

A6: Indexes, triggers, user functions and population

workpad is a project management platform. This document contains the physical schema of the database, and will show the database population, indexes and triggers, as well as the user functions.

1. Database Workload

Understanding the nature of the workload of the application, and the performance goals, is essential to developing a good database design. The workload includes:

- the most important queries (SELECT) and how often they arise
 - the most important updates (UPDATE, DELETE) and how often they arise
 - the desired performance for these queries and updates
 - an estimate of the number of tuples for each relation
-

1.1. Tuple Estimation

This table includes our estimation of the number of tuples in each relation, as well as our estimation of its growth.

Relation reference	Relation name	Order of magnitude	Estimated growth
R01	member	millions	thousand per day
R02	default_auth	millions	thousand per day
R03	google_auth	millions	thousand per day
R04	project	millions	thousands per day
R05	project_member	tens of millions	tens of thousands per day
R06	task	millions	thousands per day
R07	assigned_to	millions	thousands per day
R08	subtask	millions	thousands per day
R09	task_comment	hundreds of thousands	hundreds per day
R10	forum	hundreds of thousands	hundreds per day
R11	forum_comment	millions	thousands per day
R12	notification	tens of millions	hundreds of thousands per day
R13	admin	units	none

1.2. Frequent Queries

Query	SELECT01
-------	----------

Query **SELECT01**

Description Select the details of a user to display on the profile page or on other pages

Frequency hundreds of thousands per day

SQL code

```
SELECT *  
FROM member  
WHERE username = $username;
```

Query **SELECT02**

Description Select the details of a project to display on the dashboard

Frequency hundreds of thousands per day

SQL code

```
SELECT *  
FROM project  
WHERE id_project = $id_project;
```

Query **SELECT03**

Description Select all the developers (non-Managers) of a project to display in the project settings

Frequency thousands per day

SQL code

```
SELECT member.username  
FROM project, member, project_member  
WHERE project.id_project = $id_project AND project.id_project =  
project_member.id_project AND member.id_member = project_member.id_member  
AND project_member.manager = FALSE;
```

Query **SELECT04**

Description Select all the Managers of a project to display in the project settings

Frequency thousands per day

SQL code

```
SELECT member.username
FROM project, member, project_member
WHERE project.id_project = $id_project AND project.id_project =
project_member.id_project AND member.id_member = project_member.id_member
AND project_member.manager = TRUE;
```

Query	SELECT05
-------	----------

Description	Select all the tasks of a project
-------------	-----------------------------------

Frequency	hundreds of thousands per day
-----------	-------------------------------

SQL code

```
SELECT *
FROM task
WHERE id_project = $id_project
```

Query	SELECT06
-------	----------

Description	Select all the developers assigned to a task
-------------	--

Frequency	tens of thousands per day
-----------	---------------------------

SQL code

```
SELECT member.username
FROM member, task, assigned_to
WHERE task.id_task = $id_task AND task.id_task = assigned_to.id_task AND
assigned_to.id_member = member.id_member;
```

Query	SELECT07
-------	----------

Description	Select all the tasks assigned to a user
-------------	---

Frequency	hundreds of thousands per day
-----------	-------------------------------

SQL code

```
SELECT task.id_task
FROM member, task, assigned_to
WHERE task.id_task = assigned_to.id_task AND member.id_member =
assigned_to.id_member AND member.username = $username;
```

Query	SELECT08
Description	Select all the subtasks of a task
Frequency	tens of thousands per day

SQL code

```
SELECT brief
FROM task, subtask
WHERE task.id_task = subtask.id_task AND task.id_task = $id_task;
```

Query	SELECT09
Description	Select all the comments of a task
Frequency	tens of thousands per day

SQL code

```
SELECT task_comment.content, task_comment.date, member.name
FROM task_comment, task, member
WHERE task_comment.id_task = task.id_task AND task_comment.id_member =
member.id_member AND task.id_task = $id_task;
```

Query	SELECT10
Description	Select all the forums of a project
Frequency	tens of thousands per day

SQL code

```
SELECT forum.topic
FROM project, forum
WHERE forum.project_id = project.id_project AND project.id_project =
$id_project;
```

Query	SELECT11
Description	Select all the comments from a forum
Frequency	tens of thousands per day

SQL code

```
SELECT forum_comment.content, forum_comment.date, member.name
FROM forum, forum_comment, member
WHERE forum.id_forum = forum_comment.id_forum AND forum_comment.id_member =
member.id_member AND forum.id_forum = $id_forum;
```

Query	SELECT12
Description	Select all the notifications of a user
Frequency	hundreds of thousands per day
SQL code	

```
SELECT notification.content, notification.interactable
FROM notification, member
WHERE notification.id_member = member.id_member AND member.username =
$username;
```

Query	SELECT13
Description	Select all the member whose name/username match the search string
Frequency	thousands per day
SQL code	

```
SELECT name, username, about, location, phone_number
FROM to_tsquery($search) AS query, to_tsvector('english', ' ' || name ||
username) AS textsearch
WHERE query @@ textsearch
ORDER BY name ASC
```

Query	SELECT14
Description	Select all the member whose description match the search string
Frequency	thousands per day
SQL code	

```
SELECT name, username, about, location, phone_number
FROM to_tsquery($search) AS query, to_tsvector('english', ' ' || about ||
description || location) AS textsearch
WHERE query @@ textsearch
ORDER BY location ASC
```

1.3. Frequent Updates

Query	UPDATE01
Description	Update the list_name of a task
Frequency	tens of thousands per day
SQL code	

```
UPDATE task
SET list_name = 'Done'
WHERE task.id_task = $id_task;
```

Query	UPDATE02
Description	Update the status of a member (banned)
Frequency	hundreds per day
SQL code	

```
UPDATE member
SET banned = true
WHERE
username = $username;
```

Query	UPDATE03
Description	Update the status of a project (deleted)
Frequency	hundreds per day
SQL code	

```
UPDATE project
SET deleted = true
WHERE
username = $name;
```

Query	UPDATE04
Description	Update the information of a user

Query	UPDATE04
Frequency	thousands per day
SQL code	

```
UPDATE member
SET description = 'Loves to code',
    phone_number = 912345678
WHERE
username = $username;
```

Query	UPDATE05
Description	Edit the text of a comment
Frequency	tens of thousands per day
SQL code	

```
UPDATE task_comment
SET content = 'I agree. Edit: I believe that this information is now
outdated'
WHERE id_task_comment = $id_task_comment;
OR
UPDATE forum_comment
SET content = 'I agree. Edit: I believe that this information is now
outdated'
WHERE id_forum_comment = $id_forum_comment;
```

Query	UPDATE06
Description	Update the color of a project
Frequency	thousands per day
SQL code	

```
UPDATE project
SET color = 'Lilac'
WHERE
id_project = $id_project;
```

Query	UPDATE07
Description	Upgrade a developer to manager

Query	UPDATE07
Frequency	thousands per day

SQL code

```
UPDATE project_member
SET manager = true
FROM member, project
WHERE
project_member.id_member = member.id_member AND
project_member.id_project = project.id_project AND
member.username = $username AND
project.id_project = $project_id
```

Query	UPDATE08
Description	Update the description of a task
Frequency	tens of thousands per day

SQL code

```
UPDATE task
SET description = 'Fix front-end'
WHERE id_task = $id_task;
```

1.4. Frequent Deletes

Query	DELETE01
Description	Delete a task
Frequency	thousands per day

SQL code

```
DELETE FROM task
WHERE id_task = $id_task
```

Query	DELETE02
Description	Delete a subtask
Frequency	thousands per day

Query DELETE02

SQL code

```
DELETE FROM subtask
WHERE id_subtask = $id_subtask
```

Query DELETE03

Description	Remove a member from a task
--------------------	-----------------------------

Frequency	thousands per day
------------------	-------------------

SQL code

```
DELETE FROM assigned_to
WHERE id_member = $id_member AND
      id_task = $id_task
```

Query DELETE04

Description	Remove a member from a project
--------------------	--------------------------------

Frequency	thousands per day
------------------	-------------------

SQL code

```
DELETE FROM project_member
WHERE id_member = $id_member AND
      id_project = $id_project
```

1.5. Frequent Inserts

Query INSERT01

Description	Create an account
--------------------	-------------------

Frequency	tens of thousand per day
------------------	--------------------------

SQL code

```
INSERT INTO member (name, username, email)
VALUES($name, $username, $email);
```

```
INSERT INTO default_auth (id_member, password)
VALUES($id_member, $password);
```

Note: Will later require a transaction

Query	INSERT02
Description	Create a project
Frequency	thousands per day
SQL code	

```
INSERT INTO project (name, color)
VALUES ($name, $color);
```

Query	INSERT03
Description	Adding a member to a project
Frequency	tens of thousands per day
SQL code	

```
INSERT INTO project_member (id_project, id_member, manager)
VALUES ($id_project, $id_member, $manager_val);
```

Query	INSERT04
Description	Create a task
Frequency	tens of thousands per day
SQL code	

```
INSERT INTO task (id_project, name)
VALUES ($id_project, $name);
```

Query	INSERT05
Description	Assign a user to a task
Frequency	thousands per day
SQL code	

```
INSERT INTO assigned_to (id_member, id_task)
VALUES ($id_member, $id_task);
```

Query	INSERT06
-------	----------

Description	Create a subtask
-------------	------------------

Frequency	thousands per day
-----------	-------------------

SQL code

```
INSERT INTO subtask (id_task, brief)
VALUES ($id_task, $brief);
```

Query	INSERT07
-------	----------

Description	Create a task comment
-------------	-----------------------

Frequency	hundreds per day
-----------	------------------

SQL code

```
INSERT INTO task_comment (id_member, id_task, content)
VALUES ($id_member, $id_task, $content);
```

Query	INSERT08
-------	----------

Description	Create a forum comment
-------------	------------------------

Frequency	thousands per day
-----------	-------------------

SQL code

```
INSERT INTO forum_comment (id_member, id_task, content)
VALUES ($id_member, $id_task, $content);
```

Query	INSERT09
-------	----------

Description	Creating a forum topic
-------------	------------------------

Frequency	hundreds per day
-----------	------------------

SQL code

```
INSERT INTO forum (id_project, topic)
VALUES ($id_project, $topic);
```

Query	INSERT10
Description	Creating a notification
Frequency	hundreds of thousands per day
SQL code	

```
INSERT INTO notification (id_member, content, interactable)
VALUES ($id_member, $content, $interactable);
```

2. Proposed Indexes

2.1. Performance Indexes

Index	IDX01
Related queries	SELECT03, SELECT04
Relation	project_member
Attribute	id_project, manager
Type	B-Tree
Cardinality	High
Clustering	No
Justification	Used to search the developers or managers of a project, so it has to be fast because it's executed several times; Doesn't need range query support; Cardinality is high, so there is no need for clustering.
SQL code	

```
CREATE INDEX developer_project ON project_member (id_project, manager);
```

Index	IDX02
Related queries	SELECT05

Index	IDX02
Relation	task
Attribute	id_project
Type	Hash
Cardinality	High
Clustering	No
Justification	Used to search the tasks of a project, so it has to be fast because it's executed several times; Doesn't need range query support; Cardinality is high, so there is no need for clustering.
SQL code	

```
CREATE INDEX task_project ON task USING hash(id_project);
```

Index	IDX03
Related queries	SELECT06
Relation	assigned_to
Attribute	id_task
Type	Hash
Cardinality	High
Clustering	No
Justification	Used to search the developers of a task, so it has to be fast because it's executed several times; Doesn't need range query support; Cardinality is high, so there is no need for clustering.
SQL code	

```
CREATE INDEX member_task ON assigned_to USING hash(id_task);
```

Index	IDX04
Related queries	SELECT07
Relation	assigned_to

Index	IDX04
Attribute	id_member
Type	Hash
Cardinality	High
Clustering	No
Justification	Used to search all the tasks a developer is assigned to, so it has to be fast because it's executed several times; Doesn't need range query support; Cardinality is high, so there is no need for clustering.

SQL code

```
CREATE INDEX task_member ON assigned_to USING hash(id_member);
```

Index	IDX05
Related queries	SELECT08
Relation	subtask
Attribute	id_task
Type	Hash
Cardinality	High
Clustering	No
Justification	Used to search the subtasks of a task, so it has to be fast because it's executed several times; Doesn't need range query support; Cardinality is high, so there is no need for clustering.

SQL code

```
CREATE INDEX subtask_task ON subtask USING hash(id_task);
```

Index	IDX06
Related queries	SELECT09
Relation	task_comment
Attribute	id_task

Index	IDX06
Type	Hash
Cardinality	High
Clustering	No
Justification	Used to search the developers of a task, so it has to be fast because it's executed many times and the table is very large; Doesn't need range query support; Cardinality is high, so there is no need for clustering.

SQL code

```
CREATE INDEX comment_task ON task_comment USING hash(id_task);
```

Index	IDX07
Related queries	SELECT10
Relation	forum
Attribute	id_project
Type	Hash
Cardinality	High
Clustering	No
Justification	Used to search the topics of a project's forum so it has to be fast because it's executed several times; Doesn't need range query support; Cardinality is high, so there is no need for clustering.

SQL code

```
CREATE INDEX forum_project ON forum USING hash(id_project);
```

Index	IDX08
Related queries	SELECT11
Relation	forum_comment
Attribute	id_forum
Type	Hash

Index	IDX08
Cardinality	High
Clustering	No
Justification	Used to search the comments of a forum so it has to be fast because it's executed several times; Doesn't need range query support; Cardinality is high, so there is no need for clustering.
SQL code	
<pre>CREATE INDEX comment_forum ON forum_comment USING hash(id_forum);</pre>	

Index	IDX09
Related queries	SELECT12
Relation	notification
Attribute	id_member
Type	Hash
Cardinality	High
Clustering	No
Justification	Used to search for all the notifications of a member, so it has to be fast because it's executed several times; Doesn't need range query support; Cardinality is high, so there is no need for clustering.
SQL code	
<pre>CREATE INDEX notification_member ON notification USING hash(id_member);</pre>	

2.2. Full-text Search Indexes

Index	IDX01
Related queries	SELECT13
Relation	member
Attribute	name
Type	GiST

Index	IDX01
Clustering	Yes
Justification	To improve the performance of full text searches while searching for members by their names or usernames; GiST because it's better for dynamic data.
SQL code	
<pre>CREATE INDEX search_idx_name ON MEMBER USING GIST (to_tsvector(' ' name username))</pre>	

Index	IDX02
Related queries	SELECT14
Relation	member
Attribute	location
Type	GiST
Clustering	Yes
Justification	To improve the performance of full text searches while searching for members and their abouts, descriptions or locations; GiST because it's better for dynamic data.
SQL code	
<pre>CREATE INDEX search_idx_desc ON MEMBER USING GIST (to_tsvector(' ' about description location))</pre>	

3. Triggers

Trigger	TRIGGER01
Description	When a Member is banned or deletes his account, all tasks assigned to him and his connections to projects are removed
SQL code	
<pre>CREATE OR REPLACE FUNCTION inactive_user_remove_assigns() RETURNS TRIGGER AS \$BODY\$ begin IF NEW.banned = true THEN</pre>	

```

DELETE FROM assigned_to WHERE id_member = NEW.id_member;
DELETE FROM project_member WHERE id_member = NEW.id_member;
END IF;
IF NEW.deleted = true THEN
DELETE FROM assigned_to WHERE id_member = NEW.id_member;
DELETE FROM project_member WHERE id_member = NEW.id_member;
END IF;
return new;
end;
$BODY$
language plpgsql;

CREATE TRIGGER inactive_user_remove_assigns
AFTER UPDATE ON member
FOR EACH ROW
EXECUTE PROCEDURE inactive_user_remove_assigns();

```

Trigger	TRIGGER02
----------------	------------------

Description	In a forum, there can't be duplicate topic names
--------------------	--

SQL code

```

CREATE OR REPLACE FUNCTION unique_forum_titles()
RETURNS TRIGGER AS
$BODY$
begin
IF EXISTS (SELECT * FROM forum WHERE id_project = NEW.id_project AND topic
= NEW.topic) THEN
RAISE EXCEPTION 'A topic with the same name already exists';
END IF;
return new;
end;
$BODY$
language plpgsql;

CREATE TRIGGER unique_forum_titles
BEFORE INSERT ON forum
FOR EACH ROW
EXECUTE PROCEDURE unique_forum_titles();

```

Trigger	TRIGGER03
----------------	------------------

Description	When a task is deleted, also delete its subtasks, assignments and comments
--------------------	--

SQL code

```

CREATE OR REPLACE FUNCTION delete_task()
RETURNS TRIGGER AS
$BODY$
begin
    DELETE FROM assigned_to WHERE OLD.id_task = assigned_to.id_task;
    DELETE FROM subtask WHERE OLD.id_task = subtask.id_task;
    DELETE FROM task_comment WHERE OLD.id_task = task_comment.id_task;
return new;
end;
$BODY$
language plpgsql;

CREATE TRIGGER delete_task
BEFORE DELETE ON task
FOR EACH ROW
EXECUTE PROCEDURE delete_task();

```

Trigger	TRIGGER04
---------	-----------

Description	If a manager is the only one in a project, he cannot renounce its role
--------------------	--

SQL code

```

CREATE OR REPLACE FUNCTION only_manager()
RETURNS TRIGGER AS
$BODY$
begin
    IF (SELECT COUNT(*) FROM(SELECT id_member FROM project_member WHERE
id_project = NEW.id_project AND manager = true) AS id) = 0 THEN
        RAISE EXCEPTION 'Cannot remove only manager';
    END IF;
return new;
end;
$BODY$
language plpgsql;

CREATE TRIGGER only_manager
AFTER UPDATE ON project_member
FOR EACH ROW
EXECUTE PROCEDURE only_manager();

```

Trigger	TRIGGER05
---------	-----------

Description	If a user deletes his account/is banned, all project where he was the only manager must have all their other developers become managers, to avoid having projects with no managers
--------------------	--

SQL code

```
CREATE OR REPLACE FUNCTION make_everyone_manager()
RETURNS TRIGGER AS
$BODY$
begin
    IF (old.manager = true AND (SELECT count(*) FROM project_member WHERE
id_project = old.id_project AND manager = true) = 0) THEN
        UPDATE project_member SET manager = true WHERE id_project =
old.id_project;
    END IF;
return old;
end;
$BODY$
language plpgsql;

CREATE TRIGGER make_everyone_manager
AFTER DELETE ON project_member
FOR EACH ROW
EXECUTE PROCEDURE make_everyone_manager();
```

4. Complete SQL Code

Link to Database Code: [sqlscript.pgsql](#)

```
-- Drop Types

DROP TYPE IF EXISTS List CASCADE;
DROP TYPE IF EXISTS Color CASCADE;

-- Drop Triggers

DROP TRIGGER IF EXISTS inactive_user_remove_assigns on member;
DROP TRIGGER IF EXISTS unique_forum_titles on forum;
DROP TRIGGER IF EXISTS delete_task on task;
DROP TRIGGER IF EXISTS only_manager on project_member;
DROP TRIGGER IF EXISTS make_everyone_manager on project_member;

DROP FUNCTION IF EXISTS inactive_user_remove_assigns();
DROP FUNCTION IF EXISTS unique_forum_titles();
DROP FUNCTION IF EXISTS delete_task();
DROP FUNCTION IF EXISTS only_manager();
DROP FUNCTION IF EXISTS make_everyone_manager();

-- Drop Tables
```

```
DROP TABLE IF EXISTS member, default_auth, google_auth, project,
project_member, task, assigned_to, subtask, task_comment, forum,
forum_comment, notification, admin CASCADE;

-- TYPES

CREATE TYPE List AS ENUM ('To Do', 'In Progress', 'Pending Approval',
'Done');

CREATE TYPE Color AS ENUM ('Orange', 'Yellow', 'Red', 'Green', 'Lilac',
'Sky', 'Blue', 'Purple', 'Emerald', 'Bordeaux', 'Golden', 'Brown');

-- TABLES

CREATE TABLE member (
    id_member SERIAL PRIMARY KEY,
    name text NOT NULL,
    username text NOT NULL CONSTRAINT user_username_uk UNIQUE,
    email text NOT NULL CONSTRAINT user_email_uk UNIQUE,
    about text,
    description text,
    location text,
    phone_number INTEGER,
    region_code INTEGER,
    banned BOOLEAN NOT NULL DEFAULT FALSE,
    deleted BOOLEAN NOT NULL DEFAULT FALSE
);

CREATE TABLE default_auth (
    id_member INTEGER NOT NULL REFERENCES member (id_member) ON UPDATE
CASCADE,
    password TEXT NOT NULL
);

CREATE TABLE google_auth (
    id_member INTEGER NOT NULL REFERENCES member (id_member) ON UPDATE
CASCADE,
    password TEXT NOT NULL
);

CREATE TABLE project (
    id_project SERIAL PRIMARY KEY,
    name text NOT NULL,
    color Color NOT NULL DEFAULT 'Sky',
    end_date TIMESTAMP WITH TIME zone,
    deleted BOOLEAN DEFAULT FALSE
);

CREATE TABLE project_member (
    id_project INTEGER NOT NULL REFERENCES project (id_project) ON UPDATE
CASCADE,
```

```
    id_member INTEGER NOT NULL REFERENCES member (id_member) ON UPDATE
    CASCADE,
    manager BOOLEAN NOT NULL
);

CREATE TABLE task (
    id_task SERIAL PRIMARY KEY,
    id_project INTEGER NOT NULL REFERENCES project (id_project) ON UPDATE
    CASCADE,
    list_name List DEFAULT 'To Do' NOT NULL,
    name text NOT NULL,
    description text,
    creation_date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
    due_date TIMESTAMP WITH TIME zone,
    CONSTRAINT task_date_ck CHECK (due_date > creation_date),
    issue text
);

CREATE TABLE assigned_to (
    id_member INTEGER NOT NULL REFERENCES member (id_member) ON UPDATE
    CASCADE,
    id_task INTEGER NOT NULL REFERENCES task (id_task) ON UPDATE CASCADE
);

CREATE TABLE subtask(
    id_subtask SERIAL PRIMARY KEY,
    id_task INTEGER NOT NULL REFERENCES task (id_task) ON UPDATE CASCADE,
    brief text NOT NULL
);

CREATE TABLE task_comment (
    id_task_comment SERIAL PRIMARY KEY,
    id_member INTEGER NOT NULL REFERENCES member (id_member) ON UPDATE
    CASCADE,
    id_task INTEGER NOT NULL REFERENCES task (id_task) ON UPDATE CASCADE,
    content text NOT NULL,
    "date" TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL
);

CREATE TABLE forum (
    id_forum SERIAL PRIMARY KEY,
    id_project INTEGER NOT NULL REFERENCES project (id_project) ON UPDATE
    CASCADE,
    topic text NOT NULL
);

CREATE TABLE forum_comment (
    id_forum_comment SERIAL PRIMARY KEY,
    id_member INTEGER NOT NULL REFERENCES member (id_member) ON UPDATE
    CASCADE,
    id_forum INTEGER NOT NULL REFERENCES forum (id_forum) ON UPDATE
    CASCADE,
    content text NOT NULL,
    "date" TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL
```

```

);

CREATE TABLE notification (
    id_notification SERIAL PRIMARY KEY,
    id_member INTEGER NOT NULL REFERENCES member (id_member) ON UPDATE
    CASCADE,
    content text NOT NULL,
    seen BOOLEAN DEFAULT FALSE NOT NULL,
    interactable BOOLEAN NOT NULL,
    link text
);

CREATE TABLE admin (
    id_admin SERIAL PRIMARY KEY,
    username text NOT NULL CONSTRAINT admin_name UNIQUE,
    password text NOT NULL
);

-- INDEXES

CREATE INDEX developer_project ON project_member (id_project,manager);
CREATE INDEX task_project ON task USING hash(id_project);
CREATE INDEX member_task ON assigned_to USING hash(id_task);
CREATE INDEX task_member ON assigned_to USING hash(id_member);
CREATE INDEX subtask_task ON subtask USING hash(id_task);
CREATE INDEX comment_task ON task_comment USING hash(id_task);
CREATE INDEX forum_project ON forum USING hash(id_project);
CREATE INDEX comment_forum ON forum_comment USING hash(id_forum);
CREATE INDEX notification_member ON notification USING hash(id_member);

CREATE INDEX search_idx_name ON MEMBER USING GIST (to_tsvector('english', '
' || name || username));
CREATE INDEX search_idx_desc ON MEMBER USING GIST (to_tsvector('english', '
' || about || description || location));

-- TRIGGERS and UDFs

CREATE OR REPLACE FUNCTION inactive_user_remove_assigns()
RETURNS TRIGGER AS
$BODY$
begin
IF NEW.banned = true THEN
    DELETE FROM assigned_to WHERE id_member = NEW.id_member;
    DELETE FROM project_member WHERE id_member = NEW.id_member;
END IF;
IF NEW.deleted = true THEN
    DELETE FROM assigned_to WHERE id_member = NEW.id_member;
    DELETE FROM project_member WHERE id_member = NEW.id_member;
END IF;
return new;
end;

```

```
$BODY$
language plpgsql;

CREATE TRIGGER inactive_user_remove_assigns
  AFTER UPDATE ON member
  FOR EACH ROW
  EXECUTE PROCEDURE inactive_user_remove_assigns();

CREATE OR REPLACE FUNCTION unique_forum_titles()
RETURNS TRIGGER AS
$BODY$
begin
IF EXISTS (SELECT * FROM forum WHERE id_project = NEW.id_project AND topic
= NEW.topic) THEN
  RAISE EXCEPTION 'A topic with the same name already exists';
END IF;
return new;
end;
$BODY$
language plpgsql;

CREATE TRIGGER unique_forum_titles
  BEFORE INSERT ON forum
  FOR EACH ROW
  EXECUTE PROCEDURE unique_forum_titles();

CREATE OR REPLACE FUNCTION delete_task()
RETURNS TRIGGER AS
$BODY$
begin
  DELETE FROM assigned_to WHERE OLD.id_task = assigned_to.id_task;
  DELETE FROM subtask WHERE OLD.id_task = subtask.id_task;
  DELETE FROM task_comment WHERE OLD.id_task = task_comment.id_task;
return new;
end;
$BODY$
language plpgsql;

CREATE TRIGGER delete_task
  BEFORE DELETE ON task
  FOR EACH ROW
  EXECUTE PROCEDURE delete_task();

CREATE OR REPLACE FUNCTION only_manager()
RETURNS TRIGGER AS
$BODY$
begin
  IF (SELECT COUNT(*) FROM (SELECT id_member FROM project_member WHERE
id_project = NEW.id_project AND manager = true) AS id) = 0 THEN
    RAISE EXCEPTION 'Cannot remove only manager';
  END IF;
```



```
return new;
end;
$BODY$
language plpgsql;

CREATE TRIGGER only_manager
  AFTER UPDATE ON project_member
  FOR EACH ROW
  EXECUTE PROCEDURE only_manager();

CREATE OR REPLACE FUNCTION make_everyone_manager()
RETURNS TRIGGER AS
$BODY$
begin
  IF (old.manager = true AND (SELECT count(*) FROM project_member WHERE
id_project = old.id_project AND manager = true) = 0) THEN
    UPDATE project_member SET manager = true WHERE id_project =
old.id_project;
  END IF;
return old;
end;
$BODY$
language plpgsql;

CREATE TRIGGER make_everyone_manager
  AFTER DELETE ON project_member
  FOR EACH ROW
  EXECUTE PROCEDURE make_everyone_manager();
```

Link to Database Population Code: [data.sql](#)

```
-- deletes (order is important!)

DELETE FROM admin;
DELETE FROM subtask;
DELETE FROM task_comment;
DELETE FROM forum_comment;
DELETE FROM default_auth;
DELETE FROM google_auth;
DELETE FROM project_member;
DELETE FROM assigned_to;
DELETE FROM task;
DELETE FROM forum;
DELETE FROM project;
DELETE FROM notification;
```

```
DELETE FROM member;
```

```
-- member (id_member, name, username, email, about, description, location,  
phone_number, region_code, banned)
```

```
INSERT INTO member (id_member, name, username, email, about, description,  
location, phone_number, region_code, banned) VALUES(1, 'Cláudio Lemos',  
'claudiolemos', 'claudio@mail.com', 'Aliquam vestibulum, sem et congue  
tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.',  
'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse  
euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam  
vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida  
ante tortor eget magna. Curabitur euismod enim quis accumsan semper.  
Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.',  
'Porto, Portugal', '910000001', '351', false);
```

```
INSERT INTO member (id_member, name, username, email, about, description,  
location, phone_number, region_code, banned) VALUES(2, 'Fernando Alves',  
'fernandoalves', 'fernando@mail.com', 'Aliquam vestibulum, sem et congue  
tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.',  
'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse  
euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam  
vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida  
ante tortor eget magna. Curabitur euismod enim quis accumsan semper.  
Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.',  
'Porto, Portugal', '910000002', '351', false);
```

```
INSERT INTO member (id_member, name, username, email, about, description,  
location, phone_number, region_code, banned) VALUES(3, 'João Maduro',  
'joaomaduro', 'joao@mail.com', 'Aliquam vestibulum, sem et congue  
tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.',  
'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse  
euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam  
vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida  
ante tortor eget magna. Curabitur euismod enim quis accumsan semper.  
Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.',  
'Porto, Portugal', '910000003', '351', false);
```

```
INSERT INTO member (id_member, name, username, email, about, description,  
location, phone_number, region_code, banned) VALUES(4, 'Pedro Gonçalves',  
'pedrogoncalves', 'pedro@mail.com', 'Aliquam vestibulum, sem et congue  
tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.',  
'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse  
euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam  
vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida  
ante tortor eget magna. Curabitur euismod enim quis accumsan semper.  
Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.',  
'Porto, Portugal', '910000004', '351', false);
```

```
INSERT INTO member (id_member, name, username, email, about, description,  
location, phone_number, region_code, banned) VALUES(5, 'Alexandra Mendes',  
'alexandramendes', 'alexandra@mail.com', 'Aliquam vestibulum, sem et congue  
tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.',  
'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse  
euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam  
vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida  
ante tortor eget magna. Curabitur euismod enim quis accumsan semper.  
Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.',  
'Porto, Portugal', '910000005', '351', false);
```

```
INSERT INTO member (id_member, name, username, email, about, description, location, phone_number, region_code, banned) VALUES(6, 'Fábio Bernardo', 'fabiobernardo', 'fabio@mail.com', 'Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna. Curabitur euismod enim quis accumsan semper. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.', 'Porto, Portugal', '9100000006', '351', false);
INSERT INTO member (id_member, name, username, email, about, description, location, phone_number, region_code, banned) VALUES(7, 'Duarte Oliveira', 'duarteoliveira', 'duarte@mail.com', 'Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna. Curabitur euismod enim quis accumsan semper. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.', 'Porto, Portugal', '9100000007', '351', false);
INSERT INTO member (id_member, name, username, email, about, description, location, phone_number, region_code, banned) VALUES(8, 'Diogo Silva', 'diogosilva', 'diogo@mail.com', 'Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna. Curabitur euismod enim quis accumsan semper. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.', 'Porto, Portugal', '9100000008', '351', false);
INSERT INTO member (id_member, name, username, email, about, description, location, phone_number, region_code, banned) VALUES(9, 'Carolina Soares', 'carolinasoares', 'carolina@mail.com', 'Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna. Curabitur euismod enim quis accumsan semper. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.', 'Porto, Portugal', '9100000009', '351', false);
INSERT INTO member (id_member, name, username, email, about, description, location, phone_number, region_code, banned) VALUES(10, 'Rúben Barreiro', 'rubenbarreiro', 'ruben@mail.com', 'Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna. Curabitur euismod enim quis accumsan semper. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.', 'Porto, Portugal', '9100000010', '351', false);
INSERT INTO member (id_member, name, username, email, about, description, location, phone_number, region_code, banned) VALUES(11, 'Vitor Lopes', 'vitorlopes', 'vitor@mail.com', 'Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.',
```

```
'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna. Curabitur euismod enim quis accumsan semper. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.', 'Porto, Portugal', '910000011', '351', false);
INSERT INTO member (id_member, name, username, email, about, description, location, phone_number, region_code, banned) VALUES(12, 'António Marques', 'antoniomarkes', 'antonio@mail.com', 'Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna. Curabitur euismod enim quis accumsan semper. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.', 'Porto, Portugal', '910000012', '351', false);
INSERT INTO member (id_member, name, username, email, about, description, location, phone_number, region_code, banned) VALUES(13, 'Rita Pinto', 'ritapinto', 'rita@mail.com', 'Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna. Curabitur euismod enim quis accumsan semper. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.', 'Porto, Portugal', '910000013', '351', false);
INSERT INTO member (id_member, name, username, email, about, description, location, phone_number, region_code, banned) VALUES(14, 'Daniela Pereira', 'danielapereira', 'daniela@mail.com', 'Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna. Curabitur euismod enim quis accumsan semper. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.', 'Porto, Portugal', '910000014', '351', false);
INSERT INTO member (id_member, name, username, email, about, description, location, phone_number, region_code, banned) VALUES(15, 'Joana Ramos', 'joanaramos', 'joana@mail.com', 'Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna. Curabitur euismod enim quis accumsan semper. Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.', 'Porto, Portugal', '910000015', '351', false);
INSERT INTO member (id_member, name, username, email, about, description, location, phone_number, region_code, banned) VALUES(16, 'Mariana Azevedo', 'marianaazevedo', 'mariana@mail.com', 'Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget. Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna. Curabitur euismod enim quis accumsan semper.
```

```

Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.',
'Porto, Portugal', '910000016', '351', false);
INSERT INTO member (id_member, name, username, email, about, description,
location, phone_number, region_code, banned) VALUES(17, 'Maria Cardoso',
'mariacardoso', 'maria@mail.com', 'Aliquam vestibulum, sem et congue
tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.',
'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
eismod commodo velit, ut tincidunt urna consectetur eget. Aliquam
vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida
ante tortor eget magna. Curabitur euismod enim quis accumsan semper.
Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.',
'Porto, Portugal', '910000017', '351', false);
INSERT INTO member (id_member, name, username, email, about, description,
location, phone_number, region_code, banned) VALUES(18, 'Helena Sampaio',
'helenasampaio', 'helen@mail.com', 'Aliquam vestibulum, sem et congue
tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.',
'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
eismod commodo velit, ut tincidunt urna consectetur eget. Aliquam
vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida
ante tortor eget magna. Curabitur euismod enim quis accumsan semper.
Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.',
'Porto, Portugal', '910000018', '351', false);
INSERT INTO member (id_member, name, username, email, about, description,
location, phone_number, region_code, banned) VALUES(19, 'Paula Rocha',
'paularocha', 'paula@mail.com', 'Aliquam vestibulum, sem et congue
tincidunt, nibh ligula commodo urna, at gravida ante tortor eget magna.',
'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
eismod commodo velit, ut tincidunt urna consectetur eget. Aliquam
vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at gravida
ante tortor eget magna. Curabitur euismod enim quis accumsan semper.
Interdum et malesuada fames ac ante ipsum primis in faucibus. Proin.',
'Porto, Portugal', '910000019', '351', false);
INSERT INTO member (id_member, name, username, email, about, description,
location, phone_number, region_code, banned) VALUES(20, 'Beatriz
Henriques', 'beatrizhenriques', 'beatriz@mail.com', 'Aliquam vestibulum,
sem et congue tincidunt, nibh ligula commodo urna, at gravida ante tortor
eget magna.', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Suspendisse euismod commodo velit, ut tincidunt urna consectetur eget.
Aliquam vestibulum, sem et congue tincidunt, nibh ligula commodo urna, at
gravida ante tortor eget magna. Curabitur euismod enim quis accumsan
semper. Interdum et malesuada fames ac ante ipsum primis in faucibus.
Proin.', 'Porto, Portugal', '910000020', '351', false);

SELECT setval(pg_get_serial_sequence('member', 'id_member'), (SELECT
MAX(id_member) FROM member));

-- default_auth (id_member, password)
INSERT INTO default_auth (id_member, password) VALUES(1,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO default_auth (id_member, password) VALUES(2,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO default_auth (id_member, password) VALUES(3,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');

```



```
INSERT INTO default_auth (id_member, password) VALUES(4,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO default_auth (id_member, password) VALUES(5,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO default_auth (id_member, password) VALUES(6,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO default_auth (id_member, password) VALUES(7,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO default_auth (id_member, password) VALUES(8,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO default_auth (id_member, password) VALUES(9,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO default_auth (id_member, password) VALUES(10,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');

-- google_auth (id_member, password)
INSERT INTO google_auth (id_member, password) VALUES(11,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO google_auth (id_member, password) VALUES(12,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO google_auth (id_member, password) VALUES(13,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO google_auth (id_member, password) VALUES(14,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO google_auth (id_member, password) VALUES(15,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO google_auth (id_member, password) VALUES(16,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO google_auth (id_member, password) VALUES(17,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO google_auth (id_member, password) VALUES(18,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO google_auth (id_member, password) VALUES(19,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO google_auth (id_member, password) VALUES(20,
'03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');

-- project (id_project, name, color, end_date, deleted)
INSERT INTO project (id_project, name, color, end_date, deleted) VALUES(1,
'workpad', 'Green', '2019-06-02 12:00:00', false);
INSERT INTO project (id_project, name, color, end_date, deleted) VALUES(2,
'NIFEUP', 'Orange', '2019-06-02 12:00:00', false);
INSERT INTO project (id_project, name, color, end_date, deleted) VALUES(3,
'IEEE', 'Blue', '2019-06-02 12:00:00', false);
INSERT INTO project (id_project, name, color, end_date, deleted) VALUES(4,
'AEFEUP', 'Yellow', '2019-06-02 12:00:00', false);
INSERT INTO project (id_project, name, color, end_date, deleted) VALUES(5,
'Tuga POP', 'Red', '2019-06-02 12:00:00', false);

SELECT setval(pg_get_serial_sequence('project', 'id_project'), (SELECT
MAX(id_project) FROM project));
```

```
-- project_member (id_project, id_member, manager)
INSERT INTO project_member (id_project, id_member, manager) VALUES(1, 1,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(1, 2,
true);
INSERT INTO project_member (id_project, id_member, manager) VALUES(1, 3,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(1, 4,
true);
INSERT INTO project_member (id_project, id_member, manager) VALUES(2, 4,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(2, 5,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(2, 7,
true);
INSERT INTO project_member (id_project, id_member, manager) VALUES(2, 10,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(2, 11,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(2, 12,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(2, 13,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(2, 14,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(2, 15,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(3, 8,
true);
INSERT INTO project_member (id_project, id_member, manager) VALUES(3, 9,
true);
INSERT INTO project_member (id_project, id_member, manager) VALUES(3, 16,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(3, 17,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(3, 18,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(3, 19,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(3, 20,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(4, 1,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(4, 2,
true);
INSERT INTO project_member (id_project, id_member, manager) VALUES(4, 3,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(4, 4,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(4, 5,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(4, 6,
false);
```

```
INSERT INTO project_member (id_project, id_member, manager) VALUES(4, 7,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(4, 8,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(4, 9,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(4, 10,
false);
INSERT INTO project_member (id_project, id_member, manager) VALUES(5, 1,
true);

-- task (id_task, id_project, list_name, name, description, creation_date,
due_date, issue)
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(1, 1, 'To Do', 'Create database',
null, '2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(2, 1, 'In Progress', 'Storyboards',
null, '2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(3, 1, 'Pending Approval', 'Create
workpad logo', null, '2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(4, 1, 'Done', 'Create team', null,
'2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(5, 2, 'To Do', 'Start projects',
null, '2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(6, 2, 'In Progress', 'Decide
projects', null, '2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(7, 2, 'Pending Approval',
'Brainstorm ideas for projects', null, '2019-04-01 12:00:00', '2019-06-02
12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(8, 2, 'Done', 'Create teams', null,
'2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(9, 3, 'To Do', 'Start projects',
null, '2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(10, 3, 'In Progress', 'Decide
projects', null, '2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(11, 3, 'Pending Approval',
'Brainstorm ideas for projects', null, '2019-04-01 12:00:00', '2019-06-02
12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(12, 3, 'Done', 'Create teams', null,
'2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(13, 4, 'To Do', 'Book artists',
```



```

null, '2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(14, 4, 'In Progress', 'Decide party
themes', null, '2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(15, 4, 'Pending Approval',
'Brainstorm ideas for parties', null, '2019-04-01 12:00:00', '2019-06-02
12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(16, 4, 'Done', 'Create teams', null,
'2019-04-01 12:00:00', '2019-06-02 12:00:00', null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(17, 5, 'To Do', 'Choose theme',
null, '2019-04-01 12:00:00', null, null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(18, 5, 'In Progress', 'Implement
Wordpress', null, '2019-04-01 12:00:00', null, null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(19, 5, 'Pending Approval', 'Create
logo', null, '2019-04-01 12:00:00', null, null);
INSERT INTO task (id_task, id_project, list_name, name, description,
creation_date, due_date, issue) VALUES(20, 5, 'Done', 'Buy domain', null,
'2019-04-01 12:00:00', null, null);

```

```

SELECT setval(pg_get_serial_sequence('task', 'id_task'), (SELECT
MAX(id_task) FROM task));

```

```

-- assigned_to (id_member, id_task)
INSERT INTO assigned_to (id_member, id_task) VALUES (1,1);
INSERT INTO assigned_to (id_member, id_task) VALUES (2,2);
INSERT INTO assigned_to (id_member, id_task) VALUES (3,3);
INSERT INTO assigned_to (id_member, id_task) VALUES (4,4);
INSERT INTO assigned_to (id_member, id_task) VALUES (4,5);
INSERT INTO assigned_to (id_member, id_task) VALUES (5,6);
INSERT INTO assigned_to (id_member, id_task) VALUES (7,7);
INSERT INTO assigned_to (id_member, id_task) VALUES (10,8);
INSERT INTO assigned_to (id_member, id_task) VALUES (8,9);
INSERT INTO assigned_to (id_member, id_task) VALUES (9,10);
INSERT INTO assigned_to (id_member, id_task) VALUES (16,11);
INSERT INTO assigned_to (id_member, id_task) VALUES (17,12);
INSERT INTO assigned_to (id_member, id_task) VALUES (1,13);
INSERT INTO assigned_to (id_member, id_task) VALUES (2,14);
INSERT INTO assigned_to (id_member, id_task) VALUES (3,15);
INSERT INTO assigned_to (id_member, id_task) VALUES (4,16);
INSERT INTO assigned_to (id_member, id_task) VALUES (1,17);
INSERT INTO assigned_to (id_member, id_task) VALUES (1,18);
INSERT INTO assigned_to (id_member, id_task) VALUES (1,19);
INSERT INTO assigned_to (id_member, id_task) VALUES (1,20);

```

```

-- subtask (id_subtask, id_task, brief)
INSERT INTO subtask (id_subtask, id_task, brief) VALUES (1, 3, 'Sketch');
INSERT INTO subtask (id_subtask, id_task, brief) VALUES (2, 3,

```

```
'Vectorize');
INSERT INTO subtask (id_subtask, id_task, brief) VALUES (3, 13, 'Contact
booking agencies');
INSERT INTO subtask (id_subtask, id_task, brief) VALUES (4, 13, 'Ask for
price');
INSERT INTO subtask (id_subtask, id_task, brief) VALUES (5, 13, 'Email
rider');

SELECT setval(pg_get_serial_sequence('subtask', 'id_subtask'), (SELECT
MAX(id_subtask) FROM subtask));

-- task_comment (id_task_comment, id_member, id_task, content, date)
INSERT INTO task_comment (id_task_comment, id_member, id_task, content,
date) VALUES (1, 1, 3, 'Make the w bigger. The rest is perfect. Good
work!', '2019-03-25 12:00:00');

SELECT setval(pg_get_serial_sequence('task_comment', 'id_task_comment'),
(SELECT MAX(id_task_comment) FROM task_comment));

-- forum (id_forum, id_project, topic)
INSERT INTO forum (id_forum, id_project, topic) VALUES (1, 1, 'General');
INSERT INTO forum (id_forum, id_project, topic) VALUES (2, 2, 'Team 1');
INSERT INTO forum (id_forum, id_project, topic) VALUES (3, 2, 'Team 2');
INSERT INTO forum (id_forum, id_project, topic) VALUES (4, 3, 'Team 1');
INSERT INTO forum (id_forum, id_project, topic) VALUES (5, 3, 'Team 2');
INSERT INTO forum (id_forum, id_project, topic) VALUES (6, 4, 'Team 1');
INSERT INTO forum (id_forum, id_project, topic) VALUES (7, 4, 'Team 2');

SELECT setval(pg_get_serial_sequence('forum', 'id_forum'), (SELECT
MAX(id_forum) FROM forum));

-- forum_comment (id_forum_comment, id_member, id_forum, content, date)
INSERT INTO forum_comment (id_forum_comment, id_member, id_forum, content,
date) VALUES (1, 1, 1, 'Hello everyone!', '2019-03-25 12:00:00');
INSERT INTO forum_comment (id_forum_comment, id_member, id_forum, content,
date) VALUES (2, 7, 2, 'Hello everyone!', '2019-03-25 12:00:00');
INSERT INTO forum_comment (id_forum_comment, id_member, id_forum, content,
date) VALUES (3, 7, 3, 'Hello everyone!', '2019-03-25 12:00:00');
INSERT INTO forum_comment (id_forum_comment, id_member, id_forum, content,
date) VALUES (4, 8, 4, 'Hello everyone!', '2019-03-25 12:00:00');
INSERT INTO forum_comment (id_forum_comment, id_member, id_forum, content,
date) VALUES (5, 8, 5, 'Hello everyone!', '2019-03-25 12:00:00');
INSERT INTO forum_comment (id_forum_comment, id_member, id_forum, content,
date) VALUES (6, 2, 6, 'Hello everyone!', '2019-03-25 12:00:00');
INSERT INTO forum_comment (id_forum_comment, id_member, id_forum, content,
date) VALUES (7, 2, 7, 'Hello everyone!', '2019-03-25 12:00:00');

SELECT setval(pg_get_serial_sequence('forum_comment', 'id_forum_comment'),
(SELECT MAX(id_forum_comment) FROM forum_comment));
```

```
-- notification (id_notification, id_member, content, seen, interactable, link)
INSERT INTO notification (id_notification, id_member, content, seen, interactable, link) VALUES (1, 1, 'Pedro commented on your task', true, false, '/');
INSERT INTO notification (id_notification, id_member, content, seen, interactable, link) VALUES (2, 2, 'Claudio commented on your task', true, false, '/');

SELECT setval(pg_get_serial_sequence('notification', 'id_notification'), (SELECT MAX(id_notification) FROM notification));

-- admin (id_admin, username, password)
INSERT INTO admin (id_admin, username, password) VALUES (1, 'claudio', '03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO admin (id_admin, username, password) VALUES (2, 'fernando', '03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO admin (id_admin, username, password) VALUES (3, 'joao', '03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
INSERT INTO admin (id_admin, username, password) VALUES (4, 'pedro', '03AC674216F3E15C761EE1A5E255F067953623C8B388B4459E13F978D7C846F4');
```

Revision history

Changes made to the first submission:

GROUP1863, 10/04/2019

- Cláudio Fischer Lemos, up201603542@fe.up.pt
- Fernando Jorge Coelho Barreira Calheiros Alves, up201605270@fe.up.pt (Editor)
- João Carlos Cardoso Maduro, up201605219@fe.up.pt
- Pedro Casais da Silva e Sousa Gonçalves, up201604643@fe.up.pt