

**BUAN 6320 – Database Foundations for Business Analytics****PROJECT – TECHNICAL REPORT****BUAN6320**

**Title: Influencer Partnership Management System**

**GROUP – 11**

**Zoe Christopherson  
Nikkitha Doppalapudi  
Juhee Makwana  
Chaitanya Nimmagadda  
Sri Sai Praneetha Ravipati**

**JSOM – UTD**

# "Influencer Partnership Management System"

## Introduction

This Database Design Document provides a comprehensive overview of the design and implementation of a specialized system developed to streamline and optimize the management and analysis of influencer partnerships. The system is designed to handle and process detailed information related to influencers, their content creation activities, the products they promote, contractual agreements, associated brands, and performance metrics derived from their campaigns. By centralizing and organizing this data, the database facilitates efficient tracking, reporting, and analysis, empowering businesses to evaluate the success of influencer collaborations and make data-driven decisions.

## Overview

The Influencer Partnership Management System is a centralized database solution designed to facilitate the efficient management and analysis of influencer collaborations for businesses. This system addresses the complexities of modern influencer marketing by organizing and storing critical data related to influencers, their content, promoted products, contractual agreements, associated brands, and performance metrics.

The system ensures seamless tracking and evaluation of influencer activities, providing businesses with insights into the effectiveness of their marketing efforts. It captures and analyses key performance indicators such as engagement metrics, revenue generated from promotional campaigns, and overall contract outcomes. With detailed data on content creation and campaign performance, businesses can identify high-performing influencers, measure ROI, and refine their marketing strategies.

The system is built with scalability in mind, allowing businesses to manage multiple influencers, campaigns, and products concurrently. It also ensures data integrity and security, as the information stored is considered proprietary and critical for maintaining a competitive edge in influencer marketing. This robust solution empowers businesses to make informed, data-driven decisions while fostering successful influencer collaborations.

## Literature Review

The growing influence of social media and digital platforms has transformed the marketing landscape, making influencer partnerships a cornerstone of modern marketing strategies. The **Influencer Partnership Management System** addresses the increasing need for businesses to streamline and optimize their collaboration with influencers by leveraging structured data management and analytics.

In the research paper, "The Rise of Influencers as Opinion Leaders", Freberg et al. (2011) explored the role of influencers as digital opinion leaders in shaping consumer behavior. Their research highlighted that influencers establish trust and credibility among their followers, leading to higher engagement and purchase intent compared to traditional advertisements. This study emphasized the importance of selecting influencers who align with brand values and the need for tools to evaluate their reach and authenticity. The findings underscore the necessity of

## PROJECT TECHNICAL REPORT

systems that streamline the selection and performance tracking of influencers, forming a foundation for the proposed Influencer Partnership Management System.

The author of the paper “Social Media Platforms in Influencer Marketing”, Gretzel (2018) discussed the evolution of social media platforms like Instagram and YouTube in driving influencer marketing. The paper examined how platforms enable influencers to engage with niche audiences and deliver tailored marketing messages. Gretzel emphasized the scalability of influencer marketing and the challenges businesses face in managing multiple campaigns simultaneously. The insights call for an integrated system to centralize data, enabling businesses to efficiently manage influencer collaborations across diverse platforms.

Hughes et al. (2019) the author of the paper “Challenges in Managing Influencer Collaborations”, analyzed the operational challenges businesses face in influencer marketing, including contract negotiation, performance tracking, and ROI measurement. The paper highlighted inefficiencies caused by decentralized data and manual processes. It recommended adopting technology-driven solutions to address these challenges. This study directly supports the development of the proposed system by emphasizing the importance of a relational database that ensures seamless tracking and evaluation of influencer partnerships.

Kapitan & Silvera (2016) the authors of the paper “Selecting the Right Influencers for Campaigns”, investigated the criteria for selecting influencers based on audience demographics, engagement rates, and alignment with brand identity. Their findings demonstrated that improper selection could result in diminished campaign effectiveness. The paper stressed the role of data analytics in identifying influencers who maximize ROI. The proposed system incorporates these insights by using structured data to match influencers with brands and monitor their campaign outcomes.

In the paper “Data Analytics in Marketing Strategies”, Chaffey and Ellis-Chadwick (2019) explored the application of data analytics in optimizing marketing strategies. They argued that performance metrics, such as engagement rates and revenue attribution, are critical for assessing campaign success. The paper highlighted the role of centralized data systems in providing actionable insights. These findings align with the objectives of the proposed system, which focuses on tracking and analyzing key metrics to evaluate the impact of influencer collaborations.

Silberschatz et al. (2010) in the paper “Relational Database Systems for Data Management”, provided a detailed examination of relational database management systems (RDBMS) and their applications in handling structured datasets. The authors emphasized the importance of data integrity, consistency, and scalability in designing database solutions. The concepts outlined in this paper directly inform the technical structure of the proposed system, ensuring efficient organization and retrieval of influencer-related data.

Lanz et al. (2022) the author of “Automation in Influencer Marketing”, discussed the impact of automation tools and AI in streamlining influencer marketing processes. Their research demonstrated how technology reduces manual effort, improves data accuracy, and supports real-time performance tracking. These findings highlight the potential of integrating automation features into the proposed system to enhance its functionality and reduce administrative overhead.

## PROJECT TECHNICAL REPORT

In the research paper, “The Role of Performance Metrics in Campaign Evaluation”, Lou and Yuan (2019), investigated the significance of performance metrics, such as click-through rates, engagement levels, and sales conversions, in influencer marketing. The paper emphasized that systematic collection and analysis of these metrics are essential for determining the ROI of campaigns. The proposed system builds on these insights by integrating comprehensive performance tracking capabilities, enabling businesses to make informed, data-driven decisions.

Each study contributes critical insights into the challenges and opportunities of influencer marketing. Together, they highlight the importance of a centralized, data-driven system to manage influencer partnerships effectively. The proposed Influencer Partnership Management System leverages these findings to offer a comprehensive solution, addressing operational inefficiencies and enabling businesses to optimize their marketing efforts.

## Assumptions and Special Considerations

### Assumptions:

- Influencers are allowed to create an unlimited number of content posts for each product throughout the duration of a contract, but they will be allowed to release only one content post per product per day.
- Brands can engage with multiple influencers for different contracts.
- Influencers are able to create content about any products sold by the brands they have signed a contract with.
- Performance metrics are collected periodically to analyze collaboration success.
- Metrics will only be collected on content posted between the contract start and end dates for each partnership.

### Special Considerations:

- The Revenue attribute in the Performance Metrics entity represents the value related to the total products sold in the specific campaign.
- Contracts are designed to be linked uniquely to each brand-influencer collaboration.

## Statement of Work

### Overview

This project involves the design and development of a database to track and manage information related to influencer partnerships for marketing campaigns. The database will serve as the central data repository for storing and analyzing key details about influencers, their content, partnered brands, products promoted, contracts, and performance metrics. This system aims to enhance the efficiency and effectiveness of influencer marketing campaigns by providing businesses with structured, actionable data. By enabling streamlined management and robust performance analysis, the database will empower organizations to optimize their collaborations, improve ROI, and adapt to the dynamic nature of the digital marketing landscape.

## PROJECT TECHNICAL REPORT

### Purpose and Objective

The database will facilitate the creation and maintenance of a centralized system for managing influencer partnerships. It will enable businesses to search for influencers based on various criteria, such as platform, follower count, engagement rates, and expertise in specific industries. The system will also store detailed data on contracts, content created, and associated metrics such as likes, shares, link clicks, and revenue generated. By supporting advanced data queries, the database will allow users to analyze campaign success and make informed decisions about future collaborations. Moreover, the database will be designed with scalability in mind, allowing for the easy addition of new features and search criteria to accommodate evolving business needs and user feedback.

### Project Scope

The scope of this project is strictly limited to the design and implementation of the database for influencer partnership management. In-scope activities include:

1. **Requirement Documentation:** Gathering and documenting the functional and technical requirements for the database.
2. **Entity-Relationship Modeling:** Designing the database schema using entity-relationship diagrams to ensure all business relationships are accurately represented.
3. **SQL Script Development:** Writing Data Definition Language (DDL) scripts to define and implement the database structure.
4. **Demonstrative Queries:** Creating Data Manipulation Language (DML) scripts and SQL queries to demonstrate the intended functionality, such as searching for influencers, generating performance reports, and analyzing campaign results.

### Database Goals, Expectations, and Deliverables

**Comprehensive Database Solution:** Upon completion of this project, we designed a unified, scalable, and efficient relational database that ensures high data integrity, supports quick querying, and enables flexible customization for evolving business needs. The system will centralize the management of influencers, brands, products, contracts, and performance metrics while providing accurate tools for tracking campaign performance and calculating ROI to facilitate data-driven decision-making. Deliverables include this statement of work document, an entity-relationship diagram defining the structure of the database, DDL scripts for creating the database, example DML and SQL scripts that demonstrate proper usage of the database, and a final report on the project as a whole.

### Database Benefit

The database will enable businesses to track influencer performance and measure key metrics like engagement, sales, and ROI, allowing for more informed decisions in selecting influencers and optimizing marketing strategies, ultimately improving the effectiveness of influencer partnerships.

### Project Hardware and Software Tools Diagram Tool

ER-Assistant Version 2.10, running on Windows 10

### Office Productivity Tools

Microsoft Office 365, running on Windows 10

### Database

PostgreSQL - PgAdmin4 Version 8.10 running on Windows 10

## PROJECT TECHNICAL REPORT

### General Naming Conventions

- Use only letters, numbers, and underscores in names, with no consecutive underscores.
- Replace spaces in names with underscores (e.g., "first name" → `first_name`).
- Avoid abbreviations; if necessary, use only well-known ones.
- Use plural nouns for table names and singular nouns for column names.
- Ensure table and column names are unique and don't overlap.
- For relationship tables, avoid combining the names of the related tables.
- Avoid using generic names like `id` for primary keys; be more specific (e.g., `user_id`).
- Use lowercase for column names whenever possible.
- Add meaningful suffixes to column names for clarity, like `_id` or `_name`.
- Write SQL reserved keywords (e.g., `SELECT`, `WHERE`) in uppercase.
- Prefer full words over abbreviations in names and keywords.

### Query Syntax

- Use ANSI SQL keywords instead of database-specific ones for better compatibility.
- Retain natural language spaces for clarity.
- Align root keywords in code by adding spaces to make it easier to scan.
- Include spaces around `=` signs, after commas, and around apostrophes unless inside parentheses or followed by a comma or semicolon.
- Use newlines consistently before `AND` or `OR`, after semicolons, and to separate related sections of code.
- Align `JOIN` statements for better readability and indent subqueries to match the main query style.
- Use `BETWEEN` for range conditions instead of multiple `AND` clauses.
- Use `IN()` instead of multiple `OR` statements for compactness and readability.
- Apply `CASE` expressions to process values within the database before retrieval.
- Avoid `UNION` clauses and temporary tables unless necessary for performance.
- Prefer portable data types over vendor-specific ones for cross-database compatibility.
- Use `NUMERIC` or `DECIMAL` types instead of `REAL` or `FLOAT`, unless floating-point precision is explicitly required.

### Create Syntax

- Ensure default values match the data type of their column.
- Define default values immediately after the data type and before any `NOT NULL` constraints.
- Choose keys that are unique, stable, and unlikely to change over time.
- Use keys with values that can be validated against a standard format.
- Keep keys simple, but use compound keys when necessary.
- Every table must have at least one key to be complete and functional.
- Name constraints (other than `UNIQUE`, `PRIMARY KEY`, and `FOREIGN KEY`) descriptively for clarity.
- Place multi-column constraints near the column definitions when possible or at the end of the `CREATE TABLE` statement for complex cases.
- Table-level constraints should appear at the end of the table definition.
- Use alphabetical order for clauses like `ON DELETE` and `ON UPDATE`.
- Align query parts consistently to improve readability.
- Use `LIKE` or `SIMILAR TO` constraints to validate the format of known string patterns.

## PROJECT TECHNICAL REPORT

- Apply CHECK() constraints to ensure numerical values stay within a valid range and prevent truncation errors.
- Avoid separating values and units into different columns; the column values should clearly indicate their units.
- Use specialized tools for handling schema-less data instead of Entity Attribute Value (EAV) tables.
- Avoid splitting data across multiple tables based on external factors like location or time-based archiving.

### Project Management Methodology

The initial design of the database may be carried out in a linear fashion similar to simple interpretations of the waterfall model. This early implementation should seek to satisfy the preliminary database requirements established at the outset of the parent project. Following the completion of that implementation, the database team should switch to a project management methodology that emphasizes rapid iteration; ideally this should be the same methodology the software development team is using, to help facilitate cooperation and communication between the two teams. From that point on, the database team should revise the database design iteratively based on changes made to the software project's design and on feedback from the software development team.

## Requirement Definition Document

### Business Rules

- An influencer may create zero to many content\_creation posts; a content\_creation post must be created by only one influencer.
- A content\_creation post must refer to only one product.
- A brand may have zero to many contract; a contract must belong to only one brand.
- A brand may sell one or many product; a product may sold by only one brand.
- An influencer may have zero, one or many performance\_metrics entries; a performance\_metrics entry may relate to only one influencer.
- A contract may have zero to many performance\_metrics entries; a performance\_metrics entry may relate to only one contract.
- A product may have zero to many performance\_metrics entries; a performance\_metrics entry may relate to only one product.

### Entity and Attribute Description

Each entity in the database schema is described below, along with their attributes, primary keys (PK), and foreign keys (FK):

#### 1. Influencer

- **Description:** Represents individuals who promote products and create content.
- **Attributes:**
  - InfluencerID (PK): Unique identifier for each influencer.
  - first\_name: The first name of the influencer.
  - last\_name: The last name of the influencer.
  - Platform: The primary platform where the influencer is active.
  - Follower\_count: Total number of followers the influencer has.

## PROJECT TECHNICAL REPORT

- Email: Contact email of the influencer.

### 2. Content\_creation

- **Description:** Represents content created by influencers, linking them to specific products.
- **Attributes:**
  - PostID (PK): Unique identifier for each post.
  - InfluencerID (FK): References the influencer creating the content.
  - ProductID (FK): References the product featured in the content.
  - Content\_type: Type of content created (e.g., video, blog post).
  - Date\_posted: Date when the content was posted.
  - Discount\_link: Link to a discount or promotional page.
  - Total\_post\_likes: Total number of likes the post received.
  - Total\_post\_shared: Total number of times the post was shared.
  - Total\_link\_clicks: Number of clicks on the discount link.

### 3. Contract

- **Description:** Represents agreements between brands and influencers.
- **Attributes:**
  - ContractID (PK): Unique identifier for each contract.
  - Brand\_ID (FK): References the brand associated with the contract.
  - InfluencerID (FK): References the influencer associated with the contract.
  - Start\_date: Start date of the contract.
  - End\_date: End date of the contract.
  - Pay: Payment amount for the contract.
  - Signed\_date: Date when the contract was signed.

### 4. Brand

- **Description:** Represents brands partnering with influencers.
- **Attributes:**
  - Brand\_ID (PK): Unique identifier for each brand.
  - Brand\_name: Name of the brand.
  - Industry: Industry to which the brand belongs.
  - Contact\_Info: Contact details for the brand.
  - Website: Official website of the brand.

### 5. Product

- **Description:** Represents products promoted by influencers.
- **Attributes:**
  - ProductID (PK): Unique identifier for each product.
  - Name: Name of the product.
  - Category: Category to which the product belongs.
  - Price: Price of the product.
  - Brand\_ID (FK): References the brand producing the product.
  - Manufacturer\_country: Country where the product is manufactured.
  - Rating: Average rating for the product.

### 6. Performance Metrics

- **Description:** Represents the performance of influencer posts and contracts.
- **Attributes:**



## PROJECT TECHNICAL REPORT

- InfluencerID (PK, FK): References the influencer associated with the performance metrics.
- ProductID (PK, FK): References the product being evaluated.
- Total\_posts: Total number of posts related to the contract.
- Total\_shared: Total number of times the content was shared.
- Total\_disc\_link\_clicks: Total number of discount link clicks.
- Products\_sold: Number of products sold due to the content.
- Revenue: Revenue attributed to the products sold.

## Relationship and Cardinality Description

### **Relationship:** influencer\_content

- Cardinality: 1:M between influencer and content\_creation
- Business rule: an influencer may create zero to many content\_creation posts; a content\_creation post must be created by only one influencer.

### **Relationship:** product\_content

- Cardinality: 1:M between product and content\_creation
- Business rule: a product may be referenced zero to many times in content\_creation posts; a content\_creation post must reference only one product.

### **Relationship:** brand\_contract

- Cardinality: 1:M between brand and contract
- Business rule: a brand may have zero to many contract; a contract must belong to only one brand.

### **Relationship:** influencer\_contract

- Cardinality: 1:M between influencer and contract
- Business rule: an influencer may have zero, one or many contract; a contract must belong to one & only one influencer.

### **Relationship:** brand\_product

- Cardinality: 1:M between brand and product
- Business rule: a brand may sell one or many product; a product may sold by only one brand.

### **Relationship:** influencer\_performance

- Cardinality: 1:M between influencer and performance\_metrics
- Business rule: an influencer may have zero, one or many performance\_metrics entries; a performance\_metrics entry may relate to only one influencer.

### **Relationship:** contract\_performance

- Cardinality: 1:M between contract and performance\_metrics

## PROJECT TECHNICAL REPORT

- Business rule: a contract may have zero to many performance\_metrics entries; a performance\_metrics entry may relate to only one contract.

### **Relationship:** product\_performance

- Cardinality: 1:M between product and performance\_metrics
- Business rule: a product may have zero to many performance\_metrics entries; a performance\_metrics entry may relate to only one product.

## Normalization Steps

### Step 1: Create Unnormalized Tables with Performance Attributes Integrated

#### **1. Influencer Table (Unnormalized)**

- Attributes:
  - InfluencerID (PK)
  - first\_name
  - last\_name
  - Platform
  - Follower\_count
  - Email

#### **2. Content\_creation Table (Unnormalized)**

- Attributes:
  - PostID (PK)
  - InfluencerID (FK)
  - ProductID (FK)
  - Content\_type
  - Date\_posted
  - Discount\_link
  - Total\_post\_likes
  - Total\_post\_shared
  - Total\_link\_clicks
  - Products\_sold: Number of products sold via the content.
  - Revenue: Revenue generated from the content.

## PROJECT TECHNICAL REPORT

### 3. Contract Table (Unnormalized)

- Attributes:
  - ContractID (PK)
  - Brand\_ID (FK)
  - InfluencerID (FK)
  - Start\_date
  - End\_date
  - Pay
  - Signed\_date
  - Total\_posts: Total number of posts under the contract.
  - Total\_shared: Total number of shares for posts under the contract.
  - Total\_disc\_link\_clicks: Total number of discount link clicks related to the contract.

### 4. Brand Table (Unnormalized)

- Attributes:
  - Brand\_ID (PK)
  - Brand\_name
  - Industry
  - Contact\_Info
  - Website

### 5. Product Table (Unnormalized)

- Attributes:
  - ProductID (PK)
  - Name
  - Category
  - Price
  - Brand\_ID (FK)
  - Manufacturer\_country
  - Rating

### **Explanation:**

- The **Content\_creation** table contains attributes related to the performance of individual posts, such as Products\_sold and Revenue.

## PROJECT TECHNICAL REPORT

- The **Contract** table holds performance attributes related to the entire contract, including Total\_posts, Total\_shared, and Total\_disc\_link\_clicks.

### Step 2: Apply 1NF (First Normal Form)

**Definition:** Ensure that all attributes have atomic values, no repeating groups, and that each record has a unique identifier.

#### Normalized Tables in 1NF:

1. **Influencer Table:**
  - Attributes: InfluencerID (PK), first\_name, last\_name, Platform, Follower\_count, Email.
2. **Content\_creation Table:**
  - Attributes: PostID (PK), InfluencerID (FK), ProductID (FK), Content\_type, Date\_posted, Discount\_link, Total\_post\_likes, Total\_post\_shared, Total\_link\_clicks, Products\_sold, Revenue.
3. **Contract Table:**
  - Attributes: ContractID (PK), Brand\_ID (FK), InfluencerID (FK), Start\_date, End\_date, Pay, Signed\_date, Total\_posts, Total\_shared, Total\_disc\_link\_clicks.
4. **Brand Table:**
  - Attributes: Brand\_ID (PK), Brand\_name, Industry, Contact\_Info, Website.
5. **Product Table:**
  - Attributes: ProductID (PK), Name, Category, Price, Brand\_ID (FK), Manufacturer\_country, Rating.

### Step 3: Apply 2NF (Second Normal Form)

**Definition:** Ensure that all non-prime attributes are fully functionally dependent on the whole primary key, eliminating partial dependencies.

#### Changes for 2NF:

- Move performance-related attributes from the **Content\_creation** and **Contract** tables that do not depend solely on their primary keys to a new table.

#### Updated Tables in 2NF:

1. **Influencer Table:** No changes needed.
2. **Content\_creation Table:**

## PROJECT TECHNICAL REPORT

- Remove Products\_sold and Revenue as they are dependent on both PostID and InfluencerID. These attributes will be moved to a new table.

### 3. Contract Table:

- Remove Total\_posts, Total\_shared, and Total\_disc\_link\_clicks as they are performance-related and not fully dependent on just ContractID.

### New Table: Performance Metrics (2NF)

- **Attributes:**
  - InfluencerID (FK)
  - ProductID (FK)
  - Total\_posts
  - Total\_shared
  - Total\_disc\_link\_clicks
  - Products\_sold
  - Revenue

## Step 4: Apply 3NF (Third Normal Form)

**Definition:** Eliminate transitive dependencies so that non-prime attributes are only dependent on primary keys.

### Changes for 3NF:

- Ensure that attributes like Brand\_ID in the **Product** table only reference the **Brand** table without duplicating brand-specific details.
- Ensure that Revenue in the **Performance Metrics** table is not repeated from the **Product** table and is correctly referenced.

### Final Normalized Tables in 3NF:

#### 1. Influencer Table:

- Attributes: InfluencerID (PK), first\_name, last\_name, Platform, Follower\_count, Email.

#### 2. Content\_creation Table:

- Attributes: PostID (PK), InfluencerID (FK), ProductID (FK), Content\_type, Date\_posted, Discount\_link, Total\_post\_likes, Total\_post\_shared, Total\_link\_clicks.

#### 3. Contract Table:

## PROJECT TECHNICAL REPORT

- Attributes: ContractID (PK), Brand\_ID (FK), InfluencerID (FK), Start\_date, End\_date, Pay, Signed\_date.

### 4. Brand Table:

- Attributes: Brand\_ID (PK), Brand\_name, Industry, Contact\_Info, Website.

### 5. Product Table:

- Attributes: ProductID (PK), Name, Category, Price, Brand\_ID (FK), Manufacturer\_country, Rating.

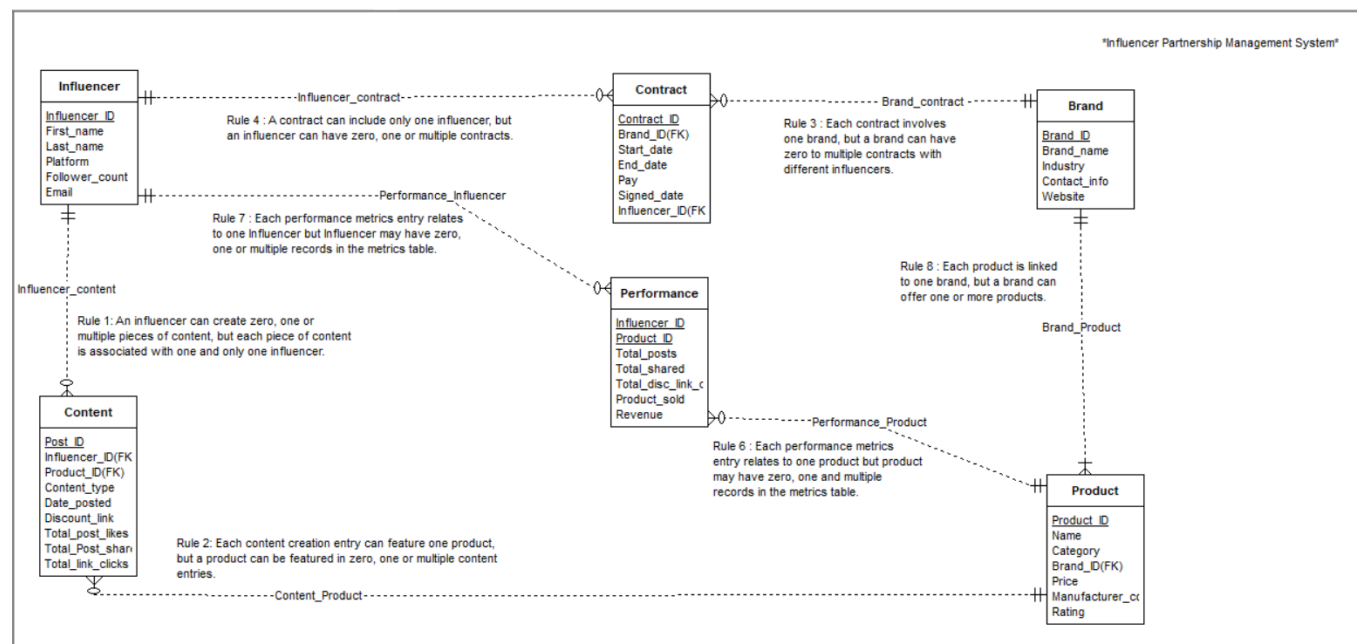
### 6. Performance Metrics Table:

- Attributes: InfluencerID (PK, FK), ProductID (PK, FK), Total\_posts, Total\_shared, Total\_disc\_link\_clicks, Products\_sold, Revenue.

This is the final structure ensures all tables comply with **3NF**, maintaining data integrity and eliminating redundancy.

## Detailed Database Design

### Entity Relationship Diagram (ER- Diagram)



### Data Definition Language Source Code (DDL)

```

--drop triggers
DROP TRIGGER IF EXISTS TRG_influencer on influencer ;
DROP TRIGGER IF EXISTS TRG_brand on brand ;
DROP TRIGGER IF EXISTS TRG_product on product ;
DROP TRIGGER IF EXISTS TRG_contract on contract ;
  
```

## PROJECT TECHNICAL REPORT

DROP TRIGGER IF EXISTS TRG\_content on content\_creation ;

DROP FUNCTION IF EXISTS trg\_influencer() ;  
DROP FUNCTION IF EXISTS trg\_brand() ;  
DROP FUNCTION IF EXISTS trg\_product() ;  
DROP FUNCTION IF EXISTS rg\_contract() ;  
DROP FUNCTION IF EXISTS trg\_content() ;

--drop sequences

DROP SEQUENCE IF EXISTS influencer\_id\_seq ;  
DROP SEQUENCE IF EXISTS brand\_id\_seq ;  
DROP SEQUENCE IF EXISTS product\_id\_seq ;  
DROP SEQUENCE IF EXISTS contract\_id\_seq ;  
DROP SEQUENCE IF EXISTS content\_id\_seq ;

--drop views

DROP VIEW IF EXISTS InfluencerInfo ;  
DROP VIEW IF EXISTS BrandInfo ;  
DROP VIEW IF EXISTS ProductInfo ;  
DROP VIEW IF EXISTS ContractInfo ;  
DROP VIEW IF EXISTS PerformanceSummary ;

--drop indices

DROP INDEX IF EXISTS IDX\_influencer\_name ;  
DROP INDEX IF EXISTS IDX\_influencer\_platform ;  
DROP INDEX IF EXISTS IDX\_brand\_name ;  
DROP INDEX IF EXISTS IDX\_brand\_industry ;  
DROP INDEX IF EXISTS IDX\_brand\_contact\_info ;  
DROP INDEX IF EXISTS IDX\_product\_name ;  
DROP INDEX IF EXISTS IDX\_product\_brand\_id\_fk ;  
DROP INDEX IF EXISTS IDX\_product\_category ;  
DROP INDEX IF EXISTS IDX\_product\_price ;  
DROP INDEX IF EXISTS IDX\_product\_rating ;  
DROP INDEX IF EXISTS IDX\_contract\_brand\_id\_fk ;  
DROP INDEX IF EXISTS IDX\_contract\_influencer\_id\_fk ;  
DROP INDEX IF EXISTS IDX\_contract\_start\_date ;  
DROP INDEX IF EXISTS IDX\_contract\_end\_date ;  
DROP INDEX IF EXISTS IDX\_contract\_pay ;  
DROP INDEX IF EXISTS IDX\_content\_influencer\_id\_fk ;  
DROP INDEX IF EXISTS IDX\_content\_product\_id\_fk ;  
DROP INDEX IF EXISTS IDX\_content\_creation\_total\_post\_likes ;  
DROP INDEX IF EXISTS IDX\_content\_creation\_total\_post\_shares ;  
DROP INDEX IF EXISTS IDX\_content\_creation\_total\_link\_clicks ;  
DROP INDEX IF EXISTS IDX\_performance\_influencer\_id\_fk ;  
DROP INDEX IF EXISTS IDX\_performance\_product\_id\_fk ;  
DROP INDEX IF EXISTS IDX\_performance\_metrics\_total\_posts ;  
DROP INDEX IF EXISTS IDX\_performance\_metrics\_total\_shares ;  
DROP INDEX IF EXISTS IDX\_performance\_metrics\_products\_sold ;  
DROP INDEX IF EXISTS IDX\_performance\_metrics\_revenue ;

## PROJECT TECHNICAL REPORT

--drop tables

DROP TABLE IF EXISTS performance\_metrics ;

DROP TABLE IF EXISTS content\_creation ;

DROP TABLE IF EXISTS contract ;

DROP TABLE IF EXISTS product ;

DROP TABLE IF EXISTS brand ;

DROP TABLE IF EXISTS influencer ;

--drop schema

DROP SCHEMA IF EXISTS Influencer\_partnership CASCADE;

/\*Create schema named project and tables\*/

CREATE SCHEMA Influencer\_partnership;

set search\_path to Influencer\_partnership;

--CREATE TABLES

--create influencer table

```
CREATE TABLE influencer (
    influencer_id INT PRIMARY KEY,
    first_name   VARCHAR(25) NOT NULL,
    last_name    VARCHAR(25) NOT NULL,
    platform     VARCHAR(25) NOT NULL,
    follower_count INT NOT NULL,
    email        VARCHAR(50) NOT NULL
);
```

--create brand table

```
CREATE TABLE brand (
    brand_id INT PRIMARY KEY,
    brand_name VARCHAR(50) NOT NULL,
    industry   VARCHAR(50) NOT NULL,
    contact_info VARCHAR(50) NOT NULL,
    website    VARCHAR(100) NOT NULL
);
```

--create product table

```
CREATE TABLE product (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(50) NOT NULL,
    category    VARCHAR(50) NOT NULL,
    price       NUMERIC(7,2) NOT NULL,
    brand_id    INT NOT NULL,
    manufacturer_country VARCHAR(50) NOT NULL,
    rating      NUMERIC(2,1) NOT NULL,
    CONSTRAINT fk_product_brand FOREIGN KEY (brand_id) REFERENCES
brand(brand_id)
);
```



## PROJECT TECHNICAL REPORT

--create contract table

```
CREATE TABLE contract(
    contract_id INT PRIMARY KEY,
    brand_id INT NOT NULL,
    influencer_id INT NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE NOT NULL,
    pay NUMERIC(7,2) NOT NULL,
    signed_date DATE NOT NULL,
    CONSTRAINT fk_contract_brand FOREIGN KEY (brand_id) REFERENCES
brand(brand_id),
    CONSTRAINT fk_contract_influencer FOREIGN KEY (influencer_id)
REFERENCES influencer(influencer_id)
);
```

--create content\_creation table

```
CREATE TABLE content_creation(
    post_id INT PRIMARY KEY,
    influencer_id INT NOT NULL,
    product_id INT NOT NULL,
    content_type VARCHAR(50) NOT NULL,
    date_posted DATE NOT NULL,
    discount_link VARCHAR(255) NOT NULL,
    total_post_likes INT NOT NULL,
    total_post_shares INT NOT NULL,
    total_link_clicks INT NOT NULL,
    CONSTRAINT fk_content_influencer FOREIGN KEY (influencer_id)
REFERENCES influencer(influencer_id),
    CONSTRAINT fk_content_product FOREIGN KEY (product_id) REFERENCES
product(product_id)
);
```

--create performance\_metrics table

```
CREATE TABLE performance_metrics (
    influencer_id INT NOT NULL,
    product_id INT NOT NULL,
    total_posts INT NOT NULL,
    total_shares INT NOT NULL,
    total_clicks INT NOT NULL,
    products_sold INT NOT NULL,
    revenue NUMERIC(8,2) NOT NULL,
    PRIMARY KEY (influencer_id, product_id),
    CONSTRAINT fk_performance_influencer FOREIGN KEY (influencer_id)
REFERENCES influencer(influencer_id),
    CONSTRAINT fk_performance_product FOREIGN KEY (product_id)
REFERENCES product(product_id)
);
```

--CREATE INDICES

## PROJECT TECHNICAL REPORT

```

--influencer
--natural keys
CREATE INDEX IDX_influencer_name ON influencer(first_name, last_name) ;
--frequently queried
CREATE INDEX IDX_influencer_platform ON influencer(platform) ;

--brand
--natural keys
CREATE INDEX IDX_brand_name ON brand(brand_name) ;
--frequently queried
CREATE INDEX IDX_brand_industry ON Brand (industry);
CREATE INDEX IDX_brand_contact_info ON Brand (contact_info);

--product
CREATE INDEX IDX_product_name ON product(product_name) ;
--foreign keys
CREATE INDEX IDX_product_brand_id_fk ON product(brand_id) ;
--frequently queried
CREATE INDEX IDX_product_category ON product(category);
CREATE INDEX IDX_product_price ON product(price);
CREATE INDEX IDX_product_rating ON product(rating);

--contract
--foreign keys
CREATE INDEX IDX_contract_brand_id_fk ON contract(brand_id) ;
CREATE INDEX IDX_contract_influencer_id_fk ON contract(influencer_id) ;
--frequently queried
CREATE INDEX IDX_contract_start_date ON contract(start_date);
CREATE INDEX IDX_contract_end_date ON contract(end_date);
CREATE INDEX IDX_contract_pay ON contract(pay);

--content
--foreign keys
CREATE INDEX IDX_content_influencer_id_fk ON content_creation(influencer_id) ;
CREATE INDEX IDX_content_product_id_fk ON content_creation(product_id) ;
--frequently queried
CREATE INDEX IDX_content_creation_total_post_likes ON
content_creation(total_post_likes);
CREATE INDEX IDX_content_creation_total_post_shares ON
content_creation(total_post_shares);
CREATE INDEX IDX_content_creation_total_link_clicks ON
content_creation(total_link_clicks);

--performance metrics
--foreign keys
CREATE INDEX IDX_performance_influencer_id_fk ON
performance_metrics(influencer_id) ;
CREATE INDEX IDX_performance_product_id_fk ON performance_metrics(product_id) ;
--frequently queried

```

## PROJECT TECHNICAL REPORT

```

CREATE INDEX IDX_performance_metrics_total_posts ON
performance_metrics(total_posts);
CREATE INDEX IDX_performance_metrics_total_shares ON
performance_metrics(total_shares);
CREATE INDEX IDX_performance_metrics_products_sold ON
performance_metrics(products_sold);
CREATE INDEX IDX_performance_metrics_revenue ON performance_metrics(revenue);

```

## --CREATE VIEWS

```

CREATE OR REPLACE VIEW InfluencerInfo AS
    SELECT influencer_id, first_name, last_name, platform, follower_count
    FROM influencer ;

```

```

CREATE OR REPLACE VIEW BrandInfo AS
    SELECT brand_id, brand_name, industry, website
    FROM brand ;

```

```

CREATE OR REPLACE VIEW ProductInfo AS
    SELECT b.brand_name, p.product_id, p.product_name, p.category, p.price, p.rating
    FROM product p LEFT JOIN brand b
    ON p.brand_id = b.brand_id ;

```

```

CREATE OR REPLACE VIEW ContractInfo AS
    SELECT c.contract_id, b.brand_name, i.first_name, i.last_name, c.start_date,
c.end_date
    FROM contract c LEFT JOIN brand b
    ON c.brand_id = b.brand_id
    LEFT JOIN influencer i
    ON c.influencer_id = i.influencer_id ;

```

```

CREATE OR REPLACE VIEW PerformanceSummary AS
    SELECT CONCAT(i.first_name, ' ', i.last_name) AS influencer_name,
p.product_name, pm.total_posts, pm.total_shares, pm.products_sold, pm.revenue
    FROM performance_metrics pm LEFT JOIN influencer i
    ON pm.influencer_id = i.influencer_id
    LEFT JOIN product p
    ON pm.product_id = p.product_id ;

```

## --CREATE SEQUENCES

```

--create sequence influencer_id_seq
CREATE SEQUENCE influencer_id_seq
    INCREMENT BY 1
    START WITH 101
    NO MAXVALUE
    MINVALUE 101
    NO CYCLE;

```

```

--create sequence brand_id_seq
CREATE SEQUENCE brand_id_seq
    INCREMENT BY 1

```

## PROJECT TECHNICAL REPORT

```
START WITH 101
NO MAXVALUE
MINVALUE 101
NO CYCLE;
```

```
--create sequence product_id_seq
CREATE SEQUENCE product_id_seq
    INCREMENT BY 1
    START WITH 4501
    NO MAXVALUE
    MINVALUE 4501
    NO CYCLE;
```

```
--create sequence contract_id_seq
CREATE SEQUENCE contract_id_seq
    INCREMENT BY 1
    START WITH 3101
    NO MAXVALUE
    MINVALUE 3101
    NO CYCLE;
```

```
--create sequence content_id_seq
CREATE SEQUENCE content_id_seq
    INCREMENT BY 1
    START WITH 10031
    NO MAXVALUE
    MINVALUE 10031
    NO CYCLE;
```

```
--create trigger functions
--create trigger function trg_influencer
CREATE OR REPLACE FUNCTION trg_influencer() RETURNS TRIGGER AS $$
BEGIN
    IF NEW.influencer_id IS NULL THEN
        NEW.influencer_id := nextval('influencer_id_seq');
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
--create trigger function trg_brand
CREATE OR REPLACE FUNCTION trg_brand() RETURNS TRIGGER AS $$
BEGIN
    IF NEW.brand_id IS NULL THEN
        NEW.brand_id := nextval('brand_id_seq');
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

## PROJECT TECHNICAL REPORT

```
--create trigger function trg_product
CREATE OR REPLACE FUNCTION trg_product() RETURNS TRIGGER AS $$
BEGIN
    IF NEW.product_id IS NULL THEN
        NEW.product_id := nextval('product_id_seq');
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
--create trigger function trg_contract
CREATE OR REPLACE FUNCTION trg_contract() RETURNS TRIGGER AS $$
BEGIN
    IF NEW.contract_id IS NULL THEN
        NEW.contract_id := nextval('contract_id_seq');
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
--create trigger function trg_content
CREATE OR REPLACE FUNCTION trg_content() RETURNS TRIGGER AS $$
BEGIN
    IF NEW.post_id IS NULL THEN
        NEW.post_id := nextval('content_id_seq');
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
--create triggers
--create trigger TRG_influencer
CREATE TRIGGER TRG_influencer
    BEFORE INSERT OR UPDATE ON influencer
    FOR EACH ROW
    EXECUTE FUNCTION trg_influencer();
```

```
--create trigger TRG_brand
CREATE TRIGGER TRG_brand
    BEFORE INSERT OR UPDATE ON brand
    FOR EACH ROW
    EXECUTE FUNCTION trg_brand();
```

```
--create trigger TRG_product
CREATE TRIGGER TRG_product
    BEFORE INSERT OR UPDATE ON product
    FOR EACH ROW
    EXECUTE FUNCTION trg_product();
```

```
--create trigger TRG_contract
```

## PROJECT TECHNICAL REPORT

```

CREATE TRIGGER TRG_contract
  BEFORE INSERT OR UPDATE ON contract
  FOR EACH ROW
  EXECUTE FUNCTION trg_contract();

--create trigger TRG_content
CREATE TRIGGER TRG_content
  BEFORE INSERT OR UPDATE ON content_creation
  FOR EACH ROW
  EXECUTE FUNCTION trg_content();
-- Check the database for tables related to the Influencer Partnership Management System
SELECT TABLE_NAME
FROM information_schema.tables
WHERE TABLE_SCHEMA = 'influencer_partnership'
  AND TABLE_NAME IN ('influencer', 'content_creation', 'contract', 'brand', 'product',
'performance_metrics');

-- Check the status and creation details of objects related to the Influencer Partnership
Management System
-- Check table details in the influencer_partnership schema
SELECT TABLE_NAME,
       TABLE_SCHEMA,
       TABLE_TYPE
FROM information_schema.tables
WHERE TABLE_SCHEMA = 'influencer_partnership'
  AND TABLE_NAME IN ('influencer', 'content_creation', 'contract', 'brand', 'product',
'performance_metrics');
-- Check indices details in influencer_partnership schema
SELECT indexname AS INDEX_NAME,
       tablename AS TABLE_NAME
FROM pg_indexes
WHERE schemaname = 'influencer_partnership';
--Check Views details in influencer_partnership schema
SELECT TABLE_NAME AS VIEW_NAME
FROM information_schema.views
WHERE TABLE_SCHEMA = 'influencer_partnership';
--Check trigger details in influencer_partnership schema
SELECT tname AS TRIGGER_NAME,
       relname AS TABLE_NAME
FROM pg_trigger
JOIN pg_class ON pg_trigger.tgrelid = pg_class.oid
JOIN pg_namespace ON pg_class.relnamespace = pg_namespace.oid
WHERE pg_namespace.nspname = 'influencer_partnership';
--Check function details in influencer_partnership schema
SELECT routine_name AS FUNCTION_NAME,
       routine_type AS FUNCTION_TYPE,
       data_type AS RETURN_TYPE
FROM information_schema.routines
WHERE specific_schema = 'influencer_partnership';

```

## PROJECT TECHNICAL REPORT

-- Check for sequences in influencer\_partnership schema

SELECT sequence\_name

FROM information\_schema.sequences

WHERE sequence\_schema = 'influencer\_partnership';

OUTPUT OF DDL CODE:

### ➔ Check the database for tables related to influencer partnership management system

```

350 -- Check the database for tables related to the Influencer Partnership Management System
351 SELECT TABLE_NAME
352 FROM information_schema.tables
353 WHERE TABLE_SCHEMA = 'influencer_partnership'
354 AND TABLE_NAME IN ('influencer', 'content_creation', 'contract', 'brand', 'product', 'performance_metrics');

```

Data Output Messages Notifications

	table_name
1	brand
2	content_creation
3	contract
4	influencer
5	performance_metrics
6	product

Total rows: 6 of 6 Query complete 00:00:01.547 Ln 355, Col 1

### ➔ Check the table details related to influencer partnership management system

```

357 -- Check the status and creation details of objects related to the Influencer Partnership Management System
358 -- Check table details in the influencer_partnership schema
359 SELECT TABLE_NAME,
360        TABLE_SCHEMA,
361        TABLE_TYPE
362 FROM information_schema.tables
363 WHERE TABLE_SCHEMA = 'influencer_partnership'
364 AND TABLE_NAME IN ('influencer', 'content_creation', 'contract', 'brand', 'product', 'performance_metrics');

```

Data Output Messages Notifications

	table_name	table_schema	table_type
1	brand	influencer_partnership	BASE TABLE
2	content_creation	influencer_partnership	BASE TABLE
3	contract	influencer_partnership	BASE TABLE
4	influencer	influencer_partnership	BASE TABLE
5	performance_metrics	influencer_partnership	BASE TABLE
6	product	influencer_partnership	BASE TABLE

Total rows: 6 of 6 Query complete 00:00:00.160 Ln 364, Col 111

### ➔ Check the views related to influencer partnership management system

```

370 --Check Views details in influencer_partnership schema
371 SELECT TABLE_NAME AS VIEW_NAME
372 FROM information_schema.views
373 WHERE TABLE_SCHEMA = 'influencer_partnership';

```

Data Output Messages Notifications

	view_name
1	influencerinfo
2	brandinfo
3	productinfo
4	contractinfo
5	performancesummary

Total rows: 5 of 5 Query complete 00:00:00.102 Ln 373, Col 47

## PROJECT TECHNICAL REPORT

## ➔ Check the triggers related to influencer partnership management system

```

381 --Check function details in influencer_partnership schema
382 SELECT routine_name AS FUNCTION_NAME,
383        routine_type AS FUNCTION_TYPE,
384        data_type AS RETURN_TYPE
385 FROM information_schema.routines
386 WHERE specific_schema = 'influencer_partnership';

```

Data Output Messages Notifications

	function_name name	function_type character varying	return_type character varying
1	trg_influencer	FUNCTION	trigger
2	trg_brand	FUNCTION	trigger
3	trg_product	FUNCTION	trigger
4	trg_contract	FUNCTION	trigger
5	trg_content	FUNCTION	trigger

Total rows: 5 of 5 Query complete 00:00:00.089 Ln 382, Col 1

## ➔ Check the sequences related to influencer partnership management system

```

387 -- Check for sequences in influencer_partnership schema
388 SELECT sequence_name
389 FROM information_schema.sequences
390 WHERE sequence_schema = 'influencer_partnership';
391

```

Data Output Messages Notifications

	sequence_name name
1	influencer_id_seq
2	brand_id_seq
3	product_id_seq
4	contract_id_seq
5	content_id_seq

Total rows: 5 of 5 Query complete 00:00:00.082 Ln 388, Col 1

## Data Manipulation Language Source Code (DML)

set search\_path to influencer\_partnership;

--populate all tables

--influencer

```

INSERT INTO influencer (first_name, last_name, platform, follower_count, email)
VALUES ('Alice', 'Smith', 'Instagram', 50000, 'alice.smith@hotmail.com') ;
INSERT INTO influencer (first_name, last_name, platform, follower_count, email)
VALUES ('Bob', 'Johnson', 'YouTube', 150000, 'bob.johnson@gmail.com') ;
INSERT INTO influencer (first_name, last_name, platform, follower_count, email)
VALUES ('Charlie', 'Williams', 'TikTok', 75000, 'charlie.williams@yahoo.com') ;

```



## PROJECT TECHNICAL REPORT

```
INSERT INTO influencer (first_name, last_name, platform, follower_count, email)
VALUES ('Diana', 'Brown', 'Instagram', 120000, 'diana.brown@gmail.com') ;
INSERT INTO influencer (first_name, last_name, platform, follower_count, email)
VALUES ('Ethan', 'Davis', 'YouTube', 200000, 'ethan.davis@gmail.com') ;
```

```
--brand
```

```
INSERT INTO brand (brand_name, industry, contact_info, website)
VALUES ('TechGadgets', 'Technology', 'contact@techgadgets.com',
'https://www.techgadgets.com') ;
INSERT INTO brand (brand_name, industry, contact_info, website)
VALUES ('FashionHub', 'Fashion', 'info@fashionhub.com',
'https://www.fashionhub.com') ;
INSERT INTO brand (brand_name, industry, contact_info, website)
VALUES ('HealthPlus', 'Healthcare', 'support@healthplus.com',
'https://www.healthplus.com') ;
INSERT INTO brand (brand_name, industry, contact_info, website)
VALUES ('AutoDrive', 'Automotive', 'sales@autodrive.com',
'https://www.autodrive.com') ;
INSERT INTO brand (brand_name, industry, contact_info, website)
VALUES ('Foodies', 'Food & Beverage', 'hello@foodies.com',
'https://www.foodies.com') ;
```

```
--product
```

```
INSERT INTO product (product_name, category, price, brand_id, manufacturer_country,
rating)
VALUES ('Smartphone X', 'Electronics', 999.99, 101, 'USA', 4.5) ;
INSERT INTO product (product_name, category, price, brand_id, manufacturer_country,
rating)
VALUES ('Designer Dress', 'Apparel', 199.99, 102, 'Italy', 4.7) ;
INSERT INTO product (product_name, category, price, brand_id, manufacturer_country,
rating)
VALUES ('Vitamin Supplements', 'Health', 29.99, 103, 'Germany', 4.3) ;
INSERT INTO product (product_name, category, price, brand_id, manufacturer_country,
rating)
VALUES ('Electric Car Model S', 'Vehicles', 49999.99, 104, 'USA', 4.8) ;
INSERT INTO product (product_name, category, price, brand_id, manufacturer_country,
rating)
VALUES ('Organic Coffee', 'Beverages', 15.99, 105, 'Colombia', 4.6) ;
```

```
--contract
```

```
INSERT INTO contract (brand_id, influencer_id, start_date, end_date, pay, signed_date)
VALUES (101, 101, '2024-01-01', '2024-06-30', 5000.00, '2023-12-15') ;
INSERT INTO contract (brand_id, influencer_id, start_date, end_date, pay, signed_date)
VALUES (102, 102, '2024-02-01', '2024-07-31', 7000.00, '2024-01-10') ;
INSERT INTO contract (brand_id, influencer_id, start_date, end_date, pay, signed_date)
VALUES (103, 103, '2024-03-01', '2024-08-31', 6000.00, '2024-02-20') ;
INSERT INTO contract (brand_id, influencer_id, start_date, end_date, pay, signed_date)
```

## PROJECT TECHNICAL REPORT

```

VALUES (104, 104, '2024-04-01', '2024-09-30', 8000.00, '2024-03-25') ;
INSERT INTO contract (brand_id, influencer_id, start_date, end_date, pay, signed_date)
VALUES (105, 105, '2024-05-01', '2024-10-31', 5500.00, '2024-04-15') ;
INSERT INTO contract (brand_id, influencer_id, start_date, end_date, pay, signed_date)
VALUES (101, 103, '2024-06-01', '2024-11-30', 7500.00, '2024-05-15') ;
INSERT INTO contract (brand_id, influencer_id, start_date, end_date, pay, signed_date)
VALUES (102, 104, '2024-06-01', '2024-11-30', 7500.00, '2024-05-15') ;
INSERT INTO contract (brand_id, influencer_id, start_date, end_date, pay, signed_date)
VALUES (103, 105, '2024-07-01', '2024-12-31', 9000.00, '2024-06-01') ;
INSERT INTO contract (brand_id, influencer_id, start_date, end_date, pay, signed_date)
VALUES (104, 101, '2024-03-01', '2024-08-30', 4000.00, '2024-02-15') ;
INSERT INTO contract (brand_id, influencer_id, start_date, end_date, pay, signed_date)
VALUES (105, 102, '2024-02-01', '2024-05-31', 7000.00, '2024-01-20') ;

--content_creation
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (101, 4501, 'Photo', '2024-01-15',
'https://www.techgadgets.com/smartphone-x-101', 1000, 150, 200) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (102, 4502, 'Long Form Video', '2024-02-20',
'https://www.fashionhub.com/designer-dress-102', 2000, 250, 300) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (102, 4502, 'Short Form Video', '2024-03-30',
'https://www.fashionhub.com/designer-dress-102', 2000, 250, 300) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (103, 4503, 'Short Form Video', '2024-03-25',
'https://www.healthplus.com/vitamin-supplements-103', 1500, 200, 250) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (104, 4504, 'Photo', '2024-04-30', 'https://www.autodrive.com/electric-car-
model-s-104', 2500, 300, 400) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (105, 4505, 'Short Form Video', '2024-05-10',
'https://www.foodies.com/organic-coffee-105', 1800, 220, 280) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (105, 4505, 'Short Form Video', '2024-07-13',
'https://www.foodies.com/organic-coffee-105', 1600, 250, 300) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (105, 4505, 'Short Form Video', '2024-08-26',
'https://www.foodies.com/organic-coffee-105', 2000, 300, 430) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)

```

## PROJECT TECHNICAL REPORT

```

VALUES (103, 4501, 'Short Form Video','2024-07-
13','https://www.techgadgets.com/smartphone-x-103', 1700, 130, 150) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (104, 4502, 'Photo', '2024-08-20', 'https://www.fashionhub.com/designer-
dress-104', 5000, 430, 640) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (105, 4503, 'Long Form Video', '2024-09-25',
'https://www.healthplus.com/vitamin-supplements-105', 5700, 400, 1700) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (101, 4504, 'Photo', '2024-05-24', 'https://www.autodrive.com/electric-car-
model-s-101', 1050, 100, 540) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (101, 4504, 'Short Form Video', '2024-07-24',
'https://www.autodrive.com/electric-car-model-s-101', 2070, 240, 1005) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (102, 4505, 'Long Form Video', '2024-05-10',
'https://www.foodies.com/organic-coffee-102', 3100, 420, 570) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (102, 4505, 'Short Form Video', '2024-07-10',
'https://www.foodies.com/organic-coffee-102', 2300, 260, 700) ;
INSERT INTO content_creation (influencer_id, product_id, content_type, date_posted,
discount_link, total_post_likes, total_post_shares, total_link_clicks)
VALUES (101, 4501, 'Short Form Video', '2024-01-15',
'https://www.techgadgets.com/smartphone-x-101', 1300, 210, 370) ;

--performance_metrics
INSERT INTO performance_metrics (influencer_id, product_id, total_posts, total_shares,
total_clicks, products_sold, revenue)
VALUES (101, 4501, 2, 360, 570, 50, 49999.50);
INSERT INTO performance_metrics (influencer_id, product_id, total_posts, total_shares,
total_clicks, products_sold, revenue)
VALUES (102, 4502, 2, 500, 600, 30, 5999.70) ;
INSERT INTO performance_metrics (influencer_id, product_id, total_posts, total_shares,
total_clicks, products_sold, revenue)
VALUES (103, 4503, 1, 200, 250, 40, 1199.60) ;
INSERT INTO performance_metrics (influencer_id, product_id, total_posts, total_shares,
total_clicks, products_sold, revenue)
VALUES (104, 4504, 1, 300, 400, 20, 999999.80) ;
INSERT INTO performance_metrics (influencer_id, product_id, total_posts, total_shares,
total_clicks, products_sold, revenue)
VALUES (105, 4505, 3, 770, 1010, 60, 959.40) ;
INSERT INTO performance_metrics (influencer_id, product_id, total_posts, total_shares,
total_clicks, products_sold, revenue)
VALUES (103, 4501, 1, 130, 150, 30, 29999.70) ;

```

## PROJECT TECHNICAL REPORT

```
INSERT INTO performance_metrics (influencer_id, product_id, total_posts, total_shares,
                                total_clicks, products_sold, revenue)
VALUES (104, 4502, 1, 430, 640, 45, 8999.55) ;
INSERT INTO performance_metrics (influencer_id, product_id, total_posts, total_shares,
                                total_clicks, products_sold, revenue)
VALUES (105, 4503, 1, 400, 1700, 160, 4798.40) ;
INSERT INTO performance_metrics (influencer_id, product_id, total_posts, total_shares,
                                total_clicks, products_sold, revenue)
VALUES (101, 4504, 2, 340, 1545, 10, 499999.90) ;
INSERT INTO performance_metrics (influencer_id, product_id, total_posts, total_shares,
                                total_clicks, products_sold, revenue)
VALUES (102, 4505, 2, 680, 1270, 230, 3677.70) ;
```

## Querying Source Code

--Q1: Select all columns and all rows from one table (5 points)

```
SELECT * FROM content_creation;
```

--Q2: Select five columns and all rows from one table (5 points)

```
SELECT product_id, product_name, category, price, manufacturer_country
FROM product;
```

--Q3: Select all columns from all rows from one view (5 points)

```
SELECT * FROM ProductInfo;
```

--Q4: Using a join on 2 tables, select all columns and all rows from the tables without the use of a Cartesian product (5 points)

```
SELECT * FROM product p
JOIN brand b
ON p.brand_id = b.brand_id ;
```

--Q5: Select and order data retrieved from one table (5 points)

```
SELECT * FROM contract
ORDER BY pay DESC ;
```

--Q6: Using a join on 3 tables, select 5 columns from the 3 tables. Use syntax that would limit the output to 3 rows (5 points)

```
SELECT c.contract_id, i.first_name, i.last_name, p.product_name, p.price
FROM contract c
JOIN influencer i ON c.influencer_id = i.influencer_id
JOIN product p ON c.brand_id = p.brand_id
LIMIT 3;
```

--Q7: Select distinct rows using joins on 3 tables (5 points)

```
SELECT DISTINCT
i.first_name, i.last_name, p.product_name, b.brand_name, c.pay
FROM contract c
JOIN influencer i ON c.influencer_id = i.influencer_id
JOIN product p ON c.brand_id = p.brand_id
```

## PROJECT TECHNICAL REPORT

```
JOIN brand b ON p.brand_id = b.brand_id;
```

--Q8: Use GROUP BY and HAVING in a select statement using one or more tables (5 points)

```
SELECT i.first_name, i.last_name, COUNT(c.contract_id) AS total_contracts, SUM(c.pay)
AS total_earnings
FROM influencer i
JOIN contract c ON i.influencer_id = c.influencer_id
GROUP BY i.first_name, i.last_name
HAVING SUM(c.pay) > 10000;
```

--Q9: Use IN clause to select data from one or more tables (5 points)

```
SELECT p.product_name, p.category, p.price
FROM product p
WHERE p.brand_id IN (
    SELECT b.brand_id FROM brand b
    WHERE b.industry IN ('Technology', 'Fashion')
);
```

--Q10: Select length of one column from one table (use LENGTH function) (5 points)

```
SELECT first_name, LENGTH(first_name) AS name_length
FROM influencer;
```

--Q11: Delete one record from one table. Use select statements to demonstrate the table contents before and after the DELETE statement.

-- Make sure you use ROLLBACK afterwards so that the data will not be physically removed (5 points)

```
SELECT * FROM influencer;
-- Delete related records from the contract table
DELETE FROM contract
WHERE influencer_id = 101;
-- Delete the influencer record
SELECT * FROM content_creation;
BEGIN;
DELETE FROM content_creation
WHERE post_id = 10031;
-- View Table Contents After Delete
SELECT * FROM content_creation;
-- Rollback the Transaction
ROLLBACK;
-- Verify Table Contents After Rollback
SELECT * FROM content_creation;
```

--Q12: Update one record from one table. Use select statements to demonstrate the table contents before and after the UPDATE statement.

-- Make sure you use ROLLBACK afterwards so that the data will not be physically removed (5 points)

-- Q12: Update a record and demonstrate using ROLLBACK

-- View Table Contents Before Update

## PROJECT TECHNICAL REPORT

```

SELECT * FROM influencer;
-- Start a Transaction
BEGIN;
-- Update a Record
UPDATE influencer
SET platform = 'TikTok'
WHERE influencer_id = 101;
-- View Table Contents After Update
SELECT * FROM influencer;
-- Rollback the Transaction
ROLLBACK;
-- Verify Table Contents After Rollback
SELECT * FROM influencer;

```

--Q13: Classify each influencer by type (nano, micro, macro, or celebrity) based on their follower count.

```

SELECT
    i.influencer_id,
    i.first_name,
    i.last_name,
    CASE
        WHEN follower_count >= 1000000 THEN 'Celebrity'
        WHEN follower_count >= 100000 THEN 'Macro'
        WHEN follower_count >= 10000 THEN 'Micro'
        ELSE 'Nano'
    END AS influencer_type
FROM
    influencer i ;

```

--Q14: Identify the Top 3 Most Profitable Products per Brand

```

SELECT
    b.brand_name,
    p.product_name,
    SUM(pm.revenue) AS total_revenue
FROM
    product p
JOIN
    performance_metrics pm ON p.product_id = pm.product_id
JOIN
    brand b ON p.brand_id = b.brand_id
GROUP BY
    b.brand_name, p.product_name
HAVING
    SUM(pm.revenue) > 0
ORDER BY
    b.brand_name, total_revenue DESC
LIMIT 3;

```

## PROJECT TECHNICAL REPORT

## Query Output

549 --Q1: Select all columns and all rows from one table (5 points)  
 550 **SELECT \* FROM content\_creation;**  
 551

Data Output Messages Notifications

	post_id [PK] integer	influencer_id integer	product_id integer	content_type character varying (50)	date_posted date	discount_link character varying (255)	total_post_likes integer	total_post_shares integer	total_link_clicks integer
1	10031	101	4501	Photo	2024-01-15	https://www.techgadgets.com/smartphone-x-101	1000	150	200
2	10032	102	4502	Long Form Video	2024-02-20	https://www.fashionhub.com/designer-dress-102	2000	250	300
3	10033	102	4502	Short Form Video	2024-03-30	https://www.fashionhub.com/designer-dress-102	2000	250	300
4	10034	103	4503	Short Form Video	2024-03-25	https://www.healthplus.com/vitamin-supplements-...	1500	200	250
5	10035	104	4504	Photo	2024-04-30	https://www.autodrive.com/electric-car-model-s-104	2500	300	400
6	10036	105	4505	Short Form Video	2024-05-10	https://www.foodies.com/organic-coffee-105	1800	220	280
7	10037	105	4505	Short Form Video	2024-07-13	https://www.foodies.com/organic-coffee-105	1600	250	300
8	10038	105	4505	Short Form Video	2024-08-26	https://www.foodies.com/organic-coffee-105	2000	300	430
9	10039	103	4501	Short Form Video	2024-07-13	https://www.techgadgets.com/smartphone-x-103	1700	130	150
10	10040	104	4502	Photo	2024-08-20	https://www.fashionhub.com/designer-dress-104	5000	430	640

Total rows: 16 of 16 Query complete 00:00:00.107 Ln 550, Col 1

Data Output Messages Notifications

	post_id integer	influencer_id integer	product_id integer	content_type character varying (50)	date_posted date	discount_link character varying (255)	total_post_likes integer	total_post_shares integer	total_link_clicks integer
7	10037	105	4505	Short Form Video	2024-07-13	https://www.foodies.com/organic-coffee-105	1600	250	300
8	10038	105	4505	Short Form Video	2024-08-26	https://www.foodies.com/organic-coffee-105	2000	300	430
9	10039	103	4501	Short Form Video	2024-07-13	https://www.techgadgets.com/smartphone-x-103	1700	130	150
10	10040	104	4502	Photo	2024-08-20	https://www.fashionhub.com/designer-dress-104	5000	430	640
11	10041	105	4503	Long Form Video	2024-09-25	https://www.healthplus.com/vitamin-supplements-...	5700	400	1700
12	10042	101	4504	Photo	2024-05-24	https://www.autodrive.com/electric-car-model-s-101	1050	100	540
13	10043	101	4504	Short Form Video	2024-07-24	https://www.autodrive.com/electric-car-model-s-101	2070	240	1005
14	10044	102	4505	Long Form Video	2024-05-10	https://www.foodies.com/organic-coffee-102	3100	420	570
15	10045	102	4505	Short Form Video	2024-07-10	https://www.foodies.com/organic-coffee-102	2300	260	700
16	10046	101	4501	Short Form Video	2024-01-15	https://www.techgadgets.com/smartphone-x-101	1300	210	370

Total rows: 16 of 16 Query complete 00:00:00.107 Ln 550, Col 1

552 --Q2: Select five columns and all rows from one table (5 points)  
 553 **SELECT product\_id, product\_name, category, price, manufacturer\_country**  
 554 **FROM product;**  
 555

Data Output Messages Notifications

	product_id [PK] integer	product_name character varying (50)	category character varying (50)	price numeric (7,2)	manufacturer_country character varying (50)
1	4501	Smartphone X	Electronics	999.99	USA
2	4502	Designer Dress	Apparel	199.99	Italy
3	4503	Vitamin Supplements	Health	29.99	Germany
4	4504	Electric Car Model S	Vehicles	49999.99	USA
5	4505	Organic Coffee	Beverages	15.99	Colombia

Total rows: 5 of 5 Query complete 00:00:00.101 Ln 553, Col 1

## PROJECT TECHNICAL REPORT

```

556 --Q3: Select all columns from all rows from one view (5 points)
557 SELECT * FROM ProductInfo;
558

```

Data Output Messages Notifications

	brand_name character varying (50)	product_id integer	product_name character varying (50)	category character varying (50)	price numeric (7,2)	rating numeric (2,1)
1	TechGadgets	4501	Smartphone X	Electronics	999.99	4.5
2	FashionHub	4502	Designer Dress	Apparel	199.99	4.7
3	HealthPlus	4503	Vitamin Supplements	Health	29.99	4.3
4	AutoDrive	4504	Electric Car Model S	Vehicles	49999.99	4.8
5	Foodies	4505	Organic Coffee	Beverages	15.99	4.6

Total rows: 5 of 5 Query complete 00:00:00.084 Ln 557, Col 1

```

559 --Q4: Using a join on 2 tables, select all columns and all rows from the tables without the use of a Cartesian product (5 points)
560 SELECT * FROM product p
561 JOIN brand b
562 ON p.brand_id = b.brand_id ;

```

Data Output Messages Notifications

	product_id integer	product_name character varying (50)	category character varying (50)	price numeric (7,2)	brand_id integer	manufacturer_country character varying (50)	rating numeric (2,1)	brand_id integer	brand_name character varying (50)	industry character varying (50)
1	4501	Smartphone X	Electronics	999.99	101	USA	4.5	101	TechGadgets	Technology
2	4502	Designer Dress	Apparel	199.99	102	Italy	4.7	102	FashionHub	Fashion
3	4503	Vitamin Supplements	Health	29.99	103	Germany	4.3	103	HealthPlus	Healthcare
4	4504	Electric Car Model S	Vehicles	49999.99	104	USA	4.8	104	AutoDrive	Automotive
5	4505	Organic Coffee	Beverages	15.99	105	Colombia	4.6	105	Foodies	Food & Beverage

Total rows: 5 of 5 Query complete 00:00:00.072 Ln 560, Col 1

```

559 --Q4: Using a join on 2 tables, select all columns and all rows from the tables without the use of a Cartesian product (5 points)
560 SELECT * FROM product p
561 JOIN brand b
562 ON p.brand_id = b.brand_id ;

```

Data Output Messages Notifications

	price numeric (7,2)	brand_id integer	manufacturer_country character varying (50)	rating numeric (2,1)	brand_id integer	brand_name character varying (50)	industry character varying (50)	contact_info character varying (50)	website character varying (100)
1	999.99	101	USA	4.5	101	TechGadgets	Technology	contact@techgadgets.com	https://www.techgadgets.com
2	199.99	102	Italy	4.7	102	FashionHub	Fashion	info@fashionhub.com	https://www.fashionhub.com
3	29.99	103	Germany	4.3	103	HealthPlus	Healthcare	support@healthplus.com	https://www.healthplus.com
4	49999.99	104	USA	4.8	104	AutoDrive	Automotive	sales@autodrive.com	https://www.autodrive.com
5	15.99	105	Colombia	4.6	105	Foodies	Food & Beverage	hello@foodies.com	https://www.foodies.com

Total rows: 5 of 5 Query complete 00:00:00.072 Ln 560, Col 1



## PROJECT TECHNICAL REPORT

```

564 --Q5: Select and order data retrieved from one table (5 points)
565 SELECT * FROM contract
566 ORDER BY pay DESC ;
567

```

Data Output Messages Notifications

	contract_id [PK] integer	brand_id integer	influencer_id integer	start_date date	end_date date	pay numeric (7,2)	signed_date date
1	3108	103	105	2024-07-01	2024-12-31	9000.00	2024-06-01
2	3104	104	104	2024-04-01	2024-09-30	8000.00	2024-03-25
3	3107	102	104	2024-06-01	2024-11-30	7500.00	2024-05-15
4	3106	101	103	2024-06-01	2024-11-30	7500.00	2024-05-15
5	3110	105	102	2024-02-01	2024-05-31	7000.00	2024-01-20
6	3102	102	102	2024-02-01	2024-07-31	7000.00	2024-01-10
7	3103	103	103	2024-03-01	2024-08-31	6000.00	2024-02-20
8	3105	105	105	2024-05-01	2024-10-31	5500.00	2024-04-15

Total rows: 8 of 8 Query complete 00:00:00.096 Ln 565, Col 1

```

568 --Q6: Using a join on 3 tables, select 5 columns from the 3 tables. Use syntax that would limit the output to 3 rows (5 points)
569 SELECT c.contract_id, i.first_name, i.last_name, p.product_name, p.price
570 FROM contract c
571 JOIN influencer i ON c.influencer_id = i.influencer_id
572 JOIN product p ON c.brand_id = p.brand_id
573 LIMIT 3;

```

Data Output Messages Notifications

	contractId integer	first_name character varying (25)	last_name character varying (25)	product_name character varying (50)	price numeric (7,2)
1	3106	Charlie	Williams	Smartphone X	999.99
2	3102	Bob	Johnson	Designer Dress	199.99
3	3107	Diana	Brown	Designer Dress	199.99

Total rows: 3 of 3 Query complete 00:00:00.078 Ln 569, Col 1

```

575 --Q7: Select distinct rows using joins on 3 tables (5 points)
576 SELECT DISTINCT
577 i.first_name, i.last_name, p.product_name, b.brand_name, c.pay
578 FROM contract c
579 JOIN influencer i ON c.influencer_id = i.influencer_id
580 JOIN product p ON c.brand_id = p.brand_id
581 JOIN brand b ON p.brand_id = b.brand_id;

```

Data Output Messages Notifications

	first_name character varying (25)	last_name character varying (25)	product_name character varying (50)	brand_name character varying (50)	pay numeric (7,2)
1	Diana	Brown	Electric Car Model S	AutoDrive	8000.00
2	Bob	Johnson	Organic Coffee	Foodies	7000.00
3	Ethan	Davis	Organic Coffee	Foodies	5500.00
4	Diana	Brown	Designer Dress	FashionHub	7500.00
5	Bob	Johnson	Designer Dress	FashionHub	7000.00
6	Ethan	Davis	Vitamin Supplements	HealthPlus	9000.00
7	Charlie	Williams	Smartphone X	TechGadgets	7500.00
8	Charlie	Williams	Vitamin Supplements	HealthPlus	6000.00

Total rows: 8 of 8 Query complete 00:00:00.084 Ln 575, Col 1

## PROJECT TECHNICAL REPORT

```

583 --Q8: Use GROUP BY and HAVING in a select statement using one or more tables (5 points)
584 v SELECT i.first_name, i.last_name, COUNT(c.contract_id) AS total_contracts, SUM(c.pay) AS total_earnings
585 FROM influencer i
586 JOIN contract c ON i.influencer_id = c.influencer_id
587 GROUP BY i.first_name, i.last_name
588 HAVING SUM(c.pay) > 10000;
589

```

Data Output Messages Notifications

	first_name character varying (25)	last_name character varying (25)	total_contracts bigint	total_earnings numeric
1	Charlie	Williams	2	13500.00
2	Ethan	Davis	2	14500.00
3	Diana	Brown	2	15500.00
4	Bob	Johnson	2	14000.00

✓ Successfu

Total rows: 4 of 4 Query complete 00:00:00.087 Ln 584, Col 1

```

590 --Q9: Use IN clause to select data from one or more tables (5 points)
591 v SELECT p.product_name, p.category, p.price
592 FROM product p
593 WHERE p.brand_id IN (
594     SELECT b.brand_id FROM brand b
595     WHERE b.industry IN ('Technology', 'Fashion')
596 );
597

```

Data Output Messages Notifications

	product_name character varying (50)	category character varying (50)	price numeric (7,2)
1	Smartphone X	Electronics	999.99
2	Designer Dress	Apparel	199.99

Total rows: 2 of 2 Query complete 00:00:00.082 Ln 591, Col 1

```

598 --Q10: Select length of one column from one table (use LENGTH function) (5 points)
599 v SELECT first_name, LENGTH(first_name) AS name_length
600 FROM influencer;
601

```

Data Output Messages Notifications

	first_name character varying (25)	name_length integer
1	Alice	5
2	Bob	3
3	Charlie	7
4	Diana	5
5	Ethan	5

Total rows: 5 of 5 Query complete 00:00:00.083 Ln 599, Col 1

## PROJECT TECHNICAL REPORT

```

602 --Q11: Delete one record from one table. Use select statements to demonstrate the table contents before and after the DELETE statement.
603 -- Make sure you use ROLLBACK afterwards so that the data will not be physically removed (5 points)
604 SELECT * FROM influencer;
605 -- Delete related records from the contract table
606 DELETE FROM contract
607 WHERE influencer_id = 101;
608 -- Delete the influencer record
609 SELECT * FROM content_creation;
610 BEGIN;
611 DELETE FROM content_creation
612 WHERE post_id = 10031;
613 -- View Table Contents After Delete
614 SELECT * FROM content_creation;
615 -- Rollback the Transaction
616 ROLLBACK;
617 -- Verify Table Contents After Rollback
618 SELECT * FROM content_creation;
619

```

Query History

```

608 -- Delete the influencer record
609 SELECT * FROM content_creation;
610 BEGIN;
611 DELETE FROM content_creation
612 WHERE post_id = 10031;
613 -- View Table Contents After Delete
614 SELECT * FROM content_creation;
615 -- Rollback the Transaction
616 ROLLBACK;
617 -- Verify Table Contents After Rollback
618 SELECT * FROM content_creation;

```

Data Output Messages Notifications

	post_id [PK] integer	influencer_id integer	product_id integer	content_type character varying (50)	date_posted date	discount_link character varying (255)	total_post_likes integer	total_post_shares integer	total_link_clicks integer
1	10032	102	4502	Long Form Video	2024-02-20	https://www.fashionhub.com/designer-dress-102	2000	250	300
2	10033	102	4502	Short Form Video	2024-03-30	https://www.fashionhub.com/designer-dress-102	2000	250	300
3	10034	103	4503	Short Form Video	2024-03-25	https://www.healthplus.com/vitamin-supplements-...	1500	200	250
4	10035	104	4504	Photo	2024-04-30	https://www.autodrive.com/electric-car-model-s-104	2500	300	400
5	10036	105	4505	Short Form Video	2024-05-10	https://www.foodies.com/organic-coffee-105	1800	220	280
6	10037	105	4505	Short Form Video	2024-07-13	https://www.foodies.com/organic-coffee-105	1600	250	300
7	10038	105	4505	Short Form Video	2024-08-26	https://www.foodies.com/organic-coffee-105	2000	300	430
8	10039	103	4501	Short Form Video	2024-07-13	https://www.techgadgets.com/smartphone-x-103	1700	130	150
9	10040	104	4502	Photo	2024-08-20	https://www.fashionhub.com/designer-dress-104	5000	430	640
10	10041	105	4503	Long Form Video	2024-09-25	https://www.healthplus.com/vitamin-supplements-...	5700	400	1700

Total rows: 15 of 15 Query complete 00:00:00.088 Ln 614, Col 1

```

615 -- Rollback the Transaction
616 ROLLBACK;
617 -- Verify Table Contents After Rollback
618 SELECT * FROM content_creation;
619

```

Data Output Messages Notifications

ROLLBACK

Query returned successfully in 100 msec.

```

617 -- Verify Table Contents After Rollback
618 SELECT * FROM content_creation;

```

Data Output Messages Notifications

	post_id [PK] integer	influencer_id integer	product_id integer	content_type character varying (50)	date_posted date	discount_link character varying (255)	total_post_likes integer	total_post_shares integer	total_link_clicks integer
1	10031	101	4501	Photo	2024-01-15	https://www.techgadgets.com/smartphone-x-101	1000	150	200
2	10032	102	4502	Long Form Video	2024-02-20	https://www.fashionhub.com/designer-dress-102	2000	250	300
3	10033	102	4502	Short Form Video	2024-03-30	https://www.fashionhub.com/designer-dress-102	2000	250	300
4	10034	103	4503	Short Form Video	2024-03-25	https://www.healthplus.com/vitamin-supplements-...	1500	200	250
5	10035	104	4504	Photo	2024-04-30	https://www.autodrive.com/electric-car-model-s-104	2500	300	400
6	10036	105	4505	Short Form Video	2024-05-10	https://www.foodies.com/organic-coffee-105	1800	220	280
7	10037	105	4505	Short Form Video	2024-07-13	https://www.foodies.com/organic-coffee-105	1600	250	300
8	10038	105	4505	Short Form Video	2024-08-26	https://www.foodies.com/organic-coffee-105	2000	300	430
9	10039	103	4501	Short Form Video	2024-07-13	https://www.techgadgets.com/smartphone-x-103	1700	130	150
10	10040	104	4502	Photo	2024-08-20	https://www.fashionhub.com/designer-dress-104	5000	430	640

Total rows: 16 of 16 Query complete 00:00:00.099 Ln 618, Col 1

## PROJECT TECHNICAL REPORT

Query

Query History

625

SELECT \* FROM influencer;

626

-- Start a Transaction

627

BEGIN;

628

-- Update a Record

629

UPDATE influencer

630

SET platform = 'TikTok'

631

WHERE influencer\_id = 101;

632

-- View Table Contents After Update

633

SELECT \* FROM influencer;

634

-- Rollback the Transaction

635

ROLLBACK;

Data Output

Messages

Notifications

UPDATE 1

Query returned successfully in 74 msec.

Total rows: 16 of 16

Query complete 00:00:00.074

Ln 631, Col 27

632

-- View Table Contents After Update

633

SELECT \* FROM influencer;

634

-- Rollback the Transaction

635

ROLLBACK;

Data Output

Messages

Notifications

SQL

	<div><div>influencer_id</div><div>[PK] integer</div></div>	<div><div>first_name</div><div>character varying (25)</div></div>	<div><div>last_name</div><div>character varying (25)</div></div>	<div><div>platform</div><div>character varying (25)</div></div>	<div><div>follower_count</div><div>integer</div></div>	<div><div>email</div><div>character varying (50)</div></div>
1	102	Bob	Johnson	YouTube	150000	bob.johnson@gmail.com
2	103	Charlie	Williams	TikTok	75000	charlie.williams@yahoo.com
3	104	Diana	Brown	Instagram	120000	diana.brown@gmail.com
4	105	Ethan	Davis	YouTube	200000	ethan.davis@gmail.com
5	101	Alice	Smith	TikTok	50000	alice.smith@hotmail.com

Total rows: 5 of 5

Query complete 00:00:00.091

Ln 633, Col 1

## PROJECT TECHNICAL REPORT

```

634 -- Rollback the Transaction
635 ROLLBACK;
636 -- Verify Table Contents After Rollback
637 SELECT * FROM influencer;
638

```

Data Output Messages Notifications

ROLLBACK

Query returned successfully in 71 msec.

```

636 -- Verify Table Contents After Rollback
637 SELECT * FROM influencer;
638

```

Data Output Messages Notifications

	influencer_id [PK] integer	first_name character varying (25)	last_name character varying (25)	platform character varying (25)	follower_count integer	email character varying (50)
1	101	Alice	Smith	Instagram	50000	alice.smith@hotmail.com
2	102	Bob	Johnson	YouTube	150000	bob.johnson@gmail.com
3	103	Charlie	Williams	TikTok	75000	charlie.williams@yahoo.com
4	104	Diana	Brown	Instagram	120000	diana.brown@gmail.com
5	105	Ethan	Davis	YouTube	200000	ethan.davis@gmail.com

✓ Successfully run. Total qu

Total rows: 5 of 5 Query complete 00:00:00.088 Ln 637, Col 1

```

639 --Q13: Classify each influencer by type (nano, micro, macro, or celebrity) based on their follower count.
640 SELECT
641     i.influencer_id,
642     i.first_name,
643     i.last_name,
644     CASE
645         WHEN follower_count >= 1000000 THEN 'Celebrity'
646         WHEN follower_count >= 100000 THEN 'Macro'
647         WHEN follower_count >= 10000 THEN 'Micro'
648         ELSE 'Nano'
649     END AS influencer_type
650 FROM
651     influencer i ;
652

```

Data Output Messages Notifications

	influencer_id [PK] integer	first_name character varying (25)	last_name character varying (25)	influencer_type text
1	101	Alice	Smith	Micro
2	102	Bob	Johnson	Macro
3	103	Charlie	Williams	Micro
4	104	Diana	Brown	Macro
5	105	Ethan	Davis	Macro

Total rows: 5 of 5 Query complete 00:00:00.111 Ln 651, Col 16

## PROJECT TECHNICAL REPORT

```

653 --Q14: Identify the Top 3 Most Profitable Products per Brand
654 SELECT
655     b.brand_name,
656     p.product_name,
657     SUM(pm.revenue) AS total_revenue
658 FROM
659     product p
660 JOIN
661     performance_metrics pm ON p.product_id = pm.product_id
662 JOIN
663     brand b ON p.brand_id = b.brand_id
664 GROUP BY
665     b.brand_name, p.product_name
666 HAVING
667     SUM(pm.revenue) > 0
668 ORDER BY
669     b.brand_name, total_revenue DESC
670 LIMIT 3;
671

```

brand_name	product_name	total_revenue
AutoDrive	Electric Car Model S	1499999.70
FashionHub	Designer Dress	14999.25
Foodies	Organic Coffee	4637.10

Total rows: 3 of 3    Query complete 00:00:00.081    Ln 654, Col 1

## CONCLUSION

The Influencer Partnership Management System will provide businesses with a centralized platform to manage influencer collaborations, track campaign performance, and optimize marketing strategies. By integrating data analytics, performance metrics, and contract management, the system aims to streamline the influencer selection process, improve ROI measurement, and ensure efficient tracking of influencer-related data. This system addresses the challenges of decentralized data and manual processes, offering a scalable and customizable solution that helps businesses make informed, data-driven decisions, ultimately enhancing the effectiveness of influencer marketing campaigns.

## REFERENCES

- Freberg, K., Graham, K., McGaughey, K., & Freberg, L. A. (2011). *Who are the social media influencers? A study of public perceptions of personality*. Public Relations Review, 37(1), 90-92. <https://doi.org/10.1016/j.pubrev.2010.11.002>
- Gretzel, U. (2018). *Influencer marketing and its impact on tourism*. Journal of Tourism Futures, 4(3), 213-225. <https://doi.org/10.1108/JTF-07-2018-0072>
- Hughes, C. S., Swaminathan, V., & Brooks, C. M. (2019). *The role of influencer marketing in advertising: Perspectives of social media influencers and brand managers*. Journal of Advertising Research, 59(3), 263-277. <https://doi.org/10.2501/JAR-2019-030>
- Kapitan, S., & Silvera, D. H. (2016). *The role of social influence in influencer marketing*. Journal of Consumer Psychology, 26(4), 437-448. <https://doi.org/10.1016/j.jcps.2015.12.002>
- Chaffey, D., & Ellis-Chadwick, F. (2019). *Digital marketing: Strategy, implementation, and practice* (7th ed.). Pearson Education.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). *Database system concepts* (6th ed.). McGraw-Hill.
- Lanz, A., Pappas, I. O., & Sivarajah, U. (2022). *Automation in influencer marketing: A systematic review of tools and platforms*. Journal of Business Research, 140, 529-547. <https://doi.org/10.1016/j.jbusres.2021.10.016>
- Lou, C., & Yuan, S. (2019). *Influencer marketing: Implications for business and research*. Journal of Advertising, 48(2), 131-144. <https://doi.org/10.1080/00913367.2019.1608958>