A6: Indexes, triggers, user functions and population

1. Database workload

A study of the predicted system load (database load), organized in subsections

1.1. Tuple Estimation

Relation Reference	Relation Name	Order of magnitude	Estimated growth
R01	user	tens	Units per day
R02	event	tens	Units per day
R03	location	tens	Units per day
R04	image	tens	Units per day
R05	city	tens	Units per day
R06	country	tens	Units per day
R07	post	hundreds	Dozens per day
R08	poll	tens	Units per day
R09	option	tens	Units per day
R10	friendRequest	hundreds	Dozens per day
R11	friendActivity	hundreds	Dozens per day
R12	eventInvite	hundreds	Dozens per day
R13	owner	tens	Units per day
R14	participant	tens	Units per day
R15	done	tens	Units per day
R16	notDone	tens	Units per day
R17	admin	units	No growth
R18	friendship	tens	Units per day
R19	eventWarning	tens	Tens per day
R20	currentDate	units	Hundreds per day

1.2. Most frequent queries

Querie reference	SELECT01
Querie description	User’s information
Querie frequency	Hundreds per day

```
SELECT username, last_name, first_name, email, image_path, city_id
  FROM users
  WHERE users.id = $user_id;
```

<table>
<tr><td>Querie reference</td><td>SELECT02</td></tr>
<tr><td>Querie description</td><td>Event’s information</td></tr>
<tr><td>Querie frequency</td><td>Hundreds per day</td></tr>
<tr><td colspan="2">SELECT events.id, events.name, events.category, events.image_id, events.description, events."date" users.username
FROM events, users
WHERE events.owner_id = users.id AND events.id = $event_id;

SELECT posts.description, posts.id, posts.image_id, posts.user_id
FROM posts,events
WHERE posts.event_id = $event_id;

SELECT users.username, users.image_path
FROM participants
WHERE users.id = participants.user_id AND participants.event_id=$event_id;</td></tr>
</table>

<table>
<tr><td>Querie reference</td><td>SELECT03</td></tr>
<tr><td>Querie description</td><td>Search event</td></tr>
<tr><td>Querie frequency</td><td>Hundreds per day</td></tr>
<tr><td colspan="2">SELECT id, "name", image_id, "date", localization, category
FROM events
WHERE "name" LIKE %$search% OR localization LIKE %$search% AND event_type = 'public'
ORDER BY "name";</td></tr>
</table>

<table>
<tr><td>Querie reference</td><td>SELECT04</td></tr>
<tr><td>Querie description</td><td>Search user</td></tr>
<tr><td>Querie frequency</td><td>Hundreds per day</td></tr>
<tr><td colspan="2">SELECT username, email, image_path
FROM users
WHERE username LIKE %$search% OR email LIKE %$search%
ORDER BY username;</td></tr>
</table>

Querie reference	SELECT05
Querie description	Search category
Querie frequency	Hundreds per day

```
SELECT id, "name", image_id, "date", localization, category
FROM events
  WHERE "name" LIKE %$search% OR localization LIKE %$search% AND event_type = 'public'
AND events.category LIKE %$categories%
  ORDER BY "name";
```

Querie reference	SELECT06
Querie description	User’s notifications
Querie frequency	Hundreds per day

```
SELECT sender_id
  FROM friend_requests
  WHERE receiver_id = $user_id;

SELECT sender_id, event_id
  FROM friend_activities
  WHERE receiver_id = $user_id;

SELECT owner_id, event_id
  FROM event_invites
  WHERE receiver_id = $user_id;

SELECT event_id, "message"
  FROM event_warnings
  WHERE receiver_id = $user_id;
```

Querie reference	SELECT07
Querie description	Search category
Querie frequency	Hundreds per day

```
SELECT id, "name", image_id, "date", localization, category
FROM events
  WHERE "name" LIKE %$search% OR localization LIKE %$search% AND event_type = 'public'
AND events.category LIKE %$categories%
  ORDER BY "name";
```

Querie reference	SELECT08
Querie description	User’s events
Querie frequency	Hundreds per day

```
SELECT event_id
  FROM participants
  WHERE user_id = $user_id;

SELECT event_id
  FROM owners
  WHERE user_id = $user_id;
```

Querie reference	SELECT09
Querie description	User’s friends
Querie frequency	Hundreds per day

```
SELECT user_id_1, user_id_2
  FROM friendships
  WHERE user_id_1 = $user_id OR user_id_2 = $user_id;
```

Querie reference	SELECT010
Querie description	Who can be invited to the event
Querie frequency	Hundreds per day

```
SELECT users.username
  FROM users, events
  WHERE users.id!= event.owner_id AND users.id  NOT IN (
    SELECT user_id
    FROM participants
    WHERE user_id IS NOT NULL AND participants.event_id=$event_id) ;
```

Querie reference	SELECT11
Querie description	Rating of a past event
Querie frequency	Hundreds per day

```
SELECT AVG(rating)
  FROM dones
  WHERE dones.event_id= $event_id;
```

1.3. Most frequent modifications

Querie reference	UPDATE01
Querie description	Update users information
Querie frequency	Hundreds per month

```
UPDATE "users"
SET password = $password,
   email = #email,
   first_name = $first_name,
   last_name = $last_name,
   image_path = $image_path,
   city_id = $city_id
WHERE id = $id;
```

Querie reference	UPDATE02
Querie description	Update events information
Querie frequency	Hundreds per month

```
UPDATE events
SET name = $name,
   date = $date,
   description = $description,
   localization_id = $localization_id,
   image_id = $image_id,
   event_type = $event_type,
   category = $category
WHERE id = $id;
```

Querie reference	UPDATE03
Querie description	Update dones rating
Querie frequency	Tens per month

```
UPDATE dones
SET rating = $rating
WHERE event_id = $event_id;
```

Querie reference	UPDATE04
Querie description	Update options description
Querie frequency	Tens per month

UPDATE options
SET description = $description
WHERE id = $id;	

Querie reference	UPDATE05
Querie description	Update posts information
Querie frequency	Hundreds per month

UPDATE posts
SET description = $description.
date = $date,
image_id = $image_id
WHERE id = $id;	

Querie reference	UPDATE06
Querie description	Update current_date
Querie frequency	Thousands per month

UPDATE current_date
SET date = $date
WHERE id = $id;	

Querie reference	INSERT01
Querie description	Regist new user
Querie frequency	Tens per month

INSERT INTO users
(id,username,password,email,regist_date,first_name,last_name, image_path,city_id)
VALUES ($id, $username, $password, $email, $regist_date, $first_name, $last_name, $image_path, $city_id);	

Querie reference	INSERT02
Querie description	Create new event
Querie frequency	Tens per month
INSERT INTO events (id,name,date,description,owner_id,localization_id,image_id,type,category) VALUES ($id,$name,$date,$description,$owner_id,$localization_id,$image_id, $type,$category);	

Querie reference	INSERT03
Querie description	Create new post
Querie frequency	Tens per month
INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES ($id,$description,$date,$event_id, $user_id, $image_id);	

Querie reference	INSERT04
Querie description	Create new event done
Querie frequency	Tens per month

INSERT INTO dones (event_id,rating)
VALUES ($event, $rating);	

Querie reference	INSERT05
Querie description	Create new event not_done
Querie frequency	Tens per month

INSERT INTO dones (event_id)
VALUES ($event);	

Querie reference	INSERT06
Querie description	Create new participant
Querie frequency	Tens per month

INSERT INTO participants (id,user_id,event_id)
VALUES ($id,$user_id,$event_id);	

Querie reference	INSERT07
Querie description	Create new owner
Querie frequency	Units per month

INSERT INTO owners (id,user_id,event_id)
VALUES ($id,$user_id,$event_id);	

Querie reference	INSERT08
Querie description	Create new image
Querie frequency	Tens per month
INSERT INTO images (id,path) VALUES ($id,$path);	

Querie reference	INSERT09
Querie description	Create new localization
Querie frequency	Units per month
INSERT INTO localizations (id,name,address,latitude,longitude,city_id) VALUES ($id,$name,$address,$latitude,$longitude,$city_id);	

Querie reference	INSERT10
Querie description	Create new city
Querie frequency	Units per month
INSERT INTO cities (id,name,country_id) VALUES ($id,$name,$country_id);	

Querie reference	INSERT11
Querie description	Create new country

Querie frequency	Units per month
INSERT INTO countries (id,name) VALUES ($id,$name);	

Querie reference	INSERT12
Querie description	Create new poll
Querie frequency	Units per month
INSERT INTO polls (id,post_id) VALUES ($id,$posts_id);	

Querie reference	INSERT13
Querie description	Create new option
Querie frequency	Tens per month
INSERT INTO options (id,description,poll_id) VALUES ($id,$description,$poll_id);	

Querie reference	INSERT14
Querie description	Create new friend request
Querie frequency	Tens per month
INSERT INTO friend_requests (id,answer,sender_id,receiver_id) VALUES ($id,$answer,$sender_id,$receiver_id);	

Querie reference	INSERT15
Querie description	Create new friend activity
Querie frequency	Tens per month
INSERT INTO friend_activities (id,sender_id,receiver_id,event_id) VALUES ($id,$sender_id,$receiver_id,$event_id);	

Querie reference	INSERT16
Querie description	Create new event invite
Querie frequency	Tens per month

INSERT INTO event_invites (id,answer,event_id,owner_id,receiver_id) VALUES ($id,$answer,$event_id,$owner_id,$receiver_id);

Querie reference	INSERT17
Querie description	Create new friendship
Querie frequency	Tens per month
INSERT INTO friendships (id, user_id_1, user_id_2) VALUES ($id, $user_id_1, $user_id_2);	

Querie reference	INSERT18
Querie description	Create new event warning
Querie frequency	Tens per month
INSERT INTO event_warnings (id, event_id, receiver_id, message) VALUES ($id, $event_id, $receiver_id, $message);	

Querie reference	DELETE01
Querie description	Delete an user
Querie frequency	Units per month

DELETE FROM "users"
WHERE id = $id;	

Querie reference	DELETE02
Querie description	Delete an event
Querie frequency	Units per month

DELETE FROM events
WHERE id = $id;	

Querie reference	DELETE03
Querie description	Delete a post
Querie frequency	Units per month

DELETE FROM posts
WHERE id = $id;	

Querie reference	DELETE04
Querie description	Delete an option
Querie frequency	Tens per month

DELETE FROM options
WHERE id = $id;	

Querie reference	DELETE05
Querie description	Delete a poll
Querie frequency	Units per month

DELETE FROM polls
WHERE id = $id;	

Querie reference	DELETE06
Querie description	Delete a friendship
Querie frequency	Units per month

DELETE FROM friendships
WHERE id = $id;	

Querie reference	DELETE07
Querie description	Delete a friend_request
Querie frequency	Tens per month

DELETE FROM friend_requests
WHERE id = $id;	

Querie reference	DELETE08
Querie description	Delete an event_invite
Querie frequency	Tens per month

DELETE FROM event_invites
WHERE id = $id;	

2. Proposed Indexes

2.1. Performance Indexes

Index reference	IDX01
Related queries	SELECT01
Índex relation	user
Índex atribute	username
Index type	Hash
Cardinality	High
Clustering	No
Justification	Query SELECT01 has to be fast as it is executed many times; doesn't need range query support; cardinality is high because email is an unique key; it's not a good candidate for clustering.

CREATE INDEX user_username ON users USING hash (username);

Index reference	IDX02
Related queries	SELECT02
Índex relation	events
Índex atribute	Owner_id
Index type	Hash
Cardinality	high
Clustering	No
Justification	Query has to be fast as it is executed many times; doesn't need range query support; cardinality is high because owner_id is an unique key; it's not a good candidate for clustering.
CREATE INDEX owner_events ON events USING hash(owner_id);	

2.2. Full-text search indexes

Index reference	IDX03
Related queries	SELECT03
Índex relation	event
Índex atribute	name
Index type	GIST
Clustering	No
Justification	To improve the performance of full text searches while searching for events and their names; GiST because it's better for dynamic data.
CREATE INDEX search_events ON events USING GIST (to_tsvector('english', name));	

3. Triggers

Trigger reference	TRIGGER01

Trigger description	When the current_date is updated, it is verified if it is greater than the event_date. In that case, the event is added to the dones table and removed from the not_dones.

```
CREATE FUNCTION set_event_as_done() RETURNS TRIGGER AS
$BODY$
BEGIN
  IF EXISTS (SELECT event_id FROM not_done WHERE NEW.event_id = id)
  THEN
    INSERT INTO dones (NEW.event_id, NULL);
    DELETE FROM not_dones WHERE id = NEW.event_id;
  END IF;
  RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER set_event_as_done
  BEFORE UPDATE OF date ON current_date
  FOR EACH ROW
  WHEN NEW.date = now()
    EXECUTE PROCEDURE set_event_as_done();
```

Trigger reference	TRIGGER02
Trigger description	When an event is deleted a notification is sent to the user

```
CREATE FUNCTION notificate_event_delete() RETURNS TRIGGER AS
$BODY$
WHILE( SELECT id FROM participants WHERE participants.event_id = OLD.event_id)
BEGIN
  INSERT INTO event_warnings(OLD.event_id, id)
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER notificate_event_delete
  FOR DELETE OR UPDATE ON events
  FOR EACH ROW
    EXECUTE PROCEDURE notificate_event_delete();
```

4. SQL code

```
--Types Enums

DROP TYPE IF EXISTS categories CASCADE;
CREATE TYPE categories AS ENUM(
   'Music',
   'Sports',
   'Entertainment',
   'Educational',
   'Business',
   'Other'
);
DROP TYPE IF EXISTS types_of_event CASCADE;
CREATE TYPE types_of_event AS ENUM(
   'Public',
   'Private'
);

--Tables
DROP TABLE IF EXISTS admins CASCADE;
CREATE TABLE admins (
   id SERIAL NOT NULL,
   username text NOT NULL,
   password text NOT NULL,
   email text NOT NULL,
   CONSTRAINT admins_pk PRIMARY KEY (id),
   CONSTRAINT admins_email_uk UNIQUE (email)
);
```

```
DROP TABLE IF EXISTS cities CASCADE;
CREATE TABLE cities (
   id SERIAL NOT NULL,
   name text NOT NULL,
   country_id INTEGER NOT NULL,
   CONSTRAINT cities_pk PRIMARY KEY (id),
   CONSTRAINT cities_name_uk UNIQUE (name)
);

DROP TABLE IF EXISTS current_date CASCADE;
CREATE TABLE current_date (
   id SERIAL NOT NULL,
   date TIMESTAMP WITH TIME zone NOT NULL,
   CONSTRAINT current_date_pk PRIMARY KEY (id),
);

DROP TABLE IF EXISTS countries CASCADE;
CREATE TABLE countries (
   id SERIAL NOT NULL,
   name text NOT NULL,
   CONSTRAINT countries_pk PRIMARY KEY (id),
   CONSTRAINT countries_name_uk UNIQUE (name)
);

DROP TABLE IF EXISTS dones CASCADE;
CREATE TABLE dones (
   event_id INTEGER NOT NULL,
   rating INTEGER,
   CONSTRAINT dones_pk PRIMARY KEY (event_id),
   CONSTRAINT rating_ck CHECK (((rating >= 1) AND (rating <= 5)))
);
```

```
DROP TABLE IF EXISTS events CASCADE;

CREATE TABLE events (

   id SERIAL NOT NULL,

   name text NOT NULL,

   date TIMESTAMP WITH TIME zone NOT NULL,

   description text NOT NULL,

   owner_id INTEGER NOT NULL,

   localization_id INTEGER NOT NULL,

   image_id INTEGER NOT NULL,

   type event_type NOT NULL,

   category categories NOT NULL,

   CONSTRAINT events_pk PRIMARY KEY (id),

   CONSTRAINT date_ck CHECK ((date > now()))

);


DROP TABLE IF EXISTS event_invites CASCADE;

CREATE TABLE event_invites (

   id SERIAL NOT NULL,

   answer text NOT NULL,

   event_id INTEGER NOT NULL,

   owner_id INTEGER NOT NULL,

   receiver_id INTEGER NOT NULL,

   CONSTRAINT event_invites_pk PRIMARY KEY (id)

);
```

```
DROP TABLE IF EXISTS event_warnings CASCADE;
CREATE TABLE event_warnings (
   id SERIAL NOT NULL,
   event_id INTEGER NOT NULL,
   receiver_id INTEGER NOT NULL,
   message text NOT NULL,
   CONSTRAINT event_warnings_pk PRIMARY KEY (id),
   CONSTRAINT event_warnings_event_id_fk FOREIGN KEY (event_id) REFERENCES events(id),
   CONSTRAINT event_warnings_user_id_fk FOREIGN KEY (user_id) REFERENCES users(id)
);

DROP TABLE IF EXISTS friend_activities CASCADE;
CREATE TABLE friend_activities (
   id SERIAL NOT NULL,
   sender_id INTEGER NOT NULL,
   receiver_id INTEGER NOT NULL,
   event_id INTEGER NOT NULL,
   CONSTRAINT friend_activities_pk PRIMARY KEY (id),
   CONSTRAINT friend_activities_event_id_fk FOREIGN KEY (event_id) REFERENCES
   events(id) ON DELETE CASCADE
);

DROP TABLE IF EXISTS friend_requests CASCADE;
CREATE TABLE friend_requests (
   id SERIAL NOT NULL,
   answer text NOT NULL,
   sender_id INTEGER NOT NULL,
   receiver_id INTEGER NOT NULL,
   CONSTRAINT friend_requests_pk PRIMARY KEY (id)
);
```

```
DROP TABLE IF EXISTS friendships CASCADE;
CREATE TABLE friendships (
   id SERIAL NOT NULL,
   user_id_1 INTEGER NOT NULL,
   user_id_2 INTEGER NOT NULL,
   CONSTRAINT friendships_users_ids_uk UNIQUE (user_id_1, user_id_2)
)

DROP TABLE IF EXISTS images CASCADE;
CREATE TABLE images (
   id SERIAL NOT NULL,
   path text NOT NULL,
   CONSTRAINT images_pk PRIMARY KEY (id),
   CONSTRAINT images_path_uk UNIQUE (path)
);

DROP TABLE IF EXISTS localizations CASCADE;
CREATE TABLE localizations (
   id SERIAL NOT NULL,
   name text NOT NULL,
   address text NOT NULL,
   latitude FLOAT,
   longitude FLOAT,
   city_id INTEGER NOT NULL,
   CONSTRAINT localizations_pk PRIMARY KEY (id),
   CONSTRAINT localizations_city_id_fk FOREIGN KEY (city_id) REFERENCES
   cities(id) ON DELETE SET NULL
);
```

```
DROP TABLE IF EXISTS not_dones CASCADE;
CREATE TABLE not_dones (
   event_id INTEGER NOT NULL,
   CONSTRAINT not_dones_pk PRIMARY KEY (event_id),
   CONSTRAINT not_dones_event_id_fk FOREIGN KEY (event_id) REFERENCES
   events(id) ON UPDATE CASCADE
);

DROP TABLE IF EXISTS options CASCADE;
CREATE TABLE options (
   id SERIAL NOT NULL,
   description text NOT NULL,
   poll_id INTEGER NOT NULL,
   CONSTRAINT options_pk PRIMARY KEY (id)
);

DROP TABLE IF EXISTS owners CASCADE;
CREATE TABLE owners (
   id SERIAL NOT NULL,
   user_id INTEGER NOT NULL,
   event_id INTEGER NOT NULL,
   CONSTRAINT owners_pk PRIMARY KEY (id),
   CONSTRAINT owners_user_id_event_id_uk UNIQUE (user_id, event_id),
   CONSTRAINT owners_event_id_fk FOREIGN KEY (event_id) REFERENCES
   events(id) ON DELETE SET NULL
);
```

```
DROP TABLE IF EXISTS participants CASCADE;
CREATE TABLE participants (
   id SERIAL NOT NULL,
   user_id INTEGER NOT NULL,
   event_id INTEGER NOT NULL,
   CONSTRAINT participants_pk PRIMARY KEY (id),
   CONSTRAINT participants_user_id_event_id_uk UNIQUE (user_id, event_id),
   CONSTRAINT participants_event_id_fk FOREIGN KEY (event_id) REFERENCES
   events(id) ON DELETE SET NULL
);
DROP TABLE IF EXISTS polls CASCADE;
CREATE TABLE polls (
   id SERIAL NOT NULL,
   post_id INTEGER NOT NULL,
   CONSTRAINT polls_pk PRIMARY KEY (id)
);

DROP TABLE IF EXISTS posts CASCADE;
CREATE TABLE posts (
   id SERIAL NOT NULL,
   description text NOT NULL,
   date TIMESTAMP WITH TIME zone NOT NULL,
   event_id INTEGER NOT NULL,
   user_id INTEGER NOT NULL,
   image_id INTEGER NOT NULL,
   CONSTRAINT posts_pk PRIMARY KEY (id),
   CONSTRAINT posts_event_id_fk FOREIGN KEY (event_id) REFERENCES
   events(id) ON DELETE CASCADE,
   CONSTRAINT posts_image_id_fk FOREIGN KEY (image_id) REFERENCES
   images(id) ON DELETE SET NULL
);
```

```
DROP TABLE IF EXISTS users CASCADE;
CREATE TABLE users (
   id SERIAL NOT NULL,
   username text NOT NULL,
   password text NOT NULL,
   email text NOT NULL,
   regist_date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
   first_name text NOT NULL,
   last_name text NOT NULL,
   image_path text NOT NULL,
   city_id INTEGER NOT NULL,
   CONSTRAINT users_pk PRIMARY KEY (id),
   CONSTRAINT users_name_uk UNIQUE (username),
   CONSTRAINT users_email_uk UNIQUE (email),
   CONSTRAINT users_city_id_fk FOREIGN KEY (city_id) REFERENCES
   cities(id) ON DELETE SET NULL
);


ALTER TABLE ONLY cities
   ADD CONSTRAINT cities_country_id_fk FOREIGN KEY (country_id) REFERENCES
   countries(id) ON DELETE SET NULL;

ALTER TABLE ONLY dones
   ADD CONSTRAINT dones_event_id_fk FOREIGN KEY (event_id) REFERENCES
    events(id) ON UPDATE CASCADE;

ALTER TABLE ONLY events
   ADD CONSTRAINT events_owner_id_fk FOREIGN KEY (owner_id) REFERENCES
   users(id) ON UPDATE CASCADE;
```

```
ALTER TABLE ONLY events
    ADD CONSTRAINT events_localization_id_fk FOREIGN KEY (localization_id) REFERENCES
    localizations(id) ON DELETE SET NULL;

ALTER TABLE ONLY events
    ADD CONSTRAINT events_image_id_fk FOREIGN KEY (image_id) REFERENCES
    images(id) ON DELETE SET NULL;

ALTER TABLE ONLY event_invites
    ADD CONSTRAINT event_invites_event_id_fk FOREIGN KEY (event_id) REFERENCES
    not_dones(event_id) ON DELETE CASCADE;

ALTER TABLE ONLY event_invites
    ADD CONSTRAINT event_invites_owner_id_fk FOREIGN KEY (owner_id) REFERENCES
    owners(id) ON DELETE SET NULL;

ALTER TABLE ONLY event_invites
    ADD CONSTRAINT event_invites_receiver_id_fk FOREIGN KEY (receiver_id) REFERENCES
    users(id) ON DELETE SET NULL;

ALTER TABLE ONLY friend_activities
    ADD CONSTRAINT friend_activities_sender_id_fk FOREIGN KEY (sender_id) REFERENCES
    users(id) ON DELETE SET NULL;

ALTER TABLE ONLY friend_activities
    ADD CONSTRAINT friend_activities_receiver_id_fk FOREIGN KEY (receiver_id) REFERENCES
    users(id) ON DELETE SET NULL;

ALTER TABLE ONLY friend_requests
    ADD CONSTRAINT friend_requests_sender_id_fk FOREIGN KEY (sender_id) REFERENCES
    participants(id) ON DELETE CASCADE;
```

```
ALTER TABLE ONLY friend_requests
   ADD CONSTRAINT friend_requests_receiver_id_fk FOREIGN KEY (receiver_id) REFERENCES
   users(id) ON DELETE CASCADE;

ALTER TABLE ONLY friendships
   ADD CONSTRAINT friendships_user_id_1 FOREIGN KEY (user_id_1) REFERENCES
   users(id) ON UPDATE CASCADE;

ALTER TABLE ONLY friendships
   ADD CONSTRAINT friendships_user_id_2 FOREIGN KEY (user_id_2) REFERENCES
   users(id) ON UPDATE CASCADE;

ALTER TABLE ONLY options
   ADD CONSTRAINT options_poll_id_fk FOREIGN KEY (poll_id) REFERENCES
   polls(id) ON DELETE CASCADE;

ALTER TABLE ONLY owners
   ADD CONSTRAINT owners_user_id_fk FOREIGN KEY (user_id) REFERENCES
   users(id) ON UPDATE CASCADE;


ALTER TABLE ONLY participants
   ADD CONSTRAINT participants_user_id_fk FOREIGN KEY (user_id) REFERENCES
   users(id) ON UPDATE CASCADE;

ALTER TABLE ONLY polls
   ADD CONSTRAINT polls_post_id_fk FOREIGN KEY (post_id) REFERENCES
   posts(id) ON DELETE CASCADE;
```

```
ALTER TABLE ONLY posts
    ADD CONSTRAINT posts_user_id_fk FOREIGN KEY (user_id) REFERENCES
    users(id) ON DELETE CASCADE;

UPDATE "users"
SET password = $password,
    email = #email,
    first_name = $first_name,
    last_name = $last_name,
    image_path = $image_path,
    city_id = $city_id
WHERE id = $id;

UPDATE events
SET name = $name,
    date = $date,
    description = $description,
    localization_id = $localization_id,
    image_id = $image_id,
    event_type = $event_type,
    category = $category
WHERE id = $id;

UPDATE dones
SET rating = $rating
WHERE event_id = $event_id;

UPDATE options
SET description = $description
WHERE id = $id;
```

```
UPDATE posts
SET description = $description.
    date = $date,
    image_id = $image_id
WHERE id = $id;

UPDATE current_date
SET date = $date
WHERE id = $id;

DELETE FROM "users"
WHERE id = $id;

DELETE FROM events
WHERE id = $id;

DELETE FROM posts
WHERE id = $id;

DELETE FROM options
WHERE id = $id;

DELETE FROM polls
WHERE id = $id;

DELETE FROM friendships
WHERE id = $id;

DELETE FROM friend_requests
WHERE id = $id;
```

```
DELETE FROM event_invites

WHERE id = $id;


CREATE FUNCTION set_event_as_done() RETURNS TRIGGER AS

$BODY$

BEGIN

  IF EXISTS (SELECT event_id FROM not_done WHERE NEW.event_id = id)

  THEN

    INSERT INTO dones (NEW.event_id, NULL);

    DELETE FROM not_dones WHERE id = NEW.event_id;

  END IF;

  RETURN NEW;

END

$BODY$

LANGUAGE plpgsql;


CREATE TRIGGER set_event_as_done

  BEFORE UPDATE OF date ON current_date

  FOR EACH ROW

  WHEN NEW.date = now()

    EXECUTE PROCEDURE set_event_as_done();



CREATE FUNCTION notificate_event_delete() RETURNS TRIGGER AS

$BODY$

WHILE( SELECT id FROM participants WHERE participants.event_id = OLD.event_id)

BEGIN

  INSERT INTO event_warnings(OLD.event_id, id)

END

$BODY$

LANGUAGE plpgsql;
```

```
CREATE TRIGGER notificate_event_delete
  FOR DELETE OR UPDATE ON events
  FOR EACH ROW
    EXECUTE PROCEDURE notificate_event_delete();

--> QUERIES

--> user profile
SELECT username, last_name, first_name, email, image_path, city_id
  FROM users
  WHERE users.id = $user_id;

--> user friends
SELECT user_id_1, user_id_2
  FROM friendships
  WHERE user_id_1 = $user_id OR user_id_2 = $user_id;

--> user events
SELECT event_id
  FROM participants
  WHERE user_id = $user_id;

SELECT event_id
  FROM owners
  WHERE user_id = $user_id;

--> notifications
SELECT sender_id
  FROM friend_requests
  WHERE receiver_id = $user_id;
```

```
SELECT sender_id, event_id
  FROM friend_activities
  WHERE receiver_id = $user_id;


SELECT owner_id, event_id
  FROM event_invites
  WHERE receiver_id = $user_id;


SELECT event_id, "message"
  FROM event_warnings
  WHERE receiver_id = $user_id;


 --> search user
SELECT id, username, image_path
FROM users
  WHERE username LIKE %$search%
  ORDER BY username;


--> search event
SELECT id, "name", image_id, "date", localization, category
FROM events
  WHERE "name" LIKE %$search% OR localization LIKE %$search% AND event_type = 'public'
  ORDER BY "name";


--> filter by category
SELECT id, "name", image_id, "date", localization, category
FROM events
  WHERE "name" LIKE %$search% OR localization LIKE %$search% AND event_type = 'public'
AND events.category LIKE %$categories%
  ORDER BY "name";
```

```
--> search user
SELECT username, email, image_path
  FROM users
  WHERE username LIKE %$search% OR email LIKE %$search%
ORDER BY username;


--> event page
SELECT events.id, events.name, events.category, events.image_id, events.description,
events."date" users.username
  FROM events, users
  WHERE events.owner_id = users.id AND events.id = $event_id;


SELECT posts.description, posts.id, posts.image_id, posts.user_id
  FROM posts,events
  WHERE posts.event_id = $event_id;


SELECT users.username, users.image_path
  FROM participants
  WHERE users.id = participants.user_id AND participants.event_id=$event_id;
--> Rating of a event
SELECT AVG(rating)
  FROM dones
  WHERE dones.event_id= $event_id;


-->who can i invite to the event
SELECT users.username
  FROM users, events
  WHERE users.id!= event.owner_id AND users.id  NOT IN (
    SELECT user_id
    FROM participants
    WHERE user_id IS NOT NULL AND participants.event_id=$event_id) ;
```

```
--> INDEXES


 CREATE INDEX user_username ON users USING hash (username);


 CREATE INDEX owner_events ON events USING hash(owner_id);


 CREATE INDEX search_events ON events USING GIST (to_tsvector('english', name));


--> INSERTS


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)
                              VALUES (1,'sodales','GUL95ZXR9EX','sodales.at@curae.co.uk','2018-03-
07 22:23:34','Zeph','Griffin','/imgs/natu.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)
                              VALUES (2,'uso1','12345','aliquam.iaculis.lacus@amet.co.uk','2018-04-
07 11:23:34','Ben','Warren','/imgs/natur.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)
                              VALUES (3,'robin','pass123','amet.ante@faucibusleo.net','2018-02-12
15:55:12','Robin','Wright','/imgs/natur.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)
                              VALUES (4,'bar123','semper123','ut.dolor@gmail.com','2017-12-12
23:55:12','Barry','Allen','/imgs/natur.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)
                              VALUES (5,'reddevil','LBAW','Nulla@et.net','2018-02-12
15:55:12','Andrew','Irons','/imgs/november.jpg');
```

```
INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                              VALUES (6,'rpedro10','lbaw1765','rpedro10@iol.pt','2018-01-01
23:55:12','Rui','Araujo','/imgs/fer.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                              VALUES (7,'joss123','CKB15AAW5MM','ante@fleo.com','2018-03-12
12:55:12','Joss','Stone','/imgs/natur.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                              VALUES (8,'top123','SXT16TTW3MH','cursus.et@orciUt.co.uk','2018-
01-03 13:55:12','Chris','Harris','/imgs/natur.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                              VALUES
(9,'roland1','ZYS24FHN5GR','eget.dictum@orciDonec.edu','2018-03-12
09:55:12','Roland','Schitt','/imgs/natur.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                              VALUES (10,'david123','ZUS29FRTGVJ','amet@faucibusleo.net','2018-
01-22 15:07:12','David','Rose','/imgs/natur.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                              VALUES (11,'catones',
'QQQ8EFHNGNR','amec.donec@faucibusleo.net','2017-11-12
15:55:12','Carlos','Antonio','/imgs/november.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                              VALUES (12,'emanem','ASDFGHJKL','donex@sapo.net','2018-01-13
15:55:12','Raheem','Sterling','/imgs/natur.jpg');
```

```
INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                                VALUES (13,'ufoExtis','ZXCVBNM','amet@iol.net','2018-02-12
15:55:12','Delle','Alli','/imgs/pyr.jpg');

INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                                VALUES (14,'ragnar','QAZWSXEDC','risus.In.mi@egestas.com','2018-02-
09 15:55:12','Thor','Ragnarok','/imgs/fer.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                                VALUES
(15,'seth','QWERTYUIOP','aliquet.diam.Sed@tinciduntnibh.co.uk','2018-01-12
15:55:12','Seth','Byers','/imgs/natu.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                                VALUES
(16,'edNorton','TYT71DOD7YN','scelerisque.scelerisque.dui@arcuiaculisenim.ca','2018-02-23
15:55:12','Ed','Norton','/imgs/natur.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                                VALUES (17,'Pacquiao','SXT16TTW3MH','magnis@cursuset.edu','2018-
04-12 15:55:12','Paky','Barret','/imgs/natur.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                                VALUES
(18,'steven','MPS10QPK6UE','arcu.Vestibulum@amet.org','2018-01-02
12:55:12','Donovan','Stevenson','/imgs/pyr.jpg');


INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                                VALUES
(19,'porter123','QIG24ZOK3EM','dui.nec@ultriciesadipiscing.co.uk','2018-02-12
15:55:12','Porter','Osborn','/imgs/natu.jpg');
```

```
INSERT INTO users (id,username,password,email,regist_date,first_name,last_name,
image_path,city_id)

                            VALUES
(20,'human','ZYG87WQA6FX','facilisis.magna.tellus@sociis.net','2018-04-01
11:55:12','Hu','Randolphe','/imgs/pyr.jpg');



INSERT INTO events
(id,name,date,description,owner_id,localization_id,image_id,type,category)

                            VALUES (1,'Antonys Birthday Party', '2018-03-04 12:30:19.000000',
'nunc ac mattis ornare, lectus',1,1,1,'Public','Sports');


INSERT INTO events
(id,name,date,description,owner_id,localization_id,image_id,type,category)

                            VALUES (2,'ENEI','2018-04-04 12:30:19.000000','Nunc quis arcu vel
quam',2,2,2,'Public','Sports');


INSERT INTO events
(id,name,date,description,owner_id,localization_id,image_id,type,category)

                            VALUES (3,'RockInRio','2018-06-04 12:30:19.000000','tempus eu,
ligula. Aenean euismod',3,3,3,'Public','Sports');


INSERT INTO events
(id,name,date,description,owner_id,localization_id,image_id,type,category)

                            VALUES (4,'Nos Alive','2018-08-04 12:30:19.000000','dignissim
pharetra. Nam ac nulla.',3,4,4,'Public','Sports');


INSERT INTO events
(id,name,date,description,owner_id,localization_id,image_id,type,category)

                            VALUES (5,'Christmas Dinner','2018-10-04 12:30:19.000000','dignissim
pharetra. Nam ac nulla.',4,5,5,'Public','Sports');


INSERT INTO events
(id,name,date,description,owner_id,localization_id,image_id,type,category)

                            VALUES (6,'Mark Birthday Party','2018-04-04
12:30:19.000000','dignissim pharetra. Nam ac nulla.',1,1,6,'Public','Sports');
```

```
INSERT INTO events
(id,name,date,description,owner_id,localization_id,image_id,type,category)
                         VALUES (7,'WebSummit','2018-04-04 12:30:19.000000','lorem lorem,
luctus ut, pellentesque',6,6,7,'Public','Sports');


INSERT INTO events
(id,name,date,description,owner_id,localization_id,image_id,type,category)
                         VALUES (8,'Ted Talk','2018-04-04 12:30:19.000000','sollicitudin orci
sem eget massa.',12,2,8,'Public','Sports');



INSERT INTO dones (event_id,rating)
                         VALUES (2,5);
INSERT INTO dones (event_id,rating)
                         VALUES (7,3);


INSERT INTO not_dones (event_id)
                         VALUES (1);
INSERT INTO not_dones (event_id)
                         VALUES (3);
INSERT INTO not_dones (event_id)
                         VALUES (4);
INSERT INTO not_dones (event_id)
                         VALUES (5);
INSERT INTO not_dones (event_id)
                         VALUES (6);
INSERT INTO not_dones (event_id)
                         VALUES (8);
```

```
INSERT INTO participants (id,user_id,event_id)
                        VALUES (1,1,1);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (2,2,1);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (3,2,2);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (4,3,1);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (5,4,1);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (6,5,8);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (7,6,8);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (8,7,3);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (9,12,3);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (10,13,3);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (11,14,3);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (12,15,1);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (13,16,2);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (14,17,3);
INSERT INTO participants (id,user_id,event_id)
                        VALUES (15,18,2);
```

```
INSERT INTO participants (id,user_id,event_id)
                              VALUES (16,19,1);
INSERT INTO participants (id,user_id,event_id)
                              VALUES (17,20,3);
INSERT INTO participants (id,user_id,event_id)
                              VALUES (18,12,6);
INSERT INTO participants (id,user_id,event_id)
                              VALUES (19,12,5);
INSERT INTO participants (id,user_id,event_id)
                              VALUES (20,10,4);



INSERT INTO owners (id,user_id,event_id)
                              VALUES (1,1,1);
INSERT INTO owners (id,user_id,event_id)
                              VALUES (2,2,2);
INSERT INTO owners (id,user_id,event_id)
                              VALUES (3,3,3);
INSERT INTO owners (id,user_id,event_id)
                              VALUES (4,3,4);
INSERT INTO owners (id,user_id,event_id)
                              VALUES (5,5,5);
INSERT INTO owners (id,user_id,event_id)
                              VALUES (6,1,6);
INSERT INTO owners (id,user_id,event_id)
                              VALUES (7,6,7);
INSERT INTO owners (id,user_id,event_id)
                              VALUES (8,12,8);
```

```sql
INSERT INTO images (id,path) VALUES (1,'/imgs/natur.jpg');

INSERT INTO images (id,path) VALUES (2,'/imgs/natu.jpg');

INSERT INTO images (id,path) VALUES (3,'/imgs/pyr.jpg');

INSERT INTO images (id,path) VALUES (4,'/imgs/november.jpg');

INSERT INTO images (id,path) VALUES (5,'/imgs/taj.jpg');

INSERT INTO images (id,path) VALUES (6,'/imgs/fer.jpg');

INSERT INTO images (id,path) VALUES (7,'/imgs/fa1.jpg');

INSERT INTO images (id,path) VALUES (8,'/imgs/fa2.jpg');


INSERT INTO localizations (id,name,address,latitude,longitude,city_id) VALUES (1,'Restaurante O Pirata','Rua da Isabelinha',41.452993,-8.5775364,1);

INSERT INTO localizations (id,name,address,latitude,longitude,city_id) VALUES (2,'FEUP','Rua Roberto Frias',41.1779401,-8.5998763,2);

INSERT INTO localizations (id,name,address,latitude,longitude,city_id) VALUES (3,'Parque da BelaVista','Av. Arlindo Vicente',38.7507558,-9.1265431,3);

INSERT INTO localizations (id,name,address,latitude,longitude,city_id) VALUES (4,'Passeio Maritimo de Alges','Alges',38.697318,-9.2375993,3);

INSERT INTO localizations (id,name,address,latitude,longitude,city_id) VALUES (5,'Hotel Douro','Rua de Agramonte',41.1564707,-8.6288115,2);

INSERT INTO localizations (id,name,address,latitude,longitude,city_id) VALUES (6,'Norte shopping','Matosinhos',41.1825143,-8.6803795,2);



INSERT INTO cities (id,name,country_id) VALUES (1,'Braga',1);

INSERT INTO cities (id,name,country_id) VALUES (1,'Porto',1);

INSERT INTO cities (id,name,country_id) VALUES (1,'Lisboa',1);


INSERT INTO countries (id,name) VALUES (1,'Portugal');

INSERT INTO countries (id,name) VALUES (2,'Espanha');

INSERT INTO countries (id,name) VALUES (3,'USA');
INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (1,'Lorem ipsum
dolor sit amet. ','2018-02-12 15:55:12',1,1,1);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (2,'Lorem ipsum
dolor sit amet. ','2018-02-12 15:55:12',1,3,1);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (3,'Lorem ipsum
dolor sit amet. ','2018-02-12 15:55:12',1,4,1);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (4,'Lorem ipsum
dolor sit amet. ''2018-02-12 15:55:12',,1,16,1);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (5,'Lorem ipsum
dolor sit amet. ','2018-02-12 15:55:12',2,18,1);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (6,'Lorem ipsum
dolor sit amet. ','2018-02-12 15:55:12',4,10,2);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (7,'Lorem ipsum
dolor sit amet. ','2018-02-12 15:55:12',3,17,2);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (8,'Lorem ipsum
dolor sit amet. ','2018-02-12 15:55:12',3,14,2);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (9,'Lorem ipsum
dolor sit amet. ','2018-02-12 15:55:12',4,5,3);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (10,'Lorem ipsum
dolor sit amet. ','2018-01-12 15:55:12',2,16,2);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (11,'Lorem ipsum
dolor sit amet. ','2018-02-12 15:55:12',3,12,2);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (12,'Lorem ipsum
dolor sit amet. ','2018-02-12 15:55:12',1,15,2);

INSERT INTO posts (id,description,date,event_id, user_id, image_id) VALUES (13,'Lorem ipsum
dolor sit amet. ','2018-02-12 15:55:12',5,12,3);


INSERT INTO polls (id,post_id) VALUES (1,1);

INSERT INTO polls (id,post_id) VALUES (2,2);

INSERT INTO polls (id,post_id) VALUES (3,3);

INSERT INTO polls (id,post_id) VALUES (4,4);


INSERT INTO options (id,description,poll_id) VALUES (1,'Bar',1);

INSERT INTO options (id,description,poll_id) VALUES (2,'Cafe',1);

INSERT INTO options (id,description,poll_id) VALUES (3,'Club',1);

INSERT INTO options (id,description,poll_id) VALUES (4,'Home',1);
INSERT INTO options (id,description,poll_id) VALUES (5,'12/05/2018',2);

INSERT INTO options (id,description,poll_id) VALUES (6,'13/05/2018',2);

INSERT INTO options (id,description,poll_id) VALUES (7,'Great',3);

INSERT INTO options (id,description,poll_id) VALUES (8,'Good',3);

INSERT INTO options (id,description,poll_id) VALUES (9,'Available',4);

INSERT INTO options (id,description,poll_id) VALUES (10,'Not Available',4);


INSERT INTO admin (id,username,password,email) VALUES
(1,'admin','admin123','amec.reset@edu.pt');


INSERT INTO friend_requests (id,answer,sender_id,receiver_id) VALUES (1,'YES',1,2);

INSERT INTO friend_requests (id,answer,sender_id,receiver_id) VALUES (3,'YES',1,3);

INSERT INTO friend_requests (id,answer,sender_id,receiver_id) VALUES (4,'NO',1,4);

INSERT INTO friend_requests (id,answer,sender_id,receiver_id) VALUES (5,'YES',1,5);



INSERT INTO friend_activities (id,sender_id,receiver_id,event_id) VALUES (1,1,2,1);

INSERT INTO friend_activities (id,sender_id,receiver_id,event_id) VALUES (2,2,3,2);



INSERT INTO event_invites (id,answer,event_id,owner_id,receiver_id) VALUES (1,'YES',1,1,2);

INSERT INTO event_invites (id,answer,event_id,owner_id,receiver_id) VALUES (2,'YES',1,1,3);

INSERT INTO event_invites (id,answer,event_id,owner_id,receiver_id) VALUES (3,'YES',1,1,4);

INSERT INTO event_invites (id,answer,event_id,owner_id,receiver_id) VALUES (4,'YES',1,1,19);


INSERT INTO friendships (id, user_id_1, user_id_2) VALUES (1,2,3);

INSERT INTO friendships (id, user_id_1, user_id_2) VALUES (2,4,7);


INSERT INTO event_warnings (id, event_id, receiver_id, message) VALUES (1,4,1,"Canceled");


INSERT INTO current_date (id, date) VALUES (1,'2018-04-04 12:30:19.000000');
```

GRUPO1765, 21/03/2018

- Mariana Duarte Guimarães, up201307777@fe.up.pt
- Rui Emanuel Cabral de Almeida Quaresma, up201503005@fe.up.pt
- Rui Pedro Machado Araújo, up201403263@fe.up.pt
- Tiago Duarte Carvalho, up201504461@fe.up.pt