Group 16 - COMP 421 - Database Systems
Demetrios Koziris (260584555)
Sami Riaz (260630430)
Brennan Nichyporuk (260504951)
Vivien Traineau (260623886)

COMP 421 Project Deliverable 2

Please note that for this deliverable we created bash scripts that take our sql files, execute all the assignment questions, and output logs. These scripts as well as the logs have been included in the submission as full proof of our work and are referenced in this document. They can also be found at the following github repo: comp421-project.

Question 1:

We did not have corrections from feedback on part 1
For reference, here is our relational model:

Creator(handle, wall_id)(wall_id ref Wall)
User(user_handle, display_name, email, firstname, lastname, gender, birthdate) (user_handle ref Creator(handle))
Page(page_handle, description, title, follower_count, user_handle) (page_handle ref Creator(handle)) (user_handle ref User)
Submission(submission_id, text_content, like_count, creation_time)
Comment(comment_id, user_handle, post_id) (comment_id ref Submission(submission_id)) (user_handle ref User) (post_id ref Post)
Post(post_id, view_count, attachment_link, image, handle, wall_id) (post_id ref Submission(submission_id)) (handle ref Creator) (wall_id ref Wall)
Event(event_id, handle, wall_id, creation_time, start_time, end_time, title, description, location) (handle ref Creator) (wall_id ref Wall)
Wall(wall_id, description)
Notification(notif_id, read_status, notif_link, notif_text, notif_time, user_handle) (user_handle ref User)
PageFollower(user_handle, page_handle) (user_handle ref User) (page_handle ref Page)
EventSubscription(user_handle, event_id, rsvp_status)(user_handle ref User)(event_id ref Event)
SubmissionLike(user_handle, submission_id) (user_handle ref User) (submission_id ref Submission)
FeedView(user_handle, post_id) (user_handle ref User) (post_id ref Post)
UserFriend(user_handle, friend_handle) (user_handle ref User**)** (friend_handle ref User(user_handle)**)**

Question 2:

Table Creation statements:

Please see **`part2.log`** for a complete log of the table creation

wall:

```
CREATE TABLE wall(
      wall_id SERIAL,
      description VARCHAR(1023),
      PRIMARY KEY(wall_id)
);
```

creator:

```
CREATE TABLE creator(
      handle VARCHAR(255),
      wall_id INTEGER UNIQUE NOT NULL,
      PRIMARY KEY (handle),
      FOREIGN KEY (wall_id) REFERENCES wall
);
```

usert:

```
CREATE TABLE usert(
      usert_handle VARCHAR(255),
      display_name VARCHAR(255),
      email VARCHAR(255) UNIQUE,
      firstname VARCHAR(255),
      lastname VARCHAR(255),
      gender VARCHAR(255),
      birth_date TIMESTAMP,
      PRIMARY KEY (usert_handle),
      FOREIGN KEY (usert_handle) REFERENCES creator(handle)
);
```

page:

```
CREATE TABLE page(
      page_handle VARCHAR(255),
      description VARCHAR(1023),
      title VARCHAR(255),
      follower_count INTEGER NOT NULL DEFAULT 0,
      usert_handle VARCHAR(255) NOT NULL,
      PRIMARY KEY (page_handle),
      FOREIGN KEY (page_handle) REFERENCES creator(handle),
      FOREIGN KEY (usert_handle) REFERENCES usert
);
```

submission:

```
CREATE TABLE submission(
      submission_id SERIAL,
      text_content VARCHAR(8000),
      like_count INTEGER NOT NULL DEFAULT 0,
      creation_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
      PRIMARY KEY (submission_id)
);
```

post:

```
CREATE TABLE post(
      post_id INTEGER,
      view_count INTEGER NOT NULL DEFAULT 0,
      attachment_link VARCHAR(2083),
      image VARCHAR(2083),
      handle VARCHAR(255) NOT NULL,
      wall_id INTEGER NOT NULL,
      PRIMARY KEY(post_id),
      FOREIGN KEY(post_id) REFERENCES submission(submission_id),
      FOREIGN KEY(handle) REFERENCES creator,
      FOREIGN KEY(wall_id) REFERENCES wall
);
```

commentt:

```
CREATE TABLE commentt(
      commentt_id INTEGER,
      usert_handle VARCHAR(255) NOT NULL,
      post_id INTEGER NOT NULL,
      PRIMARY KEY (commentt_id),
      FOREIGN KEY (commentt_id) REFERENCES submission(submission_id),
      FOREIGN KEY (usert_handle) REFERENCES usert,
      FOREIGN KEY (post_id) REFERENCES post
);
```

eventt:

```
CREATE TABLE eventt(
      eventt_id SERIAL,
      handle VARCHAR(255) NOT NULL,
      wall_id INTEGER UNIQUE NOT NULL,
      creation_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
      start_time TIMESTAMP NOT NULL,
      end_time TIMESTAMP NOT NULL,
      title VARCHAR(255) NOT NULL,
      description VARCHAR(1023),
      location VARCHAR(255),
      PRIMARY KEY(eventt_id),
      FOREIGN KEY(handle) REFERENCES creator,
      FOREIGN KEY(wall_id) REFERENCES wall
);
```

notification:

```
CREATE TABLE notification(
      notif_id SERIAL,
      read_status BOOLEAN,
      notif_link VARCHAR(2083),
      notif_text VARCHAR(1023),
      notif_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
      usert_handle VARCHAR(255) NOT NULL,
      PRIMARY KEY (notif_id),
      FOREIGN KEY (usert_handle) REFERENCES usert
);
```

page_follower:

```
CREATE TABLE page_follower(
      usert_handle VARCHAR(255),
      page_handle VARCHAR(255),
      PRIMARY KEY(usert_handle, page_handle),
      FOREIGN KEY(usert_handle) REFERENCES usert,
      FOREIGN KEY(page_handle) REFERENCES page
);
```

eventt_subscription:

```
CREATE TABLE eventt_subscription(
      usert_handle VARCHAR(255),
      eventt_id INTEGER,
      rsvp_status VARCHAR(255),
      PRIMARY KEY(usert_handle, eventt_id),
      FOREIGN KEY(usert_handle) REFERENCES usert,
      FOREIGN KEY(eventt_id) REFERENCES eventt
);
```

submission_like:

```
CREATE TABLE submission_like(
      usert_handle VARCHAR(255),
      submission_id INTEGER,
      PRIMARY KEY(usert_handle, submission_id),
      FOREIGN KEY(usert_handle) REFERENCES usert,
      FOREIGN KEY(submission_id) REFERENCES submission
);
```

feed_view:

```
CREATE TABLE feed_view(
      usert_handle VARCHAR(255),
      post_id INTEGER,
      PRIMARY KEY(usert_handle, post_id),
      FOREIGN KEY(usert_handle) REFERENCES usert,
      FOREIGN KEY(post_id) REFERENCES post
);
```

usert_friend:

```
CREATE TABLE usert_friend(
      usert_handle VARCHAR(255),
      friend_handle VARCHAR(255),
      PRIMARY KEY(usert_handle, friend_handle),
      FOREIGN KEY(usert_handle) REFERENCES usert,
      FOREIGN KEY(friend_handle) REFERENCES usert(usert_handle),
      CHECK (usert_handle <> friend_handle)
);
```

Table Descriptions:

Please see **`part2.log`** for a complete log of the table creation.

wall:

```
                                    Table "cs421g16.wall"
   Column    |          Type           |                       Modifiers
-------------+-------------------------+-----------------------------------------
------------------
 wall_id     | integer                 | not null default
nextval('wall_wall_id_seq'::regclass)
 description | character varying(1023) |
Indexes:
    "wall_pkey" PRIMARY KEY, btree (wall_id)
```

```
Referenced by:
    TABLE "creator" CONSTRAINT "creator_wall_id_fkey" FOREIGN KEY (wall_id)
REFERENCES wall(wall_id)
    TABLE "eventt" CONSTRAINT "eventt_wall_id_fkey" FOREIGN KEY (wall_id)
REFERENCES wall(wall_id)
    TABLE "post" CONSTRAINT "post_wall_id_fkey" FOREIGN KEY (wall_id)
REFERENCES wall(wall_id)
```

creator:

```
           Table "cs421g16.creator"
 Column  |          Type          | Modifiers
---------+------------------------+-----------
 handle  | character varying(255) | not null
 wall_id | integer                | not null
Indexes:
    "creator_pkey" PRIMARY KEY, btree (handle)
    "creator_wall_id_key" UNIQUE CONSTRAINT, btree (wall_id)
Foreign-key constraints:
    "creator_wall_id_fkey" FOREIGN KEY (wall_id) REFERENCES wall(wall_id)
Referenced by:
    TABLE "eventt" CONSTRAINT "eventt_handle_fkey" FOREIGN KEY (handle)
REFERENCES creator(handle)
    TABLE "page" CONSTRAINT "page_page_handle_fkey" FOREIGN KEY (page_handle)
REFERENCES creator(handle)
    TABLE "post" CONSTRAINT "post_handle_fkey" FOREIGN KEY (handle) REFERENCES
creator(handle)
    TABLE "usert" CONSTRAINT "usert_usert_handle_fkey" FOREIGN KEY
(usert_handle) REFERENCES creator(handle)
```

usert:

```
                 Table "cs421g16.usert"
    Column    |            Type             | Modifiers
--------------+-----------------------------+-----------
 usert_handle | character varying(255)      | not null
 display_name | character varying(255)      |
 email        | character varying(255)      |
 firstname    | character varying(255)      |
 lastname     | character varying(255)      |
 gender       | character varying(255)      |
 birth_date   | timestamp without time zone |
Indexes:
    "usert_pkey" PRIMARY KEY, btree (usert_handle)
    "usert_email_key" UNIQUE CONSTRAINT, btree (email)
Foreign-key constraints:
    "usert_usert_handle_fkey" FOREIGN KEY (usert_handle) REFERENCES
```

```
creator(handle)
Referenced by:
    TABLE "commentt" CONSTRAINT "commentt_usert_handle_fkey" FOREIGN KEY
(usert_handle) REFERENCES usert(usert_handle)
    TABLE "eventt_subscription" CONSTRAINT
"eventt_subscription_usert_handle_fkey" FOREIGN KEY (usert_handle) REFERENCES
usert(usert_handle)
    TABLE "feed_view" CONSTRAINT "feed_view_usert_handle_fkey" FOREIGN KEY
(usert_handle) REFERENCES usert(usert_handle)
    TABLE "notification" CONSTRAINT "notification_usert_handle_fkey" FOREIGN
KEY (usert_handle) REFERENCES usert(usert_handle)
    TABLE "page_follower" CONSTRAINT "page_follower_usert_handle_fkey" FOREIGN
KEY (usert_handle) REFERENCES usert(usert_handle)
    TABLE "page" CONSTRAINT "page_usert_handle_fkey" FOREIGN KEY
(usert_handle) REFERENCES usert(usert_handle)
    TABLE "submission_like" CONSTRAINT "submission_like_usert_handle_fkey"
FOREIGN KEY (usert_handle) REFERENCES usert(usert_handle)
    TABLE "usert_friend" CONSTRAINT "usert_friend_friend_handle_fkey" FOREIGN
KEY (friend_handle) REFERENCES usert(usert_handle)
    TABLE "usert_friend" CONSTRAINT "usert_friend_usert_handle_fkey" FOREIGN
KEY (usert_handle) REFERENCES usert(usert_handle)
```

page:

```
                     Table "cs421g16.page"
     Column     |          Type           |     Modifiers
----------------+-------------------------+--------------------
 page_handle    | character varying(255)  | not null
 description    | character varying(1023) |
 title          | character varying(255)  |
 follower_count | integer                 | not null default 0
 usert_handle   | character varying(255)  | not null
Indexes:
    "page_pkey" PRIMARY KEY, btree (page_handle)
Foreign-key constraints:
    "page_page_handle_fkey" FOREIGN KEY (page_handle) REFERENCES
creator(handle)
    "page_usert_handle_fkey" FOREIGN KEY (usert_handle) REFERENCES
usert(usert_handle)
Referenced by:
    TABLE "page_follower" CONSTRAINT "page_follower_page_handle_fkey" FOREIGN
KEY (page_handle) REFERENCES page(page_handle)
```

submission:

```
                                           Table "cs421g16.submission"
    Column     |            Type              |
Modifiers
---------------+-----------------------------+----------------------------------
------------------------------------
 submission_id | integer                     | not null default
nextval('submission_submission_id_seq'::regclass)
 text_content  | character varying(8000)     |
 like_count    | integer                     | not null default 0
 creation_time | timestamp without time zone | default now()
Indexes:
    "submission_pkey" PRIMARY KEY, btree (submission_id)
Referenced by:
    TABLE "commentt" CONSTRAINT "commentt_commentt_id_fkey" FOREIGN KEY
(commentt_id) REFERENCES submission(submission_id)
    TABLE "post" CONSTRAINT "post_post_id_fkey" FOREIGN KEY (post_id)
REFERENCES submission(submission_id)
    TABLE "submission_like" CONSTRAINT "submission_like_submission_id_fkey"
FOREIGN KEY (submission_id) REFERENCES submission(submission_id)
```

post:

```
                     Table "cs421g16.post"
     Column      |          Type           |     Modifiers
-----------------+-------------------------+--------------------
 post_id         | integer                 | not null
 view_count      | integer                 | not null default 0
 attachment_link | character varying(2083) |
 image           | character varying(2083) |
 handle          | character varying(255)  | not null
 wall_id         | integer                 | not null
Indexes:
    "post_pkey" PRIMARY KEY, btree (post_id)
Foreign-key constraints:
    "post_handle_fkey" FOREIGN KEY (handle) REFERENCES creator(handle)
    "post_post_id_fkey" FOREIGN KEY (post_id) REFERENCES
submission(submission_id)
    "post_wall_id_fkey" FOREIGN KEY (wall_id) REFERENCES wall(wall_id)
Referenced by:
    TABLE "commentt" CONSTRAINT "commentt_post_id_fkey" FOREIGN KEY (post_id)
REFERENCES post(post_id)
    TABLE "feed_view" CONSTRAINT "feed_view_post_id_fkey" FOREIGN KEY
(post_id) REFERENCES post(post_id)
```

commentt:

```
             Table "cs421g16.commentt"
    Column    |          Type          | Modifiers
--------------+------------------------+-----------
```

```
 commentt_id  | integer                | not null
 usert_handle | character varying(255) | not null
 post_id      | integer                | not null
Indexes:
    "commentt_pkey" PRIMARY KEY, btree (commentt_id)
Foreign-key constraints:
    "commentt_commentt_id_fkey" FOREIGN KEY (commentt_id) REFERENCES
submission(submission_id)
    "commentt_post_id_fkey" FOREIGN KEY (post_id) REFERENCES post(post_id)
    "commentt_usert_handle_fkey" FOREIGN KEY (usert_handle) REFERENCES
usert(usert_handle)
```

eventt:

```
                                         Table "cs421g16.eventt"
    Column     |            Type             |
Modifiers
---------------+-----------------------------+---------------------------------
----------------------------
 eventt_id     | integer                     | not null default
nextval('eventt_eventt_id_seq'::regclass)
 handle        | character varying(255)      | not null
 wall_id       | integer                     | not null
 creation_time | timestamp without time zone | default now()
 start_time    | timestamp without time zone | not null
 end_time      | timestamp without time zone | not null
 title         | character varying(255)      | not null
 description   | character varying(1023)     |
 location      | character varying(255)      |
Indexes:
    "eventt_pkey" PRIMARY KEY, btree (eventt_id)
    "eventt_wall_id_key" UNIQUE CONSTRAINT, btree (wall_id)
Foreign-key constraints:
    "eventt_handle_fkey" FOREIGN KEY (handle) REFERENCES creator(handle)
    "eventt_wall_id_fkey" FOREIGN KEY (wall_id) REFERENCES wall(wall_id)
Referenced by:
    TABLE "eventt_subscription" CONSTRAINT
"eventt_subscription_eventt_id_fkey" FOREIGN KEY (eventt_id) REFERENCES
eventt(eventt_id)
```

notification:

```
                                        Table "cs421g16.notification"
    Column    |            Type             |
Modifiers
--------------+-----------------------------+---------------------------------
--------------------------------
```

```
 notif_id     | integer                     | not null default
nextval('notification_notif_id_seq'::regclass)
 read_status  | boolean                     |
 notif_link   | character varying(2083)     |
 notif_text   | character varying(1023)     |
 notif_time   | timestamp without time zone | default now()
 usert_handle | character varying(255)      | not null
Indexes:
    "notification_pkey" PRIMARY KEY, btree (notif_id)
Foreign-key constraints:
    "notification_usert_handle_fkey" FOREIGN KEY (usert_handle) REFERENCES
usert(usert_handle)
```

page_follower:

```
          Table "cs421g16.page_follower"
    Column    |          Type          | Modifiers
--------------+------------------------+-----------
 usert_handle | character varying(255) | not null
 page_handle  | character varying(255) | not null
Indexes:
    "page_follower_pkey" PRIMARY KEY, btree (usert_handle, page_handle)
Foreign-key constraints:
    "page_follower_page_handle_fkey" FOREIGN KEY (page_handle) REFERENCES
page(page_handle)
    "page_follower_usert_handle_fkey" FOREIGN KEY (usert_handle) REFERENCES
usert(usert_handle)
```

eventt_subscription:

```
       Table "cs421g16.eventt_subscription"
    Column    |          Type          | Modifiers
--------------+------------------------+-----------
 usert_handle | character varying(255) | not null
 eventt_id    | integer                | not null
 rsvp_status  | character varying(255) |
Indexes:
    "eventt_subscription_pkey" PRIMARY KEY, btree (usert_handle, eventt_id)
Foreign-key constraints:
    "eventt_subscription_eventt_id_fkey" FOREIGN KEY (eventt_id) REFERENCES
eventt(eventt_id)
    "eventt_subscription_usert_handle_fkey" FOREIGN KEY (usert_handle)
REFERENCES usert(usert_handle)
```

submission_like:

```
          Table "cs421g16.submission_like"
```

```
    Column     |          Type          | Modifiers
---------------+------------------------+-----------
 usert_handle  | character varying(255) | not null
 submission_id | integer                | not null
Indexes:
    "submission_like_pkey" PRIMARY KEY, btree (usert_handle, submission_id)
Foreign-key constraints:
    "submission_like_submission_id_fkey" FOREIGN KEY (submission_id)
REFERENCES submission(submission_id)
    "submission_like_usert_handle_fkey" FOREIGN KEY (usert_handle) REFERENCES
usert(usert_handle)
```

feed_view:

```
            Table "cs421g16.feed_view"
    Column    |          Type          | Modifiers
--------------+------------------------+-----------
 usert_handle | character varying(255) | not null
 post_id      | integer                | not null
Indexes:
    "feed_view_pkey" PRIMARY KEY, btree (usert_handle, post_id)
Foreign-key constraints:
    "feed_view_post_id_fkey" FOREIGN KEY (post_id) REFERENCES post(post_id)
    "feed_view_usert_handle_fkey" FOREIGN KEY (usert_handle) REFERENCES
usert(usert_handle)
```

usert_friend:

```
           Table "cs421g16.usert_friend"
    Column     |          Type          | Modifiers
---------------+------------------------+-----------
 usert_handle  | character varying(255) | not null
 friend_handle | character varying(255) | not null
Indexes:
    "usert_friend_pkey" PRIMARY KEY, btree (usert_handle, friend_handle)
Check constraints:
    "usert_friend_check" CHECK (usert_handle::text <> friend_handle::text)
Foreign-key constraints:
    "usert_friend_friend_handle_fkey" FOREIGN KEY (friend_handle) REFERENCES
usert(usert_handle)
    "usert_friend_usert_handle_fkey" FOREIGN KEY (usert_handle) REFERENCES
usert(usert_handle)
```

Question 3:

Note that before inserting tuples into our usert table, we had to first insert into our wall table and our creator table due to key constraints. These setups statements and a log of their execution can be found in **`part3.log`**.

Insertions:

Please see **`part3.log`** for a complete log of the executions of this question.

```
-- Insert statements
INSERT INTO usert VALUES
('John','johnny','john.doe@mail.mcgill.ca','John','Doe','M','12-1-1990');
INSERT 0 1
INSERT INTO usert VALUES
('spidey','mary','mary.jane@mail.mcgill.ca','Mary','Jane','F','4-20-1995');
INSERT 0 1
INSERT INTO usert VALUES
('Potus','realDonaldTrump','donald.trump@mail.mcgill.ca','Donald','Trump','M',
'3-3-1950');
INSERT 0 1
INSERT INTO usert VALUES
('OB','Obama','barack.obama@mail.mcgill.ca','Barack','Obama','M','8-6-1960');
INSERT 0 1
INSERT INTO usert VALUES
('lio','messi','lionel.messi@mail.mcgill.ca','Lionel','Messi','M','10-1-1987')
;
INSERT 0 1
```

Results:

```
-- Print usert table with inserted data
SELECT * FROM usert;
 usert_handle |  display_name   |            email            | firstname |
lastname | gender |     birth_date
--------------+-----------------+-----------------------------+-----------+---
-------+--------+---------------------
 John         | johnny          | john.doe@mail.mcgill.ca     | John      |
Doe      | M      | 1990-12-01 00:00:00
 spidey       | mary            | mary.jane@mail.mcgill.ca    | Mary      |
Jane     | F      | 1995-04-20 00:00:00
 Potus        | realDonaldTrump | donald.trump@mail.mcgill.ca | Donald    |
Trump    | M      | 1950-03-03 00:00:00
 OB           | Obama           | barack.obama@mail.mcgill.ca | Barack    |
Obama    | M      | 1960-08-06 00:00:00
 lio          | messi           | lionel.messi@mail.mcgill.ca | Lionel    |
Messi    | M      | 1987-10-01 00:00:00
(5 rows)
```

Question 4:

Please see **part4_insert.log** for the complete log of data insertion.

Results:

Please see **part4_print.log** for the complete log of results after data insertion.

wall:

```
SELECT * FROM wall LIMIT 10;
 wall_id |                description
---------+------------------------------------------
       1 | implement revolutionary functionalities
       2 | maximize B2C systems
       3 | disintermediate open-source niches
       4 | leverage ubiquitous infomediaries
       5 | visualize cross-platform markets
       6 | target e-business communities
       7 | morph frictionless eyeballs
       8 | visualize collaborative networks
       9 | grow compelling web-readiness
      10 | recontextualize bricks-and-clicks niches
(10 rows)
```

creator:

```
SELECT * FROM creator LIMIT 10;
    handle    | wall_id
--------------+---------
 dbeinke0     |       1
 owharlton1   |       2
 ggentry2     |       3
 awinyard3    |       4
 kswanwick4   |       5
 vweatherill5 |       6
 dbecerro6    |       7
 ckolak7      |       8
 noxer8       |       9
 hfishenden9  |      10
(10 rows)
```

usert:

```
SELECT * FROM usert LIMIT 10;
 usert_handle  |     display_name      |              email               |
firstname    |  lastname   | gender |     birth_date
---------------+-----------------------+-------------------------------+------
-----------+------------+--------+--------------------
 wambrozewicz0 | Warren Ambrozewicz     | wambrozewicz0@accuweather.com |
Warren          | Ambrozewicz | Male   | 1979-11-18 05:44:37
 kjanko1       | Kerrill Janko          | kjanko1@mediafire.com         |
Kerrill         | Janko       | Female | 1965-11-29 06:58:48
 hmckeighan2   | Hamish McKeighan       | hmckeighan2@wikispaces.com    |
Hamish          | McKeighan   | Male   | 1989-01-06 01:59:53
 dnarracott3   | Debra Narracott        | dnarracott3@nps.gov           | Debra
| Narracott    | Female | 1977-06-25 05:24:14
 bworboy4      | Baldwin Worboy         | bworboy4@columbia.edu         |
Baldwin         | Worboy      | Male   | 1999-11-05 11:28:37
 sgreenleaf5   | Staford Greenleaf      | sgreenleaf5@webeden.co.uk     |
Staford         | Greenleaf   | Male   | 1985-08-23 03:47:41
 dtolotti6     | Domingo Tolotti        | dtolotti6@webnode.com         |
Domingo         | Tolotti     | Male   | 1997-10-24 15:33:48
 omcmenamy7    | Obie McMenamy          | omcmenamy7@yellowbook.com     | Obie
| McMenamy     | Male   | 1960-12-16 22:17:05
 aleyden8      | Alfred Leyden          | aleyden8@wunderground.com     |
Alfred          | Leyden      | Male   | 1937-04-05 14:02:53
 hcheel9       | Helen-elizabeth Cheel | hcheel9@virginia.edu          |
Helen-elizabeth | Cheel       | Female | 1996-11-03 19:44:45
(10 rows)
```

page:

```
SELECT * FROM page LIMIT 10;
 page_handle  |            description             |                    title
| follower_count | usert_handle
--------------+------------------------------------+--------------------------
-------------------+----------------+---------------
 dbeinke0     | leverage open-source convergence  | Willard
|              0 | wambrozewicz0
 owharlton1   | generate end-to-end deliverables  | Diamonds Are Forever
|              0 | kjanko1
 ggentry2     | exploit transparent infomediaries | Tiger and the Snow, The
(La tigre e la neve) |              0 | hmckeighan2
 awinyard3    | expedite distributed schemas      | Skinwalkers
|              0 | dnarracott3
 kswanwick4   | maximize best-of-breed models     | Man Apart, A
|              0 | bworboy4
 vweatherill5 | utilize compelling content        | Pokemon 4 Ever (a.k.a.
Pokémon 4: The Movie) |              0 | sgreenleaf5
 dbecerro6    | extend efficient portals          | Don McKay
|              0 | dtolotti6
 ckolak7      | maximize next-generation content  | Jerichow
```

```
|              0 | omcmenamy7
 noxer8       | incubate world-class content      | Ann Carver’s Profession
|              0 | aleyden8
 hfishenden9  | leverage B2B systems              | Employee of the Month
|              0 | hcheel9
(10 rows)
```

submission:

```
SELECT * FROM submission LIMIT 10;
 submission_id |             text_content              | like_count |
creation_time
---------------+---------------------------------------+------------+---------
------------
             1 | maximize frictionless functionalities |          0 |
2017-10-30 19:56:44
             2 | generate viral supply-chains          |          0 |
2017-11-23 06:00:22
             3 | enhance dynamic e-business            |          0 |
2017-10-15 03:56:45
             4 | cultivate cross-media markets         |          0 |
2017-09-06 00:36:37
             5 | matrix mission-critical deliverables  |          0 |
2017-12-01 02:14:59
             6 | exploit 24/7 ROI                      |          0 |
2017-10-08 18:44:28
             7 | generate enterprise initiatives       |          0 |
2017-03-29 10:51:56
             8 | unleash open-source partnerships      |          0 |
2017-10-27 06:52:16
             9 | disintermediate B2C web-readiness     |          0 |
2017-03-28 19:08:41
            10 | benchmark turn-key ROI                |          0 |
2017-11-17 14:06:54
(10 rows)
```

post:

Note that because the attachment_link and image values are large, we only have printed the first 3 rows of the post table. Please see **`part4_print.log`** to view the first 10 rows of this table.

```
SELECT * FROM post LIMIT 10;
 post_id | view_count |
attachment_link
|
image
|    handle     | wall_id
---------+------------+-------------------------------------------------------
```

```
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
--------------------------------------------------------------------+---------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
------------------------------------------------------------------------------
---------------------------------------------------------------+--------------
-+---------
       1 |          0 |
https://technorati.com/amet/sapien/dignissim/vestibulum/vestibulum/ante/ipsum.
json?fringilla=nunc&rhoncus=purus&mauris=phasellus&enim=in&leo=felis&rhoncus=d
onec&sed=semper&vestibulum=sapien&sit=a&amet=libero&cursus=nam&id=dui&turpis=p
roin&integer=leo&aliquet=odio&massa=porttitor&id=id&lobortis=consequat&convall
is=in&tortor=consequat&risus=ut&dapibus=nulla&augue=sed&vel=accumsan&accumsan=
felis&tellus=ut&nisi=at&eu=dolor&orci=quis
|
data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf8/9hAAAABGdBTUEA
AK/INwWK6QAAABl0RVh0U29mdHdhcmUAQWRvYmUgSW1hZ2VSZWFkeXHJZTwAAAJ0SURBVDjLlZNbSB
RRGMe/2XRWZ2/D7G4sSuYaKVurhMZaIRbm6iaGERSSbz0EPfTSQ4j0ZNRLQq8WPaQgPfQaIWmUJWwr
SYolkW2ul7S0Rrbcy+zM7vQ/EovJ2uXAj/+c833nu5xzhtN1nXKN+h6Jc1Sqma/fhPHn574cpG2GYT
uDt9quHbe0U0vRiZrqXvHZfwXoHqnTfaZWg8ceII90jPy7mo5W9Vjv5fLltrZwa6RJLzbVkstWQys/
FiiakEng7TQ6N0iD7x4vhK+mSjb7522NmFb56PRqKF+OyYLbcQQlFtCT8H0aW5ygHUTiX1uYX75WL6
90C/PRScwyZDDoNCtPU1vlKB0ueGBpvXy76o8BTpUmu1x5CpGWT6Rn0CMrk6fIdIQcXJwaSjMXfjuD
qampXk3T6gGpqsqDMq/Xy918eoaUtEJriSjF1QQdiHcmTjYcKhweHo6nUqk5RVEIumrARs3pdHoQzI
```

```
OFKzzPc8Fg8GWj2EW1+qX209IN8lPn2d0OoXBoaOiuyWQSksnk9YqKCg+ClHGhUGgPx3EfYrFYMQLI
YB9YgnEnywTc+I5Ai6CfoS6wZjQal2RZfmTw+Xzh9fX176jgExxaYBhHBr8gCJPQgNlsfg1thr6FPW
Cz2Zg2ut1u1sKLjXcwiIHJeRhYBeXQZWgJ9COrAMyyOWDrTpyTDCLovm3jFpDhzqYKJlkGq9X6imUU
RZFVFJAkaQLzJugbBGlGgCUwln2J/f39KgwWOJWy04WWQ2fAXvAezm6wCFzIvB9c7Ovrq8u+RGzIAw
k4068ryoJNWdh149ApnU4/zPkv/Mvo6OjgmQ4MDKR+Apt6owU5Oz7IAAAAAElFTkSuQmCC
| wambrozewicz0 |      40
       2 |          0 |
http://rediff.com/pede.jsp?cras=magna&non=at&velit=nunc&nec=commodo&nisi=place
rat&vulputate=praesent&nonummy=blandit&maecenas=nam&tincidunt=nulla&lacus=inte
ger&at=pede&velit=justo&vivamus=lacinia&vel=eget&nulla=tincidunt&eget=eget&ero
s=tempus&elementum=vel&pellentesque=pede&quisque=morbi&porta=porttitor&volutpa
t=lorem&erat=id&quisque=ligula&erat=suspendisse&eros=ornare&viverra=consequat&
eget=lectus&congue=in&eget=est&semper=risus&rutrum=auctor&nulla=sed&nunc=trist
ique&purus=in&phasellus=tempus&in=sit&felis=amet&donec=sem&semper=fusce&sapien
=consequat&a=nulla&libero=nisl&nam=nunc&dui=nisl&proin=duis&leo=bibendum&odio=
felis&porttitor=sed&id=interdum&consequat=venenatis&in=turpis&consequat=enim&u
t=blandit&nulla=mi&sed=in&accumsan=porttitor&felis=pede&ut=justo&at=eu
|
data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf8/9hAAAABGdBTUEA
AK/INwWK6QAAABl0RVh0U29mdHdhcmUAQWRvYmUgSW1hZ2VSZWFkeXHJZTwAAAK/SURBVDjLY/j//z
8DJRiFozbrLk/aqkc76/a8eDft2Ou/Ew69+lm/8/n7pMUPTsuXXlAgaAAIK/fe9Kg7/ubmsaff/h99
/O2/48y7q+Tyz2vKZJ5hJGiAUucNRv0JNycuuvLho/WU24tytz67aNl5fZFM8mlhoryg0HAlcePNz7
+06670y2aftaja8fy224SbW6SzL1lrNt+aY95776BJ593dJq13dpu13jqoWXptGUJz1WXVkp0vrs48
/e6NTNoZM+n4kzpTDr5+7T/l9gHpzAvOyhU3J/vMe/w5e+OL/5lrXvzXKb2xTjz2QhncAKOWqzM3X/
/0Z97Jdx8mHHj1YsbB128P3Pz0P3bW3TNiXgfk9BturQ+Y9+ifU+/du4nLnvyXiD7fLBZ+lo0BGEAs
wACKXXLm3We/aXf2SoYejZQIPBws7ncwb+qeF29TZt+9LJlwNiNmydP/tm13LwNtdY+Y+/i/TNT5Xn
AYAANIL3vN40uTDrx6JRF0xBDmIlHPvepJM+5czJh174Hb5Pvv3SbceykWdd4aaGtQ5MyH/1UTLywD
G9Cx8/n3aQdf/W/e+uxL8ozb20CCIu57jIN7bpxcdujN/+hJ9/4nLnnyXyzibC1YLuS0d/jU+/+1ky
9swZoOkDHQuTHR8x//T1705H/MnIf/ffvu/Q+ffO9/ytyH/7XiLmwR9DoijFtz9Hkz6/qbl716736T
izo/XSTgZIGw34kc9ajz65JnPvivF3/+oIDbYQ2cBmhmX1qTMO/Rf7Hgk83C/ie4YOKCnkeCXSpvfN
CLPn+A3+WgEoZGYCAZi4aeKXZvu/PBo+3OV6CtwUI+x1nBmj2OKAJtbXCrvPbVNufSYz6nA/EYBrh3
3v3k23f3v2/Pnf8+HXf+G6VdPAa0lRMkZ5Zy8aJXzY1/QPzfq/rGf/fyaz8ZKM3OABiskbcwY1E6AA
AAAElFTkSuQmCC                                                              |
kjanko1        |      41
(3 rows)
```

commentt:

```
SELECT * FROM commentt LIMIT 10;
 commentt_id | usert_handle  | post_id
-------------+---------------+---------
          81 | wambrozewicz0 |       1
          82 | kjanko1       |       2
          83 | hmckeighan2   |       3
          84 | dnarracott3   |       4
          85 | bworboy4      |       5
          86 | sgreenleaf5   |       6
          87 | dtolotti6     |       7
          88 | omcmenamy7    |       8
          89 | aleyden8      |       9
          90 | hcheel9       |      10
(10 rows)
```

eventt:

```
SELECT * FROM eventt LIMIT 10;
 eventt_id |    handle     | wall_id |    creation_time     |     start_time
|      end_time       |                       title                       |
description                       |         location
-----------+--------------+---------+---------------------+-------------------
--+--------------------+--------------------------------------------------------+-
-----------------------------------------+--------------------------
         1 | eboughtons    |     252 | 2017-09-27 03:37:28 | 2018-06-04
18:00:30 | 2018-09-17 15:43:47 | Superhero Movie
| strategize mission-critical web services | 6523 Sloan Way
         2 | nlevingst     |     253 | 2017-11-09 20:26:16 | 2018-10-29
19:04:55 | 2018-09-15 05:02:15 | Soul Men
| morph best-of-breed content              | 916 Burning Wood Crossing
         3 | bmerrileesu   |     254 | 2017-03-26 14:31:34 | 2018-09-11
13:39:11 | 2018-07-24 06:02:05 | The Shadow of the Eagle
| embrace vertical e-business              | 720 Shoshone Junction
         4 | bmelbyv       |     255 | 2017-05-12 15:39:37 | 2018-10-29
02:05:10 | 2018-03-20 07:27:25 | Trouble at Timpetill (Enfants de Timpelbach,
Les) | integrate enterprise niches              | 415 Dahle Drive
         5 | vskynerw      |     256 | 2017-04-03 09:12:36 | 2018-09-19
04:05:37 | 2018-12-15 04:42:16 | Falls, The
| engineer one-to-one initiatives          | 9 Dapin Pass
         6 | bdillestonex |     257 | 2017-11-29 14:15:04 | 2018-04-07
11:14:57 | 2018-10-10 08:26:57 | 10.000 Km (Long Distance)
| enhance dynamic models                   | 0 Transport Street
         7 | tvasilechkoy |     258 | 2017-05-11 12:35:10 | 2018-04-05
00:19:27 | 2019-01-26 01:29:30 | Truman Show, The
| deploy frictionless infrastructures      | 39755 Loeprich Parkway
         8 | mbrittonz     |     259 | 2017-05-05 04:39:13 | 2018-03-27
18:10:40 | 2018-10-17 10:24:19 | Invasion of the Body Snatchers
| transform open-source architectures      | 3 Vernon Point
         9 | csiebert10    |     260 | 2017-08-03 15:46:17 | 2018-03-07
10:00:38 | 2018-07-11 14:45:38 | Blank Generation, The
| deploy revolutionary systems             | 1369 Crowley Lane
        10 | iskokoe11     |     261 | 2017-03-11 12:01:52 | 2018-05-16
11:59:18 | 2018-08-08 16:30:06 | Man with One Red Shoe, The
| envisioneer e-business partnerships      | 0658 Butterfield Avenue
(10 rows)
```

notification:

```
SELECT * FROM notification LIMIT 10;
 notif_id | read_status |             notif_link             |        notif_text
|     notif_time      | usert_handle
----------+-------------+----------------------------------+-----------------
---------+--------------------+--------------
        1 | f           | c91feeb261545b0039980572e61d4708 |
asherry0@hexun.com        | 2017-07-16 16:38:29 | wambrozewicz0
        2 | t           | c351eab625bea121dda19fc38a2c9b2d |
groseman1@nymag.com       | 2017-10-30 15:17:27 | kjanko1
        3 | f           | 3cbe36024316e05222efcaa9205f1d9b |
asynan2@squarespace.com   | 2017-10-24 16:29:57 | hmckeighan2
        4 | f           | e29b338021e6035cc0501c5028a3b4af | jadne3@issuu.com
| 2017-04-30 08:04:55 | dnarracott3
        5 | f           | e44fee0318cd21f0a68a62e67bfe1764 | ccesco4@last.fm
| 2017-09-03 22:17:00 | bworboy4
        6 | t           | 1c5a40aaba06de3b1f8f80a6c9b6689e |
lruddock5@dailymail.co.uk | 2017-11-29 08:26:47 | sgreenleaf5
        7 | f           | 86e40a5444a39ba74a5824396f139ce7 |
glandall6@earthlink.net   | 2017-12-01 22:31:59 | dtolotti6
        8 | t           | 119d166274d44b04963c0acceb648e22 |
abilson7@google.com.hk    | 2017-08-21 07:52:05 | omcmenamy7
        9 | t           | 83c9436bad3eaa571f6d09af78c03b53 |
wcrockley8@google.ca      | 2017-06-12 20:35:40 | aleyden8
       10 | f           | e7afe450070225e286628a087a589cb0 |
dhaines9@amazon.co.jp     | 2017-02-27 21:19:39 | hcheel9
(10 rows)
```

page_follower:

```
SELECT * FROM page_follower LIMIT 10;
 usert_handle  | page_handle
---------------+--------------
 wambrozewicz0 | dbeinke0
 kjanko1       | owharlton1
 hmckeighan2   | ggentry2
 dnarracott3   | awinyard3
 bworboy4      | kswanwick4
 sgreenleaf5   | vweatherill5
 dtolotti6     | dbecerro6
 omcmenamy7    | ckolak7
 aleyden8      | noxer8
 hcheel9       | hfishenden9
(10 rows)
```

eventt_subscription:

```
SELECT * FROM eventt_subscription LIMIT 10;
 usert_handle  | eventt_id | rsvp_status
---------------+-----------+-------------
 abaudichon13  |         1 | going
 abaudichon13  |         2 | not going
 abidewel18    |         3 | maybe
 abidewel18    |         4 | maybe
 abrierley37   |         5 |
 abrierley37   |         6 |
 abullente     |         7 |
 abullente     |         8 | not going
 abullerwell3x |         9 |
 abullerwell3x |        10 | maybe
(10 rows)
```

submission_like:

```
SELECT * FROM submission_like LIMIT 10;
 usert_handle  | submission_id
---------------+---------------
 wambrozewicz0 |             1
 kjanko1       |             2
 hmckeighan2   |             3
 dnarracott3   |             4
 bworboy4      |             5
 sgreenleaf5   |             6
 dtolotti6     |             7
 omcmenamy7    |             8
 aleyden8      |             9
 hcheel9       |            10
(10 rows)
```

feed_view:

```
SELECT * FROM feed_view LIMIT 10;
 usert_handle  | post_id
---------------+---------
 wambrozewicz0 |       1
 kjanko1       |       2
 hmckeighan2   |       3
 dnarracott3   |       4
 bworboy4      |       5
 sgreenleaf5   |       6
 dtolotti6     |       7
 omcmenamy7    |       8
 aleyden8       |       9
 hcheel9        |      10
(10 rows)
```

usert_friend:

```
SELECT * FROM usert_friend LIMIT 10;
  usert_handle   | friend_handle
-----------------+---------------
 abaudichon13    | wminet1f
 abidewel18      | wbradberry21
 abullente       | wambrozewicz0
 aleyden8        | vmccarlie1k
 amacness1q      | tmacourek10
 apack1u         | tkerman1n
 apellantk       | tdurninn
 bportchmouth1c  | tbeever1h
 bswoffer12      | srohan1i
 bthornberm      | spretiousv
(10 rows)
```

Question 5:

Please see **part5.log** for the complete log of query execution.

Query 1)

Description: Largest age difference between two friends:
Relations: See Question 2 for the create statements for tables `usert` & `usert_friend`.
SQL:

```
SELECT MAX(ABS(EXTRACT(YEAR FROM t1.birth_date) - EXTRACT(YEAR FROM
usert.birth_date))) AS age_difference FROM
      (SELECT usert.birth_date, usert_friend.friend_handle FROM usert_friend
      LEFT JOIN usert
      ON usert.usert_handle = usert_friend.usert_handle) t1
LEFT JOIN usert ON t1.friend_handle = usert.usert_handle;
```

Result:

```
 age_difference
----------------
             63
(1 row)
```

(Please see **part5.log** `for the log of this execution)`

Query 2)

Description: All the couples of friend with age difference higher than X (50 in this example).
Relations: See Question 2 for the create statements for tables `usert` & `usert_friend`.
SQL:

```
SELECT t1.usert_handle, usert.usert_handle, ABS(EXTRACT(YEAR FROM
t1.birth_date) - EXTRACT(YEAR FROM usert.birth_date)) AS age_difference FROM
      (SELECT usert.birth_date, usert_friend.usert_handle,
      usert_friend.friend_handle FROM usert_friend LEFT JOIN usert ON
      usert.usert_handle = usert_friend.usert_handle) t1
LEFT JOIN usert ON t1.friend_handle = usert.usert_handle
GROUP BY t1.usert_handle, usert.usert_handle, t1.birth_date, usert.birth_date
HAVING ABS(EXTRACT(YEAR FROM t1.birth_date) - EXTRACT(YEAR FROM
usert.birth_date)) > 50;
```

Result:

```
  usert_handle  |  usert_handle  | age_difference
----------------+----------------+----------------
 amacness1q     | tmacourek10    |             60
 bportchmouth1c | aspikeings2w   |             55
 bportchmouth1c | tbeever1h      |             52
 cfley36        | tloughrey3i    |             57
 chilleng       | cfley36        |             52
 cmoody4g       | tglazzard4b    |             61
 eramas2c       | rhudd2d        |             56
 hcheel9        | kwordsley1y    |             63
 hsibthorpea    | hde3d          |             62
 jhowgego52     | mtitmuss28     |             52
 jmacvaugh4k    | mgauthorpp3u   |             54
 kambrogelli1a  | hpersicke31    |             57
 kbrumfield1x   | iwanklin2o     |             59
 kwordsley1y    | hcheel9        |             63
 mgauthorpp3u   | jmacvaugh4k    |             54
 mlocke11       | jprydie3n      |             53
 mtauntonl      | jshelp58       |             52
 mtitmuss28     | jhowgego52     |             52
 osadgrovey     | kgehricke3h    |             61
 rhudd2d        | eramas2c       |             56
 tbeever1h      | bportchmouth1c |             52
 tbeever1h      | mgauthorpp3u   |             61
 tglazzard4b    | cmoody4g       |             61
 tloughrey3i    | cfley36        |             57
 tmacourek10    | amacness1q     |             60
(25 rows)
```

(Please see **`part5.log`** `for the log of this execution)`

Query 3)

Description: Display names of all users who are ‘going’ to more than one event
Relations: See Question 2 for the create statements for tables `usert` & `eventt_subscription`.
SQL:

```
SELECT display_name
FROM usert
WHERE usert_handle IN
      (SELECT usert_handle FROM eventt_subscription
      WHERE rsvp_status = 'going'
      GROUP BY usert_handle HAVING COUNT(*) > 1);
```

Result:

```
    display_name
---------------------
 Bayard Thornber
 Urson Cowperthwaite
 Gabriello Nulty
 Shanon Morse
 Libbi Buckleigh
 Lula Caillou
 Jackelyn Letessier
 Maribel Wheeler
 Osmund Royan
 Kathi Ambrogelli
 Johann Kelsell
 Cristiano Shelly
 Ring Gosz
 Jesselyn Darville
(14 rows)
```

(Please see **`part5.log`** `for the log of this execution)`

Query 4)

Description: Display names of all users who commented on post 39. Also display the text of their comment.
Relations: See Question 2 for the create statements for tables `usert`, `commentt`, `submission` & `post`.
SQL:

```
SELECT display_name, text_content FROM usert T1 JOIN (SELECT C.usert_handle,
S.text_content FROM commentt C, submission S WHERE C.commentt_id =
S.submission_id AND C.post_id = 39) T2 ON T1.usert_handle = T2.usert_handle;
```

Result:

```
   display_name    |           text_content
------------------+--------------------------------
```

```
 Bengt Swoffer     | harness next-generation users
 Xerxes Gennerich | iterate strategic relationships
 Sybyl Stebbins    | innovate seamless applications
 Debra Narracott   | synergize B2C communities
 Walliw Royds      | matrix global portals
(5 rows)
```

(Please see **`part5.log`** `for the log of this execution)`

Query 5)

Description: Total number of comments by gender
Relations: See Question 2 for the create statements for tables `usert` & `commentt`.
SQL:

```
SELECT gender, COUNT(*)
FROM commentt C
LEFT JOIN usert U
ON C.usert_handle = U.usert_handle
GROUP BY gender
```

Result:

```
 gender | count
--------+-------
 Female |   157
 Male   |   176
(2 rows)
```

(Please see **`part5.log`** `for the log of this execution)`

Question 6:

Please see **`part6.log`** for the complete log of the modification execution.
All relations that are used in these modifications have already been described (See Question 3).

Modification 1)

Description: Update the following_count number for each page
SQL:

```
UPDATE page SET follower_count = (SELECT COUNT(*) FROM page_follower WHERE
page_follower.page_handle = page.page_handle);
```

Execution: (note that due to our data generation, each page has 2 followers)

```
-- Update the following_count number for each page:
UPDATE page SET follower_count = (SELECT COUNT(*) FROM page_follower WHERE
page_follower.page_handle = page.page_handle);
UPDATE 50
-- Print the results
```

```
SELECT * FROM page LIMIT 10;
 page_handle  |           description            |                    title
| follower_count | usert_handle
--------------+----------------------------------+-------------------------
------------------+--------------+--------------
 dbeinke0     | leverage open-source convergence | Willard
|              2 | wambrozewicz0
 owharlton1   | generate end-to-end deliverables | Diamonds Are Forever
|              2 | kjanko1
 ggentry2     | exploit transparent infomediaries | Tiger and the Snow, The
(La tigre e la neve) |              2 | hmckeighan2
 awinyard3    | expedite distributed schemas     | Skinwalkers
|              2 | dnarracott3
 kswanwick4   | maximize best-of-breed models    | Man Apart, A
|              2 | bworboy4
 vweatherill5 | utilize compelling content       | Pokemon 4 Ever (a.k.a.
Pokémon 4: The Movie) |              2 | sgreenleaf5
 dbecerro6    | extend efficient portals         | Don McKay
|              2 | dtolotti6
 ckolak7      | maximize next-generation content | Jerichow
|              2 | omcmenamy7
 noxer8       | incubate world-class content     | Ann Carver's Profession
|              2 | aleyden8
 hfishenden9  | leverage B2B systems             | Employee of the Month
|              2 | hcheel9
(10 rows)
```

(Please see **`part6.log`** `for the log of this execution)`

Modification 2)

Description: Delete all events with creation times earlier than 11 months (must first delete all event_subscriptions referencing these events)

SQL:

```
DELETE FROM eventt_subscription WHERE eventt_id IN
(SELECT (eventt_id) FROM eventt WHERE creation_time < CURRENT_DATE -  INTERVAL
'11 months');
DELETE FROM eventt WHERE creation_time < CURRENT_DATE - INTERVAL '11 months';
```

Execution:

```
-- Delete all event subscriptions to events older than 11 months
DELETE FROM eventt_subscription WHERE eventt_id IN
(SELECT (eventt_id) FROM eventt WHERE creation_time < CURRENT_DATE -  INTERVAL
'11 months');
DELETE 27
-- Delete all events with creation times earlier than 11 months
DELETE FROM eventt WHERE creation_time < CURRENT_DATE - INTERVAL '11 months';
DELETE 4
```

(Please see **`part6.log`** `for the log of this execution`)

Modification 3)

Description: Delete all comments with no likes then delete the corresponding submissions for these comments in order to maintain covering constraint (submission must be either post or comment).

SQL:

```
DELETE FROM commentt C WHERE NOT EXISTS (SELECT * FROM submission_like SL
WHERE C.commentt_id = SL.submission_id);
DELETE FROM submission WHERE submission_id NOT IN (SELECT post_id AS
submission_id FROM post UNION SELECT commentt_id AS submission_id FROM
commentt);
```

Execution:

```
-- Delete all comments with no likes:
DELETE FROM commentt C WHERE NOT EXISTS (SELECT * FROM submission_like SL
WHERE C.commentt_id = SL.submission_id);
DELETE 133
-- Delete the corresponding submission for each deleted comment:
DELETE FROM submission WHERE submission_id NOT IN (SELECT post_id AS
submission_id FROM post UNION SELECT commentt_id AS submission_id FROM
commentt);
DELETE 133
```

(Please see **`part6.log`** `for the log of this execution`)

Modification 4)

Description: Update the like_count number for each submission

SQL:

```
UPDATE submission SET like_count = (SELECT COUNT(*) FROM submission_like WHERE
submission.submission_id = submission_like.submission_id);
```

Execution: (note that due to our data generation, each submission has 2 likes)

```
-- Update the like_count number for each submission:
UPDATE submission SET like_count = (SELECT COUNT(*) FROM submission_like WHERE
submission.submission_id = submission_like.submission_id);
UPDATE 280
-- Print the results
SELECT * FROM submission LIMIT 10;
 submission_id |             text_content              | like_count |
creation_time
---------------+---------------------------------------+------------+----------
----------
           209 | e-enable sticky infomediaries         |          2 | 2017-12-02
01:49:17
           210 | brand innovative metrics              |          2 | 2017-07-29
```

```
05:00:59
           211 | facilitate e-business experiences   |          2 | 2017-12-18
11:16:40
           212 | embrace cutting-edge users          |          2 | 2017-03-30
07:55:15
           213 | redefine plug-and-play deliverables |          2 | 2017-06-25
04:56:11
           214 | redefine sticky e-markets           |          2 | 2017-07-14
13:49:27
           215 | grow wireless users                 |          2 | 2017-09-09
21:39:33
           216 | recontextualize front-end eyeballs  |          2 | 2017-09-17
04:34:32
           217 | cultivate killer schemas            |          2 | 2018-01-22
00:22:39
           218 | visualize distributed infomediaries |          2 | 2017-05-24
09:25:53
(10 rows)
```

Question 7:

Please see **`part7.log`** for the complete log of the following view execution.

View 1:

Informal description:
This view represents the average age per event of the people that rsvped that they are 'going' to said event.

Create the view:

```
CREATE VIEW eventt_average_ages (eventt_id, average_age) AS SELECT
t1.eventt_id, AVG(age(usert.birth_date)) AS avgage FROM (SELECT * FROM
eventt_subscription WHERE rsvp_status = 'going') t1 LEFT JOIN usert ON
usert.usert_handle = t1.usert_handle GROUP BY eventt_id ORDER BY avgage;
```

Use the view (to find the event with the max average age of people that are going)

```
SELECT * FROM eventt_average_ages WHERE average_age = (SELECT MAX(average_age)
FROM eventt_average_ages);
```

Response:

```
 eventt_id |          average_age
-----------+-------------------------------
        32 | 85 years 1 mon 1 day 22:28:04
(1 row)
```

Trying to update:

```
UPDATE eventt_average_ages SET average_age = age(CURRENT_TIMESTAMP) WHERE
eventt_id = 32;
```

Response:

```
ERROR:  cannot update view "eventt_average_ages"
DETAIL:  Views containing GROUP BY are not automatically updatable.
HINT:  To enable updating the view, provide an INSTEAD OF UPDATE trigger or an
unconditional ON UPDATE DO INSTEAD rule.
```

This view is NOT updatable. The reason why is that we are using an aggregate function (AVG, or average) which means that we can't update any specific elements as all the elements of the view are groups in the database.

View 2:

Informal description: Create a view that consists of all the users' with unread notifications usert_handle and the number of each users unread notifications.

Create the view:

```
CREATE VIEW unread_notifications (usert_handle, num_of_notifs) AS SELECT
usert_handle, COUNT(usert_handle)
FROM notification
WHERE read_status = 'f'
GROUP BY usert_handle;
```

Use the view (to find the users with highest number of unread notifications)

```
SELECT usert_handle FROM unread_notifications WHERE num_of_notifs = (SELECT
MAX(num_of_notifs) FROM unread_notifications);
```

Response:

```
usert_handle
--------------
 tbettenay33
(1 row)
```

Trying to update:

```
UPDATE unread_notifications SET num_of_notifs = 4 WHERE usert_handle =
'tbettenay33';
```

Response:

```
ERROR:  cannot update view "unread_notifications"
DETAIL:  Views containing GROUP BY are not automatically updatable.
HINT:  To enable updating the view, provide an INSTEAD OF UPDATE trigger or an
unconditional ON UPDATE DO INSTEAD rule.
```

This view is NOT updatable because we are using a GROUP BY on usert_handle which prevents us from updating a specific row.

Summary of the conditions for a view to be updatable:

- A view cannot contain an aggregate function, a window function or a set-returning function
- A view cannot contain a set operation at the top level
- A view cannot contain any of the following clauses at the top level: WITH, DISTINCT, GROUP BY, HAVING, LIMIT, OFFSET
- The view must only have one table or updatable view in its FROM list

Disclaimer: This information was taken from the official PostgreSQL documentation.

Question 8:

Please see **part8.log** for the complete log of the execution.

Check1:

Description: Check that the user is above 18

Add Constraint:

```
ALTER TABLE usert ADD CONSTRAINT minimum_age CHECK (EXTRACT(YEAR FROM
age(birth_date)) >= 18);
```

Response:

```
ALTER TABLE
```

Revised Schema:

```
                Table "cs421g16.usert"
    Column    |            Type             | Modifiers
--------------+-----------------------------+-----------
 usert_handle | character varying(255)      | not null
 display_name | character varying(255)      |
 email        | character varying(255)      |
 firstname    | character varying(255)      |
 lastname     | character varying(255)      |
 gender       | character varying(255)      |
 birth_date   | timestamp without time zone |
Indexes:
    "usert_pkey" PRIMARY KEY, btree (usert_handle)
    "usert_email_key" UNIQUE CONSTRAINT, btree (email)
Check constraints:
    "minimum_age" CHECK (date_part('year'::text, age(birth_date)) >=
18::double precision)
Foreign-key constraints:
    "usert_usert_handle_fkey" FOREIGN KEY (usert_handle) REFERENCES
creator(handle)
Referenced by:
    TABLE "commentt" CONSTRAINT "commentt_usert_handle_fkey" FOREIGN KEY
(usert_handle) REFERENCES usert(usert_handle)
    TABLE "eventt_subscription" CONSTRAINT
"eventt_subscription_usert_handle_fkey" FOREIGN KEY (usert_handle) REFERENCES
usert(usert_handle)
    TABLE "feed_view" CONSTRAINT "feed_view_usert_handle_fkey" FOREIGN KEY
(usert_handle) REFERENCES usert(usert_handle)
    TABLE "notification" CONSTRAINT "notification_usert_handle_fkey" FOREIGN
KEY (usert_handle) REFERENCES usert(usert_handle)
    TABLE "page_follower" CONSTRAINT "page_follower_usert_handle_fkey" FOREIGN
KEY (usert_handle) REFERENCES usert(usert_handle)
    TABLE "page" CONSTRAINT "page_usert_handle_fkey" FOREIGN KEY
(usert_handle) REFERENCES usert(usert_handle)
    TABLE "submission_like" CONSTRAINT "submission_like_usert_handle_fkey"
FOREIGN KEY (usert_handle) REFERENCES usert(usert_handle)
    TABLE "usert_friend" CONSTRAINT "usert_friend_friend_handle_fkey" FOREIGN
KEY (friend_handle) REFERENCES usert(usert_handle)
    TABLE "usert_friend" CONSTRAINT "usert_friend_usert_handle_fkey" FOREIGN
KEY (usert_handle) REFERENCES usert(usert_handle)
```

Trying to violate constraint (update an age to under 18)**.**

```
UPDATE usert SET birth_date = '2002-03-01' WHERE usert_handle = 'bworboy4';
```

Response:

```
ERROR:  new row for relation "usert" violates check constraint "minimum_age"
DETAIL:  Failing row contains (bworboy4, Baldwin Worboy,
bworboy4@columbia.edu, Baldwin, Worboy, Male, 2002-03-01 00:00:00).
```

Check2:

Description: Only allow event that last less than a year

Add Constraint:

```
ALTER TABLE eventt ADD CONSTRAINT eventt_time_constraint CHECK ( end_time <
(start_time + INTERVAL '1 year'));
```

Response:

```
ERROR:  check constraint "eventt_time_constraint" is violated by some row
```

Revised Schema:

```
                                       Table "cs421g16.eventt"
    Column     |            Type             |
Modifiers
---------------+-----------------------------+------------------------------
----------------------------
 eventt_id     | integer                     | not null default
nextval('eventt_eventt_id_seq'::regclass)
 handle        | character varying(255)      | not null
 wall_id       | integer                     | not null
 creation_time | timestamp without time zone | default now()
 start_time    | timestamp without time zone | not null
 end_time      | timestamp without time zone | not null
 title         | character varying(255)      | not null
 description   | character varying(1023)     |
 location      | character varying(255)      |
Indexes:
    "eventt_pkey" PRIMARY KEY, btree (eventt_id)
    "eventt_wall_id_key" UNIQUE CONSTRAINT, btree (wall_id)
Check constraints:
    "eventt_time_constraint" CHECK (end_time < (start_time + '1
year'::interval))
Foreign-key constraints:
    "eventt_handle_fkey" FOREIGN KEY (handle) REFERENCES creator(handle)
    "eventt_wall_id_fkey" FOREIGN KEY (wall_id) REFERENCES wall(wall_id)
Referenced by:
    TABLE "eventt_subscription" CONSTRAINT
"eventt_subscription_eventt_id_fkey" FOREIGN KEY (eventt_id) REFERENCES
eventt(eventt_id)
```

Trying to violate constraint (update an event to be last longer than a year):

```
UPDATE eventt SET end_time = '2020-06-05' WHERE eventt_id = 1;
```

Response:

```
ERROR:  new row for relation "eventt" violates check constraint
"eventt_time_constraint"
DETAIL:  Failing row contains (1, eboughtons, 252, 2017-09-27 03:37:28,
2018-06-04 18:00:30, 2020-06-05 00:00:00, Superhero Movie, strategize
mission-critical web services, 6523 Sloan Way).
```

Question 9:

Automation and Data:

Randomized realistic data was generated from mockaroo.com.

Bash scripts were created that take our sql files and run all parts of our assignment in a programmatic fashion so that we did not have to do it manually.
These bash scripts, sql files, and the logs that were generated (and referenced in this report) are included in a zip file along with the submission of this report and can also be found in the following github repo: comp421-project

Complex Query:

Please see **`part9.log`** for the complete log of the query execution.
Description: Find the friends with the highest age difference in terms of years
SQL:

```
SELECT t1.usert_handle, usert.usert_handle, ABS(EXTRACT(YEAR FROM
t1.birth_date) - EXTRACT(YEAR FROM usert.birth_date)) AS age_difference FROM
      (SELECT usert.birth_date, usert_friend.usert_handle,
      usert_friend.friend_handle FROM usert_friend LEFT JOIN usert ON
usert.usert_handle = usert_friend.usert_handle) t1
LEFT JOIN usert ON t1.friend_handle = usert.usert_handle GROUP BY
t1.usert_handle, usert.usert_handle, t1.birth_date, usert.birth_date HAVING
ABS(EXTRACT(YEAR FROM t1.birth_date) - EXTRACT(YEAR FROM usert.birth_date)) =
(SELECT MAX(ABS(EXTRACT(YEAR FROM t1.birth_date) - EXTRACT(YEAR FROM
usert.birth_date))) AS age_difference FROM
      (SELECT usert.birth_date, usert_friend.usert_handle,
      usert_friend.friend_handle FROM usert_friend LEFT JOIN usert ON
      usert.usert_handle = usert_friend.usert_handle) t1
LEFT JOIN usert ON t1.friend_handle = usert.usert_handle);
```

Response:

```
 usert_handle | usert_handle | age_difference
--------------+--------------+----------------
 kwordsley1y  | hcheel9      |             63
 hcheel9      | kwordsley1y  |             63
(2 rows)
```