

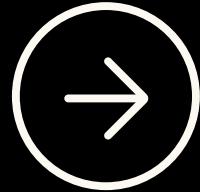


İrem | Burak | İlkay | Özgür

# INFLUENCER-MARKETING ANALYSIS SYSTEM

To enable e-commerce companies to analyze the impact of influencer collaborations on sales and profitability.

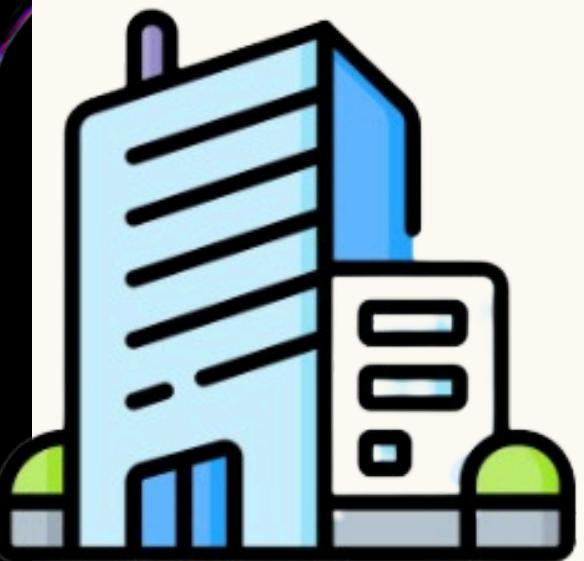
# TARGET AUDIENCE



- 1 E-Commerce Companies
- 2 Marketing and Advertising Teams
- 3 Data Analysts
- 4 Business Development Managers

# SCENARIO

Brand X



Brand X is an e-commerce business and already has its own e-commerce platform.





Brand X wants to collaborate  
with influencers to promote its  
new season products!

Advertise!

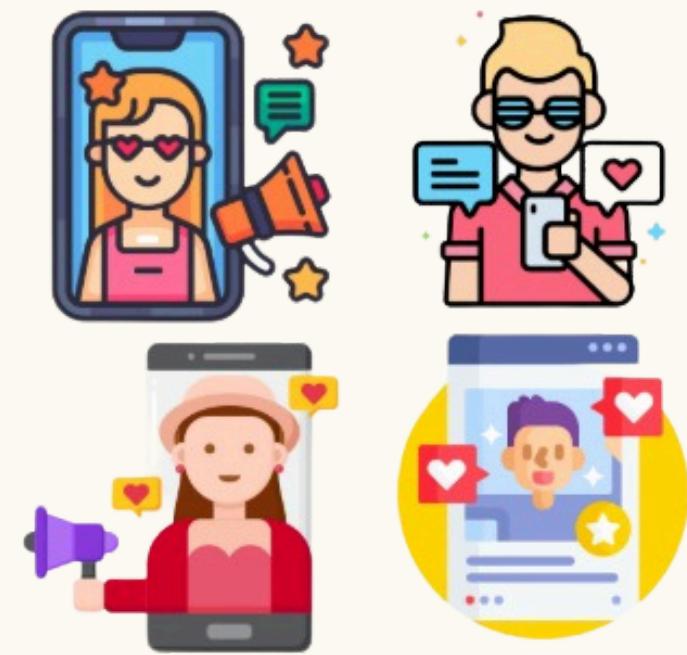


## Customers



Customers click on  
referral links of  
influencers

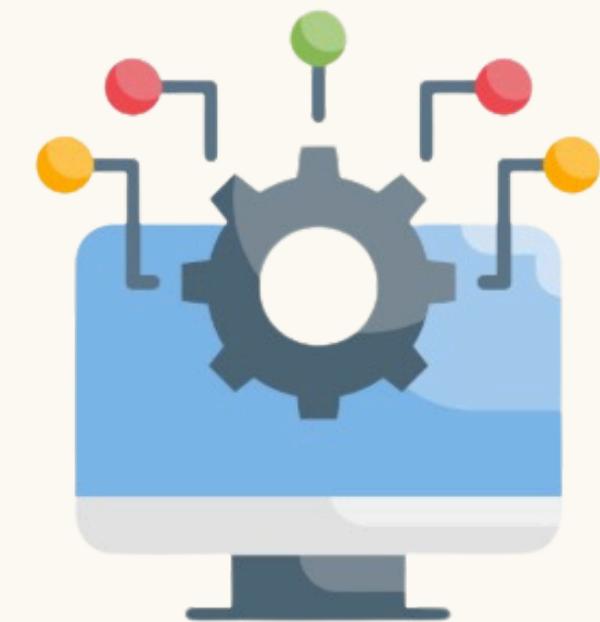
## Influencers



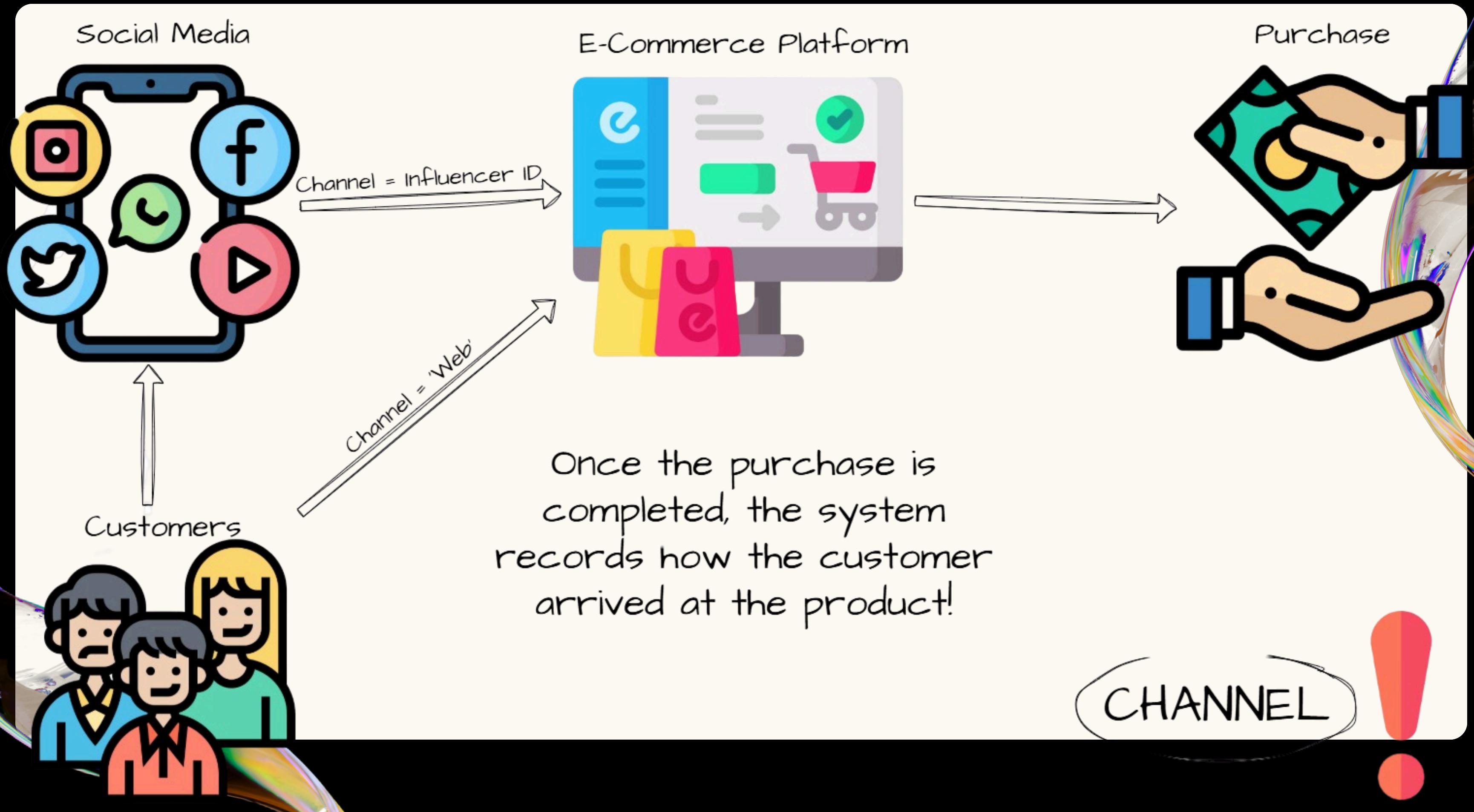
## Brand X's E-Commerce Platform



They go and buy or  
like some products



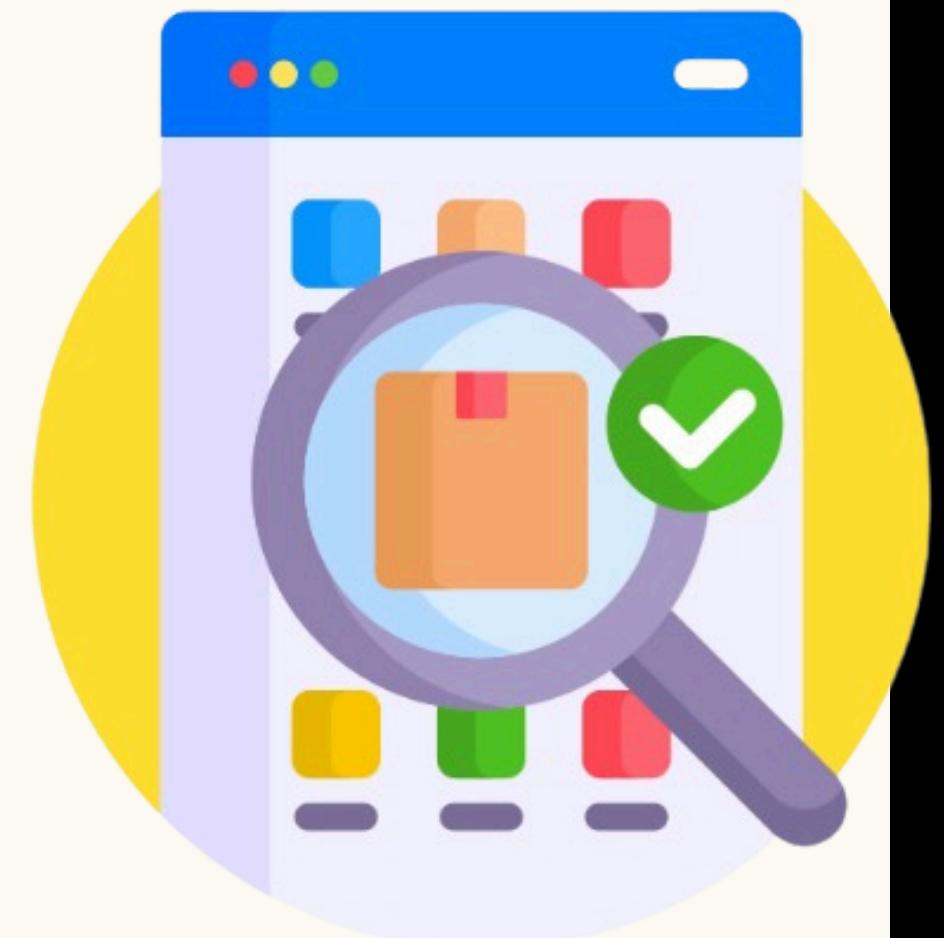
The system records all these  
actions to allow analysis of  
influencer performance!

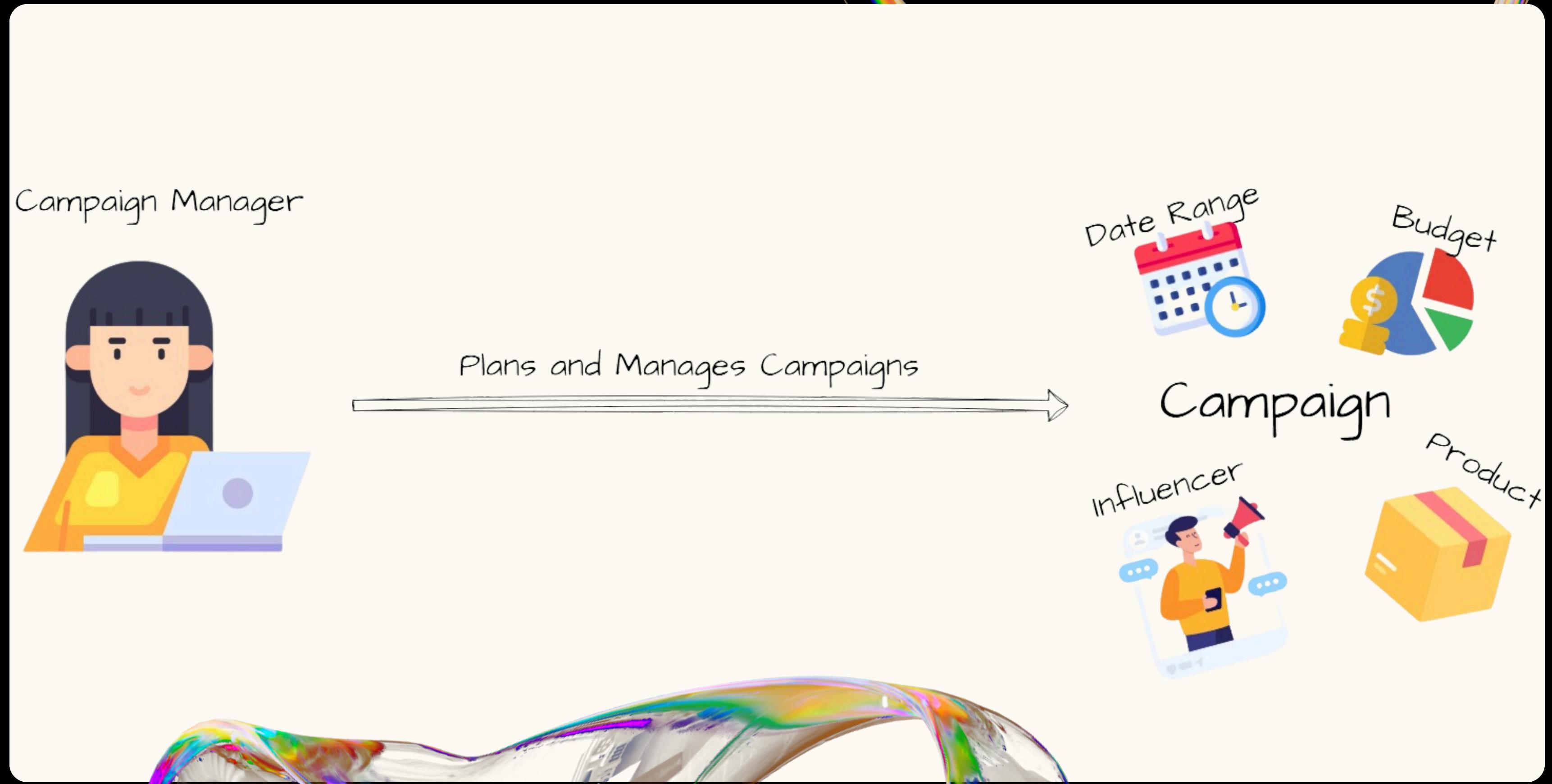


## Product Manager



Ensures products are entered  
into the system with correct  
information

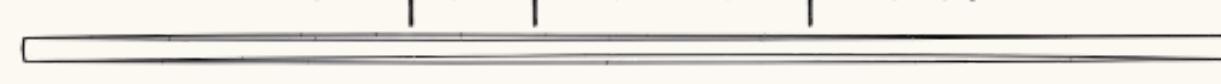




Analyst



Uses this collected data  
to prepare reports



Conversion Rates



Influencer Effectiveness



Sales Changes



Influencer ROI



Influencer Manager



Handles the influencer portfolio

Recruits  
influencers



Adjusts  
Commissions



Terminating  
Contracts





## At the end of the campaign

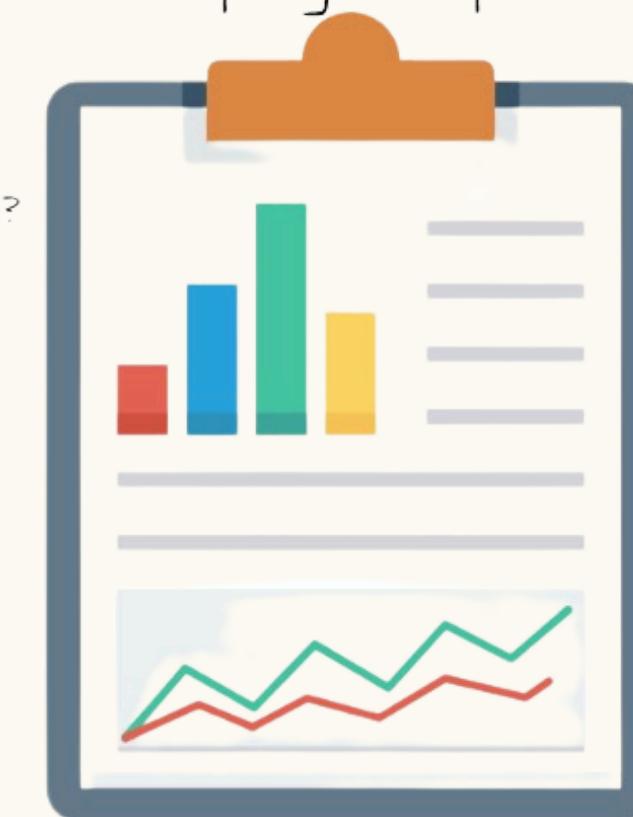
The System



Which products attracted the most attention?  
Which products bring most profit?  
How many sales each influencer made?  
How well the budget was used?

Generates detailed reports

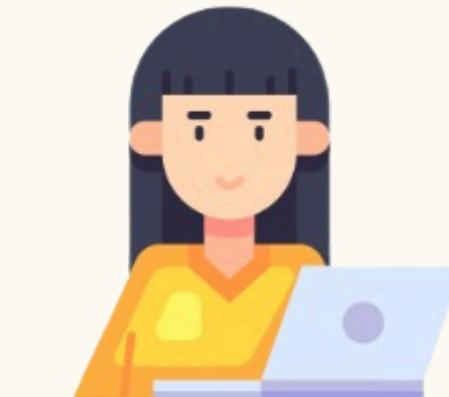
Campaign Report



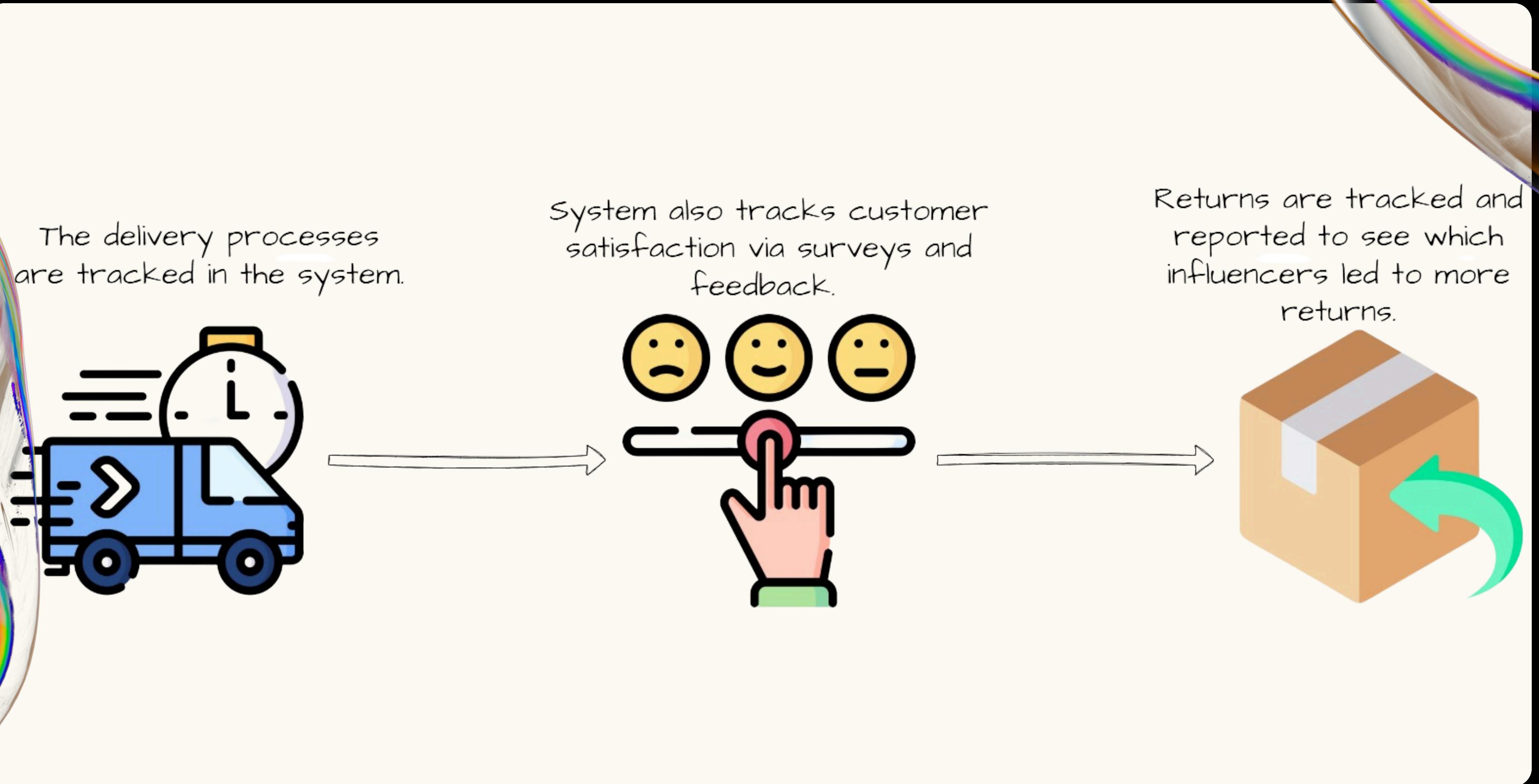
Analyst



Campaign Manager



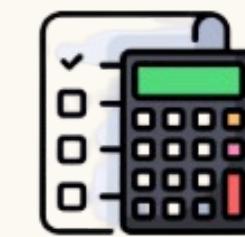
Analyst and the Campaign Manager evaluate these reports together.





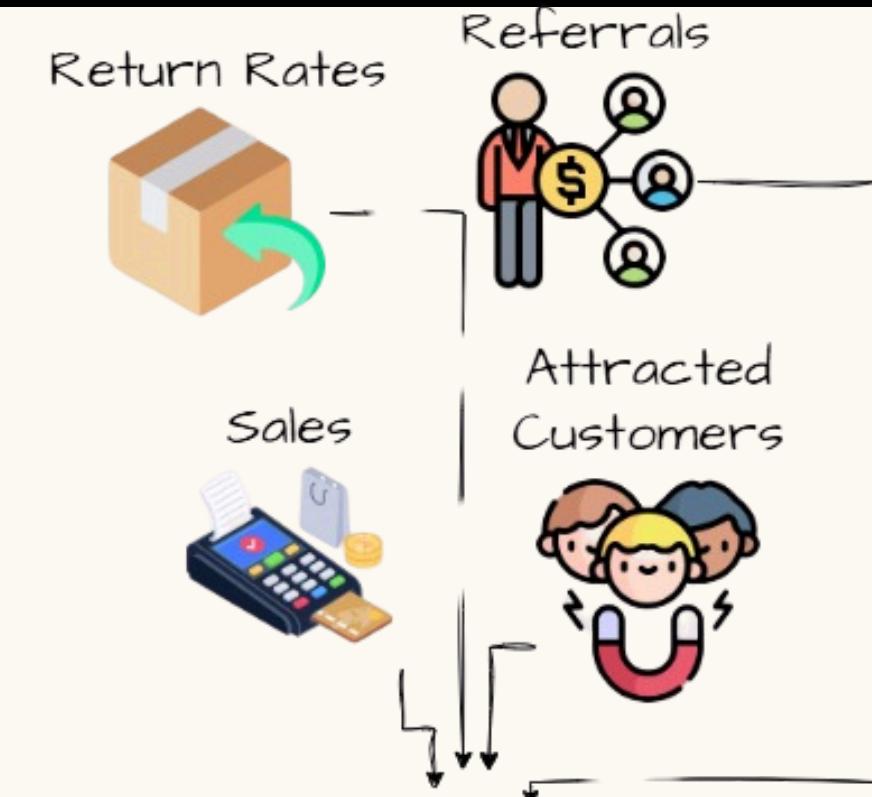
Finance Manager

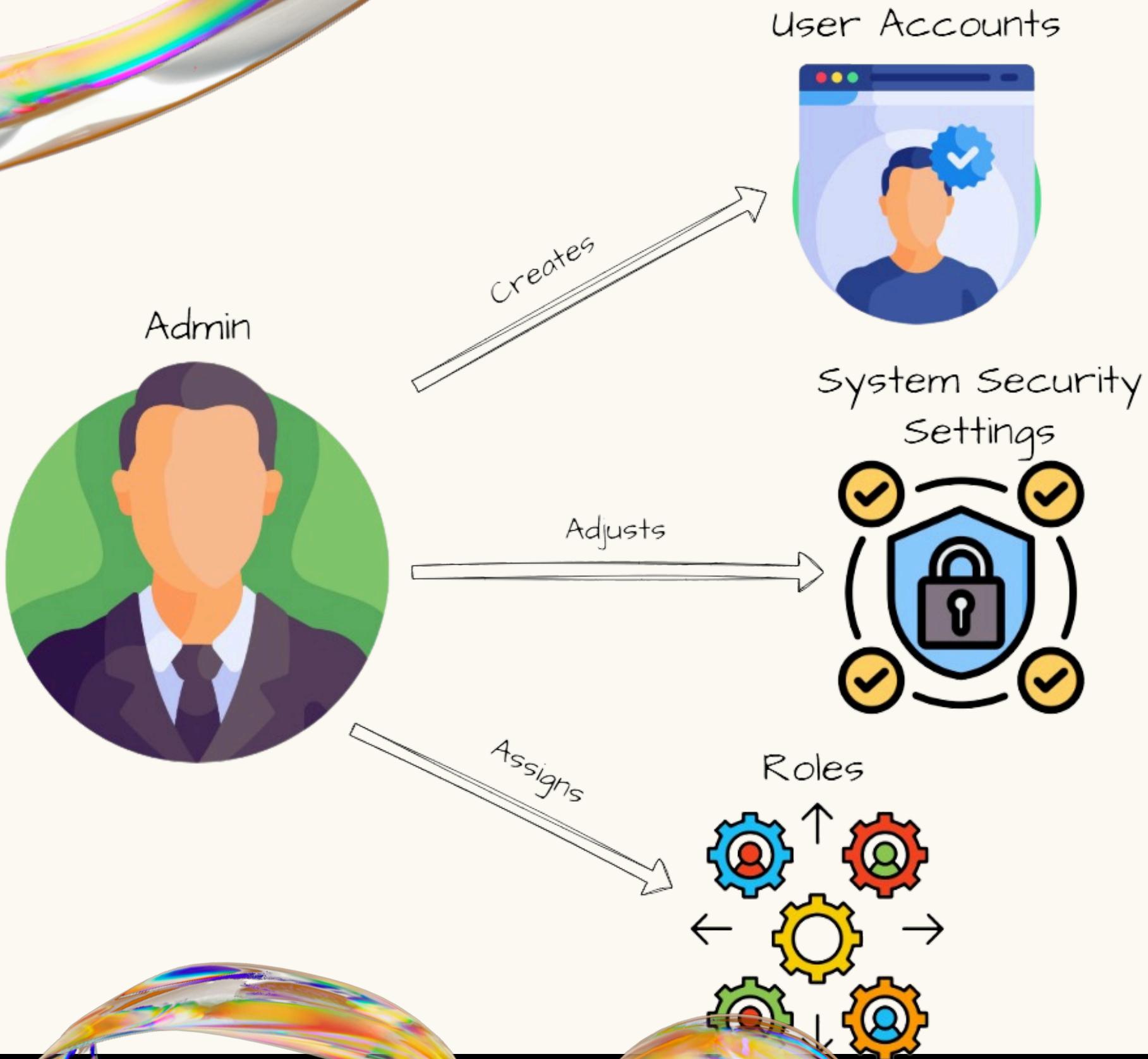
Manages influencer payments  
and profitability



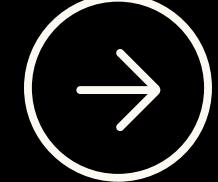
Calculating  
Profitability

Confirming  
Influencer  
Payments





All coordination and access control are managed by the Admin role.



# USERS



## Analyst

Access all statistical data, download performance metrics and reports.

## Product Manager

Add/delete products, edit product info, price, and stock.

## Campaign Manager

Create campaigns, manage influencers and products, edit budget and schedule.

## Influencer Manager

Add/delete influencers, adjust commissions.

## Finance Manager

Access all financial and payment data, make and manage payments.

## Admin

Full system control: adjust settings, user roles, and access all data.

# REQUIREMENTS AND CONSTRAINTS

Our system includes 35 functional requirements, 9 non-functional requirements, and 18 constraints.



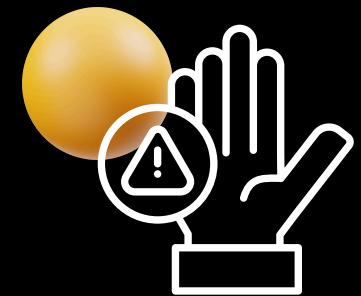
## Functional Requirements

They are the requirements that define what a DBMS should do. In other words, they describe the system's functions and the services provided to users.



## Non- Functional Requirements

They define how a DBMS should operate. In other words, they are concerned with quality attributes such as performance, security, and scalability.



## Constraints

They are mandatory limitations that must be followed during the development or operation of a DBMS. They are usually related to technological, legal, organizational, or cost factors.

# FUNCTIONAL REQUIREMENTS



## User & Access Management

UAFR-01: Upon request, the admin creates new user accounts within the company.

UAFR-02: Admin must be able to assign roles to users.

UAFR-03: Admin must be able to configure all user and system settings and access all data.

## Product Management

PFR-01: Product managers can access data about products and sales.

PFR-02: Product Manager must be able to add new products to the system.

PFR-03: Product Manager must be able to edit product information, price, and stock details in the system.

## Campaign Management

CFR-01: Campaign Manager must be able to create new campaigns and edit the campaigns.

CFR-02: The Campaign Manager must be able to stop active campaigns.

CFR-03: Campaign Manager must be able to define campaign dates and budgets.

CFR-04: Campaign managers must be able to access data and reports related to past campaigns and influencer performance.

CFR-05: Campaign managers must be able to edit the products included in the campaign.

CFR-06: Campaign managers must be able to edit the influencers involved in the campaign.

CFR-07: The system must include product, date, and budget details in the campaign record for each campaign.



# FUNCTIONAL REQUIREMENTS

## Influencer Management

- IFR-01: Influencer managers can access data and reports related to influencer performance.
- IFR-02: Influencer Manager must be able to add new influencers to the system and delete existing ones.

## Influencer Performance Tracking

- IPFR-01: The system must be able to track the number of customers referred by each influencer.
- IPFR-02: The system must analyze whether referred customers completed a purchase.
- IPFR-03: The system must record through which influencer each customer was referred.
- IPFR-04: The system must identify which influencer link each customer used to reach the site.
- IPFR-05: The system must report the number of returns per influencer caused by their referrals.

## Customer Feedback & Experience

- CFEFR-01: The system must be able to collect user feedback and satisfaction surveys.
- CFEFR-02: The system must record product return data and perform influencer-based analysis.
- CFEFR-03: The system must analyze user feedback by matching it with products and influencers.

## System Logging & Monitoring

- SLMFR-01: The system must log user activities within the system.



# FUNCTIONAL REQUIREMENTS

## Analytics & Reporting

- ARFR-01: Users with the Analyst role must be able to analyze all data and download reports.
- ARFR-02: The system must generate detailed reports using all data at the end of a campaign.
- ARFR-03: The system must be able to analyze the average spending amount of customers referred by influencers.
- ARFR-04: The system must analyze the sales performance of products included in campaigns.
- ARFR-05: The system must generate graphical analysis outputs based on SQL queries.

## Payment & Finance Management

- PFFR-01: Finance Manager must be able to perform payment transactions through the system.
- PFFR-02: Finance Manager must be able to view payment reports through the system.
- PFFR-03: The system must calculate payments by matching sales, returns, and referral data per influencer.
- PFFR-04: Finance Manager must be able to define payment dates through the system.
- PFFR-05: Finance Manager must be able to define and edit influencer budgets.
- PFFR-06: After payments are completed, the system must report the total spending and ROI ratio.



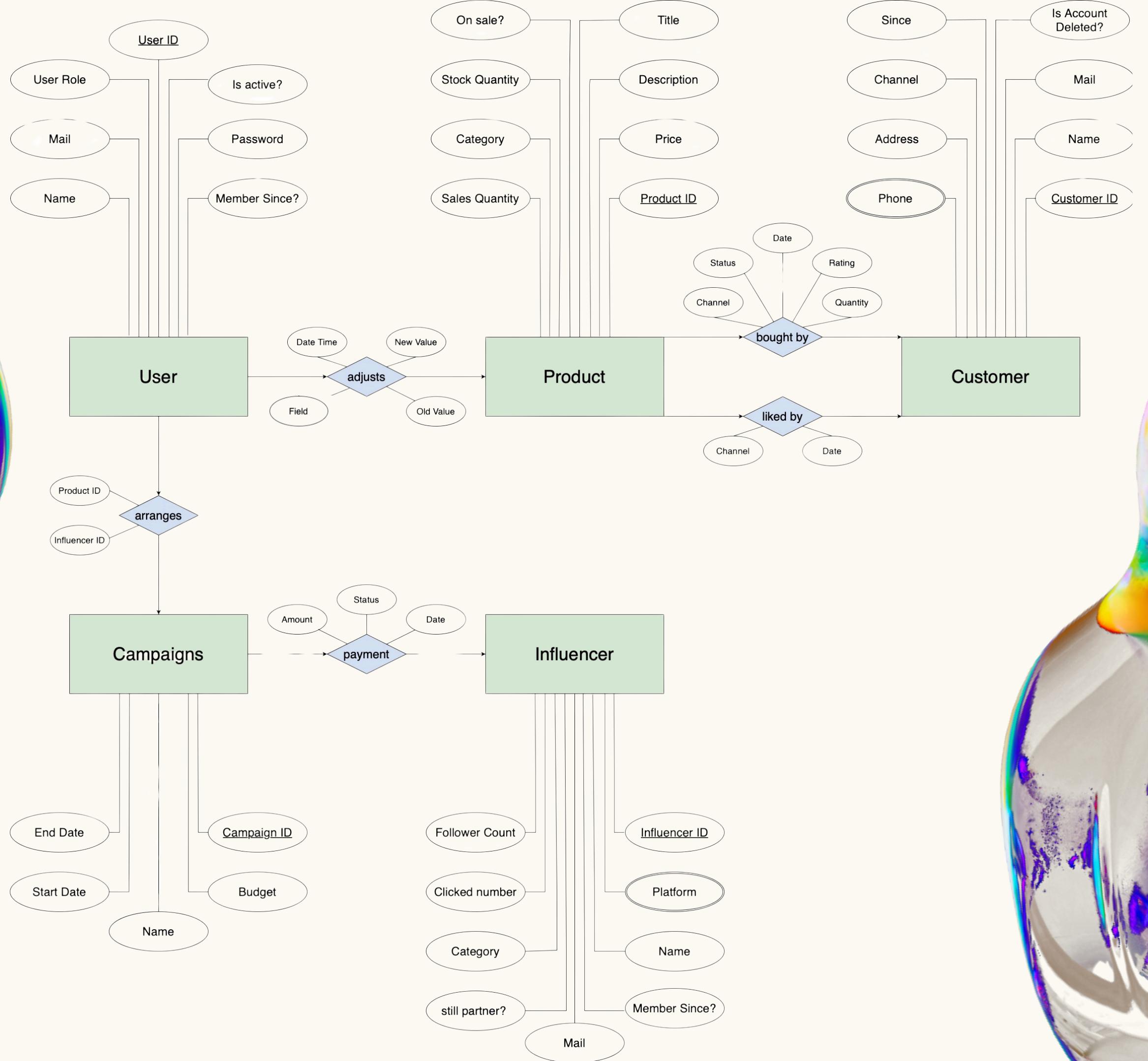
# NON-FUNCTIONAL REQUIREMENTS

- NFR-01: The system must support at least 200 concurrent users.
- NFR-02: All queries and reports must complete in under 7 seconds.
- NFR-03: User data must be stored with AES-256 encryption.
- NFR-04: The application must be fully compatible with both mobile and desktop browsers.
- NFR-05: Weekly full and daily incremental database backups must be implemented.
- NFR-06: The UI must comply with WCAG 2.1 accessibility standards.
- NFR-07: A caching mechanism (e.g., Redis or Memcached) must be used for frequent queries.
- NFR-08: There must be logging and error tracking systems to detect and resolve issues.
- NFR-09: The system must run over HTTPS and support secure authentication methods (OAuth2, JWT).



# CONSTRAINTS

- C-01: Analysts may only observe data, not update it.
- C-02: Product managers cannot access data beyond products and sales.
- C-03: Campaign managers can manage only up to 10 campaigns within 2 weeks and stay within defined budgets.
- C-04: Campaign managers can edit only the campaigns they created.
- C-05: Influencer managers can edit only the influencers they added.
- C-06: Only Admin can change system settings and user roles.
- C-07: Influencer commissions can only be adjusted within a predefined range.
- C-08: Product managers can add products only under existing categories.
- C-09: Campaign managers must choose influencers from the system pool.
- C-10: Finance managers can update sale statuses but not delete sales data.
- C-11: Finance managers cannot process payments that exceed the campaign budget.
- C-12: Users can only act within defined roles.
- C-13: Currency is limited to Turkish Lira (₺).
- C-14: Campaign data cannot be edited retroactively.
- C-15: Each user can access only authorized data.
- C-16: External payment systems are not integrated.
- C-17: Only Admins can change system settings.
- C-18: Data analysis must use internal database only.



**ER DIAGRAM**



# TABLES

## CUSTOMERS TABLE

### Purpose:

Stores all registered customer information including personal data, the channel through which they joined, their registration date, and whether the account is active or deleted.

### Why it's important:

Enables tracking of customers referred by influencers.

Supports sales, feedback, and engagement records.

Crucial for analyzing how influencer campaigns impact customer behavior.

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?

# TABLES

## PRODUCTS TABLE

### Purpose:

Holds detailed information about each product such as stock quantity, category, sales numbers, description, price, and sale status.

### Why it's important:

Products are central to campaign planning and customer interactions.

Supports analysis of popular or best-performing products.

Enables inventory control and product-level analytics.

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

# TABLES

## INFLUENCERS TABLE

### Purpose:

Stores influencer details including platform, follower count, number of clicks, category, partnership status, and more.

### Why it's important:

Essential for calculating ROI and campaign contribution.

Helps identify high-performing influencers.

Links to payment and campaign tables for performance-based decisions.

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

# TABLES

## USERS TABLE

### Purpose:

Represents internal system users such as Admins, Campaign Managers, Product Managers, Analysts, etc. Includes role, login credentials, and activity status.

### Why it's important:

Drives the Role-Based Access Control (RBAC) system.

Controls what each user can do.

Tracks user changes and system usage for auditing.

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

# TABLES

## CAMPAIGNS TABLE

### Purpose:

Captures metadata for influencer marketing campaigns including campaign name, start and end dates, and budget.

### Why it's important:

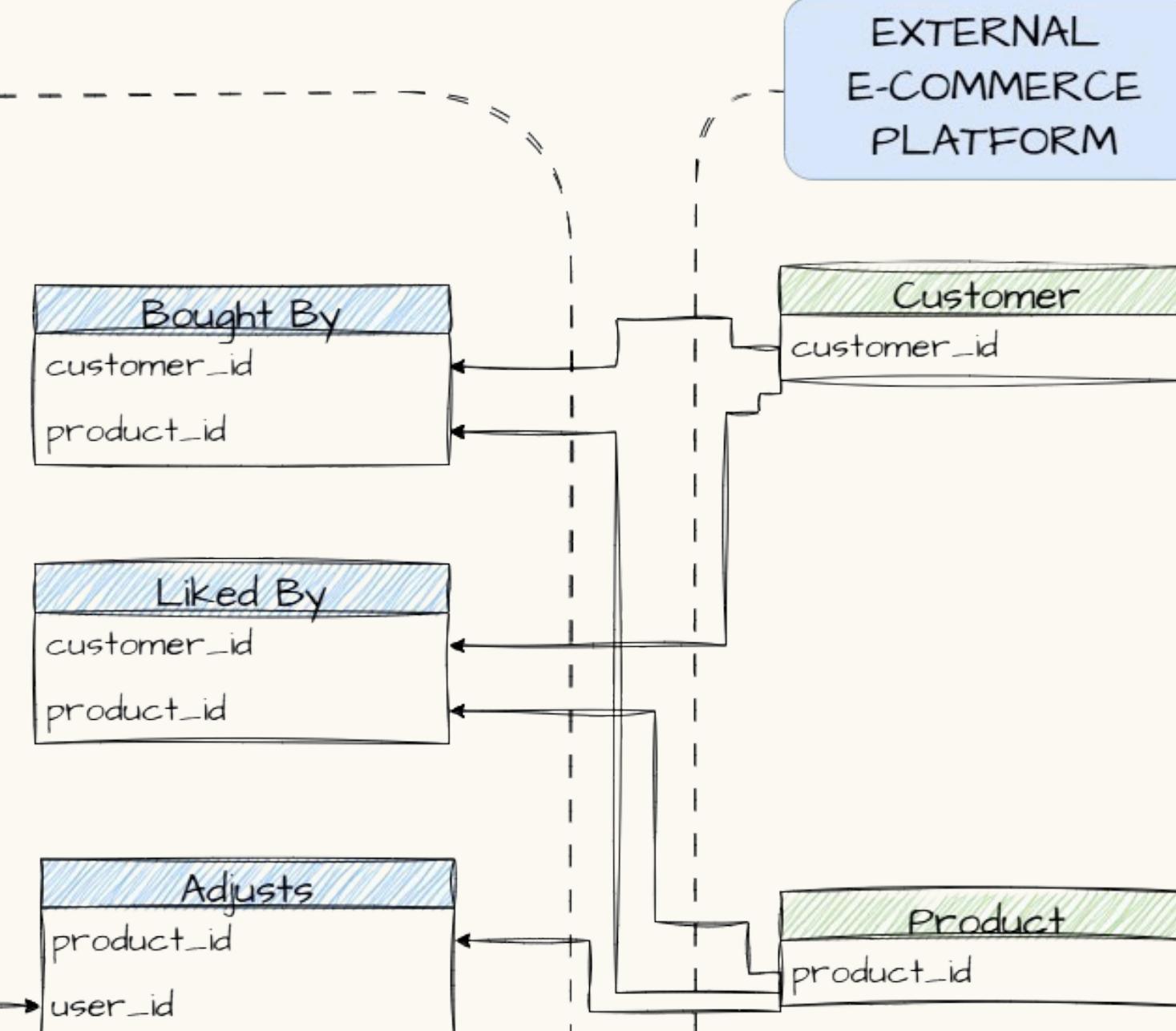
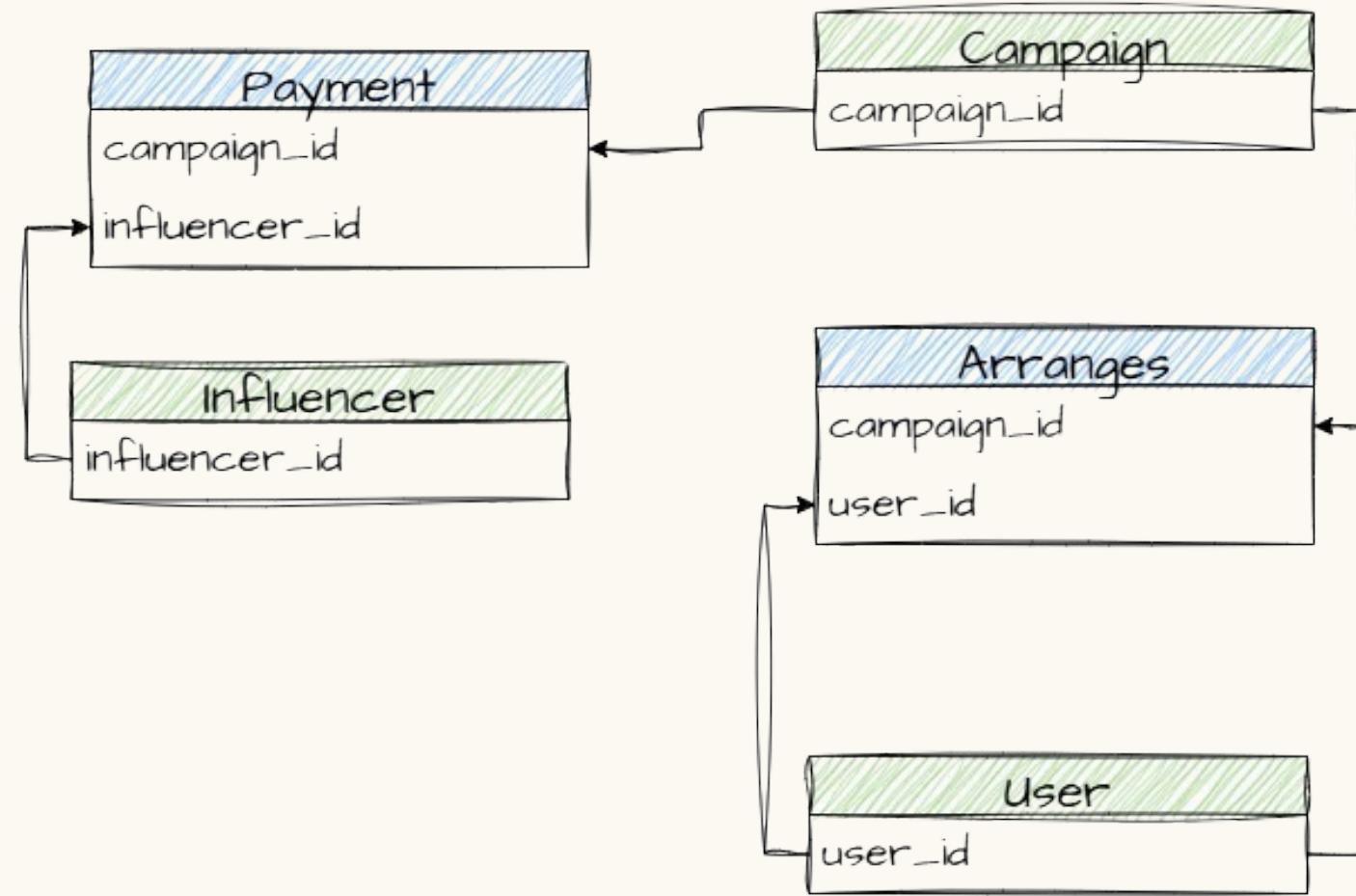
Ties together products, influencers, and performance metrics.

Forms the basis for campaign reporting and analysis.

Tracks campaign effectiveness and financial planning.

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

## DEEP PURPLE ANALYSIS SYSTEM



# TABLES

## ARRANGES TABLE

### Purpose:

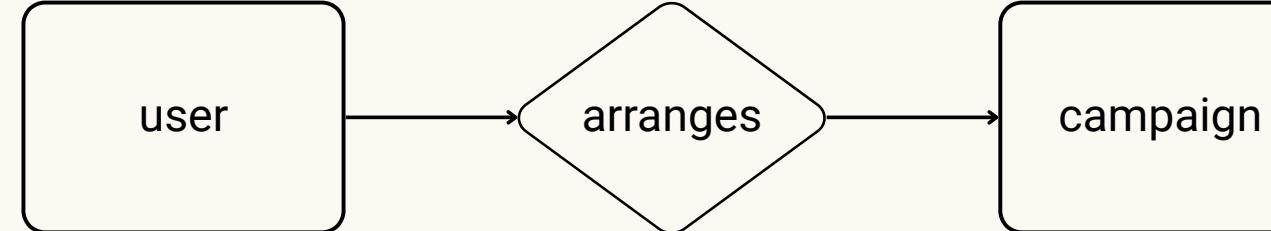
Connects products, influencers, users, and campaigns. Shows which product was promoted by which influencer under which campaign and by whom it was arranged.

### Why it's important:

Defines the composition of each campaign.

Critical for determining product-influencer combinations.

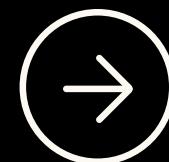
Links key entities for end-to-end campaign management.



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

# TABLES

## ADJUSTS TABLE



### Purpose:

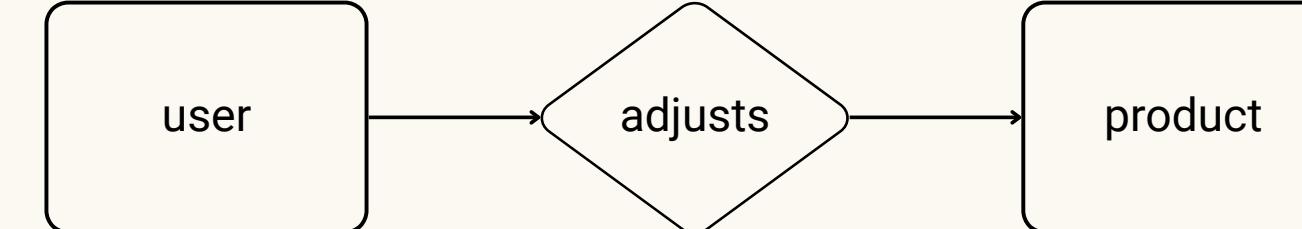
Tracks historical changes made to product data, including who made the change, when, what field was changed, and the old and new values.

### Why it's important:

Provides an audit trail for accountability.

Supports rollback and verification in case of data errors.

Essential for maintaining data integrity.



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

# TABLES

## BOUGHT BY TABLE

### Purpose:

Stores purchase history, including which customer bought which product, when, in what quantity, through which channel, and with what satisfaction rating.

### Why it's important:

Enables sales and conversion tracking.  
Links product performance to customer behavior.  
Supports influencer-based revenue analysis.



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

# TABLES

## LIKED BY TABLE

### Purpose:

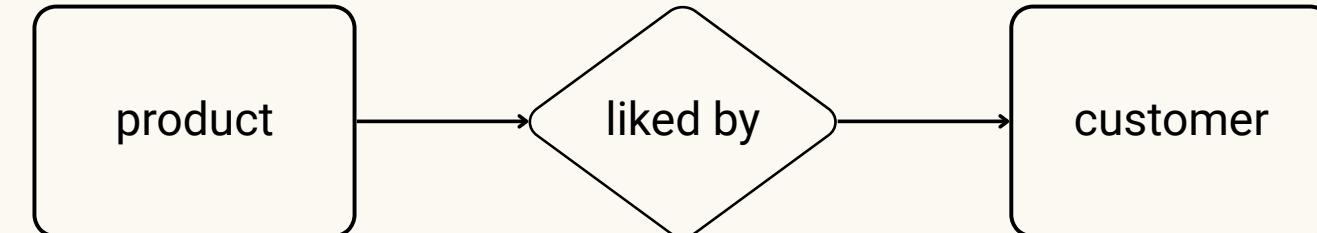
Stores product likes, showing which customer liked which product, via which channel, and when.

### Why it's important:

Measures product interest and popularity.

Assists in matching user intent with campaign success.

Helps calculate like-to-purchase conversion rates.



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

# TABLES

## PAYMENT TABLE

### Purpose:

Manages influencer payments by campaign, including amount, status, and date.

### Why it's important:

Supports performance-based payment logic.

Allows financial planning and ROI measurement.

Tracks how much was spent on each influencer.



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

# SYSTEM COMPONENTS

## User Management Module

User registration, secure login, and authentication. Role-based access: Admin, Marketing Manager, Data Analyst, etc. Password recovery, account deletion, and Two-Factor Authentication (2FA).

## Influencer Analysis Module

Automatic import of campaign data (spending, engagement, conversion rates). Performance analysis of CTR, ROI, and revenue contribution. Ranking algorithm to identify the most effective influencers.

## Campaign Management Module

Creating, editing, and managing campaigns. Measuring conversion rates, ROI, and customer engagement. Target audience analysis and detailed performance graphs.

## Finance and Payment Management Module

Tracking and calculating influencer payments. Campaign budget planning and expense reporting. Identifying the most profitable influencer collaborations.

## Product and Sales Analysis Module

Tracking return rates and sales performance. Influencer-based return analysis. Generating category-based sales reports.

## Reporting and Visualization Module

Detailed reports with filters by time range, influencer, and campaign. Graphical visualizations: bar charts, pie charts, line graphs. Report downloads in Excel, PDF, and CSV formats.

## We followed the Agile methodology with weekly sprints and used Trello to maintain an organized workflow.

The team gathered and agreed on the project idea: Influencer Marketing Analysis System.

In the first meeting, the project idea was presented to the advisor and the general structure was approved.

The initial ER diagram was created, modeling entities and their relationships in the system.

The ER diagram was revised, clarifying relationships, keys, and table structures.

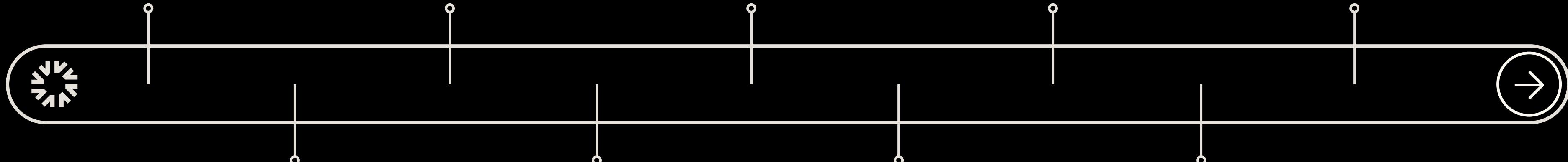
All documents were completed, ensuring the project structure was technically and conceptually integrated.

A system scenario (case) was created, and PostgreSQL was chosen as the database.

Preparation of the requirements, system design, and database documents was initiated.

In the second meeting, the ER model was presented and conceptual mistakes in the structure were identified.

Sample test data was generated using Python to simulate the dataset for the system.



# WHY DID WE CHOOSE POSTGRESQL ?

- 1** Ability to Manage Large and Complex Data
- 2** Advanced Querying and Analysis Capabilities
- 3** Data Integrity
- 4** Open-Source and Scalable
- 5** Community Support and Rich Toolset





İrem | Burak | İlkay | Özgür

# THANK YOU

for your time and attention

