

A5 – Relational schema, validation and schema refinement

This artifact contains the relational schema obtained by mapping the conceptual data model made in the previous week, which allows a different approach of the entities and relationship involved in a project management web application.

1. Relational Schema

The relational schema includes the relational schema attributes, domains, keys (primary and foreign) and other integrity rules.

R01	User (<u>id</u> , e-mail NN UK , full_name, password NN , URL NN UK , username NN UK , administrator NN DF 0 , <u>user_ban</u> → User)
R02	Project (<u>id</u> , description, end_date CK end_date > start_date, name NN , privacy DF privacy=0, start_date NN DF today, state NN CK state IN stateType DF 'In_progress', id_coordinator NN → User, <u>user_archived</u> → User)
R03	Board (<u>id</u> , description, name NN , id_creator NN → User, id_project NN → Project)
R04	Task (<u>id</u> , budget, deadline NN , description, name NN , progress CK 0 ≤ progress ≤ 100, state NN CK state IN stateType DF 'In_progress', id_creator NN → User, id_board NN → Board)
R05	Meeting (<u>id</u> , date NN CK date > today, name, place, id_board NN → Board)
R06	Personal_event (<u>id</u> , date NN CK date > today, name, place, id_user NN → User)
R07	Comment (<u>id</u> , comment NN , date NN DF today, id_user NN → User, id_task NN → Task)
R08	File (<u>id</u> , description, path NN , publish_date NN DF today, id_user NN → User, id_task NN → Task)
R09	Notification (<u>id</u> , date NN DF today, notification NN , read NN DF 0 , id_user NN → User)
R10	Progress_update (<u>id</u> , date NN DF today, new_value CK 0 ≤ new_value ≤ 100, id_user NN → User, id_task NN → Task)
R11	Message (<u>id</u> , date NN DF today, message NN , id_user NN → User, id_project NN → Project)
R12	Project_team (id_project NN → Project, id_worker NN → User)
R13	Board_team (id_board NN → Board, id_worker NN → User)
R14	Profile_picture (id_image NN , id_user NN → User, path UK)
R15	Project_picture (id_image NN , id_project NN → Project, path UK)
R16	Contact (id_user NN → User, id_contact NN → User)
R17	Ban_user (id_administrator → User, id_user → User, date)
R18	Archive_project (id_project → Project, id_administrator → User, date)

Relational schemas are presented above, where UK means UNIQUE KEY, NN means NOT NULL, DF means DEFAULT and CK means CHECK.

2. Domains

Specification of additional domains:

today	DATE DEFAULT CURRENT_DATE
stateType	ENUM ('Closed', 'In_progress', 'Archived')

3. Functional dependencies and schema validation

To validate the Relational Schema obtained from the Conceptual Model, all functional dependencies are identified, and the normalization of all relation schemas is accomplished. ~~All relations schemas are in the Boyce-Codd Normal Form, so the relational schema is also in the BCNF and there is no need to be refined using normalization.~~

Table R01 (User)	
Keys: {id}, {e-mail}, {URL}, {username}	
Functional Dependencies	
FD0101	{id} → {e-mail, full_name, password, URL, username, administrator, user_ban}
FD0102	{e-mail} → {id, full_name, password, URL, username, administrator, user_ban}
FD0103	{URL} → {id, e-mail, full_name, password, username, administrator, user_ban}
FD0104	{username} → {id, e-mail, full_name, password, URL, administrator, user_ban}
NORMAL FORM	BCNF

Formatada: Português (Portugal)

Formatada: Português (Portugal)

Formatada: Inglês (Estados Unidos)

Table R02 (Project)	
Keys: {id}	
Functional Dependencies	
FD0201	{id} → {description, end_date, name, privacy, start_date, state, id_coordinator}
NORMAL FORM	BCNF

Table R03 (Board)	
Keys: {id}	
Functional Dependencies	
FD0301	{id} → {description, name, id_creator, id_project}
NORMAL FORM	BCNF

Table R04 (Task)	
Keys: {id}	
Functional Dependencies	
FD0401	{id} → {budget, deadline, description, name, progress, state, id_creator, id_board}
NORMAL FORM	BCNF

Table R05 (Meeting)	
Keys: {id}	
Functional Dependencies	

FD0501	{id} → {date, name, place, id_board}
NORMAL FORM	BCNF

Table R06 (Personal_event)	
Keys: {id}	
Functional Dependencies	
FD0601	{id} → {date, name, place, id_user}
NORMAL FORM	BCNF

Table R07 (Comment)	
Keys: {id}	
Functional Dependencies	
FD0701	{id} → {comment, date, id_user, id_task}
NORMAL FORM	BCNF

Table R08 (File)	
Keys: {id}	
Functional Dependencies	
FD0801	{id} → {description, path, publish_date, id_user, id_task}
NORMAL FORM	BCNF

Table R09 (Notification)	
Keys: {id, id_user}	
Functional Dependencies	
FD0901	{id, id_user} → {date, notification, state}
NORMAL FORM	BCNF

Table R10 (Progress_update)	
Keys: {id}	
Functional Dependencies	
FD1001	{id} → {date, new_value, id_user, id_task}
NORMAL FORM	BCNF

Table R11 (Message)	
Keys: {id}	
Functional Dependencies	
FD1101	{id} → {date, message, id_user, id_project}
NORMAL FORM	BCNF

Table R12 (Project_team)	
Keys: {id_project, id_worker}	

Functional Dependencies	
(none)	
NORMAL FORM	BCNF

Table R13 (Board_team)	
Keys: {id_board, id_worker}	
Functional Dependencies	
(none)	
NORMAL FORM	BCNF

Table R14 (Profile_picture)	
Keys: {id_image, <u>path</u> }	
Functional Dependencies	
FD1401	id_image → {id_user, path}
<u>FD1402</u>	<u>path → {id_image, id_user}</u>
NORMAL FORM	BCNF

Table R15 (Project_picture)	
Keys: {id_image, <u>path</u> }	
Functional Dependencies	
FD1501	id_image → {id_project, path}
<u>FD1502</u>	<u>path → {id_image, id_project}</u>
NORMAL FORM	BCNF

Table R16 (Contact)	
Keys: {id_user, id_contact}	
Functional Dependencies	
(none)	
NORMAL FORM	BCNF

Table R17 (Ban_user)	
Keys: {id_user}	
Functional Dependencies	
FD1701	id_user → {id_administrator}
NORMAL FORM	BCNF

Table R18 (Archive_project)	
Keys: {id_project}	
Functional Dependencies	
FD1801	id_project → {id_administrator}
NORMAL FORM	BCNF

All relations schemas are in the Boyce-Codd Normal Form, so the relational schema is also in the BCNF and there is no need to be refined using normalization.

4. SQL Code

(<https://github.com/TeresaValerio/LBAW1734/blob/master/Artefactos/A5/database.sql>)

Código de campo alterado

```
--
-- 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 "Personal_event" (  
    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);
```

Changes made to the first submission:

1. Added NN for foreign keys where restriction was missing;
2. Deleted Archive_project and Ban_user relations;
3. Added user as foreign key of user (to represent the user that banned him) and project (to represent the user that archived that project);
4. Added keys and functional dependencies for "User";
5. Added key "path" and functional dependency for "Profile_picture" and "Project_picture";
6. Moved BCNF justification to after the tables (previously was in the beginning);
7. Added domains in SQL

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