**PROBLEM STATEMENT**

The **Hotel Management System (HMS)** requires advanced database programming to efficiently manage complex operations, such as real-time room availability tracking, automated reservation handling, and dynamic billing processes. The system must handle high volumes of concurrent transactions, ensure data integrity across multiple entities (e.g., guests, rooms, reservations, payments), and support scalability for future growth. Advanced programming techniques, such as stored procedures, triggers, and constraints, are essential to automate workflows, enforce business rules, and optimize query performance, ultimately improving operational efficiency and guest satisfaction.

**JUSTIFICATION**

1. **Triggers**

Triggers are automatically executed in response to specific database events (e.g., INSERT, UPDATE, DELETE) and are essential for automating repetitive tasks, enforcing rules, and maintaining data integrity.

* **Use Case in HMS**:
  + Automatically update room availability (availability\_status) when a guest checks in or checks out.
  + Send notifications or logs when a reservation is updated or deleted.
  + Enforce business rules, such as preventing double bookings by ensuring no two reservations overlap for the same room.
* **Benefit**: Reduces manual intervention, ensures consistent enforcement of business logic, and improves data accuracy.

1. **Cursors**

Cursors are used to retrieve and process multiple rows from a query, one row at a time, making them useful for iterative operations.

* **Use Case in HMS**:
  + Generate detailed billing statements by iterating through services availed by a guest during their stay (e.g., room service, spa, restaurant).
  + Process bulk updates, such as marking multiple reservations as “expired” for guests who did not check in on their scheduled date.
* **Benefit**: Efficiently handles complex operations involving multiple rows, ensuring data is processed sequentially and logically.

1. **Functions**

Functions are reusable blocks of code that return a single value. They are useful for encapsulating frequently used calculations or operations.

* **Use Case in HMS**:
  + Calculate the total bill for a guest, factoring in room charges, taxes, and additional services.
  + Validate guest details, such as ensuring the email address format is correct or the check-in/check-out dates are valid.
* **Benefit**: Enhances modularity and code reusability, reducing redundancy and ensuring consistent implementation of calculations or validations.

1. **Packages**

Packages group related procedures, functions, and other database objects, improving modularity and organization.

* **Use Case in HMS**:
  + Group related operations for reservation management, such as procedures for creating, updating, and canceling reservations.
  + Group billing-related operations, such as generating invoices, calculating taxes, and processing payments.
* **Benefit**: Improves maintainability, reduces dependency issues, and enhances system organization by encapsulating related functionality in a single unit.

1. **Auditing Techniques**

Auditing involves tracking and recording changes or access to the database, ensuring accountability and security.

* **Use Case in HMS**:
  + Track changes made to reservations, such as updates to check-in/check-out dates or cancellations.
  + Monitor who accessed sensitive guest information and when, ensuring compliance with data privacy regulations.
  + Detect and log suspicious activities, such as unauthorized attempts to modify billing records.
* **Benefit**: Enhances security, ensures accountability, and supports compliance with legal and regulatory requirements.