

# Relational Schema

## Entity Tables

These tables represent the relational model of the classes in the conceptual model, done in the previous artifact [A4](#).

user_table(id, name NN, username NN, email NN UK, password NN, phone_number, photo_path, birth_date, country_name → Country, city)							
Candidate Keys	id, email						
Primary Keys	id						
Foreign Keys	country_name						
Integrity Rule: UNIQUE(UK)	email						
Integrity Rule: NOT NULL	name, username, email, password						
name	username	email	password	phone_number	photo_path	birth_date	city
VARCHAR(64)	TEXT	TEXT	TEXT	VARCHAR(32)	TEXT	DATE	VARCHAR(64)
project(id, name NN, description)							
Candidate Keys		id					
Primary Keys		id					
Integrity Rule: NOT NULL		name					
name		description					
VARCHAR(40)		TEXT					
task(id, name NN, description, deadline, id_creator → user_table NN, id_assigned_to → user_table, id_completed_by → user_table, id_project→ project NN)							
Candidate Keys		id					
Primary Keys		id					
Foreign Keys		id_creator, id_assigned_to, id_completed_by (From user_table), id_project (From project)					
Integrity Rule: NOT NULL		name, id_creator, id_project					
name		description		deadline			
VARCHAR(40)		TEXT		TIMESTAMP			
comment(id, creation_date NN, content NN, id_user → user_table NN, id_task → task NN)							
Candidate Keys		id					
Primary Keys		id					
Foreign Keys		id_user (From user_table), id_tasks ( From task)					
Integrity Rule: NOT NULL		creation_date, content, id_user, id_task					
creation_date		content					
DATE		TEXT					
forum_post(id, title NN, creation_date NN,content NN, id_project→ project NN, date modified NN, id creator → user table NN)							

Candidate Keys	id		
Primary Keys	id		
Foreign Keys	id_project (From project), id_creator (From user_table)		
Integrity Rule: NOT NULL	title, creation_date, content, id_project, date_modified, id_creator		
<b>title</b>	<b>creation_date</b>	<b>content</b>	
VARCHAR(30)	DATE	TEXT	
<b>forum_reply(id, creation_date NN, content NN, id_forum_post → forum_post NN id_creator → user_table)</b>			
Candidate Keys	id		
Primary Keys	id		
Foreign Keys	id_forum_post (From forum_post), id_creator (From user_table)		
Integrity Rule: NOT NULL	creation_date, content, id_forum_post, id_creator		
<b>creation_date</b>	<b>content</b>		
DATE	TEXT		
<b>meeting(id, name NN, date NN, description, duration, id_project → project NN, id_creator → user_table NN)</b>			
Candidate Keys	id		
Primary Keys	id		
Foreign Keys	id_project (From project), id_creator ( From user_table)		
Integrity Rule: NOT NULL	name, date, id_project, id_creator		
<b>name</b>	<b>date</b>	<b>description</b>	<b>duration</b>
VARCHAR(40)	TIMESTAMP	TEXT	TIME
<b>file(id,name NN, path UK NN, upload_date NN, id_project→ project NN, id_uploader → user NN)</b>			
Candidate Keys	id,path		
Primary Keys	id		
Foreign Keys	id_project (From project), id_uploader (From User)		
Integrity Rule: NOT NULL	name, path, upload_date, id_project, id_uploader		
Integrity Rule: UNIQUE	path		
<b>name</b>	<b>path</b>	<b>upload_date</b>	
VARCHAR(60)	TEXT	TIMESTAMP	
<b>tag(id,name UK NN)</b>			
Candidate Keys	id, name		
Primary Keys	id		
Integrity Rule: NOT NULL	name		
Integrity Rule: UNIQUE	name		
<b>name</b>			
VARCHAR(60)			
<b>country(id, name NN UK)</b>			
Candidate Keys	id, name		
Primary Key	id		
Integrity Rule:UNIQUE (UK)	name		
Integrity Rule:NOT NULL (NN)	name		
<b>name</b>			
VARCHAR(50)			

# Relations Tables

These tables represent the relational model of the many-to-many relations in the conceptual model, done in [A4](#)

<b>user_project(id_user → user_table, id_project → project, is_coordinator)</b>	
Primary Keys	(id_user, id_project)
Foreign Keys	id_user (From user_table), id_project(From project)
Integrity Rule: NOT NULL	is_coordinator
<b>is_coordinator</b>	
BOOLEAN	
<b>task_tag(id_tag → tag, id_task → task)</b>	
Primary Keys	(id_tag, id_tasks)
Foreign Keys	id_tag (From tag), id_task(From tasks)
<b>user_meeting(id_user → user_table, id_meeting → meeting, is_creator NN)</b>	
Candidate Keys	(id_user, id_meeting)
Primary Keys	id
Foreign Keys	id_user (From user_table), id_meeting ( From meeting)
Integrity Rule: NOT NULL	is_creator
<b>is_creator</b>	
BOOLEAN	
<b>file_meeting(id_file → file, id_meeting → meeting)</b>	
Primary Keys	(id_file, id_meeting)
Foreign Keys	id_file (From file), id_meeting ( From meeting)
<b>file_tag (id_tag → tag, id_file → file)</b>	
Primary Keys	(id_tag, id_file)
Foreign Keys	id_tag (From tag), id_file ( From file)
<b>comment_like (id_comment → comment, id_user → user_table )</b>	
Primary Keys	(id_comment, id_user)
Foreign Keys	id_comment (From comment), id_user ( From user_table)
<b>forum_post_like(id_post → forum_post, id_user → user_table)</b>	
Primary Keys	(id_post, id_user)
Foreign Keys	id_post (From post), id_user ( From user_table)
<b>forum_reply_like(id_reply → reply, id_user → user_table)</b>	
Primary Keys	(id_reply, id_user)
Foreign Keys	id_reply (From reply), id_user ( From user_table)

# Functional Dependencies

User_table	id → username, email, password, phone-number, photo-path, birth-date, country, city
	email → id, username, password, phone-number, photo-path, birth-date, country, city
Project	id → name, description
Task	id → name, description, deadline, id-creator, id-assigned-to, id-completed-by, id-project
Comment	id → creation-date, content, id-user, id-task
Forum_post	id → title, creation-date, content, id-project
Reply	id → creation-date, content, id-forum-post
Meeting	id → name, date, description, duration, id-project, id-creator, id-project
File	id → name, path, upload-date, id-project
	path → name, id, upload-date, id-project
Tag	id → name
	name → id
Country	name

Because there aren't any functional dependencies between non-key attributes, and all table attributes functionally depend on all candidate keys of their table, then we don't need to normalise the scheme, because it is already on the Boyce-Codd Normal Form.

## SQL Code

```
-- Tables
```

```
CREATE TABLE task (
  id SERIAL NOT NULL,
  name CHARACTER VARYING(128) NOT NULL,
  description CHARACTER VARYING(512),
  deadline DATE,
  creator_id INTEGER NOT NULL,
  assigned_id INTEGER,
  completer_id INTEGER,
  project_id INTEGER NOT NULL
);
```

```
CREATE TABLE user_table (
  id SERIAL NOT NULL,
  name CHARACTER VARYING(64) NOT NULL,
  username CHARACTER VARYING(32) NOT NULL,
  email CHARACTER VARYING(64) NOT NULL,
```

```
password CHARACTER VARYING(64) NOT NULL,  
phone_number CHARACTER VARYING(32),  
photo_path CHARACTER VARYING(256),  
birth_date DATE,  
country INTEGER,  
city CHARACTER VARYING(64)  
);
```

```
CREATE TABLE comment (  
    id SERIAL NOT NULL,  
    creation_date DATE NOT NULL,  
    content CHARACTER VARYING(512) NOT NULL,  
    id_user INTEGER NOT NULL,  
    id_task INTEGER NOT NULL  
);
```

```
CREATE TABLE comment_like (  
    id_comment INTEGER NOT NULL,  
    id_user INTEGER NOT NULL  
);
```

```
CREATE TABLE file (  
    id SERIAL NOT NULL,  
    upload_date DATE NOT NULL,  
    uploader_id INTEGER NOT NULL,  
    project_id INTEGER NOT NULL,  
    name CHARACTER VARYING(64) NOT NULL,  
    path CHARACTER VARYING(256) NOT NULL  
);
```

```
CREATE TABLE file_meeting (  
    file_id INTEGER NOT NULL,  
    tag_id INTEGER NOT NULL  
);
```

```
CREATE TABLE file_tag (  
    id_tag INTEGER NOT NULL,  
    id_file INTEGER NOT NULL  
);
```

```
CREATE TABLE forum_post (  
    id SERIAL NOT NULL,  
    title CHARACTER VARYING(128) NOT NULL,  
    creation_date DATE NOT NULL,  
    content CHARACTER VARYING(512) NOT NULL,  
    id_project INTEGER NOT NULL,  
    date_modified DATE NOT NULL,
```

```
    id_creator INTEGER NOT NULL,
);

CREATE TABLE forum_post_like (
    id_post INTEGER NOT NULL,
    id_user INTEGER NOT NULL
);

CREATE TABLE forum_reply (
    id SERIAL NOT NULL,
    creation_date DATE NOT NULL,
    content CHARACTER VARYING(512) NOT NULL,
    post_id INTEGER NOT NULL,
    id_creator INTEGER NOT NULL
);

CREATE TABLE forum_reply_like (
    reply_id INTEGER NOT NULL,
    user_id INTEGER NOT NULL
);

CREATE TABLE meeting (
    id SERIAL NOT NULL,
    name CHARACTER VARYING(64) NOT NULL,
    DATE DATE NOT NULL,
    duration INTEGER,
    description CHARACTER VARYING(512),
    id_creator INTEGER NOT NULL,
    id_project INTEGER NOT NULL
);

CREATE TABLE project (
    id SERIAL NOT NULL,
    name CHARACTER VARYING(64) NOT NULL,
    description CHARACTER VARYING(512)
);

CREATE TABLE tag (
    id SERIAL NOT NULL,
    name CHARACTER VARYING(32) NOT NULL
);

CREATE TABLE task_tag (
    task_id INTEGER NOT NULL,
```

```
    tag_id INTEGER NOT NULL
);

CREATE TABLE user_meeting (
    meeting_id INTEGER,
    user_id INTEGER
);

CREATE TABLE user_project (
    id_user INTEGER NOT NULL,
    id_project INTEGER NOT NULL,
    is_coordinator BOOLEAN NOT NULL
);

-- Create constraints

-- Primary Keys

ALTER TABLE ONLY task
    ADD CONSTRAINT task_pkey PRIMARY KEY (id);

ALTER TABLE ONLY user_table
    ADD CONSTRAINT user_email_key UNIQUE (email);

ALTER TABLE ONLY user_table
    ADD CONSTRAINT user_pkey PRIMARY KEY (id);

ALTER TABLE ONLY comment
    ADD CONSTRAINT comment_pkey PRIMARY KEY (id);

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 forum_post
    ADD CONSTRAINT forum_post_pkey PRIMARY KEY (id);

ALTER TABLE ONLY forum_reply
    ADD CONSTRAINT forum_reply_pkey PRIMARY KEY (id);

ALTER TABLE ONLY meeting
    ADD CONSTRAINT meeting_pkey PRIMARY KEY (id);

ALTER TABLE ONLY project
    ADD CONSTRAINT project_pkey PRIMARY KEY (id);
```

```
ALTER TABLE ONLY tag
    ADD CONSTRAINT tag_name_key UNIQUE (name);

ALTER TABLE ONLY tag
    ADD CONSTRAINT tag_pkey PRIMARY KEY (id);

-- Foreign Keys

ALTER TABLE ONLY user_project
    ADD CONSTRAINT project_user_id_project_fkey FOREIGN KEY (id_project)
REFERENCES project(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY task
    ADD CONSTRAINT task_assigned_id_fkey FOREIGN KEY (assigned_id)
REFERENCES user_table(id);

ALTER TABLE ONLY task
    ADD CONSTRAINT task_completer_id_fkey FOREIGN KEY (completer_id)
REFERENCES user_table(id);

ALTER TABLE ONLY task
    ADD CONSTRAINT task_creator_id_fkey FOREIGN KEY (creator_id) REFERENCES
user_table(id);

ALTER TABLE ONLY task
    ADD CONSTRAINT task_project_id_fkey FOREIGN KEY (project_id) REFERENCES
project(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY comment_like
    ADD CONSTRAINT comment_like_id_comment_fkey FOREIGN KEY (id_comment)
REFERENCES comment(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY comment_like
    ADD CONSTRAINT comment_like_id_user_fkey FOREIGN KEY (id_user)
REFERENCES user_table(id);

ALTER TABLE ONLY comment
    ADD CONSTRAINT comment_id_task_fkey FOREIGN KEY (id_task) REFERENCES
task(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY comment
    ADD CONSTRAINT comment_id-user_fkey FOREIGN KEY (id_user) REFERENCES
user_table(id);

ALTER TABLE ONLY file_meeting
    ADD CONSTRAINT file_meeting_file_id_fkey FOREIGN KEY (file_id)
REFERENCES file(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY file_meeting
```



```
ADD CONSTRAINT file_meeting_tag_id_fkey FOREIGN KEY (tag_id) REFERENCES
tag(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY file
ADD CONSTRAINT file_project_id_fkey FOREIGN KEY (project_id) REFERENCES
project(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY file_tag
ADD CONSTRAINT file_tag_id_file_fkey FOREIGN KEY (id_file) REFERENCES
file(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY file_tag
ADD CONSTRAINT file_tag_id_tag_fkey FOREIGN KEY (id_tag) REFERENCES
tag(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY file
ADD CONSTRAINT file_uploader_id_fkey FOREIGN KEY (uploader_id)
REFERENCES user_table(id);

ALTER TABLE ONLY forum_post
ADD CONSTRAINT forum_post_id_creator_fkey FOREIGN KEY (id_creator)
REFERENCES user_table(id);

ALTER TABLE ONLY forum_post_like
ADD CONSTRAINT forum_post_like_id_post_fkey FOREIGN KEY (id_post)
REFERENCES forum_post(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY forum_post_like
ADD CONSTRAINT forum_post_like_id_user_fkey FOREIGN KEY (id_user)
REFERENCES user_table(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY forum_reply_like
ADD CONSTRAINT forum_reply_like_reply_id_fkey FOREIGN KEY (reply_id)
REFERENCES comment(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY forum_reply_like
ADD CONSTRAINT forum_reply_like_user_id_fkey FOREIGN KEY (user_id)
REFERENCES user_table(id);

ALTER TABLE ONLY forum_reply
ADD CONSTRAINT forum_reply_post_id_fkey FOREIGN KEY (post_id) REFERENCES
forum_post(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY forum_post
ADD CONSTRAINT forum_reply_id_creator_fkey FOREIGN KEY (id_creator)
REFERENCES user_table(id);

ALTER TABLE ONLY meeting
ADD CONSTRAINT meeting_id_creator_fkey FOREIGN KEY (id_creator)
REFERENCES user_table(id);
```

```
ALTER TABLE ONLY meeting
  ADD CONSTRAINT meeting_id_project_fkey FOREIGN KEY (id_project)
REFERENCES project(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY task_tag
  ADD CONSTRAINT task_tag_tag_id_fkey FOREIGN KEY (tag_id) REFERENCES
tag(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY task_tag
  ADD CONSTRAINT task_tag_task_id_fkey FOREIGN KEY (task_id) REFERENCES
task(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY user_project
  ADD CONSTRAINT user_id FOREIGN KEY (id_user) REFERENCES user_table(id);

ALTER TABLE ONLY user_meeting
  ADD CONSTRAINT user_meeting_meeting_id_fkey FOREIGN KEY (meeting_id)
REFERENCES meeting(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY user_meeting
  ADD CONSTRAINT user_meeting_user_id_fkey FOREIGN KEY (user_id)
REFERENCES user_table(id);
```

From:

<http://lbaw.fe.up.pt/201617/> - **L B A W :: WORK**

Permanent link:

<http://lbaw.fe.up.pt/201617/doku.php/lbaw1614/proj/a5>

Last update: **2017/03/24 15:25**

