

A6 – Integrity constraints. Indexes, triggers, user functions and database populated with data

This artefact contains the physical schema of the database necessary for the Project Management website. It contains the characterisation of the indexes, the support of data integrity rules with triggers and the definition of the database user-defined functions. In addition, the code necessary for the above-mentioned indexes, queries and triggers, and all the inserts made.

1. Database workload

1.1. Estimate of tuples

Relation reference	Relation name	Order of magnitude	Estimated growth
R01	User	Dozens	Units per week
R02	Project	Hundreds	Dozens per month
R03	Board	Hundreds	Dozens per month
R04	Task	Thousands	Dozens per week
R05	Meeting	Hundreds	Units per week
R06	Personal_event	Hundreds	Units per week
R07	Comment	Dozens	Units per week
R08	File	Dozens	Units per week
R09	Notification	Thousands	Units per day
R10	Progress_update	Thousands	Units per day
R11	Message	Hundreds	Units per week
R12	Project_team	Hundreds	Dozens per month
R13	Board_team	Hundreds	Dozens per month
R14	Profile_picture	Dozens	Units per week No growth
R15	Project_picture	Hundreds	Dozens per month No growth
R16	Contact	Dozens	Units per month
R17	Ban_user	Units	Units per year
R18	Archive_project	Dozens	Units per month

1.2. Most frequent queries

Query reference	SELECT01
Query description	Check if e-mail exists All emails
SELECT e mail FROM "User" WHERE user.e_mail = \$userEmail SELECT e_mail FROM "User"	

Query reference	SELECT02
Query description	Email's password
SELECT password FROM "User" WHERE user.e_mail = \$userEmail	

Query reference	SELECT03
Query description	All public projects
SELECT name FROM Project WHERE Project.privacy=1	

Query reference	SELECT04
Query description	All user's projects
SELECT * FROM Project INNER JOIN Project_team ON Project.id=Project_team.id_project WHERE Project.id_coordinator=\$user OR Project_team.id_user=\$user	

Query reference	SELECT05
Query description	All user's personal events
SELECT * FROM Personal_event WHERE Personal_event.id_user=\$user	

Query reference	SELECT06
Query description	All users board's meetings
SELECT * FROM Meeting INNER JOIN Board_team ON Meeting.id_board=Board_team.id_board WHERE Board_team.id_worker=\$user	

Query reference	SELECT07
Query description	User's personal info
SELECT * FROM "User" WHERE User.id=\$user	

Query reference	SELECT08
Query description	User's notifications
SELECT * FROM Notification WHERE Notification.id_user=\$user	

Query reference	SELECT09
Query description	User's number of unread notifications
SELECT COUNT id FROM Notification WHERE Notification.id_user=\$user AND Notification.read=0	

Query reference	SELECT10
Query description	Project's boards
SELECT * FROM Board WHERE Board.id_project=\$projectId	

Query reference	SELECT11
Query description	Project's info
SELECT * FROM Project WHERE Project.id=\$projectId	

Query reference	SELECT12
Query description	Project worker's info
SELECT User.full_name, Profile_picture.path, User.e_mail FROM "User" INNER JOIN Profile_picture ON Profile_picture.id_user=\$user WHERE Project.id=\$projectId AND Profile_picture.path=SELECT(MAX(Profile_picture.id))	

Query reference	SELECT13
Query description	Project's events
SELECT * FROM Meeting INNER JOIN Board ON Board.id=Meeting.id_board WHERE Board.id_project=\$projectId	

Query reference	SELECT14
Query description	Last week project's messages
SELECT * FROM Message WHERE Message.id_project=\$projectId AND Message.date >= DATEADD(day, -7, GETDATE())	

Query reference	SELECT15
Query description	Board's tasks
SELECT Task.id FROM Task WHERE Task.id_board=\$boardId	

Query reference	SELECT16
Query description	Task's info
SELECT * FROM Task WHERE Task.id=\$taskId	

Query reference	SELECT15
Query description	Last 5 task's updates
SELECT TOP 5 Comment.id, File.id, Progress_update.id FROM Task INNER JOIN Comment ON Task.id=Comment.id_task INNER JOIN File ON Task.id=File.id_task INNER JOIN Task.id=Progress_update.id_task WHERE Task.id=\$taskId	

Query reference	SELECT16
Query description	Search project
SELECT description, start_date, name FROM Project WHERE description LIKE '%\$search%' OR title LIKE '%\$search%' AND privacy=false; ORDER BY title;	

Query reference	SELECT17
-----------------	----------

Query description	Search user
<pre>SELECT username, full_name, e_mail FROM User WHERE username LIKE '%\$search%' OR full_name LIKE '%\$search%'; ORDER BY username;</pre>	

1.3. Most frequent modifications

Query reference	UPDATE01
Query description	Insert new progress update
<pre>INSERT INTO Progress_update (new_value, id_user, id_task) VALUES (\$new_value, \$id_user, \$id_task)</pre>	

Query reference	UPDATE02
Query description	Update task progress
<pre>UPDATE Task SET progress = \$progress</pre>	

Query reference	UPDATE03
Query description	Upload new file
<pre>INSERT INTO File (path, description, id_user, id_task) VALUES (\$path, \$description, \$id_user, \$id_task)</pre>	

Query reference	UPDATE04
Query description	Comment task
<pre>INSERT INTO Comment (comment, id_user, id_task) VALUES (\$comment, \$id_user, \$id_task)</pre>	

Query reference	UPDATE05
Query description	Add personal event
<pre>INSERT INTO Personal_event (date, place, name, id_user) VALUES (\$date, \$place, \$name, \$id_user)</pre>	

Query reference	UPDATE06
Query description	Add board meeting
<pre>INSERT INTO Meeting (date, name, place, id_board) VALUES (\$date, \$name, \$place, \$id_board)</pre>	

Query reference	UPDATE07
Query description	Add new project
<pre>INSERT INTO Project (description, name, id_coordinator, privacy) VALUES (\$description, \$name, \$id_coordinator, \$privacy)</pre>	

Query reference	UPDATE08
Query description	Add new board to existing project
<pre>INSERT INTO Board (description, name, id_creator, id_project) VALUES (\$description, \$name, \$id_creator, \$id_project)</pre>	

Query reference	UPDATE09
Query description	Archive or close task
UPDATE Task SET task_state=\$task_state	
Query reference	UPDATE10
Query description	Archive or close project
UPDATE Project SET project_state=\$project_state	

Query reference	UPDATE11
Query description	Add new worker to project
INSERT INTO Project_team (id_user, id_project) VALUES (\$id_user, \$id_project)	

Query reference	UPDATE12
Query description	Add new worker to board
INSERT INTO Board_team (id_board, id_user) VALUES (\$id_board, \$id_user)	

Query reference	UPDATE13
Query description	Add new user
INSERT INTO User (e_mail, password, URL, username) VALUES (\$e_mail, \$password, \$URL, \$username)	

Query reference	UPDATE14
Query description	Update user's full name
Query frequency	
UPDATE User SET full_name=\$full_name	

Query reference	UPDATE15
Query description	Change password
UPDATE User SET password=\$password	

Query reference	UPDATE16
Query description	Set user as administrator
UPDATE User SET administrator=\$administrator	

Query reference	UPDATE17
Query description	Change profile picture
Query frequency	
INSERT INTO Profile_picture (id_user, path) VALUES (\$id_user, \$path)	

Query reference	UPDATE18
Query description	Change project description

UPDATE Project
SET description=\$description

Query reference	UPDATE19
Query description	Add project end date
UPDATE Project	
SET end_date=\$end_date	
Query reference	UPDATE20
Query description	Change project picture
INSERT INTO Project_picture (id_project, path)	
VALUES (\$id_project, \$path)	

Query reference	UPDATE21
Query description	Add user to contacts
INSERT INTO Contact (id_user, id_contact)	
VALUES (\$id_user, \$id_contact)	

Query reference	UPDATE22
Query description	Send messages to project forum
INSERT INTO Message (message, id_user, id_project)	
VALUES (\$message, \$id_user, \$id_project)	

Query reference	UPDATE23
Query description	Create new notification
INSERT INTO Notification (id_user, notification)	
VALUES (\$id_user, \$notification)	

Query reference	UPDATE24
Query description	Mark notification as read or unread
Query frequency	
UPDATE Notification	
SET read=\$read	

2. Proposed Indexes

2.1. Performance indexes

Index reference	INDEX01
Related queries	SELECT07
Index relation	User
Index attribute	e_mail
Index type	Hash
Cardinality	High
Clustering	No
Justification	Query needs to be executed many times and has to be fast; cardinality is high because the

	e_mail is a unique key; not a good candidate for clustering.
<pre>CREATE INDEX email_user ON User USING hash(e_mail)</pre>	

Index reference	INDEX02
Related queries	SELECT03
Index relation	Project
Index attribute	id
Index type	Hash
Cardinality	High
Clustering	No
Justification	Query needs to be executed many times and has to be fast; cardinality is high because the id is a unique key; not a good candidate for clustering.
<pre>CREATE INDEX id_project ON Project USING hash(id)</pre>	

Index reference	INDEX03
Related queries	SELECT04
Index relation	Project_team
Index attribute	id_user
Index type	Hash
Cardinality	Medium
Clustering	Yes
Justification	Query needs to be executed many times and table is very large, so it has to be fast; cardinality is medium so it's a good candidate for clustering.
<pre>CREATE INDEX user_project ON Project_team USING hash(id_user)</pre>	

Index reference	INDEX04
Related queries	SELECT10
Index relation	Board
Index attribute	id_project
Index type	Hash
Cardinality	Medium
Clustering	Yes
Justification	Query needs to be executed many times and has to be fast because it is executed many times; cardinality is medium so it's a good candidate for clustering.
<pre>CREATE INDEX project_boards ON Board USING hash(id_project)</pre>	

Index reference	INDEX05
Related queries	SELECT14
Index relation	Message
Index attribute	date, project
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	Query needs to be executed many times and table is very large; cardinality is medium so it's a good candidate for clustering.
CREATE INDEX message_date ON Message USING btree(date, id_project)	

2.2. Full-text Search indexes

Index reference	INDEX06
Related queries	SELECT16
Index relation	Project
Index attribute	title
Index type	GiST
Clustering	No
Justification	To improve the performance of full text searches while searching for projects by their titles; GiST because it's better for dynamic data.
CREATE INDEX search_project ON Project USING GIST(title)	

Index reference	INDEX07
Related queries	SELECT17
Index relation	User
Index attribute	username
Index type	B-Tree GiST
Clustering	No
Justification	To improve the performance of full text searches while searching for users by their username; GiST because it's better for dynamic data B-Tree because it is of the form col LIKE '\$search\$%'. it's a good candidate for clustering.
CREATE INDEX search_user ON User USING btree GiST(username)	

3. Triggers

Trigger reference	TRIGGER01
Trigger description	An e-mail can only be registered once
<pre>CREATE FUNCTION check_exist_email() RETURNS TRIGGER AS \$BODY\$ BEGIN IF EXISTS (SELECT e_mail FROM "User" WHERE NEW.e_mail = e_mail) THEN RAISE EXCEPTION 'The e-mail you are trying to use is already registred' END IF; RETURN NEW; END \$BODY\$ LANGUAGE plpgsql; CREATE TRIGGER check_exist_email BEFORE INSERT ON "User" FOR EACH ROW EXECUTE PROCEDURE check_exist_user();</pre>	

Trigger reference	TRIGGER02
Trigger description	An username can only be registered once
<pre>CREATE FUNCTION check_exist_username() RETURNS TRIGGER AS \$BODY\$ BEGIN IF EXISTS (SELECT username FROM "User" WHERE NEW.username = username) THEN RAISE EXCEPTION 'The username you are trying to use is already in use' END IF; RETURN NEW; END \$BODY\$ LANGUAGE plpgsql; CREATE TRIGGER check_exist_username BEFORE INSERT ON "User"</pre>	

```

FOR EACH ROW
EXECUTE PROCEDURE check_exist_username();

```

Trigger reference	TRIGGER03
Trigger description	Two boards of the same project can't have the same name
<pre> CREATE FUNCTION check_exist_board() RETURNS TRIGGER AS \$BODY\$ BEGIN IF EXISTS (SELECT name FROM Board WHERE Board.id_project=\$projectId AND NEW.name=name) THEN RAISE EXCEPTION 'This project already has a board with that name' END IF; RETURN NEW; END \$BODY\$ LANGUAGE plpgsql; CREATE TRIGGER check_exist_board BEFORE INSERT ON Board FOR EACH ROW EXECUTE PROCEDURE check_exist_board(); </pre>	

Trigger reference	TRIGGER04
Trigger description	Can't add user to team if he already belongs to it
<pre> CREATE FUNCTION check_exist_worker() RETURNS TRIGGER AS \$BODY\$ BEGIN IF EXISTS (SELECT name FROM Project_team WHERE Project_team.id_project=\$projectId AND NEW.id_worker=id_worker) THEN RAISE EXCEPTION 'This user already belongs to this team' END IF; RETURN NEW; END \$BODY\$ </pre>	

```
LANGUAGE plpgsql;
```

```
CREATE TRIGGER check_exist_worker  
    BEFORE INSERT ON Project_team  
    FOR EACH ROW  
        EXECUTE PROCEDURE check_exist_worker();
```

Trigger reference	TRIGGER05
Trigger description	Can't add user to contact list if he already belongs to it
<pre>CREATE FUNCTION check_exist_contact() RETURNS TRIGGER AS \$BODY\$ BEGIN IF EXISTS (SELECT name FROM Contact WHERE Contact.id_user=\$userId AND NEW.id_contact=id_contact AND NEW.id_contact=\$userId) THEN RAISE EXCEPTION 'This user already is on our contact list' END IF; RETURN NEW; END \$BODY\$ LANGUAGE plpgsql; CREATE TRIGGER check_exist_contact BEFORE INSERT ON Contact FOR EACH ROW EXECUTE PROCEDURE check_exist_contact();</pre>	

4. SQL Code

Database creation

```
--
```

```
-- Domains
```

```
--
```

```
CREATE DOMAIN state AS TEXT
```

```
CHECK(
```

```
____ VALUE ~ 'In progress'  
____ OR VALUE ~ 'Closed'  
____ OR VALUE ~ 'Archived'  
____ );
```

```
--  
-- Tables  
--
```

```
CREATE TABLE "Board" (  
____ id SERIAL NOT NULL,  
____ description text,  
____ name text NOT NULL,  
____ id creator integer NOT NULL,  
____ id project integer NOT NULL  
____ );
```

```
CREATE TABLE "Board team" (  
____ id board integer NOT NULL,  
____ id user integer NOT NULL  
____ );
```

```
CREATE TABLE "Comment" (  
____ id SERIAL NOT NULL,  
____ comment text NOT NULL,  
____ date timestamp without time zone DEFAULT now() NOT NULL,  
____ id user integer NOT NULL,  
____ id task integer NOT NULL  
____ );
```

```
CREATE TABLE "Contact" (  
____ id user integer NOT NULL,  
____ id contact integer NOT NULL  
____ );
```

```
CREATE TABLE "File" (  
____ id SERIAL NOT NULL,  
____ path text NOT NULL,  
____ publish date timestamp without time zone DEFAULT now() NOT NULL,  
____ description text,
```

```
____ id user integer NOT NULL,  
____ id task integer NOT NULL  
);
```

```
CREATE TABLE "Meeting" (  
____ id SERIAL NOT NULL,  
____ date timestamp without time zone NOT NULL,  
____ name text,  
____ place text,  
____ id board integer NOT NULL,  
____ CONSTRAINT "CK1" CHECK ((date > now()))  
);
```

```
CREATE TABLE "Message" (  
____ id SERIAL NOT NULL,  
____ message text NOT NULL,  
____ date timestamp without time zone DEFAULT now() NOT NULL,  
____ id user integer NOT NULL,  
____ id project integer NOT NULL  
);
```

```
CREATE TABLE "Notification" (  
____ id SERIAL NOT NULL,  
____ id user integer NOT NULL,  
____ date timestamp without time zone DEFAULT now() NOT NULL,  
____ notification text NOT NULL,  
____ read boolean DEFAULT false NOT NULL  
);
```

```
CREATE TABLE "Personalevent" (  
____ id SERIAL NOT NULL,  
____ date timestamp without time zone NOT NULL,  
____ place text,  
____ name text,  
____ id user integer NOT NULL,  
____ CONSTRAINT "CK1" CHECK ((date > now()))  
);
```

```
CREATE TABLE "Profile picture" (  
____ id SERIAL NOT NULL,  
____ id user integer NOT NULL,
```

```
____ path text NOT NULL  
);
```

```
CREATE TABLE "Progress update" (  
____ id SERIAL NOT NULL,  
____ date timestamp without time zone DEFAULT now() NOT NULL,  
____ new value integer NOT NULL,  
____ id user integer NOT NULL,  
____ id task integer NOT NULL,  
____ CONSTRAINT "CK1" CHECK ((new value > 0)),  
____ CONSTRAINT "CK2" CHECK ((new value <= 100))  
);
```

```
CREATE TABLE "Project" (  
____ id SERIAL NOT NULL,  
____ description text,  
____ start date timestamp without time zone DEFAULT now() NOT NULL,  
____ end date date,  
____ name text NOT NULL,  
____ id coordinator integer NOT NULL,  
____ project state state DEFAULT 'In progress' NOT NULL,  
____ privacy boolean DEFAULT false NOT NULL,  
____ user archived integer,  
____ CONSTRAINT "CK1" CHECK ((end date > start date))  
);
```

```
CREATE TABLE "Project picture" (  
____ id SERIAL NOT NULL,  
____ id project integer NOT NULL,  
____ path text NOT NULL  
);
```

```
CREATE TABLE "Project team" (  
____ id user integer NOT NULL,  
____ id project integer NOT NULL  
);
```

```
CREATE TABLE "Task" (  
____ id SERIAL NOT NULL,  
____ budget money,  
____ deadline date NOT NULL,
```

```
description text,  
name text NOT NULL,  
progress integer DEFAULT 0 NOT NULL,  
task state state DEFAULT 'In progress' NOT NULL,  
id creator integer NOT NULL,  
id board integer NOT NULL,  
CONSTRAINT "CK1" CHECK ((progress > 0)),  
CONSTRAINT "CK2" CHECK ((progress <= 100))  
);
```

```
CREATE TABLE "User" (  
id SERIAL NOT NULL,  
"e mail" text NOT NULL,  
full name text,  
password text NOT NULL,  
"url" text NOT NULL,  
username text NOT NULL,  
administrator boolean DEFAULT false NOT NULL,  
user ban integer  
);
```

```
--  
-- Primary Keys and Uniques  
--
```

```
ALTER TABLE ONLY "Board"  
ADD CONSTRAINT "Board_pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "Board team"  
ADD CONSTRAINT "Board team_pkey" PRIMARY KEY (id board, id user);
```

```
ALTER TABLE ONLY "Comment"  
ADD CONSTRAINT "Comment_pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "Contact"  
ADD CONSTRAINT "Contact_pkey" PRIMARY KEY (id user, id contact);
```

```
ALTER TABLE ONLY "File"  
ADD CONSTRAINT "File_path key" UNIQUE (path);
```

```
ALTER TABLE ONLY "File"  
ADD CONSTRAINT "File_pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "Meeting"  
ADD CONSTRAINT "Meeting_pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "Message"  
ADD CONSTRAINT "Message_pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "Notification"  
ADD CONSTRAINT "Notification_pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "Personal event"  
ADD CONSTRAINT "Personal event_pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "Profile picture"  
ADD CONSTRAINT "Profile picture path key" UNIQUE (path);
```

```
ALTER TABLE ONLY "Profile picture"  
ADD CONSTRAINT "Profile picture_pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "Progress update"  
ADD CONSTRAINT "Progress update_pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "Project picture"  
ADD CONSTRAINT "Project picture path key" UNIQUE (path);
```

```
ALTER TABLE ONLY "Project picture"  
ADD CONSTRAINT "Project picture_pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "Project"  
ADD CONSTRAINT "Project_pkey" PRIMARY KEY (id);
```



```
ALTER TABLE ONLY "Project team"  
ADD CONSTRAINT "Project team pkey" PRIMARY KEY (id user, id project);
```

```
ALTER TABLE ONLY "Task"  
ADD CONSTRAINT "Task pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "User"  
ADD CONSTRAINT "User url key" UNIQUE ("url");
```

```
ALTER TABLE ONLY "User"  
ADD CONSTRAINT "User e mail key" UNIQUE ("e mail");
```

```
ALTER TABLE ONLY "User"  
ADD CONSTRAINT "User pkey" PRIMARY KEY (id);
```

```
ALTER TABLE ONLY "User"  
ADD CONSTRAINT "User username key" UNIQUE (username);
```

```
--  
-- Foreign Keys  
--
```

```
ALTER TABLE ONLY "Board"  
ADD CONSTRAINT "Board id creator fkey" FOREIGN KEY (id creator)  
REFERENCES "User"(id);
```

```
ALTER TABLE ONLY "Board"  
ADD CONSTRAINT "Board id project fkey" FOREIGN KEY (id project)  
REFERENCES "Project"(id);
```

```
ALTER TABLE ONLY "Board team"  
ADD CONSTRAINT "Board team id board fkey" FOREIGN KEY (id board)  
REFERENCES "Board"(id);
```

```
ALTER TABLE ONLY "Board team"
```

```
ADD CONSTRAINT "Board team id user fkey" FOREIGN KEY (id user)  
REFERENCES "User"(id);
```

```
ALTER TABLE ONLY "Comment"  
ADD CONSTRAINT "Comment id task fkey" FOREIGN KEY (id task) REFERENCES  
"Task"(id);
```

```
ALTER TABLE ONLY "Comment"  
ADD CONSTRAINT "Comment id user fkey" FOREIGN KEY (id user) REFERENCES  
"User"(id);
```

```
ALTER TABLE ONLY "Contact"  
ADD CONSTRAINT "Contact id contact fkey" FOREIGN KEY (id contact)  
REFERENCES "User"(id);
```

```
ALTER TABLE ONLY "Contact"  
ADD CONSTRAINT "Contact id user fkey" FOREIGN KEY (id user) REFERENCES  
"User"(id);
```

```
ALTER TABLE ONLY "File"  
ADD CONSTRAINT "File id task fkey" FOREIGN KEY (id task) REFERENCES  
"Task"(id);
```

```
ALTER TABLE ONLY "File"  
ADD CONSTRAINT "File id user fkey" FOREIGN KEY (id user) REFERENCES  
"User"(id);
```

```
ALTER TABLE ONLY "Meeting"  
ADD CONSTRAINT "Meeting id board fkey" FOREIGN KEY (id board)  
REFERENCES "Board"(id);
```

```
ALTER TABLE ONLY "Message"  
ADD CONSTRAINT "Message id project fkey" FOREIGN KEY (id project)  
REFERENCES "Project"(id);
```

```
ALTER TABLE ONLY "Message"
```

ADD CONSTRAINT "Message id user fkey" FOREIGN KEY (id user) REFERENCES "User"(id);

ALTER TABLE ONLY "Notification"

ADD CONSTRAINT "Notification id user fkey" FOREIGN KEY (id user) REFERENCES "User"(id);

ALTER TABLE ONLY "Personal event"

ADD CONSTRAINT "Personal event id user fkey" FOREIGN KEY (id user) REFERENCES "User"(id);

ALTER TABLE ONLY "Profile picture"

ADD CONSTRAINT "Profile picture id user fkey" FOREIGN KEY (id user) REFERENCES "User"(id);

ALTER TABLE ONLY "Progress update"

ADD CONSTRAINT "Progress update id task fkey" FOREIGN KEY (id task) REFERENCES "Task"(id);

ALTER TABLE ONLY "Progress update"

ADD CONSTRAINT "Progress update id user fkey" FOREIGN KEY (id user) REFERENCES "User"(id);

ALTER TABLE ONLY "Project"

ADD CONSTRAINT "Project id coordinator fkey" FOREIGN KEY (id coordinator) REFERENCES "User"(id);

ALTER TABLE ONLY "Project"

ADD CONSTRAINT "Project user archived fkey" FOREIGN KEY (user archived) REFERENCES "User"(id);

ALTER TABLE ONLY "Project picture"

ADD CONSTRAINT "Project picture id project fkey" FOREIGN KEY (id project) REFERENCES "Project"(id);

ALTER TABLE ONLY "Project team"

```
ADD CONSTRAINT "Project team id project fkey" FOREIGN KEY (id project)  
REFERENCES "Project"(id);
```

```
ALTER TABLE ONLY "Project team"
```

```
ADD CONSTRAINT "Project team id user fkey" FOREIGN KEY (id user)  
REFERENCES "User"(id);
```

```
ALTER TABLE ONLY "Task"
```

```
ADD CONSTRAINT "Task id board fkey" FOREIGN KEY (id board) REFERENCES  
"Board"(id);
```

```
ALTER TABLE ONLY "Task"
```

```
ADD CONSTRAINT "Task id creator fkey" FOREIGN KEY (id creator)  
REFERENCES "User"(id);
```

```
ALTER TABLE ONLY "User"
```

```
ADD CONSTRAINT "User user ban fkey" FOREIGN KEY (user ban) REFERENCES  
"User"(id);
```

4- Database population

```
---  
USE  
RS  
---
```

```
-----
```

```
INSERT INTO "User" (e_mail, full_name, password, url, username,  
administrator) VALUES ('up201405655@fe.up.pt', 'Teresa Valério',  
'teresaMaya19', 444, 'TeresaValerio', true)  
INSERT INTO "User" (e_mail, full_name, password, url, username,  
administrator) VALUES ('up201402784@fe.up.pt', 'Inês Gonçalves',  
'inesggtunafe8', 123, 'InesGoncalves', true)  
INSERT INTO "User" (e_mail, full_name, password, url, username,  
administrator) VALUES ('up201405085@fe.up.pt', 'Sara Gomes', 'sara23bolota',  
635, 'SaraGomes', true)  
INSERT INTO "User" (e_mail, full_name, password, url, username,  
administrator) VALUES ('joana.monteiro@hotmail.com', 'Joana Monteiro',  
'QWERTY', 883, 'JoanaMonteiro45', false)
```

```

INSERT INTO "User" (e_mail, full_name, password, url, username) VALUES
('maria.costa12@hotmail.com', 'Maria João Costa', 'palavrapasse123',
713, 'MJCosta14')
INSERT INTO "User" (e_mail, full_name, password, url, username) VALUES
('joaoo_santos@hotmail.com', 'João Costa Santos', 'SG234StbtQ',
790, 'JoaoCS18')
INSERT INTO "User" (e_mail, full_name, password, url, username) VALUES
('luis134castro@hotmail.com', 'Luís Castro', 'BB13fsRg6', 253, 'LuisCastro')
INSERT INTO "User" (e_mail, full_name, password, url, username) VALUES
('up201404541@fe.up.pt', 'Francisca Seabra', 'BETHDDD', 662, 'FSeabra')
INSERT INTO "User" (e_mail, full_name, password, url, username,
administrator) VALUES ('lacinia.orci.consectetuer@velarcu.ca', 'Noelle
Moran', 'PMR97BRV5EC', 621, 'NoelleMoran44', false)
INSERT INTO "User" (e_mail, full_name, password, url, username,
administrator) VALUES ('ipsum.nunc.id@scelerisquemollis.edu', 'MacKenzie
Garrison', 'YUY23IU09RC', 412, 'MKenzieG', false)
INSERT INTO "User" (e_mail, full_name, password, url, username) VALUES
('urna.Vivamus.molestie@Fusce.net', 'Isadora Hewitt', 'BUE71FOD50I',
914, 'HIsadora23')
INSERT INTO "User" (e_mail, full_name, password, url, username,
administrator) VALUES ('tincidunt@tinciduntduiaugue.com', 'Lacota Page',
'GPB35FPL2TB', 515, 'LacotaPagee', false)
INSERT INTO "User" (e_mail, full_name, password, url, username,
administrator) VALUES ('ac.risus.Morbi@mauris.co.uk', 'Priscilla Silva',
'UPK14PEJ2DZ', 328, 'SilvaP47', false)
INSERT INTO "User" (e_mail, full_name, password, url, username,
administrator) VALUES ('eu@tellus.net', 'Lynn Burch', 'IGU81EUU7UB',
555, 'lynn29', false)
INSERT INTO "User" (e_mail, full_name, password, url, username) VALUES
('nascetur.ridiculus@nonmassa.net', 'Jocelyn Fulton', 'IEV88RCI7LT',
181, 'Fulton')
INSERT INTO "User" (e_mail, full_name, password, url, username,
administrator) VALUES ('sem@consequatauctornunc.co.uk', 'Tashya Deleon',
'LZJ76NCB9QP', 741, 'deleonTashya', false)
INSERT INTO "User" (e_mail, full_name, password, url, username,
administrator) VALUES ('et.euismod.et@pharetraNamac.net', 'Lillian Snow',
'MYX34BPA2RK', 301, 'lillian', false)
INSERT INTO "User" (e_mail, full_name, password, url, username) VALUES
('cursus.Nunc@vitaeeratVivamus.edu', 'Margaret Mack', 'NRZ16ICN4NL',
215, 'Margarety')
INSERT INTO "User" (e_mail, full_name, password, url, username) VALUES
('vel.faucibus.id@lacinia.edu', 'Martha Cain', 'OGM48BTU8UO', 590,
'MarthaCain')
INSERT INTO "User" (e_mail, full_name, password, url, username) VALUES
('fi1996@gmail.com', 'Filipa Soares', 'GLsn36d', 506, 'FiSoares45')

```

--- PROJECTS ---

```
INSERT INTO "Project" (description, start_date, end_date, name,
id_coordinator, project_state, privacy) VALUES ('Projeto de LIEB para
detecção de apneia do sono', TIMESTAMP '2018/03/04', timestamp '2019/05/18',
'Hypnos', 1, 'In_progress', true)
INSERT INTO "Project" (start_date, end_date, name, id_coordinator,
project_state) VALUES (TIMESTAMP '2018/02/20', TIMESTAMP '2019/05/25
24:00:00', 'Fotopletismógrafo portátil', 1, 'In_progress')
INSERT INTO "Project" (start_date, end_date, name, id_coordinator) VALUES
(TIMESTAMP '2018/03/1', TIMESTAMP '2019/06/10 9:00:00', 'Estudo da doença de
Alzheimer através de análise de imagens do cérebro', 3)
INSERT INTO "Project" (description, start_date, end_date, name,
id_coordinator, project_state, privacy) VALUES ('Detecção de emoções através
da análise de sinais de EEG', TIMESTAMP '2018/03/14', timestamp
'2019/09/18', 'EEG feelings', 4, 'In_progress', true)
INSERT INTO "Project" (start_date, end_date, name, id_coordinator,
project_state, privacy) VALUES (TIMESTAMP '2018/02/15', TIMESTAMP
'2019/05/30', 'Projeto MINV', 20, 'In_progress', false)
```

--- PROJECT'S TEAMS ---

```
INSERT INTO "Project_team" (id_user, id_project) VALUES (3,1)
INSERT INTO "Project_team" (id_user, id_project) VALUES (8,1)
INSERT INTO "Project_team" (id_user, id_project) VALUES (3,2)
INSERT INTO "Project_team" (id_user, id_project) VALUES (16,3)
INSERT INTO "Project_team" (id_user, id_project) VALUES (5,4)
INSERT INTO "Project_team" (id_user, id_project) VALUES (6,4)
INSERT INTO "Project_team" (id_user, id_project) VALUES (7,4)
INSERT INTO "Project_team" (id_user, id_project) VALUES (8,4)
INSERT INTO "Project_team" (id_user, id_project) VALUES (9,4)
INSERT INTO "Project_team" (id_user, id_project) VALUES (10,4)
INSERT INTO "Project_team" (id_user, id_project) VALUES (11,4)
INSERT INTO "Project_team" (id_user, id_project) VALUES (1,5)
INSERT INTO "Project_team" (id_user, id_project) VALUES (3,5)
```

--- PROJECT'S BOARDS ---

```
INSERT INTO "Board" (description, name, id_creator, id_project) VALUES
('Pesquisa de fundamentos teóricos que servirão de base à implementação de
código', 'Pesquisa', 1, 1)
INSERT INTO "Board" (description, name, id_creator, id_project) VALUES
('Pesquisa sobre funcionamento da fotopletismografia e métodos a adotar',
'Pesquisa', 3, 2)
INSERT INTO "Board" (name, id_creator, id_project) VALUES ('Construção
circuito', 3, 2)
INSERT INTO "Board" (description, name, id_creator, id_project) VALUES
('Pesquisa de possíveis técnicas a adotar', 'Pesquisa', 3, 3)
INSERT INTO "Board" (description, name, id_creator, id_project) VALUES
('Procurar artigos sobre projetos semelhantes', 'Artigos', 7, 4)
INSERT INTO "Board" (description, name, id_creator, id_project) VALUES
('Pesquisar acerca do EEC e possíveis maneiras de o analisar', 'Pesquisa
EEG', 9, 4)
INSERT INTO "Board" (description, name, id_creator, id_project) VALUES
('Pesquisa de artigos relacionados com o tema', 'Pesquisa bibliográfica', 1,
5)
INSERT INTO "Board" (description, name, id_creator, id_project) VALUES
('Elaboração do questionário sob a forma de Google Form', 'Questionário',
20, 5)
INSERT INTO "Board" (description, name, id_creator, id_project) VALUES
('Elaboração do guião da entrevista', 'Entrevista', 3, 5)
```

--- BOARD'S TASKS ---

```
INSERT INTO "Task" (deadline, description, name, progress, task_state,
id_creator, id_board) VALUES(TIMESTAMP '2019/05/1', 'Pesquisa em várias
fontes bibliográficas relativamente ao que causa a apneia e como poderá ser
detetada' , 'Pesquisa sobre apneia', 10, 'In_progress', 1, 1)
INSERT INTO "Task" (deadline, description, name, progress, task_state,
id_creator, id_board) VALUES(TIMESTAMP '2019/05/10', 'Pesquisa sobre
fotopletismografia (como funciona, que método utilizar, etc)' , 'Pesquisa
fotopletismografia', 40, 'In_progress', 3, 2)
```

```

INSERT INTO "Task" (deadline, description, name, progress, task_state,
id_creator, id_board) VALUES(TIMESTAMP '2019/05/15', 'Pesquisar melhor
sensor a utilizar e possíveis circuitos a implementar' , 'Pesquisa
circuito', 30, 'In_progress', 3, 2)
INSERT INTO "Task" (deadline, description, name, progress, task_state,
id_creator, id_board) VALUES(TIMESTAMP '2019/05/10', 'Pesquisa sobre o que é
a doença de Alzheimer e que efeitos tem no cérebro humano' , 'Pesquisa sobre
Alzheimer', 30, 'In_progress', 3, 3)
INSERT INTO "Task" (deadline, description, name, progress, task_state,
id_creator, id_board) VALUES(TIMESTAMP '2019/05/20', 'Pesquisa sobre
possíveis técnicas de análise de imagem a adotar e base de dados a utilizar'
, 'Técnicas de análise de imagem', 20, 'In_progress', 3, 3)
INSERT INTO "Task" (deadline, description, name, progress, task_state,
id_creator, id_board) VALUES(TIMESTAMP '2019/06/24', 'O que é o EEG, como é
obtido e que informação nos dá?', 'Fundamentos EEG', 60, 'In_progress', 8,
4)
INSERT INTO "Task" (deadline, description, name, progress, task_state,
id_creator, id_board) VALUES(TIMESTAMP '2019/06/30', 'Pesquisa de
processamentos a adotar para analisar o sinal EEG', 'Processamento EEG', 10,
'In_progress', 6, 4)
INSERT INTO "Task" (deadline, description, name, progress, task_state,
id_creator, id_board) VALUES(TIMESTAMP '2019/04/22', 'Pesquisa sobre a
estrutura que um questionário deve ter (quantidade/tipo de perguntas, etc)',
'Estrutura questionário', 40, 'In_progress', 1, 8)
INSERT INTO "Task" (deadline, name, progress, task_state, id_creator,
id_board) VALUES(TIMESTAMP '2019/04/30', 'Elaboração questionário', 10,
'In_progress', 20, 8)
INSERT INTO "Task" (deadline, name, progress, task_state, id_creator,
id_board) VALUES(TIMESTAMP '2019/04/25', 'Como fazer uma entrevista?', 50,
'In_progress', 3, 9)
INSERT INTO "Task" (deadline, name, progress, task_state, id_creator,
id_board) VALUES(TIMESTAMP '2019/04/27', 'Guião da entrevista', 10,
'In_progress', 1, 9)
INSERT INTO "Task" (deadline, name, progress, task_state, id_creator,
id_board) VALUES(TIMESTAMP '2019/05/10', 'Realização das entrevista', 10,
'In_progress', 20, 9)

```

--- BOARD'S TEAMS ---

```

INSERT INTO "Board_team" VALUES (3,1)
INSERT INTO "Board_team" VALUES (2,1)

```



```

INSERT INTO "Board_team" VALUES (4,16)
INSERT INTO "Board_team" VALUES (5,5)
INSERT INTO "Board_team" VALUES (5,1)
INSERT INTO "Board_team" VALUES (5,8)
INSERT INTO "Board_team" VALUES (5,11)
INSERT INTO "Board_team" VALUES (6,6)
INSERT INTO "Board_team" VALUES (6,1)
INSERT INTO "Board_team" VALUES (6,8)
INSERT INTO "Board_team" VALUES (6,9)
INSERT INTO "Board_team" VALUES (6,10)
INSERT INTO "Board_team" VALUES (6,11)

```

```

--- COMMENTS ---
-----

```

```

INSERT INTO "Comment" (comment, date, id_user, id_task) VALUES ('SFH7050 já
não pode ser utilizado!', TIMESTAMP '2018/03/25 12:50:04', 1, 3)
INSERT INTO "Comment" (comment, date, id_user, id_task) VALUES ('Não está a
ser possível fazer o download da base de dados', TIMESTAMP '2018/03/19
19:24:34', 3, 5)
INSERT INTO "Comment" (comment, date, id_user, id_task) VALUES ('Ver
apontamentos da aula prática!', TIMESTAMP '2018/03/27 17:58:02', 20, 8)

```

```

--- MEETINGS ---
-----

```

```

INSERT INTO "Meeting" (date, place, id_board) VALUES (TIMESTAMP '2018/04/10
11:00:00', 'Departamento de Bioengenharia', 1)
INSERT INTO "Meeting" (date, place, id_board) VALUES (TIMESTAMP '2018/04/17
09:30:00', 'Departamento de Bioengenharia', 1)
INSERT INTO "Meeting" (date, place, id_board) VALUES (TIMESTAMP '2018/04/07
14:00:00', 'I007', 2)
INSERT INTO "Meeting" (date, place, id_board) VALUES (TIMESTAMP '2018/06/20
14:00:00', 'I007', 3)
INSERT INTO "Meeting" (date, place, id_board) VALUES (TIMESTAMP '2018/04/20
15:30:00', 'INESC', 4)
INSERT INTO "Meeting" (date, place, id_board) VALUES (TIMESTAMP '2018/04/23
10:30:00', 'B118', 5)

```

```
INSERT INTO "Meeting" (date, place, id_board) VALUES (TIMESTAMP '2018/04/20
10:00:00', 'B005', 6)
INSERT INTO "Meeting" (date, place, id_board) VALUES (TIMESTAMP '2018/04/06
17:00:00', 'Sala de bio', 7)
INSERT INTO "Meeting" (date, place, id_board) VALUES (TIMESTAMP '2018/04/10
15:00:00', 'B223', 8)
INSERT INTO "Meeting" (date, place, id_board) VALUES (TIMESTAMP '2018/04/13
09:20:00', 'B019', 9)
```

--- MESSAGE ---

```
INSERT INTO "Message" (date, message, id_user, id_project) VALUES (TIMESTAMP
'2018/04/02 15:02:46', 'Não esquecer que a interface do programa deve ser
feita em Java!!', 3, 1)
INSERT INTO "Message" (date, message, id_user, id_project) VALUES (TIMESTAMP
'2018/03/21 10:24:06', 'Ter em conta que o circuito deve ser de dimensões
reduzidas, uma vez que deve ser portátil', 1, 2)
```

--- PERSONAL EVENTS ---

```
INSERT INTO "Personal_event" (date, name, id_user) VALUES (TIMESTAMP
'2018/04/14', 'Aniversário mãe', 1)
INSERT INTO "Personal_event" (date, name, place, id_user) VALUES (TIMESTAMP
'2018/04/10', 'Almoço com o António', 'Casa do António', 2)
INSERT INTO "Personal_event" (date, name, id_user) VALUES (TIMESTAMP
'2018/04/17 13:20:00', 'Ir buscar o João ao Aeroporto', 7)
INSERT INTO "Personal_event" (date, name, id_user) VALUES (TIMESTAMP
'2018/04/21 18:00:00', 'Levar o João ao Aeroporto', 7)
INSERT INTO "Personal_event" (date, name, id_user) VALUES (TIMESTAMP
'2018/04/10 18:30:00', 'Café com a Maria', 5)
INSERT INTO "Personal_event" (date, name, id_user) VALUES (TIMESTAMP
'2018/05/18', 'Visitar a Joana', 5)
INSERT INTO "Personal_event" (date, name, id_user) VALUES (TIMESTAMP
'2018/05/06', 'Dia da mãe!!', 8)
INSERT INTO "Personal_event" (date, name, id_user) VALUES (TIMESTAMP
'2018/05/05', 'Aniversário da Maya', 3)
```

--- CONTACT ---

```
INSERT INTO "Contact" (id_user, id_contact) VALUES (1,2)
INSERT INTO "Contact" (id_user, id_contact) VALUES (1,3)
INSERT INTO "Contact" (id_user, id_contact) VALUES (1,4)
INSERT INTO "Contact" (id_user, id_contact) VALUES (1,8)
INSERT INTO "Contact" (id_user, id_contact) VALUES (2,3)
INSERT INTO "Contact" (id_user, id_contact) VALUES (2,1)
INSERT INTO "Contact" (id_user, id_contact) VALUES (2,5)
INSERT INTO "Contact" (id_user, id_contact) VALUES (3,16)
INSERT INTO "Contact" (id_user, id_contact) VALUES (4,6)
INSERT INTO "Contact" (id_user, id_contact) VALUES (4,7)
INSERT INTO "Contact" (id_user, id_contact) VALUES (4,9)
INSERT INTO "Contact" (id_user, id_contact) VALUES (4,11)
INSERT INTO "Contact" (id_user, id_contact) VALUES (5,6)
INSERT INTO "Contact" (id_user, id_contact) VALUES (5,9)
INSERT INTO "Contact" (id_user, id_contact) VALUES (6,4)
INSERT INTO "Contact" (id_user, id_contact) VALUES (6,10)
INSERT INTO "Contact" (id_user, id_contact) VALUES (7,11)
INSERT INTO "Contact" (id_user, id_contact) VALUES (8,5)
INSERT INTO "Contact" (id_user, id_contact) VALUES (8,9)
```

--- FILE ---

```
INSERT INTO "File" (description, path, publish_date, id_user, id_task)
VALUES ('Referências bibliográficas
apneia','https://docs.google.com/document/d/1KZ00yc2JSSUsC3EdrPWqTRZqUhBn4oI
B-IR61YU7EHs/edit' , TIMESTAMP '2018/04/03 10:35:07', 1, 1 )
INSERT INTO "File" (description, path, publish_date, id_user, id_task)
VALUES
('Fotopletismografia','http://www.peb.ufrj.br/teses/Tese0030_2006_03_29.pdf'
, TIMESTAMP '2018/04/03 10:35:07', 1, 2 )
INSERT INTO "File" (description, path, publish_date, id_user, id_task)
VALUES ('Datasheet
SFH7050','https://www.mouser.com/ds/2/311/SFH%207050,%20Lead%20(Pb)%20Free%2
```

```

0Product%20-%20RoHS%20Compliant-608232.pdf' , TIMESTAMP '2018/03/21
19:14:44', 1, 3 )
INSERT INTO "File" (description, path, publish_date, id_user, id_task)
VALUES ('Datasheet
TCRT1000', 'https://www.vishay.com/docs/83752/tcrt1000.pdf' , TIMESTAMP
'2018/03/29 22:51:00', 1, 3 )
INSERT INTO "File" (description, path, publish_date, id_user, id_task)
VALUES ('EEG', 'https://pt.wikipedia.org/wiki/Eletroencefalografia' ,
TIMESTAMP '2018/03/17 20:58:12', 8, 6 )
INSERT INTO "File" (description, path, publish_date, id_user, id_task)
VALUES
('EEG', 'https://docs.google.com/forms/d/e/1FAIpQLSfa60Ny5pTFxZGpUwyCp6_LiL5h
KLWMZExoCpicrAqdcR2uw/viewform' , TIMESTAMP '2018/03/03 17:49:22', 20, 9 )

```

```

--- NOTIFICATION ---

```

```

INSERT INTO "Notification" (date, notification, read, id_user) VALUES
(TIMESTAMP '2018/03/04', 'Foi adicionado a um novo projeto "Hypnos" ',
false, 3)
INSERT INTO "Notification" (date, notification, read, id_user) VALUES
(TIMESTAMP '2018/03/04', 'Foi adicionado a um novo projeto "Hypnos" ',
false, 8)
INSERT INTO "Notification" (date, notification, read, id_user) VALUES
(TIMESTAMP '2018/03/02', 'Um utilizador deseja adiciona um novo colaborador
ao seu projeto "Estudo da doença de Alzheimer através de análise de imagens
do cérebro" ', false, 3)
INSERT INTO "Notification" (date, notification, read, id_user) VALUES
(TIMESTAMP '2018/03/04', 'Utilizador "Joana Monteiro" pediu para o adicionar
à sua lista de contactos', false, 6)
INSERT INTO "Notification" (date, notification, read, id_user) VALUES
(TIMESTAMP '2018/03/04', 'Utilizador "Noelle Moran" pediu para o adicionar à
sua lista de contactos', false, 8)

```

```

--- PROGRESS UPDATE ---

```

```

INSERT INTO "Progress_update" (date, new_value, id_user, id_task) VALUES
(TIMESTAMP '2018/04/03 10:40:45', 30, 1, 1)
INSERT INTO "Progress_update" (date, new_value, id_user, id_task) VALUES
(TIMESTAMP '2018/04/03 10:37:23', 60, 3, 2)
INSERT INTO "Progress_update" (date, new_value, id_user, id_task) VALUES
(TIMESTAMP '2018/04/04 11:07:15', 80, 3, 2)
INSERT INTO "Progress_update" (date, new_value, id_user, id_task) VALUES
(TIMESTAMP '2018/03/21 19:16:35', 50, 1, 3)
INSERT INTO "Progress_update" (date, new_value, id_user, id_task) VALUES
(TIMESTAMP '2018/03/29 22:52:05', 70, 1, 3)
INSERT INTO "Progress_update" (date, new_value, id_user, id_task) VALUES
(TIMESTAMP '2018/03/27 14:46:35', 70, 8, 6)
INSERT INTO "Progress_update" (date, new_value, id_user, id_task) VALUES
(TIMESTAMP '2018/03/03 18:35:17', 40, 20, 8)

```

```

--- PROFILE PICTURES ---
-----

```

```

INSERT INTO "Profile_picture" (id_user, path) VALUES (1, 'https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTsAUelg7yg0mepGMf-
N93hh6E9XrLnKwsKa4YEQ6swhDknk4F9eQ')
INSERT INTO "Profile_picture" (id_user, path) VALUES (2, 'https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcRHMdytMDJqBYz4eR2PpDUBMepUR7SxtOmPCNdIO5
SUMQ2vdeH8')
INSERT INTO "Profile_picture" (id_user, path) VALUES (3,
'https://cdn4.iconfinder.com/data/icons/green-shopper/1068/user.png')
INSERT INTO "Profile_picture" (id_user, path) VALUES (4,
'http://icons.iconarchive.com/icons/paomedia/small-n-flat/1024/user-male-
icon.png')
INSERT INTO "Profile_picture" (id_user, path) VALUES (5, 'https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQMrYqax_J2xys7057n0QcYyBXR-AGNwPGBmaw6j-
xJXHfFwAu5')
INSERT INTO "Profile_picture" (id_user, path) VALUES (6,
'http://wfarm2.dataknet.com/static/resources/icons/set108/b5cdab07.png')
INSERT INTO "Profile_picture" (id_user, path) VALUES (7, 'https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQrLV_ettEako62R_lp4RcYFLKo0WKqxKaGpYlTzo
aEeRVZXHnGfw')
INSERT INTO "Profile_picture" (id_user, path) VALUES (8,
'https://365psd.com/images/istock/previews/9353/93539553-flat-vector-avatar-
face-character-person-portrait-user-icon.jpg')

```

```
INSERT INTO "Profile_picture" (id_user, path) VALUES (9, 'https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQMj09lsBeDLzlm9780HyMXWa0c07DnpzPy_CCYoc
3kqX85WGh4Mw')
```

```
--- PROJECT PICTURES ---
-----
```

```
INSERT INTO "Project_picture" (id_project, path) VALUES (1,
'https://image.flaticon.com/icons/svg/552/552413.svg')
INSERT INTO "Project_picture" (id_project, path) VALUES (2,
'https://image.flaticon.com/icons/svg/490/490348.svg')
INSERT INTO "Project_picture" (id_project, path) VALUES (3,
'https://image.flaticon.com/icons/svg/40/40488.svg')
INSERT INTO "Project_picture" (id_project, path) VALUES (4,
'https://image.flaticon.com/icons/png/512/228/228437.png')
INSERT INTO "Project_picture" (id_project, path) VALUES (5,
'https://image.flaticon.com/icons/svg/124/124945.svg')
```

Changes made to the first submission:

1. Changed Profile_picture and Project_picture estimated growth;
2. Changed SELECT01, in order to check if the e-mail already exists;
3. Removed indexes based on primary keys (INDEX02);
4. Changed INDEX07 to B-tree;
5. Removed TRIGGER02;
6. Added SQL code for the creation of the database;
7. Identified the document.

GROUP1734, 18/03/2018

- Ilona Generalova, up201400035@fe.up.pt
- Maria Inês Gonçalves, up201402784@fe.up.pt
- Maria Teresa Valério, up201405655@fe.up.pt
- Sara Gomes, up201405085@fe.up.pt