Designing a Database System for Koodo : A Case Study

Winlove Chryseis T. Navarro

Professional Certificate – Data Science

Canada

2025

Linkedin: Winlove Navarro

Table of Contents

Abstract

This article presents the design and implementation of a comprehensive database system for Koodo's telecommunications operations. The system addresses critical data management challenges in subscription-based telecommunications services through an integrated approach that encompasses customer relationship management, subscription lifecycle tracking, inventory control, and financial transaction processing.

Using a systematic methodology involving entity identification, relationship mapping, and constraint implementation, we developed an eight-entity relational database schema that supports real-time operational needs and strategic business intelligence. The resulting system integrates customer data, subscription management, product catalog, shipment tracking, payment processing, support ticketing, and employee management into a cohesive platform.

Key outcomes include streamlined data access through optimized SQL views, enhanced data integrity through comprehensive constraints, and improved operational efficiency through integrated business intelligence capabilities. This design demonstrates how strategic database architecture can transform telecommunications operations by providing a single source of truth for customer lifecycle management and enabling data-driven decision making across all operational departments.

I. Introduction

Telecommunications have become an essential part of society, connecting people, businesses, governments globally. They play a crucial role in social interaction, emergency response and social interaction. In today's digital age, the telecommunication industry generates massive volumes of data every second – from customer details, billing information to call records, and network performance.

Efficiently managing and analyzing these data is crucial for service reliability and customer satisfaction, and this use case highlights the importance of databases in addressing these challenges.

This use case addresses these challenges by designing and implementing a comprehensive relational database system specifically tailored to telecommunications operations. The system consolidates customer relationship management, subscription lifecycle tracking, inventory control, payment processing, and support services into a single, coherent platform. By establishing clear data relationships and implementing robust integrity constraints, the database ensures data accuracy while enabling sophisticated business intelligence capabilities.

II. Mission and Objectives

This use case aims to transform raw business data into actionable insights by implementing strategic database views that integrate customer lifecycle management, order fulfillment operations, inventory control, and financial tracking, enabling comprehensive business oversight and data-driven decision making across all operational departments.

To achieve this mission, the specific objectives are:

- to establish single source of truth for customer lifecycle metrics
- to standardize customer data definitions across all departments
- to create scalable view architecture that supports future business requirements
- to enable real-time dashboards for subscription metrics and revenue tracking

III. Methodology

This database was developed using a methodology that models business operations and its comprehensive operational analysis.

a. Entity Identification Through Business Analysis

This phase involves analyzing Koodo's business operations to identify the system's components. Key processes such as subscription life cycle, integrated customer support, product sales, recurring billing and customer service operations lead to the identification of the following entities: customers. subscriptions, products, shipments, supporttickets, payments, employees and inventory. Each entity contains data essential for specific business operations.

b. Attribute Specification Based on User Needs

This phase defines all the attributes and fields per entity considering all the business requirements and user needs. For example, customers table contains all necessary customer information such as customer name, address, contact information and status which is vital for daily operations and analysis.

c. Defining Relationships and Cardinality

After defining the entities and its attributes, the inter-table relationships were then analyzed to identify how it interacts with from different parts of the business. For this use case, the following cardinalities were identified (one-to-one, one-to-many, and many-to-many) and is directly reflected from the defined business rules. Here are several examples that is identified: (1:M) - One customer can have multiple subscriptions; (M:M) - Many tickets assigned to one employee.

d. Creating an Entity Relationship Diagram (ERD)

This phase covers the construction of an Entity-Relationship Diagram (ERD) based on the defined entities, attributes, and relationships from the previous phases. This visual representation maps the database's logical structure, displaying tables, keys, and relationship flows. The ERD functions as a blueprint for physical database implementation.

e. Enforcing Data Integrity Constraints

This phase enforces comprehensive integrity constraints to maintain data accuracy and consistency. These rules ensure all system data remains valid and reliable throughout operations.

f. Implementing Database Views for Business Insights

SQL views were developed to streamline access to consolidated business intelligence. These pre-configured queries enable rapid reporting while preserving system performance and delivering clear insights into operational metrics.

IV. Database Design

Database design forms the foundation of effective data management systems. This systematic process involves analyzing Koodo’s business requirements and structuring data to support operational needs.

a. Entities and Fields

The database structure comprises eight core entities, each containing specific fields that capture essential business data.

Table 1

customers

Field Name	Data Type	Constraints	Description
customer_id	int	PRIMARY KEY, AUTO_INCREMENT, NOT NULL	Unique identifier for each customer
first_name	varchar(50)	NOT NULL	Customer's first name
last_name	varchar(50)	NOT NULL	Customer's last name
contact_info	varchar(20)		Primary contact number (phone)
email_address	varchar(100)	UNIQUE, NOT NULL	Customer's email address for communication
street_address	text	NOT NULL	Customer's street address line
city_address	varchar(50)	NOT NULL	Customer's city address
postal_code	varchar(50)	NOT NULL	Customer's postal/zip code
Status	varchar(20)	DEFAULT 'Active'	Customer account status (Active, Inactive, Suspended)
CreatedDate	date	DEFAULT CURRENT_DATE	Date when customer account was created

Table 1 is an entity table that stores customer information, addresses and account status and is essential for the customer relationship management of this organization.

Table 2

subscriptions

Field Name	Data Type	Constraints	Description
subscription_id	int	PRIMARY KEY, AUTO_INCREMENT, NOT NULL	Unique identifier for each subscription
customer_id	int	FOREIGN KEY (Customer.customer_id), NOT NULL	Reference to subscribing customer
shipment_id	int	FOREIGN KEY (Shipment.shipment_id), NOT NULL	Reference to the shipped product
product_id	int	FOREIGN KEY (Product.product_id), NOT NULL	Reference to subscribed product
start_date	date	NOT NULL	Date when subscription begins
end_date	date		Date when subscription ends (NULL for ongoing)
billing_amount	decimal(10,2)	NOT NULL, CHECK (BillingAmount >= 0)	Amount charged per billing cycle
next_billing_date	date	NOT NULL	Date of next scheduled billing
status	varchar(20)	DEFAULT 'Active'	Subscription status (Active, Paused, Cancelled, Expired)

Table 2 is a fact table that manages subscription services including billing cycles, pricing tiers, customer assignments and subscription status. It also handles the recurring revenue management and subscription lifecycle tracking.

Table 3
products

Field Name	Data Type	Constraints	Description
product_id	int	PRIMARY KEY, AUTO_INCREMENT, NOT NULL	Unique identifier for each product
inventory_id	int	FOREIGN KEY (Inventory.inventory_id), NOT NULL	Reference to Inventory
product_name	varchar(100)	NOT NULL	Name/title of the product
product_type	varchar(50)	NOT NULL	Category or type of product (Plan, Device)
price	decimal(10,2)	NOT NULL, CHECK (Price >= 0)	Product price in base currency
is_active	boolean	DEFAULT TRUE	Whether product is currently available for sale

Table 3 is an entity table that stores core product information including product names, types, pricing and availability status. It serves as the foundation for inventory management and sales operations.

Table 4
shipments

Field Name	Data Type	Constraints	Description
shipment_id	int	PRIMARY KEY, AUTO_INCREMENT, NOT NULL	Unique identifier for each shipment
product_id	int	FOREIGN KEY (Product.product_id), NOT NULL	Reference to shipped product
tracking_number	varchar(100)	UNIQUE, NULL	Carrier-provided tracking number
carrier	varchar(100)		Shipping company name (UPS, FedEx, USPS, etc.)
status	varchar(20)	DEFAULT 'Preparing'	Shipment status (Preparing, Shipped, In Transit, Delivered)
ship_date	date		Date when item was shipped
delivery_date	date		Date when item was delivered

Table 4 is an entity table that manages the physical delivery of products from warehouses to customers. This is important for order fulfillment and delivery tracking.

Table 5

supporttickets

Field Name	Data Type	Constraints	Description
ticket_id	int	PRIMARY KEY, AUTO_INCREMENT, NOT NULL	Unique identifier for each support ticket
customer_id	int	FOREIGN KEY (Customer.customer_id), NOT NULL	Reference to customer who created ticket
employee_id	int	FOREIGN KEY (Employee.employee_id)	Reference to assigned support employee
category	varchar(100)	NOT NULL	Ticket category (Technical, Billing, General Inquiry, etc.)
priority	varchar(20)	DEFAULT 'Medium'	Ticket priority level (Low, Medium, High, Urgent)
status	varchar(20)	DEFAULT 'Open'	Ticket status (Open, In Progress, Resolved, Closed)
resolution	text		Description of how the issue was resolved
customer_satisfaction	int	CHECK (CustomerSatisfaction BETWEEN 1 AND 5), NULL	Customer satisfaction rating (1-5 scale)
created_date	date	DEFAULT CURRENT_DATE	Date when ticket was created
resolved_date	date	NULL	Date when ticket was resolved

Table 5 is an entity table that tracks customer inquiries, complaints and service requests. This manages customer service and issue resolution.

Table 6

payments

Field Name	Data Type	Constraints	Description
payment_id	int	PRIMARY KEY, AUTO_INCREMENT, NOT NULL	Unique identifier for each payment transaction
subscription_id	int	FOREIGN KEY (Subscription.subscription_id)	Reference to related subscription
payment_date	date	DEFAULT CURRENT_DATE	Date when payment was processed
amount	decimal(10,2)	NOT NULL, CHECK (Amount >= 0)	Payment amount
method	varchar(50)	NOT NULL	Payment method used (Credit Card, PayPal, Bank Transfer, etc.)
status	varchar(20)	DEFAULT 'Pending'	Payment status (Pending, Completed, Failed, Refunded)

Table 6 is an entity table that records all payment transactions, methods, amounts and status. This links subscription billing to actual payment collection.

Table 7

employees

Field Name	Data Type	Constraints	Description
employee_id	int	PRIMARY KEY, AUTO_INCREMENT, NOT NULL	Unique identifier for each employee
first_name	varchar(100)	NOT NULL	Employee's first name
last_name	varchar(100)	NOT NULL	Employee's last name
role	varchar(100)	NOT NULL	Employee's job role/position
is_active	boolean	DEFAULT TRUE	Whether employee is currently active

Table 7 is an entity table that manages all support team members, their roles and availability status for ticket assignments and workload distribution.

Table 8

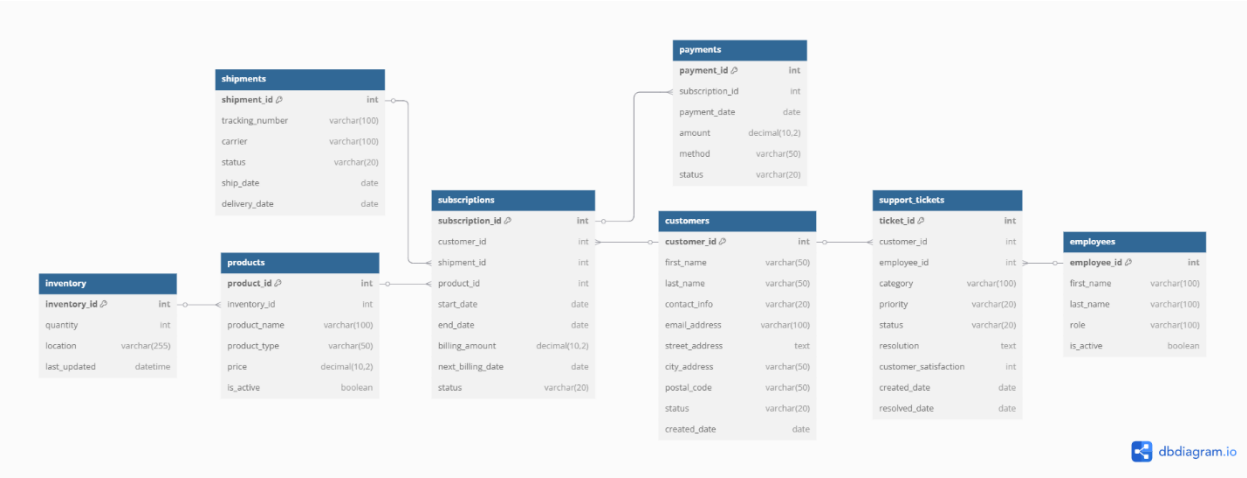
inventory

Field Name	Data Type	Constraints	Description
inventory_id	INT	PRIMARY KEY, AUTO_INCREMENT, NOT NULL	Unique inventory record identifier
quantity	INT	NOT NULL, CHECK (Quantity >= 0)	Current stock quantity
location	VARCHAR	DEFAULT 'Main Warehouse'	Storage location
last_updated	DATETIME	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	Last inventory update

Table 8 is an entity table that tracks real-time stock management and warehouse operations. This is essential for fulfillment and supply chain management.

b. Relationships

This section displays the use case's Entity Relationship Diagram (ERD) that provides a visual representation of how each entity relates to one another.



Based on Koodo's operations and processes, the following relationships were defined.

- **products ←→ inventory (1:1):** One product has one inventory record
- **products → subscriptions (1:M):** One product can be shipped multiple times
- **products → shipments (1:M):** One product can be shipped multiple times
- **customers → subscriptions (1:M):** One customer can have multiple subscriptions
- **subscriptions → payments (1:M):** One subscription generates multiple payment transactions

- **customers → support_tickets (1:M):** One customer can create multiple support tickets
- **employees → support_tickets (1:M):** One employee can handle multiple support tickets
- **subscriptions → customers (M:M):** Many subscriptions belong to one customer
- **payments → subscriptions (M:M):** Many payments belong to one subscription
- **support_tickets → customers (M:M):** Many tickets belong to one customer
- **support_tickets → employees (M:M):** Many tickets assigned to one employee

V. Data Integrity and Constraints

The database incorporates multiple constraint types to maintain data integrity.

- Primary keys ensure unique identification across all entities while preventing null values and duplicate records.
- Foreign keys establish referential integrity by linking related tables and maintaining consistent relationships.
- Check constraints validate business rules, such as ensuring positive values for prices and quantities.
- Not null constraints guarantee that critical fields always contain required information.
- Unique constraints applied to fields like email addresses and tracking numbers prevent duplicate entries within specific columns.

These constraints collectively safeguard data quality and enforce business logic at the database level.

VI. SQL Queries

SQL queries were developed to support operational reporting and strategic decision-making. These queries enable rapid data retrieval for key business metrics, performance analysis, and operational insights, streamlining the reporting process while maintaining system efficiency.

Figure 1

Active Subscriptions View

```sql
-- Active Subscriptions View
CREATE VIEW active_subscriptions AS
SELECT
    s.subscription_id,
    c.first_name,
    c.last_name,
    p.product_name,
    s.start_date,
    s.end_date,
    s.billing_amount,
    s.status
FROM subscriptions s
JOIN customers c ON s.customer_id = c.customer_id
JOIN products p ON s.product_id = p.product_id
WHERE s.status = 'active';
```

Results | Messages

	subscription_id	first_name	last_name	product_name	start_date	end_date	billing_amount	status
1	1	John	Smith	Koodo Basic Plan	2024-01-15	NULL	35.00	Active
2	2	Sarah	Johnson	Koodo Premium Plan	2024-02-20	NULL	65.00	Active
3	3	Michael	Brown	Koodo Family Plan	2024-01-30	NULL	120.00	Active
4	4	Emily	Davis	Koodo Student Plan	2024-03-10	NULL	25.00	Active

Figure 1 is an SQL view that displays all currently active subscriptions with customer and product details, providing a clear overview of revenue-generating subscriptions.

Figure 2

Open Support Tickets View

```sql
-- Open Support Tickets View
CREATE VIEW open_support_tickets AS
SELECT
    st.ticket_id,
    c.first_name,
    c.last_name,
    c.email_address,
    st.category,
    st.priority,
    st.status,
    st.created_date,
    e.first_name AS employee_first_name,
    e.last_name AS employee_last_name
FROM support_tickets st
JOIN customers c ON st.customer_id = c.customer_id
LEFT JOIN employees e ON st.employee_id = e.employee_id
WHERE st.status IN ('open', 'in_progress', 'pending');
```

Results | Messages

	ticket_id	first_name	last_name	email_address	category	priority	status	created_date	employee_first_name	employee_l
1	8	Jennifer	Martinez	jennifer.martinez@email.com	Billing	Medium	Open	2024-06-02	Maria	Rodriguez
2	12	Michelle	Thompson	michelle.thompson@email.com	Billing	Medium	Open	2024-06-03	Sophie	Martin

Figure 2 is an SQL view that shows all unresolved support tickets with customer information and assigned employees, enabling efficient ticket management and response tracking.

Figure 3

Customer Health Score View

```sql
-- Customer Health Score View
CREATE VIEW customer_health_score AS
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    c.email_address,
    COUNT(DISTINCT s.subscription_id) AS active_subscriptions,
    COUNT(DISTINCT st.ticket_id) AS total_tickets,
    COALESCE(AVG(st.customer_satisfaction), 0) AS avg_satisfaction,
    CASE
        WHEN COUNT(DISTINCT s.subscription_id) > 2 AND
             COALESCE(AVG(st.customer_satisfaction), 5) >= 4 AND
             COUNT(DISTINCT st.ticket_id) <= 2 THEN 'Healthy'
        WHEN COUNT(DISTINCT s.subscription_id) >= 1 AND
             COALESCE(AVG(st.customer_satisfaction), 5) >= 3 THEN 'Moderate'
        ELSE 'At Risk'
    END AS health_score,
    c.created_date AS customer_since
FROM customers c
LEFT JOIN subscriptions s ON c.customer_id = s.customer_id AND s.status = 'active'
LEFT JOIN support_tickets st ON c.customer_id = st.customer_id
WHERE c.status = 'active'
GROUP BY c.customer_id, c.first_name, c.last_name, c.email_address, c.created_date;
```

Results | Messages

	customer_id	first_name	last_name	email_address	active_subscriptions	total_tickets	avg_satisfaction	health_score	customer
1	1	John	Smith	john.smith@email.com	1	2	4.0000	Moderate	2024-01
2	2	Sarah	Johnson	sarah.johnson@email.com	1	1	5.0000	Moderate	2024-02
3	3	Michael	Brown	michael.brown@email.com	1	2	4.5000	Moderate	2024-01
4	4	Emily	Davis	emily.davis@email.com	1	1	3.0000	Moderate	2024-03

Figure 3 is an SQL view that calculates a composite health score based on subscription count, support ticket volume, and satisfaction ratings, helping identify customers who may need attention or are at risk of churning.

Each view joins relevant tables to provide comprehensive information while maintaining query simplicity and performance.

Conclusion

The development of Koodo's integrated database system represents a comprehensive solution to the complex data management challenges facing modern telecommunications providers. Through systematic analysis of business operations and careful application of relational database principles, this project has successfully created a unified platform that consolidates eight critical business functions into a cohesive data ecosystem.

Key technical achievements include the establishment of a scalable eight-entity schema with properly defined relationships, implementation of robust data integrity mechanisms that prevent inconsistencies, and development of strategic SQL views that provide immediate access to critical business metrics. The database architecture successfully addresses key telecommunications industry requirements: real-time

subscription management, integrated customer support, inventory optimization, and financial transaction tracking.

The successful implementation of this database system illustrates that well-designed database serves as a strategic business asset in the telecommunications industry, enabling organizations to achieve operational excellence while positioning themselves for continued growth and innovation in an increasingly competitive market.