Case Study: Bus Booking System Implementation

Background

The **Bus Booking System** is designed to help passengers book tickets, make payments, and track bus schedules across different routes. This system stores and manages data regarding buses, routes, schedules, bookings, and payments to optimize bus operations. The goal is to create a reliable and efficient system that helps customers book their travel efficiently while allowing the bus operators to track bookings, payments, and bus status in real-time.

System Components

1. **Bus Management**
    - **Buses Table**: Stores information about buses including the bus name, total seats, bus type (AC/Non-AC), and bus status (Active, Full).
    - **Routes Table**: Defines the routes with origin, destination, distance, and travel time for each route. This helps in managing travel logistics.
    - **Schedules Table**: Links buses to routes and schedules, detailing departure and arrival times. This table is essential for scheduling bus departures on specific routes at given times.
2. **Booking & Payment Management**
    - **Bookings Table**: Keeps track of ticket bookings, including which bus schedule a booking is associated with, the number of seats booked, booking date, and the total amount.
    - **Payments Table**: Records the payment information for each booking, including payment date, payment status, amount paid, and payment method (Credit/Debit Card). This table ensures accurate tracking of financial transactions.
3. **Triggers and Functions**
    - **Triggers**:
        - **After Insert Trigger**: Updates the bus status to 'Full' once all the seats of a bus are booked for a schedule.
        - **After Update Trigger**: Inserts a new payment record when a booking is updated.
        - **After Delete Trigger**: Inserts a cancellation payment record when a booking is deleted.
    - **Table-Valued Functions**:
        - **GetSchedulesByRoute**: Allows users to retrieve available bus schedules for a specific route (from origin to destination).

- Scalar-Valued Function:
  - CalculateFare: Calculates the fare based on the distance of the route. In this case, it assumes a fare of 1.5 per kilometer.

4. Views and Procedures
   - View for Booking Status: Provides a view of all bookings with information about booking size (Small, Medium, Large) and booking value (Low, Medium, High) based on the number of seats booked and total amount.
   - Stored Procedures:
     - InsertBooking: Handles insertion of new bookings with transactional support, ensuring data integrity.
     - UpdateBooking: Allows updates to existing bookings, updating the number of seats and the total amount, again with transactional support.
     - DeleteBooking: Deletes a booking record with transactional support, ensuring that cancellations are handled properly.

System Features

1. Booking Functionality
   - A customer can choose a bus schedule, book seats, and proceed to payment.
   - The InsertBooking procedure is used to insert booking data into the Bookings table and the after_booking_insert trigger updates the bus status if the bus becomes full.
   - The Check seat availability query checks the number of available seats for a given schedule before confirming a booking.
2. Payment Processing
   - After a booking is made, the payment information is recorded in the Payments table. The after_booking_update and after_booking_delete triggers log payment updates and cancellations respectively.
   - Customers can pay using either credit or debit cards, and the View payment status for a booking query allows the operator to track payment status for any booking.
3. Schedule Management
   - The Schedules table connects buses with routes and their departure and arrival times.
   - The GetSchedulesByRoute function is particularly useful for retrieving available bus schedules for a specific route, helping both customers and operators manage travel information efficiently.
4. Bus Availability
   - The system tracks whether a bus is fully booked or has available seats. The Check seat availability query helps in determining how many seats are available for a given

schedule, and the after_booking_insert trigger ensures that buses are marked as 'Full' when there are no seats available.

5. Financial Reports
    - The BookingSummary query provides useful financial insights, such as the total number of bookings, seats booked, and revenue generated for each route. This helps operators assess which routes are most popular and profitable.
    - The Payments table helps operators track the status of all payments, including amounts paid and payment methods.

Use Case Scenario:

Scenario: A customer wants to book a ticket for a bus traveling from Naogaon to Dhaka. The customer accesses the bus booking system and searches for available buses. The system shows all the active buses with routes from Naogaon to Dhaka and their available seats.

1. The customer selects a bus with available seats and proceeds to book 2 seats for the scheduled departure on 26th November 2024 at 8:00 AM.
2. The InsertBooking procedure is invoked, and a new booking record is created in the Bookings table.
3. The after_booking_insert trigger checks if the total booked seats have reached the bus's capacity. If the bus is fully booked, its status is updated to 'Full'.
4. The customer then proceeds to make a payment using their credit card. A new record is created in the Payments table, and the payment status is set to 'Paid'.
5. The customer receives confirmation of their booking, including payment details, and can later check their payment status or booking details through the system.

Business Impact:

- Real-time Booking Updates: Customers can view live availability and make bookings instantly, ensuring that no double bookings occur.
- Efficient Payment Tracking: Payment transactions are logged and tracked, which minimizes the chances of payment discrepancies.
- Operational Efficiency: Bus operators can monitor seat availability, track booking status, and assess the profitability of routes in real-time.

Conclusion

The Bus Booking System designed in this case study offers an efficient, user-friendly solution for managing bus bookings and payments. By incorporating advanced features like triggers,

functions, and stored procedures, the system ensures smooth operations, timely updates on booking status, and seamless payment processing. This implementation would be especially useful for bus companies looking to digitize their operations and provide a better customer experience.