

University Of Mauritius

Faculty of Information, Communication and Digital Technologies

Travel Agency

Prepared by team SQL-icious :

Student Name	Student ID
RAGHOONUNDUN Rishab	2412024
THUMMANAH Kentish	2411831
SOOBRAYEN Keshia	2412920
BEEDASSY Nirvana Luxmi	2413850
AUCKEL Ayush Manav	2412549






Course :	BSc (Hons) Computer Science – level 1
Module :	ICDT1202Y Database Systems
Assignment :	Mini Project
Lecturer Name :	Mrs. Sudha Cheerkoot-Jalim

Date : 11th March 2025

Table of Contents

Distribution Of Tasks	2
1.0 Project Overview	3
2.0 Database Description and Structure	3
2.1 Database Structure	3
2.2 Table Description	4
2.3 Assumptions	5
3.0 Entity Relation Diagram	6
4.0 Normalisation	7
5.0 Functional Dependencies	8
6.0 Database Schema	9
7.0 Database Design	10
8.0 Insert Data Set	12
9.0 Stored Procedures	16
10.0 Triggers	18
11.0 Security Measures using DCL	20

Distribution Of Tasks

Student ID	Student Name	Task	Signature
2412024	RAGHOONUNDUN Rishab	Project Overview ,Stored Procedure, Trigger, ERD, Functional Dependencies	
2411831	THUMMANAH Kentish	Project Overview, Database Description and Structure, Normalisation, Insert Data Set, Functional Dependencies	
2412920	SOOBRAYEN Keshia	Project Overview ,Stored Procedure, Trigger, Database Design, DCL	
2413850	BEEDASSY Nirvana Luxmi	Project Overview ,Stored Procedure, Database Schema, DCL, Report Compilation	
2412549	AUCKEL Ayush Manav	Project Overview	

Note : The diagrams were designed by using draw.io

1.0 Project Overview

In the travel agency business, efficiently managing customer bookings, travel packages, and financial transactions is essential for smooth operations. Our project aims to design and implement a well-structured database system for a Travel Agency using SQL Server to ensure seamless data management.

The system will focus on handling key aspects such as customer information, travel packages, reservations, payments, and trip itineraries. By organizing the data effectively, we will improve accessibility and accuracy, ensuring a better experience for both the agency and its clients.

To achieve this, we will create an Entity Relationship Diagram (ERD) to visualize the database structure and apply normalization up to Third Normal Form (3NF) to eliminate data redundancy.

Additionally, we will implement stored procedures to automate key tasks such as booking confirmations, and trip cancellations. Triggers will be used to enforce data integrity. To enhance security, Data Control Language (DCL) measures will be applied, ensuring that only authorized users can access or modify sensitive information.

By implementing this database system, our goal is to streamline the travel agency's operations, improve customer service, and ensure efficient data management for future scalability.

2.0 Database Description and Structure

2.1 Database Structure

The Epic Explorers Travel Agency database system consists of multiple interconnected tables, each playing a crucial role in managing different aspects of the travel business. Although the database model primarily centers on key entities related to customer service and operations, it has been optimized to include 7 essential tables.

The tables are:

1. Customers : Stores customer details.
2. Booking : Stores trip booking details.
3. Travel_Packages : Contains information about available packages.
4. Destinations : Stores details of destinations offered by the agency.

5. Guide : Stores guide details offered by the agency.
6. Flight : Stores flight details for travel packages.
7. Hotel : Stores hotel details for travel packages.

Each table is connected using primary keys (PK) and foreign keys (FK) to ensure data consistency and minimize redundancy.

2.2 Table Description

1. Customers

The Customer table stores essential information about clients, ensuring efficient management. The attributes are **CustID, Fname, Lname, DOB, Email, Address, PassportNum.**

2. Booking

The Booking table stores information about trip bookings made by customers. The attributes are **BookID, CustID, PackageID, BDate, TotalPrice.**

3. Travel_Packages

The Travel_Packages table stores details about available travel packages. The attributes are **PackageID, PackageName, PackagePrice, StartDate, EndDate ,PDuration.**

4. Destinations

The Destinations table stores information about destinations offered by the agency. The attributes are **DestID, DestName, Country, Desc.**

5. Guide

The Guide tables stores information about guides offered by the agency. The attributes are **GuideID, BookID, GuideName, CostPerDay, PhoneNumber.**

6. Flight

The Flight table stores flight details if applicable. The attributes are **FlightID, Airline, DepartureTime, ArrivalTime, FDuration, Status, FPrice.**

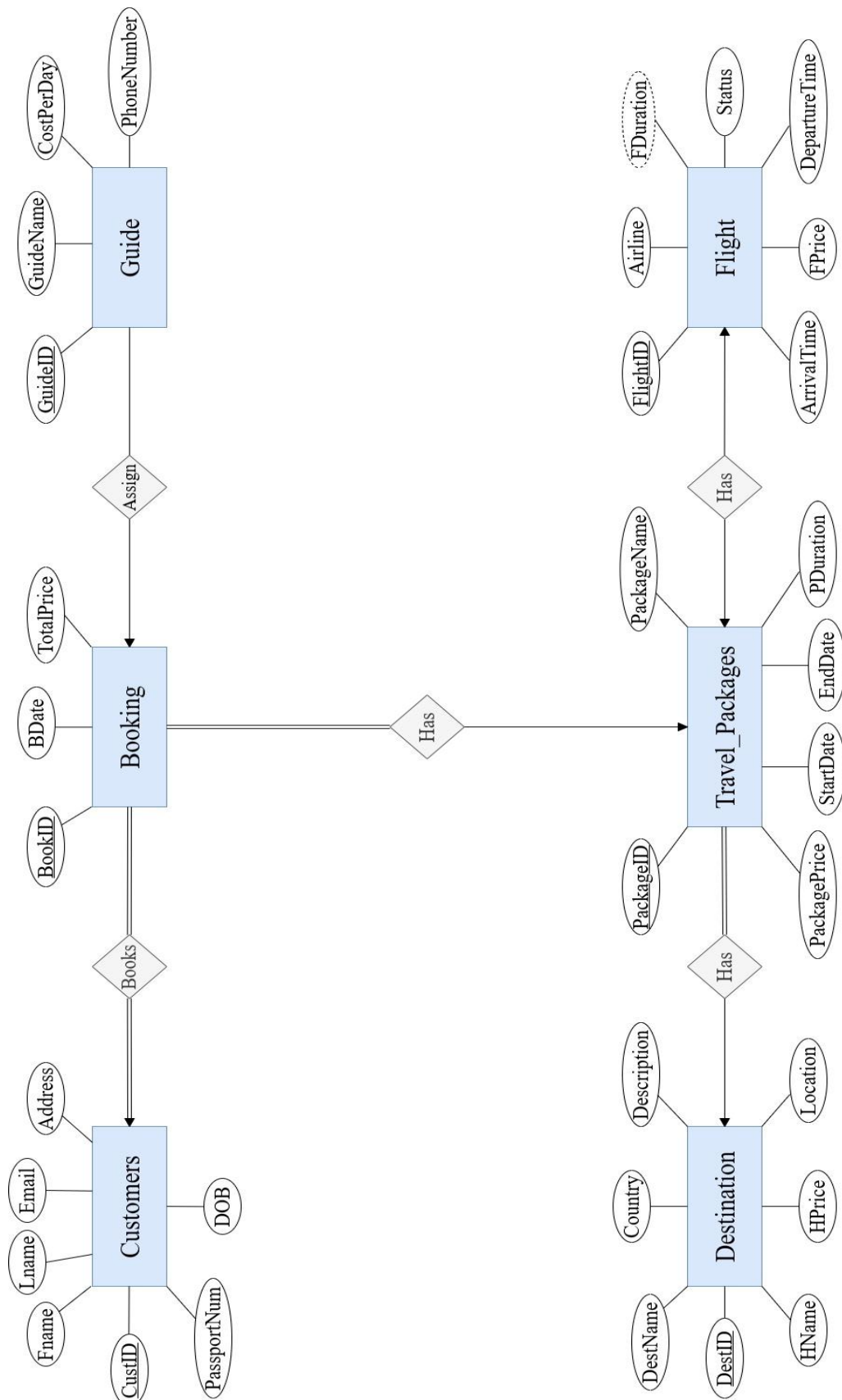
7. Hotel

The hotel table stores details about hotels offered by the agency. The attributes are **HotelID, HName, HPrice, Location, DestID.**

2.3 Assumptions

1. Customers must be at least 18 years old to book a travel package and they must provide a valid passport or national identity card when booking trips. The system will validate document authenticity before confirming the reservation.
2. Each booking must be associated with a valid customer. Customers cannot make bookings without registering in the system.
3. The total price of a booking includes the travel package price and any additional services such as flights, hotels and guides. Also, payments must be made in full before the booking is confirmed. No installment payments are allowed.
4. Travel packages have fixed durations and prices(\$), which cannot be changed by customers.
5. Flights are booked as part of a travel package, meaning customers cannot book flights separately through the system.
6. Each travel package is linked to a specific destination. A destination may have multiple travel packages, but each package includes a predefined set of services (e.g., flights, hotel, guide).
7. Hotels are assigned based on the selected travel package. Customers cannot choose their own hotel; instead, they receive accommodations included in their package.
8. Guides are optional. Customers can request a guide, but the system does not guarantee one unless confirmed.
9. Flight details, including departure, arrival times and status are predefined in the system and cannot be changed by customers.

3.0 Entity Relation Diagram



4.0 Normalisation

0NF:

Travel_Agency {CustID, Fname, Lname, DOB, Email, Address, PassportNum, BookID, BDate, TotalPrice, GuideID, GuideName, CostPerDay, PhoneNumber, FlightID, Airline, ArrivalTime, FPrice, DepartureTime, Status, FDuration, PackageID, PackageName, PackagePrice, StartDate, EndDate, PDuration, DestID, DestName, Country, Description, HName, HPrice, Location}

1NF: (Identify Primary Keys)

Travel_Agency {CustID, Fname, Lname, DOB, Email, Address, PassportNum, BookID, BDate, TotalPrice, GuideID, GuideName, CostPerDay, PhoneNumber, FlightID, Airline, ArrivalTime, FPrice, DepartureTime, FDuration, Status, PackageID, PackageName, PackagePrice, StartDate, EndDate, PDuration, DestID, DestName, Country, Description, HName, HPrice, Location}

2NF: (Remove Partial Dependencies)

Customers {CustID (PK), Fname, Lname, DOB, Email, Address, PassportNum}

Booking {BookID (PK), BDate, TotalPrice, CustID (FK), PackageID (FK)}

Guide {GuideID (PK), GuideName, CostPerDay, PhoneNumber, BookID (FK)}

Flight {FlightID (PK), Airline, ArrivalTime, FPrice, DepartureTime, FDuration, Status, PackageID (FK)}

Travel_Packages {PackageID (PK), PackageName, PackagePrice, StartDate, EndDate, PDuration, DestID (FK)}

Destination {DestID (PK), DestName, Country, Description, HName, HPrice, Location}

3NF: (Remove Transitive Dependencies)

Customers {CustID (PK), Fname, Lname, DOB, Email, Address, PassportNum}

Booking {BookID (PK), BDate, TotalPrice, CustID (FK), PackageID (FK)}

Guide {GuideID (PK), GuideName, CostPerDay, PhoneNumber, BookID (FK)}

Flight {FlightID (PK), Airline, ArrivalTime, FPrice, DepartureTime, FDuration, Status
PackageID (FK)}

Travel_Packages {PackageID (PK), PackageName, PackagePrice, StartDate, EndDate,
PDuration, DestID (FK)}

Destination {DestID (PK), DestName, Country, Description}

Hotel {HotelID (PK), HName, HPrice, Location, DestID (FK)}

5.0 Functional Dependencies

1. Customers table

CustID \rightarrow {Fname, Lname, DOB, Email, Address, PassportNum}

2. Booking table

BookID \rightarrow {BDate, TotalPrice, CustID, PackageID}

3. Travel Packages table

PackageID \rightarrow {PackageName, PackagePrice, PDuration, DestID, FlightID}

4. Guide table

GuideID \rightarrow {GuideName, PhoneNumber, CostPerDay, BookID}

5. Destination table

DestID \rightarrow {DestName, Country, Desc, Location, HName, Price, FlightID}

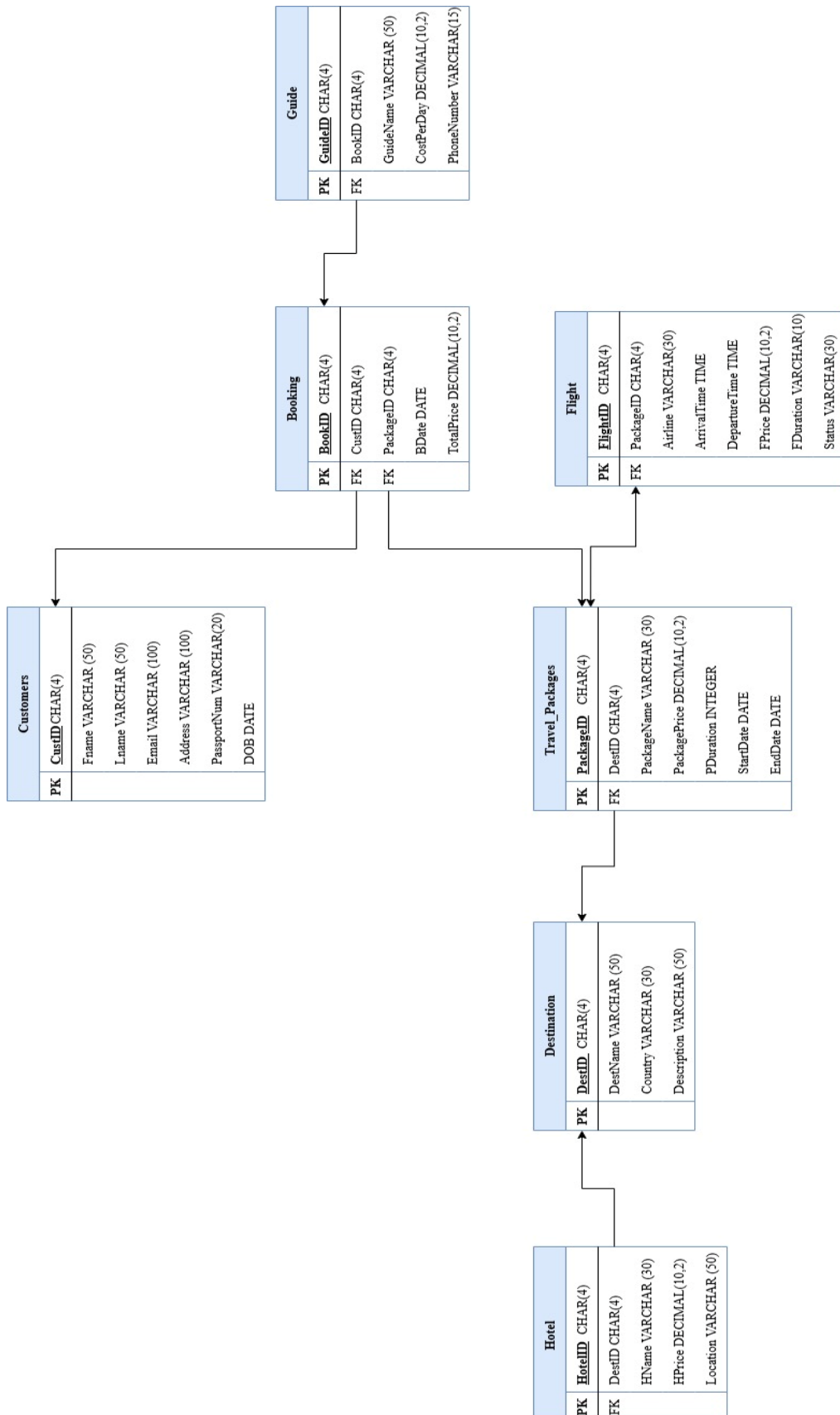
6. Flight table

FlightID \rightarrow {Airline, ArrivalTime, DepartureTime, FDuration, Status, FPrice}

7. Hotel table

HotelID \rightarrow {HName, HPrice, Location, DestID}

6.0 Database Schema



7.0 Database Design

Customers

ATTRIBUTE	DATA TYPE	CONSTRAINT
CustID	CHAR(4)	PRIMARY KEY, Starts with 'C'
Fname	VARCHAR (50)	NOT NULL
Lname	VARCHAR (50)	NOT NULL
DOB	DATE	NOT NULL
Email	VARCHAR (100)	LIKE %@%.%
Address	VARCHAR (100)	NOT NULL
PassportNum	VARCHAR(20)	NOT NULL

Booking

ATTRIBUTE	DATA TYPE	CONSTRAINT
BookID	CHAR(4)	PRIMARY KEY, Starts with 'B'
BDate	DATE	NOT NULL
TotalPrice	DECIMAL(10,2)	
CustID	CHAR(4)	References Customers table Starts with 'C'
PackageID	CHAR(4)	References Travel_Packages table ,Starts with 'P'

Guide

ATTRIBUTE	DATA TYPE	CONSTRAINT
GuideID	CHAR(4)	PRIMARY KEY, Starts with 'G'
GuideName	VARCHAR (50)	NOT NULL
CostPerDay	DECIMAL(10,2)	NOT NULL
PhoneNumber	VARCHAR(15)	NOT NULL
BookID	CHAR(4)	References Booking table, Starts with 'B'

Flight

ATTRIBUTE	DATA TYPE	CONSTRAINT
FlightID	CHAR(4)	PRIMARY KEY, Starts with 'F'
Airline	VARCHAR (30)	NOT NULL
ArrivalTime	TIME	NOT NULL
FPrice	DECIMAL(10,2)	NOT NULL
DepartureTime	TIME	NOT NULL
FDuration	VARCHAR(10)	LIKE %h %m
PackageID	CHAR(4)	References Travel_Packages table ,Starts with 'P'

Travel_Packages

ATTRIBUTE	DATA TYPE	CONSTRAINT
PackageID	CHAR(4)	PRIMARY KEY, Starts with 'P'
PackageName	VARCHAR (30)	NOT NULL
PackagePrice	DECIMAL(10,2)	NOT NULL
StartDate	DATE	NOT NULL
EndDate	DATE	NOT NULL
PDuration	INTEGER	NOT NULL
DestID	CHAR(4)	References Destination table ,Starts with 'D'

Destination

ATTRIBUTE	DATA TYPE	CONSTRAINT
DestID	CHAR(4)	PRIMARY KEY, Starts with 'D'
DestName	VARCHAR (50)	NOT NULL
Country	VARCHAR (30)	NOT NULL
Description	VARCHAR (50)	

Hotel

ATTRIBUTE	DATA TYPE	CONSTRAINT
HotelID	CHAR(4)	PRIMARY KEY, Starts with 'H'
HName	VARCHAR (30)	NOT NULL
HPrice	DECIMAL(10,2)	NOT NULL
Location	VARCHAR (50)	
DestID	CHAR(4)	References Destination table, Starts with 'D'

8.0 Insert Data Set

Customers

CustID	Fname	Lname	DOB	Email	Address	PassportNum
C001	John	Doe	1999-05-15	johndoe@example.com	123 Maple St, NY	AB123456
C002	Jane	Smith	1988-08-22	janesmith@example.com	456 Oak St, LA	CD789012
C003	Alice	Johnson	2000-03-10	alicej@example.com	789 Pine St, TX	EF345678
C004	Robert	Brown	1980-11-30	robbrown@example.com	321 Birch St, FL	GH901234
C005	Emma	Davis	2001-07-18	emmad@example.com	654 Cedar St, IL	IJ567890
C006	Michael	Wilson	1985-09-25	mikewilson@example.com	987 Spruce St	KL345678
C007	Sophia	Martinez	1997-02-14	sophiam@example.com	852 Elm St, WA	MN901234
C008	Daniel	Taylor	2004-06-08	dantaylor@example.com	963 Walnut St, TX	OP567890
C009	Olivia	Anderson	1991-04-03	oliviana@example.com	741 Ash St, GA	QR345678
C010	Ethan	Thomas	1989-12-20	ethant@example.com	159 Fir St, CA	ST901234

Destination

DestID	DestName	Country	Description
D001	Eiffel Tower	France	Iconic landmark in Paris
D002	Great Wall	China	Ancient wall with scenic views
D003	Machu Picchu	Peru	Historic Incan city
D004	Statue of Liberty	USA	Famous landmark in New York
D005	Colosseum	Italy	Ancient Roman amphitheater
D006	Sydney Opera House	Australia	Famous performing arts venue
D007	Taj Mahal	India	Marble mausoleum in Agra
D008	Santorini	Greece	Beautiful whitewashed island
D009	Mount Fuji	Japan	Sacred mountain and pilgrimage site
D010	Niagara Falls	Canada	Massive waterfalls on the border

Travel_Packages

PackageID	PackageName	PackagePrice	StartDate	EndDate	PDuration	DestID
P001	Paris Adventure	1500.00	2025-06-01	2025-06-07	7	D001
P002	China Wonders	2000.00	2025-07-10	2025-07-16	6	D002
P003	Incan Escape	1800.00	2025-05-20	2025-05-30	10	D003
P004	NYC Explorer	1700.00	2025-08-05	2025-08-13	8	D004
P005	Roman Holiday	2200.00	2025-09-01	2025-09-12	12	D005
P006	Sydney Highlights	1600.00	2025-06-10	2025-06-24	14	D006
P007	India Royal Tour	2100.00	2025-07-01	2025-07-05	5	D007
P008	Greek Islands	2300.00	2025-10-01	2025-10-07	6	D008
P009	Japan Journey	2500.00	2025-11-01	2025-11-09	9	D009
P010	Canadian Adventure	1900.00	2025-12-01	2025-12-07	6	D010

Hotel

HotelID	HName	HPrice	Location	DestID
H001	Paris Grand	200.00	Near Eiffel Tower	D001
H002	Beijing Royal	180.00	Near Great Wall	D002
H003	Cusco Retreat	150.00	Near Machu Picchu	D003
H004	Liberty Inn	175.00	Near Statue of Liberty	D004
H005	Colosseum Suites	190.00	Near Colosseum	D005
H006	Sydney Harbour	210.00	Near Opera House	D006
H007	Agra Palace	160.00	Near Taj Mahal	D007
H008	Santorini Blue	230.00	Island View	D008
H009	Fuji Heights	250.00	Near Mount Fuji	D009
H010	Falls Lodge	200.00	Near Niagara Falls	D010

Booking

BookID	BDate	TotalPrice	CustID	PackageID
B001	2025-04-01	1700.00	C001	P001
B002	2025-04-05	2200.00	C002	P002
B003	2025-04-10	1900.00	C003	P003
B004	2025-04-15	1800.00	C004	P004
B005	2025-04-20	2500.00	C005	P005
B006	2025-04-25	1600.00	C006	P006
B007	2025-04-30	2300.00	C007	P007
B008	2025-05-05	2400.00	C008	P008
B009	2025-05-10	2600.00	C009	P009
B010	2025-05-15	2000.00	C010	P010

Guide

GuideID	GuideName	CostPerDay	PhoneNumber	BookID
G001	Pierre Dupont	100.00	123-456-7890	B001
G002	Li Wei	120.00	234-567-8901	B002
G003	Carlos Mendez	110.00	345-678-9012	B003
G004	Emma Wilson	105.00	456-789-0123	B004
G005	Luca Romano	130.00	567-890-1234	B005
G006	Sydney Carter	115.00	678-901-2345	B006
G007	Raj Patel	140.00	789-012-3456	B007
G008	Dimitrios Kostas	125.00	890-123-4567	B008
G009	Haruto Tanaka	135.00	901-234-5678	B009
G010	Ethan Williams	110.00	012-345-6789	B010
G011	Rishab Raghoo	120.00	110-543-9876	NULL
G012	Keke Soso	125.00	321-543-1856	NULL
G013	Khushi Bee	115.00	654-597-5269	NULL
G014	Kentish Thum	130.00	167-956-1256	NULL
G015	Ayush Auckel	120.00	678-345-8012	NULL

Flight

FlightID	Airline	ArrivalTime	Fprice	DepartureTime	FDuration	PackageIID	Status
F001	Air France	10:30:00	500.00	05:00:00	5h 30m	P001	Scheduled
F002	China Airlines	14:15:00	700.00	08:00:00	6h 15m	P002	Scheduled
F003	LATAM Airlines	12:45:00	600.00	08:00:00	4h 45m	P003	Scheduled
F004	Delta Airlines	15:20:00	550.00	10:00:00	5h 20m	P004	Scheduled
F005	Alitalia	09:30:00	750.00	05:00:00	4h 30m	P005	Scheduled
F006	Qantas	18:10:00	650.00	12:00:00	6h 10m	P006	Scheduled
F007	Air India	20:45:00	720.00	15:00:00	5h 45m	P007	Scheduled
F008	Aegean Airlines	13:00:00	800.00	06:00:00	7h 00m	P008	Scheduled
F009	Japan Airlines	17:30:00	900.00	11:00:00	6h 30m	P009	Scheduled
F010	Air Canada	22:00:00	620.00	16:00:00	6h 00m	P010	Scheduled

9.0 Stored Procedures

Number	Name	Operation
1	SP_add_customer	Inserts new customer records into the Customers table while automatically generating a unique PK. It also uses the trigger TRG_check_customer_age to check the age of the customer.
2	SP_view_available_guides	Retrieves and display all available guides.
3	SP_assign_guide_to_booking	Assigns an available guide to a new booking. It then checks if a guide is available. If not found, it prints 'No guide available' and exits. If found, it updates the Guide table by assigning the provided Booking ID to the selected guide.
4	SP_create_booking	It creates a new booking by using the current date as the booking date. Then it assigns an available guide using SP_assign_guide_to_booking and calculates the total price using SP_total_price_booking.
5	SP_cancel_booking	The SP_cancel_booking procedure deletes a booking from the Booking table using a given BookID . It first checks if the booking exists; if not, it prints an error message. If found, it deletes the booking .
6	SP_total_price_booking	This procedure calculates the final price for a booking including the guide.
7	SP_customer_budget	This procedure finds travel packages within a customer's budget. It uses a cursor to loop through Travel_Packages and calls SP_price_without_guide to calculate package cost excluding guide. If a guide is available, it calculates the total price.

		Results are stored in #BudgetResults (temporary table).
8	SP_view_travel_package	Retrieves and display all travel packages.
9	SP_view_customer	Retrieves and display all registered customers.
10	SP_delete_customer	The SP_delete_customer procedure deletes a customer from the Customers table based on their ID (@cid). It first checks if the customer exists; if not, it prints an error message. If the customer is found, it deletes the record and confirms the deletion.
11	SP_price_without_guide	This procedure calculates the total package price excluding the guide. It retrieves the hotel price, flight price, and package price.
12	SP_update_flight_status	The SP_update_flight_status procedure updates the status of a flight in the Flight table using a given FlightID. It first checks if the flight exists; if not, it prints an error message and exits. If the flight is found, it updates its status to the new value and confirms the update.
13	SP_view_booking	The SP_view_booking procedure retrieves all the bookings and displays it.

10.0 Triggers

Number	Name	Operation
1	TRG_check_customer_age	It prevents the insertion of customers under 18 years old in the Customers table. It runs instead of an insert operation, first extracting the new customer's ID and DOB from the inserted table. It calculates the customer's age using DATEDIFF() adjusting if their birthday has not occurred yet in the current year. If the age is below 18, it prints an error message and prevents insertion; otherwise, it inserts the new customer's details into the Customers table.
2	TRG_set_BookID_null_in_Guide	This trigger runs instead of a delete operation on the Booking table. Before deleting a booking, it updates the Guide table by setting BookID to NULL wherever the deleted BookID exists, ensuring referential integrity. After this update, it proceeds with deleting the booking from the Booking table. Finally, it prints a confirmation message indicating that BookID was set to NULL in the Guide table before deletion.
3	TRG_delete_customer_booking	Whenever a customer is deleted, this trigger is called to delete the booking corresponding to the deleted customer. Before deleting a customer, it retrieves all their BookID values from the Booking table using a cursor and calls the SP_cancel_booking stored procedure for each booking to ensure proper cancellation. After processing all bookings, it proceeds

		with deleting the customer from the Customers table. This ensures that all related bookings are properly handled before removing the customer.
4	TRG_notify_customer	The TRG_notify_customer trigger runs after an update on the Flight table to notify customers about flight status changes. It checks if a flight is cancelled, delayed, or rescheduled, retrieves affected customers, and displays the appropriate message.

11.0 Security Measures using DCL

Users	Access
TravelAdmin	<p>GRANT statements assign the TravelAdmin user full access (SELECT, INSERT, UPDATE, DELETE) to key tables in the database, including Customers, Destination, Travel_Packages, Hotel, Booking, Guide, and Flight. This means TravelAdmin can view, add, modify, and remove records in these tables, giving them full control over managing customer details, travel destinations, packages, hotel information, bookings, guides, and flight data within the system.</p> <p><u>Access to stored procedures :</u></p> <p>GRANT EXECUTE statements give the TravelAdmin user permission to execute specific stored procedures related to customer and booking management. This allows TravelAdmin to add customers, view available guides, assign guides to bookings, create and cancel bookings, calculate total booking prices, check customer budgets, view travel packages and customer details, delete customers, calculate prices without a guide, and update flight statuses.</p> <p><u>Access to triggers:</u></p> <p>TravelAdmin gains permission to handle TRG_check_customer_age, which prevents underage customers from being added, TRG_set_BookID_null_in_Guide, which updates related records before deleting bookings, and TRG_delete_customer_booking, which ensures customer bookings are properly canceled before deletion.</p>
TravelAgent	TravelAgent can view, insert, and update records in the Booking and Customers tables, allowing them to create

	<p>and modify bookings and customer details. They can view and update Travel_Packages, Hotel, Guide, Flight, and Destination tables, enabling them to update travel package prices, hotel details, guide assignments, flight statuses, and destination information as needed.</p> <p><u>Access to stored procedures :</u></p> <p>GRANT EXECUTE statements allow the TravelAgent user to run specific stored procedures for managing bookings and customer information. TravelAgent can create new bookings (SP_create_booking), view existing ones (SP_view_booking), check available guides (SP_view_available_guides), add new customers (SP_add_customer), and view travel packages (SP_view_travel_package).</p>
CustomerSupport	<p>GRANT SELECT statements allow the CustomerSupport user to view data from the Booking and Customers tables. This means CustomerSupport can access customer details and booking information but cannot modify or delete records.</p> <p><u>Access to stored procedures :</u></p> <p>GRANT EXECUTE statements allow the CustomerSupport user to run specific stored procedures for retrieving customer and booking information. With access to SP_view_booking, they can view booking details, and with SP_view_customer, they can retrieve customer information.</p>
Guest	<p>GRANT SELECT statements allow the Guest user to view data from the Travel_Packages, Hotel, and Flight tables. This means Guest can browse available travel packages, hotels, and flight details but cannot make any changes.</p>

	<p><u>Access to stored procedures :</u></p> <p>GRANT EXECUTE statement allows the Guest user to run the SP_view_travel_package stored procedure, enabling them to retrieve and view details about available travel packages.</p>
guide	<p>GRANT SELECT statements allow the guide user to view data from the Guide,Booking and Travel_Packages tables. This means guide can browse guide details, travel packages and bookings made but cannot make any changes.</p> <p><u>Access to stored procedures :</u></p> <p>GRANT EXECUTE statement allows the Guest user to run the SP_view_travel_package stored procedure, enabling them to retrieve and view details about available travel packages. With access to SP_view_booking, they can view booking details.</p>