



HEALTHCARE MANAGEMENT SYSTEM

Database Design & Billing Automation Documentation

1. Project Overview

The Healthcare Management System is a relational database designed to manage hospital operations such as:

- Patient management
- Doctor management
- Encounter handling (OPD & IPD)
- Medical records
- Diagnostic tests
- Procedures and surgeries
- Prescription management
- Automated billing and payments
- Insurance claim processing

The system focuses on **automation of the billing workflow** using **triggers, stored procedures, and generated columns** to ensure:

- Data consistency
 - Reduced manual effort
 - Real-time financial calculations
-

2. Project Objectives

- To design a scalable and normalized healthcare database
 - To automate invoice generation
 - To implement event-driven billing
 - To handle partial and insurance-based payments
 - To maintain audit-ready financial data
-

3. Core Modules

3.1 Patient Management

Stores demographic and contact details of patients.

3.2 Doctor Management

Maintains doctor profile, specialization, employment type, and license details.

3.3 Encounter Management

Each hospital visit is stored as an encounter:

- OPD
- IPD

This acts as the **central link** between clinical and billing modules.

3.4 Medical Records

Stores patient history:

- Symptoms
- Diagnosis
- Allergies
- Past medical history
- Ongoing conditions

3.5 Clinical Services

- Tests
- Procedures
- Surgeries
- Prescriptions

4. Automated Billing Architecture

The system uses **event-driven automation**.

4.1 Invoice Auto Generation

When a new encounter is inserted:

Trigger:

`after insert on encounter`

→ Automatically creates an invoice.

4.2 Invoice Number Generation

Handled using:

- Stored function: `invoice_number_generator()`

Features:

- Auto increment logic
 - First invoice handling
 - Unique invoice format (INV1, INV2...)
-

4.3 Auto Addition of Billable Items

When the following are inserted:

Clinical Entry Automatically Added to Invoice

Test Report	Test charge
Procedure Report	Procedure charge
Surgery Report	Surgery charge
Prescription	Medicine charge

Each trigger:

- Fetches `encounter_id`
 - Finds related invoice
 - Fetches cost
 - Inserts into `invoice_item`
-

4.4 Auto Calculation of Item Total

Trigger:

`before insert on invoice_item`

Formula:

`total_price = quantity × unit_price`

4.5 Auto Calculation of Invoice Total

Trigger:

`after insert on invoice_item`

Recalculates:

```
total_amount = sum of all invoice items
```

4.6 Net Payable Calculation (Generated Column)

Net payable is **not manually updated**.

It is defined as:

```
net_payable_amount = total_amount - discount_amount + tax_amount
```

This ensures:

- No inconsistency
 - No trigger recursion
 - Real-time calculation
-

5. 💰 Tax & Discount Management

Handled using stored procedures:

setDiscountAmount(invoiceId, amount)

- Validates invoice existence
- Prevents negative values

setTaxAmount(invoiceId, amount)

setTaxByPercentage(invoiceId, percentage)

- Calculates tax automatically
-

6. 🛡️ Error Handling & Validation

The system uses:

```
SIGNAL SQLSTATE '45000'
```

To handle:

- Invalid specialization
- Invalid employment type
- Incorrect invoice ID
- Negative tax or discount

This ensures controlled and secure operations.

7. Insurance & Payment Flow

Supports:

- Full payment by patient
- Partial payment
- Insurance claim settlement
- Claim rejection scenarios

Payment status is updated based on:

- Total paid amount
 - Net payable amount
-

8. System Workflow

Step 1 – Insert Encounter



Step 2 – Invoice Auto Generated



Step 3 – Insert Test / Procedure / Surgery / Prescription



Step 4 – Invoice Item Auto Added



Step 5 – Total Amount Auto Calculated



Step 6 – Apply Tax / Discount



Step 7 – Net Payable Auto Calculated



Step 8 – Insert Payment



Step 9 – Payment Status Updated

9. Advanced Concepts Used

- Normalized relational schema
 - Foreign key constraints
 - Event-driven triggers
 - Stored procedures
 - Stored functions
 - Generated columns
 - Transaction-safe operations
 - Custom error handling
-

10. Key Features

- ✓ Fully automated billing
 - ✓ Real-time invoice updates
 - ✓ Insurance claim handling
 - ✓ Partial payment support
 - ✓ Scalable encounter-based design
 - ✓ Data consistency without redundancy
 - ✓ Production-style validation
-

11. Advantages of the System

- Reduces manual billing effort
 - Eliminates calculation errors
 - Ensures data integrity
 - Improves hospital workflow efficiency
 - Provides real-time financial visibility
-

12. Future Enhancements

- Medicine stock management
 - Role-based access control
 - Doctor revenue analytics
 - Monthly financial reporting
 - Appointment scheduling module
 - Dashboard & visualization integration
-

13. Conclusion

This system demonstrates how a healthcare database can be transformed into an **intelligent, automated, and transaction-safe billing engine** using MySQL.

It reflects real-world hospital workflows and applies advanced database concepts for efficient data management.

14. Developer Details

Developer Name: Pritish Ramesh Pawar

Course: Java Full Stack

Project Type: HEALTHCARE MANAGEMENT SYSTEM (MySQL)

Year: 2026