'] } Table Hotels { HotelID integer [primary key] HotelName varchar Location varchar Rating float ContactInfo varchar } Table Transports { TransportID integer [primary key] VehicleType varchar [note: 'Bus, Van, Car'] Capacity integer Company varchar ContactInfo varchar } Table TourPackageGuides { PackageID integer [ref: > TourPackages.PackageID] GuideID integer [ref: > Guides.GuideID] AssignedDate timestamp } Table TourPackageHotels { PackageID integer [ref: > TourPackages.PackageID] HotelID integer [ref: > Hotels.HotelID] } Table TourPackageTransports { PackageID integer [ref: > TourPackages.PackageID] TransportID integer [ref: > Transports.TransportID] } create sql queries and insert data accordingly

**ChatGPT said:**

Here’s the complete SQL setup based on your schema, including:

1. **Table creation queries**
2. **Insert sample data (minimal but representative)**

**🔸 1. Create Tables**

sql

CopyEdit

CREATE TABLE Customers (

CustomerID INTEGER PRIMARY KEY,

Name VARCHAR(100),

Email VARCHAR(100),

Phone VARCHAR(15),

Address TEXT,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE TourPackages (

PackageID INTEGER PRIMARY KEY,

PackageName VARCHAR(100),

Source VARCHAR(100),

Destination VARCHAR(100),

Duration INTEGER,

Cost DECIMAL(10, 2),

Description TEXT,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE Bookings (

BookingID INTEGER PRIMARY KEY,

CustomerID INTEGER NOT NULL,

PackageID INTEGER NOT NULL,

BookingDate TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

NoOfPersons INTEGER,

TotalCost DECIMAL(10, 2),

Status VARCHAR(20),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (PackageID) REFERENCES TourPackages(PackageID)

);

CREATE TABLE Payments (

PaymentID INTEGER PRIMARY KEY,

BookingID INTEGER NOT NULL,

PaymentDate TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

Amount DECIMAL(10, 2),

PaymentMode VARCHAR(50),

PaymentStatus VARCHAR(20),

FOREIGN KEY (BookingID) REFERENCES Bookings(BookingID)

);

CREATE TABLE Guides (

GuideID INTEGER PRIMARY KEY,

Name VARCHAR(100),

Contact VARCHAR(15),

Language VARCHAR(50),

Experience INTEGER

);

CREATE TABLE Hotels (

HotelID INTEGER PRIMARY KEY,

HotelName VARCHAR(100),

Location VARCHAR(100),

Rating FLOAT,

ContactInfo VARCHAR(100)

);

CREATE TABLE Transports (

TransportID INTEGER PRIMARY KEY,

VehicleType VARCHAR(50),

Capacity INTEGER,

Company VARCHAR(100),

ContactInfo VARCHAR(100)

);

CREATE TABLE TourPackageGuides (

PackageID INTEGER,

GuideID INTEGER,

AssignedDate TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (PackageID) REFERENCES TourPackages(PackageID),

FOREIGN KEY (GuideID) REFERENCES Guides(GuideID)

);

CREATE TABLE TourPackageHotels (

PackageID INTEGER,

HotelID INTEGER,

FOREIGN KEY (PackageID) REFERENCES TourPackages(PackageID),

FOREIGN KEY (HotelID) REFERENCES Hotels(HotelID)

);

CREATE TABLE TourPackageTransports (

PackageID INTEGER,

TransportID INTEGER,

FOREIGN KEY (PackageID) REFERENCES TourPackages(PackageID),

FOREIGN KEY (TransportID) REFERENCES Transports(TransportID)

);

**🔸 2. Insert Sample Data**

sql

CopyEdit

-- Customers

INSERT INTO Customers VALUES

(1, 'Ravi Kumar', 'ravi@example.com', '9876543210', 'Hyderabad', CURRENT\_TIMESTAMP),

(2, 'Meena Shah', 'meena@example.com', '9876543211', 'Chennai', CURRENT\_TIMESTAMP);

-- TourPackages

INSERT INTO TourPackages VALUES

(1, 'Hyderabad to Goa Tour', 'Hyderabad', 'Goa', 5, 15000.00, '5-day Goa beach tour', CURRENT\_TIMESTAMP),

(2, 'Chennai to Munnar Hills', 'Chennai', 'Munnar', 3, 12000.00, 'Munnar hill station trip', CURRENT\_TIMESTAMP);

-- Bookings

INSERT INTO Bookings VALUES

(1, 1, 1, CURRENT\_TIMESTAMP, 2, 30000.00, 'Confirmed'),

(2, 2, 2, CURRENT\_TIMESTAMP, 3, 36000.00, 'Pending');

-- Payments

INSERT INTO Payments VALUES

(1, 1, CURRENT\_TIMESTAMP, 30000.00, 'UPI', 'Success'),

(2, 2, CURRENT\_TIMESTAMP, 10000.00, 'Card', 'Pending');

-- Guides

INSERT INTO Guides VALUES

(1, 'Amit Singh', '9876001234', 'Hindi, English', 5),

(2, 'Latha Nair', '9876005678', 'Malayalam, English', 3);

-- Hotels

INSERT INTO Hotels VALUES

(1, 'Sea View Resort', 'Goa', 4.5, 'sea@example.com'),

(2, 'Hilltop Inn', 'Munnar', 4.2, 'hilltop@example.com');

-- Transports

INSERT INTO Transports VALUES

(1, 'Bus', 40, 'VRL Travels', 'contact@vrl.com'),

(2, 'Car', 4, 'ZoomCar', 'zoom@example.com');

-- TourPackageGuides

INSERT INTO TourPackageGuides VALUES

(1, 1, CURRENT\_TIMESTAMP),

(2, 2, CURRENT\_TIMESTAMP);

-- TourPackageHotels

INSERT INTO TourPackageHotels VALUES

(1, 1),

(2, 2);

-- TourPackageTransports

INSERT INTO TourPackageTransports VALUES

(1, 1),

(2, 2);

**✅ Notes**

* The above inserts are **just sample rows** to validate your schema. You can extend them for real-world data.
* All CreatedAt/BookingDate/PaymentDate values use CURRENT\_TIMESTAMP for simplicity. You can manually enter specific dates if needed.

Would you like:

* ER diagram image based on this schema?
* Sample **queries** like total revenue, most booked packages, etc.?
* Export-ready .sql file?

Let me know!

**You said:**

example queries regarding this system

**ChatGPT said:**

Here are **example SQL queries** for your Tour Package Booking System. These cover practical scenarios like total revenue, booking details, pending payments, and more.

**🔹 1. List All Customers and Their Bookings**

sql

CopyEdit

SELECT

c.CustomerID, c.Name, b.BookingID, b.BookingDate, b.Status

FROM

Customers c

JOIN

Bookings b ON c.CustomerID = b.CustomerID;

**🔹 2. Total Revenue Generated from Successful Payments**

sql

CopyEdit

SELECT

SUM(p.Amount) AS TotalRevenue

FROM

Payments p

WHERE

p.PaymentStatus = 'Success';

**🔹 3. Most Popular Tour Package (by number of bookings)**

sql

CopyEdit

SELECT

tp.PackageName, COUNT(b.BookingID) AS BookingCount

FROM

TourPackages tp

JOIN

Bookings b ON tp.PackageID = b.PackageID

GROUP BY

tp.PackageName

ORDER BY

BookingCount DESC

LIMIT 1;

**🔹 4. All Bookings with Hotel, Guide, and Transport Details**

sql

CopyEdit

SELECT

b.BookingID, c.Name AS CustomerName, tp.PackageName,

h.HotelName, g.Name AS GuideName, t.VehicleType

FROM

Bookings b

JOIN

Customers c ON b.CustomerID = c.CustomerID

JOIN

TourPackages tp ON b.PackageID = tp.PackageID

LEFT JOIN

TourPackageHotels tph ON tp.PackageID = tph.PackageID

LEFT JOIN

Hotels h ON tph.HotelID = h.HotelID

LEFT JOIN

TourPackageGuides tpg ON tp.PackageID = tpg.PackageID

LEFT JOIN

Guides g ON tpg.GuideID = g.GuideID

LEFT JOIN

TourPackageTransports tpt ON tp.PackageID = tpt.PackageID

LEFT JOIN

Transports t ON tpt.TransportID = t.TransportID;

**🔹 5. Customers with Pending or Failed Payments**

sql

CopyEdit

SELECT

c.Name, b.BookingID, p.Amount, p.PaymentStatus

FROM

Payments p

JOIN

Bookings b ON p.BookingID = b.BookingID

JOIN

Customers c ON b.CustomerID = c.CustomerID

WHERE

p.PaymentStatus IN ('Pending', 'Failed');

**🔹 6. Guides Assigned to More Than One Tour Package**

sql

CopyEdit

SELECT

g.GuideID, g.Name, COUNT(tpg.PackageID) AS PackageCount

FROM

Guides g

JOIN

TourPackageGuides tpg ON g.GuideID = tpg.GuideID

GROUP BY

g.GuideID, g.Name

HAVING

COUNT(tpg.PackageID) > 1;

**🔹 7. Tour Packages Without Any Bookings**

sql

CopyEdit

SELECT

tp.PackageID, tp.PackageName

FROM

TourPackages tp

LEFT JOIN

Bookings b ON tp.PackageID = b.PackageID

WHERE

b.BookingID IS NULL;

**🔹 8. List of Tour Packages with Total Earnings**

sql

CopyEdit

SELECT

tp.PackageName,

SUM(p.Amount) AS TotalEarnings

FROM

TourPackages tp

JOIN

Bookings b ON tp.PackageID = b.PackageID

JOIN

Payments p ON b.BookingID = p.BookingID

WHERE

p.PaymentStatus = 'Success'

GROUP BY

tp.PackageName;

**🔹 9. Details of Bookings Made in the Last 7 Days**

sql

CopyEdit

SELECT

b.BookingID, c.Name, tp.PackageName, b.BookingDate

FROM

Bookings b

JOIN

Customers c ON b.CustomerID = c.CustomerID

JOIN

TourPackages tp ON b.PackageID = tp.PackageID

WHERE

b.BookingDate >= DATE\_SUB(CURRENT\_DATE, INTERVAL 7 DAY);

**🔹 10. Transport Usage Frequency (by package count)**

sql

CopyEdit

SELECT

t.VehicleType, COUNT(tpt.PackageID) AS UsedInPackages

FROM

Transports t

JOIN

TourPackageTransports tpt ON t.TransportID = tpt.TransportID

GROUP BY

t.VehicleType

ORDER BY

UsedInPackages DESC;

Let me know if you want any of the following:

* Visual **ER Diagram** of this system
* Queries specific to date ranges or customer IDs
* **Stored procedures** or **triggers** for automation
* Export-ready .sql file with schema + data + queries

I'm happy to help further!

Top of Form

Bottom of Form