University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 04

Date: 2025/07/26

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

Final Project Description

With a focus on social media influencer marketing, this project provided users with the ability to view, edit, and analyze data related to social media influencer economy. The primary functionality involves allowing users to add, delete, and update the values stored in the database. In addition, users would be able to filter out the data based on certain conditions so they can view only the part of the data that they are interested in. Moreover, the application provided some interesting analytical data for users to find out more information from the database.

Schema Change (if applicable)

- We changed any data type that is MONEY, TEXT, and BIGINT as they are not build-in data types in Oracle, the changes are as followings based on data type matching documentation of oracle:
    - MONEY -> NUMBER(19,4)
    - TEXT -> LONG
    - BIGINT -> NUMBER(20)
    - Documentation reference: https://docs.oracle.com/en/database/oracle/oracle-database/18/gmswn/database-gateway-sqlserver-data-type-conversion.html
- We removed UPDATE ON CASCADE because Oracle doesn't support it

Relational Model & Instances

Note:

1. For relational model, primary key is underlined and foreign key is bolded
2. The instance screenshot is taken from a local db for better visual effect (postgreSQL - pdAdmin4). Thus, column data type still says money, text, bigint (which is equivalent to those type we updated to in Oracle)

Relational Model	Instance Screenshot

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

influencerid [PK] integer	influencername character varying (50)	location character varying (100)	age integer	niche character varying (50)
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
11	Bang Chan	AUS	28	Idol
12	Lee Minho	KOR	27	Dance
13	Seo Changbin	KOR	26	Rap
14	Hwang Hyunjin	KOR	25	Dance
15	Han Jisung	MAL	25	Rap
16	Lee Yongbok	AUS	25	Dance
17	Kim Seungmin	KOR	25	Vocal
18	Yang Jeongin	KOR	24	Vocal
21	Kim Minji	BUN	20	Idol
22	Phan Hanni	BUN	19	Idol
23	Danille Marsh	BUN	18	Idol
24	Kang Haerin	BUN	17	Idol
25	Lee Hyein	BUN	16	Idol

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

username [PK] character varying (100)	platformname [PK] character varying (50)	influencerid integer	followercount integer	activationdate date
addrae	TikTok	1	240000	2025-01-01
addraeins	Instagram	1	980000	2025-01-01
addrae	YouTube	1	240000	2024-04-15
addr	Twitter	1	9800	2015-01-01
addraeFacebook	Facebook	1	100	2019-02-01
kyljen	Instagram	2	21700	2013-07-21
jennerxky	Twitter	2	294000	2011-07-01
kenjen	Instagram	3	286000	2011-07-02
kenjen	YouTube	3	2230	2015-01-01
kimxkim	Twitter	4	294000	2011-07-01
shawn	Instagram	5	357	2016-10-09
lilbieber	Instagram	6	294000	2011-07-01
justinx	Twitter	6	294000	2013-08-01
cb1	TikTok	11	325	2018-03-25
cb2	Instagram	11	915	2019-08-06
cb3	YouTube	11	914	2014-04-25
cb4	Twitter	11	103	2020-01-03
cb5	Facebook	11	922	2021-11-20
lk1	TikTok	12	325	2018-03-25
lk2	Instagram	12	915	2019-08-06
lk3	YouTube	12	914	2014-04-25
lk4	Twitter	12	103	2020-01-03
lk5	Facebook	12	922	2021-11-20

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

postid [PK] integer	timestamp timestamp without time zone	productioncost money	views integer	likes integer	comments integer
1	2023-10-02 00:00:00	$102.30	24	4	1
2	2023-01-02 00:00:00	$302.40	2804	1000	15
3	2022-02-02 00:00:00	$10.00	100004	3894	1233
4	2020-01-10 00:00:00	$1,099.00	12	1	0
5	2023-12-22 00:00:00	$80.00	9238	2399	123
6	2022-06-14 00:00:00	$150.00	502	120	15
7	2022-07-01 00:00:00	$250.00	745	230	34
8	2022-08-12 00:00:00	$95.00	823	300	45
9	2022-09-20 00:00:00	$180.00	1200	450	56
10	2022-10-05 00:00:00	$210.00	1350	520	62
11	2022-11-15 00:00:00	$70.00	400	150	20
12	2022-12-01 00:00:00	$85.00	520	200	25
13	2023-01-25 00:00:00	$120.00	600	250	30
14	2023-02-10 00:00:00	$140.00	700	280	35
15	2023-03-05 00:00:00	$160.00	800	300	40
16	2013-03-25 00:00:00	$2,160.00	1800	520	130

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

views [PK] integer	likes [PK] integer	comments [PK] integer	engagementrate double precision
24	4	1	0.2083
2804	1000	15	0.362
100004	3894	1233	0.0513
12	1	0	0.0833
9238	2399	123	0.273
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	0.425
520	200	25	0.4327
600	250	30	0.4667
700	280	35	0.45
800	300	40	0.425
1800	520	130	0.3611

AccountHoldsPost(**<u>postID</u>**: int, **<u>username</u>**: varchar(100), **<u>platformName</u>**: varchar(50))

postid [PK] integer	username [PK] character varying (100)	platformname [PK] character varying (50)
1	addrae	TikTok
2	addrae	TikTok
16	addrae	TikTok
3	kyljen	Instagram
4	lilbieber	Instagram
5	kenjen	YouTube
6	lilbieber	Instagram
7	addraeins	Instagram
8	kenjen	Instagram
9	kenjen	Instagram
10	kenjen	Instagram
11	justinx	Twitter
12	kimxkim	Twitter
13	justinx	Twitter
14	jennerxky	Twitter
15	jennerxky	Twitter

VideoPost(**<u>postID</u>**: int, videoLength: float)

postid [PK] integer	videolength double precision
1	3.5
2	10
3	1.2
4	7.8
5	5
16	25

ImagePost(**<u>postID</u>**: int, picNum: int)

postid [PK] integer	picnum integer
6	3
7	1
8	5
9	2
10	4

TextPost(**postID**: int, wordCount: int, textContent: text)

postid [PK] integer	wordcount integer	textcontent text
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!

Platform(platformName: varchar(50), numUsers: bigint)

platformname [PK] character varying (50)	numusers bigint
Instagram	1500000000
YouTube	2500000000
TikTok	1200000000
Facebook	450000000
Twitter	500000000

SponsorCompany(companyID: int, numEmployees: int)

companyid [PK] integer	numemployees integer
1	200
2	350
3	50
4	1200
5	75

Product(productName: varchar(100), **companyID**: int, price: money, category: varchar(50))

productname [PK] character varying (100)	companyid [PK] integer	price money	category character varying (50)
Monster Energy Drink	1	$2.99	Beverage
Smartwatch Alpha	2	$199.99	Electronics
Choco Protein Bar	3	$1.50	Food
Silent Keyboard	4	$79.99	Gaming
Dove Shampoo	5	$8.99	Personal Care

BrandDealOne(brandDealID: int, **adType**: varchar(50), paymentRate: money, **companyID**: int, **postID**: int)

branddealid [PK] integer	adtype character varying (50)	paymentrate money	companyid integer	postid integer
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	16
6	Review	$0.30	1	13
7	Shoutout	$100.00	1	15

BrandDealTwo(adType: varchar(50), paymentType: varchar(50))

adtype [PK] character varying (50)	paymenttype character varying (50)
Sponsored Post	Flat Fee
Shoutout	Flat Fee
Giveaway Collaboration	Flat Fee
Review	Per View
Product Placement	Per View

Agency(agencyID: int, establishedDate: date, staffSize: int, location: varchar(100))

agencyid [PK] integer	establisheddate date	staffsize integer	location character varying (100)
1	2010-05-12	25	USA
2	2015-08-20	40	USA
3	2008-03-15	15	CAN
4	2020-11-01	10	CAN
5	2012-07-30	30	CAN

ManagementContractOne(contractID: int, influencerBaseSalary: money, **influencerPayout**: float, **influencerID**: int, **agencyID**: int)

contractid [PK] integer	influencerbasesalary money	influencerpayout double precision	influencerid integer	agencyid integer
1	$50,000.00	0.7	4	1
2	$60,000.00	0.75	5	2
3	$55,000.00	0.8	3	3
4	$70,000.00	0.85	1	4
5	$65,000.00	0.9	2	5

ManagementContractTwo(influencerPayout: float, companyCommission: float)

influencerpayout [PK] double precision	companycommission double precision
0.7	0.3
0.75	0.25
0.8	0.2
0.85	0.15
0.9	0.1

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

agencyid [PK] integer	companyid [PK] integer	numcollaborations integer
1	1	5
1	2	3
2	3	4
3	4	2
4	5	6

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

productname [PK] character varying (100)	companyid [PK] integer	postid [PK] integer
Monster Energy Drink	1	6
Smartwatch Alpha	2	2
Choco Protein Bar	3	11
Silent Keyboard	4	1
Dove Shampoo	5	16

SQL Queries & Code Position

INSERT Operation

SQL query:

```
INSERT INTO Account (username, platformName, influencerID,
followerCount, activationDate)
VALUES (:username, :platform, :influencer, :followers,
TO_DATE(:actDate, 'yyyy-mm-dd'));
```

Code Position:
appService.js line 149-150

DELETE Operation

SQL query:

```
DELETE FROM Influencer WHERE influencerID = :deleteID
```

Code Position:
appService.js line 126

UPDATE Operation

SQL query:

```
UPDATE BrandDealOne SET adType=:adType,
paymentRate=:paymentRate, companyID=:companyID, postID=:postID
WHERE brandDealID=:brandDealID
```

Code Position:
appService.js line 164

Selection AND

SQL query:

```
SELECT * FROM Influencer WHERE ${whereClauses.join(' ')}
```

Code Position:
appService.js line 194

Selection OR

SQL query:

```
SELECT * FROM Influencer WHERE ${whereClauses.join(' ')}
```

Code Position:
appService.js line 255

Projection

SQL query:

```
SELECT table_name FROM user_tables
SELECT column_name FROM USER_TAB_COLUMNS WHERE table_name
=:tableName
SELECT ${attributes} FROM ${tableName}
```

Code Position:
appService.js line 206, 216, 228

Join

Find the advertisement type of a post whose production cost is above certain threshold

SQL query:

```
SELECT BrandDealOne.adType, PostOne.productionCost
            FROM BrandDealOne, PostOne
            WHERE BrandDealOne.postID = PostOne.postID AND
PostOne.productionCost > :productionCost
```

Code Position:
appService.js line 268-270

Aggregation with group by

Find the average age of influencers on each platform

SQL query:

```
SELECT A.platformName, AVG(I.age)
```

```
                FROM Influencer I, Account A
                WHERE I.influencerID = A.influencerID
                GROUP BY A.platformName
```

Code Position:
appService.js line 283-286

Aggregation with having

Find the average post engagement rate for each category of advertised product with engagement rates above the lower bound

SQL query:

```
SELECT PR.category, AVG((P.likes + P.comments)/CAST(P.views AS
FLOAT)) AS avgEngagementRate
            FROM PostOne P, Advertise A, Product PR
            WHERE P.postID = A.postID AND A.productName =
PR.productName AND A.companyID = PR.companyID AND P.views > 0
            GROUP BY PR.category
            HAVING AVG((P.likes + P.comments)/CAST(P.views AS
FLOAT)) > :engagementRate
```

Code Position:
appService.js line 297-301

Nested aggregation with group by

Find the account that has made the most sponsored posts under brand deals

SQL query:

```
CREATE OR REPLACE VIEW NumBDsPerAccount(username, platform,
numBDs) as
                SELECT A.username, A.platformName,
COUNT(B.brandDealID) AS numBDs
                FROM BrandDealOne B, PostOne P, AccountHoldsPost
A
                WHERE B.postID = P.postID AND P.postID =
A.postID
                GROUP BY A.username, A.platformName


SELECT *
            FROM NumBDsPerAccount N
            WHERE N.numBDs = (SELECT MAX(numBDs) FROM
NumBDsPerAccount)
```

Code Position:
appService.js line 313-317, 320-322

Division

Find all IDs of influencers who have accounts on every platform

SQL query:

```
CREATE OR REPLACE VIEW influencerAccount (influencerID,
username, platformName) as
                SELECT I.influencerID, A.username,
A.platformName
                FROM Influencer I, Account A
                Where I.influencerID = A. influencerID

SELECT influencerID
                FROM influencerAccount
                GROUP BY influencerID
                HAVING COUNT (DISTINCT platformName) = (SELECT
COUNT (*)
                FROM Platform)
```

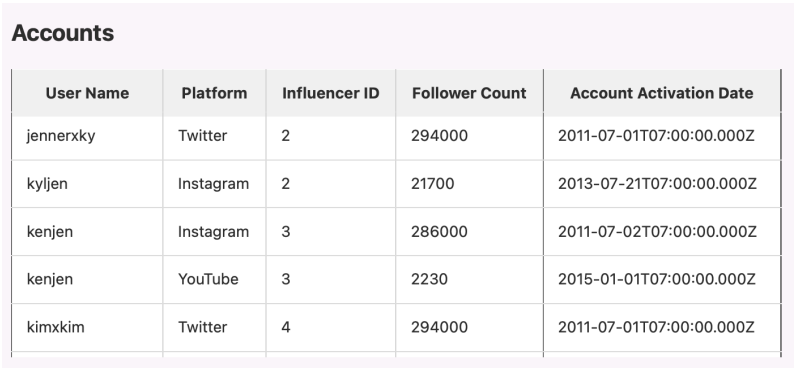
Code Position:

appService.js line 333-336, 339-343

Functionality GUI Screenshot Demo

INSERT

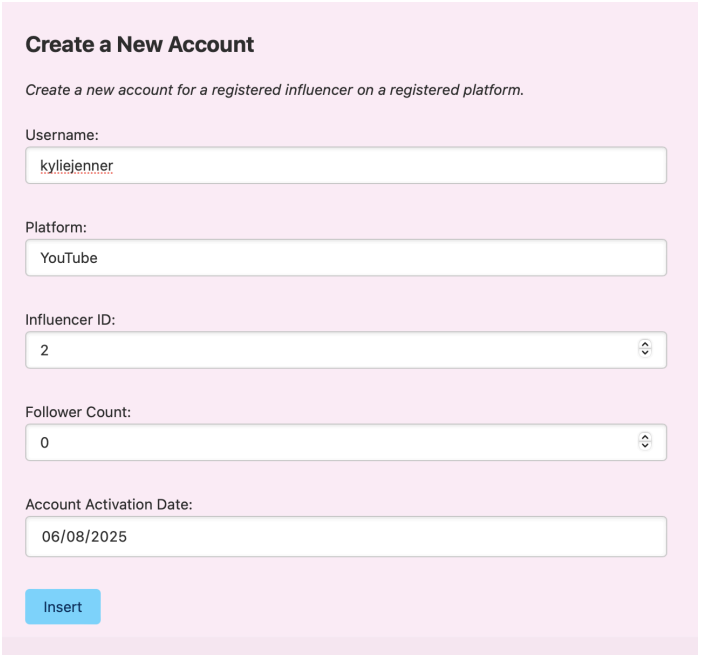
- Before:



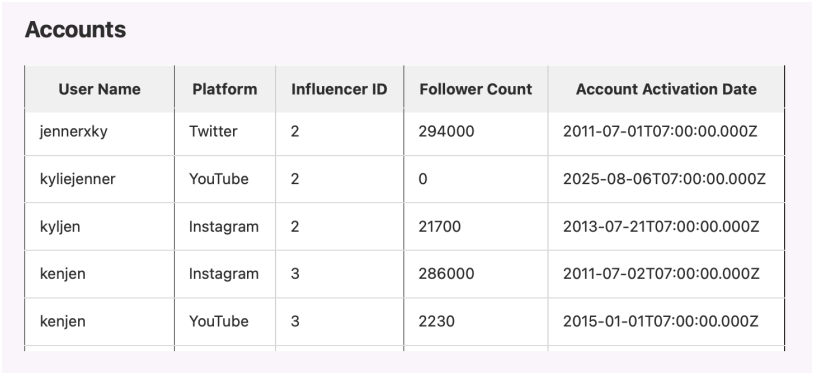
Accounts

User Name	Platform	Influencer ID	Follower Count	Account Activation Date
jennerxky	Twitter	2	294000	2011-07-01T07:00:00.000Z
kyljen	Instagram	2	21700	2013-07-21T07:00:00.000Z
kenjen	Instagram	3	286000	2011-07-02T07:00:00.000Z
kenjen	YouTube	3	2230	2015-01-01T07:00:00.000Z
kimxkim	Twitter	4	294000	2011-07-01T07:00:00.000Z

- During:



- After:



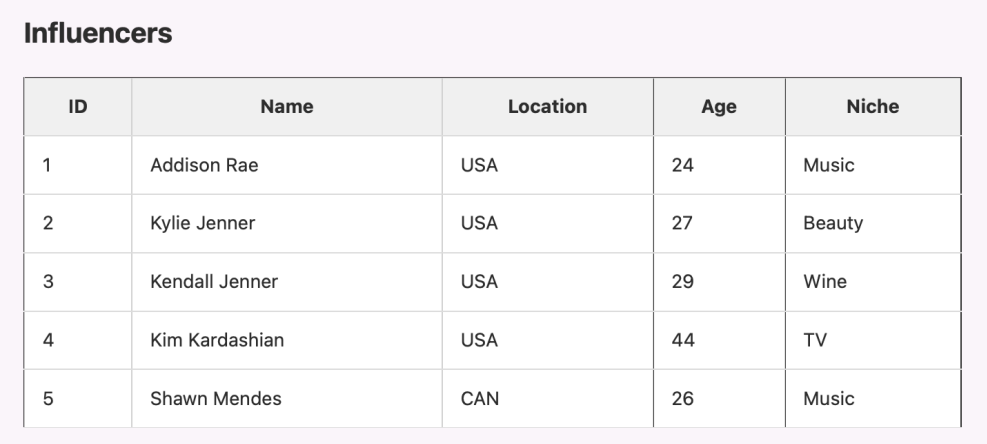
Accounts

User Name	Platform	Influencer ID	Follower Count	Account Activation Date
jennerxky	Twitter	2	294000	2011-07-01T07:00:00.000Z
kyliejenner	YouTube	2	0	2025-08-06T07:00:00.000Z
kyljen	Instagram	2	21700	2013-07-21T07:00:00.000Z
kenjen	Instagram	3	286000	2011-07-02T07:00:00.000Z
kenjen	YouTube	3	2230	2015-01-01T07:00:00.000Z

** Table is scrollable, so not all tuples are currently in frame

DELETE

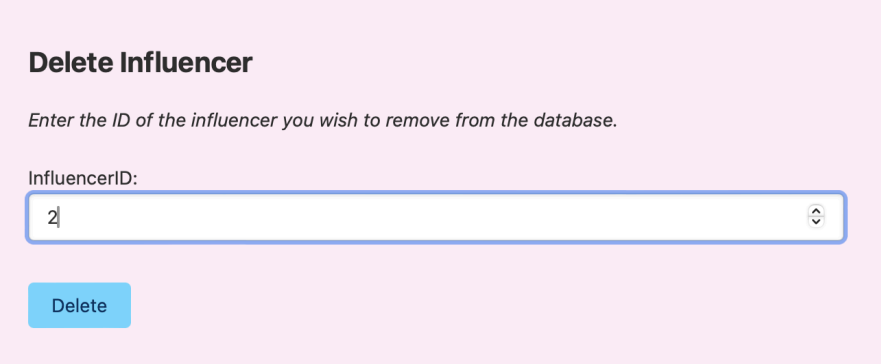
- Before:



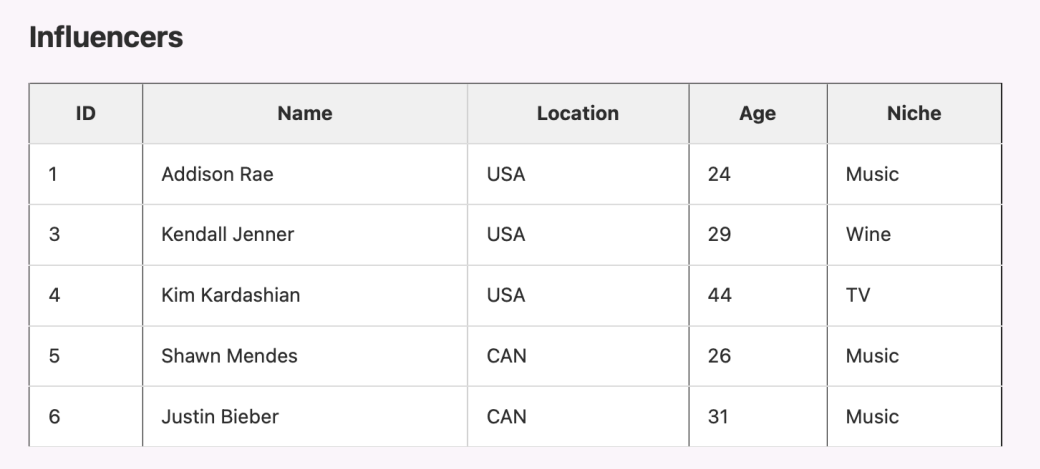
Influencers

ID	Name	Location	Age	Niche
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

- During:



- After:



Influencers

ID	Name	Location	Age	Niche
1	Addison Rae	USA	24	Music
3	Kendall Jenner	USA	29	Wine
4	Kim Kardashian	USA	44	TV
5	Shawn Mendes	CAN	26	Music
6	Justin Bieber	CAN	31	Music

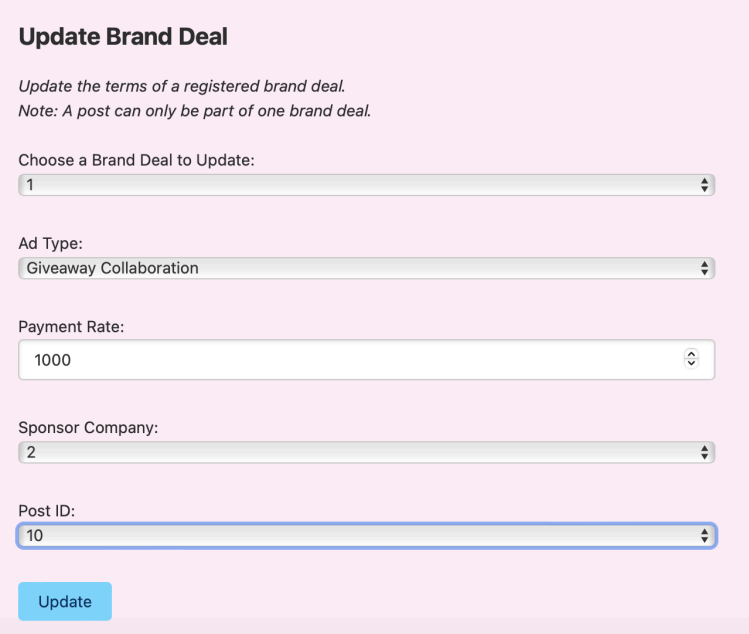
UPDATE

- Before:

Brand Deals

ID	Ad Type	Payment Rate	Company ID	Post ID
1	Sponsored Post	500	1	1
2	Shoutout	300	2	2
3	Giveaway Collaboration	750	3	3
4	Review	0.05	4	4
5	Product Placement	0.1	5	16

- During:



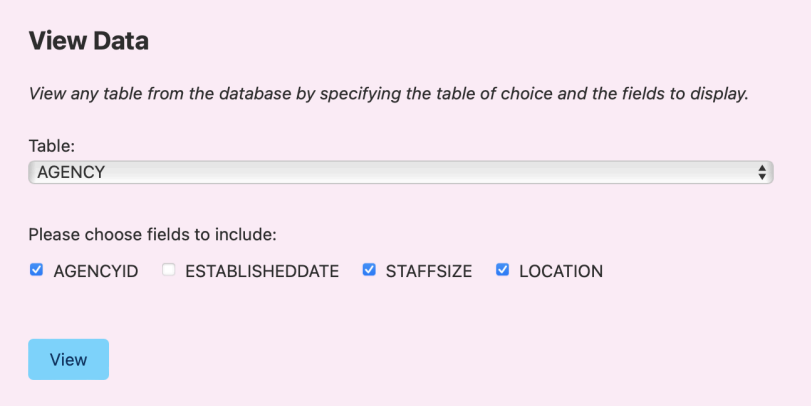
- After:

Brand Deals

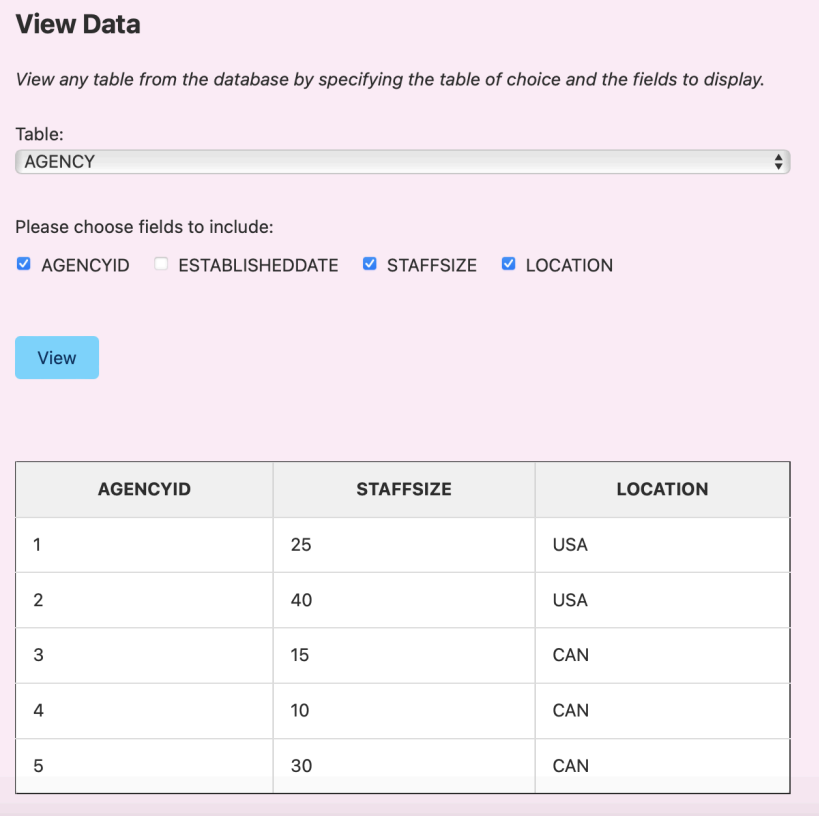
ID	Ad Type	Payment Rate	Company ID	Post ID
1	Giveaway Collaboration	1000	2	10
2	Shoutout	300	2	2
3	Giveaway Collaboration	750	3	3
4	Review	0.05	4	4
5	Product Placement	0.1	5	16

PROJECTION

- Before: no table(s)
- During:

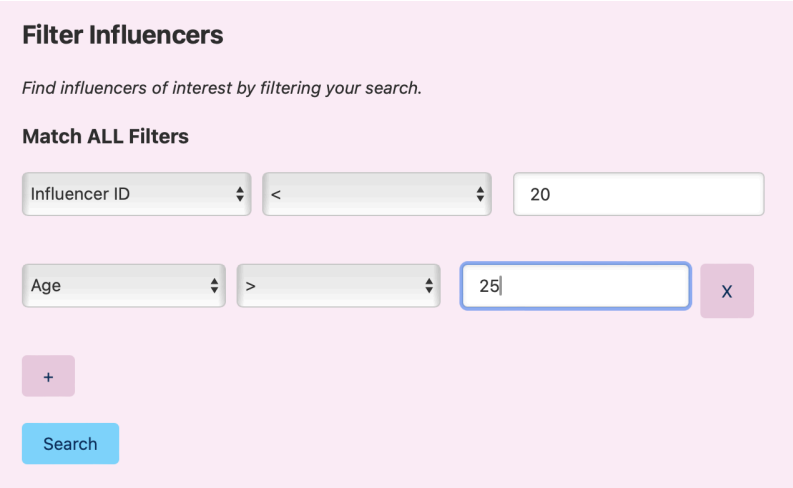


- After:

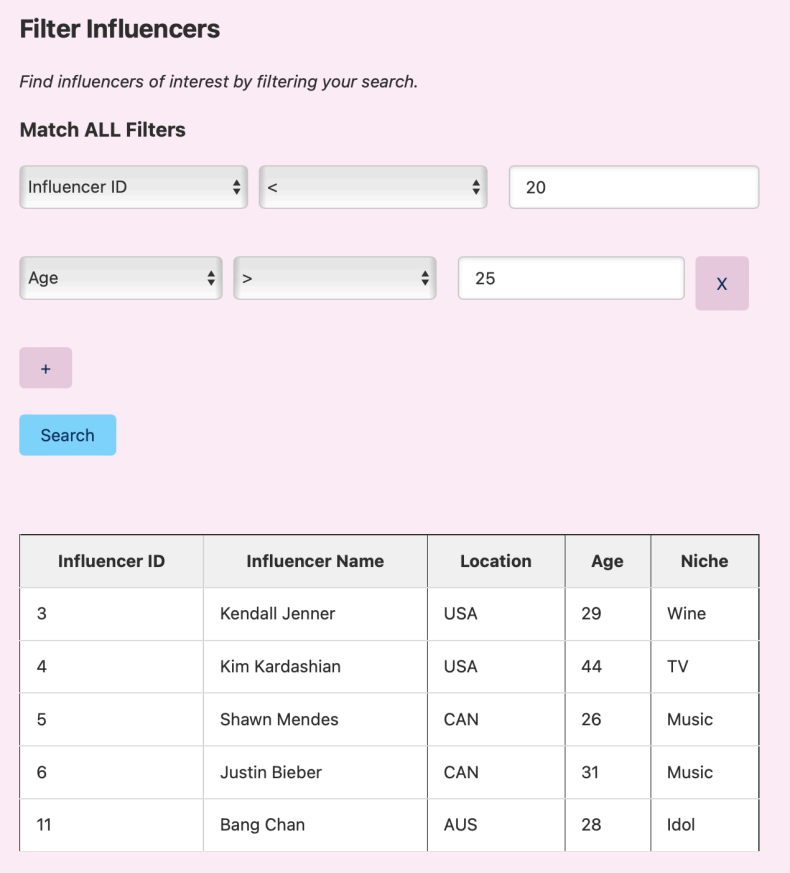


SELECTION (AND)

- Before: no table(s)
- During:



- After:



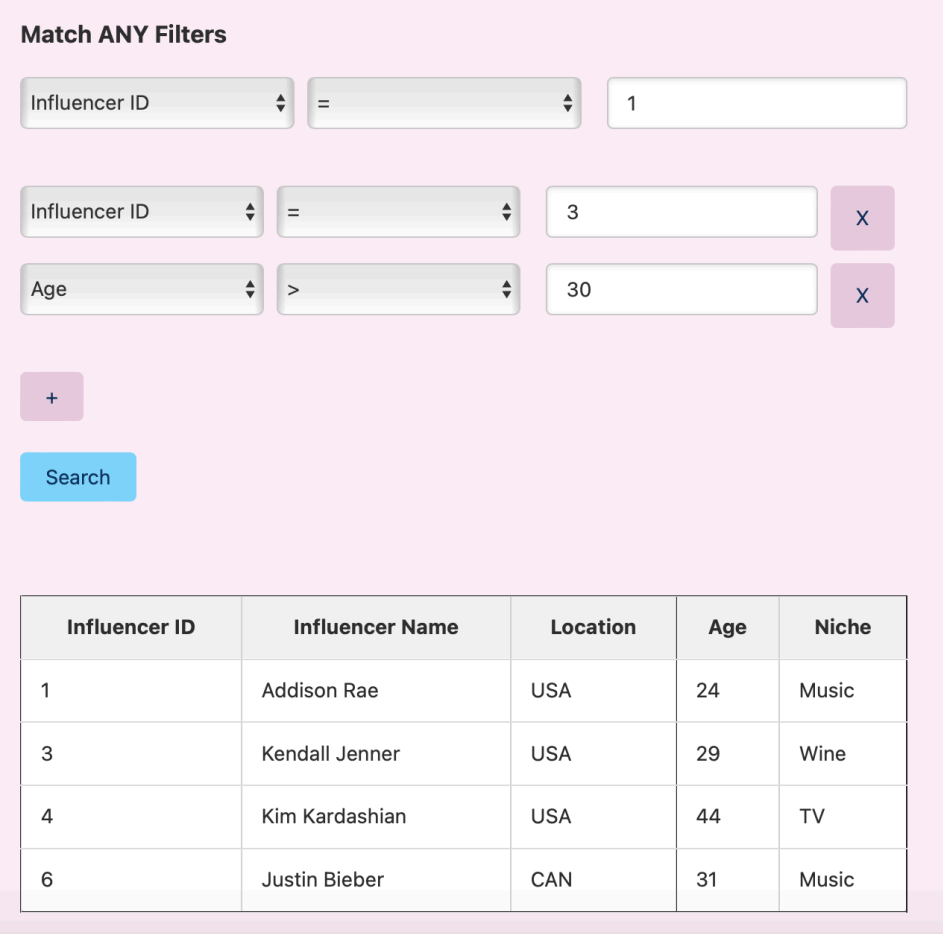
Influencer ID	Influencer Name	Location	Age	Niche
3	Kendall Jenner	USA	29	Wine
4	Kim Kardashian	USA	44	TV
5	Shawn Mendes	CAN	26	Music
6	Justin Bieber	CAN	31	Music
11	Bang Chan	AUS	28	Idol

** Table is scrollable, so not all tuples are currently in frame

SELECTION (OR)

- Before: no table(s)
- During:

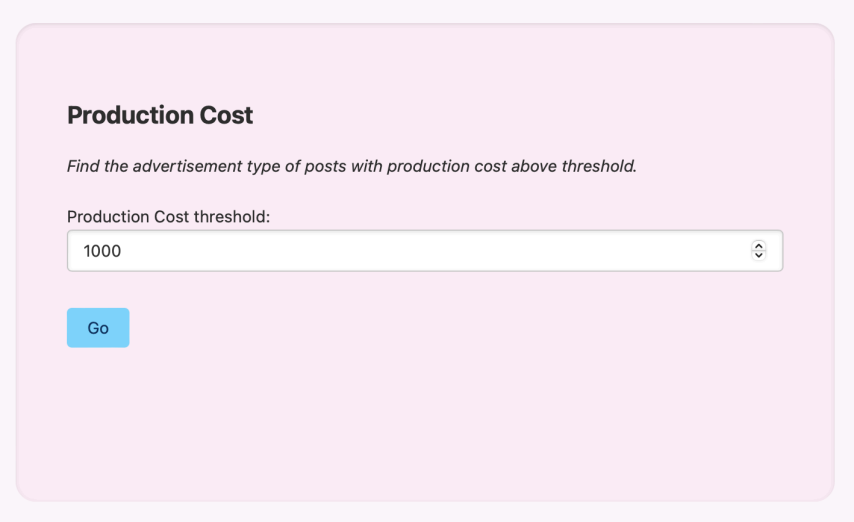
- After:



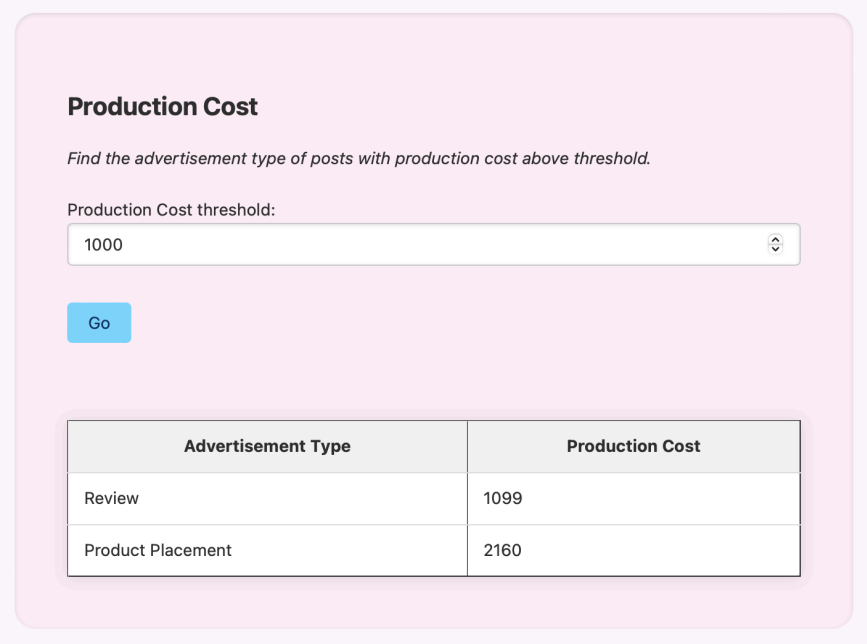
Influencer ID	Influencer Name	Location	Age	Niche
1	Addison Rae	USA	24	Music
3	Kendall Jenner	USA	29	Wine
4	Kim Kardashian	USA	44	TV
6	Justin Bieber	CAN	31	Music

JOIN

- Before: no table(s)
- During:

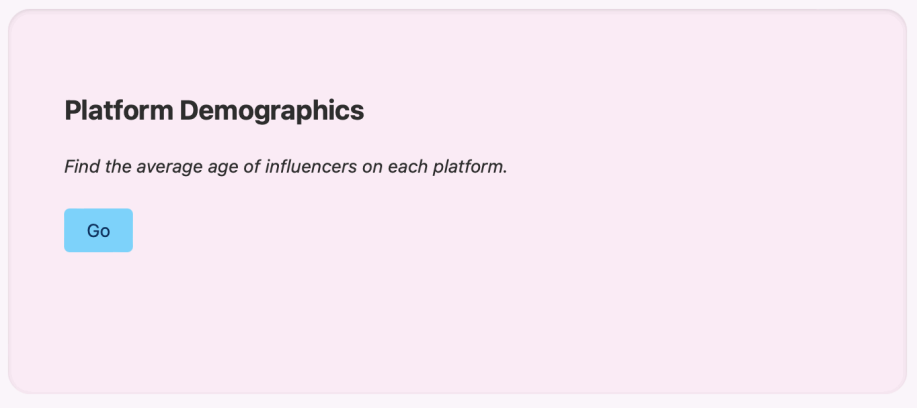


- After:

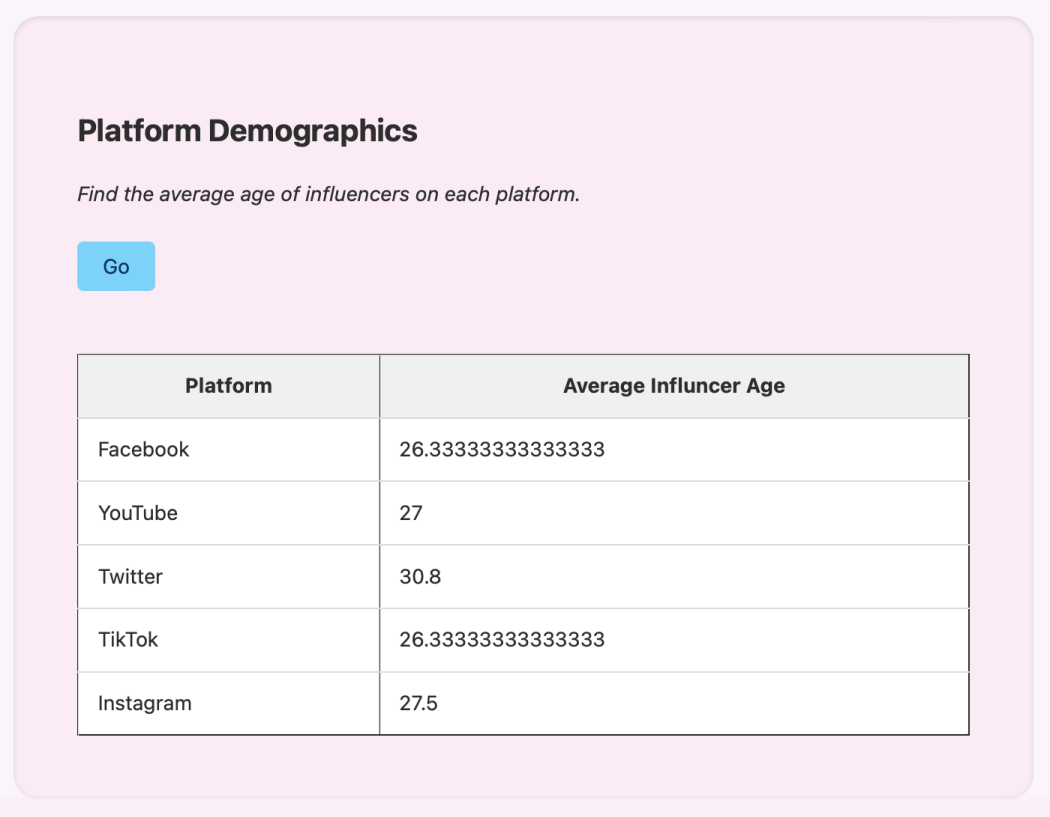


Aggregation with GROUP BY

- Before: no table(s)
- During:

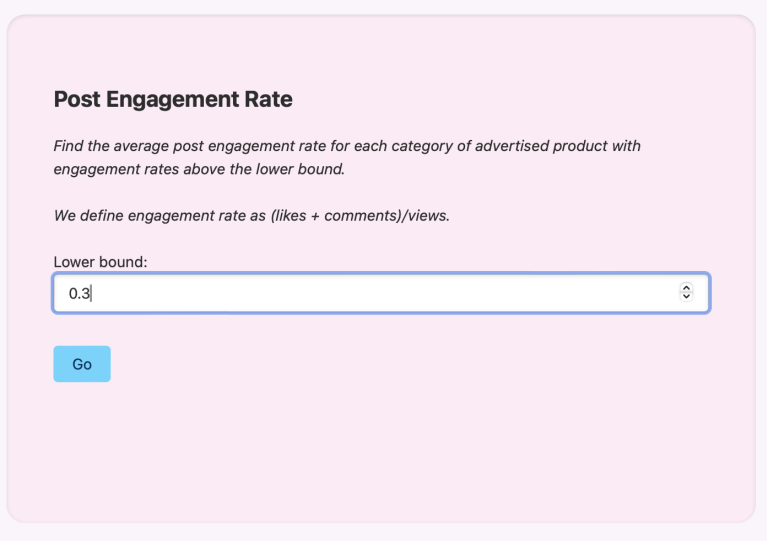


- After:

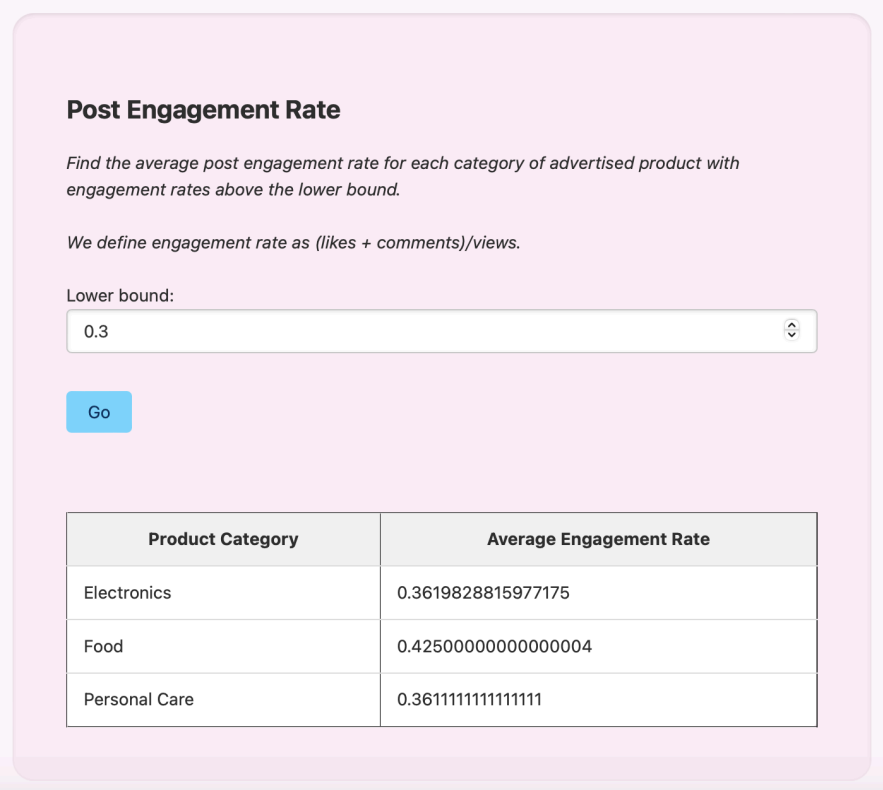


Aggregation with HAVING

- Before: no table(s)
- During:

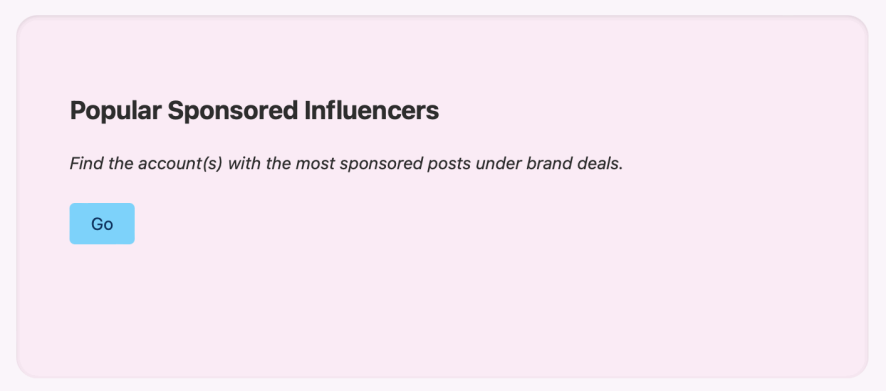


- After:

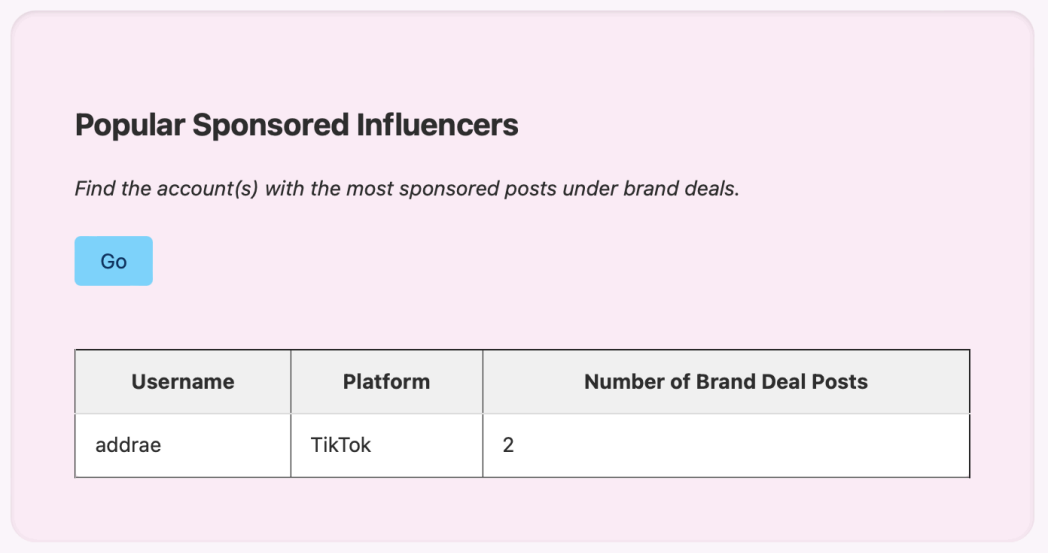


Nested aggregation with GROUP BY

- Before: no table(s)
- During:

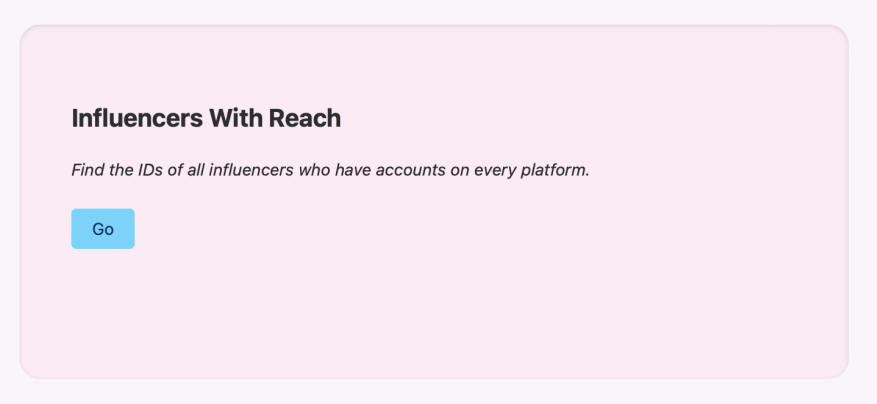


- After:



Division

- Before: no table(s)
- During:



- After: