МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Национальный исследовательский ядерный университет «МИФИ»

Институт

интеллектуальных кибернетических систем

Кафедра кибернетики (№ 22)

Отчёт о работе по курсу

«Базы данных (теоретические основы баз данных)»

Вариант «Instagram»

Выполнил	Ле Дык Ань
Группа	Б20-205
Вариант	Instagram
Преподаватель	Петровская А.В.
Проверяющий	Старков С.В.
Оценка	95/100

2021

Table of contents

1. Task formulation

Design a database for Instagram-style application, which allows users to shares photos and videos to other people. The database should contain information about users, posts and interactions between users and users, along with interactions between users and post.

2. Conceptual model of the database

During the work on the task, the subject area was analyzed, the features of which are reflected by the constructed conceptual model:

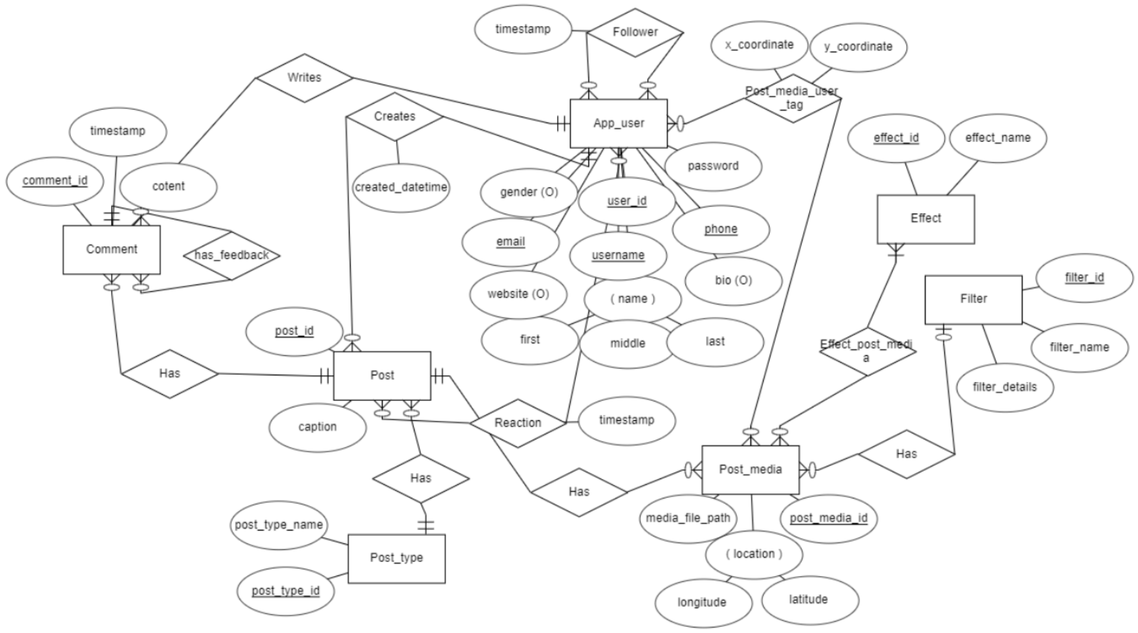


Figure 1 Conceptual database model for the Instagram-like application

a. Concretization of subject area
   It’s necessary to create a system that reflects the basic information about the users like username, email, password, etc. Users can create posts, like or unlike posts, comment on posts and tag other users in a post, our system need to store all the necessary information about interaction between users and posts. Our application allows users to follow each other, so database need to be able to keep track of the relationships among the users. And the application allows users to assign a filter for each of their photo or video, and also allows adding some effects on one photo or video, so it is necessary for out database to store information about the filters and the effects and their appearances on photo or video.

b. Description of the subject area
   Let’s consider further functionality that an user have:

i. Creating new post. As you know, Instagram is a social network that allows people to share their photos and videos, so this creating new post is the main functionality of the application.
ii. Among a huge amount of posts, there may be some posts that really impresses you, so like and comment functionality come in place, the app allows users to leave comments on posts, reply to comments and like or unlike posts.
iii. If you find normal photo or video boring, Instagram allow you to add effects or filter to your photos and videos.
iv. Sometimes you find a person who shares same interest with you or you are just interested in that person and you don't want to miss any of his/her post, then Instagram provides a functionality that users can follow each other.
v. Taking a photo of a friend, you want to let him/her know he/she is in your photo when posting it in your page, Instagram has tagging feature that allows user to tag others in their posts.
vi. Post can be represent by one of 3 types: normal post, story or reel.

Base on the description of the functionalities provided by our Instagram-like application, the following entities were identified:

1. App_user (User)
2. Post (Post)
3. Post_media (Photo or Video)
4. Post_type (Post, story or reel)
5. Filter (Filter)
6. Effect (Effect)
7. Comment (Comment)

c. Attribute description

For User entity:

Attribute	Description
user_id	Unique user ID
first	First name
middle	Middle name
last	Last name
username	Username
website	Website like personal or link other social media platform
bio	Short biographical profile of someone
email	Email
phone	Phone number
gender	Gender
password	User's password

For Post entity:

Attribute	Description

post_id	Unique post ID
caption	Caption

For Post_media entity:

Attribute	Description
post_media_id	PK
Media_pfile_path	Path to file
Longitude	Longitude of location of media
Latitude	Latitude of location of media

For Comment entity:

Attribute	Description
Comment_id	PK
Created_at	Timestamp when comment is created
Content	Content

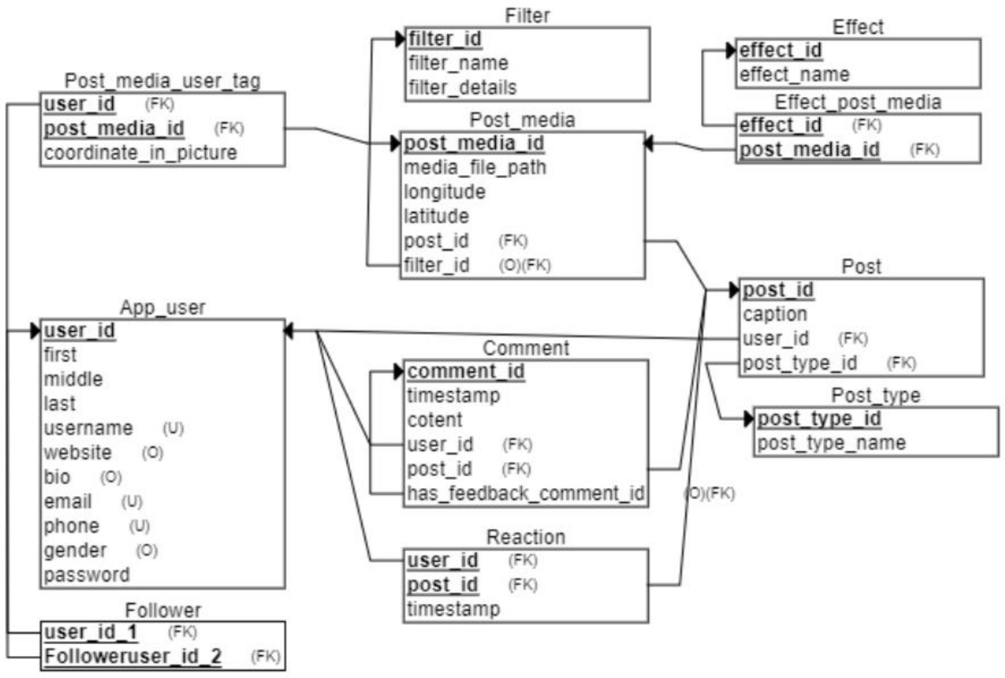
For Filter entity:

Attribute	Description
Filter_id	PK
Filter_name	Name of the filter
Filter_detail	Detail about filter

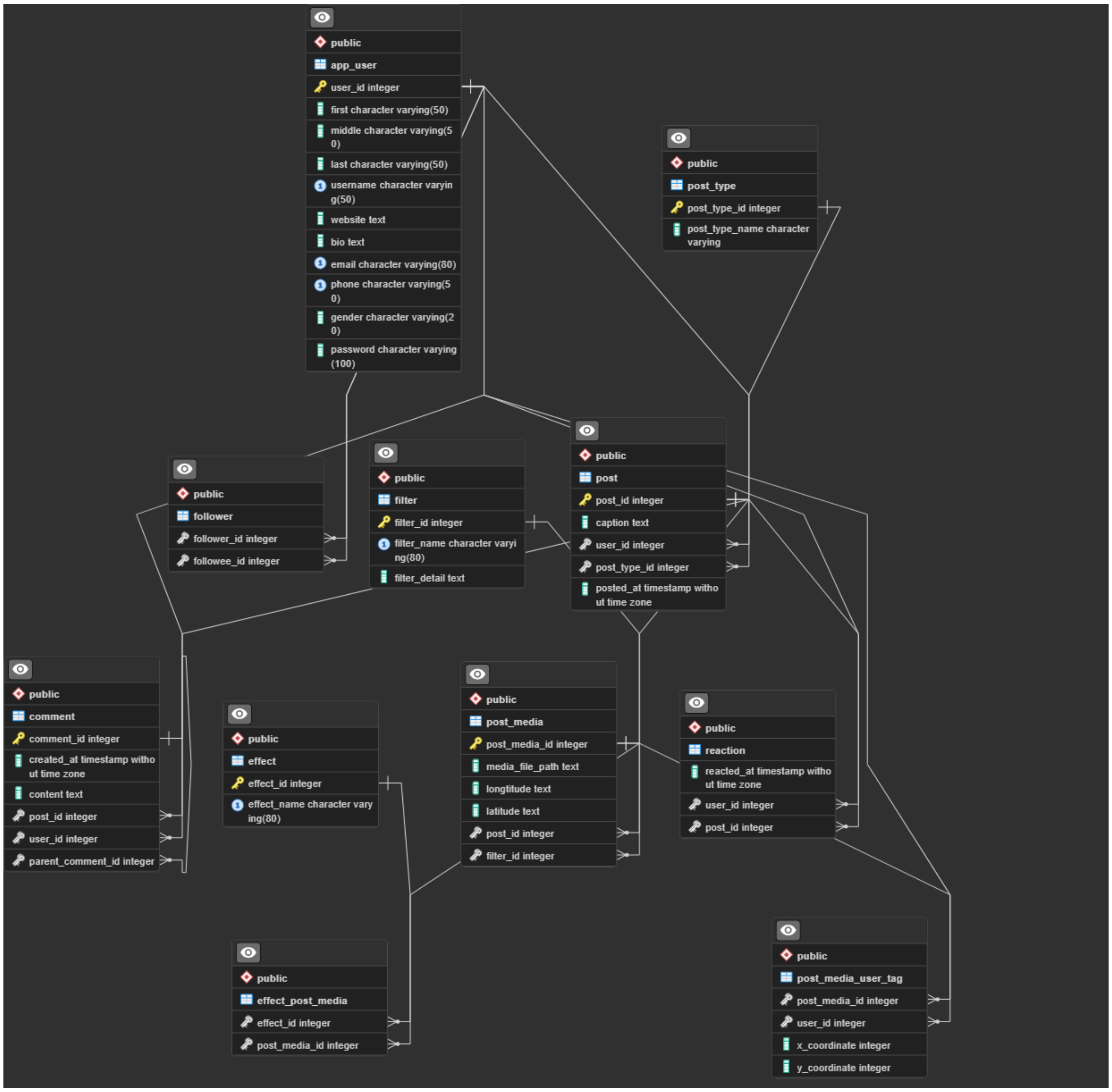
For Effect entity:

Attribute	Description
Effect_id	PK
Effect_name	Name of effect

3. Logical design



4. Physical design
   PostgreSQL was chosen as the DBMS for the database implementation. Based on the above logical model, the following physical database model was built:



a. Creating tables

Create App_user table:

```sql
CREATE TABLE IF NOT EXISTS App_user
(
  user_id SERIAL PRIMARY KEY,
  first VARCHAR(50) NOT NULL,
  middle VARCHAR(50),
  last VARCHAR(50) NOT NULL,
  username VARCHAR(50) CONSTRAINT valid_len_un CHECK (char_length(username) >=
4) UNIQUE NOT NULL,
  website TEXT,
  bio TEXT,
  email VARCHAR(80) CONSTRAINT valid_email CHECK(email LIKE '%@%.%') UNIQUE NOT
NULL,
  phone VARCHAR(50) CONSTRAINT valid_phone CHECK(phone NOT LIKE '%[^0-9]%')
UNIQUE NOT NULL,
  gender VARCHAR(20) CONSTRAINT valid_gender CHECK(gender IN('Male', 'Female',
'Unknown')) DEFAULT 'Unknown',
  password VARCHAR(100) CHECK (char_length(password) >= 4) NOT NULL
);
```

Create Post_type table:

```sql
CREATE TABLE IF NOT EXISTS Post_type
(
  post_type_id SERIAL PRIMARY KEY,
  post_type_name VARCHAR CHECK(post_type_name IN('post', 'story', 'reel')) NOT
NULL
);
```

Create Post table:

```sql
CREATE TABLE IF NOT EXISTS Post
(
  post_id SERIAL PRIMARY KEY,
  caption TEXT,
  user_id INTEGER REFERENCES App_user ON DELETE CASCADE,
  post_type_id INTEGER REFERENCES Post_type ON DELETE RESTRICT,
  posted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Create Comment table:

```sql
CREATE TABLE IF NOT EXISTS Comment
(
    comment_id SERIAL PRIMARY KEY,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    content TEXT NOT NULL,
    post_id INTEGER REFERENCES Post ON DELETE CASCADE,
    user_id INTEGER REFERENCES App_user ON DELETE CASCADE,
    parent_comment_id INTEGER REFERENCES Comment ON DELETE CASCADE
);
```

Create Reaction table:

```sql
CREATE TABLE IF NOT EXISTS Reaction
(
    reacted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    user_id INTEGER REFERENCES App_user ON DELETE NO ACTION,
    post_id INTEGER REFERENCES Post ON DELETE CASCADE,
    CONSTRAINT no_duplicate_like_from_one_user UNIQUE (user_id, post_id)
);
```

Creat Post_media table:

```
CREATE TABLE IF NOT EXISTS Post_media
(
    post_media_id SERIAL PRIMARY KEY,
    media_file_path TEXT NOT NULL,
    longtitude TEXT NOT NULL,
    latitude TEXT NOT NULL,
    post_id INTEGER REFERENCES Post ON DELETE CASCADE
);
```

Create Post_media_User_tag table:

```
CREATE TABLE IF NOT EXISTS Post_media_User_tag
(
    post_media_id INTEGER REFERENCES Post_media ON DELETE CASCADE,
    user_id INTEGER REFERENCES App_user ON DELETE CASCADE,
    x_coordinate INTEGER NOT NULL,
    y_coordinate INTEGER NOT NULL
);
```

Create Filter table:

```
CREATE TABLE IF NOT EXISTS Filter
(
    filter_id SERIAL PRIMARY KEY,
    filter_name VARCHAR(80) UNIQUE NOT NULL,
    filter_detail text
);
```

Add CONSTRAINT FOREIGN KEY, referencing to table Filter, to table Post_media:

```
ALTER TABLE Post_media
ADD COLUMN IF NOT EXISTS filter_id INTEGER REFERENCES Filter ON DELETE NO
ACTION;
```

Create Effect table:

```sql
CREATE TABLE IF NOT EXISTS Effect
(
    effect_id SERIAL PRIMARY KEY,
    effect_name VARCHAR(80) UNIQUE NOT NULL
);
```

Create table Effect_Post_media:

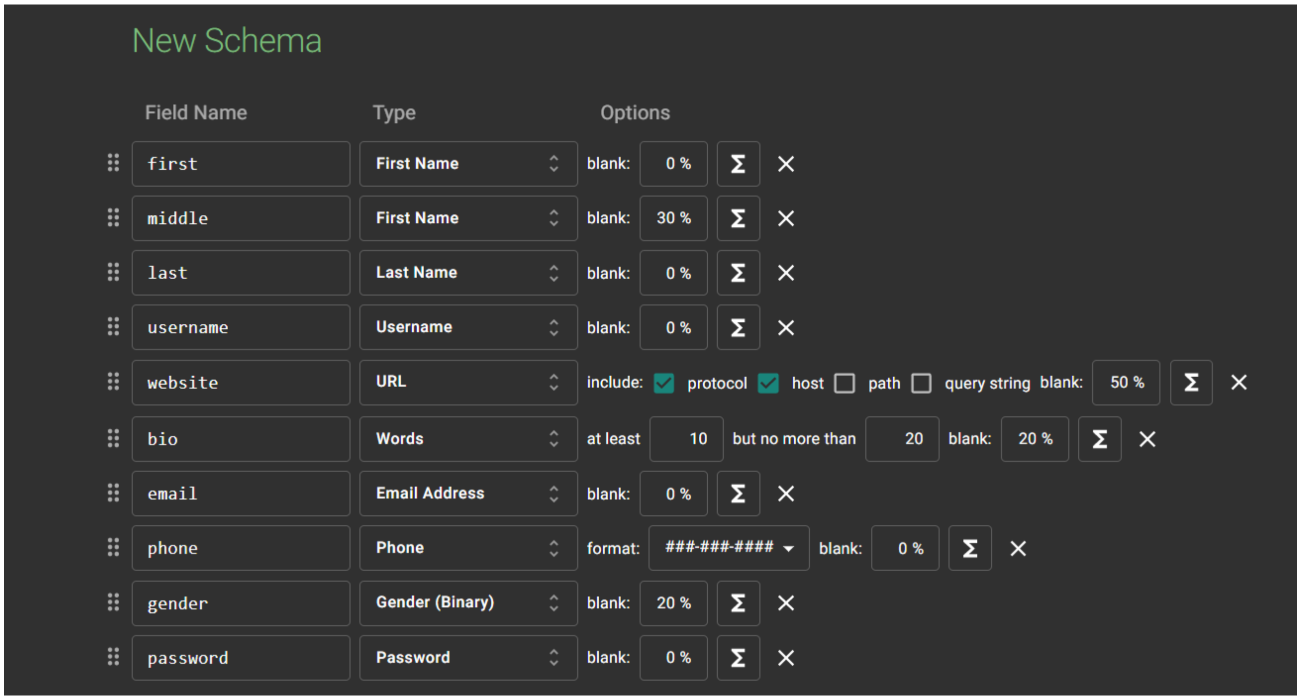
```sql
CREATE TABLE IF NOT EXISTS Effect_Post_media
(
    effect_id INTEGER REFERENCES Effect ON DELETE CASCADE,
    post_media_id INTEGER REFERENCES Post_media ON DELETE CASCADE
);

ALTER TABLE Effect_Post_media
ADD CONSTRAINT unique_tuple_E_Pmedia UNIQUE NULLS NOT DISTINCT (effect_id,
post_media_id);
```

Create table Follower:

```sql
CREATE TABLE IF NOT EXISTS Follower
(
    follower_id INTEGER REFERENCES App_user ON DELETE CASCADE,
    followee_id INTEGER REFERENCES App_user ON DELETE CASCADE
);

ALTER TABLE Follower
ADD CONSTRAINT unique_follow UNIQUE NULLS NOT DISTINCT(follower_id,
followee_id);
```

b. Populating the database

Insert into App_user table:



New Schema

Field Name	Type	Options
first	First Name	blank: 0 %
middle	First Name	blank: 30 %
last	Last Name	blank: 0 %
username	Username	blank: 0 %
website	URL	include: ✓ protocol ✓ host ☐ path ☐ query string blank: 50 %
bio	Words	at least 10 but no more than 20 blank: 20 %
email	Email Address	blank: 0 %
phone	Phone	format: ###-###-#### blank: 0 %
gender	Gender (Binary)	blank: 20 %
password	Password	blank: 0 %

```
instagram1=# \i C:/Users/1235455/Downloads/App_user.sql
```

Insert into Post_type table:

```
import psycopg2 as ps

# establishing the connection
conn = ps.connect(database="instagram1",
                  user='postgres',
                  password='Icandoit2706',
                  host='127.0.0.1', port='5432')

conn.autocommit = False

cursor = conn.cursor()

cursor.execute('''INSERT INTO Post_type(post_type_name) VALUES
('reel');''')
cursor.execute('''INSERT INTO Post_type(post_type_name) VALUES
('post');''')
cursor.execute('''INSERT INTO Post_type(post_type_name) VALUES
('story');''')

conn.commit()
print("Records inserted........")
cursor.close()
conn.close()
```

Insert into Post table:

```python
def generate_timestamp():
    MONTHS = ['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG',
'SEP', 'OCT', 'NOV', 'DEC']
    month = random.choice(MONTHS)
    if month == 'FEB':
        date = random.randint(1, 28)
    else:
        date = random.randint(1, 30)
    year = random.randint(2015, 2021)
    raw_hour = random.randint(0, 23)
    hour = ''
    if raw_hour < 10:
        hour = hour + '0'
        hour = hour + str(raw_hour)
    else:
        hour = str(raw_hour)

    raw_minute = random.randint(0, 59)
    minute = ''
    if raw_minute < 10:
        minute = minute + '0'
        minute = minute + str(raw_minute)
    else:
        minute = str(raw_minute)

    time_stamp = f'\'{date}-{month}-{year} {hour}:{minute}\''
    return time_stamp
```

```
def fill_Post():
    INT = ['Hello! ', 'HEllo. ', 'hello ', 'HeLlO ', 'Hi ', 'HI! ', 'hI.
', 'Here is my ', 'What a nice day ',
           'How are you guys? ', 'Good morning, ', 'Let me introduce to
you ', 'Have a nice day ', 'Nice to meet you! ',
           'Say hi to my ', 'This is ', 'I have a ', 'Good night, ']
    CONT = ['my cat! ', 'my new puppy! ', 'the picture of nature! ',
'something ', 'my girl friend ', 'our group. ',
            'Happy classmates ', 'nice weather ', 'feeling sleepy ', 'a
book ', 'my lover ', 'my dream ', 'a bike ',
            'a car ',
            'new book ', 'new ps5 ', 'my friend ', 'oh dear ', 'Santa
Clause ', 'oops ']
    EMOJI = ['=(', '=)', ':3', ';)', '.-.', '-_-', '+_+', '=_=', ':P',
'KEK', ':/', 'o_0', 'UwU', '*v*']

    query = 'INSERT INTO Post(caption, user_id, post_type_id, posted_at)
VALUES '

    for i in range(2):
        caption = random.choice(INT) + random.choice(CONT)
        if random.randint(0, 1) == 1:
            caption += random.choice(EMOJI)
        posted_at = generate_timestamp()
        user_id = random.randint(1, 600)
        post_type_id = random.randint(1, 3)
        query += f'(\'{caption}\', {user_id}, {post_type_id},
TO_TIMESTAMP({posted_at}, \'DD MON YYYY HH24:MI\')), '
    query = query[:-2]
    query += ';'
    return query
```

```
def main():
    conn = ps.connect(database="instagram1",
                      user='postgres',
                      password='Icandoit2706')
    conn.autocommit = False
    cursor = conn.cursor()
    cursor.execute(fill_Post())
    conn.commit()
    cursor.close()
    conn.close()
```

Insert into Reaction table:

```
def fill_Reaction():
    query = 'INSERT INTO Reaction(reacted_at, user_id, post_id) VALUES '
    for user_id in range(1, 601):
        reacted_at = generate_timestamp()
        for post_id in range(36229, 37429):
            if random.randint(0, 100) > 66:
                query += f'(TO_TIMESTAMP({reacted_at}, \'DD MON YYYY
HH24:MI\'), {user_id}, {post_id}), '
    query = query[:-2]
    query += ';'
    return query
```

Insert into Filter table:

```
INSERT INTO Filter (filter_name, filter_detail) VALUES
('Stereoscopy','This filter is perfect for boomerang lovers that want to speed up their mainstream boomerangs
 and take them to the next level.'),

('The little mermaid','If you have always dreamed about being a cute princess with an amazing voice such
 as The Little Mermaid this filter is the one for you. Just try it out to see yourself as the famous red hair
 and to even hear her pronouncing your words!'),

('Not so basic','This Instagram filter gives you a great glow up with this artistic makeup around your eyes.
Besides, it also softens your skin and applies a cute lipstick!'),

('Snow White','Keeping up with the ‘I want to be a princess’ trend! This Snow White Instagram filter provides
 you with this princess’s eyes, headband, and charming smile.'),

('BW Vibe','This Instagram filter gives you a black and white look with a mysterious blurred vibe. Add some
 mystical music and you’ll rule Instagram stories!'),

('Big City Life','Soft skin, white shiny teeth, and great lightning: this Instagram filter is basically
everything
 we have been dreaming about!'),

('Bubblegum','This Instagram filter provides you with a moving background of sparkling shapes and colors. It
doesn’t
 change anything on your face, which can be a true advantage since you don’t always need to look like a
perfect baby
 doll.'),
('Red Berries','This is the Instagram filter we all need when it comes to making indoor stories since it
highlights
 the whites in your room and gives a sparkling vibe to make it look more appealing. It also softens your skin
and
 makes your nose look a little bit thinner.'),

('Cute Baby','If you want to look pretty and cool you should try out this Instagram filter that softens your
skin,
 makes your nose look thinner, brightens your eyes, applies eyeliner, and places a cute little sky blue
butterfly
 on top of your nose!'),

('Lil Anime Doll','As the name suggests, this is definitely not one of those filters that you almost can't
tell if
 someone is using them or not. However, it makes you look both funny and pretty!'),

('Dreamy Summer','Tanned skin, bright light eyes, and a bunch of sparkling butterflies around your head is
just what
 you need to look stunning. This filter does it for you!'),

('Print','This is honestly one of the most useful Instagram filters we've seen.
It lets you 'print' or 'scan' anything you want and then uses it as a filter to create your story.
For instance, you could scan some line of your favorite book and then record your story with that phrase
applied!'),

('Daisies','If you wish to set your Instagram story up you should try Daisies, an Instagram filter that
provides a
 great makeup and many cute flowers around your head (or just a single one if you tap the screen).'),

('Lens','Lens is also one of our favorite Instagram filters.
It keeps your footage small right in the middle of your screen and makes a large, upturned replica in black
and white background.'),

('Smile','If you are feeling full of joy you should try out this Instagram filter that adds a 'positive vibes'
sign on
```

```
 top of your head and cool smiley faces melting on your cheeks.
Even though it might sound childish, once you’ve tried you’ll see that it makes you look pretty cool.');
```

Insert into Post_media:

```
def fill_Post_media():
    DISK = ['C:/', 'D:/', 'E:/', 'F:/', 'http://www.var.com/',
'http://www.google_picture.com/', 'http://www.wiki.com/',
            'http://www.example.ru/', 'http://www.facebook.com/',
'http://www.instagram.com/',
            'http://www.nothing.com/',
            'http://www.pages.com/', 'http://www.example.com/', 'User/']
    FOLDER = ['Pictures/', 'Forum/', 'Home/', 'Desktop/', 'something/',
'another/', 'apilibrary/', 'TravelBrochure',
              'Newsletters/', 'January/', 'February/', 'utilities/',
'folder1/']
    FILE_NAME = ['MorbiNon', 'NuncProin', 'Est', 'AliquetUltricesErat',
'Dapibus', 'FeugiatNon', 'Elementum',
                 'OrciLuctusEt', 'CongueRisus'
                                 'AnteIpsumPrimis', 'RutrumAt',
'ProinLeoOdio', 'Felis', 'Adipiscing', 'TurpisInteger',
                 'EuOrci', 'VivamusTortor', 'TempusVelPede']
    FILE_FORMAT = ['.jpeg', '.jpg', '.png', '.gif', '.tiff', '.psd',
'.pdf', '.eps', '.ai', '.indd', '.raw',
                   '.mp4', '.mov', '.wmv', '.avchd', '.flv', '.f4v',
'.swf', '.mkv', '.mpeg']
    query = 'INSERT INTO Post_media(media_file_path, longtitude,
latitude, post_id, filter_id) VALUES '

    for i in range(2400):
        media_file_path = (random.choice(DISK) + random.choice(FOLDER) +
random.choice(FILE_NAME) +
                           random.choice(FILE_FORMAT))
        longitude = str(np.random.uniform(-180, 180))
        latitude = str(np.random.uniform(-90, 90))
        post_id = random.randint(36229, 37428)
        filter_id = random.randint(1, 15)
        query += f'(\'{media_file_path}\', \'{longitude}\',
\'{latitude}\', {post_id}, {filter_id}), '
    query = query[:-2]
    query += ';'
    return query
```

Insert into Post_media_User_tag table:

```
def fill_Post_media_User_tag():
    query = 'INSERT INTO Post_media_User_tag(post_media_id, user_id,
x_coordinate, y_coordinate) VALUES '
    for i in range(2000):
        post_media_id = random.randint(1, 2402)
        user_id = random.randint(1, 600)
        x_coordinate = random.randint(1, 1000)
        y_coordinate = random.randint(1, 1000)
        query += f'({post_media_id}, {user_id}, {x_coordinate},
{y_coordinate}), '
    query = query[:-2]
    query += ';'
    return query
```

Insert into Effect:

```
INSERT INTO Effect(effect_name) VALUES

('Hat'),

('Mickey Mouse'),

('Big mouth'),

('Red nose'),

('Galsses'),

('Tom Holland'),

('Dragon'),

('Horse'),

('Cat'),

('Karate'),

('Pig'),

('At beach'),

('Smile'),

('Creepy'),

('Shiba Inu'),

('Cucumber'),

('Elvis Presley'),

('Metro'),

('Google'),

('Flower');
```

Insert into Effect_Post_media table:

```
def fill_Effect_Post_media():
    query = 'INSERT INTO Effect_Post_media(effect_id, post_media_id)
VALUES '
    for post_media_id in range(1, 2042):
        for effect_id in range(1, 21):
            if random.randint(0, 100) <= 10:
                query += f'({effect_id}, {post_media_id}), '
    query = query[:-2]
    query += ';'
    return query
```

Insert into Comment table:

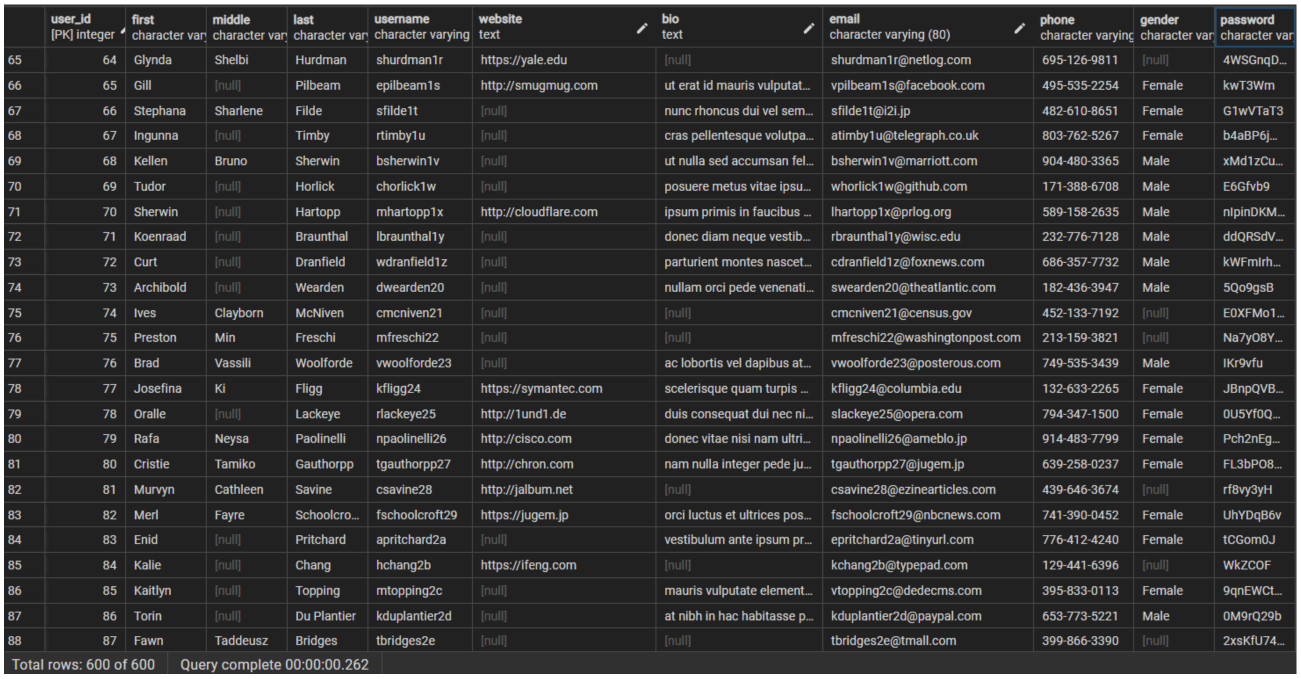
```
def fill_Comment():
    INT = ['Hello! ', 'HEllo. ', 'hello ', 'HeLlO ', 'Hi ', 'HI! ', 'hI.
', 'Here is my ', 'What a nice day ',
           'How are you guys? ', 'Good morning, ', 'Let me introduce to
you ', 'Have a nice day ', 'Nice to meet you! ',
           'Say hi to my ', 'This is ', 'I have a ', 'Good night, ']
    CONT = ['my cat! ', 'my new puppy! ', 'the picture of nature! ',
'something ', 'my girl friend ', 'our group. ',
            'Happy classmates ', 'nice weather ', 'feeling sleepy ', 'a
book ', 'my lover ', 'my dream ', 'a bike ',
            'a car ',
            'new book ', 'new ps5 ', 'my friend ', 'oh dear ', 'Santa
Clause ', 'oops ']
    EMOJI = ['=(', '=)', ':3', ';)', '.-.', '-_-', '+_+', '=_=', ':P',
'KEK', ':/', 'o_0', 'UwU', '*v*']
    query = 'INSERT INTO Comment(created_at, content, post_id, user_id)
VALUES '
    for i in range(3598):
        created_at = generate_timestamp()
        content = random.choice(INT) + random.choice(CONT)
        if random.randint(0, 1) == 1:
            content += random.choice(EMOJI)
        post_id = random.randint(36229, 37428)
        user_id = random.randint(1, 600)
        query += f'(TO_TIMESTAMP({created_at}, \'DD MON YYYY HH24:MI\'),
\'{content}\', {post_id}, {user_id}), '
    query = query[:-2]
    query += ';'
    return query


def fill_Reply():
    INT = ['Hello! ', 'HEllo. ', 'hello ', 'HeLlO ', 'Hi ', 'HI! ', 'hI.
', 'Here is my ', 'What a nice day ',
           'How are you guys? ', 'Good morning, ', 'Let me introduce to
you ', 'Have a nice day ', 'Nice to meet you! ',
           'Say hi to my ', 'This is ', 'I have a ', 'Good night, ']
    CONT = ['my cat! ', 'my new puppy! ', 'the picture of nature! ',
'something ', 'my girl friend ', 'our group. ',
            'Happy classmates ', 'nice weather ', 'feeling sleepy ', 'a
book ', 'my lover ', 'my dream ', 'a bike ',
            'a car ',
            'new book ', 'new ps5 ', 'my friend ', 'oh dear ', 'Santa
Clause ', 'oops ']
    EMOJI = ['=(', '=)', ':3', ';)', '.-.', '-_-', '+_+', '=_=', ':P',
'KEK', ':/', 'o_0', 'UwU', '*v*']
    query = 'INSERT INTO Comment(created_at, content, post_id, user_id,
parent_comment_id) VALUES '
    for i in range(1, 10801):
        created_at = generate_timestamp()
        content = random.choice(INT) + random.choice(CONT)
        if random.randint(0, 1) == 1:
            content += random.choice(EMOJI)
        # post_id = random.randint(36229, 37428)
```

Insert into Follower table:

```python
def fill_Follower():
    query = 'INSERT INTO Follower(follower_id, followee_id) VALUES '
    for follower_id in range(1, 601):
        for followee_id in range(1, 601):
            if follower_id is not followee_id:
                if random.randint(0, 100) <= 10:
                    query += f'({follower_id}, {followee_id}), '
    query = query[:-2]
    query += ';'
    return query
```

c. Filling results

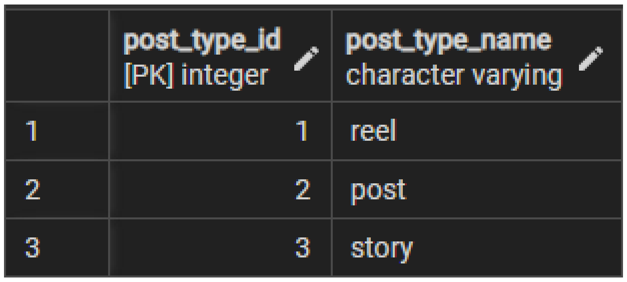
Result of filling the App_user table:



	user_id [PK] integer	first character vary	middle character vary	last character vary	username character varying	website text	bio text	email character varying (80)	phone character varying	gender character vary	password character vary
65	64	Glynda	Shelbi	Hurdman	shurdman1r	https://yale.edu	[null]	shurdman1r@netlog.com	695-126-9811	[null]	4WSGnqD...
66	65	Gill	[null]	Pilbeam	epilbeam1s	http://smugmug.com	ut erat id mauris vulputat...	vpilbeam1s@facebook.com	495-535-2254	Female	kwT3Wm
67	66	Stephana	Sharlene	Filde	sfilde1t	[null]	nunc rhoncus dui vel sem...	sfilde1t@i2i.jp	482-610-8651	Female	G1wVTaT3
68	67	Ingunna	[null]	Timby	rtimby1u	[null]	cras pellentesque volutpa...	atimby1u@telegraph.co.uk	803-762-5267	Female	b4aBP6j...
69	68	Kellen	Bruno	Sherwin	bsherwin1v	[null]	ut nulla sed accumsan fel...	bsherwin1v@marriott.com	904-480-3365	Male	xMd1zCu...
70	69	Tudor	[null]	Horlick	chorlick1w	[null]	posuere metus vitae ipsu...	whorlick1w@github.com	171-388-6708	Male	E6Gfvb9
71	70	Sherwin	[null]	Hartopp	mhartopp1x	http://cloudflare.com	ipsum primis in faucibus ...	lhartopp1x@prlog.org	589-158-2635	Male	nIpinDKM...
72	71	Koenraad	[null]	Braunthal	lbraunthal1y	[null]	donec diam neque vestib...	rbraunthal1y@wisc.edu	232-776-7128	Male	ddQRSdV...
73	72	Curt	[null]	Dranfield	wdranfield1z	[null]	parturient montes nascet...	cdranfield1z@foxnews.com	686-357-7732	Male	kWFmIrh...
74	73	Archibold	[null]	Wearden	dwearden20	[null]	nullam orci pede venenati...	swearden20@theatlantic.com	182-436-3947	Male	5Qo9gsB
75	74	Ives	Clayborn	McNiven	cmcniven21	[null]	[null]	cmcniven21@census.gov	452-133-7192	[null]	E0XFMo1...
76	75	Preston	Min	Freschi	mfreschi22	[null]	[null]	mfreschi22@washingtonpost.com	213-159-3821	[null]	Na7yO8Y...
77	76	Brad	Vassili	Woolforde	vwoolforde23	[null]	ac lobortis vel dapibus at...	vwoolforde23@posterous.com	749-535-3439	Male	IKr9vfu
78	77	Josefina	Ki	Fligg	kfligg24	https://symantec.com	scelerisque quam turpis ...	kfligg24@columbia.edu	132-633-2265	Female	JBnpQVB...
79	78	Oralle	[null]	Lackeye	rlackeye25	http://1und1.de	duis consequat dui nec ni...	slackeye25@opera.com	794-347-1500	Female	0U5Yf0Q...
80	79	Rafa	Neysa	Paolinelli	npaolinelli26	http://cisco.com	donec vitae nisi nam ultri...	npaolinelli26@ameblo.jp	914-483-7799	Female	Pch2nEg...
81	80	Cristie	Tamiko	Gauthorpp	tgauthorpp27	http://chron.com	nam nulla integer pede ju...	tgauthorpp27@jugem.jp	639-258-0237	Female	FL3bPO8...
82	81	Murvyn	Cathleen	Savine	csavine28	http://jalbum.net	[null]	csavine28@ezinearticles.com	439-646-3674	[null]	rf8vy3yH
83	82	Merl	Fayre	Schoolcro...	fschoolcroft29	https://jugem.jp	orci luctus et ultrices pos...	fschoolcroft29@nbcnews.com	741-390-0452	Female	UhYDqB6v
84	83	Enid	[null]	Pritchard	apritchard2a	[null]	vestibulum ante ipsum pr...	epritchard2a@tinyurl.com	776-412-4240	Female	tCGom0J
85	84	Kalie	[null]	Chang	hchang2b	https://ifeng.com	[null]	kchang2b@typepad.com	129-441-6396	[null]	WkZCOF
86	85	Kaitlyn	[null]	Topping	mtopping2c	[null]	mauris vulputate element...	vtopping2c@dedecms.com	395-833-0113	Female	9qnEWCt...
87	86	Torin	[null]	Du Plantier	kduplantier2d	[null]	at nibh in hac habitasse p...	kduplantier2d@paypal.com	653-773-5221	Male	0M9rQ29b
88	87	Fawn	Taddeusz	Bridges	tbridges2e	[null]	[null]	tbridges2e@tmall.com	399-866-3390	[null]	2xsKfU74...

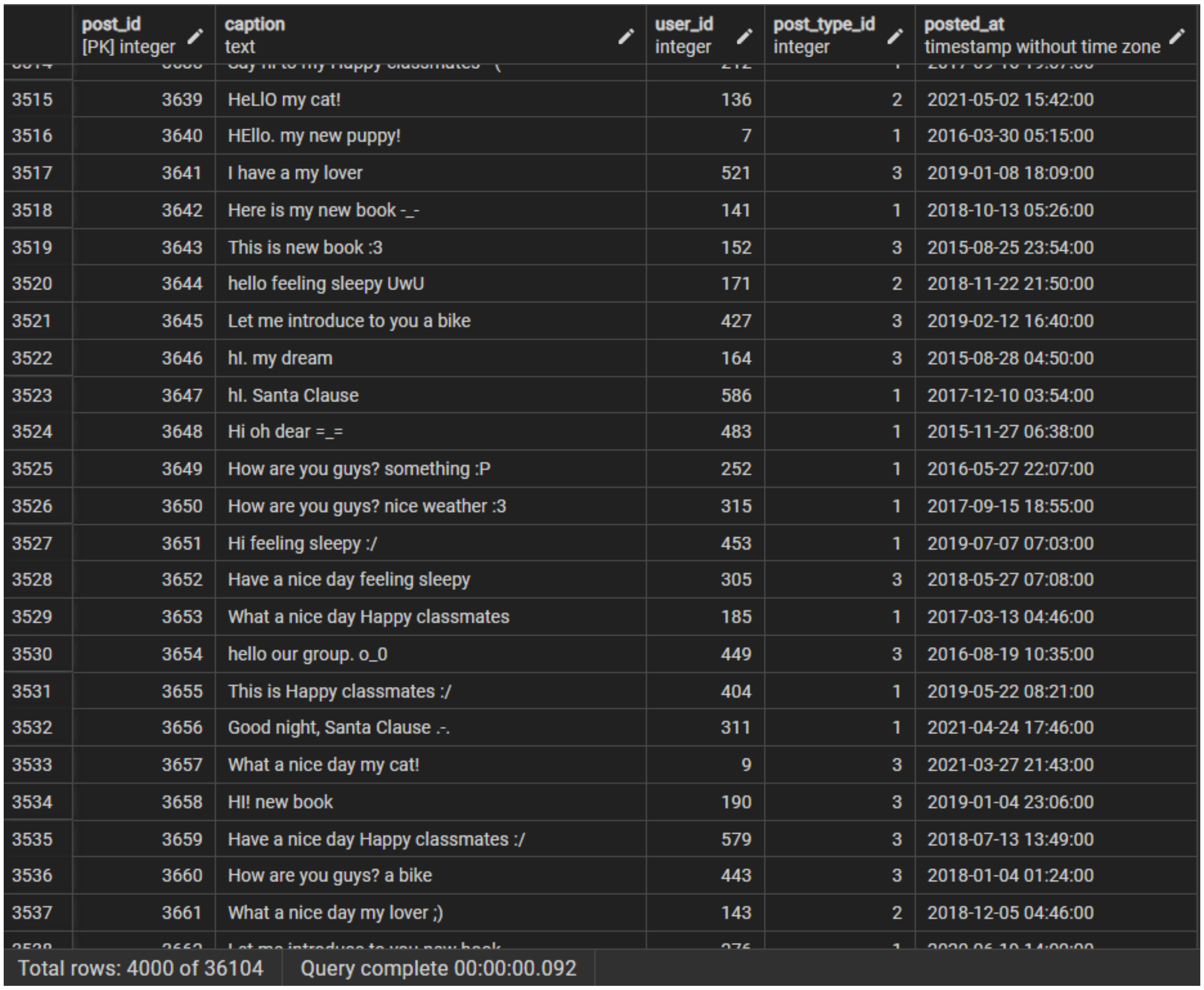
Total rows: 600 of 600 Query complete 00:00:00.262

Result of filling Post_type table:



	post_type_id [PK] integer	post_type_name character varying
1	1	reel
2	2	post
3	3	story

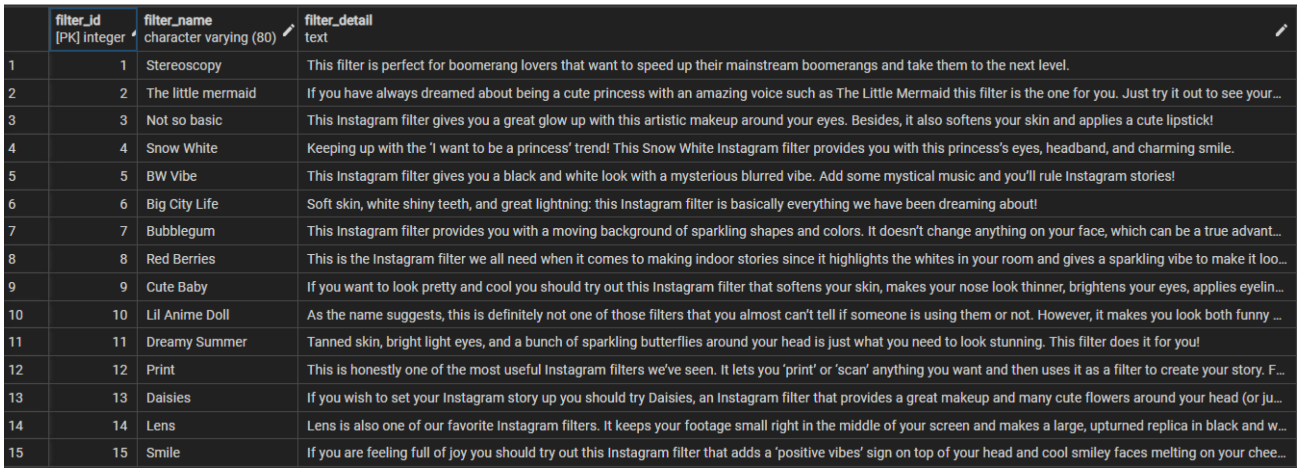
Result of filling Post table:



	post_id [PK] integer	caption text	user_id integer	post_type_id integer	posted_at timestamp without time zone
3515	3639	HeLlO my cat!	136	2	2021-05-02 15:42:00
3516	3640	HEllo. my new puppy!	7	1	2016-03-30 05:15:00
3517	3641	I have a my lover	521	3	2019-01-08 18:09:00
3518	3642	Here is my new book -_-	141	1	2018-10-13 05:26:00
3519	3643	This is new book :3	152	3	2015-08-25 23:54:00
3520	3644	hello feeling sleepy UwU	171	2	2018-11-22 21:50:00
3521	3645	Let me introduce to you a bike	427	3	2019-02-12 16:40:00
3522	3646	hI. my dream	164	3	2015-08-28 04:50:00
3523	3647	hI. Santa Clause	586	1	2017-12-10 03:54:00
3524	3648	Hi oh dear =_=	483	1	2015-11-27 06:38:00
3525	3649	How are you guys? something :P	252	1	2016-05-27 22:07:00
3526	3650	How are you guys? nice weather :3	315	1	2017-09-15 18:55:00
3527	3651	Hi feeling sleepy :/	453	1	2019-07-07 07:03:00
3528	3652	Have a nice day feeling sleepy	305	3	2018-05-27 07:08:00
3529	3653	What a nice day Happy classmates	185	1	2017-03-13 04:46:00
3530	3654	hello our group. o_0	449	3	2016-08-19 10:35:00
3531	3655	This is Happy classmates :/	404	1	2019-05-22 08:21:00
3532	3656	Good night, Santa Clause .-.	311	1	2021-04-24 17:46:00
3533	3657	What a nice day my cat!	9	3	2021-03-27 21:43:00
3534	3658	HI! new book	190	3	2019-01-04 23:06:00
3535	3659	Have a nice day Happy classmates :/	579	3	2018-07-13 13:49:00
3536	3660	How are you guys? a bike	443	3	2018-01-04 01:24:00
3537	3661	What a nice day my lover ;)	143	2	2018-12-05 04:46:00

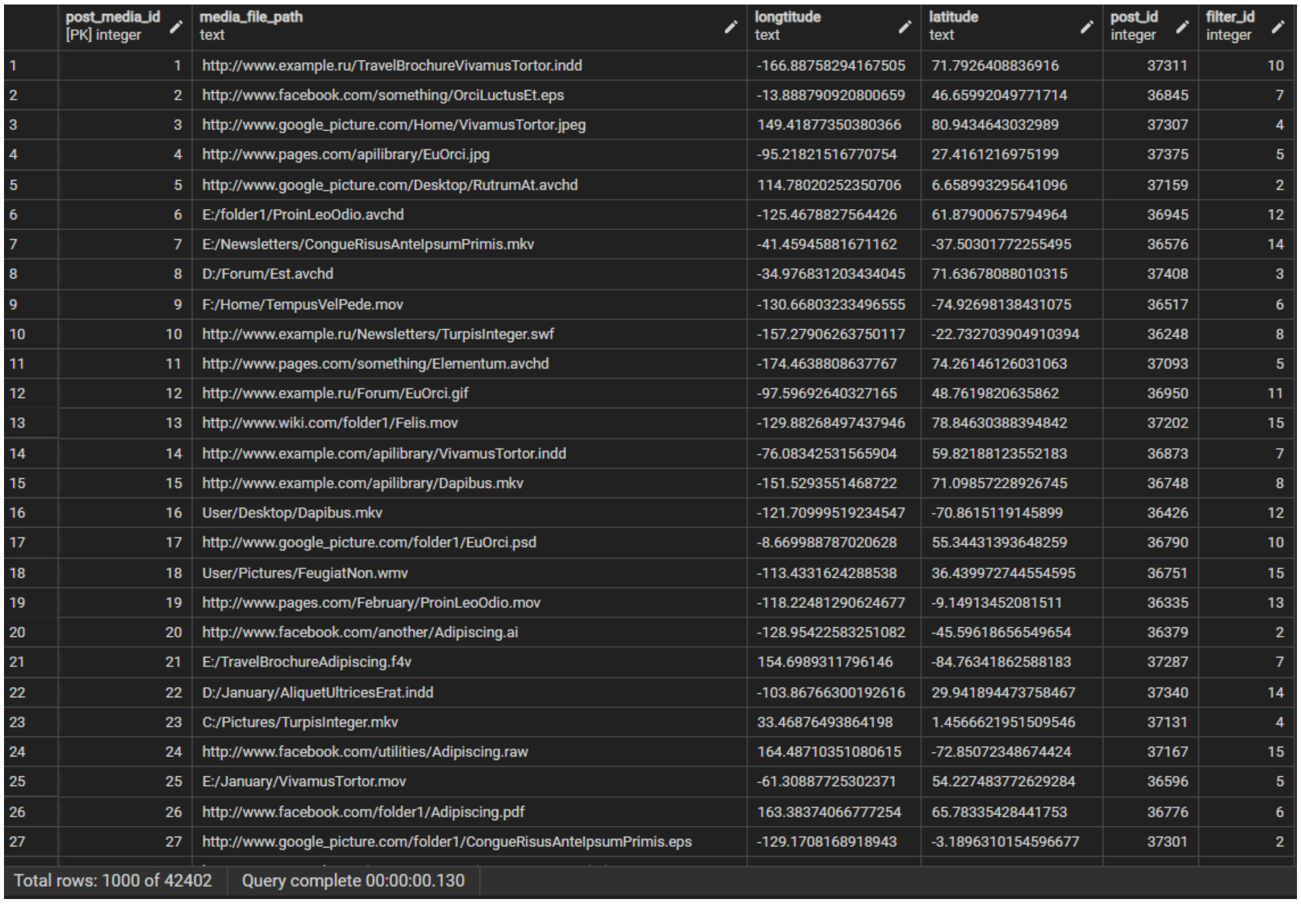
Total rows: 4000 of 36104   Query complete 00:00:00.092

Result of filling Filter table:



	filter_id [PK] integer	filter_name character varying (80)	filter_detail text
1	1	Stereoscopy	This filter is perfect for boomerang lovers that want to speed up their mainstream boomerangs and take them to the next level.
2	2	The little mermaid	If you have always dreamed about being a cute princess with an amazing voice such as The Little Mermaid this filter is the one for you. Just try it out to see your...
3	3	Not so basic	This Instagram filter gives you a great glow up with this artistic makeup around your eyes. Besides, it also softens your skin and applies a cute lipstick!
4	4	Snow White	Keeping up with the 'I want to be a princess' trend! This Snow White Instagram filter provides you with this princess's eyes, headband, and charming smile.
5	5	BW Vibe	This Instagram filter gives you a black and white look with a mysterious blurred vibe. Add some mystical music and you'll rule Instagram stories!
6	6	Big City Life	Soft skin, white shiny teeth, and great lightning: this Instagram filter is basically everything we have been dreaming about!
7	7	Bubblegum	This Instagram filter provides you with a moving background of sparkling shapes and colors. It doesn't change anything on your face, which can be a true advant...
8	8	Red Berries	This is the Instagram filter we all need when it comes to making indoor stories since it highlights the whites in your room and gives a sparkling vibe to make it loo...
9	9	Cute Baby	If you want to look pretty and cool you should try out this Instagram filter that softens your skin, makes your nose look thinner, brightens your eyes, applies eyelin...
10	10	Lil Anime Doll	As the name suggests, this is definitely not one of those filters that you almost can't tell if someone is using them or not. However, it makes you look both funny ...
11	11	Dreamy Summer	Tanned skin, bright light eyes, and a bunch of sparkling butterflies around your head is just what you need to look stunning. This filter does it for you!
12	12	Print	This is honestly one of the most useful Instagram filters we've seen. It lets you 'print' or 'scan' anything you want and then uses it as a filter to create your story. F...
13	13	Daisies	If you wish to set your Instagram story up you should try Daisies, an Instagram filter that provides a great makeup and many cute flowers around your head (or ju...
14	14	Lens	Lens is also one of our favorite Instagram filters. It keeps your footage small right in the middle of your screen and makes a large, upturned replica in black and w...
15	15	Smile	If you are feeling full of joy you should try out this Instagram filter that adds a 'positive vibes' sign on top of your head and cool smiley faces melting on your chee...

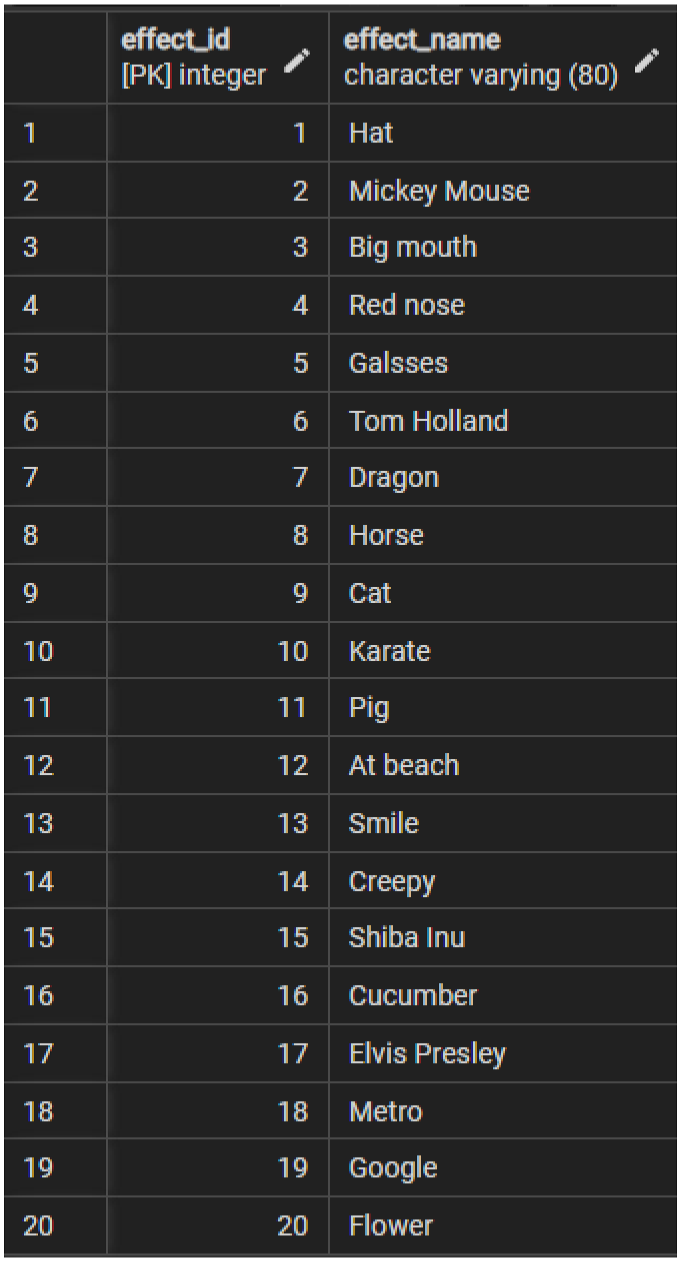
Result of filling Post_media table:



	post_media_id [PK] integer	media_file_path text	longtitude text	latitude text	post_id integer	filter_id integer
1	1	http://www.example.ru/TravelBrochureVivamusTortor.indd	-166.88758294167505	71.7926408836916	37311	10
2	2	http://www.facebook.com/something/OrciLuctusEt.eps	-13.888790920800659	46.65992049771714	36845	7
3	3	http://www.google_picture.com/Home/VivamusTortor.jpeg	149.41877350380366	80.9434643032989	37307	4
4	4	http://www.pages.com/apilibrary/EuOrci.jpg	-95.21821516770754	27.4161216975199	37375	5
5	5	http://www.google_picture.com/Desktop/RutrumAt.avchd	114.78020252350706	6.658993295641096	37159	2
6	6	E:/folder1/ProinLeoOdio.avchd	-125.4678827564426	61.87900675794964	36945	12
7	7	E:/Newsletters/CongueRisusAnteIpsumPrimis.mkv	-41.45945881671162	-37.50301772255495	36576	14
8	8	D:/Forum/Est.avchd	-34.976831203434045	71.63678088010315	37408	3
9	9	F:/Home/TempusVelPede.mov	-130.66803233496555	-74.92698138431075	36517	6
10	10	http://www.example.ru/Newsletters/TurpisInteger.swf	-157.27906263750117	-22.732703904910394	36248	8
11	11	http://www.pages.com/something/Elementum.avchd	-174.4638808637767	74.26146126031063	37093	5
12	12	http://www.example.ru/Forum/EuOrci.gif	-97.59692640327165	48.7619820635862	36950	11
13	13	http://www.wiki.com/folder1/Felis.mov	-129.88268497437946	78.84630388394842	37202	15
14	14	http://www.example.com/apilibrary/VivamusTortor.indd	-76.08342531565904	59.82188123552183	36873	7
15	15	http://www.example.com/apilibrary/Dapibus.mkv	-151.5293551468722	71.09857228926745	36748	8
16	16	User/Desktop/Dapibus.mkv	-121.70999519234547	-70.8615119145899	36426	12
17	17	http://www.google_picture.com/folder1/EuOrci.psd	-8.669988787020628	55.34431393648259	36790	10
18	18	User/Pictures/FeugiatNon.wmv	-113.4331624288538	36.439972744554595	36751	15
19	19	http://www.pages.com/February/ProinLeoOdio.mov	-118.22481290624677	-9.14913452081511	36335	13
20	20	http://www.facebook.com/another/Adipiscing.ai	-128.95422583251082	-45.59618656549654	36379	2
21	21	E:/TravelBrochureAdipiscing.f4v	154.6989311796146	-84.76341862588183	37287	7
22	22	D:/January/AliquetUltricesErat.indd	-103.86766300192616	29.941894473758467	37340	14
23	23	C:/Pictures/TurpisInteger.mkv	33.46876493864198	1.4566621951509546	37131	4
24	24	http://www.facebook.com/utilities/Adipiscing.raw	164.48710351080615	-72.85072348674424	37167	15
25	25	E:/January/VivamusTortor.mov	-61.30887725302371	54.227483772629284	36596	5
26	26	http://www.facebook.com/folder1/Adipiscing.pdf	163.38374066777254	65.78335428441753	36776	6
27	27	http://www.google_picture.com/folder1/CongueRisusAnteIpsumPrimis.eps	-129.1708168918943	-3.1896310154596677	37301	2

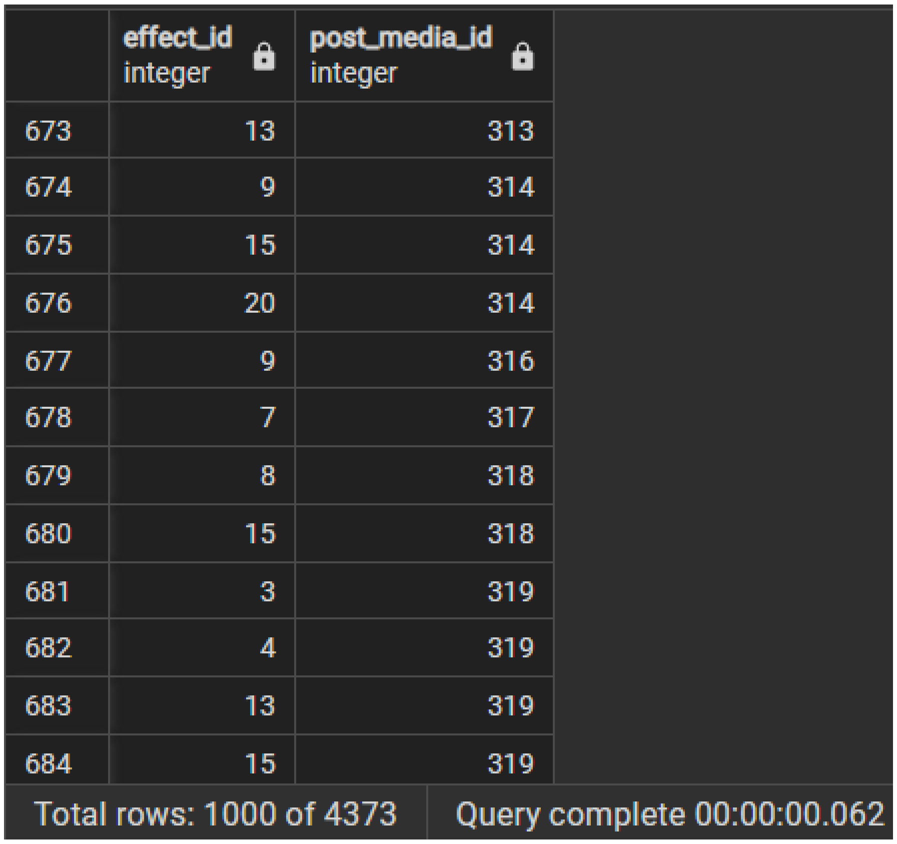
Total rows: 1000 of 42402 Query complete 00:00:00.130

Result of filling Effect table:



	effect_id [PK] integer	effect_name character varying (80)
1	1	Hat
2	2	Mickey Mouse
3	3	Big mouth
4	4	Red nose
5	5	Galsses
6	6	Tom Holland
7	7	Dragon
8	8	Horse
9	9	Cat
10	10	Karate
11	11	Pig
12	12	At beach
13	13	Smile
14	14	Creepy
15	15	Shiba Inu
16	16	Cucumber
17	17	Elvis Presley
18	18	Metro
19	19	Google
20	20	Flower

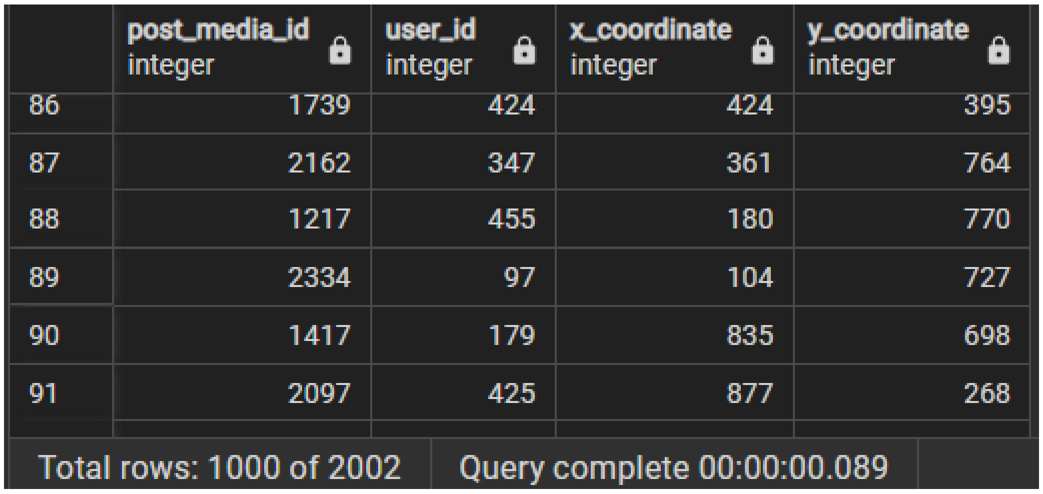
Result of filling Effect_Post_media table:



	effect_id integer	post_media_id integer
673	13	313
674	9	314
675	15	314
676	20	314
677	9	316
678	7	317
679	8	318
680	15	318
681	3	319
682	4	319
683	13	319
684	15	319

Total rows: 1000 of 4373 Query complete 00:00:00.062

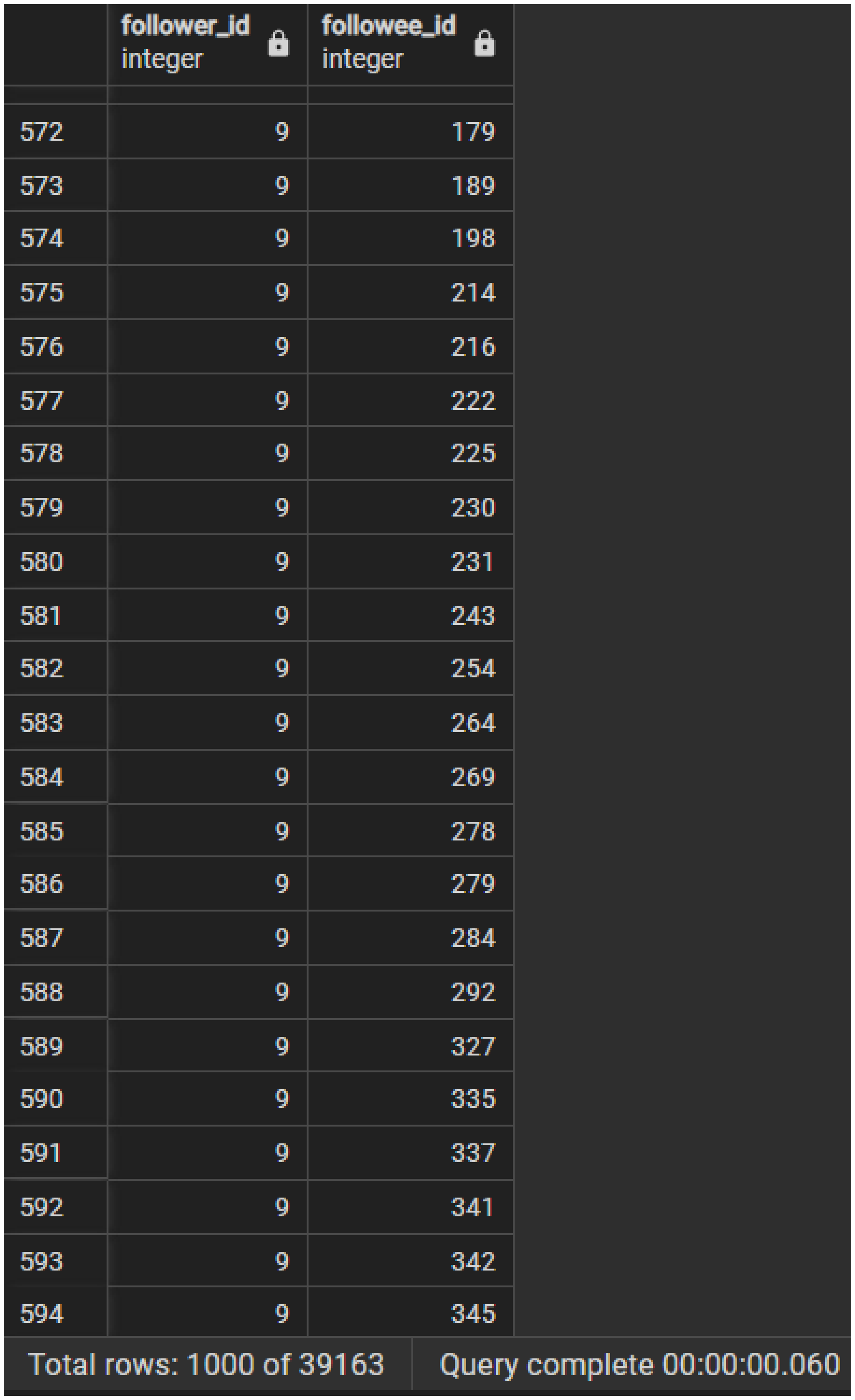
Result of filling Post_media_User_tag table:



	post_media_id integer	user_id integer	x_coordinate integer	y_coordinate integer
86	1739	424	424	395
87	2162	347	361	764
88	1217	455	180	770
89	2334	97	104	727
90	1417	179	835	698
91	2097	425	877	268

Total rows: 1000 of 2002 Query complete 00:00:00.089

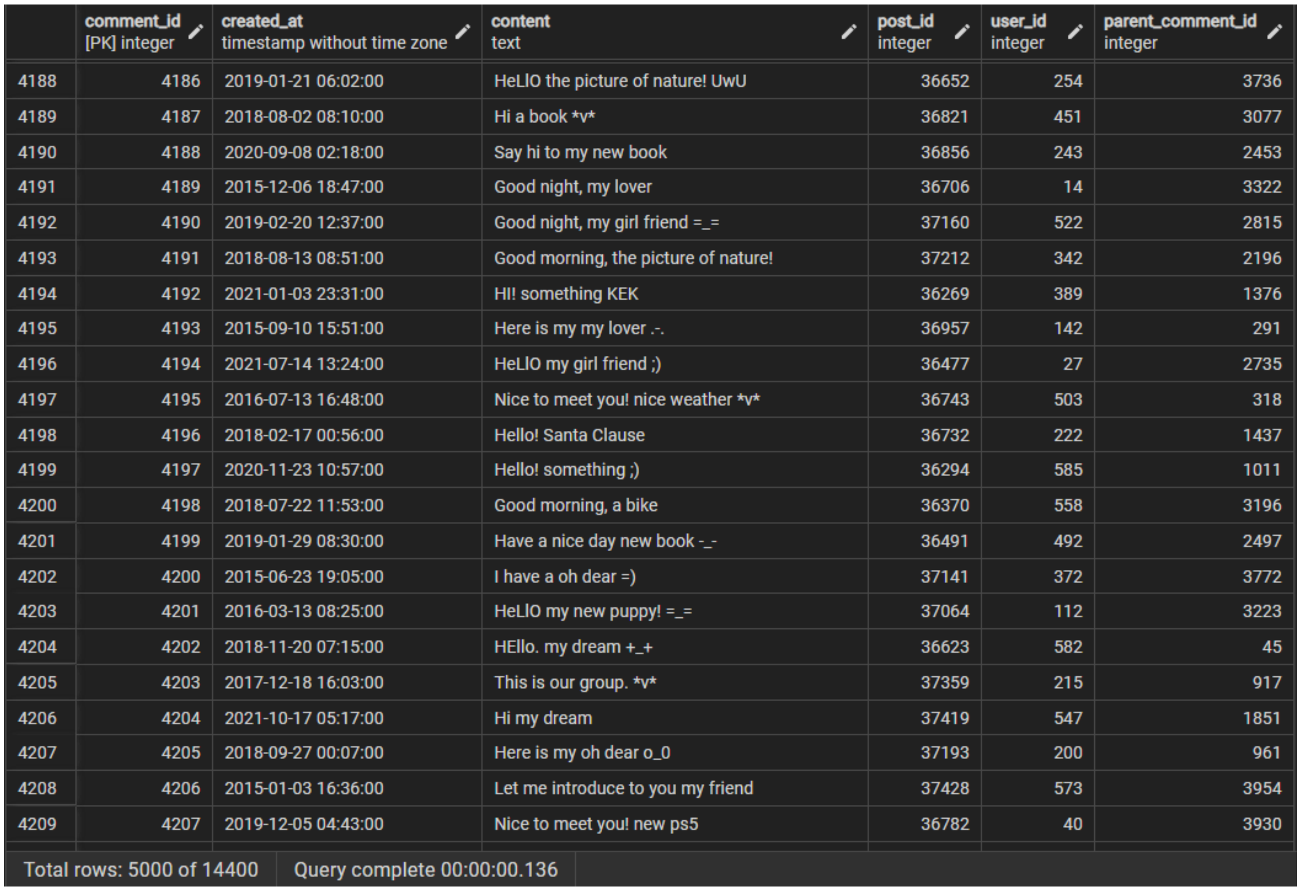
Result of filling Follower table:



	follower_id integer	followee_id integer
572	9	179
573	9	189
574	9	198
575	9	214
576	9	216
577	9	222
578	9	225
579	9	230
580	9	231
581	9	243
582	9	254
583	9	264
584	9	269
585	9	278
586	9	279
587	9	284
588	9	292
589	9	327
590	9	335
591	9	337
592	9	341
593	9	342
594	9	345

Total rows: 1000 of 39163 Query complete 00:00:00.060

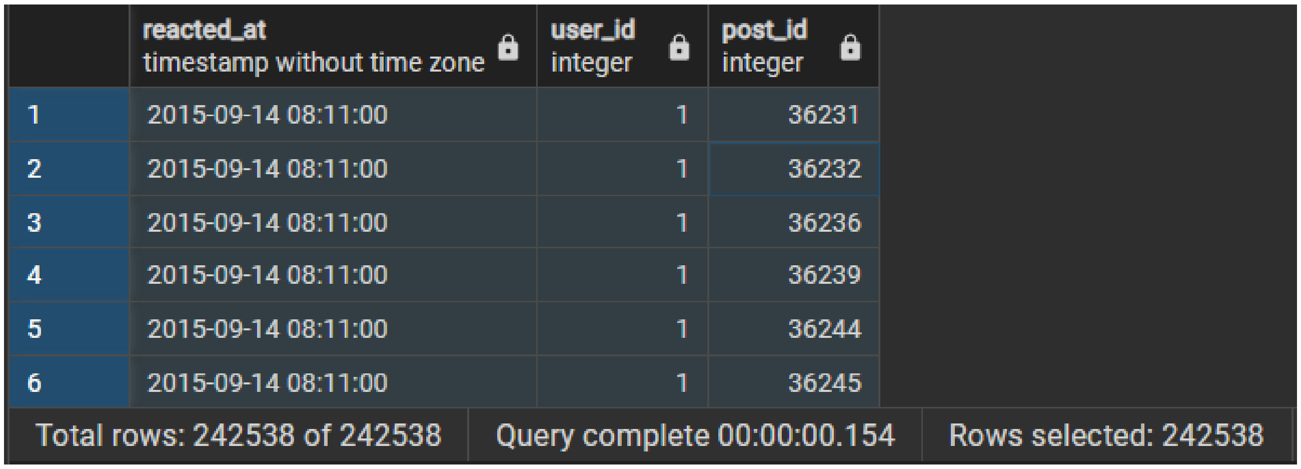
Result of filling Comment table:



	comment_id [PK] integer	created_at timestamp without time zone	content text	post_id integer	user_id integer	parent_comment_id integer
4188	4186	2019-01-21 06:02:00	HeLlO the picture of nature! UwU	36652	254	3736
4189	4187	2018-08-02 08:10:00	Hi a book *v*	36821	451	3077
4190	4188	2020-09-08 02:18:00	Say hi to my new book	36856	243	2453
4191	4189	2015-12-06 18:47:00	Good night, my lover	36706	14	3322
4192	4190	2019-02-20 12:37:00	Good night, my girl friend =_=	37160	522	2815
4193	4191	2018-08-13 08:51:00	Good morning, the picture of nature!	37212	342	2196
4194	4192	2021-01-03 23:31:00	HI! something KEK	36269	389	1376
4195	4193	2015-09-10 15:51:00	Here is my my lover .-.	36957	142	291
4196	4194	2021-07-14 13:24:00	HeLlO my girl friend ;)	36477	27	2735
4197	4195	2016-07-13 16:48:00	Nice to meet you! nice weather *v*	36743	503	318
4198	4196	2018-02-17 00:56:00	Hello! Santa Clause	36732	222	1437
4199	4197	2020-11-23 10:57:00	Hello! something ;)	36294	585	1011
4200	4198	2018-07-22 11:53:00	Good morning, a bike	36370	558	3196
4201	4199	2019-01-29 08:30:00	Have a nice day new book -_-	36491	492	2497
4202	4200	2015-06-23 19:05:00	I have a oh dear =)	37141	372	3772
4203	4201	2016-03-13 08:25:00	HeLlO my new puppy! =_=	37064	112	3223
4204	4202	2018-11-20 07:15:00	HEllo. my dream +_+	36623	582	45
4205	4203	2017-12-18 16:03:00	This is our group. *v*	37359	215	917
4206	4204	2021-10-17 05:17:00	Hi my dream	37419	547	1851
4207	4205	2018-09-27 00:07:00	Here is my oh dear o_0	37193	200	961
4208	4206	2015-01-03 16:36:00	Let me introduce to you my friend	37428	573	3954
4209	4207	2019-12-05 04:43:00	Nice to meet you! new ps5	36782	40	3930

Total rows: 5000 of 14400 Query complete 00:00:00.136

Result of filling Reaction table:



	reacted_at timestamp without time zone	user_id integer	post_id integer
1	2015-09-14 08:11:00	1	36231
2	2015-09-14 08:11:00	1	36232
3	2015-09-14 08:11:00	1	36236
4	2015-09-14 08:11:00	1	36239
5	2015-09-14 08:11:00	1	36244
6	2015-09-14 08:11:00	1	36245

Total rows: 242538 of 242538 Query complete 00:00:00.154 Rows selected: 242538

5. Queries
   a. Easy
      i. Find post_type of the post with post_id 37000.

```
select post_type_name from post_type
Inner join  post on post.post_type_id = post_type.post_ty
where post_id = 37000;
```

      ii. Find all posts which are posted after 2018.

```
select post_id, content from comment
where 'timestamp' > '2018-01-01 00:00:00';
```

      iii. Find comments which mention about “Santa Clause”.

```
select content from comment
where content like '%Santa Clause%';
```

   b. Medium
      i. Find average, max, min number of followers for each user.

```
with count_ as (select count(*) as a from follower group
follower_id)
select avg(a), max(a), min(a) from count_; --??
```

      ii. Count the number of posts created by each user.

```
select user_id as user , count(post_id) as _cnt from post
group by user_id
Union
select user_id, 0 from App_user
where user_id not in (select distinct user_id from post);
```

      iii. Modify the second query to show only those users who created more than 5 posts.

```
select tbl.user from (select user_id as user , count(post
_cnt from post
group by user_id) as tbl
where _cnt >5;
```

c. Hard

  i. Find the most valuable comment in a post

  ii. Group users to categories:

```
SELECT 'Celebrity' type, 85 min_follows, 9999999 max_follows
UNION ALL
SELECT 'Influencer' type, 70 min_follows, 84 max_follows
UNION ALL
SELECT 'Normal user' type, 0 min_follows, 69 max_follows
```

The subquery against the app_user table should count the number of rows for each user using group by user_id, and the count should be compared to the min_follows/max_follows columns to determine which type each user belongs to.

  iii. Sort users by the total number of reactions their posts have.

  iv. Show number of comment of the most liked post.

```
with likes as (select post.post_id, count ('timestamp') a
from post
join reaction as r on r.post_id = post.post_id
group by post.post_id
order by num desc
limit 10)

select post.post_id, likes.num as num_of_likes, count
(c.comment_id) as comments from post
join comment as c on c.post_id = post.post_id
join likes on likes.post_id = post.post_id
where post.post_id in (select post_id from likes)
group by post.post_id, likes.num
order by num_of_likes desc;
```

d. Solution

```sql
--MEDIUM 2
SELECT au.user_id, count(*) FROM app_user au
INNER JOIN post p
ON au.user_id = p.user_id
GROUP BY au.user_id ORDER BY 2 DESC;
```

```sql
--MEDIUM 3
SELECT au.user_id, count(*) FROM app_user au
INNER JOIN post p
ON au.user_id = p.user_id
GROUP BY au.user_id
HAVING count(*) > 5
ORDER BY 2 DESC;
```

```sql
--HARD 1
DELETE FROM comment c
WHERE c.created_at < (SELECT p.posted_at FROM post p WHERE p.post_id =
c.post_id);
```

```sql
--HARD 2
WITH u_numf AS
(SELECT au.user_id, count(*) numf FROM app_user au
INNER JOIN follower f ON au.user_id = f.followee_id
GROUP BY au.user_id)
SELECT max(u_numf.numf), min(u_numf.numf), avg(u_numf.numf) FROM u_numf;

SELECT max(u_numf.numf), min(u_numf.numf), avg(u_numf.numf)
FROM (SELECT au.user_id, count(*) numf FROM app_user au
INNER JOIN follower f ON au.user_id = f.followee_id
GROUP BY au.user_id) u_numf;
```

```
--HARD 3
SELECT grp.type, count(*) num_user FROM
(SELECT au.user_id, count(*) numf FROM app_user au
INNER JOIN follower f ON au.user_id = f.followee_id
GROUP BY au.user_id) u_numf
INNER JOIN
(SELECT 'Celebrity' type, 85 min_follows, 9999999 max_follows
UNION ALL
SELECT 'Influencer' type, 70 min_follows, 84 max_follows
UNION ALL
SELECT 'Normal user' type, 0 min_follows, 69 max_follows) grp
ON u_numf.numf BETWEEN grp.min_follows AND grp.max_follows
GROUP BY grp.type;

SELECT u_numf.user_id,
(SELECT au.first FROM app_user au WHERE u_numf.user_id = au.user_id) first_r
(SELECT au.middle FROM app_user au WHERE u_numf.user_id = au.user_id)
middle_name,
(SELECT au.last FROM app_user au WHERE u_numf.user_id = au.user_id) last_nar
grp.type
FROM
(
    SELECT au.user_id, count(*) numf FROM app_user au
    INNER JOIN follower f ON au.user_id = f.followee_id
    GROUP BY au.user_id
) u_numf
INNER JOIN
(
    SELECT 'Celebrity' type, 85 min_follows, 9999999 max_follows
    UNION ALL
    SELECT 'Influencer' type, 70 min_follows, 84 max_follows
    UNION ALL
    SELECT 'Normal user' type, 0 min_follows, 69 max_follows
) grp
ON u_numf.numf BETWEEN grp.min_follows AND grp.max_follows;
```

```sql
--HARD 4
SELECT au.user_id, au.first, au.last, count(*) number_of_post,
sum(post_nrp.num_react_post) total_react_on_post
FROM app_user au
INNER JOIN
(SELECT p.user_id, p.post_id, count(*) num_react_post
FROM post p
INNER JOIN reaction r ON p.post_id = r.post_id
GROUP BY p.post_id) post_nrp
ON au.user_id = post_nrp.user_id
GROUP BY au.user_id
ORDER BY 5 DESC;
```

```sql
--HARD 5
WITH p_num_react AS
(
    SELECT p.post_id, count(*) num_react
    FROM post p
    INNER JOIN reaction r
    ON p.post_id = r.post_id
    GROUP BY p.post_id
),
most_reaction_post AS
(
    SELECT p.post_id FROM p_num_react p
    WHERE p.num_react = (SELECT max(num_react) FROM p_num_react)
)
SELECT count(*) num_comment
FROM most_reaction_post mrp
INNER JOIN post p
ON p.post_id = mrp.post_id
INNER JOIN comment c
ON c.post_id = p.post_id
GROUP BY mrp.post_id;
```