University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 02

Date: 2025/07/16

Group Number: 4

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Hejia Qiu	55761555	hqiu09	q9month@163.com
Alice Peng	95587275	apeng09	alice.peng129@gmail.com
Yinuo Sun (Preferred: Enora Sun)	83070482	ysun137	enorasun1120@gmail.com

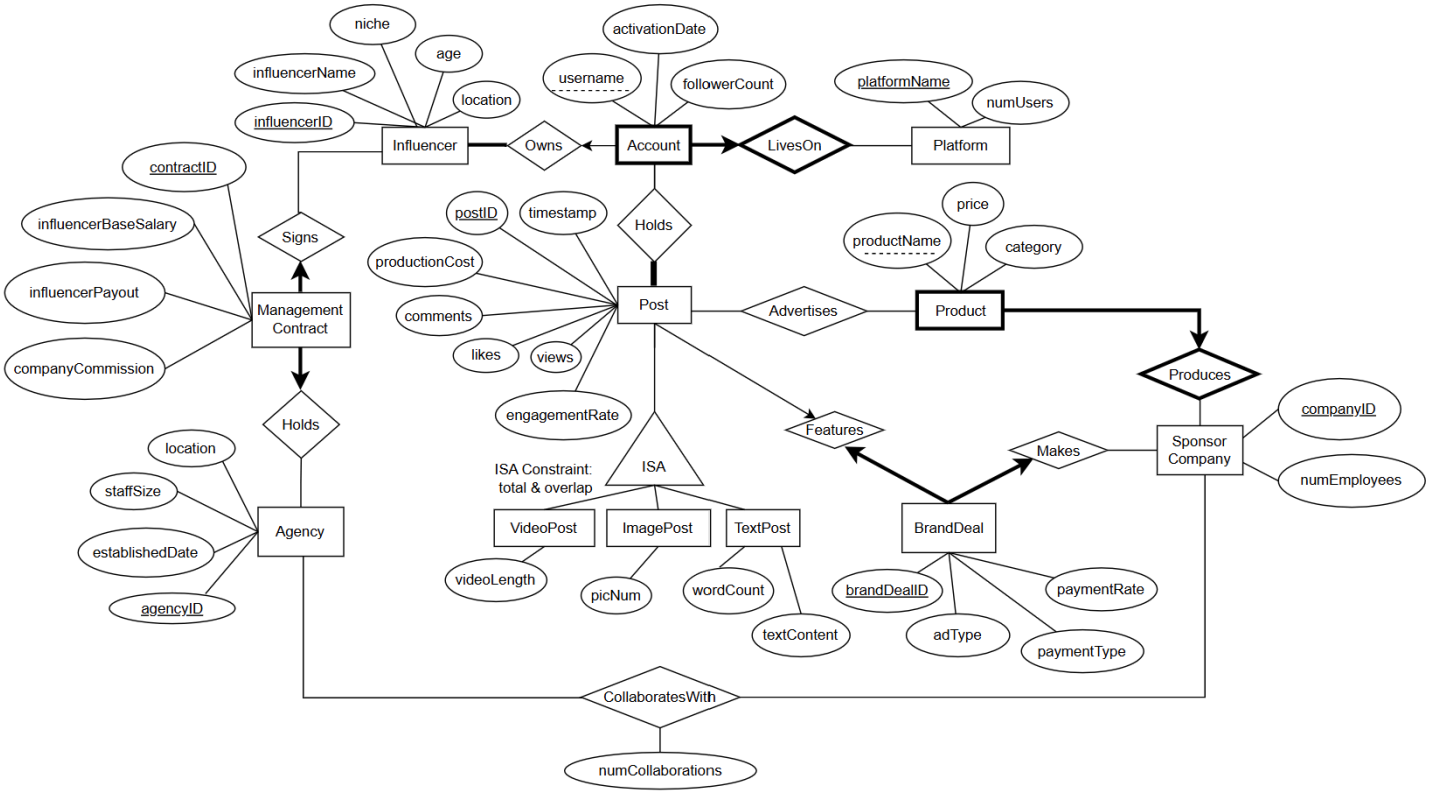
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project summary

Our project focuses on the data related to influencers on social media and their collaboration with sponsors and agencies. The application enables people to view, manipulate, and analyze data over brand deals, social media posts, influencers, etc.

ER Diagram



Differences in ER diagram compared to milestone 01 submission (mainly adding additional attributes and updating some names)

- Changed entity titles Video and Image to VideoPost and ImagePost to make the name more informative and consistent (to match the TextPost entity)
- Added “niche” attribute for influencer (which means the domain that the influencers is mainly working on) since we found that to be an interesting attribute to have
- Changed the relationship name between account and platform from “Holds” to “LivesOn” to make it more informative
- We edited and added attributes to ManagementContract to introduce more interesting functional dependencies: influencerBaseSalary, influencerPayout (percentage of brand

deal revenue that the influencer gets), CompanyCommision (percentage of brand deal revenue that the agency gets)

- We added “comments” (number of comments) and “engagement rate” attributes to introduce another interesting functional dependency (engagementRate = (likes + comments)/views)
- We changed the relationship between Agency and Sponsor company from “NegotiatesWith” to “CollaboratesWith”, and changes the attribute to number of collaborations, as we want that relationship to focus on how many times an agency has collaborated with a sponsor company
- We added “adType” (such as affiliate link, sponsored mentioning, etc.) attribute for BrandDeal entity, and updated the attribute title “rate” to “paymentRate” to make it more informative
- We removed the descriptive attribute “type” from relationship Advertises because we decided advertisement type would be determined in the BrandDeal

Relational model (Before Any Normalization)

*primary key attributes are underlined, and foreign key attributes are bolded

Influencer(<u>influencerID</u>: int, influencerName: varchar(50), location: varchar(100), age: int, niche: varchar(50))

- Primary key: {influencerID}
- Foreign key: N/A
- Candidate key excluding primary key: N/A
- Other constraints: N/A

Account(<u>username</u>: varchar(100), **<u>platformName</u>**: varchar(50), **influencerID**: int, followerCount: int, activationDate: date)

- Primary key: {username, platformName}
- Foreign key:
  - platformName, which references to platformName in relation “Platform”
  - influencerID, which references to influencerID in relation “Influencer”
- Candidate key excluding primary key: N/A
- Other constraints: N/A

*total participation on Influencer in Influencer-Owns-Account relationship cannot be represented without assertions

*Relation’s name is just Account instead of Account + LivesOn + Owns because we think Account is already informative enough on what the relation is about

Post(<u>postID</u>: int, timeStamp: timestamp, productionCost: money, views: int, likes: int, comments: int, engagementRate: float)

- Primary key: {postID}
- Foreign key: N/A
- Candidate key excluding primary key: N/A

- Other constraints: N/A

AccountHoldsPost(**postID**: int, **username**: varchar(100), **platformName**: varchar(50))
- Primary key: {postID, username, platformName}
- Foreign key:
    - postID, which references to postID in relation “Post”
    - {username, platformName} which references to {username, platformName} in relation “Account”
- Candidate key excluding primary key: N/A
- Other constraints: N/A

*AccountHoldsPost is used as the name instead of just Holds to be more descriptive
*total participation on Post in Account-Holds-Post relationship cannot be represented without assertions

VideoPost(**postID**: int, videoLength: float)
- Primary key: {postID}
- Foreign key:
    - postID, which references to postID in relation “Post”
- Candidate key excluding primary key: N/A
- Other constraints: N/A

ImagePost(**postID**: int, picNum: int)
- Primary key: {postID}
- Foreign key:
    - postID, which references to postID in relation “Post”
- Candidate key excluding primary key: N/A
- Other constraints: N/A

TextPost(**postID**: int, wordCount: int, textContent: text)
- Primary key: {postID}
- Foreign key:
    - postID, which references to postID in relation “Post”
- Candidate key excluding primary key: N/A
- Other constraints: N/A

Platform(platformName: varchar(50), numUsers: bigint)
- Primary key: {platformName}
- Foreign key: N/A
- Candidate key excluding primary key: N/A
- Other constraints: N/A

SponsorCompany(companyID: int, numEmployees: int)
- Primary key: {companyID}
- Foreign key: N/A

- Candidate key excluding primary key: N/A
- Other constraints: N/A

Product(productName: varchar(100), **companyID**: int, price: money, category: varchar(50))
- Primary key: {companyID, productName}
- Foreign key:
    - companyID, which references to the companyID in relation “SponsorCompany”
- Candidate key excluding primary key: N/A
- Other constraints: N/A

*Relation’s name is just Product instead of Product + produces because we think Product is already informative enough on what the relation is about

BrandDeal(brandDealID: int, adType: varchar(50), paymentType: varchar(50), paymentRate: money, **companyID**: int, **postID**: int)
- Primary key: { brandDealID }
- Foreign key:
    - companyID, which references to the companyID in relation “SponsorCompany”
    - postID, which references to the postID in relation “Post”
- Candidate key excluding primary key: postID
- Other constraints:
    - companyID needs to be not null
    - postID needs to be not null AND unique

* Relation’s name is just BrandDeal instead of BrandDeal+ Makes + Features because we think BrandDeal is already informative enough on what the relation is about

Agency(agencyID: int, establishedDate: date, staffSize: int, location: varchar(100))
- Primary key: { agencyID }
- Foreign key: N/A
- Candidate key excluding primary key: N/A
- Other constraints: N/A

ManagementContract(contractID: int, influencerBaseSalary: money, influencerPayout: float, companyCommission: float, **influencerID**: int, **agencyID**: int)
- Primary key: { contractID }
- Foreign key:
    - agencyID, which references to the agencyID in relation “Agency”
    - influencerID, which references to the influencerID in relation “Influencer”
- Candidate key excluding primary key: N/A
- Other constraints:
    - influencerID needs to be not null
    - agencyID needs to be no null

* Relation’s name is just ManagementContract instead of ManagementContract + Signs+ Holds because we think ManagementContract is already informative enough on what the relation is about

CollaboratesWith(**agencyID**: int, **companyID**: int, numCollaborations: int)

- Primary key: { agencyID, companyID}
- Foreign key:
    - agencyID, which references to the agencyID in relation “Agency”
    - companyID, which references to the companyID in relation “SponsorCompany”
- Candidate key excluding primary key: N/A
- Other constraints: N/A

Advertise(**productName**: varchar(100), **companyID**: int, **postID**: int)

- Primary key: { productName, companyID, postID}
- Foreign key:
    - {productName, companyID} which references {productName, companyID} in relation “Product”
    - postID, which references to the postID in relation “Post”
- Candidate key excluding primary key: N/A
- Other constraints: N/A

Functional Dependencies

*Valid FDs other than those identified by a PK or CK are in blue.

Influencer

- influencerID → influencerName
- influencerID → location
- influencerID → age
- influencerID → niche

Account

- username, platformName → influencerID
- username, platformName → followerCount
- username, platformName → activationDate

Post

- postID → timeStamp
- postID → productionCost
- postID → views
- postID → likes
- postID → comments
- postID → engagementRate
- comments, likes, view → engagementRate
    - We define engagement rate as (comments + likes)/view

VideoPost
- postID → videoLength

ImagePost
- postID → picNum

TextPost
- postID → wordCount, textContent

Platform
- platformName → numUsers

SponsorCompany
- companyID → numEmployees

Product
- productName, companyID → price, category

BrandDeal
- brandDealID → adType
- brandDealID → paymentType
- brandDealID → paymentRate
- brandDealID → companyID
- brandDealID → postID
- postID → brandDealID
- adType → paymentType
    - We made a simplifying assumption that 'Sponsored Post', 'Shoutout', 'Giveaway Collaboration' types of advertisements would be paid out as a flat fee; 'Review' and 'Product Placement' types would have per view payment type

Agency:
- agencyID → establishedDate
- agencyID → staffSize
- agencyID → location

ManagementContract
- contractID → influencerBaseSalary
- contractID → influencerPayout
- contractID → companyCommission
- contractID → influencerID
- contractID → agencyID
- influencerPayout → companyCommission
- companyCommission → influencerPayout

    - influencerPayout is the percentage that influencer will get from the revenue they gain from a brand deal; companyCommision is the the percentage that the agency will get from the revenue they gain from a brand deal

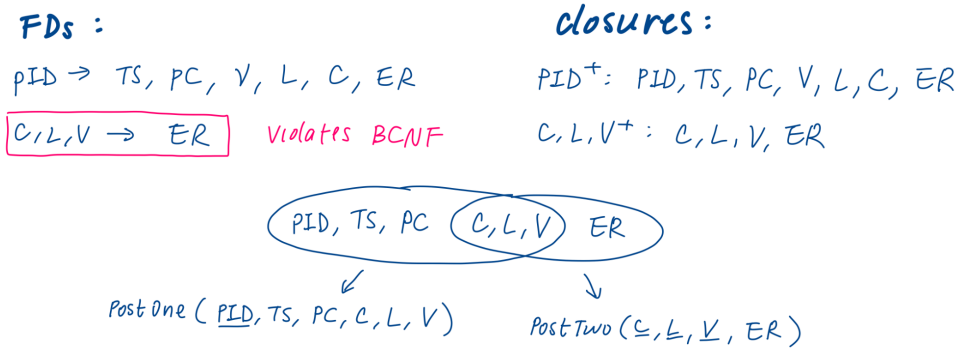
CollaboratesWith

- agencyID, companyID → numCollaborations

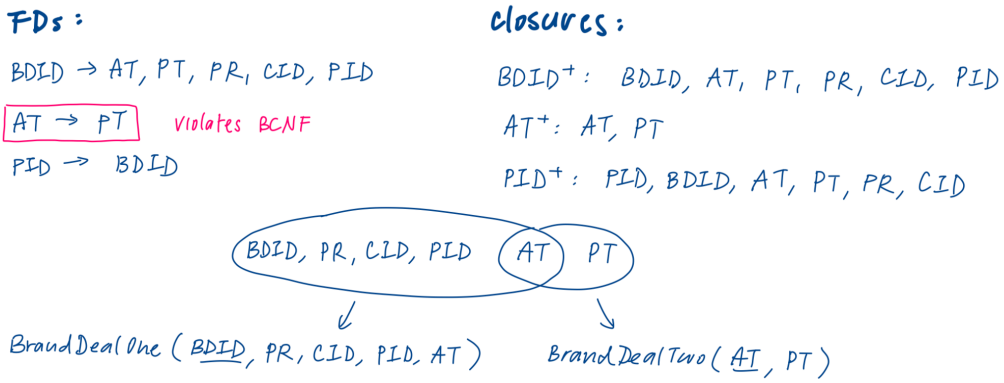
Normalization

We chose to decompose each of our tables to be in BCNF.

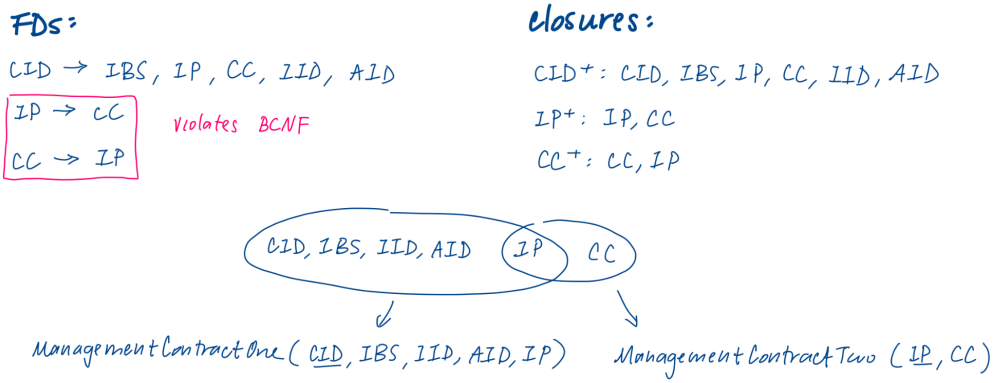
1. Post(<u>postID</u>: int, timeStamp: timestamp, productionCost: money, views: int, likes: int, comments: int, engagementRate: float)
    - Attribute names shortened for brevity, in the same order as above: Post(PID, TS, PC, V, L, C, ER)
    - Decomposing on (C, L, V → ER)



2. BrandDeal(<u>brandDealID</u>: int, adType: varchar(50), paymentType: varchar(50), paymentRate: money, **companyID**: int, **postID**: int)
    - Attribute names shortened for brevity, in the same order as above: BrandDeal(BDID, AT, PT, PR, CID, PID)
    - Decomposing on (AT → PT)



3. ManagementContract(<u>contractID</u>: int, influencerBaseSalary: money, influencerPayout: float, companyCommission: float, **influencerID**: int, **agencyID**: int)
    - Attribute names shortened for brevity, in the same order as above: ManagementContract(CID, IBS, IP, CC, IID, AID)
    - Decomposing on (IP → CC)



Post-Normalization Schemas

*The new schemas post-normalization are the ones in blue.

Influencer(<u>influencerID</u>: int, influencerName: varchar(50), location: varchar(100), age: int, niche: varchar(50))

- Primary key: {influencerID}

Account(<u>username</u>: varchar(100), **<u>platformName</u>**: varchar(50), **influencerID**: int, followerCount: int, activationDate: date)

- Primary key: {username, platformName}
- Foreign key:
    - platformName references platformName in Platform
    - influencerID references influencerID in Influencer

*total participation on Influencer in Influencer-Owns-Account relationship cannot be represented without assertions

PostOne(<u>postID</u>: int, timeStamp: timestamp, productionCost: money, **views**: int, **likes**: int, **comments**: int)

- Primary key: {postID}
- Foreign key:
    - { views, likes, comments } references { views, likes, comments } in PostTwo

PostTwo(<u>views</u>: int, <u>likes</u>: int, <u>comments</u>: int, engagementRate: float)

- Primary key: { views, likes, comments }

AccountHoldsPost(**postID**: int, **username**: varchar(100), **platformName**: varchar(50))
- Primary key: { postID, username, platformName }
- Foreign key:
    - postID references postID in PostOne
    - { username, platformName } references { username, platformName } in Account

*total participation on Post in Account-Holds-Post relationship cannot be represented without assertions

VideoPost(**postID**: int, videoLength: float)
- Primary key: {postID}
- Foreign key:
    - postID references postID in PostOne

ImagePost(**postID**: int, picNum: int)
- Primary key: {postID}
- Foreign key:
    - postID references postID in PostOne

TextPost(**postID**: int, wordCount: int, textContent: text)
- Primary key: {postID}
- Foreign key:
    - postID references postID in PostOne

Platform(platformName: varchar(50), numUsers: bigint)
- Primary key: { platformName }

SponsorCompany(companyID: int, numEmployees: int)
- Primary key: { companyID }

Product(productName: varchar(100), **companyID**: int, price: money, category: varchar(50))
- Primary key: {companyID, productName}
- Foreign key:
    - companyID references companyID in SponsorCompany

BrandDealOne(brandDealID: int, **adType**: varchar(50), paymentRate: money, **companyID**: int, **postID**: int)
- Primary key: { brandDealID }
- Foreign key:
    - companyID references companyID in SponsorCompany
    - postID references postID in PostOne
    - adType references adType in BrandDealTwo
- Candidate key excluding primary key: postID

- Other constraints:
    - companyID is not null
    - postID is not null AND unique

BrandDealTwo(adType: varchar(50), paymentType: varchar(50))
- Primary key: { adType }

Agency(agencyID: int, establishedDate: date, staffSize: int, location: varchar(100))
- Primary key: { agencyID }

ManagementContractOne(contractID: int, influencerBaseSalary: money, **influencerPayout**: float, **influencerID**: int, **agencyID**: int)
- Primary key: { contractID }
- Foreign key:
    - agencyID, which references to the agencyID in relation “Agency”
    - influencerID, which references to the influencerID in relation “Influencer”
    - influencerPayout references influencerPayout in ManagementContractTwo
- Candidate key excluding primary key: N/A
- Other constraints:
    - influencerID needs to be not null
    - agencyID needs to be no null

ManagementContractTwo(influencerPayout: float, companyCommission: float)
- Primary key: { influencerPayout }

CollaboratesWith(**agencyID**: int, **companyID**: int, numCollaborations: int)
- Primary key: { agencyID, companyID}
- Foreign key:
    - agencyID references agencyID in Agency
    - companyID, which references to the companyID in relation “SponsorCompany”

Advertise(**productName**: varchar(100), **companyID**: int, **postID**: int)
- Primary key: { productName, companyID, postID}
- Foreign key:
    - { productName, companyID } references { productName, companyID } in Product
    - postID references postID in PostOne

SQL for Table Creation

```
CREATE TABLE Influencer (
        influencerID INT PRIMARY KEY,
```

```
    influencerName VARCHAR(50),
    location VARCHAR(100),
    age INT,
    niche VARCHAR(50)
);

CREATE TABLE Account (
    username VARCHAR(100),
    platformName VARCHAR(50),
    influencerID INT,
    followerCount INT,
    activationDate DATE,
    PRIMARY KEY (username, platformName),
    FOREIGN KEY (platformName) REFERENCES Platform(platformName),
    FOREIGN KEY (influencerID) REFERENCES Influencer(influencerID)
);
```

*total participation on Influencer in Influencer-Owns-Account relationship cannot be represented without assertions

```
CREATE TABLE PostOne (
    postID INT PRIMARY KEY,
    timeStamp TIMESTAMP,
    productionCost MONEY,
    views INT,
    likes INT,
    comments INT,
    FOREIGN KEY (views, likes, comments) REFERENCES PostTwo(views, likes, comments)
);

CREATE TABLE PostTwo (
    views INT,
    likes INT,
    comments INT,
    engagementRate FLOAT,
    PRIMARY KEY (views, likes, comments)
);

CREATE TABLE AccountHoldsPost (
    postID INT,
    username VARCHAR(100),
    platformName VARCHAR(50),
    PRIMARY KEY (postID, username, platformName),
    FOREIGN KEY (postID) REFERENCES PostOne(postID),
    FOREIGN KEY (username, platformName) REFERENCES Account(username, platformName)
```

```
);
```
*total participation on Post in Account-Holds-Post relationship cannot be represented without assertions

```
CREATE TABLE VideoPost (
        postID INT PRIMARY KEY,
        videoLength FLOAT,
        FOREIGN KEY (postID) REFERENCES PostOne(postID)
);

CREATE TABLE ImagePost (
        postID INT PRIMARY KEY,
        picNum INT,
        FOREIGN KEY (postID) REFERENCES PostOne(postID)
);

CREATE TABLE TextPost (
        postID INT PRIMARY KEY,
        wordCount INT,
        textContent TEXT,
        FOREIGN KEY (postID) REFERENCES PostOne(postID)
);

CREATE TABLE Platform (
        platformName VARCHAR(50) PRIMARY KEY,
        numUsers BIGINT
);

CREATE TABLE SponsorCompany (
        companyID INT PRIMARY KEY,
        numEmployees INT
);

CREATE TABLE Product (
        productName VARCHAR(100),
        companyID INT,
        price MONEY,
        category VARCHAR(50),
        PRIMARY KEY (companyID, productName),
        FOREIGN KEY (companyID) REFERENCES SponsorCompany(companyID)
);

CREATE TABLE BrandDealOne (
        brandDealID INT PRIMARY KEY,
```

```
        adType VARCHAR(50),
        paymentRate MONEY,
        companyID INT NOT NULL,
        postID INT NOT NULL UNIQUE,
        FOREIGN KEY (companyID) REFERENCES SponsorCompany(companyID),
        FOREIGN KEY (postID) REFERENCES PostOne(postID),
        FOREIGN KEY (adType) REFERENCES BrandDealTwo(adType)
);

CREATE TABLE BrandDealTwo (
        adType VARCHAR(50) PRIMARY KEY,
        paymentType VARCHAR(50),
);

CREATE TABLE Agency (
        agencyID INT PRIMARY KEY,
        establishedDate DATE,
        staffSize INT,
        location VARCHAR(100)
);

CREATE TABLE ManagementContractOne (
        contractID INT PRIMARY KEY,
        influencerBaseSalary MONEY,
        influencerPayout FLOAT,
        influencerID INT NOT NULL,
        agencyID INT NOT NULL,
        FOREIGN KEY (agencyID) REFERENCES Agency(agencyID),
        FOREIGN KEY (influencerID) REFERENCES Influencer(influencerID),
        FOREIGN KEY (influencerPayout) REFERENCES ManagementContractTwo(influencerPayout)
);

CREATE TABLE ManagementContractTwo (
        influencerPayout FLOAT PRIMARY KEY,
        companyCommission FLOAT,
);

CREATE TABLE CollaboratesWith(
        agencyID INT,
        companyID INT,
        numCollaborations INT,
        PRIMARY KEY (agencyID, companyID),
        FOREIGN KEY (agencyID) REFERENCES Agency(agencyID),
        FOREIGN KEY (companyID) REFERENCES SponsorCompany(companyID)
```

```
);

CREATE TABLE Advertise (
        productName VARCHAR(100),
        companyID INT,
        postID INT,
        PRIMARY KEY (productName, companyID, postID),
        FOREIGN KEY (companyID, productName) REFERENCES Product(companyID, productName),
        FOREIGN KEY (postID) REFERENCES PostOne(postID)
);
```

SQL Insert Statement

```
INSERT INTO Influencer (influencerID, influencerName, location, age, niche) VALUES
        (1, 'Addison Rae', 'USA', 24, 'Music'),
        (2, 'Kylie Jenner', 'USA', 27, 'Beauty'),
        (3, 'Kendall Jenner', 'USA', 29, 'Wine'),
        (4, 'Kim Kardashian', 'USA', 44, 'TV'),
        (5, 'Shawn Mendes', 'CAN', 26, 'Music'),
        (6, 'Justin Bieber', 'CAN', 31, 'Music');

INSERT INTO Account (username, platformName, influencerID, followerCount, activationDate)
VALUES
        ('addrae', 'Tiktok', 1, 240000, 2025-01-01),
        ('kyljen', 'Instagram', 2, 21700, 2013-07-21'),
        ('kenjen', 'YouTube', 3, 2230, 2015-01-01),
        ('kenjen', 'Instagram', 3, 286000, 2011-07-02),
        ('shawn', 'Instagram', 5, 357, 2016-10-09),
        ('justinx', 'Twitter', 6, 294000, 2011-07-01),
        ('jennerxky', 'Instagram', 2, 294000, 2011-07-01);,
        ('kimxkim', 'Instagram', 4, 294000, 2011-07-01);

INSERT INTO PostOne (postID, timeStamp, productionCost, views, likes, comments) VALUES
        (1,  '2023-10-02 16:24:56', 102.30, 24, 4, 1),
        (2,  '2023-01-02 06:27:46', 302.40, 2804, 1000, 15),
        (3,  '2022-02-02 17:24:36', 10.00, 100004, 3894, 1233),
        (4,  '2020-01-10 11:04:50', 1099.00, 12, 1, 0),
        (5,  '2023-12-22 19:16:07', 80.00, 9238, 2399, 123),
        (6,  '2022-06-14 09:30:00', 150.00, 502, 120, 15),
        (7,  '2022-07-01 11:45:20', 250.00, 745, 230, 34),
        (8,  '2022-08-12 13:15:40', 95.00, 823, 300, 45),
        (9,  '2022-09-20 17:00:00', 180.00, 1200, 450, 56),
        (10, '2022-10-05 19:30:10', 210.00, 1350, 520, 62),
```

```
    (11, '2022-11-15 12:10:00', 70.00, 400, 150, 20),
    (12, '2022-12-01 14:20:00', 85.00, 520, 200, 25),
    (13, '2023-01-25 15:35:00', 120.00, 600, 250, 30),
    (14, '2023-02-10 16:45:00', 140.00, 700, 280, 35),
    (15, '2023-03-05 18:00:00', 160.00, 800, 300, 40);

INSERT INTO PostTwo (views, likes, comments, engagementRate) VALUES
    (24, 4, 1, 0.2083),
    (2804, 1000, 15, 0.3620),
    (100004, 3894, 1233, 0.0513),
    (12, 1, 0, 0.0833),
    (9238, 2399, 123, 0.2730),
    (502, 120, 15, 0.2689),
    (745, 230, 34, 0.3544),
    (823, 300, 45, 0.4192),
    (1200, 450, 56, 0.4217),
    (1350, 520, 62, 0.4311),
    (400, 150, 20, 0.4250),
    (520, 200, 25, 0.4327),
    (600, 250, 30, 0.4667),
    (700, 280, 35, 0.4500),
    (800, 300, 40, 0.4250);

INSERT INTO AccountHoldsPost (postID INT, username, platformName) VALUES
    (1, 'addrae', 'Tiktok'),
    (2, 'addrae', 'Tiktok'),
    (3, 'kyljen', 'Instagram'),
    (4, 'lilbieber', 'Instagram'),
    (5, 'kenjen', 'Facebook’')
    (6, 'addrae', 'Tiktok'),
    (7, 'addrae', 'Tiktok'),
    (8, 'lilbieber', 'Instagram'),
    (9, 'kenjen', 'Instagram'),
    (10, 'kenjen', 'Facebook’')
    (11, 'justinx', 'Twitter'),
    (12, 'kimxkim', 'Twitter'),
    (13, 'justinx', 'Twitter'),
    (14, 'jennerxky', 'Twitter'),
    (15, 'jennerxky', 'Twitter');

INSERT INTO VideoPost (postID, videoLength) VALUES
    (1, 3.5),
    (2, 10.0),
    (3, 1.2),
```

```
    (4, 7.8),
    (5, 5.0);

INSERT INTO ImagePost (postID, picNum) VALUES
    (6, 3),
    (7, 1),
    (8, 5),
    (9, 2),
    (10, 4);

INSERT INTO TextPost (postID, wordCount, textContent) VALUES
    (11, 120, 'Excited to announce my new partnership!'),
    (12, 80, 'Check out my latest review of the smartwatch.'),
    (13, 200, 'Tips on healthy lifestyle and meal prep.'),
    (14, 150, 'Behind the scenes of my recent campaign.'),
    (15, 90, 'Thank you all for your support!');

INSERT INTO Platform (platformName, numUsers) VALUES
    ('Instagram', 1500000000),
    ('YouTube', 2500000000),
    ('TikTok', 1200000000),
    ('Facebook', 450000000),
    ('Twitter', 500000000);

INSERT INTO SponsorCompany (companyID, numEmployees) VALUES
    (1, 200),
    (2, 350),
    (3, 50),
    (4, 1200),
    (5, 75);

INSERT INTO Product (companyID, productName, price, category) VALUES
    (1, 'Monster Energy Drink', 2.99, 'Beverage'),
    (2, 'Smartwatch Alpha', 199.99, 'Electronics'),
    (3, 'Choco Protein Bar', 1.50, 'Food'),
    (4, 'Silent Keyboard', 79.99, 'Gaming'),
    (5, 'Dove Shampoo', 8.99, 'Personal Care');

INSERT INTO BrandDealOne (brandDealID, adType, paymentRate, companyID, postID) VALUES
    (1, 'Sponsored Post', 500.00, 1, 1),
    (2, 'Shoutout', 300.00, 2, 2),
    (3, 'Giveaway Collaboration', 750.00, 3, 3),
    (4, 'Review', 0.05, 4, 4),
    (5, 'Product Placement', 0.10, 5, 5);
```

```
INSERT INTO BrandDealTwo (adType, paymentType) VALUES
        ('Sponsored Post', 'Flat Fee'),
        ('Shoutout', 'Flat Fee'),
        ('Giveaway Collaboration', 'Flat Fee'),
        ('Review', 'Per View'),
        ('Product Placement', 'Per View');

INSERT INTO ManagementContractOne (contractID, influencerBaseSalary, influencerPayout,
influencerID, agencyID) VALUES
        (1, 50000.00, 0.70, 4, 1),
        (2, 60000.00, 0.75, 5, 2),
        (3, 55000.00, 0.80, 3, 3),
        (4, 70000.00, 0.85, 1, 4),
        (5, 65000.00, 0.90, 2, 5);

INSERT INTO ManagementContractTwo (influencerPayout, companyCommission) VALUES
        (0.70, 0.30),
        (0.75, 0.25),
        (0.80, 0.20),
        (0.85, 0.15),
        (0.90, 0.10);

INSERT INTO Agency (agencyID, establishedDate, staffSize, location) VALUES
        (1, '2010-05-12', 25, 'New York, USA'),
        (2, '2015-08-20', 40, 'Los Angeles, USA'),
        (3, '2008-03-15', 15, 'Toronto, Canada'),
        (4, '2020-11-01', 10, 'Vancouver, Canada'),
        (5, '2012-07-30', 30, 'London, UK');

INSERT INTO CollaboratesWith(agencyID, companyID, numCollaborations) VALUES
        (1, 1, 5),
        (1, 2, 3),
        (2, 3, 4),
        (3, 4, 2),
        (4, 5, 6);

INSERT INTO Advertise (productName, companyID, postID) VALUES
        ('Monster Energy Drink', 1, 6),
        ('Smartwatch Alpha', 2, 2),
        ('Choco Protein Bar', 3, 11),
        (''Silent Keyboard', 4, 1),
        ('Dove Shampoo', 5, 9);
```