Telecom Management System - Database Requirements Document

1. Introduction

This document outlines the detailed specifications for the design and implementation of the Telecom Management System (TMS) database. The system is intended to support and streamline the following operations:

- Customer account and billing management
- SIM card provisioning
- Service plan and subscription handling
- Usage tracking and quota enforcement
- Support ticket management
- Automated business workflows
- Comprehensive data reporting

The primary goal is to maintain data integrity and automate essential telecom business processes.

2. System Overview

The database will support the following core functions:

- **Customer Management:** Creating and managing accounts, tracking balances, and updating customer profiles.
- **SIM Card Operations:** SIM activation, deactivation, and linking to customer accounts.
- **Service Provisioning:** Managing service plans and subscriptions and enforcing quota policies.
- **Usage Tracking:** Real-time monitoring of data, SMS, and voice usage.
- **Support System:** Handling support ticket lifecycle from creation to resolution.
- **Employee Management:** Managing departments and employee roles.
- **Automation:** Performing balance deductions, status updates, and notifications.
- **Reporting:** Generating reports on active subscriptions, usage analytics, and plan popularity.

3. Entity Definitions

3.1 Customer

- **Personal Info:** First Name, Middle Name, Last Name, Gender, Birth Date
- **Contact Info:** Email, City, Street Address, Door Number
- **Account Info:** Registration Date, National ID (unique), Balance
- **Payment Info:** Encrypted Card Number

3.2 SIM Card

- **Identifiers:** ICCID (PK), IMSI (15 digits), Phone Number (unique)
- **Status:** Active, Suspended, Terminated

3.3 Service Plan

- **Plan Info:** Name, Plan ID, Monthly Fee
- **Service Limits:** Data (MB/GB), SMS Count, Voice Minutes

3.4 Subscription

- **Period:** Start Date, End Date (30-day default)
- **Status:** Active, Suspended, Expired
- **Remaining Quotas:** Data, SMS, Minutes

3.5 Usage Record

- **Consumption:** Data Used, SMS Sent, Minutes Consumed
- **Timestamp:** Date and Time

3.6 Department

- **Info:** Department Name, ID, Manager ID
- **Contact:** Email, Phone Number

3.7 Employee

- **Personal Info:** First Name, Middle Name, Last Name
- **Employment Info:** Role, Hire Date, Salary

3.8 Support Ticket

- **Ticket Info:** ID, Creation Date, Resolved Date
- **Issue Details:** Type (e.g., Network, Billing), Priority (High/Medium/Low)
- **Status:** Open, In-Progress, Closed

4. Relationships

- One **Customer** can have multiple **SIM Cards** and **Subscriptions**.
- One **Service Plan** can be linked to multiple **Subscriptions**.
- One **Subscription** may have multiple **Usage Records**.
- One **Department** can have multiple **Employees**.
- One **Employee** can handle multiple **Support Tickets**.
- One **Customer** can submit multiple **Support Tickets**.

5. Functional Requirements

5.1 Stored Procedures:

- The `Customer Subscription` procedure checks if a customer's balance is sufficient for the selected plan. If the balance is enough, the system subscribes the customer to the specified plan.
- The `Insert into Usage Record` procedure captures and stores all usage details for a customer. It ensures that the insertion occurs only when the customer still has remaining quota, preventing overuse beyond the subscribed plan limits.
- The `Update Status Support Ticket` procedure changes the status of a support ticket. This update is triggered after the reported issue has been successfully resolved.
- The `Get Remaining Quota` procedure retrieves the unused quota for each active plan subscribed to by a specific customer. It can be executed at any time to monitor current quota availability.
- The `Get Customer Usage` procedure retrieves detailed usage data for all plans, both active and inactive, associated with a specific customer. It can be executed at any time to review the customer's full usage history.

5.2 Triggers:

- The `UsageRecords` **Trigger** automatically updates the remaining quota in the **Subscriptions** table whenever a new usage record is inserted. This ensures real-time synchronization between recorded usage and available quota.

- The `Subscriptions` **Trigger** automatically updates the customer's balance in the **Customers** table whenever changes occur in the **Subscriptions** table. This keeps the customer's balance accurately reflected after subscription modifications.

- The `new_support_ticket` trigger automatically sends a notification to the assigned employee whenever a new support ticket is created. This ensures timely awareness and prompt handling of incoming support requests.

- The trigger automatically updates the `Subscriptions_status` in the **Subscriptions** table when either the subscription end date is reached or the quota is fully used. This ensures that expired or exhausted subscriptions are properly marked as inactive.

5.3 Views:

- The `v_active_customers` view displays all customers who currently have active subscriptions or ongoing services. It provides a quick reference to identify customers with active engagement in the system.

- The `v_get_num_of_tickets_per_employee` view shows the total number of support tickets assigned to each employee. It helps track employee workload and monitor ticket distribution.

- The `v_active_support_tickets` view lists all currently open or unresolved support tickets. It provides an overview of ongoing customer issues requiring attention.

- The `v_active_Subscriptions` view displays all subscriptions that are currently active and valid. It helps monitor which customers are actively enrolled in plans or services.

- The `v_num_of_active_Subscriptions_per_plan` view shows the count of active subscriptions grouped by each plan. It provides insights into the popularity and usage of different plans.