VACATION DATABASE MANAGEMENT SYSTEM

A FINAL PROJECT REPORT SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR COURSE STAT311 – MODERN DATABASE SYSTEMS

DEPARTMENT OF STATISTICS OF

MIDDLE EAST TECHNICAL UNIVERSITY

BY

İSMAİL EREN ÇAKIR

NURAY TAGHIYEVA

SİNEM SARICA

SU ALADAĞ

January 2025

TABLE OF CONTENT

Introduction

There are several platforms to book a hotel reservation, plan travel for a vacation or find a ticketk for an event. However, most of the websites do not combine all of these features.

In Vacation Database Management System, our goal is to design a platform that can achieve the followings:

- Providing different type of accommodation,
- Finding all possible trasnport options and booking,
- Planning activities take place in accommoditons,
- Booking a hotel,
- Displaying reservation history.

Only the system was built to reach these goals, yet there is no an application form for these system. This structure solely a practical demonstrations of planned functions.

Here the functions of In the Vacation Database Management System:

- With predefined restrictions, tables are connected each other meaning that changes in one table directly affect the related table.
- Reservation detail changes can be displayed with past reseravation table via trigger functions.

Entire project was built on phpMyAdmin on Ubuntu Virtual Machine via VMware Workstation Pro.

ER Diagram

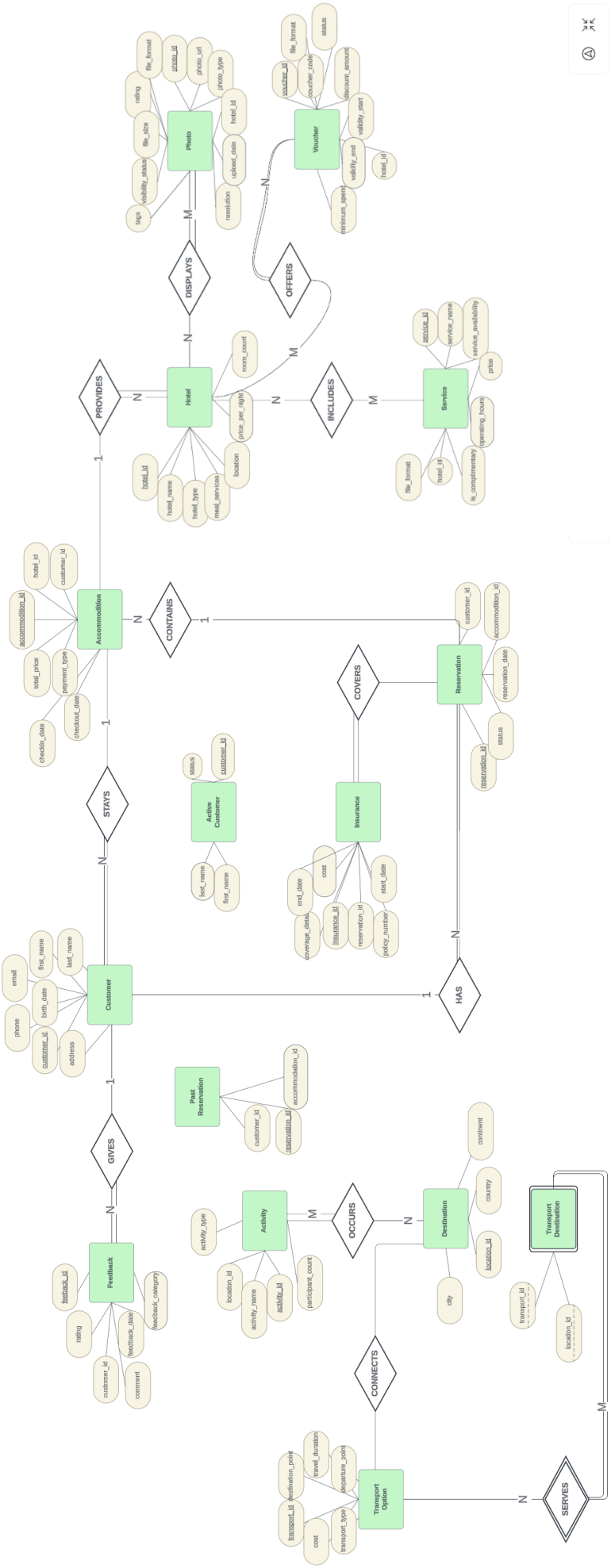
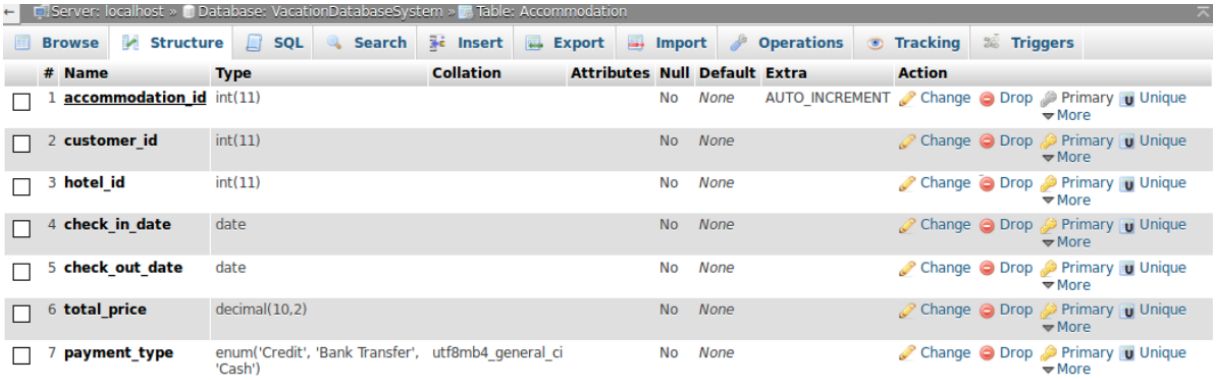


Photo
photo_id
photo_url
photo_type
file_format
rating
file_size
hotel_id
upload_date
resolution
visibility_status
tags
Voucher
voucher_id
file_format
status
voucher_code
discount_amount
validity_start
validity_end
hotel_id
minimum_spend
DISPLAYS
OFFERS
Hotel
room_count
price_per_night
hotel_id
hotel_name
hotel_type
meal_services
location
PROVIDES
INCLUDES
Service
service_id
service_name
service_availability
price
operating_hours
is_complimentary
hotel_id
file_format
Accommodation
hotel_id
customer_id
accommodation_id
total_price
payment_type
checkout_date
checkin_date
CONTAINS
Reservation
customer_id
accommodation_id
reservation_date
status
reservation_id
COVERS
STAYS
Active Customer
status
customer_id
last_name
first_name
Insurance
cost
start_date
end_date
coverage_details
insurance_id
reservation_id
policy_number
Customer
email
first_name
last_name
birth_date
phone
customer_id
address
HAS
Past Reservation
accommodation_id
customer_id
reservation_id
GIVES
Feedback
feedback_id
rating
feedback_date
feedback_category
customer_id
comment
Activity
activity_type
location_id
activity_name
activity_id
participant_count
OCCURS
Destination
continent
country
location_id
city
Transport Destination
transport_id
location_id
CONNECTS
SERVES
Transport Option
travel_duration
destination_point
departure_point
transport_id
transport_type
cost

Schema Design

The design and its table properties are built in phpmyadmin tool. Here are the tables and design properties.

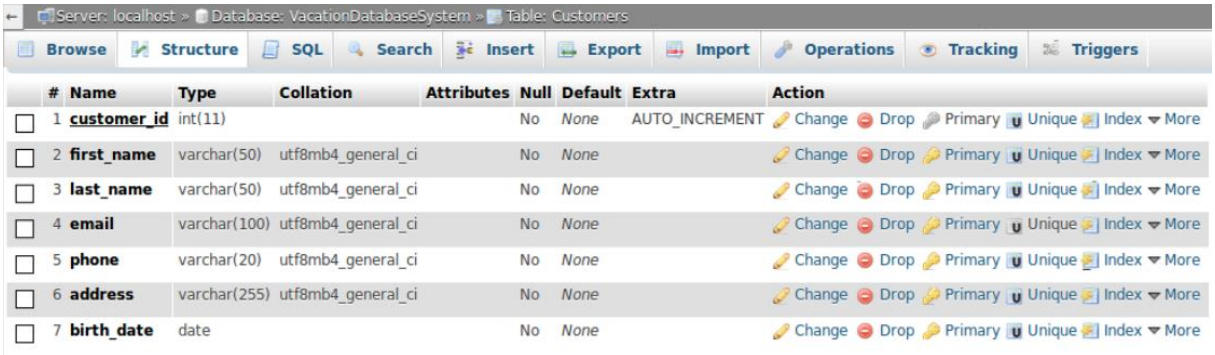
Accommodation Schema



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	accommodation_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique More
2	customer_id	int(11)			No	None		Change Drop Primary Unique More
3	hotel_id	int(11)			No	None		Change Drop Primary Unique More
4	check_in_date	date			No	None		Change Drop Primary Unique More
5	check_out_date	date			No	None		Change Drop Primary Unique More
6	total_price	decimal(10,2)			No	None		Change Drop Primary Unique More
7	payment_type	enum('Credit', 'Bank Transfer', 'Cash')	utf8mb4_general_ci		No	None		Change Drop Primary Unique More

accommoadtion_id is the primary key of the Accommodation table. customer_id is unique IDs of each visitors. hotel_id is the indentifier of hotels. check_in_date and check_out_date are the dates that visitors come and leave the facilities. total_price is the total amount they paid. Lastly, payment _type keeps information about the differen payment types such as ‘credit’, ‘bank transfer’ and ‘cash’. customer_id, hotel_id are the foreign keys.

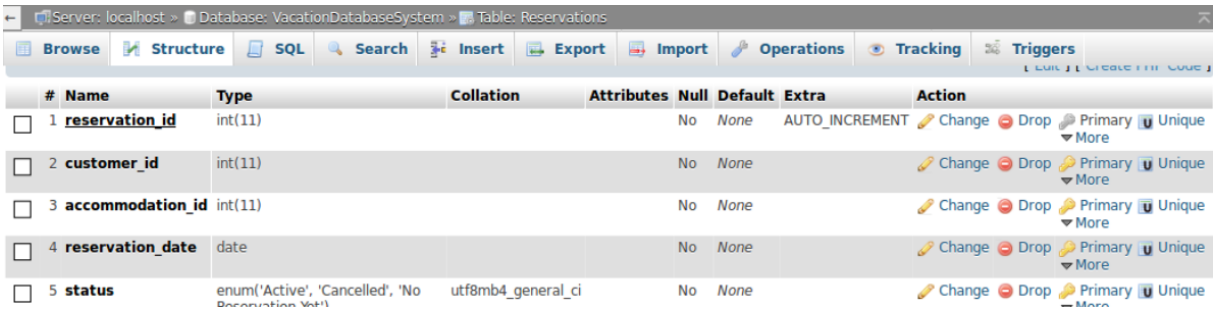
Customer Schema



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	customer_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index More
2	first_name	varchar(50)	utf8mb4_general_ci		No	None		Change Drop Primary Unique Index More
3	last_name	varchar(50)	utf8mb4_general_ci		No	None		Change Drop Primary Unique Index More
4	email	varchar(100)	utf8mb4_general_ci		No	None		Change Drop Primary Unique Index More
5	phone	varchar(20)	utf8mb4_general_ci		No	None		Change Drop Primary Unique Index More
6	address	varchar(255)	utf8mb4_general_ci		No	None		Change Drop Primary Unique Index More
7	birth_date	date			No	None		Change Drop Primary Unique Index More

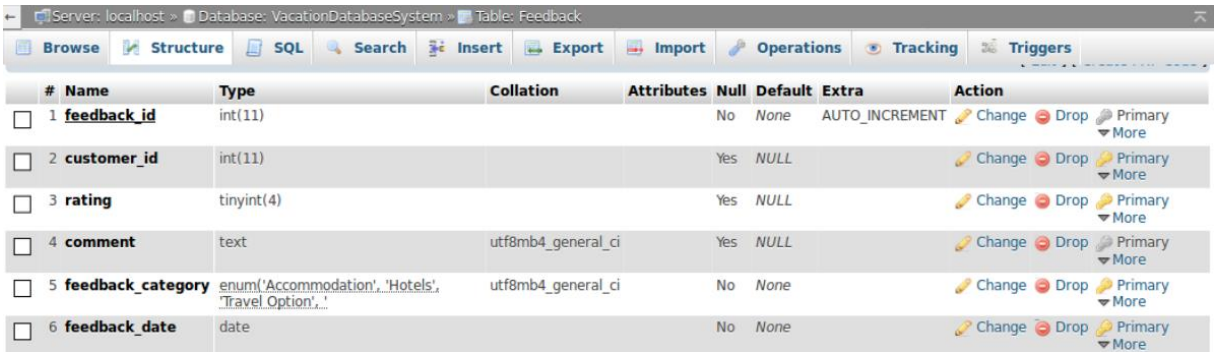
The Customers table is part of a database designed to store information about individuals who make bookings in a vacation or hotel system. The primary key, customer_id, is an auto-incrementing integer that uniquely identifies each customer in the table. The table includes fields such as first_name and last_name, both defined as varchar(50), to store the customer’s name. The email column, a varchar(100), stores the customer’s email address and may be used for communication or login purposes. The phone column, a varchar(20), records the customer’s contact number. The address column, defined as varchar(255), stores the

Reservation Schema



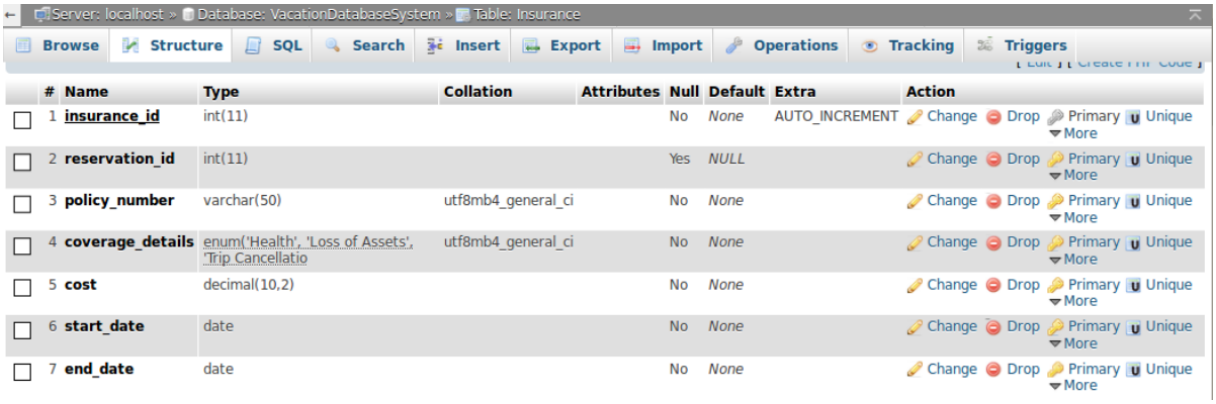
This schema tracks the booking details for accommodations. The reservation_id, is the primary key and uniquely identifies each reservation. The customer_id, links the reservation to a specific customer, referencing the customer in the "Customer" table. The accommodation_id, associates the reservation with a particular accommodation option, such as a hotel room or other facility, by referencing the "Accommodation" table. The reservation_date, a date, records the date when the reservation was made. The status, an enum field, reflects the current state of the reservation, with options like "Active," "Cancelled," or "No Reservation Yet." This table is essential for tracking and managing reservation records in the system, ensuring that all booking details are properly associated and up-to-date.

Feedback Schema



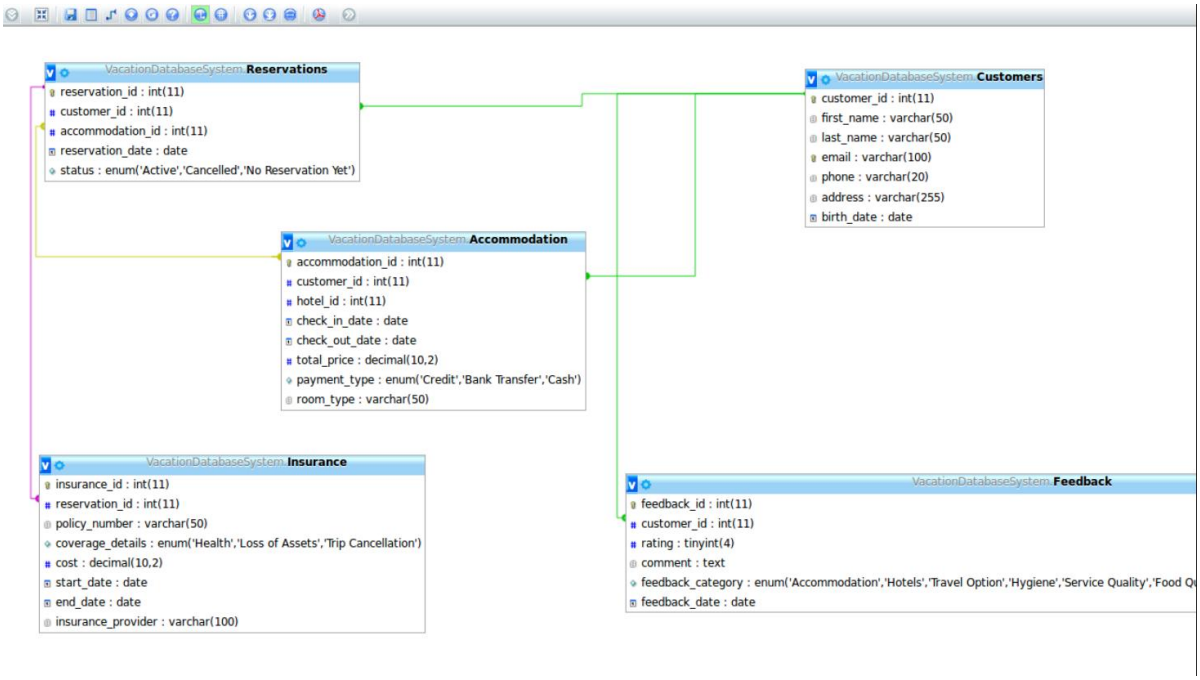
The Feedback table is designed to store customer feedback related to their experiences in a vacation or hotel booking system. The primary key, feedback_id, is an auto-incrementing integer that uniquely identifies each feedback entry. The customer_id column, an integer that allows NULL, links the feedback to a specific customer, likely referencing the Customers table. The rating column, of type tinyint(4), stores a numeric rating provided by the customer, representing their evaluation of the service or experience. The comment column, defined as text, allows customers to provide detailed written feedback. The feedback_category column, an enum, specifies the category of feedback, such as "Accommodation," "Hotels," or "Travel Option," for better organization and classification. Finally, the feedback_date column, of type date, records when the feedback was submitted. This schema is well-suited for managing customer feedback in the system.

Insurance Schema



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	insurance_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique More
2	reservation_id	int(11)			Yes	NULL		Change Drop Primary Unique More
3	policy_number	varchar(50)	utf8mb4_general_ci		No	None		Change Drop Primary Unique More
4	coverage_details	enum('Health', 'Loss of Assets', 'Trip Cancellatio	utf8mb4_general_ci		No	None		Change Drop Primary Unique More
5	cost	decimal(10,2)			No	None		Change Drop Primary Unique More
6	start_date	date			No	None		Change Drop Primary Unique More
7	end_date	date			No	None		Change Drop Primary Unique More

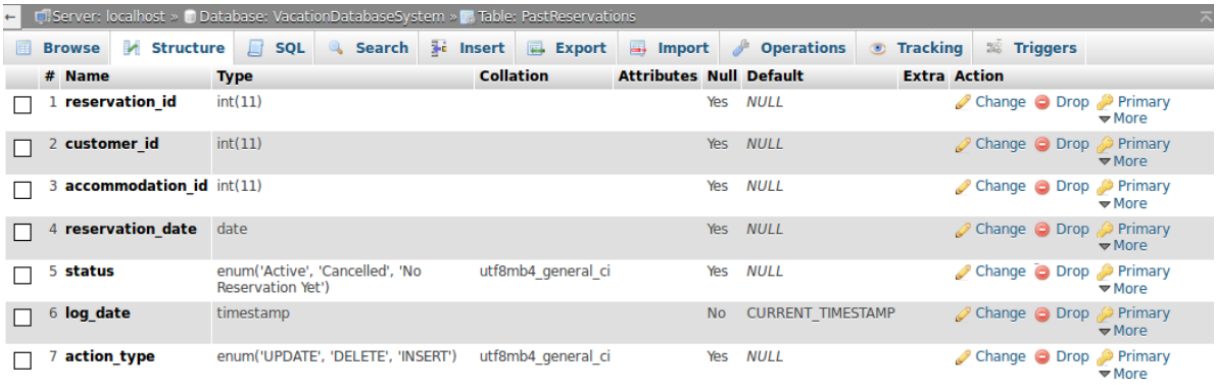
This table, named Insurance, is part of the Vacation Database System and manages insurance details for reservations. The primary key is insurance_id, an auto-incrementing integer that uniquely identifies each insurance record. It is linked to the reservation_id column, establishing a relationship with the Reservations table. The policy_number column stores a unique identifier for each insurance policy. The coverage_details column, defined as an ENUM, specifies the type of coverage, such as "Health," "Loss of Assets," or "Trip Cancellation." The cost column represents the price of the insurance policy with a decimal precision of two. The start_date and end_date columns track the validity period of the insurance. Together, these attributes provide comprehensive information on insurance policies associated with customer reservations.



The Customers table stores personal details of customers, such as their name, email, and address, identified by a unique customer_id. The Reservations table links customers with accommodations and tracks reservation status. The Accommodation table details booking information, such as check-in/out dates, room types, and payment types. The Insurance table is connected to reservations, providing policy details and coverage types. The Feedback table

collects customer ratings and comments on various categories like accommodation and service quality. Relationships between these tables ensure that customers, reservations, and feedback are interconnected.

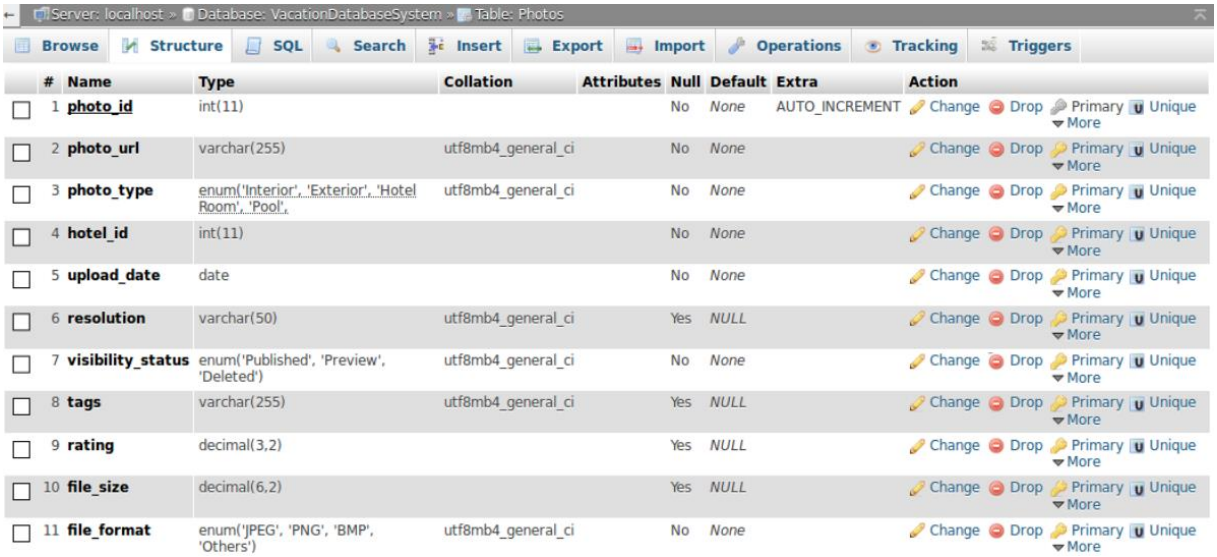
Past Reservation Schema



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	reservation_id	int(11)			Yes	NULL		Change Drop Primary More
2	customer_id	int(11)			Yes	NULL		Change Drop Primary More
3	accommodation_id	int(11)			Yes	NULL		Change Drop Primary More
4	reservation_date	date			Yes	NULL		Change Drop Primary More
5	status	enum('Active', 'Cancelled', 'No Reservation Yet')	utf8mb4_general_ci		Yes	NULL		Change Drop Primary More
6	log_date	timestamp			No	CURRENT_TIMESTAMP		Change Drop Primary More
7	action_type	enum('UPDATE', 'DELETE', 'INSERT')	utf8mb4_general_ci		Yes	NULL		Change Drop Primary More

This table is used to store logs of changes made to the "Reservation" table, specifically recording INSERT, UPDATE, and DELETE actions. The reservation_id records the ID of the reservation being modified. The customer_id references the customer involved in the reservation change. The accommodation_id, links to the accommodation that was affected. The reservation_date, a date, keeps track of the reservation's original date. The status, an enum, reflects the reservation's state at the time of change (e.g., "Active," "Cancelled," "No Reservation Yet"). The log_date, a timestamp, automatically records the exact date and time of the change using timestamp data type. The action_type, an enum, stores the type of action performed on the reservation, “INSERT”, "UPDATE" or "DELETE." This table helps track and maintain a history of modifications to the reservation records, ensuring data accuracy and keeping a record of system changes.

Photos Schema



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	photo_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique More
2	photo_url	varchar(255)	utf8mb4_general_ci		No	None		Change Drop Primary Unique More
3	photo_type	enum('Interior', 'Exterior', 'Hotel Room', 'Pool',	utf8mb4_general_ci		No	None		Change Drop Primary Unique More
4	hotel_id	int(11)			No	None		Change Drop Primary Unique More
5	upload_date	date			No	None		Change Drop Primary Unique More
6	resolution	varchar(50)	utf8mb4_general_ci		Yes	NULL		Change Drop Primary Unique More
7	visibility_status	enum('Published', 'Preview', 'Deleted')	utf8mb4_general_ci		No	None		Change Drop Primary Unique More
8	tags	varchar(255)	utf8mb4_general_ci		Yes	NULL		Change Drop Primary Unique More
9	rating	decimal(3,2)			Yes	NULL		Change Drop Primary Unique More
10	file_size	decimal(6,2)			Yes	NULL		Change Drop Primary Unique More
11	file_format	enum('JPEG', 'PNG', 'BMP', 'Others')	utf8mb4_general_ci		No	None		Change Drop Primary Unique More

This table stores images associated with hotels. The photo_id is the primary key that uniquely identifies each photo. The photo_url contains the link to the photo, while photo_type (enum)

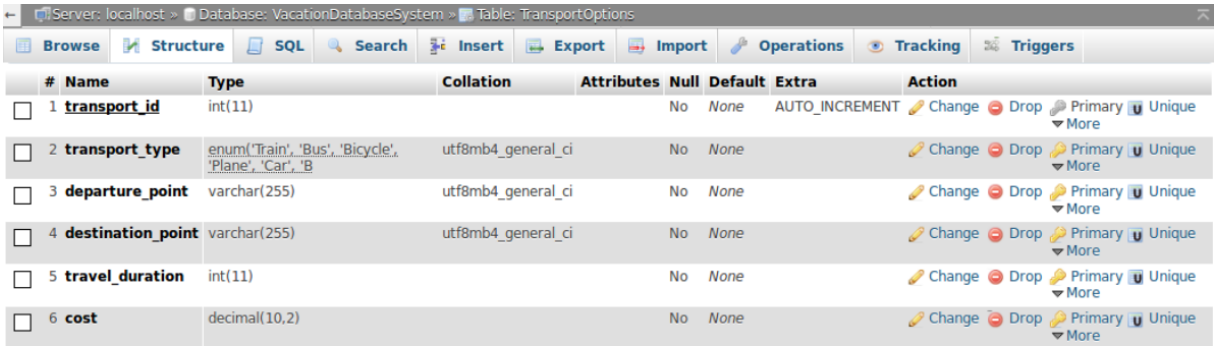
categorizes the photo as "Interior," "Exterior," "Hotel Room," or other types. The hotel_id connects the photo to a specific hotel. The upload_date marks when the photo was uploaded. The visibility_status (enum) indicates whether the photo is "Published," "Preview," or "Deleted." The table also stores optional fields like resolution for photo quality, tags for categorization, rating (decimal) for user feedback, file_size (decimal) to track the photo size, and file_format (enum) to specify the image format (e.g., "JPEG," "PNG"). This structure helps organize photos and manage their visibility and quality efficiently. Also, this table is connected to the "Hotel" table through a "display" relationship, linking hotel images to specific hotels.

Service Schema

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	service_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index More
2	service_name	varchar(100)	utf8mb4_general_ci		No	None		Change Drop Primary Unique Index More
3	service_availability	tinyint(1)			No	None		Change Drop Primary Unique Index More
4	price	decimal(10,2)			Yes	NULL		Change Drop Primary Unique Index More
5	operating_hours	varchar(50)	utf8mb4_general_ci		Yes	NULL		Change Drop Primary Unique Index More
6	is_complimentary	tinyint(1)			No	None		Change Drop Primary Unique Index More
7	hotel_id	int(11)			No	None		Change Drop Primary Unique Index More

This table is designed to manage the various services provided within the vacation management system. The service_id is the primary key of service table, uniquely identifies each service. The service_name stores the name of the service, such as "Spa Treatment" or "Gym Access." The service_availability attribute, a tinyint(1) type, indicates whether the service is currently available (1) or not (0). The price, defined as decimal (10,2), specifies the service's cost, with complimentary services leaving this field empty. The operating_hours defines the operational times of the service. The is_complimentary, a tinyint(1) type, signifies if a service is free (1) or not (0). The hotel_id acts as a foreign key referencing the hotel_id in the "Hotel" table, linking the service to a specific hotel or accommodation. This table provides detailed information about the types of services offered at each hotel, including their availability, cost, and operating hours. The "Service" table is connected to the "Hotel" table through an "includes" relationship, indicating which services are available at each hotel.

Transport Options Schema

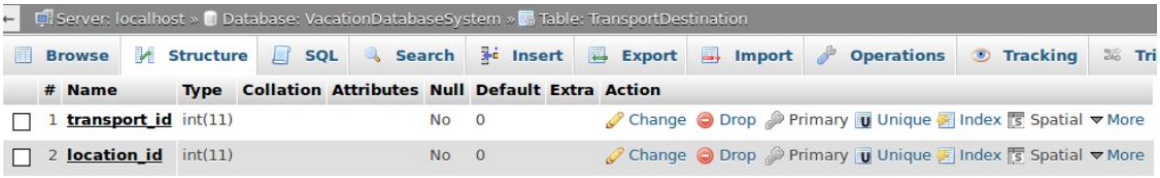


#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	transport_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique More
2	transport_type	enum('Train', 'Bus', 'Bicycle', 'Plane', 'Car', 'B	utf8mb4_general_ci		No	None		Change Drop Primary Unique More
3	departure_point	varchar(255)	utf8mb4_general_ci		No	None		Change Drop Primary Unique More
4	destination_point	varchar(255)	utf8mb4_general_ci		No	None		Change Drop Primary Unique More
5	travel_duration	int(11)			No	None		Change Drop Primary Unique More
6	cost	decimal(10,2)			No	None		Change Drop Primary Unique More

This table is designed to manage and organize various transportation methods available within the vacation management system. The transport_id is the primary key, in which uniquely identifies each transport option. The transport_type, defined as an enum data type, specifies

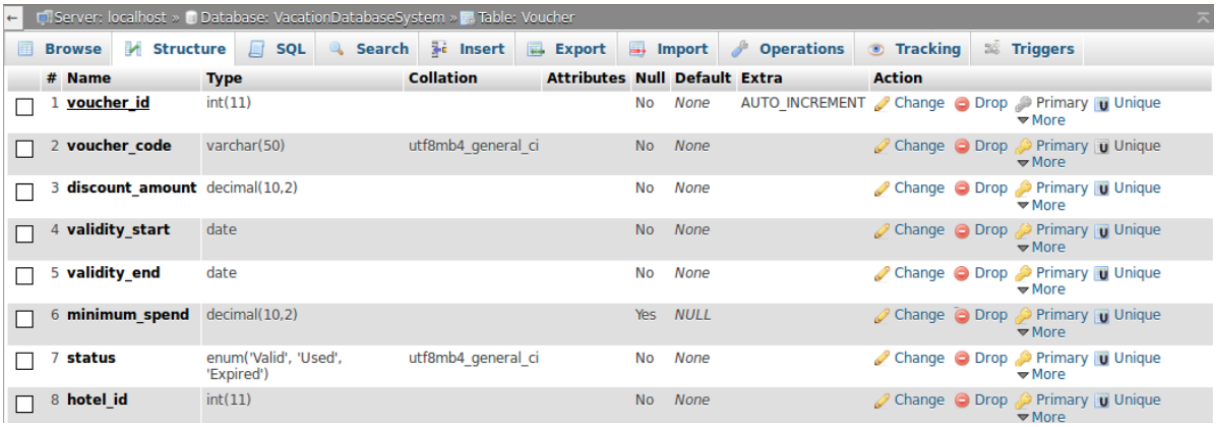
the type of transportation, including options such as "Train," "Bus," "Bicycle," "Plane," and "Car." The departure_point indicates the starting location of the journey, while the destination_point specifies the endpoint of the trip. The travel_duration records the estimated duration of the journey in minutes. The cost, defined as decimal (10,2), represents the price of the transport option. This table allows for easy tracking of transportation details, enabling users to access information about available transport types, departure and destination points, travel durations, and associated costs. This table also connects to the "Destination" table via a "connects" relationship and to the "Transport Destination" table via a "serves" relationship to map routes and transport options effectively.

Transport Destination Schema



This table serves as a bridge between transportation options and specific destinations. It includes the following attributes: transport_id acts as a foreign key linking to the unique identifier of a transport option in the “Transport Option” table (e.g., bus, train, or airplane). This field does not allow null values and has a default value of 0 to ensure data consistency. The location_id acts as a foreign key referencing the unique identifier of a location in the “Destination” table. Similar to transport_id, this field does not allow null values and is set to a default value of 0. Together, these two attributes form the composite primary key of the table, ensuring that each transport option and its associated destination are uniquely identified. This table is essential for organizing transport routes and showing available transport options for each destination. . This table connects to the "Transport Options" table via a "serves" relationship to map transport options effectively.

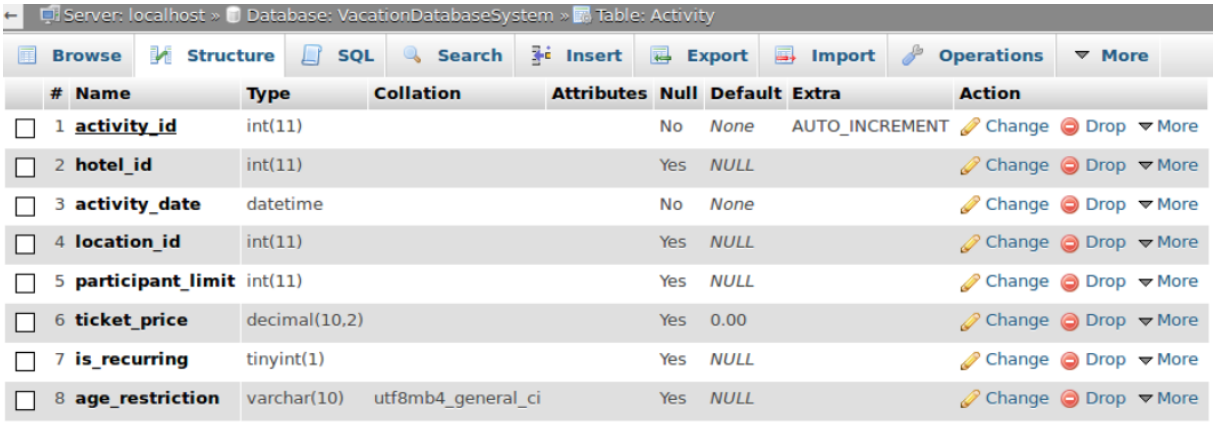
Voucher Schema



The "Voucher" table is designed to manage discount vouchers offered within the vacation management system. The voucher_id is the primary key, in which uniquely identifies each voucher. The voucher_code stores the unique code associated with the voucher, which users can redeem for discounts. The discount_amount, defined as a decimal (10,2) type, specifies the monetary value of the discount provided by the voucher. The validity_start and

validity_end, both date type, define the time period during which the voucher can be used. The minimum_spend represents the minimum amount required to be eligible for using the voucher, if applicable. The status, an enum field, tracks the current state of the voucher, with possible values such as "Valid," "Used," or "Expired." The hotel_idacts as a foreign key referencing the hotel_id in the "Hotel" table, linking each voucher to a specific hotel. This table provides a structured way to manage and track vouchers, ensuring seamless integration of discounts and promotions within the system. This table is connected to the "Hotel" table through an "offer" relationship, indicating which hotel the voucher applies to.

Activity Schema



The "Activity" table is used to record different activity types. The table includes fields such as activity_idis the primary key to uniquely identify each activity, hotel_id to specify the hotel where the activity takes place. Also, the hotel_id field is a foreign key that references the hotel_id in the "Hotel" table. This establishes a relationship between the "Activity" table and the "Hotel" table, ensuring that each activity is associated with a specific hotel. activity_date (datetime) to record the date and time of the activity, location_id for the location of the event, participant_limit to set the maximum number of participants, ticket_price (decimal(10,2)) for the ticket price, is_recurring (tinyint(1)) to indicate if the activity is recurring or not, and age_restriction to specify age limits if applicable. Additionally, the log_date (timestamp) field tracks the time of the change, while action_type (enum) records the type of action (INSERT, UPDATE, DELETE). The combination of activity_id and log_date forms a composite primary key to ensure each record is unique. This structure helps maintain an audit trail and ensures data integrity for system changes. If the "Activity" table has an "occurs" relationship with the Destination table, it suggests that an activity takes place at a specific destination.

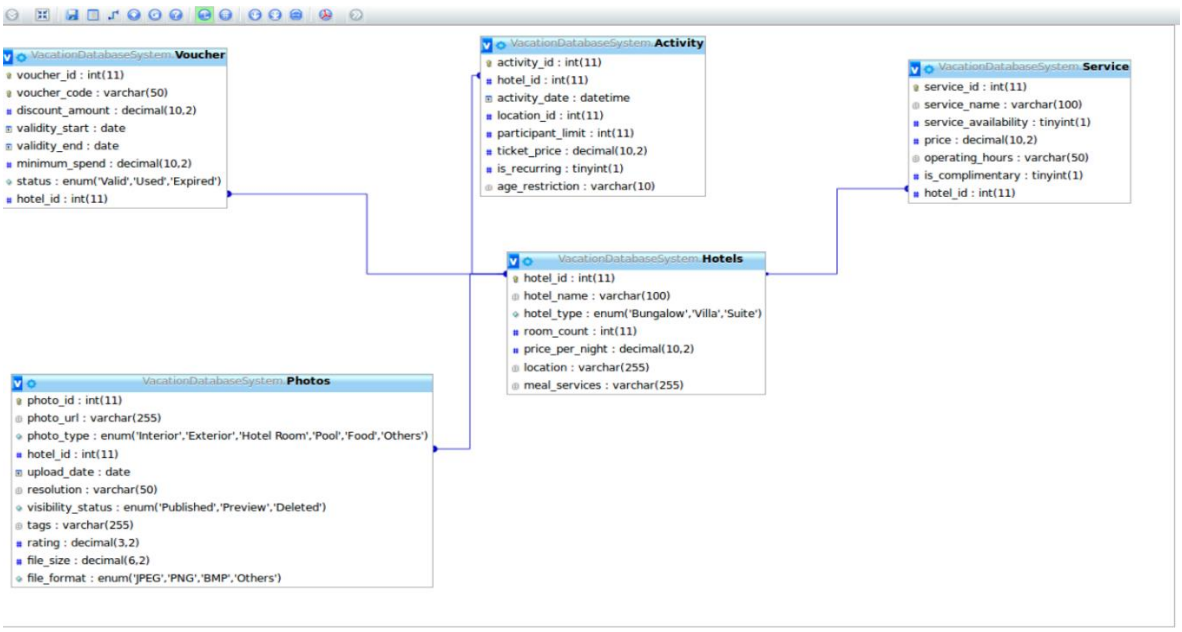
Hotels Schema

Server: localhost » Database: VacationDatabaseSystem » Table: Hotels

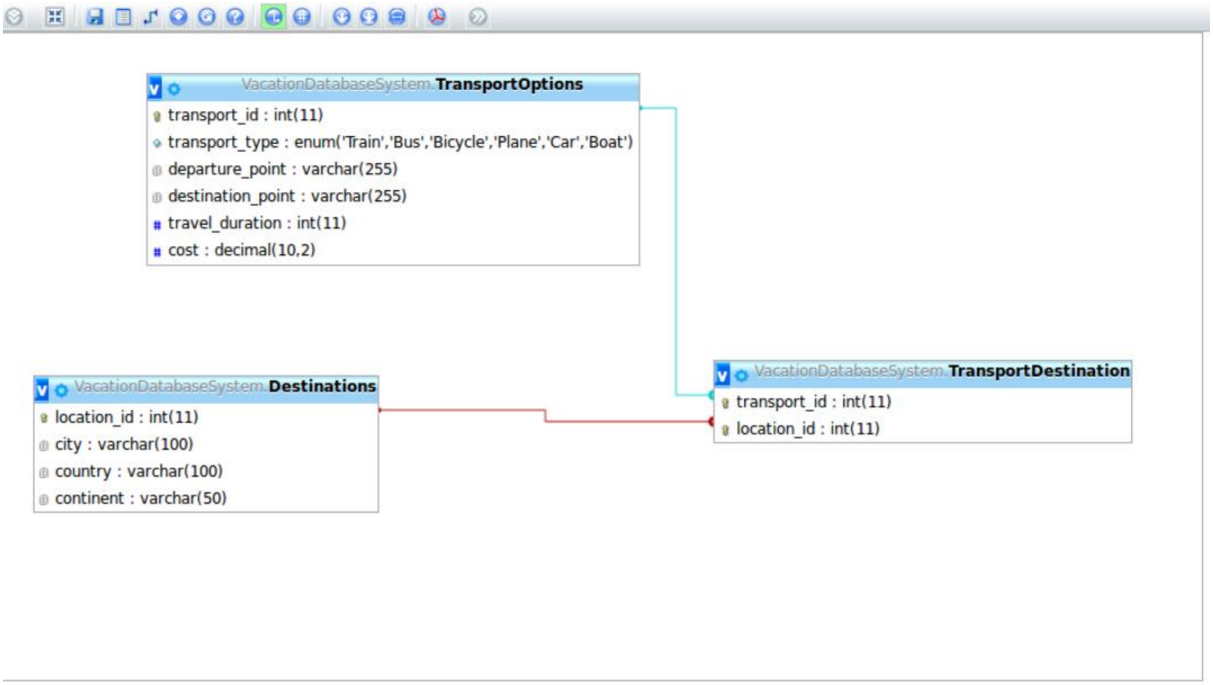
Browse | Structure | SQL | Search | Insert | Export | Import | Operations | More

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	hotel_id	int(11)			No	None	AUTO_INCREMENT	Change Drop More
2	hotel_name	varchar(100)	utf8mb4_general_ci		No	None		Change Drop More
3	hotel_type	enum('Bungalow', 'Villa', 'Suite')	utf8mb4_general_ci		No	None		Change Drop More
4	room_count	int(11)			No	None		Change Drop More
5	price_per_night	decimal(10,2)			No	None		Change Drop More
6	location	varchar(255)	utf8mb4_general_ci		No	None		Change Drop More
7	meal_services	varchar(255)	utf8mb4_general_ci		No	None		Change Drop More

The "Hotel" table is used to store information about hotels, including details such as their name, type, and amenities. The table consists of hotel_id is the primary key to uniquely identifies each hotel, hotel_name stores the name of the hotel and cannot be NULL, hotel_type (enum), which specifies the type of the hotel (such as "Bungalow," "Villa," or "Suite") and cannot be NULL, room_count tracks the number of rooms available in the hotel, and price_per_night (decimal(10,2)), which records the cost per night for staying at the hotel. Additionally, the location field stores the hotel's location and meal_services details the meal services available at the hotel, both of which are non-nullable. This structure helps maintain an organized record of hotel details, which can be used for managing and categorizing hotels in the system. The Hotel table has a "display" relationship with the Photos table where a hotel can have multiple photos, has a "offer" relationship with the Voucher table, where a hotel can offer several vouchers. has a "include" relationship with the Service table, where a hotel includes multiple services, has a "provide" relationship with the Accommodation table, where a hotel provides different types of accommodation.

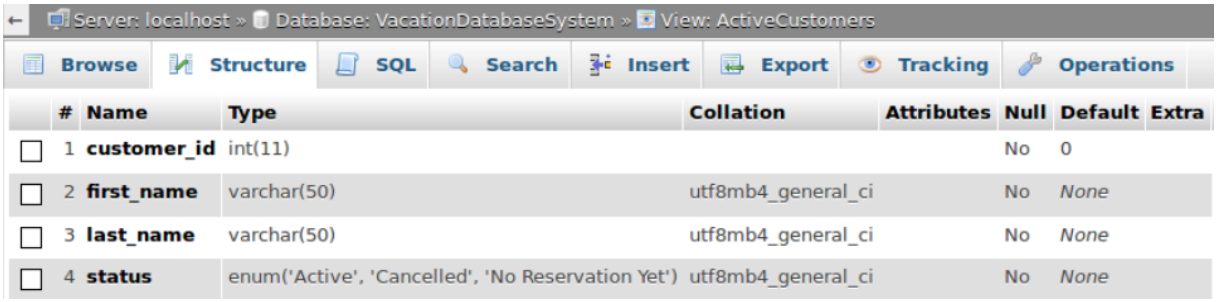


This schema models a "Vacation Database System" with interconnected tables for hotels, services, activities, vouchers, and photos. The "Hotels" table is central, storing details like name, type, and location, and linking to "Services" (hotel amenities), "Activity" (events or activities), "Voucher" (promotional offers), and "Photos" (hotel images with metadata). Foreign keys like "hotel_id" ensure data consistency across the system.



This schema models transportation options in a "Vacation Database System." The "TransportOptions" table stores details like transport type (e.g., train, bus, plane), departure and destination points, travel duration, and cost. The "Destinations" table contains information about locations, including city, country, and continent. The "TransportDestination" table links transport options to specific destinations using foreign keys, ensuring a connection between travel methods and vacation locations.

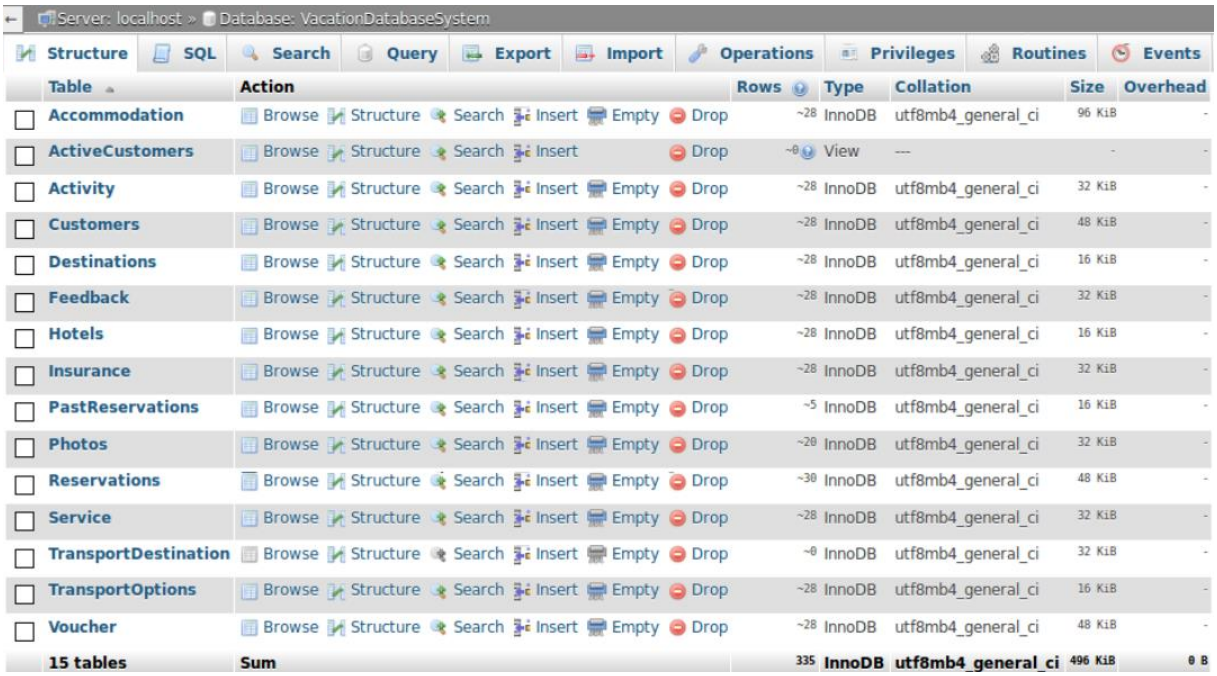
View Table: Active Customer Schema

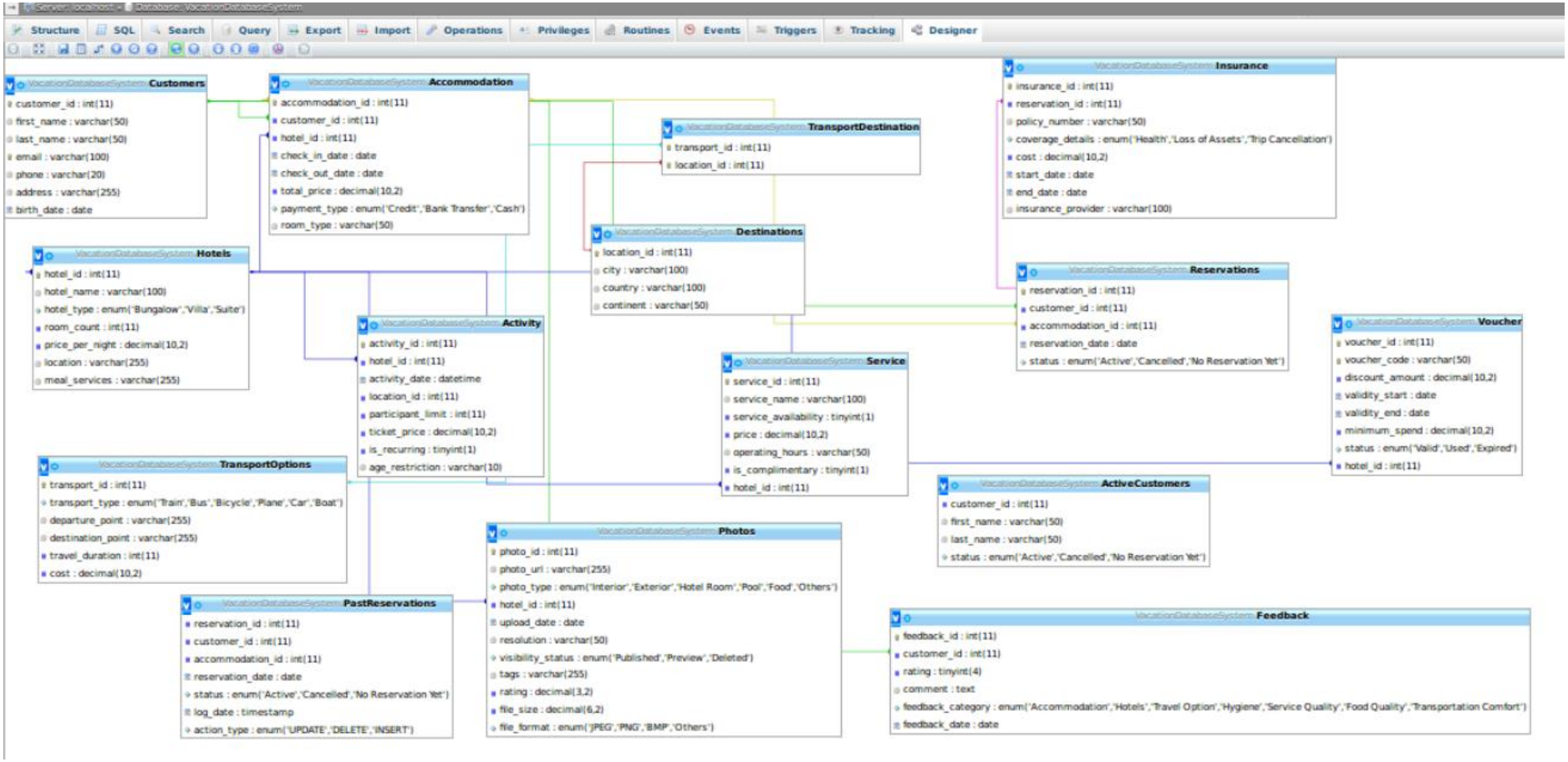


The "Active Customer" view schema is designed to provide a filtered list of customers who have an "Active" status in the system. The customer_id references the customer_id in the "Customer" table, linking each entry in the "Active Customers" view to a specific customer along with their first_name and last_name for name details. The status attribute, defined as an enum ('Active', 'Cancelled', 'No Reservation Yet'), ensures that only customers marked as

"Active" are included in the view. This schema provides an efficient way to manage and interact with active customers, simplifying operations and improving usability

General View of All Entities

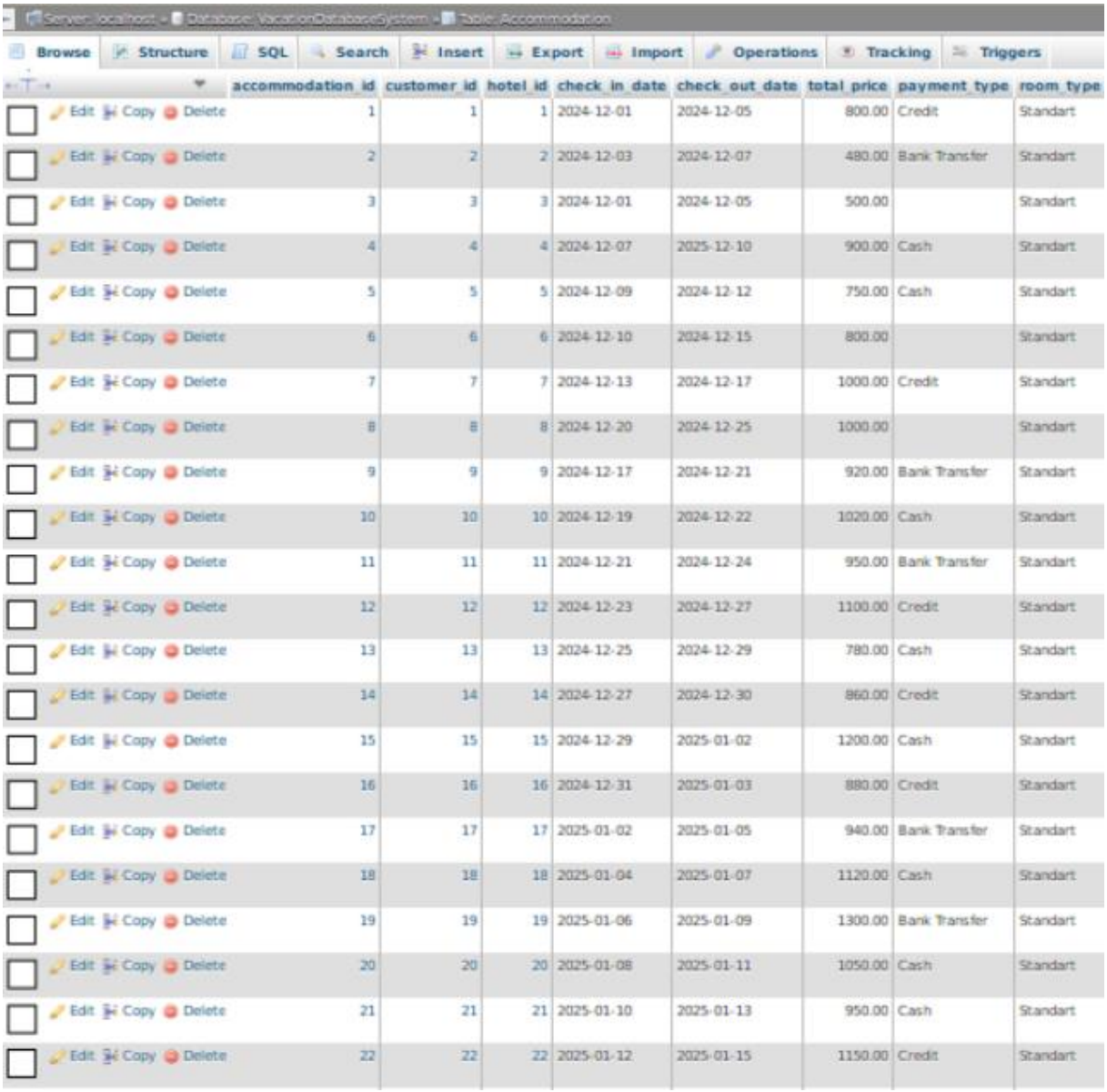



Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
Tracking
Designer
Customers
customer_id : int(11)
first_name : varchar(50)
last_name : varchar(50)
email : varchar(100)
phone : varchar(20)
address : varchar(255)
birth_date : date
Accommodation
accommodation_id : int(11)
customer_id : int(11)
hotel_id : int(11)
check_in_date : date
check_out_date : date
total_price : decimal(10,2)
payment_type : enum('Credit','Bank Transfer','Cash')
room_type : varchar(50)
TransportDestination
transport_id : int(11)
location_id : int(11)
Insurance
insurance_id : int(11)
reservation_id : int(11)
policy_number : varchar(50)
coverage_details : enum('Health','Loss of Assets','Trip Cancellation')
cost : decimal(10,2)
start_date : date
end_date : date
insurance_provider : varchar(100)
Hotels
hotel_id : int(11)
hotel_name : varchar(100)
hotel_type : enum('Bungalow','Villa','Suite')
room_count : int(11)
price_per_night : decimal(10,2)
location : varchar(255)
meal_services : varchar(255)
Destinations
location_id : int(11)
city : varchar(100)
country : varchar(100)
continent : varchar(50)
Reservations
reservation_id : int(11)
customer_id : int(11)
accommodation_id : int(11)
reservation_date : date
status : enum('Active','Cancelled','No Reservation Yet')
Activity
activity_id : int(11)
hotel_id : int(11)
activity_date : datetime
location_id : int(11)
participant_limit : int(11)
ticket_price : decimal(10,2)
is_recurring : tinyint(1)
age_restriction : varchar(10)
Service
service_id : int(11)
service_name : varchar(100)
service_availability : tinyint(1)
price : decimal(10,2)
operating_hours : varchar(50)
is_complimentary : tinyint(1)
hotel_id : int(11)
Voucher
voucher_id : int(11)
voucher_code : varchar(50)
discount_amount : decimal(10,2)
validity_start : date
validity_end : date
minimum_spend : decimal(10,2)
status : enum('Valid','Used','Expired')
hotel_id : int(11)
TransportOptions
transport_id : int(11)
transport_type : enum('Train','Bus','Bicycle','Plane','Car','Boat')
departure_point : varchar(255)
destination_point : varchar(255)
travel_duration : int(11)
cost : decimal(10,2)
ActiveCustomers
customer_id : int(11)
first_name : varchar(50)
last_name : varchar(50)
status : enum('Active','Cancelled','No Reservation Yet')
Photos
photo_id : int(11)
photo_url : varchar(255)
photo_type : enum('Interior','Exterior','Hotel Room','Pool','Food','Others')
hotel_id : int(11)
upload_date : date
resolution : varchar(50)
visibility_status : enum('Published','Preview','Deleted')
tags : varchar(255)
rating : decimal(3,2)
file_size : decimal(6,2)
file_format : enum('JPEG','PNG','BMP','Others')
PastReservations
reservation_id : int(11)
customer_id : int(11)
accommodation_id : int(11)
reservation_date : date
status : enum('Active','Cancelled','No Reservation Yet')
log_date : timestamp
action_type : enum('UPDATE','DELETE','INSERT')
Feedback
feedback_id : int(11)
customer_id : int(11)
rating : tinyint(4)
comment : text
feedback_category : enum('Accommodation','Hotels','Travel Option','Hygiene','Service Quality','Food Quality','Transportation Comfort')
feedback_date : date

Populated Tables

In order to populate the table, test records are used. These records are regularly updated during development. The final, stable versions of these tables, along with their populated data, are provided in this section as the frozen version. The tables and their corresponding records are listed, ensuring that the data used for testing is consistent and reliable for further analysis and development.

Accomodation Table

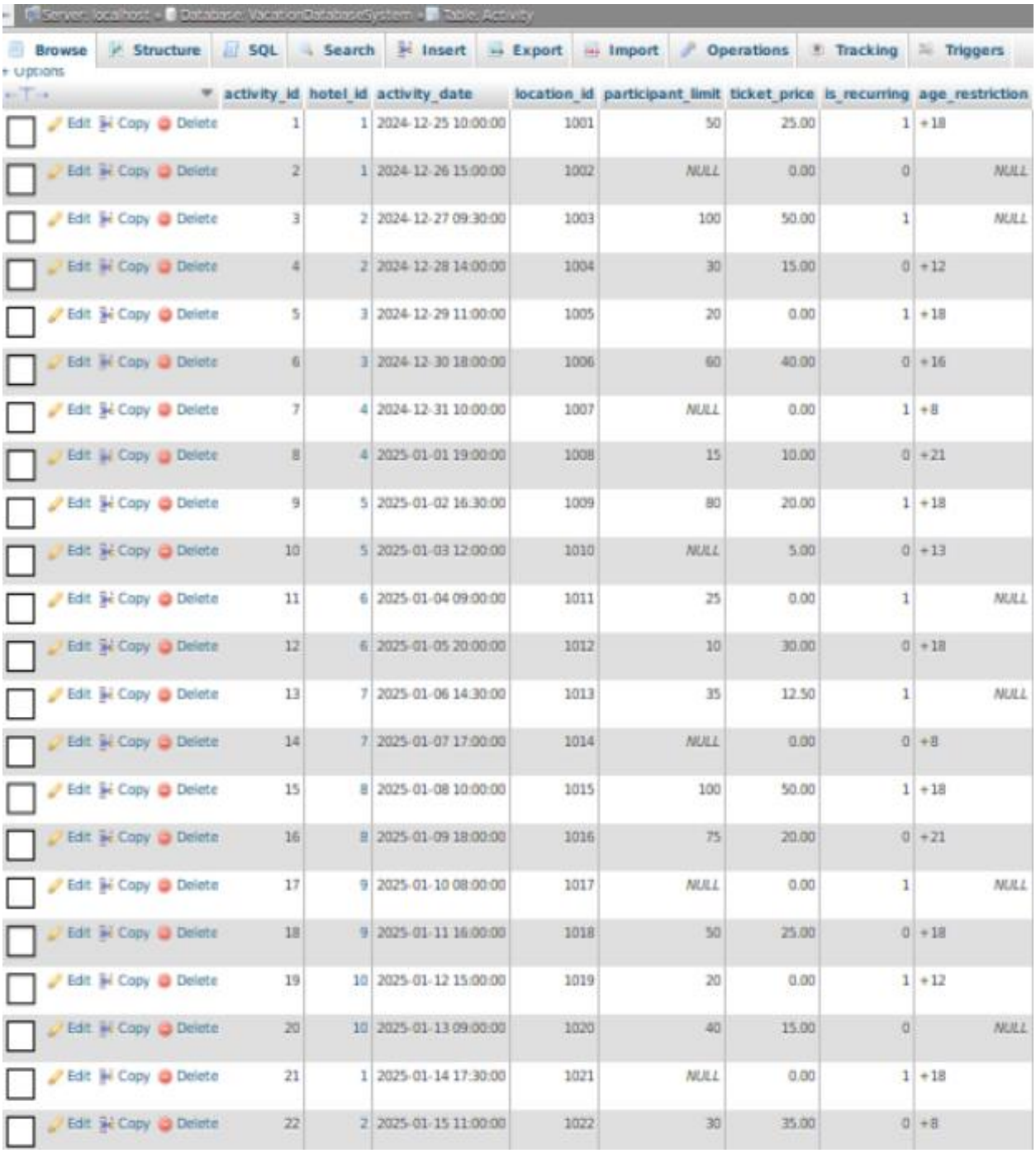


Server: localhost » Database: VacationDatabaseSystem » Table: Accommodation

Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Tracking | Triggers

	accommodation_id	customer_id	hotel_id	check_in_date	check_out_date	total_price	payment_type	room_type
Edit Copy Delete	1	1	1	2024-12-01	2024-12-05	800.00	Credit	Standart
Edit Copy Delete	2	2	2	2024-12-03	2024-12-07	480.00	Bank Transfer	Standart
Edit Copy Delete	3	3	3	2024-12-01	2024-12-05	500.00		Standart
Edit Copy Delete	4	4	4	2024-12-07	2025-12-10	900.00	Cash	Standart
Edit Copy Delete	5	5	5	2024-12-09	2024-12-12	750.00	Cash	Standart
Edit Copy Delete	6	6	6	2024-12-10	2024-12-15	800.00		Standart
Edit Copy Delete	7	7	7	2024-12-13	2024-12-17	1000.00	Credit	Standart
Edit Copy Delete	8	8	8	2024-12-20	2024-12-25	1000.00		Standart
Edit Copy Delete	9	9	9	2024-12-17	2024-12-21	920.00	Bank Transfer	Standart
Edit Copy Delete	10	10	10	2024-12-19	2024-12-22	1020.00	Cash	Standart
Edit Copy Delete	11	11	11	2024-12-21	2024-12-24	950.00	Bank Transfer	Standart
Edit Copy Delete	12	12	12	2024-12-23	2024-12-27	1100.00	Credit	Standart
Edit Copy Delete	13	13	13	2024-12-25	2024-12-29	780.00	Cash	Standart
Edit Copy Delete	14	14	14	2024-12-27	2024-12-30	860.00	Credit	Standart
Edit Copy Delete	15	15	15	2024-12-29	2025-01-02	1200.00	Cash	Standart
Edit Copy Delete	16	16	16	2024-12-31	2025-01-03	880.00	Credit	Standart
Edit Copy Delete	17	17	17	2025-01-02	2025-01-05	940.00	Bank Transfer	Standart
Edit Copy Delete	18	18	18	2025-01-04	2025-01-07	1120.00	Cash	Standart
Edit Copy Delete	19	19	19	2025-01-06	2025-01-09	1300.00	Bank Transfer	Standart
Edit Copy Delete	20	20	20	2025-01-08	2025-01-11	1050.00	Cash	Standart
Edit Copy Delete	21	21	21	2025-01-10	2025-01-13	950.00	Cash	Standart
Edit Copy Delete	22	22	22	2025-01-12	2025-01-15	1150.00	Credit	Standart

Activity Table



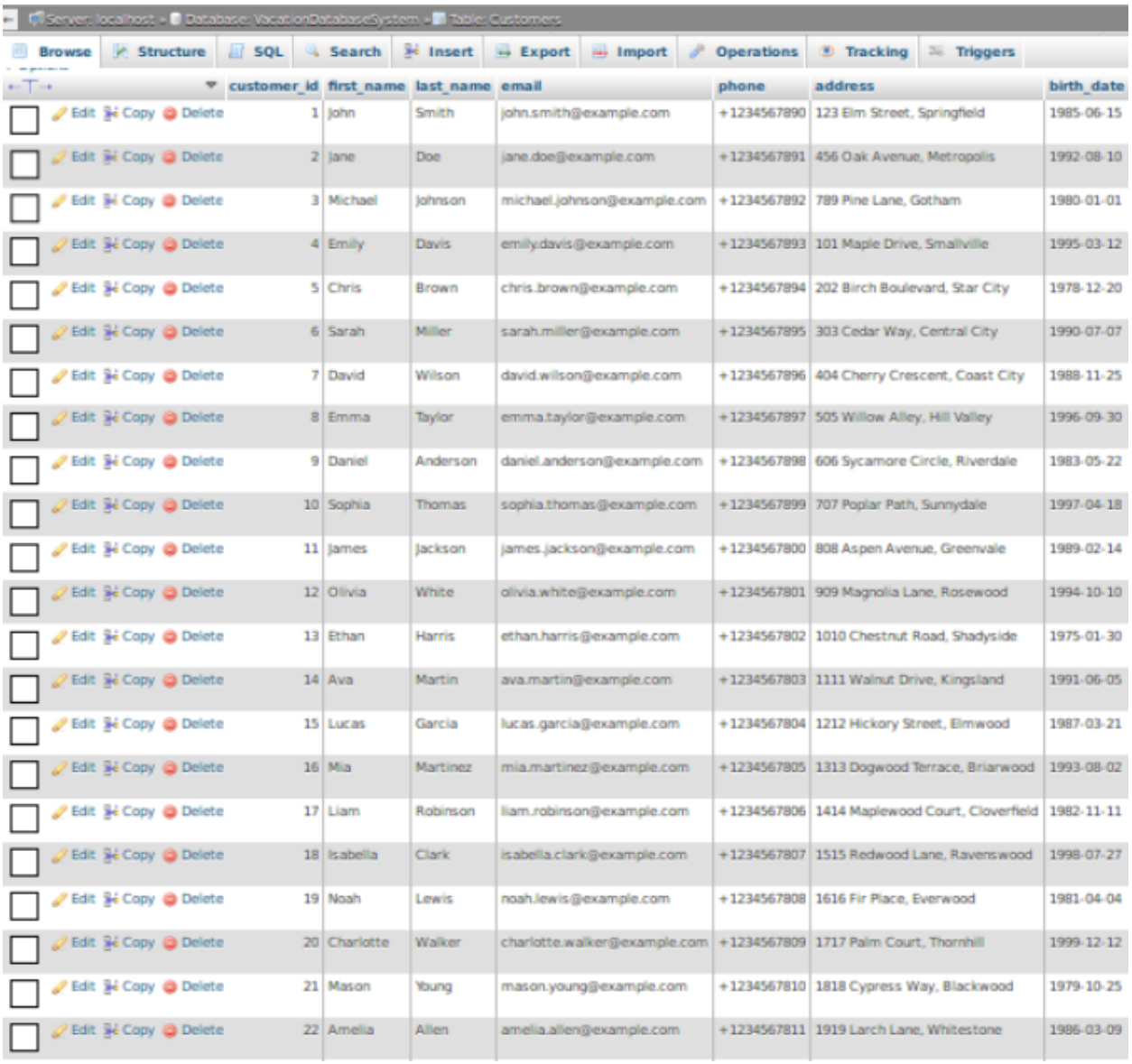
Server: localhost » Database: VacationDatabaseSystem » Table: Activity

Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Tracking | Triggers

+ Options

	activity_id	hotel_id	activity_date	location_id	participant_limit	ticket_price	is_recurring	age_restriction
Edit Copy Delete	1	1	2024-12-25 10:00:00	1001	50	25.00	1	+18
Edit Copy Delete	2	1	2024-12-26 15:00:00	1002	NULL	0.00	0	NULL
Edit Copy Delete	3	2	2024-12-27 09:30:00	1003	100	50.00	1	NULL
Edit Copy Delete	4	2	2024-12-28 14:00:00	1004	30	15.00	0	+12
Edit Copy Delete	5	3	2024-12-29 11:00:00	1005	20	0.00	1	+18
Edit Copy Delete	6	3	2024-12-30 18:00:00	1006	60	40.00	0	+16
Edit Copy Delete	7	4	2024-12-31 10:00:00	1007	NULL	0.00	1	+8
Edit Copy Delete	8	4	2025-01-01 19:00:00	1008	15	10.00	0	+21
Edit Copy Delete	9	5	2025-01-02 16:30:00	1009	80	20.00	1	+18
Edit Copy Delete	10	5	2025-01-03 12:00:00	1010	NULL	5.00	0	+13
Edit Copy Delete	11	6	2025-01-04 09:00:00	1011	25	0.00	1	NULL
Edit Copy Delete	12	6	2025-01-05 20:00:00	1012	10	30.00	0	+18
Edit Copy Delete	13	7	2025-01-06 14:30:00	1013	35	12.50	1	NULL
Edit Copy Delete	14	7	2025-01-07 17:00:00	1014	NULL	0.00	0	+8
Edit Copy Delete	15	8	2025-01-08 10:00:00	1015	100	50.00	1	+18
Edit Copy Delete	16	8	2025-01-09 18:00:00	1016	75	20.00	0	+21
Edit Copy Delete	17	9	2025-01-10 08:00:00	1017	NULL	0.00	1	NULL
Edit Copy Delete	18	9	2025-01-11 16:00:00	1018	50	25.00	0	+18
Edit Copy Delete	19	10	2025-01-12 15:00:00	1019	20	0.00	1	+12
Edit Copy Delete	20	10	2025-01-13 09:00:00	1020	40	15.00	0	NULL
Edit Copy Delete	21	1	2025-01-14 17:30:00	1021	NULL	0.00	1	+18
Edit Copy Delete	22	2	2025-01-15 11:00:00	1022	30	35.00	0	+8

Customer Table



Server: localhost » Database: VacationDatabaseSystem » Table: Customers

Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Tracking | Triggers

	customer_id	first_name	last_name	email	phone	address	birth_date
Edit Copy Delete	1	John	Smith	john.smith@example.com	+1234567890	123 Elm Street, Springfield	1985-06-15
Edit Copy Delete	2	Jane	Doe	jane.doe@example.com	+1234567891	456 Oak Avenue, Metropolis	1992-08-10
Edit Copy Delete	3	Michael	Johnson	michael.johnson@example.com	+1234567892	789 Pine Lane, Gotham	1980-01-01
Edit Copy Delete	4	Emily	Davis	emily.davis@example.com	+1234567893	101 Maple Drive, Smallville	1995-03-12
Edit Copy Delete	5	Chris	Brown	chris.brown@example.com	+1234567894	202 Birch Boulevard, Star City	1978-12-20
Edit Copy Delete	6	Sarah	Miller	sarah.miller@example.com	+1234567895	303 Cedar Way, Central City	1990-07-07
Edit Copy Delete	7	David	Wilson	david.wilson@example.com	+1234567896	404 Cherry Crescent, Coast City	1988-11-25
Edit Copy Delete	8	Emma	Taylor	emma.taylor@example.com	+1234567897	505 Willow Alley, Hill Valley	1996-09-30
Edit Copy Delete	9	Daniel	Anderson	daniel.anderson@example.com	+1234567898	606 Sycamore Circle, Riverdale	1983-05-22
Edit Copy Delete	10	Sophia	Thomas	sophia.thomas@example.com	+1234567899	707 Poplar Path, Sunnydale	1997-04-18
Edit Copy Delete	11	James	Jackson	james.jackson@example.com	+1234567800	808 Aspen Avenue, Greenvale	1989-02-14
Edit Copy Delete	12	Olivia	White	olivia.white@example.com	+1234567801	909 Magnolia Lane, Rosewood	1994-10-10
Edit Copy Delete	13	Ethan	Harris	ethan.harris@example.com	+1234567802	1010 Chestnut Road, Shadyside	1975-01-30
Edit Copy Delete	14	Ava	Martin	ava.martin@example.com	+1234567803	1111 Walnut Drive, Kingsland	1991-06-05
Edit Copy Delete	15	Lucas	Garcia	lucas.garcia@example.com	+1234567804	1212 Hickory Street, Elmwood	1987-03-21
Edit Copy Delete	16	Mia	Martinez	mia.martinez@example.com	+1234567805	1313 Dogwood Terrace, Briarwood	1993-08-02
Edit Copy Delete	17	Liam	Robinson	liam.robinson@example.com	+1234567806	1414 Maplewood Court, Cloverfield	1982-11-11
Edit Copy Delete	18	Isabella	Clark	isabella.clark@example.com	+1234567807	1515 Redwood Lane, Ravenswood	1998-07-27
Edit Copy Delete	19	Noah	Lewis	noah.lewis@example.com	+1234567808	1616 Fir Place, Everwood	1981-04-04
Edit Copy Delete	20	Charlotte	Walker	charlotte.walker@example.com	+1234567809	1717 Palm Court, Thornhill	1999-12-12
Edit Copy Delete	21	Mason	Young	mason.young@example.com	+1234567810	1818 Cypress Way, Blackwood	1979-10-25
Edit Copy Delete	22	Amelia	Allen	amelia.allen@example.com	+1234567811	1919 Larch Lane, Whitestone	1986-03-09

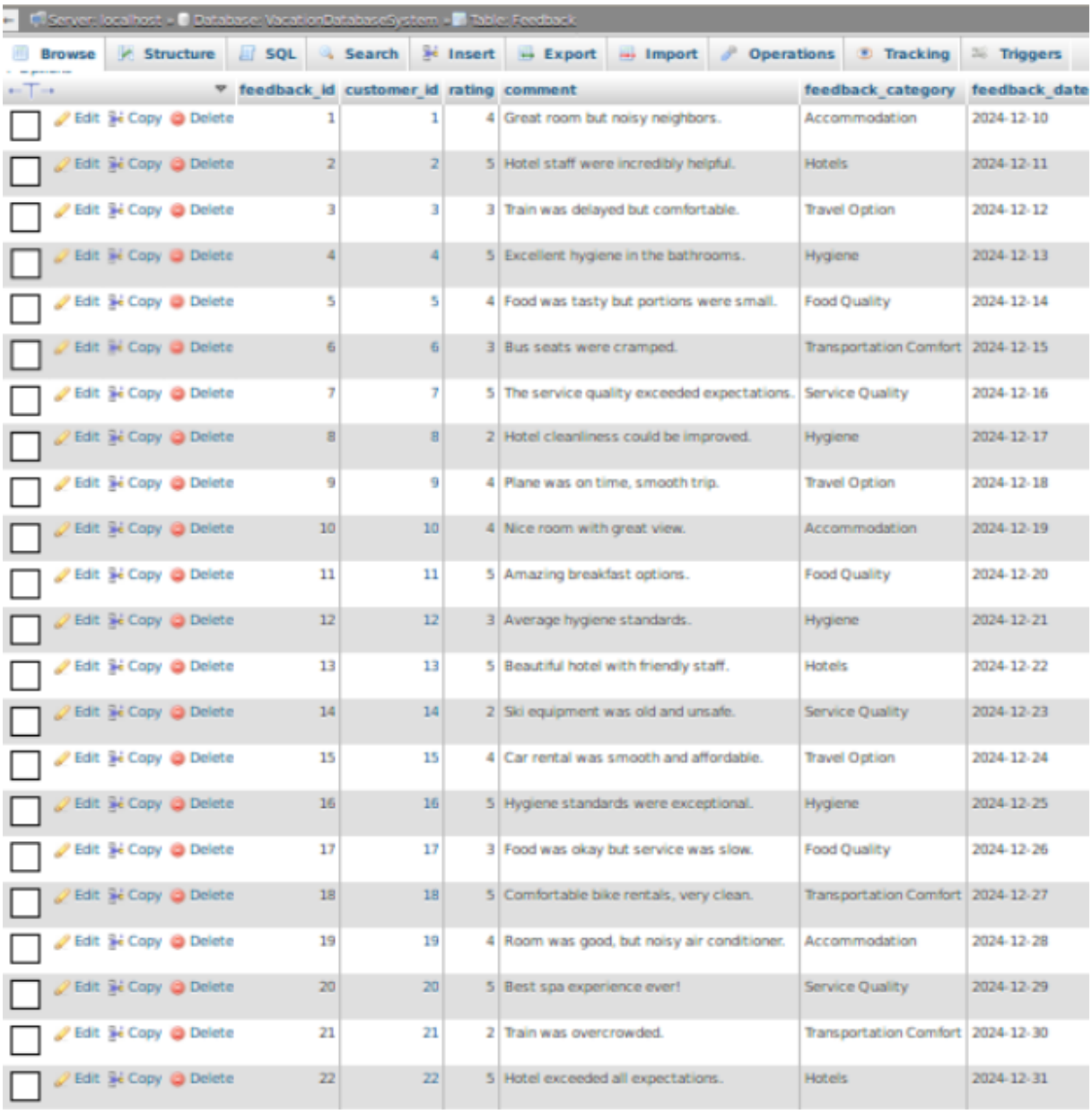
Destination Table

Browse | Structure | SQL | Search | Insert | Export | Import

Options

	location_id	city	country	continent
Edit Copy Delete	1001	Paris	France	Europe
Edit Copy Delete	1002	Malè	Maldives	Asia
Edit Copy Delete	1003	Aspen	United States	North America
Edit Copy Delete	1004	Tokyo	Japan	Asia
Edit Copy Delete	1005	Bali	Indonesia	Asia
Edit Copy Delete	1006	Zurich	Switzerland	Europe
Edit Copy Delete	1007	Cape Town	South Africa	Africa
Edit Copy Delete	1008	Rio de Janeiro	Brazil	South America
Edit Copy Delete	1009	Sydney	Australia	Oceania
Edit Copy Delete	1010	London	United Kingdom	Europe
Edit Copy Delete	1011	New York	United States	North America
Edit Copy Delete	1012	Rome	Italy	Europe
Edit Copy Delete	1013	Bangkok	Thailand	Asia
Edit Copy Delete	1014	Dubai	United Arab Emirates	Asia
Edit Copy Delete	1015	Hong Kong	China	Asia
Edit Copy Delete	1016	Singapore	Singapore	Asia
Edit Copy Delete	1017	Istanbul	Türkiye	Europe
Edit Copy Delete	1018	Moscow	Russia	Europe
Edit Copy Delete	1019	Vancouver	Canada	North America
Edit Copy Delete	1020	Buenos Aires	Argentina	South America
Edit Copy Delete	1021	Helsinki	Finland	Europe
Edit Copy Delete	1022	Vienna	Austria	Europe

Feedback Table

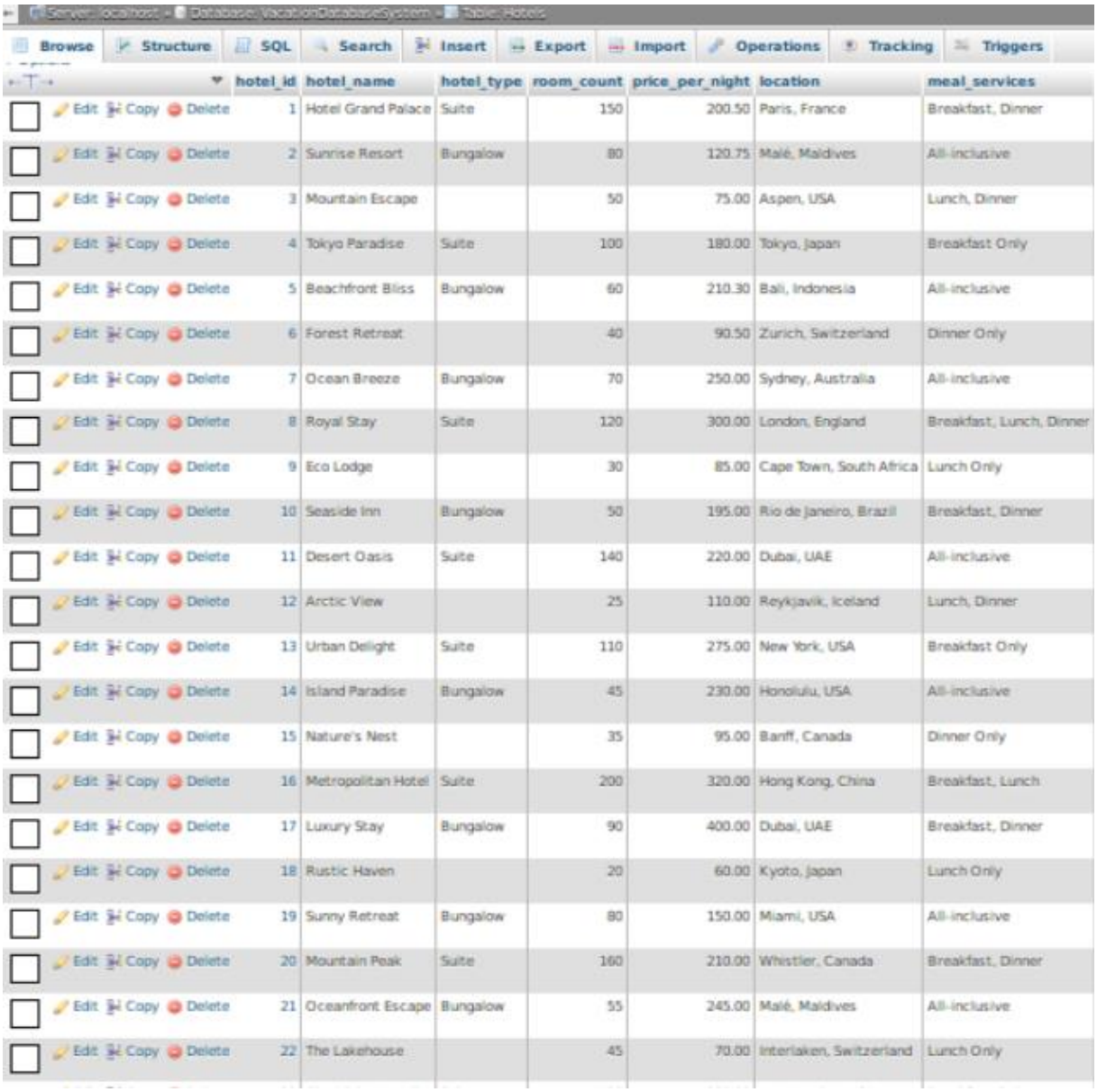


Server: localhost » Database: VacationDatabaseSystem » Table: Feedback

Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Tracking | Triggers

	feedback_id	customer_id	rating	comment	feedback_category	feedback_date
Edit Copy Delete	1	1	4	Great room but noisy neighbors.	Accommodation	2024-12-10
Edit Copy Delete	2	2	5	Hotel staff were incredibly helpful.	Hotels	2024-12-11
Edit Copy Delete	3	3	3	Train was delayed but comfortable.	Travel Option	2024-12-12
Edit Copy Delete	4	4	5	Excellent hygiene in the bathrooms.	Hygiene	2024-12-13
Edit Copy Delete	5	5	4	Food was tasty but portions were small.	Food Quality	2024-12-14
Edit Copy Delete	6	6	3	Bus seats were cramped.	Transportation Comfort	2024-12-15
Edit Copy Delete	7	7	5	The service quality exceeded expectations.	Service Quality	2024-12-16
Edit Copy Delete	8	8	2	Hotel cleanliness could be improved.	Hygiene	2024-12-17
Edit Copy Delete	9	9	4	Plane was on time, smooth trip.	Travel Option	2024-12-18
Edit Copy Delete	10	10	4	Nice room with great view.	Accommodation	2024-12-19
Edit Copy Delete	11	11	5	Amazing breakfast options.	Food Quality	2024-12-20
Edit Copy Delete	12	12	3	Average hygiene standards.	Hygiene	2024-12-21
Edit Copy Delete	13	13	5	Beautiful hotel with friendly staff.	Hotels	2024-12-22
Edit Copy Delete	14	14	2	Ski equipment was old and unsafe.	Service Quality	2024-12-23
Edit Copy Delete	15	15	4	Car rental was smooth and affordable.	Travel Option	2024-12-24
Edit Copy Delete	16	16	5	Hygiene standards were exceptional.	Hygiene	2024-12-25
Edit Copy Delete	17	17	3	Food was okay but service was slow.	Food Quality	2024-12-26
Edit Copy Delete	18	18	5	Comfortable bike rentals, very clean.	Transportation Comfort	2024-12-27
Edit Copy Delete	19	19	4	Room was good, but noisy air conditioner.	Accommodation	2024-12-28
Edit Copy Delete	20	20	5	Best spa experience ever!	Service Quality	2024-12-29
Edit Copy Delete	21	21	2	Train was overcrowded.	Transportation Comfort	2024-12-30
Edit Copy Delete	22	22	5	Hotel exceeded all expectations.	Hotels	2024-12-31

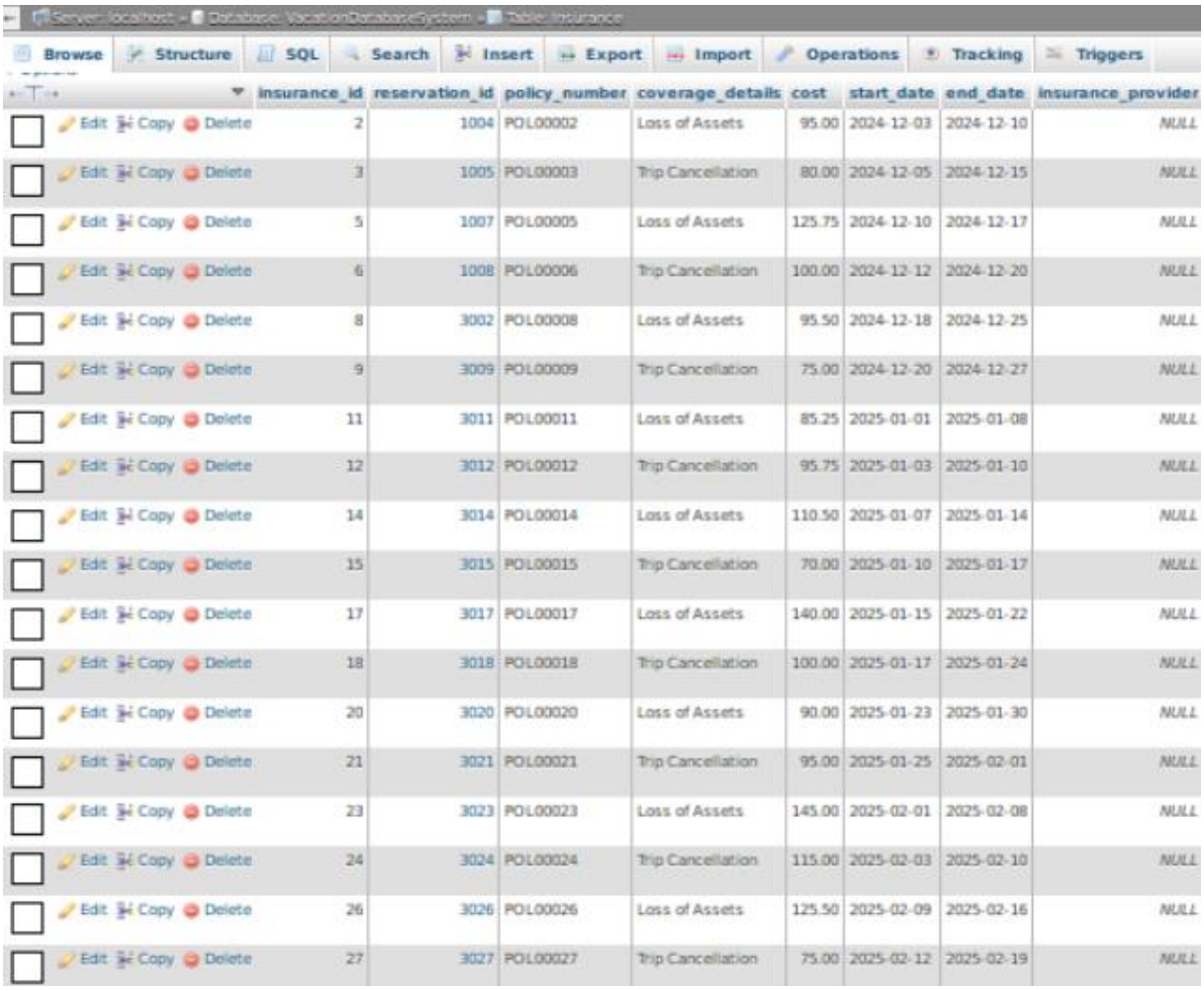
Hotel Table



Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Tracking | Triggers

	hotel_id	hotel_name	hotel_type	room_count	price_per_night	location	meal_services
Edit Copy Delete	1	Hotel Grand Palace	Suite	150	200.50	Paris, France	Breakfast, Dinner
Edit Copy Delete	2	Sunrise Resort	Bungalow	80	120.75	Malé, Maldives	All-inclusive
Edit Copy Delete	3	Mountain Escape		50	75.00	Aspen, USA	Lunch, Dinner
Edit Copy Delete	4	Tokyo Paradise	Suite	100	180.00	Tokyo, Japan	Breakfast Only
Edit Copy Delete	5	Beachfront Bliss	Bungalow	60	210.30	Bali, Indonesia	All-inclusive
Edit Copy Delete	6	Forest Retreat		40	90.50	Zurich, Switzerland	Dinner Only
Edit Copy Delete	7	Ocean Breeze	Bungalow	70	250.00	Sydney, Australia	All-inclusive
Edit Copy Delete	8	Royal Stay	Suite	120	300.00	London, England	Breakfast, Lunch, Dinner
Edit Copy Delete	9	Eco Lodge		30	85.00	Cape Town, South Africa	Lunch Only
Edit Copy Delete	10	Seaside Inn	Bungalow	50	195.00	Rio de Janeiro, Brazil	Breakfast, Dinner
Edit Copy Delete	11	Desert Oasis	Suite	140	220.00	Dubai, UAE	All-inclusive
Edit Copy Delete	12	Arctic View		25	110.00	Reykjavik, Iceland	Lunch, Dinner
Edit Copy Delete	13	Urban Delight	Suite	110	275.00	New York, USA	Breakfast Only
Edit Copy Delete	14	Island Paradise	Bungalow	45	230.00	Honolulu, USA	All-inclusive
Edit Copy Delete	15	Nature's Nest		35	95.00	Banff, Canada	Dinner Only
Edit Copy Delete	16	Metropolitan Hotel	Suite	200	320.00	Hong Kong, China	Breakfast, Lunch
Edit Copy Delete	17	Luxury Stay	Bungalow	90	400.00	Dubai, UAE	Breakfast, Dinner
Edit Copy Delete	18	Rustic Haven		20	60.00	Kyoto, Japan	Lunch Only
Edit Copy Delete	19	Sunny Retreat	Bungalow	80	150.00	Miami, USA	All-inclusive
Edit Copy Delete	20	Mountain Peak	Suite	160	210.00	Whistler, Canada	Breakfast, Dinner
Edit Copy Delete	21	Oceanfront Escape	Bungalow	55	245.00	Malé, Maldives	All-inclusive
Edit Copy Delete	22	The Lakehouse		45	70.00	Interlaken, Switzerland	Lunch Only

Insurance Table



Server: localhost » Database: VacationDatabaseSystem » Table: insurance

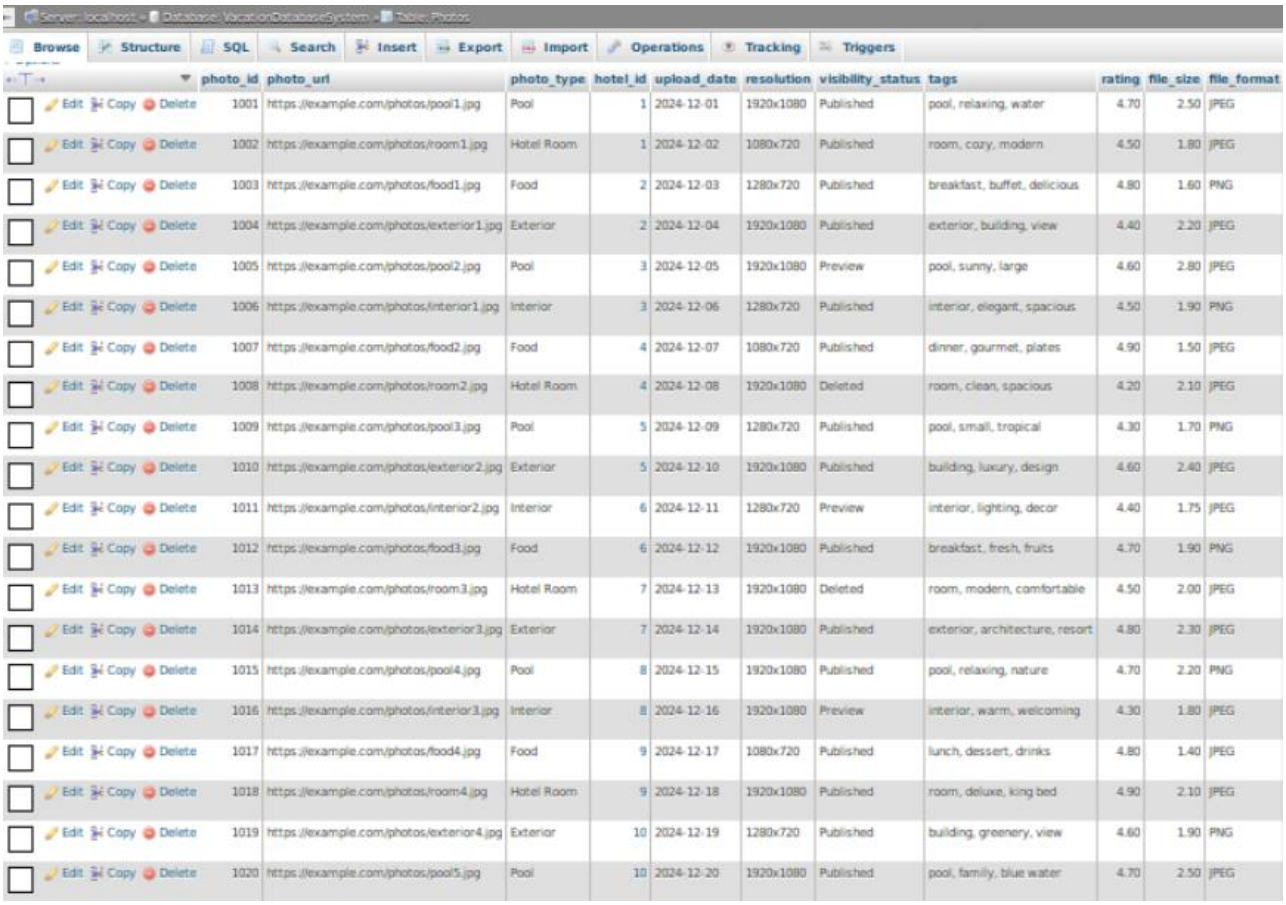
Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Tracking | Triggers

	insurance_id	reservation_id	policy_number	coverage_details	cost	start_date	end_date	insurance_provider
Edit Copy Delete	2	1004	POL00002	Loss of Assets	95.00	2024-12-03	2024-12-10	NULL
Edit Copy Delete	3	1005	POL00003	Trip Cancellation	80.00	2024-12-05	2024-12-15	NULL
Edit Copy Delete	5	1007	POL00005	Loss of Assets	125.75	2024-12-10	2024-12-17	NULL
Edit Copy Delete	6	1008	POL00006	Trip Cancellation	100.00	2024-12-12	2024-12-20	NULL
Edit Copy Delete	8	3002	POL00008	Loss of Assets	95.50	2024-12-18	2024-12-25	NULL
Edit Copy Delete	9	3009	POL00009	Trip Cancellation	75.00	2024-12-20	2024-12-27	NULL
Edit Copy Delete	11	3011	POL00011	Loss of Assets	85.25	2025-01-01	2025-01-08	NULL
Edit Copy Delete	12	3012	POL00012	Trip Cancellation	95.75	2025-01-03	2025-01-10	NULL
Edit Copy Delete	14	3014	POL00014	Loss of Assets	110.50	2025-01-07	2025-01-14	NULL
Edit Copy Delete	15	3015	POL00015	Trip Cancellation	70.00	2025-01-10	2025-01-17	NULL
Edit Copy Delete	17	3017	POL00017	Loss of Assets	140.00	2025-01-15	2025-01-22	NULL
Edit Copy Delete	18	3018	POL00018	Trip Cancellation	100.00	2025-01-17	2025-01-24	NULL
Edit Copy Delete	20	3020	POL00020	Loss of Assets	90.00	2025-01-23	2025-01-30	NULL
Edit Copy Delete	21	3021	POL00021	Trip Cancellation	95.00	2025-01-25	2025-02-01	NULL
Edit Copy Delete	23	3023	POL00023	Loss of Assets	145.00	2025-02-01	2025-02-08	NULL
Edit Copy Delete	24	3024	POL00024	Trip Cancellation	115.00	2025-02-03	2025-02-10	NULL
Edit Copy Delete	26	3026	POL00026	Loss of Assets	125.50	2025-02-09	2025-02-16	NULL
Edit Copy Delete	27	3027	POL00027	Trip Cancellation	75.00	2025-02-12	2025-02-19	NULL

Past Reservation Table

reservation_id	customer_id	accommodation_id	reservation_date	status	log_date	action_type
7001	1	3	2024-07-08	Active	2025-01-09 19:23:02	
7002	2	3	2024-01-05	No Reservation Yet	2025-01-09 19:23:02	
7001	1	3	2024-07-08	Active	2025-01-09 19:25:50	UPDATE
7001	1	3	2024-07-08	Cancelled	2025-01-09 19:27:39	UPDATE
7001	1	3	2024-07-08	Cancelled	2025-01-09 19:28:14	DELETE

Photos Table



Server: localhost » Database: VacationDatabaseSystem » Table: Photos

Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Tracking | Triggers

	photo_id	photo_url	photo_type	hotel_id	upload_date	resolution	visibility_status	tags	rating	file_size	file_format
Edit Copy Delete	1001	https://example.com/photos/pool1.jpg	Pool	1	2024-12-01	1920x1080	Published	pool, relaxing, water	4.70	2.50	JPEG
Edit Copy Delete	1002	https://example.com/photos/room1.jpg	Hotel Room	1	2024-12-02	1080x720	Published	room, cozy, modern	4.50	1.80	JPEG
Edit Copy Delete	1003	https://example.com/photos/food1.jpg	Food	2	2024-12-03	1280x720	Published	breakfast, buffet, delicious	4.80	1.60	PNG
Edit Copy Delete	1004	https://example.com/photos/exterior1.jpg	Exterior	2	2024-12-04	1920x1080	Published	exterior, building, view	4.40	2.20	JPEG
Edit Copy Delete	1005	https://example.com/photos/pool2.jpg	Pool	3	2024-12-05	1920x1080	Preview	pool, sunny, large	4.60	2.80	JPEG
Edit Copy Delete	1006	https://example.com/photos/interior1.jpg	Interior	3	2024-12-06	1280x720	Published	interior, elegant, spacious	4.50	1.90	PNG
Edit Copy Delete	1007	https://example.com/photos/food2.jpg	Food	4	2024-12-07	1080x720	Published	dinner, gourmet, plates	4.90	1.50	JPEG
Edit Copy Delete	1008	https://example.com/photos/room2.jpg	Hotel Room	4	2024-12-08	1920x1080	Deleted	room, clean, spacious	4.20	2.10	JPEG
Edit Copy Delete	1009	https://example.com/photos/pool3.jpg	Pool	5	2024-12-09	1280x720	Published	pool, small, tropical	4.30	1.70	PNG
Edit Copy Delete	1010	https://example.com/photos/exterior2.jpg	Exterior	5	2024-12-10	1920x1080	Published	building, luxury, design	4.60	2.40	JPEG
Edit Copy Delete	1011	https://example.com/photos/interior2.jpg	Interior	6	2024-12-11	1280x720	Preview	interior, lighting, decor	4.40	1.75	JPEG
Edit Copy Delete	1012	https://example.com/photos/food3.jpg	Food	6	2024-12-12	1920x1080	Published	breakfast, fresh, fruits	4.70	1.90	PNG
Edit Copy Delete	1013	https://example.com/photos/room3.jpg	Hotel Room	7	2024-12-13	1920x1080	Deleted	room, modern, comfortable	4.50	2.00	JPEG
Edit Copy Delete	1014	https://example.com/photos/exterior3.jpg	Exterior	7	2024-12-14	1920x1080	Published	exterior, architecture, resort	4.80	2.30	JPEG
Edit Copy Delete	1015	https://example.com/photos/pool4.jpg	Pool	8	2024-12-15	1920x1080	Published	pool, relaxing, nature	4.70	2.20	PNG
Edit Copy Delete	1016	https://example.com/photos/interior3.jpg	Interior	8	2024-12-16	1920x1080	Preview	interior, warm, welcoming	4.30	1.80	JPEG
Edit Copy Delete	1017	https://example.com/photos/food4.jpg	Food	9	2024-12-17	1080x720	Published	lunch, dessert, drinks	4.80	1.40	JPEG
Edit Copy Delete	1018	https://example.com/photos/room4.jpg	Hotel Room	9	2024-12-18	1920x1080	Published	room, deluxe, king bed	4.90	2.10	JPEG
Edit Copy Delete	1019	https://example.com/photos/exterior4.jpg	Exterior	10	2024-12-19	1280x720	Published	building, greenery, view	4.60	1.90	PNG
Edit Copy Delete	1020	https://example.com/photos/pool5.jpg	Pool	10	2024-12-20	1920x1080	Published	pool, family, blue water	4.70	2.50	JPEG

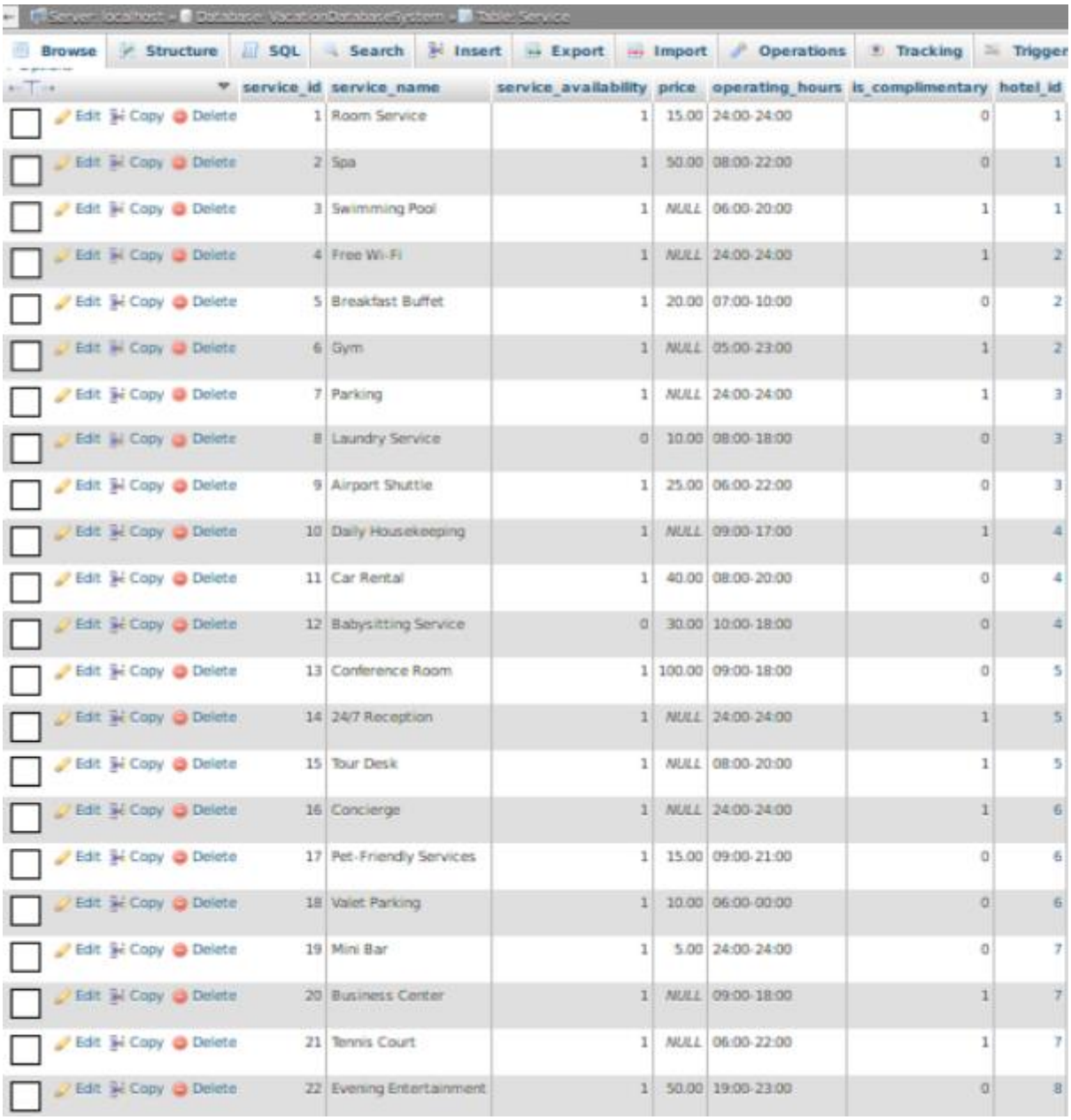
Reservation Table

Server: localhost » Database: VacationDatabaseSystem » Table: Reservations

Browse | Structure | SQL | Search | Insert | Export | Import | Operations

	reservation_id	customer_id	accommodation_id	reservation_date	status
Edit Copy Delete	1003	3	3	2024-12-03	No Reservation Yet
Edit Copy Delete	1004	4	4	2024-12-05	Cancelled
Edit Copy Delete	1005	5	5	2024-12-07	Active
Edit Copy Delete	1006	6	6	2024-12-09	No Reservation Yet
Edit Copy Delete	1007	7	7	2024-12-11	Active
Edit Copy Delete	1008	8	8	2024-12-13	No Reservation Yet
Edit Copy Delete	3001	1	5	2024-12-01	Active
Edit Copy Delete	3002	2	7	2024-12-03	Cancelled
Edit Copy Delete	3009	9	9	2024-12-17	Active
Edit Copy Delete	3010	10	10	2024-12-19	Cancelled
Edit Copy Delete	3011	11	12	2024-12-21	Active
Edit Copy Delete	3012	12	14	2024-12-23	Active
Edit Copy Delete	3013	13	11	2024-12-25	Cancelled
Edit Copy Delete	3014	14	15	2024-12-27	Active
Edit Copy Delete	3015	15	18	2024-12-29	Active
Edit Copy Delete	3016	16	20	2024-12-31	Cancelled
Edit Copy Delete	3017	17	16	2025-01-02	Active
Edit Copy Delete	3018	18	13	2025-01-04	Active
Edit Copy Delete	3019	19	17	2025-01-06	Cancelled
Edit Copy Delete	3020	20	19	2025-01-08	Active
Edit Copy Delete	3021	21	22	2025-01-10	Active
Edit Copy Delete	3022	22	24	2025-01-12	Cancelled

Service Table



Server: localhost » Database: VacationDatabaseSystem » Table: Service

Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Tracking | Trigger

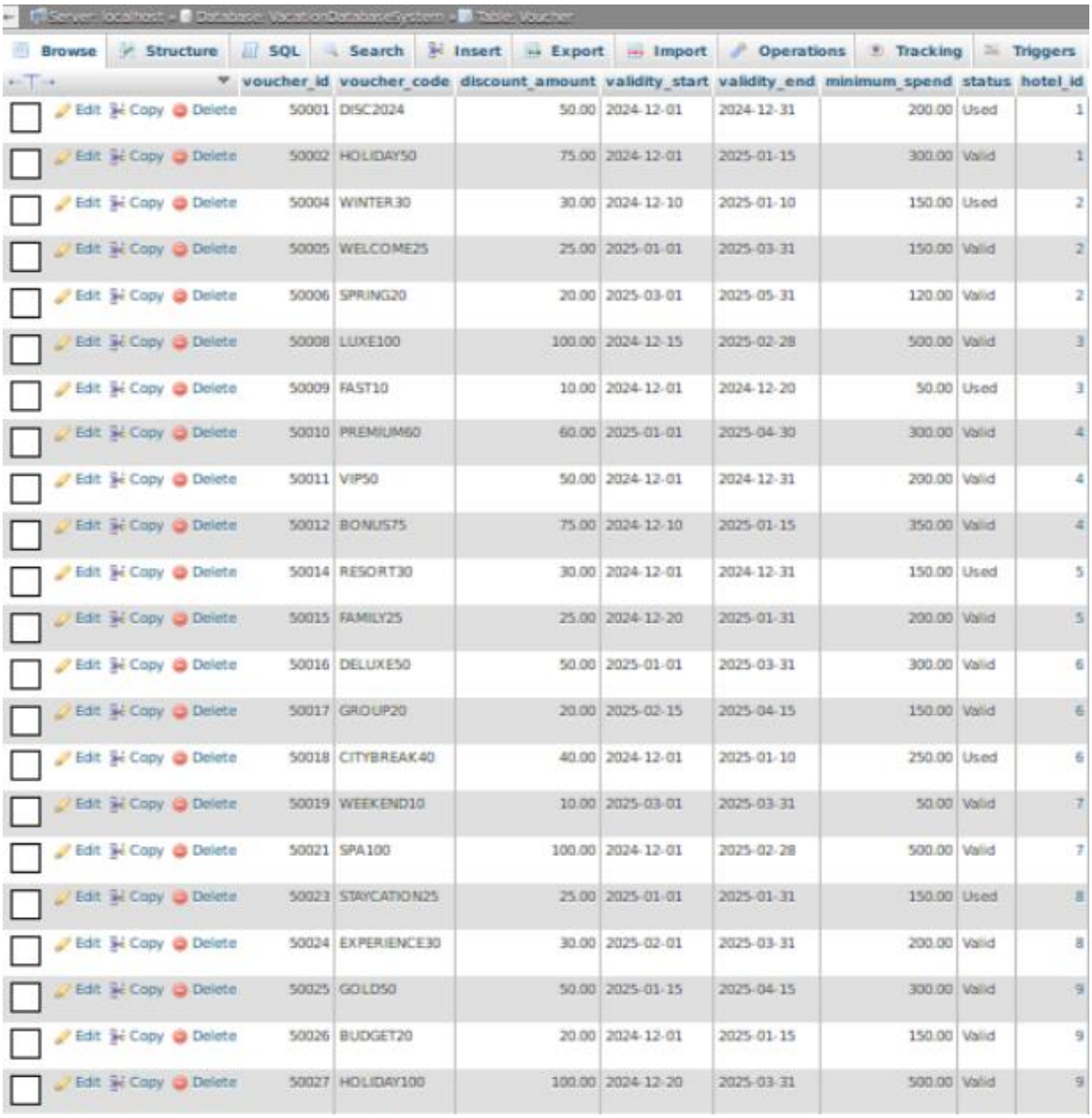
	service_id	service_name	service_availability	price	operating_hours	is_complimentary	hotel_id
Edit Copy Delete	1	Room Service	1	15.00	24:00-24:00	0	1
Edit Copy Delete	2	Spa	1	50.00	08:00-22:00	0	1
Edit Copy Delete	3	Swimming Pool	1	NULL	06:00-20:00	1	1
Edit Copy Delete	4	Free Wi-Fi	1	NULL	24:00-24:00	1	2
Edit Copy Delete	5	Breakfast Buffet	1	20.00	07:00-10:00	0	2
Edit Copy Delete	6	Gym	1	NULL	05:00-23:00	1	2
Edit Copy Delete	7	Parking	1	NULL	24:00-24:00	1	3
Edit Copy Delete	8	Laundry Service	0	10.00	08:00-18:00	0	3
Edit Copy Delete	9	Airport Shuttle	1	25.00	06:00-22:00	0	3
Edit Copy Delete	10	Daily Housekeeping	1	NULL	09:00-17:00	1	4
Edit Copy Delete	11	Car Rental	1	40.00	08:00-20:00	0	4
Edit Copy Delete	12	Babysitting Service	0	30.00	10:00-18:00	0	4
Edit Copy Delete	13	Conference Room	1	100.00	09:00-18:00	0	5
Edit Copy Delete	14	24/7 Reception	1	NULL	24:00-24:00	1	5
Edit Copy Delete	15	Tour Desk	1	NULL	08:00-20:00	1	5
Edit Copy Delete	16	Concierge	1	NULL	24:00-24:00	1	6
Edit Copy Delete	17	Pet-Friendly Services	1	15.00	09:00-21:00	0	6
Edit Copy Delete	18	Valet Parking	1	10.00	06:00-00:00	0	6
Edit Copy Delete	19	Mini Bar	1	5.00	24:00-24:00	0	7
Edit Copy Delete	20	Business Center	1	NULL	09:00-18:00	1	7
Edit Copy Delete	21	Tennis Court	1	NULL	06:00-22:00	1	7
Edit Copy Delete	22	Evening Entertainment	1	50.00	19:00-23:00	0	8

Transport Options Table

Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Track

	transport_id	transport_type	departure_point	destination_point	travel_duration	cost
Edit Copy Delete	2001	Plane	JFK Airport	Charles de Gaulle	7	480.00
Edit Copy Delete	2002	Bus	Berlin	Prague	4	45.00
Edit Copy Delete	2003	Train	Tokyo	Osaka	2	90.00
Edit Copy Delete	2004	Car	Los Angeles	San Francisco	5	70.00
Edit Copy Delete	2005	Bicycle	Amsterdam	Rotterdam	3	8.00
Edit Copy Delete	2006	Boat	Miami	Bahamas	6	280.00
Edit Copy Delete	2007	Plane	Heathrow	Dubai	6	430.00
Edit Copy Delete	2008	Train	Paris	Lyon	2	55.00
Edit Copy Delete	2009	Bus	Rome	Florence	3	35.00
Edit Copy Delete	2010	Car	New York	Boston	4	85.00
Edit Copy Delete	2011	Bicycle	Seoul	Incheon	1	12.00
Edit Copy Delete	2012	Boat	Sydney	Hobart	8	340.00
Edit Copy Delete	2013	Plane	Tokyo	Seoul	2	190.00
Edit Copy Delete	2014	Train	Berlin	Hamburg	2	45.00
Edit Copy Delete	2015	Bus	Madrid	Barcelona	5	40.00
Edit Copy Delete	2016	Car	Las Vegas	Los Angeles	4	80.00
Edit Copy Delete	2017	Bicycle	Copenhagen	Malmö	2	10.00
Edit Copy Delete	2018	Boat	Vancouver	Victoria	6	240.00
Edit Copy Delete	2019	Plane	Singapore	Bangkok	2	170.00
Edit Copy Delete	2020	Train	Vienna	Budapest	2	60.00
Edit Copy Delete	2021	Bus	Buenos Aires	Mendoza	8	50.00
Edit Copy Delete	2022	Car	Cape Town	Stellenbosch	1	25.00

Voucher Table



Server: localhost » Database: VacationDatabaseSystem » Table: Voucher

Browse | Structure | SQL | Search | Insert | Export | Import | Operations | Tracking | Triggers

	voucher_id	voucher_code	discount_amount	validity_start	validity_end	minimum_spend	status	hotel_id
Edit Copy Delete	50001	DISC2024	50.00	2024-12-01	2024-12-31	200.00	Used	1
Edit Copy Delete	50002	HOLIDAY50	75.00	2024-12-01	2025-01-15	300.00	Valid	1
Edit Copy Delete	50004	WINTER30	30.00	2024-12-10	2025-01-10	150.00	Used	2
Edit Copy Delete	50005	WELCOME25	25.00	2025-01-01	2025-03-31	150.00	Valid	2
Edit Copy Delete	50006	SPRING20	20.00	2025-03-01	2025-05-31	120.00	Valid	2
Edit Copy Delete	50008	LUXE100	100.00	2024-12-15	2025-02-28	500.00	Valid	3
Edit Copy Delete	50009	FAST10	10.00	2024-12-01	2024-12-20	50.00	Used	3
Edit Copy Delete	50010	PREMIUM60	60.00	2025-01-01	2025-04-30	300.00	Valid	4
Edit Copy Delete	50011	VIP50	50.00	2024-12-01	2024-12-31	200.00	Valid	4
Edit Copy Delete	50012	BONUS75	75.00	2024-12-10	2025-01-15	350.00	Valid	4
Edit Copy Delete	50014	RESORT30	30.00	2024-12-01	2024-12-31	150.00	Used	5
Edit Copy Delete	50015	FAMILY25	25.00	2024-12-20	2025-01-31	200.00	Valid	5
Edit Copy Delete	50016	DELUXE50	50.00	2025-01-01	2025-03-31	300.00	Valid	6
Edit Copy Delete	50017	GROUP20	20.00	2025-02-15	2025-04-15	150.00	Valid	6
Edit Copy Delete	50018	CITYBREAK40	40.00	2024-12-01	2025-01-10	250.00	Used	6
Edit Copy Delete	50019	WEEKEND10	10.00	2025-03-01	2025-03-31	50.00	Valid	7
Edit Copy Delete	50021	SPA100	100.00	2024-12-01	2025-02-28	500.00	Valid	7
Edit Copy Delete	50023	STAYCATION25	25.00	2025-01-01	2025-01-31	150.00	Used	8
Edit Copy Delete	50024	EXPERIENCE30	30.00	2025-02-01	2025-03-31	200.00	Valid	8
Edit Copy Delete	50025	GOLD50	50.00	2025-01-15	2025-04-15	300.00	Valid	9
Edit Copy Delete	50026	BUDGET20	20.00	2024-12-01	2025-01-15	150.00	Valid	9
Edit Copy Delete	50027	HOLIDAY100	100.00	2024-12-20	2025-03-31	500.00	Valid	9

Active Customers View Table

Foreign Key Table

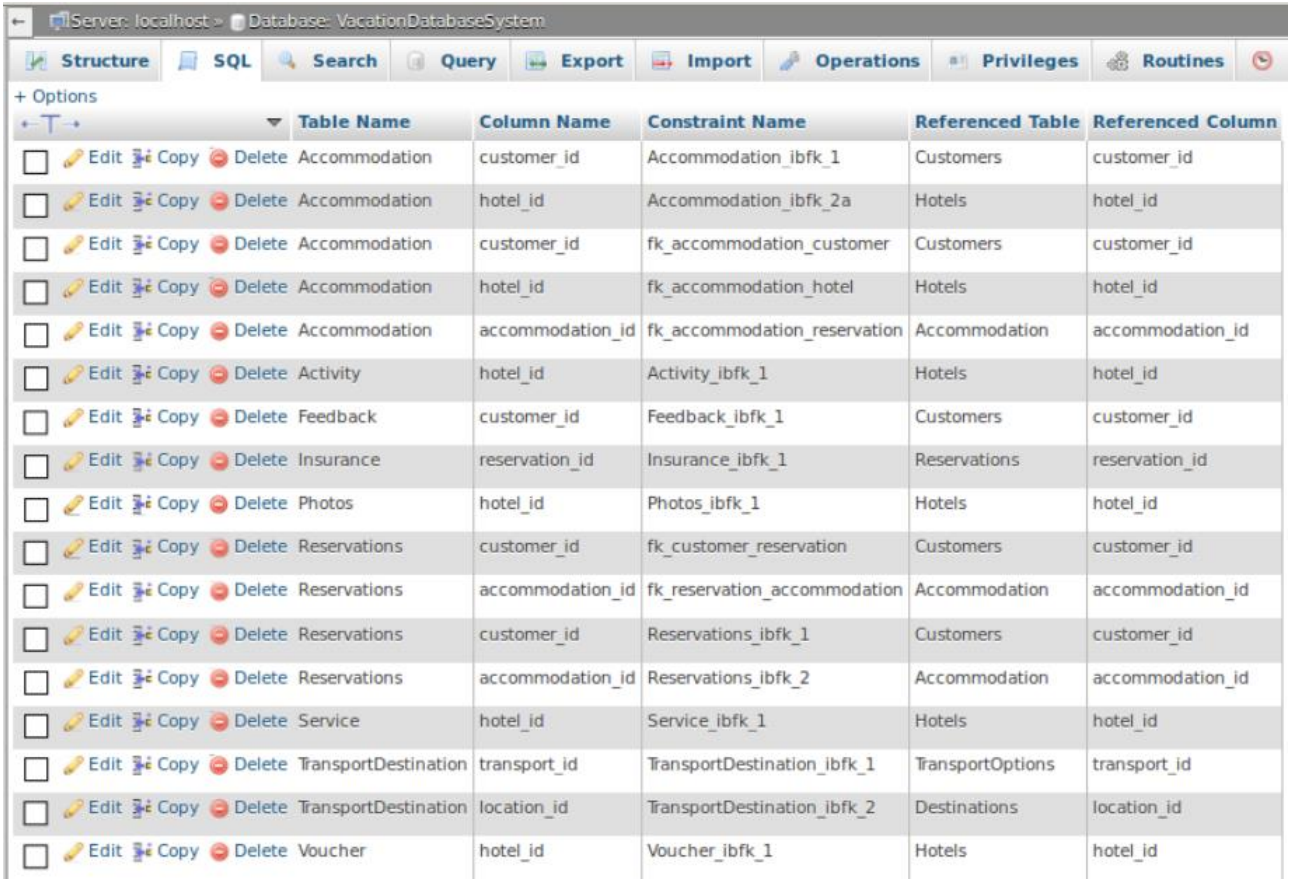


	Table Name	Column Name	Constraint Name	Referenced Table	Referenced Column
Edit Copy Delete	Accommodation	customer_id	Accommodation_ibfk_1	Customers	customer_id
Edit Copy Delete	Accommodation	hotel_id	Accommodation_ibfk_2a	Hotels	hotel_id
Edit Copy Delete	Accommodation	customer_id	fk_accommodation_customer	Customers	customer_id
Edit Copy Delete	Accommodation	hotel_id	fk_accommodation_hotel	Hotels	hotel_id
Edit Copy Delete	Accommodation	accommodation_id	fk_accommodation_reservation	Accommodation	accommodation_id
Edit Copy Delete	Activity	hotel_id	Activity_ibfk_1	Hotels	hotel_id
Edit Copy Delete	Feedback	customer_id	Feedback_ibfk_1	Customers	customer_id
Edit Copy Delete	Insurance	reservation_id	Insurance_ibfk_1	Reservations	reservation_id
Edit Copy Delete	Photos	hotel_id	Photos_ibfk_1	Hotels	hotel_id
Edit Copy Delete	Reservations	customer_id	fk_customer_reservation	Customers	customer_id
Edit Copy Delete	Reservations	accommodation_id	fk_reservation_accommodation	Accommodation	accommodation_id
Edit Copy Delete	Reservations	customer_id	Reservations_ibfk_1	Customers	customer_id
Edit Copy Delete	Reservations	accommodation_id	Reservations_ibfk_2	Accommodation	accommodation_id
Edit Copy Delete	Service	hotel_id	Service_ibfk_1	Hotels	hotel_id
Edit Copy Delete	TransportDestination	transport_id	TransportDestination_ibfk_1	TransportOptions	transport_id
Edit Copy Delete	TransportDestination	location_id	TransportDestination_ibfk_2	Destinations	location_id
Edit Copy Delete	Voucher	hotel_id	Voucher_ibfk_1	Hotels	hotel_id

Trigger

The "Past Reservation" table and its triggers are designed to keep a clear and detailed record of any changes made to the "Reservation" table. This functionality ensures that any modifications, including INSERT, UPDATE, and DELETE actions, are systematically logged for future analysis and reference By maintaining these records, the system helps track customer reservation histories and ensures the accuracy and reliability of the data for future analysis.

Triggers were designed to automatically log all changes made to the "Reservation" table into the "Past Reservation" table. This functionality supports the following objectives:

1. **Audit Trail Creation**: Tracks modifications to reservation records for accountability and operational insights.
2. **Historical Data Storage**: Maintains a record of customer reservation histories for review and analysis.
3. **Data Integrity**: Ensures a reliable mechanism for logging all relevant details of reservation changes.

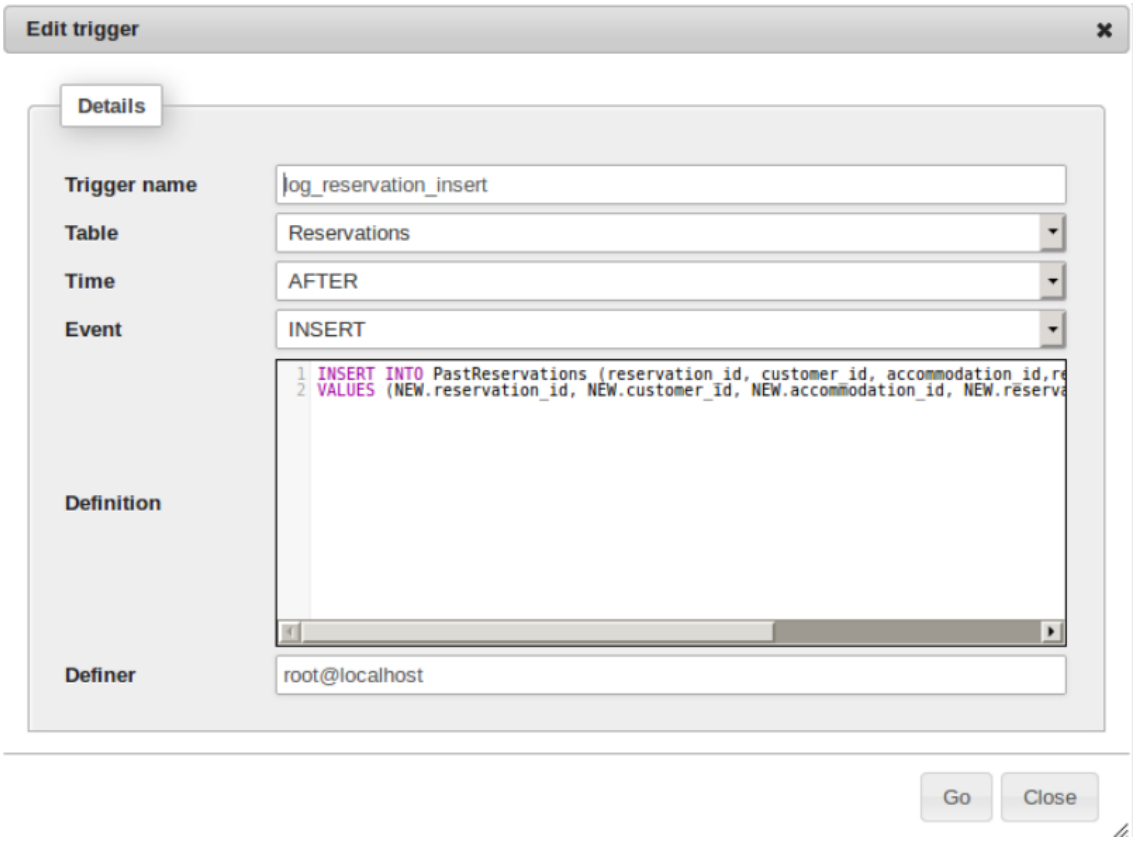
Benefits of the Trigger Implementation

- **Better Understanding of Customers:** Helps businesses gain insights into customer preferences and behaviors by analyzing reservation histories.

- **Improved Accountability:** Provides a dependable way to track and review changes in the system, ensuring transparency and meeting compliance requirements.
- **Simplified Reporting:** Makes it easier to create detailed reports and analyze trends or changes in reservations.

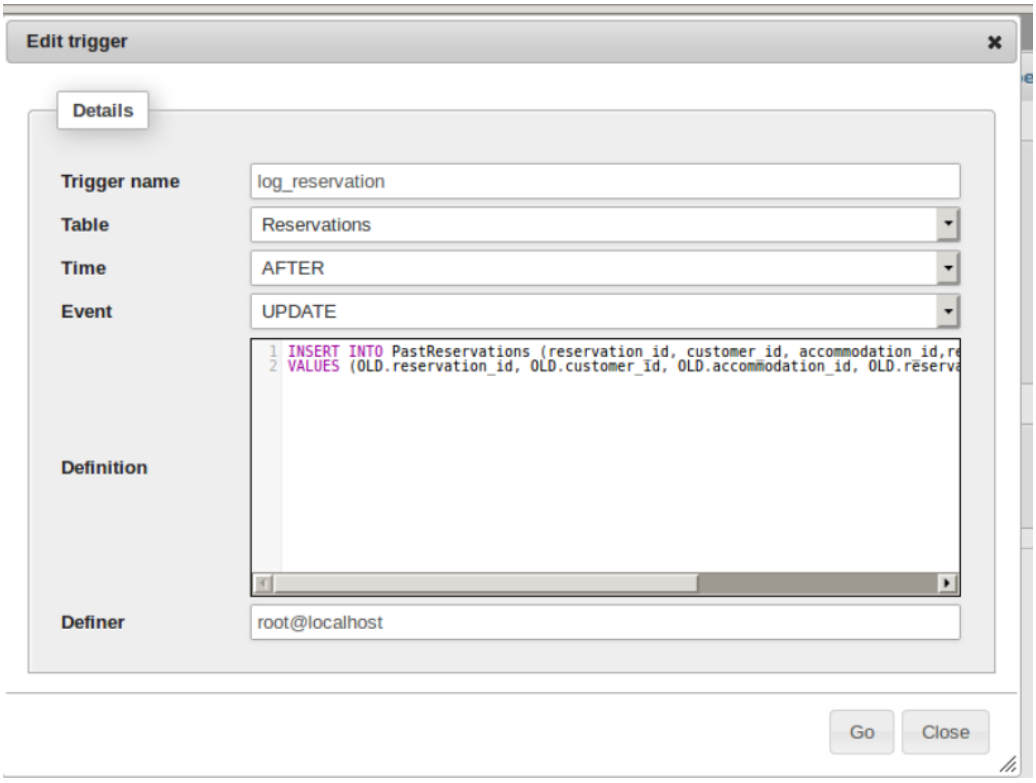
Trigger Details

1.INSERT TRIGGER



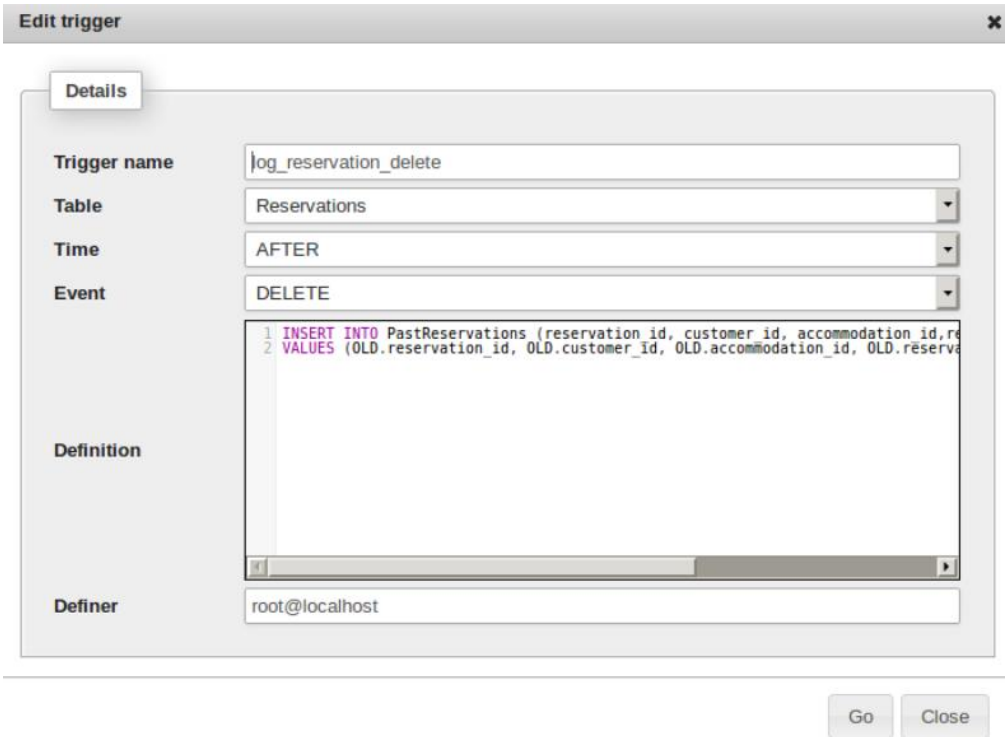
Captures details such as the reservation ID, customer ID, accommodation ID, reservation date, and assigns the status as "Active." The action type is recorded as "INSERT."

2. UPDATE TRIGGER



Records the updated details of a reservation, including the modified status. The action type is recorded as "UPDATE," and the log_date captures the precise timestamp of the change.

3.DELETE TRIGGER



Preserves details of deleted reservations, including reservation ID, customer ID, and

accommodation ID, while marking the action type as "DELETE."

The implementation of triggers for the "Past Reservation" table significantly enhances the database system's capabilities by ensuring all modifications to the "Reservation" table are logged comprehensively. This approach supports robust data analysis, operational accountability, and a higher level of system reliability.

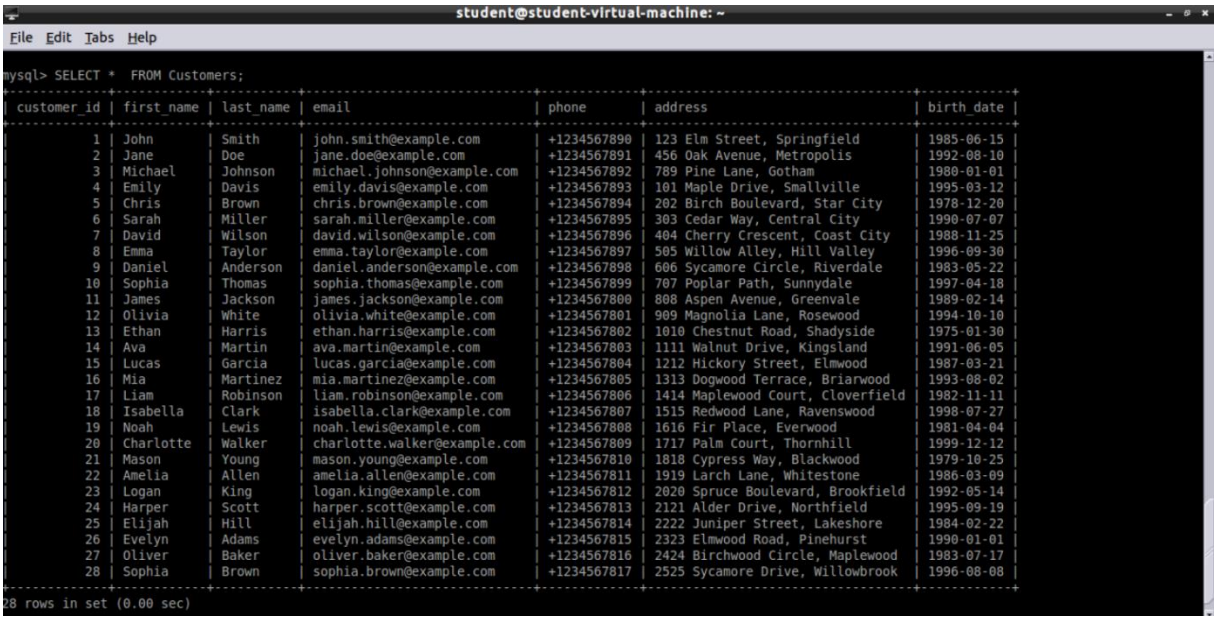
5. Query Execution

This section shows SQL operations performed on the tables. Each step is supported by screenshots from the terminal to demonstrate the execution and outcomes of the queries.

5.1 Queries and Results for Customer table

1. Viewing Existing Records

- **Query**: `SELECT * FROM Customers;`
- This query retrieves all records from the Customers table, columns such as customer_id, first_name, last_name, email, phone, address, and birth_date.
- **Output**:



customer_id	first_name	last_name	email	phone	address	birth_date
1	John	Smith	john.smith@example.com	+1234567890	123 Elm Street, Springfield	1985-06-15
2	Jane	Doe	jane.doe@example.com	+1234567891	456 Oak Avenue, Metropolis	1992-08-10
3	Michael	Johnson	michael.johnson@example.com	+1234567892	789 Pine Lane, Gotham	1980-01-01
4	Emily	Davis	emily.davis@example.com	+1234567893	101 Maple Drive, Smallville	1995-03-12
5	Chris	Brown	chris.brown@example.com	+1234567894	202 Birch Boulevard, Star City	1978-12-20
6	Sarah	Miller	sarah.miller@example.com	+1234567895	303 Cedar Way, Central City	1990-07-07
7	David	Wilson	david.wilson@example.com	+1234567896	404 Cherry Crescent, Coast City	1988-11-25
8	Emma	Taylor	emma.taylor@example.com	+1234567897	505 Willow Alley, Hill Valley	1996-09-30
9	Daniel	Anderson	daniel.anderson@example.com	+1234567898	606 Sycamore Circle, Riverdale	1983-05-22
10	Sophia	Thomas	sophia.thomas@example.com	+1234567899	707 Poplar Path, Sunnydale	1997-04-18
11	James	Jackson	james.jackson@example.com	+1234567800	808 Aspen Avenue, Greenvale	1989-02-14
12	Olivia	White	olivia.white@example.com	+1234567801	909 Magnolia Lane, Rosewood	1994-10-10
13	Ethan	Harris	ethan.harris@example.com	+1234567802	1010 Chestnut Road, Shadyside	1975-01-30
14	Ava	Martin	ava.martin@example.com	+1234567803	1111 Walnut Drive, Kingsland	1991-06-05
15	Lucas	Garcia	lucas.garcia@example.com	+1234567804	1212 Hickory Street, Elmwood	1987-03-21
16	Mia	Martinez	mia.martinez@example.com	+1234567805	1313 Dogwood Terrace, Briarwood	1993-08-02
17	Liam	Robinson	liam.robinson@example.com	+1234567806	1414 Maplewood Court, Cloverfield	1982-11-11
18	Isabella	Clark	isabella.clark@example.com	+1234567807	1515 Redwood Lane, Ravenswood	1998-07-27
19	Noah	Lewis	noah.lewis@example.com	+1234567808	1616 Fir Place, Everwood	1981-04-04
20	Charlotte	Walker	charlotte.walker@example.com	+1234567809	1717 Palm Court, Thornhill	1999-12-12
21	Mason	Young	mason.young@example.com	+1234567810	1818 Cypress Way, Blackwood	1979-10-25
22	Amelia	Allen	amelia.allen@example.com	+1234567811	1919 Larch Lane, Whitestone	1986-03-09
23	Logan	King	logan.king@example.com	+1234567812	2020 Spruce Boulevard, Brookfield	1992-05-14
24	Harper	Scott	harper.scott@example.com	+1234567813	2121 Alder Drive, Northfield	1995-09-19
25	Elijah	Hill	elijah.hill@example.com	+1234567814	2222 Juniper Street, Lakeshore	1984-02-22
26	Evelyn	Adams	evelyn.adams@example.com	+1234567815	2323 Elmwood Road, Pinehurst	1990-01-01
27	Oliver	Baker	oliver.baker@example.com	+1234567816	2424 Birchwood Circle, Maplewood	1983-07-17
28	Sophia	Brown	sophia.brown@example.com	+1234567817	2525 Sycamore Drive, Willowbrook	1996-08-08

2. Inserting a New Record

- **Query**:

```
INSERT INTO Customers (customer_id, first_name,
last_name, email, phone, address, birth_date)

VALUES (29, 'Eren', 'Çakır', 'eren.cakir@example.com',
'+9876543210', '2237 Berber Street, Bolu', '2002-25-06');
```

- This query adds a new customer with customer_id = 29.

- **Output**:

```
mysql> INSERT INTO Customers (customer_id, first_name, last_name, email, phone, address, birth_date)
    -> VALUES
    -> (29, 'Eren', 'Çakır', 'eren.cakir@example.com', '+9876543210','2237 Berber Street, Bolu', '2002-25-06');
Query OK, 1 row affected, 2 warnings (0.00 sec)
```

```
mysql> SELECT * FROM Customers;
+-------------+------------+-----------+-----------------------------+------------+---------------------------------+------------+
| customer_id | first_name | last_name | email                       | phone      | address                         | birth_date |
+-------------+------------+-----------+-----------------------------+------------+---------------------------------+------------+
|           1 | John       | Smith     | john.smith@example.com      | +1234567890 | 123 Elm Street, Springfield     | 1985-06-15 |
|           2 | Jane       | Doe       | jane.doe@example.com        | +1234567891 | 456 Oak Avenue, Metropolis      | 1992-08-10 |
|           3 | Michael    | Johnson   | michael.johnson@example.com | +1234567892 | 789 Pine Lane, Gotham           | 1980-01-01 |
|           4 | Emily      | Davis     | emily.davis@example.com     | +1234567893 | 101 Maple Drive, Smallville     | 1995-03-12 |
|           5 | Chris      | Brown     | chris.brown@example.com     | +1234567894 | 202 Birch Boulevard, Star City  | 1978-12-20 |
|           6 | Sarah      | Miller    | sarah.miller@example.com    | +1234567895 | 303 Cedar Way, Central City     | 1990-07-07 |
|           7 | David      | Wilson    | david.wilson@example.com    | +1234567896 | 404 Cherry Crescent, Coast City | 1988-11-25 |
|           8 | Emma       | Taylor    | emma.taylor@example.com     | +1234567897 | 505 Willow Alley, Hill Valley   | 1996-09-30 |
|           9 | Daniel     | Anderson  | daniel.anderson@example.com | +1234567898 | 606 Sycamore Circle, Riverdale  | 1983-05-22 |
|          10 | Sophia     | Thomas    | sophia.thomas@example.com   | +1234567899 | 707 Poplar Path, Sunnydale      | 1997-04-18 |
|          11 | James      | Jackson   | james.jackson@example.com   | +1234567800 | 808 Aspen Avenue, Greenvale     | 1989-02-14 |
|          12 | Olivia     | White     | olivia.white@example.com    | +1234567801 | 909 Magnolia Lane, Rosewood     | 1994-10-10 |
|          13 | Ethan      | Harris    | ethan.harris@example.com    | +1234567802 | 1010 Chestnut Road, Shadyside   | 1975-01-30 |
|          14 | Ava        | Martin    | ava.martin@example.com      | +1234567803 | 1111 Walnut Drive, Kingsland    | 1991-06-05 |
|          15 | Lucas      | Garcia    | lucas.garcia@example.com    | +1234567804 | 1212 Hickory Street, Elmwood    | 1987-03-21 |
|          16 | Mia        | Martinez  | mia.martinez@example.com    | +1234567805 | 1313 Dogwood Terrace, Briarwood | 1993-08-02 |
|          17 | Liam       | Robinson  | liam.robinson@example.com   | +1234567806 | 1414 Maplewood Court, Cloverfield | 1982-11-11 |
|          18 | Isabella   | Clark     | isabella.clark@example.com  | +1234567807 | 1515 Redwood Lane, Ravenswood   | 1998-07-27 |
|          19 | Noah       | Lewis     | noah.lewis@example.com      | +1234567808 | 1616 Fir Place, Everwood        | 1981-04-04 |
|          20 | Charlotte  | Walker    | charlotte.walker@example.com | +1234567809 | 1717 Palm Court, Thornhill      | 1999-12-12 |
|          21 | Mason      | Young     | mason.young@example.com     | +1234567810 | 1818 Cypress Way, Blackwood     | 1979-10-25 |
|          22 | Amelia     | Allen     | amelia.allen@example.com    | +1234567811 | 1919 Larch Lane, Whitestone     | 1986-03-09 |
|          23 | Logan      | King      | logan.king@example.com      | +1234567812 | 2020 Spruce Boulevard, Brookfield | 1992-05-14 |
|          24 | Harper     | Scott     | harper.scott@example.com    | +1234567813 | 2121 Alder Drive, Northfield    | 1995-09-19 |
|          25 | Elijah     | Hill      | elijah.hill@example.com     | +1234567814 | 2222 Juniper Street, Lakeshore  | 1984-02-22 |
|          26 | Evelyn     | Adams     | evelyn.adams@example.com    | +1234567815 | 2323 Elmwood Road, Pinehurst    | 1990-01-01 |
|          27 | Oliver     | Baker     | oliver.baker@example.com    | +1234567816 | 2424 Birchwood Circle, Maplewood | 1983-07-17 |
|          28 | Sophia     | Brown     | sophia.brown@example.com    | +1234567817 | 2525 Sycamore Drive, Willowbrook | 1996-08-08 |
|          29 | Eren       | Çak?r     | eren.cakir@example.com      | +9876543210 | 2237 Berber Street, Bolu        | 0000-00-00 |
+-------------+------------+-----------+-----------------------------+------------+---------------------------------+------------+
29 rows in set (0.00 sec)
```

3. Viewing Changes After Insertion

- **Query**: `SELECT * FROM Customers WHERE customer_id=29;`
- This query verifies the insertion of the new record of the customer customer_id = 29 .
- **Output**:

```
mysql> SELECT * FROM Customers;
+-------------+------------+-----------+-----------------------------+------------+---------------------------------+------------+
| customer_id | first_name | last_name | email                       | phone      | address                         | birth_date |
+-------------+------------+-----------+-----------------------------+------------+---------------------------------+------------+
|          29 | Eren       | Çak?r     | eren.cakir@example.com      | +9876543210 | 2237 Berber Street, Bolu        | 0000-00-00 |
+-------------+------------+-----------+-----------------------------+------------+---------------------------------+------------+
29 rows in set (0.00 sec)
```

4. Updating a Record

- **Query**:

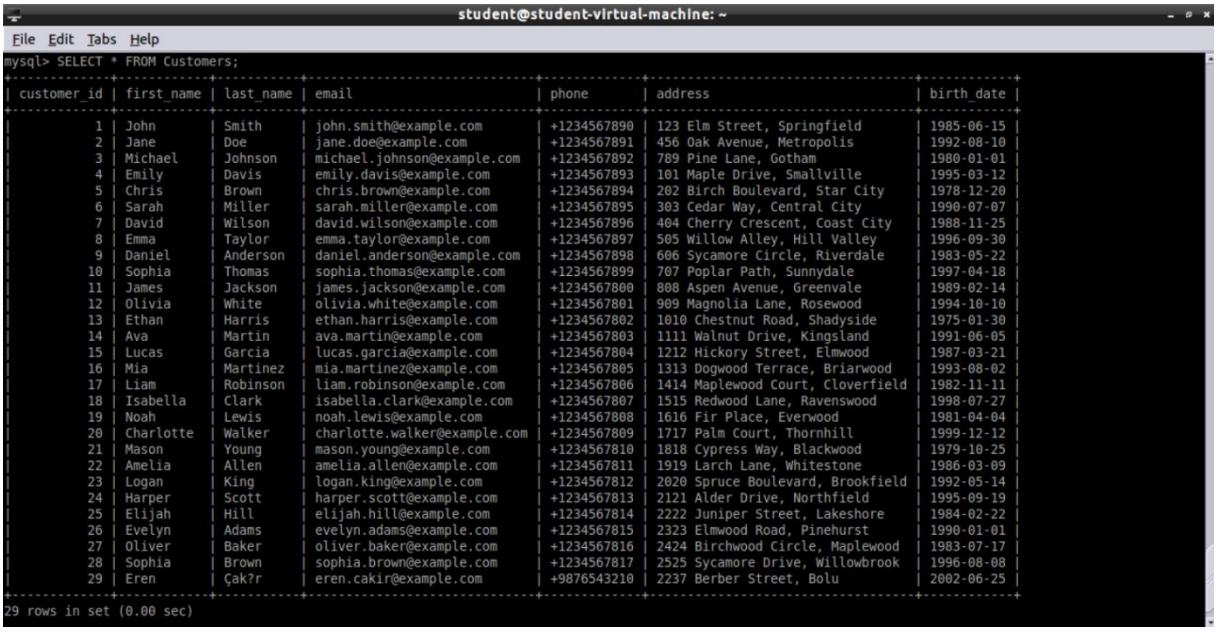
```
UPDATE Customers

SET birth_date = '2002-06-25'

WHERE customer_id = 29;
```

- This query corrects the birth_date of the new customer.

- **Output**:



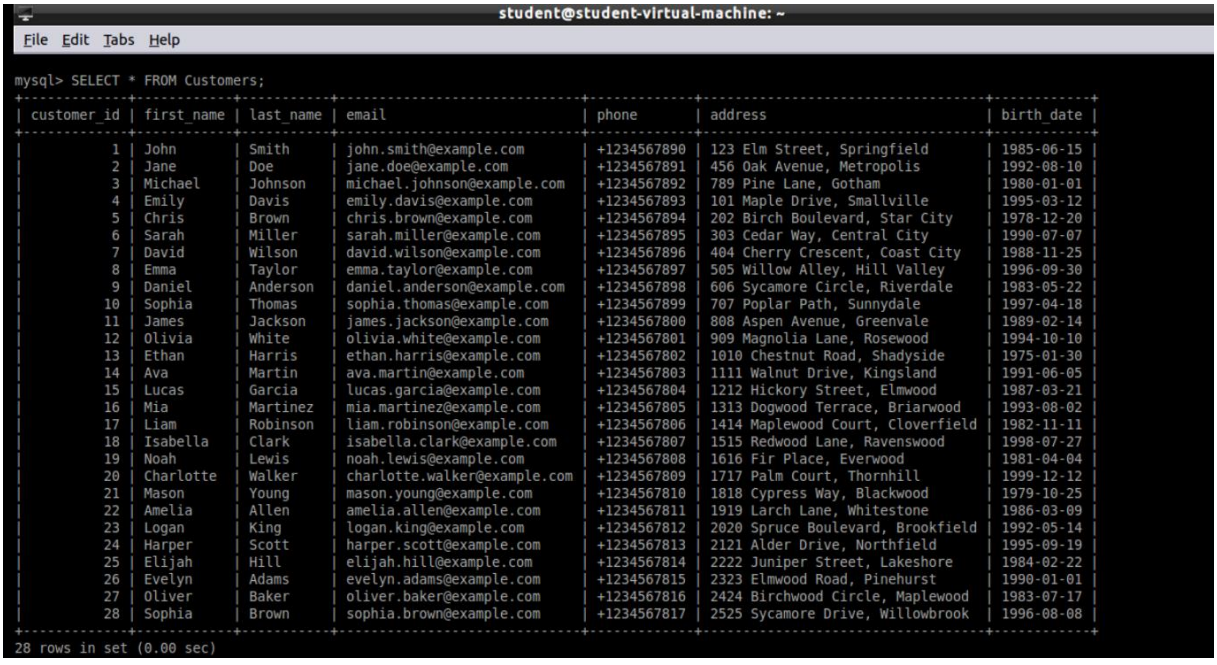
6. Deleting a Record

- **Query**:

```
DELETE FROM Customers

WHERE customer_id = 29;
```

- This query removes the record with customer_id = 29.



8. Altering the Table Structure

- **Query**:

```
ALTER TABLE Customers
ADD COLUMN customer_type ENUM('Regular', 'VIP', 'New')
DEFAULT 'Regular';
```

- This query adds a new column, customer_type.
- **Output**:

```
mysql> ALTER TABLE Customers
    -> ADD COLUMN customer_type ENUM('Regular', 'VIP', 'New') DEFAULT 'Regular';
Query OK, 28 rows affected (0.06 sec)
Records: 28  Duplicates: 0  Warnings: 0

mysql> select * from Customers;
+-------------+------------+-----------+-----------------------------+-------------+-----------------------------------+------------+---------------+
| customer_id | first_name | last_name | email                       | phone       | address                           | birth_date | customer_type |
+-------------+------------+-----------+-----------------------------+-------------+-----------------------------------+------------+---------------+
|           1 | John       | Smith     | john.smith@example.com      | +1234567890 | 123 Elm Street, Springfield       | 1985-06-15 | Regular       |
|           2 | Jane       | Doe       | jane.doe@example.com        | +1234567891 | 456 Oak Avenue, Metropolis        | 1992-08-10 | Regular       |
|           3 | Michael    | Johnson   | michael.johnson@example.com | +1234567892 | 789 Pine Lane, Gotham             | 1980-01-01 | Regular       |
|           4 | Emily      | Davis     | emily.davis@example.com     | +1234567893 | 101 Maple Drive, Smallville       | 1995-03-12 | Regular       |
|           5 | Chris      | Brown     | chris.brown@example.com     | +1234567894 | 202 Birch Boulevard, Star City    | 1978-12-20 | Regular       |
|           6 | Sarah      | Miller    | sarah.miller@example.com    | +1234567895 | 303 Cedar Way, Central City       | 1990-07-07 | Regular       |
|           7 | David      | Wilson    | david.wilson@example.com    | +1234567896 | 404 Cherry Crescent, Coast City   | 1988-11-25 | Regular       |
|           8 | Emma       | Taylor    | emma.taylor@example.com     | +1234567897 | 505 Willow Alley, Hill Valley     | 1996-09-30 | Regular       |
|           9 | Daniel     | Anderson  | daniel.anderson@example.com | +1234567898 | 606 Sycamore Circle, Riverdale    | 1983-05-22 | Regular       |
|          10 | Sophia     | Thomas    | sophia.thomas@example.com   | +1234567899 | 707 Poplar Path, Sunnydale        | 1997-04-18 | Regular       |
|          11 | James      | Jackson   | james.jackson@example.com   | +1234567800 | 808 Aspen Avenue, Greenvale       | 1989-02-14 | Regular       |
|          12 | Olivia     | White     | olivia.white@example.com    | +1234567801 | 909 Magnolia Lane, Rosewood       | 1994-10-10 | Regular       |
|          13 | Ethan      | Harris    | ethan.harris@example.com    | +1234567802 | 1010 Chestnut Road, Shadyside     | 1975-01-30 | Regular       |
|          14 | Ava        | Martin    | ava.martin@example.com      | +1234567803 | 1111 Walnut Drive, Kingsland      | 1991-06-05 | Regular       |
|          15 | Lucas      | Garcia    | lucas.garcia@example.com    | +1234567804 | 1212 Hickory Street, Elmwood      | 1987-03-21 | Regular       |
|          16 | Mia        | Martinez  | mia.martinez@example.com    | +1234567805 | 1313 Dogwood Terrace, Briarwood   | 1993-08-02 | Regular       |
|          17 | Liam       | Robinson  | liam.robinson@example.com   | +1234567806 | 1414 Maplewood Court, Cloverfield | 1982-11-11 | Regular       |
|          18 | Isabella   | Clark     | isabella.clark@example.com  | +1234567807 | 1515 Redwood Lane, Ravenswood     | 1998-07-27 | Regular       |
|          19 | Noah       | Lewis     | noah.lewis@example.com      | +1234567808 | 1616 Fir Place, Everwood          | 1981-04-04 | Regular       |
|          20 | Charlotte  | Walker    | charlotte.walker@example.com | +1234567809 | 1717 Palm Court, Thornhill        | 1999-12-12 | Regular       |
|          21 | Mason      | Young     | mason.young@example.com     | +1234567810 | 1818 Cypress Way, Blackwood       | 1979-10-25 | Regular       |
|          22 | Amelia     | Allen     | amelia.allen@example.com    | +1234567811 | 1919 Larch Lane, Whitestone       | 1986-03-09 | Regular       |
|          23 | Logan      | King      | logan.king@example.com      | +1234567812 | 2020 Spruce Boulevard, Brookfield | 1992-05-14 | Regular       |
|          24 | Harper     | Scott     | harper.scott@example.com    | +1234567813 | 2121 Alder Drive, Northfield      | 1995-09-19 | Regular       |
|          25 | Elijah     | Hill      | elijah.hill@example.com     | +1234567814 | 2222 Juniper Street, Lakeshore    | 1984-02-22 | Regular       |
|          26 | Evelyn     | Adams     | evelyn.adams@example.com    | +1234567815 | 2323 Elmwood Road, Pinehurst      | 1990-01-01 | Regular       |
|          27 | Oliver     | Baker     | oliver.baker@example.com    | +1234567816 | 2424 Birchwood Circle, Maplewood  | 1983-07-17 | Regular       |
|          28 | Sophia     | Brown     | sophia.brown@example.com    | +1234567817 | 2525 Sycamore Drive, Willowbrook  | 1996-08-08 | Regular       |
+-------------+------------+-----------+-----------------------------+-------------+-----------------------------------+------------+---------------+
```

5.2 Queries and Results for Activity table

Below is a detailed overview of the queries performed on the **Activity** table along with their corresponding outputs and purposes.

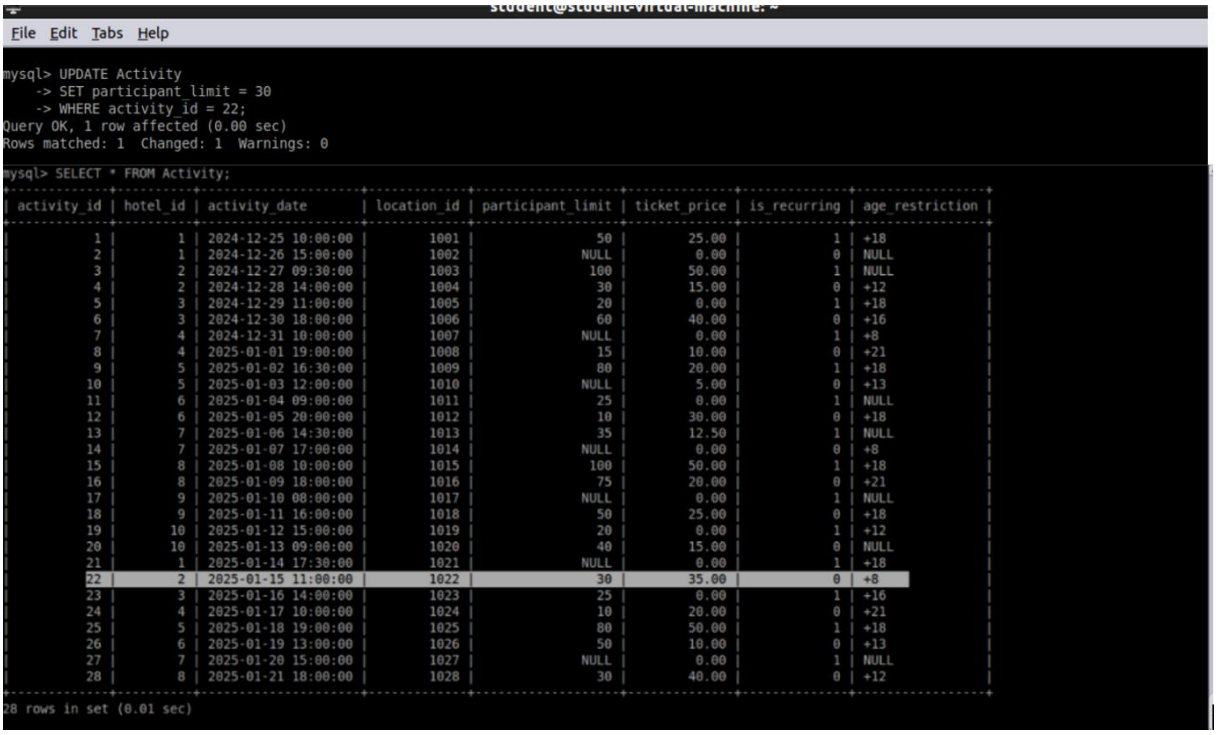
1. Update Participant Limit

- **Query**:

```
UPDATE Activity
SET participant_limit = 30
WHERE activity_id = 22;
```

- This query updates the participant_limit to the value 30 for the activity with activity_id equal to 22.

- **Output**:



2. Insert a New Activity

- **Query**:

```
INSERT INTO Activity (activity_id, hotel_id,
activity_date, location_id, participant_limit,
ticket_price, is_recurring, age_restriction)
VALUES (29, 1, '2025-05-05 15:00:00', 1011, 80, 90.00,
FALSE, '+18');
```

- Inserts a new record into the Activity table.

- **Output**:

```
mysql> INSERT INTO Activity (activity_id, hotel_id, activity_date, location_id, participant_limit, ticket_price, is_recurring, age_restriction)
    -> VALUES (29, 1, '2025-05-05 15:00:00', 1011, 80, 90.00, FALSE, '+18');
Query OK, 1 row affected (0.01 sec)
```

```
select * from Activity
SELECT * FROM Activity' at line 2
mysql> SELECT * FROM Activity;
+-------------+----------+---------------------+-------------+------------------+--------------+--------------+-------------------+
| activity_id | hotel_id | activity_date       | location_id | participant_limit | ticket_price | is_recurring | age_restriction |
+-------------+----------+---------------------+-------------+------------------+--------------+--------------+-------------------+
|           1 |        1 | 2024-12-25 10:00:00 |        1001 |               50 |        25.00 |            1 | +18               |
|           2 |        1 | 2024-12-26 15:00:00 |        1002 |             NULL |         0.00 |            0 | NULL              |
|           3 |        2 | 2024-12-27 09:30:00 |        1003 |              100 |        50.00 |            1 | NULL              |
|           4 |        2 | 2024-12-28 14:00:00 |        1004 |               30 |        15.00 |            0 | +12               |
|           5 |        3 | 2024-12-29 11:00:00 |        1005 |               20 |         0.00 |            1 | +18               |
|           6 |        3 | 2024-12-30 18:00:00 |        1006 |               60 |        40.00 |            0 | +16               |
|           7 |        4 | 2024-12-31 10:00:00 |        1007 |             NULL |         0.00 |            1 | +8                |
|           8 |        4 | 2025-01-01 19:00:00 |        1008 |               15 |        10.00 |            0 | +21               |
|           9 |        5 | 2025-01-02 16:30:00 |        1009 |               80 |        20.00 |            1 | +18               |
|          10 |        5 | 2025-01-03 12:00:00 |        1010 |             NULL |         5.00 |            0 | +13               |
|          11 |        6 | 2025-01-04 09:00:00 |        1011 |               25 |         0.00 |            1 | NULL              |
|          12 |        6 | 2025-01-05 20:00:00 |        1012 |               10 |        30.00 |            0 | +18               |
|          13 |        7 | 2025-01-06 14:30:00 |        1013 |               35 |        12.50 |            1 | NULL              |
|          14 |        7 | 2025-01-07 17:00:00 |        1014 |             NULL |         0.00 |            0 | +8                |
|          15 |        8 | 2025-01-08 10:00:00 |        1015 |              100 |        50.00 |            1 | +18               |
|          16 |        8 | 2025-01-09 18:00:00 |        1016 |               75 |        20.00 |            0 | +21               |
|          17 |        9 | 2025-01-10 08:00:00 |        1017 |             NULL |         0.00 |            1 | NULL              |
|          18 |        9 | 2025-01-11 16:00:00 |        1018 |               50 |        25.00 |            0 | +18               |
|          19 |       10 | 2025-01-12 15:00:00 |        1019 |               20 |         0.00 |            1 | +12               |
|          20 |       10 | 2025-01-13 09:00:00 |        1020 |               40 |        15.00 |            0 | NULL              |
|          21 |        1 | 2025-01-14 17:30:00 |        1021 |             NULL |         0.00 |            1 | +18               |
|          22 |        2 | 2025-01-15 11:00:00 |        1022 |               30 |        35.00 |            0 | +8                |
|          23 |        3 | 2025-01-16 14:00:00 |        1023 |               25 |         0.00 |            1 | +16               |
|          24 |        4 | 2025-01-17 10:00:00 |        1024 |               10 |        20.00 |            0 | +21               |
|          25 |        5 | 2025-01-18 19:00:00 |        1025 |               80 |        50.00 |            1 | +18               |
|          26 |        6 | 2025-01-19 13:00:00 |        1026 |               50 |        10.00 |            0 | +13               |
|          27 |        7 | 2025-01-20 15:00:00 |        1027 |             NULL |         0.00 |            1 | NULL              |
|          28 |        8 | 2025-01-21 18:00:00 |        1028 |               30 |        40.00 |            0 | +12               |
|          29 |        1 | 2025-05-05 15:00:00 |        1011 |               80 |        90.00 |            0 | +18               |
```

4. Calculate Average Ticket Price

- **Query**:

```
SELECT AVG(ticket_price) AS average_price FROM Activity;
```

- Calculates the average ticket price across all activities in the table.
- **Output**:

```
mysql> SELECT AVG(ticket_price) AS average_price FROM Activity;
+---------------+
| average_price |
+---------------+
|     19.396552 |
+---------------+
1 row in set (0.05 sec)
```

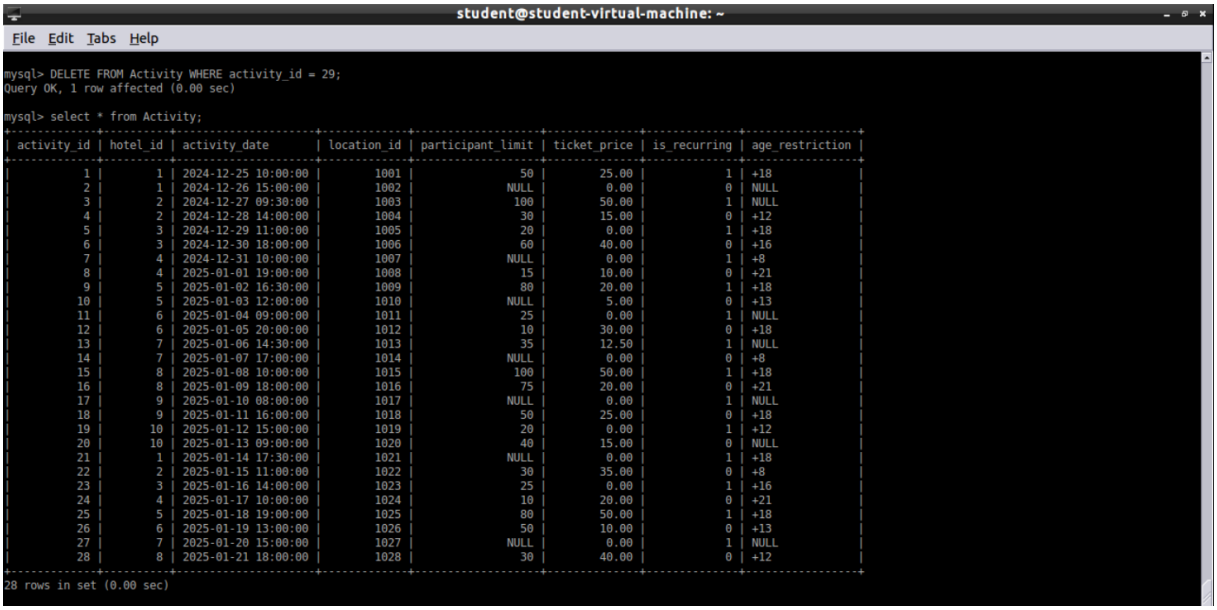
3. Delete a Specific Activity

- **Query**:

```
DELETE FROM Activity WHERE activity_id = 29;
```

- Deletes the activity with activity_id equal to 29 from the table.

- **Output**:



5.3 Queries and Results for Voucher table

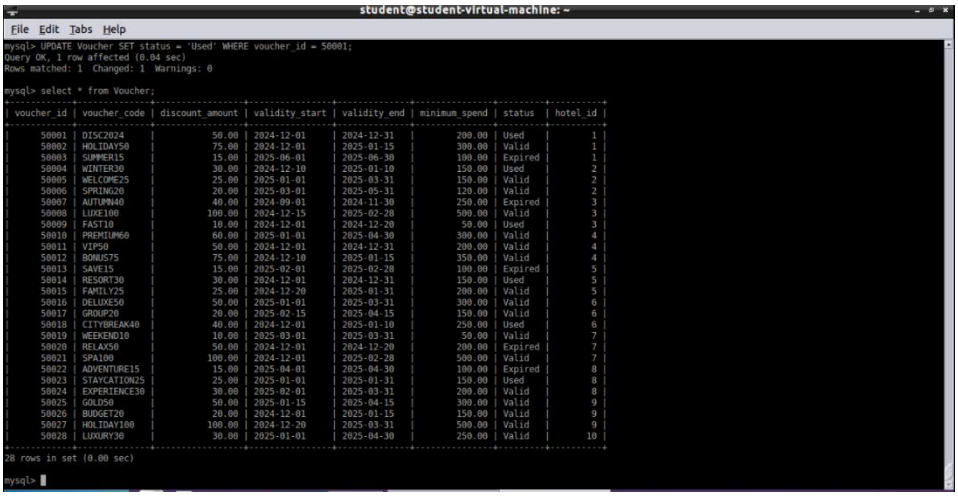
1. Updating a Voucher Record

- **Query**:

```
UPDATE Voucher SET status = 'Used' WHERE voucher_id = 50001;
```

- To update the status of a voucher (voucher ID 50001) from 'Valid' to 'Used'. This ensures that the same voucher cannot be used again.

- **Output**:

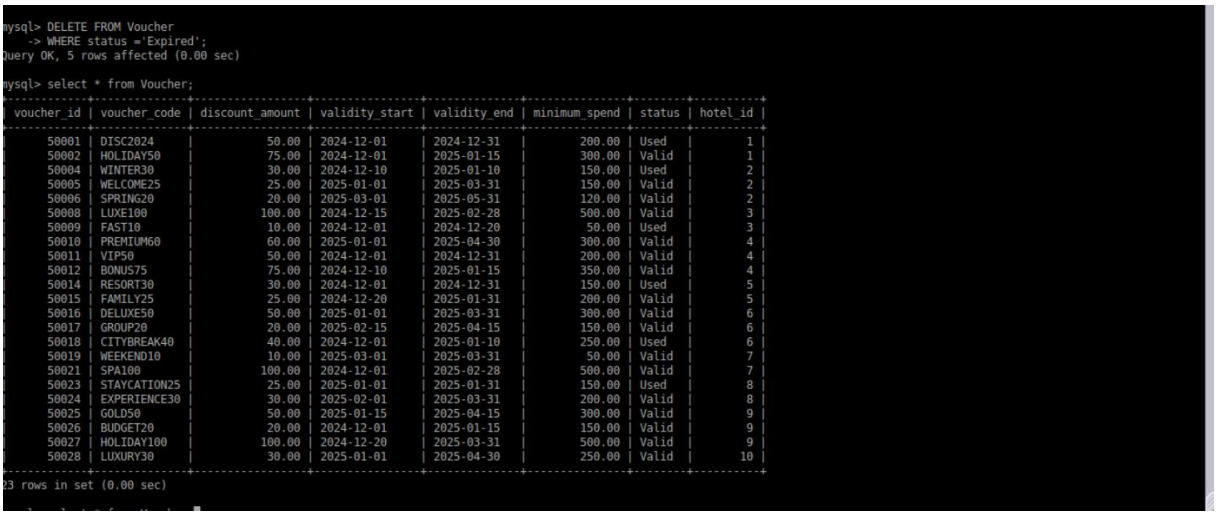


2. Deleting Expired Vouchers

- **Query**:

```
DELETE FROM Voucher WHERE status = 'Expired';
```

- To remove all records of vouchers that are no longer valid (status 'Expired') from the table.
- **Output**:



```
mysql> DELETE FROM Voucher
    -> WHERE status ='Expired';
Query OK, 5 rows affected (0.00 sec)

mysql> select * from Voucher;
+------------+--------------+-----------------+----------------+--------------+---------------+--------+----------+
| voucher_id | voucher_code | discount_amount | validity_start | validity_end | minimum_spend | status | hotel_id |
+------------+--------------+-----------------+----------------+--------------+---------------+--------+----------+
|      50001 | DISC2024     |           50.00 | 2024-12-01     | 2024-12-31   |        200.00 | Used   |        1 |
|      50002 | HOLIDAY50    |           75.00 | 2024-12-01     | 2025-01-15   |        300.00 | Valid  |        1 |
|      50004 | WINTER30     |           30.00 | 2024-12-10     | 2025-01-10   |        150.00 | Used   |        2 |
|      50005 | WELCOME25    |           25.00 | 2025-01-01     | 2025-03-31   |        150.00 | Valid  |        2 |
|      50006 | SPRING20     |           20.00 | 2025-03-01     | 2025-05-31   |        120.00 | Valid  |        2 |
|      50008 | LUXE100      |          100.00 | 2024-12-15     | 2025-02-28   |        500.00 | Valid  |        3 |
|      50009 | FAST10       |           10.00 | 2024-12-01     | 2024-12-20   |         50.00 | Used   |        3 |
|      50010 | PREMIUM60    |           60.00 | 2025-01-01     | 2025-04-30   |        300.00 | Valid  |        4 |
|      50011 | VIP50        |           50.00 | 2024-12-01     | 2024-12-31   |        200.00 | Valid  |        4 |
|      50012 | BONUS75      |           75.00 | 2024-12-10     | 2025-01-15   |        350.00 | Valid  |        4 |
|      50014 | RESORT30     |           30.00 | 2024-12-01     | 2024-12-31   |        150.00 | Used   |        5 |
|      50015 | FAMILY25     |           25.00 | 2024-12-20     | 2025-01-31   |        200.00 | Valid  |        5 |
|      50016 | DELUXE50     |           50.00 | 2025-01-01     | 2025-03-31   |        300.00 | Valid  |        6 |
|      50017 | GROUP20      |           20.00 | 2025-02-15     | 2025-04-15   |        150.00 | Valid  |        6 |
|      50018 | CITYBREAK40  |           40.00 | 2024-12-01     | 2025-01-10   |        250.00 | Used   |        6 |
|      50019 | WEEKEND10    |           10.00 | 2025-03-01     | 2025-03-31   |         50.00 | Valid  |        7 |
|      50021 | SPA100       |          100.00 | 2024-12-01     | 2025-02-28   |        500.00 | Valid  |        7 |
|      50023 | STAYCATION25 |           25.00 | 2025-01-01     | 2025-01-31   |        150.00 | Used   |        8 |
|      50024 | EXPERIENCE30 |           30.00 | 2025-02-01     | 2025-03-31   |        200.00 | Valid  |        8 |
|      50025 | GOLD50       |           50.00 | 2025-01-15     | 2025-04-15   |        300.00 | Valid  |        9 |
|      50026 | BUDGET20     |           20.00 | 2024-12-01     | 2025-01-15   |        150.00 | Valid  |        9 |
|      50027 | HOLIDAY100   |          100.00 | 2024-12-20     | 2025-03-31   |        500.00 | Valid  |        9 |
|      50028 | LUXURY30     |           30.00 | 2025-01-01     | 2025-04-30   |        250.00 | Valid  |       10 |
+------------+--------------+-----------------+----------------+--------------+---------------+--------+----------+
23 rows in set (0.00 sec)
```

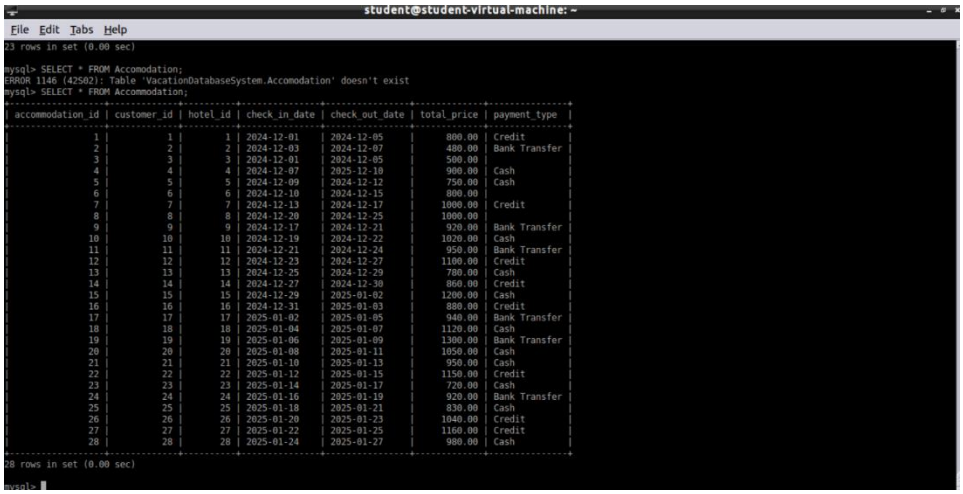
5.3 Queries and Results for Accommodation table

1. **Selecting Data from Accommodation Table**

- **Query**:

```
SELECT * FROM Accommodation;
```

- To retrieve all the records in the Accommodation table, showing details like customer ID, hotel ID, check-in and check-out dates, total price, and payment type.
- **Output**:



```
student@student-virtual-machine: ~
File  Edit  Tabs  Help
23 rows in set (0.00 sec)

mysql> SELECT * FROM Accomodation;
ERROR 1146 (42S02): Table 'VacationDatabaseSystem.Accomodation' doesn't exist
mysql> SELECT * FROM Accommodation;
+----------------+-------------+----------+---------------+----------------+-------------+---------------+
| accommodation_id | customer_id | hotel_id | check_in_date | check_out_date | total_price | payment_type  |
+----------------+-------------+----------+---------------+----------------+-------------+---------------+
|              1 |           1 |        1 | 2024-12-01    | 2024-12-05     |      800.00 | Credit        |
|              2 |           2 |        2 | 2024-12-03    | 2024-12-07     |      480.00 | Bank Transfer |
|              3 |           3 |        3 | 2024-12-01    | 2024-12-05     |      500.00 |               |
|              4 |           4 |        4 | 2024-12-07    | 2025-12-10     |      900.00 | Cash          |
|              5 |           5 |        5 | 2024-12-09    | 2024-12-12     |      750.00 | Cash          |
|              6 |           6 |        6 | 2024-12-10    | 2024-12-15     |      800.00 |               |
|              7 |           7 |        7 | 2024-12-13    | 2024-12-17     |     1000.00 | Credit        |
|              8 |           8 |        8 | 2024-12-20    | 2024-12-25     |     1000.00 |               |
|              9 |           9 |        9 | 2024-12-17    | 2024-12-21     |      920.00 | Bank Transfer |
|             10 |          10 |       10 | 2024-12-19    | 2024-12-22     |     1020.00 | Cash          |
|             11 |          11 |       11 | 2024-12-21    | 2024-12-24     |      950.00 | Bank Transfer |
|             12 |          12 |       12 | 2024-12-23    | 2024-12-27     |     1100.00 | Credit        |
|             13 |          13 |       13 | 2024-12-25    | 2024-12-29     |      780.00 | Cash          |
|             14 |          14 |       14 | 2024-12-27    | 2024-12-30     |      860.00 | Credit        |
|             15 |          15 |       15 | 2024-12-29    | 2025-01-02     |     1200.00 | Cash          |
|             16 |          16 |       16 | 2024-12-31    | 2025-01-03     |      880.00 | Credit        |
|             17 |          17 |       17 | 2025-01-02    | 2025-01-05     |      940.00 | Bank Transfer |
|             18 |          18 |       18 | 2025-01-04    | 2025-01-07     |     1120.00 | Cash          |
|             19 |          19 |       19 | 2025-01-06    | 2025-01-09     |     1300.00 | Bank Transfer |
|             20 |          20 |       20 | 2025-01-08    | 2025-01-11     |     1050.00 | Cash          |
|             21 |          21 |       21 | 2025-01-10    | 2025-01-13     |      950.00 | Cash          |
|             22 |          22 |       22 | 2025-01-12    | 2025-01-15     |     1150.00 | Credit        |
|             23 |          23 |       23 | 2025-01-14    | 2025-01-17     |      720.00 | Cash          |
|             24 |          24 |       24 | 2025-01-16    | 2025-01-19     |      920.00 | Bank Transfer |
|             25 |          25 |       25 | 2025-01-18    | 2025-01-21     |      830.00 | Cash          |
|             26 |          26 |       26 | 2025-01-20    | 2025-01-23     |     1040.00 | Credit        |
|             27 |          27 |       27 | 2025-01-22    | 2025-01-25     |     1160.00 | Credit        |
|             28 |          28 |       28 | 2025-01-24    | 2025-01-27     |      980.00 | Cash          |
+----------------+-------------+----------+---------------+----------------+-------------+---------------+
28 rows in set (0.00 sec)

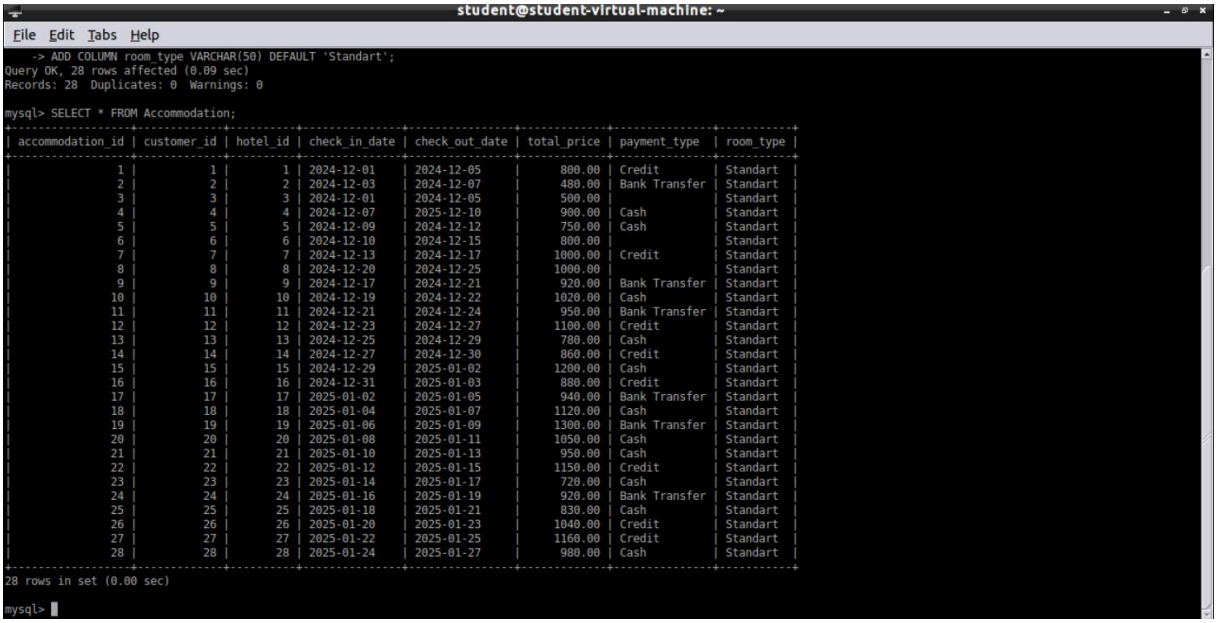
mysql>
```

2. Adding a New Column

- **Query**:

```
ALTER TABLE Accommodation ADD COLUMN room_type VARCHAR(50)
DEFAULT 'Standart';
```

- To add a new column, room_type, to specify the type of room booked by the customer. The default value is set to 'Standart'.
- **Output**:



3. Inserting a New Record

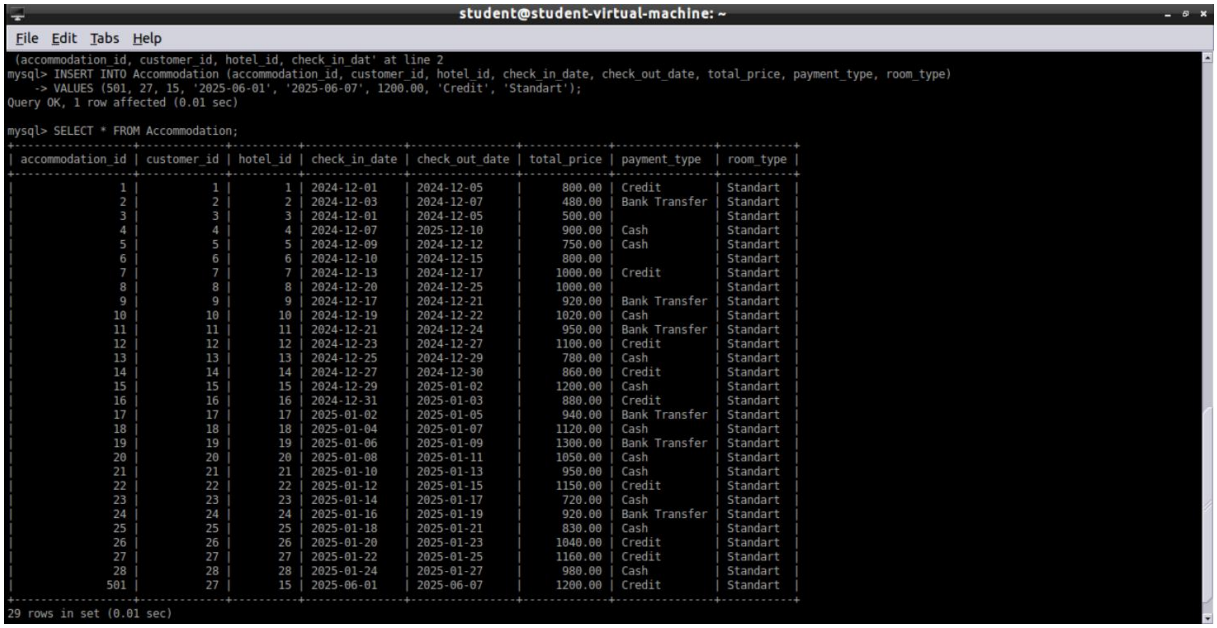
- **Query**:

```
INSERT INTO Accommodation (accommodation_id, customer_id,
hotel_id, check_in_date, check_out_date, total_price,
payment_type, room_type)

VALUES (501, 27, 15, '2025-06-01', '2025-06-07', 1200.00,
'Credit', 'Standart');
```

- To insert a new accommodation record into the table with details about the customer, hotel, check-in and check-out dates, total price, payment type, and room type.

- **Output**:



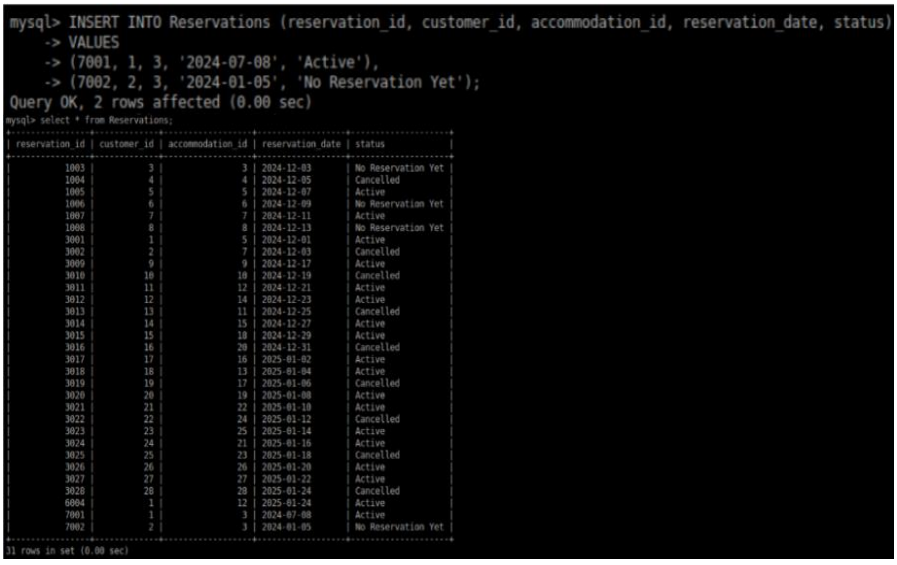
5.5 Queries and Results for Reservations Table

1. Inserting a New Record

- **Query:**

```
INSERT INTO Reservations (reservation_id,
accommodation_id, reservation_date, status)
VALUES
(7001, 1, 3, '2024-07-08', 'Active')
(7002, 2, 3, '2024-01-05', 'No Reservation Yet');
```

- This query adds 2 new reservations with `reservation_id = 7001 & 7002`
- **Output:**



2. Viewing Changes in Records

- Applied changes, inserted 2 new reservations, are recorded to the PastReservations table.

- **Query:** `select * from PastReservations ;`
- **Output:**

```
mysql> select * from PastReservations;
+----------------+-------------+------------------+------------------+-------------------+---------------------+-------------+
| reservation_id | customer_id | accommodation_id | reservation_date | status            | log_date            | action_type |
+----------------+-------------+------------------+------------------+-------------------+---------------------+-------------+
|           7001 |           1 |                3 | 2024-07-08       | Active            | 2025-01-09 19:23:02 |             |
|           7002 |           2 |                3 | 2024-01-05       | No Reservation Yet | 2025-01-09 19:23:02 |             |
+----------------+-------------+------------------+------------------+-------------------+---------------------+-------------+
```

3. Updating a Record

- **Query:**

```
UPDATE Reservations
SET status = 'Cancelled'
WHERE reservation_id = 7001;
```

- This query updates the status of the already existing reservation whose `reservation_id = 7001` with `'Cancelled'`.

- **Output:**

```
mysql> UPDATE Reservations SET status = 'Cancelled' WHERE reservation_id = 7001;
Query OK, 0 rows affected (0.01 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> select * from PastReservations;
+----------------+-------------+------------------+------------------+-------------------+---------------------+-------------+
| reservation_id | customer_id | accommodation_id | reservation_date | status            | log_date            | action_type |
+----------------+-------------+------------------+------------------+-------------------+---------------------+-------------+
|           7001 |           1 |                3 | 2024-07-08       | Active            | 2025-01-09 19:23:02 |             |
|           7002 |           2 |                3 | 2024-01-05       | No Reservation Yet | 2025-01-09 19:23:02 |             |
|           7001 |           1 |                3 | 2024-07-08       | Active            | 2025-01-09 19:25:50 | UPDATE      |
|           7001 |           1 |                3 | 2024-07-08       | Cancelled         | 2025-01-09 19:27:39 | UPDATE      |
+----------------+-------------+------------------+------------------+-------------------+---------------------+-------------+
4 rows in set (0.00 sec)
```

4. Deleting a Record

- **Query:**
-
```
DELETE FROM Reservations
WHERE reservation_id = 7001;
```

- This query removes the record with `reservation_id = 7001`.

- **Output:**

```
mysql> DELETE FROM Reservations WHERE reservation_id = 7001;
Query OK, 1 row affected (0.00 sec)

mysql> select * from PastReservations;
+----------------+-------------+------------------+------------------+--------------------+---------------------+-------------+
| reservation_id | customer_id | accommodation_id | reservation_date | status             | log_date            | action_type |
+----------------+-------------+------------------+------------------+--------------------+---------------------+-------------+
|           7001 |           1 |                3 | 2024-07-08       | Active             | 2025-01-09 19:23:02 |             |
|           7002 |           2 |                3 | 2024-01-05       | No Reservation Yet | 2025-01-09 19:23:02 |             |
|           7001 |           1 |                3 | 2024-07-08       | Active             | 2025-01-09 19:25:50 | UPDATE      |
|           7001 |           1 |                3 | 2024-07-08       | Cancelled          | 2025-01-09 19:27:39 | UPDATE      |
|           7001 |           1 |                3 | 2024-07-08       | Cancelled          | 2025-01-09 19:28:14 | DELETE      |
+----------------+-------------+------------------+------------------+--------------------+---------------------+-------------+
5 rows in set (0.01 sec)
```

5.6 Queries and Results for Destinations Table

1. Viewing Existing Records

- **Query:** `SELECT * FROM Destinations LIMIT 5;`
- This query retrieves all records from the Destinations table, columns such as location_id, city, country, continent limiting with first 5 entries.
- **Output:**

```
mysql> SELECT * FROM Destinations LIMIT 5;
+-------------+-------+---------------+---------------+
| location_id | city  | country       | continent     |
+-------------+-------+---------------+---------------+
|        1001 | Paris | France        | Europe        |
|        1002 | Malé  | Maldives      | Asia          |
|        1003 | Aspen | United States | North America |
|        1004 | Tokyo | Japan         | Asia          |
|        1005 | Bali  | Indonesia     | Asia          |
+-------------+-------+---------------+---------------+
5 rows in set (0.00 sec)
```

2. Inserting a New Record

- **Query:**
- ```
  INSERT INTO Destinations (location_id, city, country,
  continent)
  VALUES
  (2001, 'Bolu', 'Türkiye', 'Europe');
  ```

- This query adds a new destination with `location_id = 2001`.
  ```

- **Output:**

```
mysql> select * from Destinations;
+-------------+---------------+-----------------------+---------------+
| location_id | city          | country               | continent     |
+-------------+---------------+-----------------------+---------------+
|        1001 | Paris         | France                | Europe        |
|        1002 | Malé          | Maldives              | Asia          |
|        1003 | Aspen         | United States         | North America |
|        1004 | Tokyo         | Japan                 | Asia          |
|        1005 | Bali          | Indonesia             | Asia          |
|        1006 | Zurich        | Switzerland           | Europe        |
|        1007 | Cape Town     | South Africa          | Africa        |
|        1008 | Rio de Janeiro | Brazil                | South America |
|        1009 | Sydney        | Australia             | Oceania       |
|        1010 | London        | United Kingdom        | Europe        |
|        1011 | New York      | United States         | North America |
|        1012 | Rome          | Italy
 | Europe        |
|        1013 | Bangkok       | Thailand              | Asia          |
|        1014 | Dubai         | United Arab Emirates  | Asia          |
|        1015 | Hong Kong     | China                 | Asia          |
|        1016 | Singapore     | Singapore             | Asia          |
|        1017 | Istanbul      | Türkiye               | Europe        |
|        1018 | Moscow        | Russia                | Europe        |
|        1019 | Vancouver     | Canada                | North America |
|        1020 | Buenos Aires  | Argentina             | South America |
|        1021 | Helsinki      | Finland               | Europe        |
|        1022 | Vienna        | Austria               | Europe        |
|        1023 | Berlin        | Germany               | Europe        |
|        1024 | Madrid        | Spain                 | Europe        |
|        1025 | Mumbai        | India                 | Asia          |
|        1026 | Lima          | Peru                  | South America |
|        1027 | Cairo         | Egypt                 | Africa        |
|        1028 | Mexico City   | Mexico                | North America |
|        2001 | Bolu          | Türkiye               | Europe        |
+-------------+---------------+-----------------------+---------------+
29 rows in set (0.00 sec)
```

3. Deleting a Record

- **Query:**
  ```
  DELETE FROM Destinations
  WHERE location_id = 2001;
  ```

- This query removes the record with `destination_id = 2001`.
- **Output:**

```
mysql> DELETE FROM Destinations WHERE location_id = 2001;
Query OK, 1 row affected (0.00 sec)

mysql> select * from Destinations;
+-------------+---------------+-----------------------+---------------+
| location_id | city          | country               | continent     |
+-------------+---------------+-----------------------+---------------+
|        1001 | Paris         | France                | Europe        |
|        1002 | Malé          | Maldives              | Asia          |
|        1003 | Aspen         | United States         | North America |
|        1004 | Tokyo         | Japan                 | Asia          |
|        1005 | Bali          | Indonesia             | Asia          |
|        1006 | Zurich        | Switzerland           | Europe        |
|        1007 | Cape Town     | South Africa          | Africa        |
|        1008 | Rio de Janeiro | Brazil                | South America |
|        1009 | Sydney        | Australia             | Oceania       |
|        1010 | London        | United Kingdom        | Europe        |
|        1011 | New York      | United States         | North America |
|        1012 | Rome          | Italy
 | Europe        |
|        1013 | Bangkok       | Thailand              | Asia          |
|        1014 | Dubai         | United Arab Emirates  | Asia          |
|        1015 | Hong Kong     | China                 | Asia          |
|        1016 | Singapore     | Singapore             | Asia          |
|        1017 | Istanbul      | Türkiye               | Europe        |
|        1018 | Moscow        | Russia                | Europe        |
|        1019 | Vancouver     | Canada                | North America |
|        1020 | Buenos Aires  | Argentina             | South America |
|        1021 | Helsinki      | Finland               | Europe        |
|        1022 | Vienna        | Austria               | Europe        |
|        1023 | Berlin        | Germany               | Europe        |
|        1024 | Madrid        | Spain                 | Europe        |
|        1025 | Mumbai        | India                 | Asia          |
|        1026 | Lima          | Peru                  | South America |
|        1027 | Cairo         | Egypt                 | Africa        |
|        1028 | Mexico City   | Mexico                | North America |
+-------------+---------------+-----------------------+---------------+
28 rows in set (0.01 sec)
```

4. Updating a Record

- **Query:**
```
UPDATE Destinations
SET city = 'Barcelona'
WHERE location_id = 1024;
```

- This query updates the city name of the record with `location_id = 1024.`
- **Output :**

```
mysql> UPDATE Destinations SET city = 'Barcelona' WHERE location_id = 1024;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Destinations;
+-------------+---------------+----------------------+---------------+
| location_id | city          | country              | continent     |
+-------------+---------------+----------------------+---------------+
|        1001 | Paris         | France               | Europe        |
|        1002 | Malé          | Maldives             | Asia          |
|        1003 | Aspen         | United States        | North America |
|        1004 | Tokyo         | Japan                | Asia          |
|        1005 | Bali          | Indonesia            | Asia          |
|        1006 | Zurich        | Switzerland          | Europe        |
|        1007 | Cape Town     | South Africa         | Africa        |
|        1008 | Rio de Janeiro | Brazil               | South America |
|        1009 | Sydney        | Australia            | Oceania       |
|        1010 | London        | United Kingdom       | Europe        |
|        1011 | New York      | United States        | North America |
|        1012 | Rome          | Italy
              | Europe        |
|        1013 | Bangkok       | Thailand             | Asia          |
|        1014 | Dubai         | United Arab Emirates | Asia          |
|        1015 | Hong Kong     | China                | Asia          |
|        1016 | Singapore     | Singapore            | Asia          |
|        1017 | Istanbul      | Türkiye              | Europe        |
|        1018 | Moscow        | Russia               | Europe        |
|        1019 | Vancouver     | Canada               | North America |
|        1020 | Buenos Aires  | Argentina            | South America |
|        1021 | Helsinki      | Finland              | Europe        |
|        1022 | Vienna        | Austria              | Europe        |
|        1023 | Berlin        | Germany              | Europe        |
|        1024 | Barcelona     | Spain                | Europe        |
|        1025 | Mumbai        | India                | Asia          |
|        1026 | Lima          | Peru                 | South America |
|        1027 | Cairo         | Egypt                | Africa        |
|        1028 | Mexico City   | Mexico               | North America |
+-------------+---------------+----------------------+---------------+
28 rows in set (0.00 sec)
```

5. Number of Cities for Each Continent

- **Query:**
```
SELECT continent, SUM(location,id IS NOT NULL AS
destination_count
FROM Destinations
GROUP BY continent;
```

- Finds the number of cities for each continent by using sum and group by functions.

- **Output:**

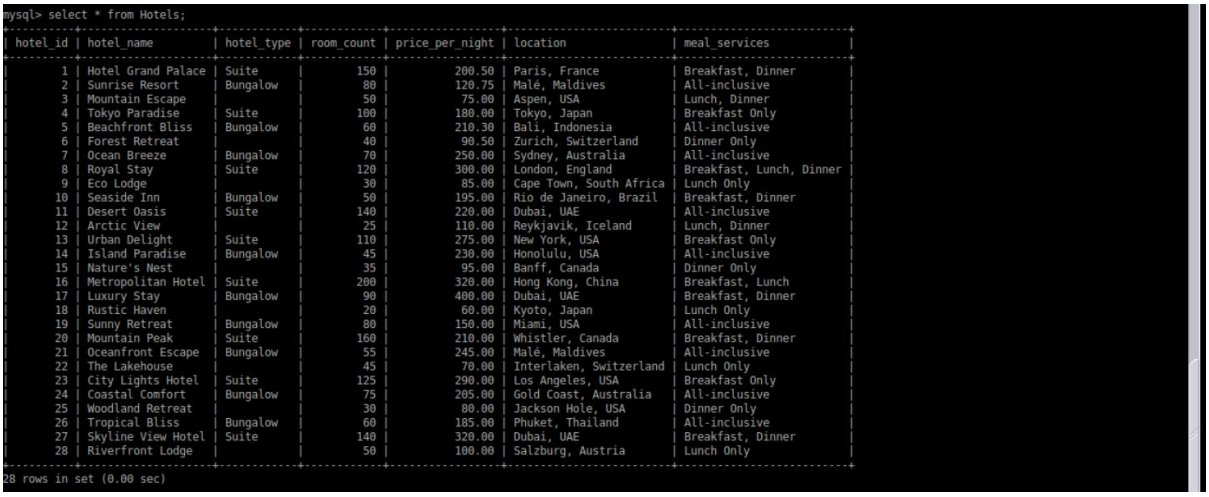
```
mysql> SELECT continent, SUM(location_id IS NOT NULL) AS destination_count
    -> FROM Destinations
    -> GROUP BY continent;
+---------------+-------------------+
| continent     | destination_count |
+---------------+-------------------+
| Africa        |                 2 |
| Asia          |                 8 |
| Europe        |                10 |
| North America |                 4 |
| Oceania       |                 1 |
| South America |                 3 |
+---------------+-------------------+
6 rows in set (0.04 sec)
```

5.7 Queries and Results for Hotels Table

1. Viewing Existing Records

- **Query:** `select * from Hotels;`

- This query retrieves all records from the `Hotels` table, columns such as `hotel_id`, `hotel_name`, `hotel_type`, `room_count`, `price_per_night`, `location`, and `meal_services`.
- **Output:**



```
mysql> select * from Hotels;
+----------+-----------------+------------+------------+----------------+-----------------------+-----------------------+
| hotel_id | hotel_name      | hotel_type | room_count | price_per_night | location              | meal_services         |
+----------+-----------------+------------+------------+----------------+-----------------------+-----------------------+
|        1 | Hotel Grand Palace | Suite      |        150 |         200.50 | Paris, France         | Breakfast, Dinner     |
|        2 | Sunrise Resort  | Bungalow   |         80 |         120.75 | Malé, Maldives        | All-inclusive         |
|        3 | Mountain Escape |            |         50 |          75.00 | Aspen, USA            | Lunch, Dinner         |
|        4 | Tokyo Paradise  | Suite      |        100 |         180.00 | Tokyo, Japan          | Breakfast Only        |
|        5 | Beachfront Bliss | Bungalow   |         60 |         210.30 | Bali, Indonesia       | All-inclusive         |
|        6 | Forest Retreat  |            |         40 |          90.50 | Zurich, Switzerland   | Dinner Only           |
|        7 | Ocean Breeze    | Bungalow   |         70 |         250.00 | Sydney, Australia     | All-inclusive         |
|        8 | Royal Stay      | Suite      |        120 |         300.00 | London, England       | Breakfast, Lunch, Dinner |
|        9 | Eco Lodge       |            |         30 |          85.00 | Cape Town, South Africa | Lunch Only          |
|       10 | Seaside Inn     | Bungalow   |         50 |         195.00 | Rio de Janeiro, Brazil | Breakfast, Dinner    |
|       11 | Desert Oasis    | Suite      |        140 |         220.00 | Dubai, UAE            | All-inclusive         |
|       12 | Arctic View     |            |         25 |         110.00 | Reykjavik, Iceland    | Lunch, Dinner         |
|       13 | Urban Delight   | Suite      |        110 |         275.00 | New York, USA         | Breakfast Only        |
|       14 | Island Paradise | Bungalow   |         45 |         230.00 | Honolulu, USA         | All-inclusive         |
|       15 | Nature's Nest   |            |         35 |          95.00 | Banff, Canada         | Dinner Only           |
|       16 | Metropolitan Hotel | Suite      |        200 |         320.00 | Hong Kong, China      | Breakfast, Lunch      |
|       17 | Luxury Stay     | Bungalow   |         90 |         400.00 | Dubai, UAE            | Breakfast, Dinner     |
|       18 | Rustic Haven    |            |         20 |          60.00 | Kyoto, Japan          | Lunch Only            |
|       19 | Sunny Retreat   | Bungalow   |         80 |         150.00 | Miami, USA            | All-inclusive         |
|       20 | Mountain Peak   | Suite      |        160 |         210.00 | Whistler, Canada      | Breakfast, Dinner     |
|       21 | Oceanfront Escape | Bungalow   |         55 |         245.00 | Malé, Maldives        | All-inclusive         |
|       22 | The Lakehouse   |            |         45 |          70.00 | Interlaken, Switzerland | Lunch Only          |
|       23 | City Lights Hotel | Suite      |        125 |         290.00 | Los Angeles, USA      | Breakfast Only        |
|       24 | Coastal Comfort | Bungalow   |         75 |         205.00 | Gold Coast, Australia | All-inclusive         |
|       25 | Woodland Retreat |            |         30 |          80.00 | Jackson Hole, USA     | Dinner Only           |
|       26 | Tropical Bliss  | Bungalow   |         60 |         185.00 | Phuket, Thailand      | All-inclusive         |
|       27 | Skyline View Hotel | Suite      |        140 |         320.00 | Dubai, UAE            | Breakfast, Dinner     |
|       28 | Riverfront Lodge |            |         50 |         100.00 | Salzburg, Austria     | Lunch Only            |
+----------+-----------------+------------+------------+----------------+-----------------------+-----------------------+
28 rows in set (0.00 sec)
```

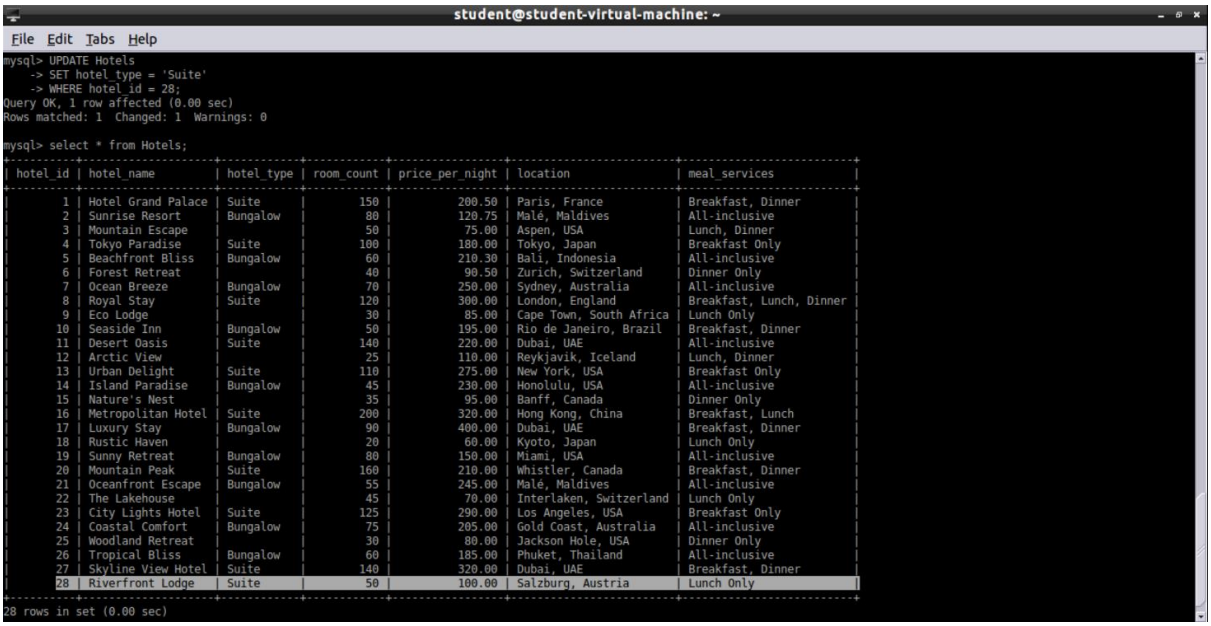
2. Updating a Record

- **Query:**

```
UPDATE Hotels
SET hotel_type='Suite'
WHERE hotel_id = 28;
```

- This query updates the `hotel_type` as '`Suite`' of the hotel with `hotel_id = 28`.

- **Output:**

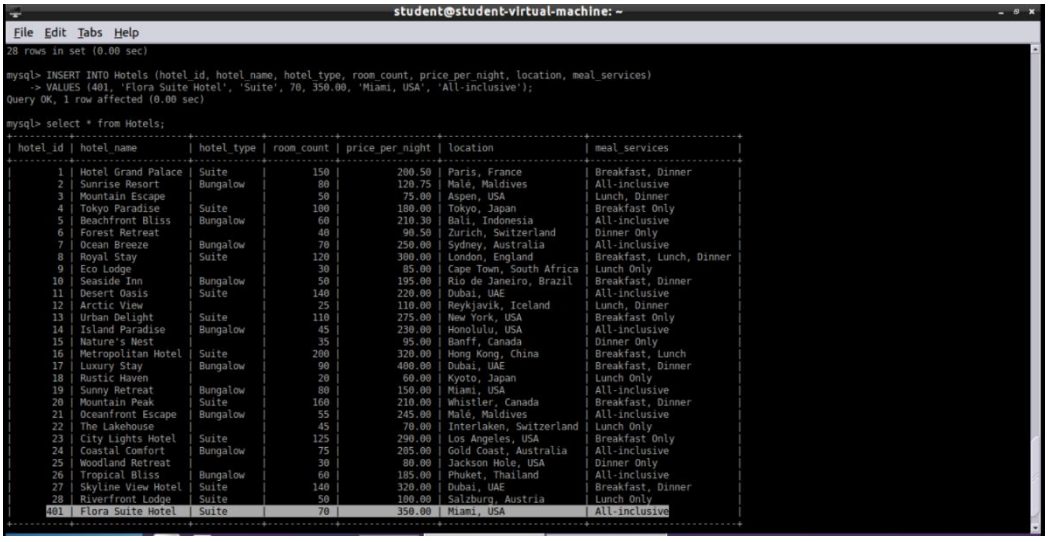


3. Inserting a New Record

- **Query:**

```
INSERT INTO Hotels (hotel_id, hotel_name,hotel_type,
room_count,price_per_night,location,meal_services)
VALUES (401, 'Flora Suite Hotel',
'Suite',70,350.00,'Miami, USA','All-inclusive');
```

- This query adds a new customer with `hotel_id = 401`.
- **Output:**

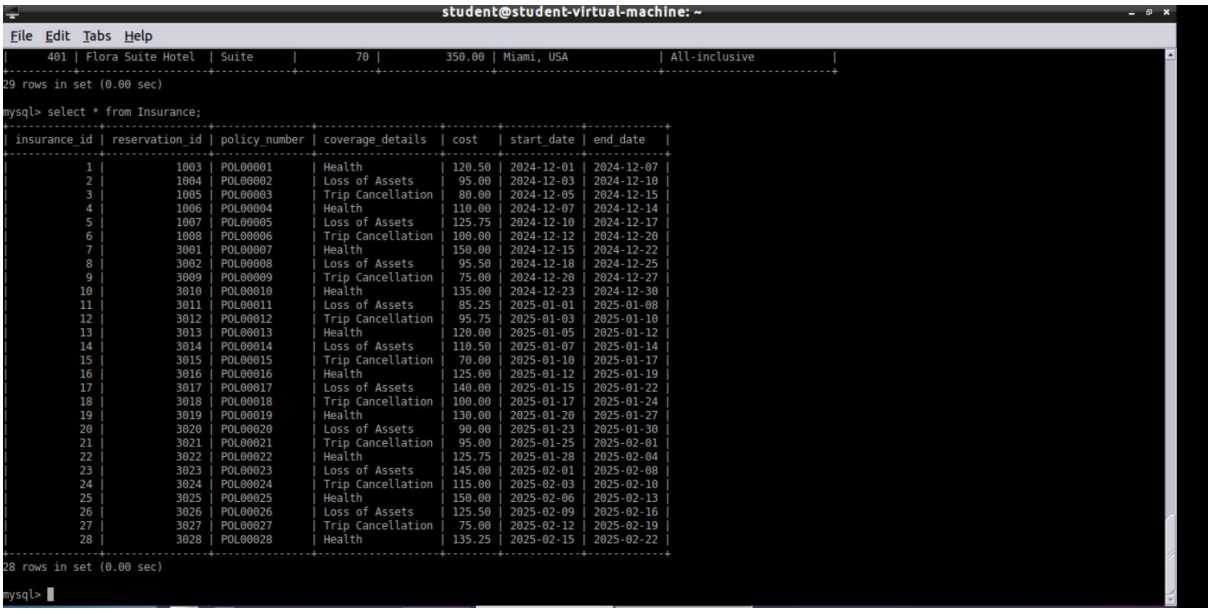


5.8 Queries and Results for Insurance table

1. Viewing Existing Records

- **Query:** `select * from Insurance;`

- This query retrieves all records from the `Insurance` table, columns such as `insurance_id`, `reservation_id`, `policy_number`, `coverage_details`, `cost`, `start_date`, and `end_date`.
- **Output:**



2. Deleting a Record

- **Query:**
```
DELETE FROM Insurance
WHERE coverage_details='Health';
```

  This query removes the entries with `coverage_details = 'Health'`.

- **Output:**

```
mysql> DELETE FROM Insurance
    -> WHERE coverage_details = 'Health';
Query OK, 10 rows affected (0.00 sec)

mysql> select * from Insurance;
+--------------+----------------+---------------+------------------+--------+------------+------------+
| insurance_id | reservation_id | policy_number | coverage_details | cost   | start_date | end_date   |
+--------------+----------------+---------------+------------------+--------+------------+------------+
|            2 |           1004 | POL00002      | Loss of Assets   |  95.00 | 2024-12-03 | 2024-12-10 |
|            3 |           1005 | POL00003      | Trip Cancellation |  80.00 | 2024-12-05 | 2024-12-15 |
|            5 |           1007 | POL00005      | Loss of Assets   | 125.75 | 2024-12-10 | 2024-12-17 |
|            6 |           1008 | POL00006      | Trip Cancellation | 100.00 | 2024-12-12 | 2024-12-20 |
|            8 |           3002 | POL00008      | Loss of Assets   |  95.50 | 2024-12-18 | 2024-12-25 |
|            9 |           3009 | POL00009      | Trip Cancellation |  75.00 | 2024-12-20 | 2024-12-27 |
|           11 |           3011 | POL00011      | Loss of Assets   |  85.25 | 2025-01-01 | 2025-01-08 |
|           12 |           3012 | POL00012      | Trip Cancellation |  95.75 | 2025-01-03 | 2025-01-10 |
|           14 |           3014 | POL00014      | Loss of Assets   | 110.50 | 2025-01-07 | 2025-01-14 |
|           15 |           3015 | POL00015      | Trip Cancellation |  70.00 | 2025-01-10 | 2025-01-17 |
|           17 |           3017 | POL00017      | Loss of Assets   | 140.00 | 2025-01-15 | 2025-01-22 |
|           18 |           3018 | POL00018      | Trip Cancellation | 100.00 | 2025-01-17 | 2025-01-24 |
|           20 |           3020 | POL00020      | Loss of Assets   |  90.00 | 2025-01-23 | 2025-01-30 |
|           21 |           3021 | POL00021      | Trip Cancellation |  95.00 | 2025-01-25 | 2025-02-01 |
|           23 |           3023 | POL00023      | Loss of Assets   | 145.00 | 2025-02-01 | 2025-02-08 |
|           24 |           3024 | POL00024      | Trip Cancellation | 115.00 | 2025-02-03 | 2025-02-10 |
|           26 |           3026 | POL00026      | Loss of Assets   | 125.50 | 2025-02-09 | 2025-02-16 |
|           27 |           3027 | POL00027      | Trip Cancellation |  75.00 | 2025-02-12 | 2025-02-19 |
+--------------+----------------+---------------+------------------+--------+------------+------------+
18 rows in set (0.00 sec)

mysql>
```

3. Altering the Table Structure

- **Query:**
  ```
  ALTER TABLE Insurance
  ADD COLUMN insurance_provider VARCHAR(100);
  ```

- This query adds a new column, `insurance_provider`.
- **Output:**

```
mysql> ALTER TABLE Insurance
    -> ADD COLUMN insurance_provider VARCHAR(100);
Query OK, 18 rows affected (0.07 sec)
Records: 18  Duplicates: 0  Warnings: 0

mysql> select * from Insurance;
+--------------+----------------+---------------+------------------+--------+------------+------------+--------------------+
| insurance_id | reservation_id | policy_number | coverage_details | cost   | start_date | end_date   | insurance_provider |
+--------------+----------------+---------------+------------------+--------+------------+------------+--------------------+
|            2 |           1004 | POL00002      | Loss of Assets   |  95.00 | 2024-12-03 | 2024-12-10 | NULL               |
|            3 |           1005 | POL00003      | Trip Cancellation |  80.00 | 2024-12-05 | 2024-12-15 | NULL               |
|            5 |           1007 | POL00005      | Loss of Assets   | 125.75 | 2024-12-10 | 2024-12-17 | NULL               |
|            6 |           1008 | POL00006      | Trip Cancellation | 100.00 | 2024-12-12 | 2024-12-20 | NULL               |
|            8 |           3002 | POL00008      | Loss of Assets   |  95.50 | 2024-12-18 | 2024-12-25 | NULL               |
|            9 |           3009 | POL00009      | Trip Cancellation |  75.00 | 2024-12-20 | 2024-12-27 | NULL               |
|           11 |           3011 | POL00011      | Loss of Assets   |  85.25 | 2025-01-01 | 2025-01-08 | NULL               |
|           12 |           3012 | POL00012      | Trip Cancellation |  95.75 | 2025-01-03 | 2025-01-10 | NULL               |
|           14 |           3014 | POL00014      | Loss of Assets   | 110.50 | 2025-01-07 | 2025-01-14 | NULL               |
|           15 |           3015 | POL00015      | Trip Cancellation |  70.00 | 2025-01-10 | 2025-01-17 | NULL               |
|           17 |           3017 | POL00017      | Loss of Assets   | 140.00 | 2025-01-15 | 2025-01-22 | NULL               |
|           18 |           3018 | POL00018      | Trip Cancellation | 100.00 | 2025-01-17 | 2025-01-24 | NULL               |
|           20 |           3020 | POL00020      | Loss of Assets   |  90.00 | 2025-01-23 | 2025-01-30 | NULL               |
|           21 |           3021 | POL00021      | Trip Cancellation |  95.00 | 2025-01-25 | 2025-02-01 | NULL               |
|           23 |           3023 | POL00023      | Loss of Assets   | 145.00 | 2025-02-01 | 2025-02-08 | NULL               |
|           24 |           3024 | POL00024      | Trip Cancellation | 115.00 | 2025-02-03 | 2025-02-10 | NULL               |
|           26 |           3026 | POL00026      | Loss of Assets   | 125.50 | 2025-02-09 | 2025-02-16 | NULL               |
|           27 |           3027 | POL00027      | Trip Cancellation |  75.00 | 2025-02-12 | 2025-02-19 | NULL               |
+--------------+----------------+---------------+------------------+--------+------------+------------+--------------------+
18 rows in set (0.00 sec)

mysql>
```

Summary

In this project, we designed and implemented the Vacation Database Management System to address the need for an integrated platform that combines hotel bookings, transport options, activity planning, and voucher management. Through this system, we have successfully demonstrated the practicality of connecting various entities using relational database concepts.

A key decision in this project was reducing the number of entities from the initially required 20 to 12. This was a deliberate choice, as we identified that these 14 entities comprehensively

covered all necessary functionalities and relationships without adding unnecessary complexity.

Key achievements of the project include:

- Developing a schema design that ensures data consistency through foreign keys and relationships between tables.
- Implementing triggers, such as the "Past Reservations" trigger, to track changes and maintain data integrity.
- Executing various queries that demonstrate the functionality of the system, such as data insertion, updates, and deletions across multiple tables.
- Establishing mechanisms for calculating aggregated values, like average ticket prices for activities, and filtering data based on predefined conditions.
- The project also highlighted some challenges, such as designing a schema that accommodates diverse functionalities while ensuring simplicity and efficiency. However, through careful planning and iteration, we achieved a robust system that meets its intended objectives.

This project sets the foundation for future work, including building a user-friendly application interface to complement the database. Additionally, incorporating advanced analytics and real-time updates can further enhance the system's capabilities. Overall, the Vacation Database Management System demonstrates the potential of relational databases to manage complex and interconnected data in a structured and efficient manner.

Constraints Example

```
ALTER TABLE `Accommodation`

  ADD CONSTRAINT `Accommodation_ibfk_1` FOREIGN KEY
(`customer_id`) REFERENCES `Customers` (`customer_id`) ON
DELETE CASCADE,

  ADD CONSTRAINT `Accommodation_ibfk_2a` FOREIGN KEY
(`hotel_id`) REFERENCES `Hotels` (`hotel_id`) ON DELETE
CASCADE ON UPDATE CASCADE,

  ADD CONSTRAINT `fk_accommodation_customer` FOREIGN KEY
(`customer_id`) REFERENCES `Customers` (`customer_id`) ON
DELETE CASCADE ON UPDATE CASCADE,

  ADD CONSTRAINT `fk_accommodation_hotel` FOREIGN KEY
(`hotel_id`) REFERENCES `Hotels` (`hotel_id`) ON DELETE
CASCADE ON UPDATE CASCADE,

  ADD CONSTRAINT `fk_accommodation_reservation` FOREIGN KEY
(`accommodation_id`) REFERENCES `Accommodation`
(`accommodation_id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

The provided ALTER TABLE statement establishes multiple foreign key constraints in the Accommodation table, enhancing data integrity and defining relationships with other tables. Specifically, it creates constraints to link the customer_id column to the Customers table and the hotel_id column to the Hotels table, ensuring that any updates or deletions in the referenced tables are cascaded to maintain consistency. Additional constraints reinforce these relationships with cascading updates and deletions for related records, and the accommodation_id column is also linked to the Accommodation table itself, likely for recursive relationships. This robust configuration guarantees that the Accommodation table consistently reflects changes in related entities, maintaining referential integrity throughout the database.

Example Queries

- From Destination Screen:

```
SELECT continent, SUM(location,id IS NOT NULL AS
destination_count

FROM Destinations

GROUP BY continent;
```

- From Customers Screen:

```
ALTER TABLE Customers

ADD COLUMN customer_type ENUM('Regular', 'VIP', 'New') DEFAULT
'Regular';
```

- From Activity Screen:

```
SELECT AVG(ticket_price) AS average_price FROM Activity;
```

- From Accommodation Screen:

```
INSERT INTO Accommodation (accommodation_id, customer_id,
hotel_id, check_in_date, check_out_date, total_price,
payment_type, room_type)

VALUES (501, 27, 15, '2025-06-01', '2025-06-07', 1200.00,
'Credit', 'Standart');
```

- From Reservations Screen:

```
UPDATE Reservations

SET status = ‘Cancelled’

WHERE reservation_id = 7001;
```

- From Hotels Screen:

```
UPDATE Hotels
SET hotel_type=’Suite’
WHERE hotel_id = 28;
```

- From Insurance Screen:

```
DELETE FROM Insurance
WHERE coverage_details='Health';
```

- From Destinations Screen:

```
UPDATE Voucher SET status = 'Used' WHERE voucher_id = 50001;
SELECT * FROM Destinations LIMIT 5;
```