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 provides users with the ability to view, edit, and analyze data related to social media influencer economy. The primary functionality allows users to add, delete, and update the values stored in the database. In addition, users are 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 provides 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 follows, 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 S	creensh	not						
Influencer(<u>influencerID</u> : int,	influencerid [PK] integer	influencerna character va	ime irying (50)	location character varyi	ng (100) 🖍	age integer	niche charac	eter varying (50)	
influencerName:	1	Addison Ra		USA		2	24 Music		
varchar(50), location:	2 Kylie Jenner		r	USA		2	27 Beaut	ty	
varchar(100), age: int,	3 Kendall Jenne		ner	USA		2	29 Wine		
niche: varchar(50))	4	Kim Kardas	hian	USA		4	14 TV		
	5	Shawn Men	des	CAN		2	26 Music		
	6	Justin Biebe	er	CAN		3	31 Music		
	11	Bang Chan		AUS		2	28 Idol		
	12	Lee Minho		KOR		2	27 Dance	:	
	13	Seo Changb	oin	KOR		2	26 Rap		
	14	Hwang Hyu	njin	KOR		2	25 Dance	•	
	15	Han Jisung		MAL		2	25 Rap		
	16	Lee Yongbo	k	AUS		2	25 Dance		
	17	Kim Seungn	nin	KOR		2	25 Vocal		
	18	Yang Jeong	in	KOR		2	24 Vocal		
	21	Kim Minji		BUN		2	20 Idol		
	22	Phan Hanni	ni BUN			19 Idol			
	23 Danille Marsh		sh	n BUN		1	ldol		
	24	Kang Haerir	1	BUN		1	17 Idol		
	25	Lee Hyein		BUN		1	l6 Idol		
Account(<u>username</u> :	username [PK] character var	ying (100) 🖍	platformname [PK] characte	e r varying (50)	influencerid integer	follow integer	ercount	activationdate date	
varchar(100),	addrae		TikTok			1	240000	2025-01-01	
platformName: varchar(50),	addraeins		Instagram			1	980000	2025-01-01	
influencerID: int,	addrae		YouTube			1	240000	2024-04-15	
followerCount: int,	addraaFaaabaak		Twitter			1	9800	2015-01-01	
activationDate: date)	addraeFacebook kyljen		Facebook Instagram			2	100 21700	2019-02-01	
	jennerxky		Twitter			2	294000	2013-07-21	
	kenjen		Instagram			3	286000	2011-07-02	
	kenjen		YouTube			3	2230	2015-01-01	
	kimxkim						294000	2011-07-01	
			Twitter			4	254000		
	shawn		Twitter Instagram			5	357	2016-10-09	
	shawn lilbieber							2016-10-09	
			Instagram			5	357		
	lilbieber		Instagram Instagram			5	357 294000	2011-07-01	
	lilbieber justinx		Instagram Instagram Twitter TikTok Instagram			5 6 6 11	357 294000 294000	2011-07-01 2013-08-01 2018-03-25 2019-08-06	
	lilbieber justinx cb1 cb2 cb3		Instagram Instagram Twitter TikTok Instagram YouTube			5 6 6 11 11	357 294000 294000 325 915 914	2011-07-01 2013-08-01 2018-03-25 2019-08-06 2014-04-25	
	lilbieber justinx cb1 cb2 cb3 cb4		Instagram Instagram Twitter TikTok Instagram YouTube Twitter			5 6 6 11 11 11 11	357 294000 294000 325 915 914	2011-07-01 2013-08-01 2018-03-25 2019-08-06 2014-04-25 2020-01-03	
	lilbieber justinx cb1 cb2 cb3 cb4 cb5		Instagram Instagram Twitter TikTok Instagram YouTube Twitter Facebook			5 6 6 11 11 11 11 11	357 294000 294000 325 915 914 103 922	2011-07-01 2013-08-01 2018-03-25 2019-08-06 2014-04-25 2020-01-03 2021-11-20	
	lilbieber justinx cb1 cb2 cb3 cb4 cb5		Instagram Instagram Twitter TikTok Instagram YouTube Twitter Facebook TikTok			5 6 6 11 11 11 11 11 11 11	357 294000 294000 325 915 914 103 922 325	2011-07-01 2013-08-01 2018-03-25 2019-08-06 2014-04-25 2020-01-03 2021-11-20 2018-03-25	
	lilbieber justinx cb1 cb2 cb3 cb4 cb5 lk1		Instagram Instagram Twitter TikTok Instagram YouTube Twitter Facebook TikTok Instagram			5 6 6 6 111 111 111 111 111 111 112 112 1	357 294000 294000 325 915 914 103 922 325 915	2011-07-01 2013-08-01 2018-03-25 2019-08-06 2014-04-25 2020-01-03 2021-11-20 2018-03-25 2019-08-06	
	lilbieber justinx cb1 cb2 cb3 cb4 cb5 lk1 lk2 lk3		Instagram Instagram Twitter TikTok Instagram YouTube Twitter Facebook TikTok Instagram YouTube			5 6 6 6 111 111 111 111 111 112 112 112 1	357 294000 294000 325 915 914 103 922 325 915	2011-07-01 2013-08-01 2018-03-25 2019-08-06 2014-04-25 2020-01-03 2021-11-20 2018-03-25 2019-08-06 2014-04-25	
	lilbieber justinx cb1 cb2 cb3 cb4 cb5 lk1		Instagram Instagram Twitter TikTok Instagram YouTube Twitter Facebook TikTok Instagram			5 6 6 6 111 111 111 111 111 111 112 112 1	357 294000 294000 325 915 914 103 922 325 915	2011-07-01 2013-08-01 2018-03-25 2019-08-06 2014-04-25 2020-01-03 2021-11-20 2018-03-25 2019-08-06	

PostOne(postID: 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(views: int, likes: int, comments: 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(postID : int, username :	postid [PK] integer	username [PK] character vary	ying (100) 🖍	platformname [PK] character varying (50)
varchar(100),	1	addrae		TikTok
platformName:	2	addrae		TikTok
varchar(50))	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(postID : int, picNum: int)	postid [PK] integer	, picnum integer		
	6	5 3		
	7	7 1		
	8	3 5		
		9 2		
	10			

TextPost(<u>postID</u> : int, wordCount: int,	postid [PK] integer	wordcount integer	<i>j</i> .	textcon	ten	t	,
textContent: text)	11	1	20	Excited	to	announce my	y new partnership!
	12 8			Check	out	my latest rev	iew of the smartwatch.
	13	2	00	Tips on	he	althy lifestyle	and meal prep.
	14	1	50	Behind	the	scenes of m	y recent campaign.
	15		90	Thank y	/ou	all for your s	upport!
Platform(<u>platformName</u> : varchar(50), numUsers:	platformname [PK] character va	arying (50) '		numuse bigint	rs	ř	
bigint)	Instagram			150000	000	000	
	YouTube			250000	000	000	
	TikTok			120000	000	000	
	Facebook			45000	000	000	
	Twitter			50000000			
SponsorCompany(<u>companyl</u> <u>D</u> : int, numEmployees: int)	companyid [PK] integer	numemp integer	loye	es			
	1			200			
	2			350			
	3			50			
	4			1200			
	5			75			
Product(<u>productName</u> : varchar(100), <u>companyID</u> :	productname [PK] character vary	ring (100)		npanyid (] integer	<i>)</i> '	price money	category character varying (50)
int, price: money, category:	Monster Energy Di	rink			1	\$2.99	Beverage
varchar(50))	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	branddealid [PK] integer	adtype	or vorv	ing (50)	paymentrat money	e 🎤	companyid integer	<i>'</i>	postid integer	<i>'</i>
: int, adType: varchar(50),	1	Sponso			\$500.00		integer	1	integer	1
paymentRate: money,	2	Shouto		31	\$300.00			2		2
companyID: int, postID: int)	3			aboration	\$750.00			3		3
	4	Review	ay oon	aboration	\$0.05			4		4
	5 Product		t Place	ment	\$0.10			5		16
	6	Review			\$0.30			1		13
	7	Shouto	ut		\$100.00			1		15
BrandDealTwo(<u>adType</u> : varchar(50), paymentType:	adtype [PK] character v	arying (5	0)	payment characte	type r varying (50)	<i>j</i>				
varchar(50))	Sponsored Pos	t		Flat Fee						
	Shoutout			Flat Fee						
	Giveaway Colla	boration		Flat Fee						
	Review			Per View	I					
	Product Placen	nent		Per View						
Agency(<u>agencyID</u> : int, establishedDate: date,	agencyid establishedo date		eddate 🥕	staffsize integer	<i>/</i>	location character varying (100)		<i>/</i> ·		
staffSize: int, location:	1 2010-		0-05-1	-05-12 25		25	USA			
varchar(100))	2 2015		5-08-2	5-08-20 40		40	USA			
	3 2008		8-03-15			15	CAN			
		4 202	0-11-0	1-01 10		10	CAN			
		5 201	2-07-3	30		30	CAN			
ManagementContractOne(<u>c</u>	contractid [PK] integer	influence money	erbases		influencerpayor		influencerid integer	j	agencyid integer	i
ontractID: int,	1	\$50,000.	.00			0.7		4		1
influencerBaseSalary:	2	\$60,000	.00			0.75		5		2
money, influencerPayout: float, influencerID: int,	3	\$55,000	.00			0.8		3		3
agencyID: int)	4	\$70,000	.00			0.85		1		4
agency is: into	5	\$65,000.	.00			0.9		2		5
ManagementContractTwo(<u>i</u> nfluencerPayout: float,				panycomm ble precisio						
companyCommission: float)		0.7			0.3					
' '		0.75			0.25					
		0.8			0.2					
		0.85			0.15					
		0.9			0.1					

CollaboratesWith(<u>agencyID</u> : int, <u>companyID</u> :	agencyid [PK] integer	companyid [PK] intege	, ,	numcolla integer	aborations
numCollaborations: int)	1		1		5
	1		2		3
	2		3		4
	3		4		2
	4		5		6
Advertise(<u>productName</u> : varchar(100), <u>companyID</u> :	productname [PK] character var	ying (100) 🖍		panyid integer	postid [PK] integer
int, postID : int)	Monster Energy Drink		1		
	Smartwatch Alph	a		2	
	Choco Protein Ba	r		3	1
	Silent Keyboard			4	
	Dove Shampoo			5	1

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 **SOL** 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 **SOL 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)
```

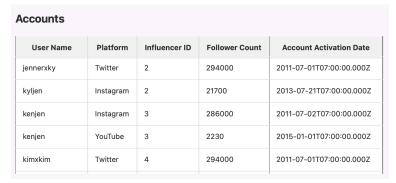
Code Position:

appService.js line 333-336, 339-343

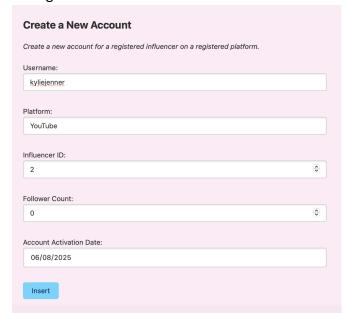
Functionality GUI Screenshot Demo

INSERT Operation

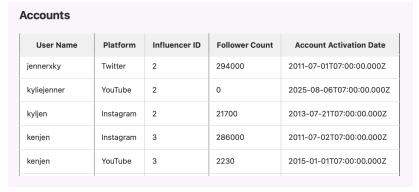
• Before:



• During:



• After:



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

DELETE Operation

• Before:

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

Accounts				
User Name	Platform	Influencer ID	Follower Count	Account Activation Date
addraeins	Instagram	1	980000	2025-01-01T07:00:00.000Z
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

• 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

Accounts

User Name	Platform	Influencer ID	Follower Count	Account Activation Date
addraeins	Instagram	1	980000	2025-01-01T07:00:00.000Z
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
shawn	Instagram	5	357	2016-10-09T07:00:00.000Z

^{**} Deleting Influencer cascades to Account

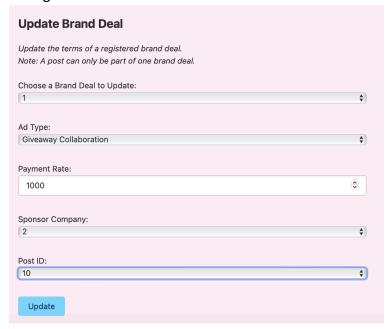
UPDATE Operation

• 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:



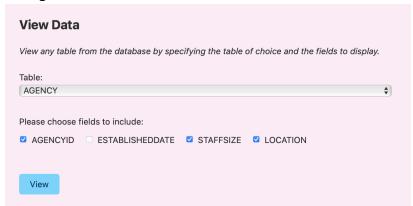
After:

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

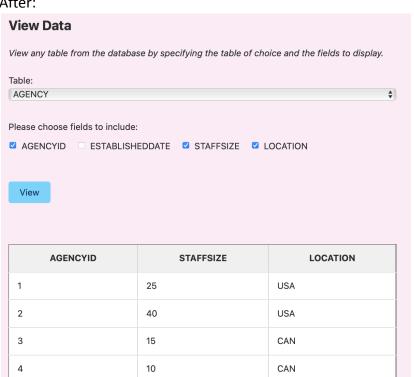
PROJECTION

- Before: no table(s)
- During:



After:

5



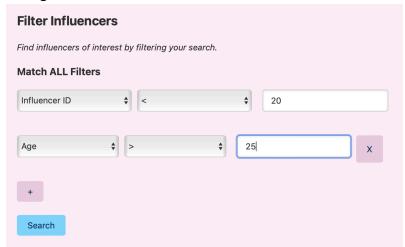
CAN

30

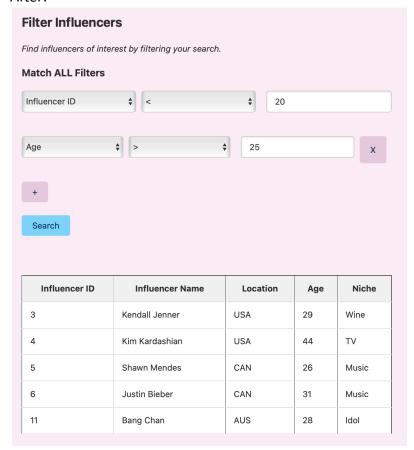
SELECTION (AND)

Before: no table(s)

During:



After:



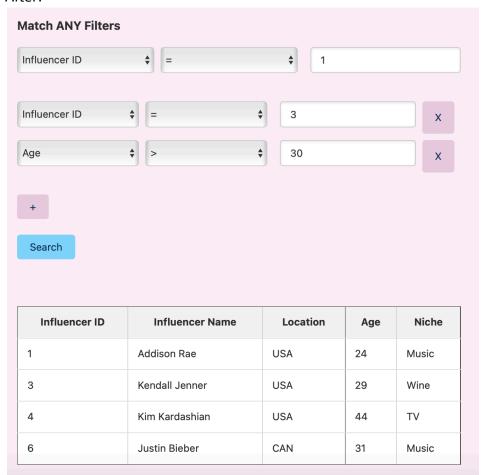
** Table is scrollable, so not all tuples are currently in frame

SELECTION (OR)

Before: no table(s)

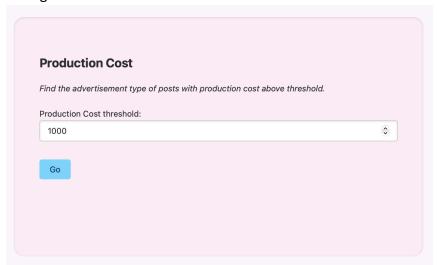
• During:



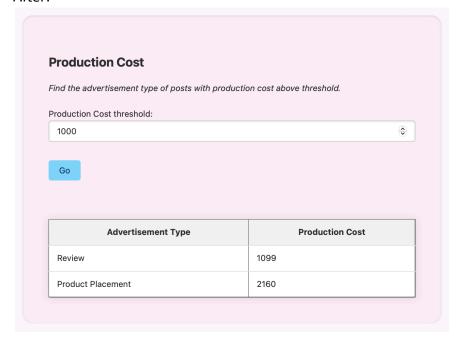


JOIN

- Before: no table(s)
- During:



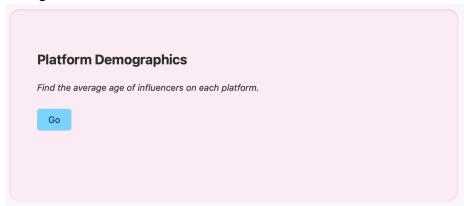
• After:

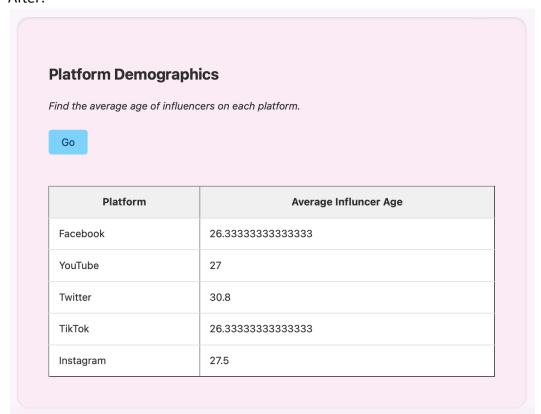


Aggregation with GROUP BY

• Before: no table(s)

• During:

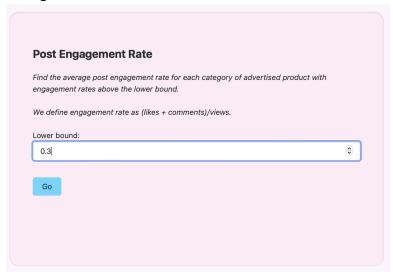


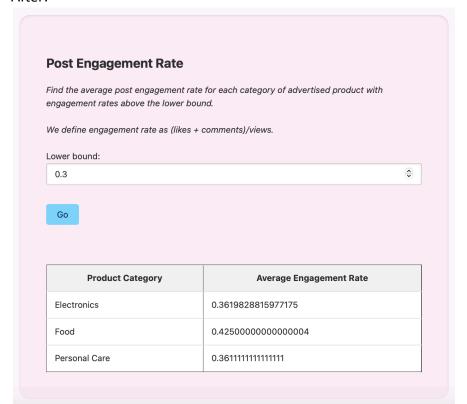


Aggregation with HAVING

• Before: no table(s)

• During:

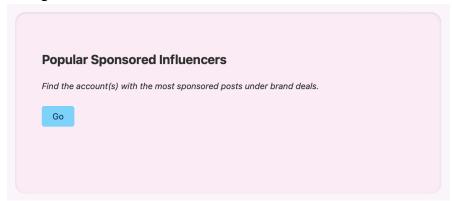


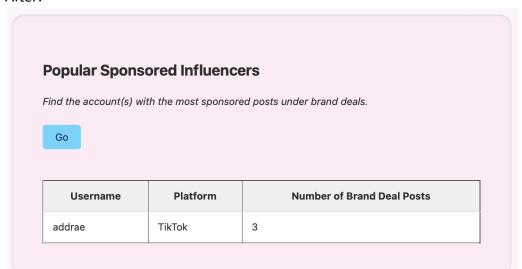


Nested aggregation with GROUP BY

• Before: no table(s)

• During:





DIVISION

- Before: no table(s)
- During:

