Task: Build an Automated Insurance Management System

Objective: You are tasked with creating a fully automated SQL-based insurance management system. The system will manage customer records, agents, policies, claims, and payments. It must handle scenarios where agents manage multiple policies, customers make multiple claims, and payments are recorded for various methods. Your implementation must incorporate advanced automation techniques using SQL.

Step 1: Design the Database Schema

Create the following tables:

1. **Customers**
2. **Agents**
3. **Policies**
4. **Claims**
5. **Payments**

Step 2: Populate Tables with Data

Populate each of the following tables with the data provided:

- **1000 customers**
- **50 agents** managing multiple policies
- **1000 policies**
- **1500 claims**
- **2000 payments**

Step 3: SQL Queries and Procedures

1. **Insert Data Automatically**:
   Write SQL scripts to insert data into each table. The primary keys (customer_id, agent_id, etc.) will auto-increment.
2. **Automated Claim Status Updates**:
   Write a trigger or scheduled job that automatically updates the status of claims based on conditions such as claim amount or random logic.
3. **Generate Reports**:
   Write queries to:
   - List policies sold by each agent.
   - Display claims with different statuses.
   - Show payment history per customer and policy.

Step 4: Advanced Automation Techniques

Triggers:

1. **Auto-expiring policies**:
   Create a trigger that automatically expires policies after the end date.
2. **Preventing duplicate claims**:
   Write a trigger to prevent customers from submitting duplicate claims for the same incident.
3. **Auto-calculating agent commissions**:
   Write a trigger that automatically calculates and updates the commission earned by agents whenever a new policy is created.
4. **Auto-approving claims below a threshold**:
   If a claim is below a certain amount (e.g., ₹10,000), write a trigger to automatically approve it.

Stored Procedures:

1. **Automating policy renewals**:
   Write a stored procedure to automatically renew policies that are expiring, if the customer has made full payments and has no outstanding claims.
2. **Automating payments**:
   Create a stored procedure to automatically process recurring payments for policies that are paid in installments.
3. **Dynamically generating reports**:
   Implement a stored procedure that dynamically generates reports on the number of policies, claims, and payments processed each day.

Audit Tables:

1. **Tracking all changes in key tables**:
   Implement audit tables to log any changes made to important tables such as Policies, Claims, and Payments. The audit tables will track what data was changed, when, and by whom.