IN-HOME NURSING BOOKINGS MANAGEMENT SYSTEM

PL/SQL Practicum Project

Student Name: GASANA Gerardine
Student ID: 28888
Course: PL/SQL
Date: December 9, 2025

1. PROJECT OVERVIEW

This project develops a complete healthcare coordination system that handles everything from patient service requests to nurse dispatching and visit logging. The system connects patients requiring home care with qualified nurses in real-time, making healthcare delivery smoother and reducing dangerous errors that happen with manual scheduling.

2. PROBLEM STATEMENT

From observing home-care services in Kigali, I've noticed several critical problems with how nursing visits are currently managed:

- **Manual Scheduling:** Coordinators use phone calls and paper diaries, leading to accidental **double-bookings**.
- **Missed Visits:** Without automated reminders, nurses sometimes forget appointments or arrive late.
- **Matching Inefficiency:** It is difficult to quickly find a nurse who has the right skills (e.g., wound care vs. pediatric) and lives near the patient.
- **Lack of Compliance:** There is no reliable "audit trail" to prove who changed a record or if a visit actually occurred.
- **Billing Disputes:** Patients are sometimes charged for visits that were shorter than promised or didn't happen at all.

These issues lead to compromised patient health, frustrated staff, and compliance risks. I want to solve this using a database system with smart automation and strict audit controls.

3. MY PROPOSED SOLUTION

I will build an intelligent nursing management system with these features:

Core Features:

- **Digital Booking Engine:** Patients or their families enter care requests directly into the system.
- **Smart Dispatcher:** Automatically matches nurses to patients based on qualification, availability, and proximity.
- **Visit Logging:** Nurses log "Actual Start" and "Actual End" times to verify service delivery.
- **Automated Reminders:** Sends alerts to nurses 24 hours and 2 hours before a visit (simulated)[7].
- **Compliance Vault:** A secure audit system that tracks every change made to sensitive data.

Key Innovations:

1. **Intelligent Matching Algorithm:** Prioritizes nurses by **Skills + Distance + Rating** rather than just availability.
2. **Strict Weekday Protection:** Implements business rules that restrict administrative configuration changes to weekends only, ensuring system stability during peak operational days.
3. **Conflict Detection:** Automatically blocks a booking if the nurse is already scheduled for that time slot (Double-booking prevention)[9].
4. **Service-Based Duration Logic:** The system calculates expected visit times based on the specific medical service (e.g., "Injection" = 30 mins, "Postnatal Care" = 2 hours).
5. **Visit Verification:** Compares Scheduled_Time vs. Actual_Time to flag late arrivals automatically.

4. DATABASE DESIGN

I will create 10 interconnected tables to support this logic:

Main Tables:

1. **USERS** - Base entity for authentication: user_id (PK), username, password_hash, role (PATIENT/NURSE/ADMIN), created_at.

2. **PATIENTS** - Medical profile: patient_id (PK/FK), DOB, home_address, medical_notes, emergency_contact[12].

3. **NURSES** - Professional profile: nurse_id (PK/FK), license_no, specialization, current_rating, hourly_rate, is_verified.

4. **SERVICES** - Service catalog: service_id (PK), service_name (e.g., Wound Care), base_price, default_duration_minutes, description.

5. **BOOKINGS** - The core transaction: booking_id (PK), patient_id (FK), nurse_id (FK), service_id (FK), status (PENDING/CONFIRMED/COMPLETED), scheduled_time, created_date[14].

6. **VISIT_LOGS** - Execution details: log_id (PK), booking_id (FK), actual_start_time, actual_end_time, clinical_notes, outcome_status.

7. **NURSE_AVAILABILITY** - Scheduling slots: slot_id (PK), nurse_id (FK), day_of_week, start_time, end_time, is_booked.

8. **AUDIT_LOGS** - Security tracking: log_id (PK), table_name, action_type (INSERT/UPDATE), user_performing, old_value, new_value, timestamp.

9. **ADMIN_CONFIG** - System settings: config_key (PK), config_value, last_updated (Target for the "Weekday Restriction" rule).

10. **PAYMENTS** - Financial records: payment_id (PK), booking_id (FK), amount, method, status, timestamp.

5. PL/SQL COMPONENTS I WILL USE

This project demonstrates advanced PL/SQL concepts:

Collections

- **VARRAY** to store a nurse's list of certified skills (e.g., ['CPR', 'Pediatrics', 'Wound Care']).
- **Nested Table** for storing the history of status changes for a single booking.
- **Associative Array** for caching service pricing to speed up calculations.

Records

- **Booking Summary Record** combining patient name, address, and service details for the nurse's view.
- **Audit Record** capturing the "Who, What, When" of system changes.

GOTO Statements

- Jump to MATCH_FAIL handler if no nurses are found within range.
- Jump to AUDIT_LOG section after a sensitive transaction is completed.

Cursors

- **Explicit Cursor** to fetch all "Pending" bookings that need matching.
- **Cursor Loop** to identify nurses who have not logged their visit notes by the end of the day.
- **Parameter Cursor** to calculate a nurse's average rating over a specific date range.

Procedures

- SP_REQUEST_BOOKING: Validates patient inputs and creates a booking request.
- SP_AUTO_MATCH_NURSE: Runs the logic to find the best available nurse for a specific request.
- SP_LOG_VISIT: Allows nurses to record their start/end times and clinical notes.
- SP_PROCESS_PAYMENT: Calculates final cost based on service duration and processes transactions.

Triggers

- TRG_PREVENT_WEEKDAY_CHANGES: **(Critical Requirement)** Prevents INSERT/UPDATE/DELETE on the ADMIN_CONFIG table during weekdays (Mon-Fri).

- TRG_PREVENT_DOUBLE_BOOKING: Checks for overlapping time slots before a booking is confirmed (BEFORE INSERT).
- TRG_AUDIT_SENSITIVE_DATA: Automatically writes to AUDIT_LOGS whenever a booking status changes (AFTER UPDATE).
- TRG_UPDATE_NURSE_RATING: Recalculates average rating when a new review is submitted.

Exception Handling

- E_NO_NURSE_AVAILABLE: Custom exception when matching logic fails to find a candidate.

- E_INVALID_VISIT_TIME: Raised if a nurse tries to log a visit time in the future.
- E_RESTRICTED_ACCESS: Raised when an admin tries to change config on a weekday.

Packages

- PKG_BOOKING_ENGINE: Groups procedures for requests, matching, and scheduling.
- PKG_NURSE_PORTAL: Contains procedures for nurses to view schedules and log visits.
- PKG_COMPLIANCE: Manages audit logging and restriction rules.

6. EXPECTED RESULTS

When I finish this system, it will:

- **Eliminate Double Bookings** completely through database constraints.

- **Ensure 100% Auditability** of all system changes for compliance.
- **Optimize Nurse Utilization** by matching them efficiently to nearby patients.
- **Reduce Missed Visits** by providing clear schedules and reminders to staff.

- **Provide Data-Driven Insights** on which medical services are most in demand.

- **Enforce Operational Discipline** through the strict weekday maintenance restriction rules.

7. CONCLUSION

This project shows how PL/SQL database features can transform healthcare coordination. By using triggers for compliance (audit trails), cursors for matching logic, and procedures for workflow automation, the system makes home care delivery safer and more reliable.

The innovation isn't just digitizing the booking form, it's adding **intelligence** that prevents conflicts, ensures visits happen on time, and strictly controls administrative access[32]. This makes the service professional and trustworthy.