



Zaid Fareed | Muhammad Tayyab | Muhammad Umair

23I-0829 | 23I-0636 | 23i-0662

Travel Ease Data Model

Instructor: Ms. Noor-ul-Ain

Course: Database Systems

Course Code: CS2006

Travel Ease Development Docs

The TravelEase platform is designed to be a centralized solution for planning, booking, and managing diverse travel experiences. It brings together travelers, tour operators, hotels, local guides, and transport providers under one unified system. To ensure smooth operations and data integrity, a well-structured and normalized relational database is essential.

This report outlines the database model developed for TravelEase, detailing its core entities, relationships, and design rationale. The model supports key functionalities such as trip customization, booking management, payment tracking, service coordination, and performance reporting. It is implemented using SQL Server and is intended to integrate seamlessly with a C#-based front-end built in Visual Studio.

Database Model

1. Users:

<u>RegNo</u>	Pass_hash	Creat_date	Last_login	Contact_email	Contact_phone	role	User_status
picUrl	Pro_desc						

- **RegNo (VARCHAR(20))** – Unique user ID based on role (TR-XXXXXX, AD-XXXXXX, OP-XXXXXX, SP-XXXXXX).
- **Password_hash (VARCHAR(255))** – Hashed password stored securely using a strong algorithm.
- **Created_date (DATETIME)** – Timestamp of account creation, defaults to GETDATE().
- **Last_login (DATETIME)** – Last successful login timestamp.
- **Contact_email (VARCHAR(100))** – User’s email address, must follow standard email format.
- **Contact_phone (VARCHAR(15))** – Phone number in formats like +92..., 0300..., or 92....
- **User_status (VARCHAR(50))** – Account state: active, inactive, pending, or deleted.
- **User_role (VARCHAR(50))** – Type of user: traveler, admin, tour_operator, or service_provider.
- **User_profile_image (VARCHAR(255))** – Optional path or URL to profile image.
- **User_profile_description (TEXT)** – Optional user bio or description.

2. Travelers:

<u>RegNo</u>	FName	LName	Nationality	CNIC	DOB	
--------------	-------	-------	-------------	------	-----	--

- **reg_no (VARCHAR(20))** – Foreign key linking to users(reg_no), uniquely identifies each traveler.
- **first_name (VARCHAR(50))** – Traveler’s first name, required and alphabetic.
- **last_name (VARCHAR(50))** – Traveler’s last name, required and alphabetic.
- **nationality (VARCHAR(50))** – Country of citizenship or origin, required field.
- **cnic (VARCHAR(15))** – National ID number (numeric only), required for identity verification.
- **date_of_birth (DATE)** – Traveler’s date of birth in YYYY-MM-DD format.

3. Tour Operator:

<u>RegNo</u>	Operator_name	Business_address	website_url	
--------------	---------------	------------------	-------------	--

- **reg_no (VARCHAR(20))** – Foreign key referencing users(reg_no), uniquely identifies each tour operator.
- **operator_name (VARCHAR(100))** – Name of the tour operator, required field.
- **business_address (VARCHAR(255))** – Address of the tour operator’s business, required field.
- **website_url (VARCHAR(100))** – Optional URL of the operator’s website.

4. Service Provider:

<u>RegNo</u>	Provider_name	Provider_location		
--------------	---------------	-------------------	--	--

- **reg_no (VARCHAR(20))** – Foreign key referencing users(reg_no), uniquely identifies each service provider.
- **provider_name (VARCHAR(100))** – Name of the service provider, required field.
- **provider_location (VARCHAR(100))** – Location of the service provider, required field.

5. Admin:

<u>RegNo</u>	Admin_name	Last_action_time	
--------------	------------	------------------	--

- **reg_no (VARCHAR(20))** – Foreign key linking to users(reg_no), uniquely identifies each admin.

- **admin_name (VARCHAR(100))** – Full name of the admin, required field.
- **last_action_timestamp (DATETIME)** – Records the last time the admin performed an action (can be null).

6. Location:

<u>Dest_id</u>	Destination_name	city	Region	Country
----------------	------------------	------	--------	---------

- **dest_id (VARCHAR(20))** – Primary key with region-based prefix (e.g., 'ASI-', 'EUR-'), uniquely identifies each location.
- **destination_name (VARCHAR(100))** – Name of the travel destination, required field.
- **city (VARCHAR(100))** – City where the destination is located, required field.
- **region (VARCHAR(100))** – Continent/region of the destination, must be one of: Asia, Africa, Europe, North America, South America, Oceania, Antarctica.
- **country (VARCHAR(100))** – Country of the destination, required field.

7. Trips:

<u>trip_id</u>	Title	description	Start_loc_id	Start_date	End_date	Duration	capacity
----------------	-------	-------------	--------------	------------	----------	----------	----------

Status(active, cancelled, completed)	category	Price_per_person	Image_url	Operator_id
--------------------------------------	----------	------------------	-----------	-------------

- **trip_id (VARCHAR(20))** – Primary key with format 'TRIP-XXXXXX', uniquely identifies each trip.
- **title (VARCHAR(100))** – Descriptive title of the trip, required field.
- **descirption (TEXT)** – Detailed description of the trip itinerary and features.
- **capacity (INT)** – Maximum number of travelers allowed on the trip.
- **duration (INT)** – Duration of the trip in days.
- **category (VARCHAR(50))** – Type of trip, must be one of: adventure, cultural, leisure, wildlife, historical, beach, mountain, cruise.
- **status (VARCHAR(50))** – Current state of the trip: active, completed, or cancelled.
- **price_per_person (DECIMAL(10, 2))** – Cost per person for the trip.
- **start_loc_id (VARCHAR(20))** – Foreign key linking to the starting location in the location table.

- **start_date (DATE)** – Date when the trip begins.
- **end_date (DATE)** – Date when the trip ends.
- **operator_id (VARCHAR(20))** – Foreign key referencing the tour_operator responsible for the trip.
- **profileTrip_image_url (VARCHAR(255))** – Optional URL of the profile image representing the trip.

8. Bookings:

<u>Booking_id</u>	Book_date	Booking_status	Traveler_id	Trip_id	
-------------------	-----------	----------------	-------------	---------	--

- **booking_id (VARCHAR(20))** – Primary key with format 'BOOK-XXXXXX', uniquely identifies each booking.
- **book_date (DATETIME)** – Timestamp indicating when the booking was made.
- **booking_status (VARCHAR(50))** – Current state of the booking: confirmed, pending, cancelled, or abandoned.
- **traveler_id (VARCHAR(20))** – Foreign key referencing the travelers table, identifies the user who made the booking.
- **trip_id (VARCHAR(20))** – Foreign key referencing the trips table, links to the booked trip.

9. Transactions:

<u>Transaction_id</u>	Acc_no	Amount	Transaction_date	Payment_method	Booking_id	status
-----------------------	--------	--------	------------------	----------------	------------	--------

- **transaction_id (VARCHAR(20))** – Primary key with format 'TXN-XXXXXX', uniquely identifies each transaction.
- **amount (DECIMAL(10, 2))** – Total amount paid in the transaction.
- **transaction_date (DATETIME)** – Timestamp indicating when the transaction occurred.
- **payment_method (VARCHAR(50))** – Mode of payment used: credit_card, debit_card, paypal, or bank_transfer.
- **booking_id (VARCHAR(20))** – Foreign key referencing the bookings table, links the transaction to a specific booking.
- **status (VARCHAR(50))** – Current status of the transaction: success, failed, or pending.

- **sending_account_number (VARCHAR(50))** – Account number from which the payment was made.

10. Preferences:

<u>Traveler_id</u>	<u>Destination_id</u>	Preference_level
--------------------	-----------------------	------------------

- **traveler_id (VARCHAR(20))** – Foreign key referencing the travelers table, links each preference to a specific traveler.
- **destination_id (VARCHAR(20))** – Foreign key referencing the location table, links each preference to a specific destination.
- **preference_level (INT)** – Indicates the level of preference for the destination, with a value between 1 and 5.

11. Trip location:

<u>Trip_id</u>	<u>Dest_id</u>	<u>Destination_order</u>	
----------------	----------------	--------------------------	--

- **trip_id (VARCHAR(20))** – Foreign key referencing the trips table, linking a trip to its associated locations.
- **location_id (VARCHAR(20))** – Foreign key referencing the location table, linking a trip to a specific destination location.
- **destination_order (INT)** – Specifies the order of the location in the trip itinerary, with values between 1 and 10.

12. Services:

<u>Service_id</u>	Service_type	Service_description	Price_per_person	capacity	Provider_id
-------------------	--------------	---------------------	------------------	----------	-------------

- **service_id (VARCHAR(20))** – Primary key for the service, uniquely identifying each service entry.
- **service_type (VARCHAR(50))** – Type of service provided, which can be one of 'hotel', 'transport', 'guide', 'activity', or 'other'.
- **service_description (TEXT)** – Detailed description of the service offered.
- **price (DECIMAL(10, 2))** – The price of the service per unit (e.g., per night, per trip, etc.).

- **provider_id (VARCHAR(20))** – Foreign key referencing the service_provider table, linking the service to the provider.
- **capacity (INT)** – The number of people that the service can accommodate or handle.

13. Hotel services:

<u>Service_id</u>	Room_type	amenities
-------------------	-----------	-----------

- **service_id (VARCHAR(20))** – Primary key for the hotel service, referencing the services table to link a service to the hotel.
- **room_type (VARCHAR(255))** – Type of room offered, which can be one of 'single', 'double', 'suite', 'deluxe', 'family', or 'studio'.
- **amenities (TEXT)** – Description of the amenities provided in the room, such as Wi-Fi, air conditioning, etc.

14. guide services:

<u>Service_id</u>	Guide_name	yearsOfExperience	certificationStatus
-------------------	------------	-------------------	---------------------

- **service_id (VARCHAR(20))** – Primary key for the guide service, referencing the services table to link a service to the guide.
- **guide_name (VARCHAR(100))** – Name of the guide providing the service.
- **years_of_experience (INT)** – Number of years the guide has been active in the industry.
- **certification_status (BIT)** – Indicates whether the guide is certified (1 for certified, 0 for not certified).

15. Transport services:

<u>Service_id</u>	Vehicle_type	Ac_available
-------------------	--------------	--------------

- **service_id (VARCHAR(20))** – Primary key for the transport service, referencing the services table to link the service to transport.
- **vehicle_type (VARCHAR(50))** – Type of vehicle used for transport, constrained to predefined options such as bus, car, bike, van, truck, and SUV.
- **ac_available (BIT)** – Indicates whether the vehicle has air conditioning (1 for available, 0 for not available).

16. Trip Services:

<u>Trid_id</u>	<u>Service_id</u>	
----------------	-------------------	--

- **trip_id (VARCHAR(20))** – Foreign key referencing the trips table, linking a service to a specific trip.
- **service_id (VARCHAR(20))** – Foreign key referencing the services table, linking a service to a specific trip.
- **PRIMARY KEY (trip_id, service_id)** – Composite primary key to uniquely identify the combination of a trip and a service.

17. Digital passes:

<u>Pass_id</u>	Date_generated	Valid_till	Document_type	File_url	Booking_id	Service_id
----------------	----------------	------------	---------------	----------	------------	------------

- **pass_id (VARCHAR(20))** – Primary key for identifying the digital pass, with a format constraint based on type (e-ticket, hotel voucher, activity pass).
- **date_generated (DATETIME)** – Timestamp when the digital pass was generated.
- **valid_till (DATETIME)** – Expiration date and time for the digital pass.
- **document_type (VARCHAR(50))** – Type of the document (e-ticket, hotel voucher, activity pass).
- **booking_id (VARCHAR(20))** – Foreign key referencing the bookings table, linking the digital pass to a booking.
- **service_id (VARCHAR(20))** – Foreign key referencing the services table, linking the digital pass to a specific service (optional).

18. Trip reviews:

<u>Review_id</u>	User_id	Rating	description	Review_date	Flag_status	isEdited	Trip_id
------------------	---------	--------	-------------	-------------	-------------	----------	---------

- **review_id (VARCHAR(20))** – Primary key for identifying each review, with a format constraint (TRVW-#####).
- **trip_id (VARCHAR(20))** – Foreign key referencing the trips table, linking the review to a specific trip.

- **traveler_id (VARCHAR(20))** – Foreign key referencing the travelers table, linking the review to a specific traveler.
- **rating (INT)** – Rating given by the traveler for the trip, between 1 and 5.
- **description (TEXT)** – Optional detailed description of the review.
- **review_date (DATETIME)** – Timestamp when the review was created, defaulting to the current date and time.
- **flag_status (VARCHAR(50))** – Status of the review indicating whether it's flagged for inappropriate content, defaulting to 'clear'.

19. Service reviews:

<u>Review_id</u>	User_id	Service_id	Rating	description	Review_date	Flag_status	isEdited
------------------	---------	------------	--------	-------------	-------------	-------------	----------

- **review_id (VARCHAR(20))** – Primary key for identifying each service review, with a format constraint (SRVW-#####).
- **service_id (VARCHAR(20))** – Foreign key referencing the services table, linking the review to a specific service.
- **traveler_id (VARCHAR(20))** – Foreign key referencing the travelers table, linking the review to the traveler who submitted it.
- **rating (INT)** – Rating given to the service, constrained to be between 1 and 5.
- **description (TEXT)** – Optional text description of the review, detailing the traveler's experience with the service.
- **review_date (DATETIME)** – Timestamp of when the review was created, defaulting to the current date and time.
- **flag_status (VARCHAR(50))** – Status indicating whether the review is 'clear' or 'flagged', defaulting to 'clear'.

20. User Approval logs:

<u>Log_id</u>	User_id	Admin_id	Log_datetime	Reject_reason	action
---------------	---------	----------	--------------	---------------	--------

- **log_id (VARCHAR(20))** – Primary key for identifying each approval log, with a format constraint (UAL-#####).
- **user_id (VARCHAR(20))** – Foreign key referencing the users table, linking the approval log to a specific user.
- **admin_id (VARCHAR(20))** – Foreign key referencing the admins table, linking the log to the admin who performed the action.
- **log_time (DATETIME)** – Timestamp of when the approval or rejection action was taken, defaulting to the current date and time.
- **action (VARCHAR(50))** – The action taken by the admin, either 'approved' or 'rejected'.
- **reason (TEXT)** – Optional detailed reason for the approval or rejection.

21. Trip Review logs:

<u>Log_id</u>	Review_id	Admin_id	Log_time	action	Reason
---------------	-----------	----------	----------	--------	--------

- **log_id (VARCHAR(20))** – Primary key for identifying each review log, with a format constraint (TRL-#####).
- **review_id (VARCHAR(20))** – Foreign key referencing the trip_reviews table, linking the log to a specific review.
- **admin_id (VARCHAR(20))** – Foreign key referencing the admins table, linking the log to the admin who performed the action.
- **log_time (DATETIME)** – Timestamp of when the review was cleared or flagged, defaulting to the current date and time.
- **action (VARCHAR(50))** – The action taken by the admin, either 'clear' or 'flagged'.
- **reason (TEXT)** – Optional detailed reason for the action taken on the review.

22. Service Review Approvals:

<u>Log_id</u>	Review_id	Admin_id	Log_time	action	Reason
---------------	-----------	----------	----------	--------	--------

- **log_id (VARCHAR(20))** – Primary key for identifying each log entry, with a format constraint (TRL-#####).
- **review_id (VARCHAR(20))** – Foreign key referencing the service_reviews table, linking the log entry to a specific review.

- **admin_id (VARCHAR(20))** – Foreign key referencing the admins table, linking the log entry to the admin who took action.
- **log_time (DATETIME)** – Timestamp of when the log entry was created, defaulting to the current date and time.
- **action (VARCHAR(50))** – Action taken on the review, either 'clear' or 'flagged'.
- **reason (TEXT)** – Optional text describing the reason for the action taken (if any).