Database Design Document: Influencer - Marketing Analysis System

1. Introduction

1.1 Purpose

This document includes the database design for the Influencer - Marketing Analysis System. The database schema, relationships between tables, indexing strategies, and data integrity constraints will be addressed. The goal is to create a high-performance and reliable database structure that can accurately measure the financial impact of influencer collaborations.

1.2 Scope

The database will store and manage customer transactions, influencer referrals, product sales, campaign management, payment processes, and system logs. The system aims to ensure uninterrupted user experience by maintaining data integrity, security, and efficient query execution.

1.3 Target Audience

Database Administrators: To ensure sustainable and optimized system performance.
Backend Developers: To integrate database operations.
System Architects: To ensure compatibility with the overall system design.
Marketing Analysts: To review campaign data and sales analysis.
Finance and Accounting Departments: For influencer payments and budget analysis.

2. Database Architecture

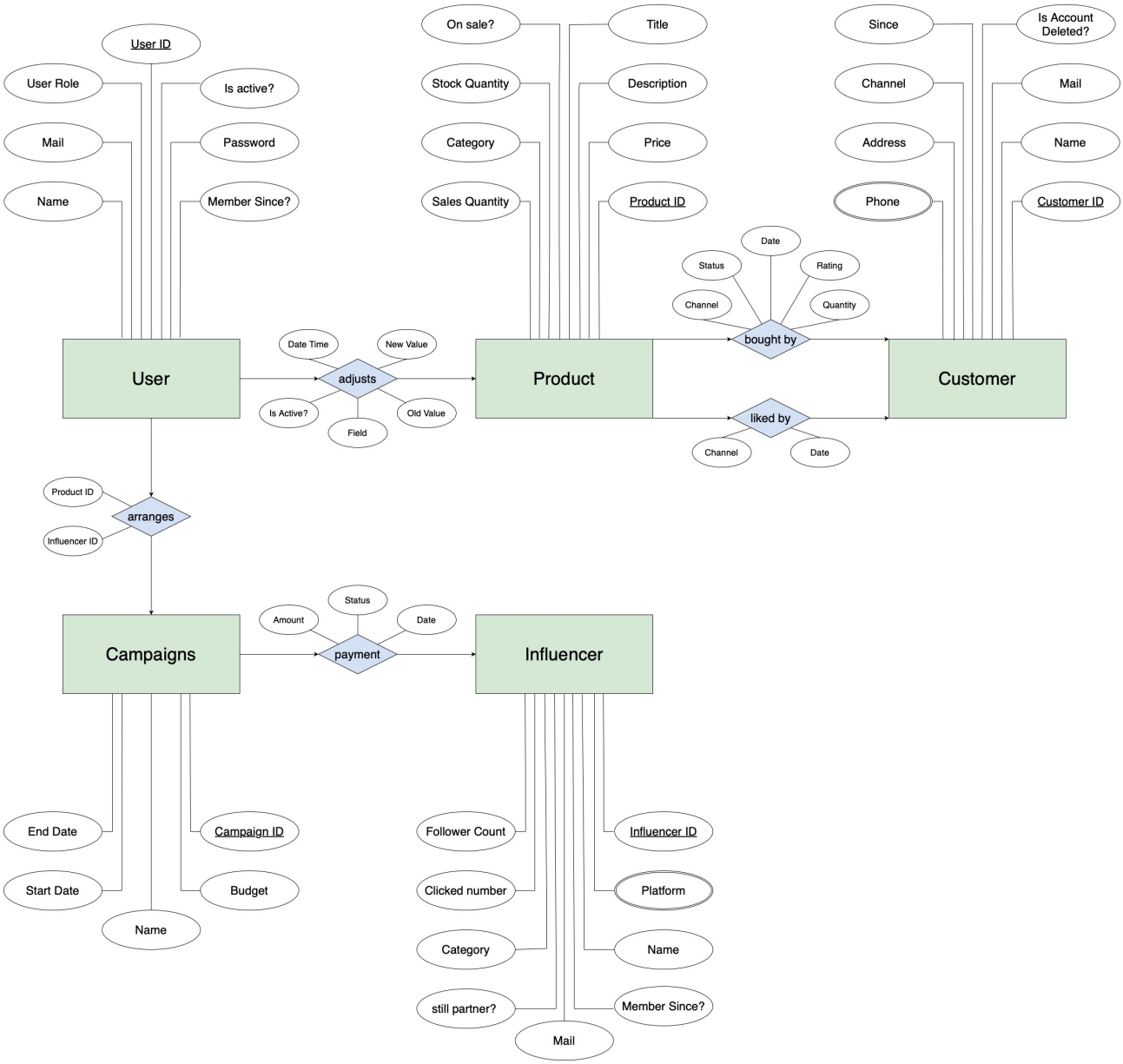
2.1 Database Management System

Type: Relational Database Management System (RDBMS)
Preferred DBMS: PostgreSQL
Normalization: Third Normal Form (3NF) will be applied. This is to optimize storage and prevent data redundancy.

2.2 Database Components

The system will include database tables with the following core components:
Customers: User information and purchase history.
Products: Information on products promoted in the system.
Influencers: Influencer profiles and performance data.
Campaigns: Influencer campaigns created by brands.
Users: Represents all defined users in the system, authorized according to their roles.

3. ER Diagram


User ID
User Role
Is active?
Mail
Password
Name
Member Since?
On sale?
Title
Stock Quantity
Description
Category
Price
Sales Quantity
Product ID
Since
Is Account Deleted?
Channel
Mail
Address
Name
Phone
Customer ID
Date
Status
Rating
Channel
Quantity
bought by
User
Date Time
New Value
adjusts
Is Active?
Old Value
Field
Product
Customer
liked by
Channel
Date
Product ID
arranges
Influencer ID
Campaigns
Status
Amount
Date
payment
Influencer
End Date
Campaign ID
Start Date
Budget
Name
Follower Count
Influencer ID
Clicked number
Platform
Category
Name
still partner?
Member Since?
Mail

4. Database Schemas

Customers Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
customer_id	SERIAL (PK)	NOT NULL	Customer Unique ID
customer_name	VARCHAR(50)	NOT NULL	Customer Name
customer_mail	VARCHAR(255)	NOT NULL, UNIQUE	Customer e-mail Address
customer_phone	VARCHAR(15)	NOT NULL,UNIQUE	Customer Phone Number
customer_address	TEXT	NOT NULL	Customer Address
channel	INTEGER	DEFAULT -1	Channel the Customer Came From
since	DATE	NOT NULL, CHECK (<= CURRENT_DATE)	Channel Registration Date
is_account_deleted	BOOLEAN	NOT NULL	Is Account Deleted?

Products Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
product_id	SERIAL (PK)	NOT NULL	Product Unique ID
stock_quantity	INTEGER	NOT NULL, CHECK (>= 0)	Product Stock Quantity
category	VARCHAR(100)	NOT NULL	Product Category
sales_quantity	INTEGER	NOT NULL, CHECK (>= 0)	Number of Sales
description	TEXT	-	Product Description
product_price	DOUBLE PRECISION	NOT NULL, CHECK (>= 0)	Product Prices
on_sale	BOOLEAN	NOT NULL	Whether the Product is on Sale or Not
title	VARCHAR(100)	NOT NULL	Name of the Product Ad

Influencers Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
influencer_id	SERIAL (PK)	NOT NULL	Influencer's Unique ID
influencer_name	VARCHAR(50)	NOT NULL	Influencer Name
platform	VARCHAR(50)	NOT NULL	Platform Active On
follower_count	INTEGER	NOT NULL, CHECK (>= 0)	Number of Followers
clicked_number	INTEGER	NOT NULL, CHECK (>= 0)	Number of Clicks
category	VARCHAR(100)	NOT NULL	Influencer's Category
influencer_mail	VARCHAR(255)	NOT NULL, UNIQUE	Influencer e-mail Address
still_partner	BOOLEAN	NOT NULL	Still a Partner?
member_since	DATE	NOT NULL, CHECK (<= CURRENT_DATE)	Influencer Participation Date

Users Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
user_id	SERIAL (PK)	NOT NULL	User ID
user_role	user_role_enum	NOT NULL	User Role (enum)
user_mail	VARCHAR(255)	NOT NULL, UNIQUE	User e-mail Address
user_name	VARCHAR(50)	NOT NULL	Username
user_password	VARCHAR(100)	NOT NULL	Password
is_active	BOOLEAN	NOT NULL	Is User Still Active or Not
member_since	DATE	NOT NULL, CHECK (<= CURRENT_DATE)	Membership Start Date

Campaigns Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
campaign_id	SERIAL (PK)	NOT NULL	Campaign ID
campaign_name	VARCHAR(50)	NOT NULL	Campaign Name
start_date	DATE	NOT NULL, CHECK (<= end_date)	Campaign Start Date
end_date	DATE	NOT NULL	Campaign End Date
budget	NUMERIC(10,2)	NOT NULL, CHECK (>= 0)	Campaign Budget

Liked By Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
customer_id	INTEGER	NOT NULL, FK (customers) ON DELETE CASCADE	Customer ID
product_id	INTEGER	NOT NULL, FK (products) ON DELETE CASCADE	Product ID
channel	INTEGER	DEFAULT -1	Channel Liked On
liked_date	DATE	CHECK (<= CURRENT_DATE)	Date Liked

Arranges Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
product_id	INTEGER	NOT NULL, FK (products) ON DELETE CASCADE	Product ID
influencer_id	INTEGER	NOT NULL, FK (influencers) ON DELETE CASCADE	Influencer ID
user_id	INTEGER	NOT NULL, FK (users) ON DELETE CASCADE	User ID
campaign_id	INTEGER	NOT NULL, FK (campaigns) ON DELETE CASCADE	Campaign ID

Adjusts Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
user_id	INTEGER	NOT NULL, FK (users) ON DELETE CASCADE	User ID
product_id	INTEGER	NOT NULL, FK (products) ON DELETE CASCADE	Product ID
date_time	TIMESTAMP	NOT NULL, CHECK (<= CURRENT_TIMESTAMP)	Edit Date
is_active	BOOLEAN	NOT NULL	Is Active?
field	adjust_field_enum	NOT NULL	Modified Field
old_value	DOUBLE PRECISION	NOT NULL, CHECK (>= 0)	Previous Value
new_value	DOUBLE PRECISION	NOT NULL, CHECK (>= 0)	New Value

Bought By Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
product_id	INTEGER	NOT NULL, FK (products) ON DELETE CASCADE	Product ID
customer_id	INTEGER	NOT NULL, FK (customers) ON DELETE CASCADE	Customer ID
status	bought_status_enum	NOT NULL	Purchase Status (enum)
channel	INTEGER	DEFAULT -1	Sales Channel
rating	INTEGER	CHECK (1 <= rating <= 5)	Product Rating
quantity	INTEGER	CHECK (quantity > 0)	Quantity Purchased
bought_date	DATE	CHECK (<= CURRENT_DATE)	Purchase Date

Payment Table

COLUMN	TYPE	CONSTRAINTS	DESCRIPTION
campaign_id	INTEGER	NOT NULL, FK (campaigns) ON DELETE CASCADE	Campaign ID
influencer_id	INTEGER	NOT NULL, FK (influencers) ON DELETE CASCADE	Influencer ID
amount	NUMERIC(10,2)	NOT NULL, CHECK (>= 0)	Payment Amount
status	enum	NOT NULL	Payment Status
date	DATE	NOT NULL	Payment Date

5. Indexing

Primary Keys:

All table ID fields are designated as primary keys (PK). Since our project involves frequent querying and reporting, this ensures uniqueness and fast access. (Except for the join tables, where a composite primary key is created using other IDs.)

Foreign Keys:

Primary keys composed of unique IDs are used as foreign keys to establish relationships between tables. This allows, for example, automatic updating or deletion of related sales records when a customer is deleted.

Indexing:

In general, unique IDs are indexed.
• **Campaigns table: I**ndexing on ‘Start Date’ and ‘End Date’ for effective date-range queries.
• **Influencers table:** Indexing on ‘Follower Count’ for quicker querying of influencers by audience size.
• **Products table:** Indexing on ‘Stock Quantity’ for efficient stock checks and updates.
• **Payments table:** Indexing on ‘Influencer ID’ and ‘Campaign ID’ to speed up campaign-based payment reports.
• **Users table:** Indexing on ‘email’ for faster login verifications as user numbers grow.
• **Campaigns table:** Indexing on ‘Budget’ to speed up budget-range analysis queries.

6. Security Measures

Data Encryption:

• **Passwords:** User passwords will be stored using bcrypt hashing.
• **Sensitive Data:** Customer and payment data will be protected using AES-256 encryption algorithm.

Access Control:

Role-Based Access Control (RBAC) will be implemented.
• **Admin:** Creates/deletes users, assigns roles, manages system settings.
• **Analyst:** Accesses all statistical data, views/downloads campaign and sales analysis reports.
• **Campaign Manager:** Creates campaigns, adds products/influencers, sets campaign dates and budgets.
• **Product Manager:** Adds, edits, deletes products; manages stock, descriptions, and prices.
• **Influencer Manager:** Adds/removes influencers, updates influencer details and commissions.

• **Finance Manager:** Accesses payment data, performs payment calculations, reviews ROI reports.

7. Scalability & Performance Optimization

• Load balancing via database replication
• Frequently accessed data cached using Redis/Memcached
• Sales and payment records partitioned by time
• Connection pooling using tools like PgBouncer

8. Backup & Disaster Recovery Plan

• Daily full and hourly incremental backups
• Product and campaign files versioned and backed up
• Auto-failover and multi-region deployment applied
• RTO: 15 minutes, RPO: 5 minutes for data recovery

9. Conclusion

This database design document provides a strong foundation in terms of security, scalability, and performance for the Influencer & Marketing System.