

# A5: Esquema relacional, validação e melhoramento

Este artefacto contém o esquema relacional obtido através do mapeamento do modelo conceptual de dados. O esquema relacional inclui o esquema de relações, atributos, domínios, chaves primárias, chaves estrangeiras e outras regras de integridade: UNIQUE, DEFAULT, NOT NULL, CHECK.

## 1. Esquema Relacional

R01	user( <u>id</u> , username <b>UK NN</b> , password <b>NN</b> , email <b>UK NN</b> , registDate <b>NN DF</b> Today, firstName <b>NN</b> , lastName <b>NN</b> , imagePath, city_id → city <b>NN</b> )
R02	event( <u>id</u> , name <b>NN</b> , date <b>NN</b> date > Today, description <b>NN</b> , owner_id → user <b>NN</b> , localization_id → localization <b>NN</b> , image_id → image, type <b>NN CK</b> type IN Types_of_Events, category <b>NN CK</b> category IN Types_of_Categories)
R03	done(event_id → event, rating <b>NN CK</b> rating > 0 AND rating <= 5)
R04	notDone(event_id → event)
R05	participant(user_id → user, event_id → event)
R06	owner(user_id → user, event_id → event)
R07	image( <u>id</u> , path <b>NN UK</b> )
R08	localization( <u>id</u> , name, address <b>NN</b> , latitude, longitude, city_id → city <b>NN</b> )
R09	city( <u>id</u> , name <b>UK NN</b> , country_id → country <b>NN</b> )
R10	country( <u>id</u> , name <b>UK NN</b> )
R11	post( <u>id</u> , description <b>NN</b> , date <b>NN DF</b> Today, event_id → event <b>NN</b> , user_id → user <b>NN</b> , image_id → image)
R12	poll( <u>id</u> , post_id → post <b>NN</b> )
R13	option( <u>id</u> , description <b>NN</b> , poll_id → poll <b>NN</b> )
R14	admin( <u>id</u> , username <b>UK NN</b> , password <b>NN</b> , email <b>UK NN</b> )
R15	friendRequest( <u>id</u> , answer, sender_id → user <b>NN</b> , receiver_id → user <b>NN</b> )
R16	friendActivity( <u>id</u> , sender_id → user <b>NN</b> , receiver_id → user <b>NN</b> , event_id → event <b>NN</b> )
R17	eventInvite( <u>id</u> , answer, event_id → notDone <b>NN</b> , owner_id → user <b>NN</b> , receiver_id → user <b>NN</b> )

## 2. Domínios

Today	DATE DEFAULT CURRENT_DATE
Types of Events	ENUM('Public', 'Private')
Types of Categories	ENUM('Music', 'Sports', 'Entertainment', 'Educational', 'Business', 'Other')

## 3. Dependências funcionais e validação do esquema

<b>Table R01 (user)</b>	
<b>Keys:</b> {id}, {username}, {email}	
<b>Dependências Funcionais</b>	
FD0101	{id} → {username, email, password, registDate, firstName, lastName, image, city}
FD0102	{username} → {id, email, password, registDate, firstName, lastName, image, city}
FD0103	{email} → {id, username, password, registDate, firstName, lastName, image, city}
<b>Forma Normal</b>	BCNF

<b>Table R02 (event)</b>	
<b>Keys:</b> {id}	
<b>Dependências Funcionais</b>	
FD0201	{id} → {name, date, type, category, description, localization, owner, image_id}
<b>Forma Normal</b>	BCNF

<b>Table R03 (done)</b>	
<b>Keys:</b> {event_id}	
<b>Dependências Funcionais</b>	
FD0301	{event_id} → {rating}
<b>Forma Normal</b>	BCNF

<b>Table R04 (notDone)</b>	
<b>Keys:</b> {event_id}	
<b>Dependências Funcionais</b>	
(none)	
<b>Forma Normal</b>	BCNF

<b>Table R05</b> (participant)	
<b>Keys:</b> {user_id, event_id}	
<b>Dependências Funcionais</b>	
(none)	
<b>Forma Normal</b>	BCNF

<b>Table R06</b> (owner)	
<b>Keys:</b> {user_id, event_id}	
<b>Dependências Funcionais</b>	
(none)	
<b>Forma Normal</b>	BCNF

<b>Table R07</b> (image)	
<b>Keys:</b> {id}	
<b>Dependências Funcionais</b>	
FD0701	{id} → {path}
<b>Forma Normal</b>	BCNF

<b>Table R08</b> (localization)	
<b>Keys:</b> {id}	
<b>Dependências Funcionais</b>	
FD0801	{id} → {name, address, latitude, longitude, city_id}
<b>Forma Normal</b>	BCNF

<b>Table R09</b> (city)	
<b>Keys:</b> {id}	
<b>Dependências Funcionais</b>	
FD0901	{id} → {name, country_id}
<b>Forma Normal</b>	BCNF

<b>Table R10</b> (country)	
<b>Keys:</b> {id}	
<b>Dependências Funcionais</b>	
FD1001	{id} → {name}
<b>Forma Normal</b>	BCNF

<b>Table R11</b> (post)	
<b>Keys:</b> {id}	
<b>Dependências Funcionais</b>	
FD1101	{id} → {description, date, event, user_id, image}
<b>Forma Normal</b>	BCNF

<b>Table R12 (poll)</b>	
<b>Keys:</b> {id}	
<b>Dependências Funcionais</b>	
FD1201	{id} → {post}
<b>Forma Normal</b>	BCNF

<b>Table R13 (option)</b>	
<b>Keys:</b> {id}	
<b>Dependências Funcionais</b>	
FD1301	{id} → {description, poll}
<b>Forma Normal</b>	BCNF

<b>Table R14 (admin)</b>	
<b>Keys:</b> {id}, {username}, {email}	
<b>Dependências Funcionais</b>	
FD1401	{id} → {username, email, password}
FD1402	{username} → {id, email, password}
FD1403	{email} → {id, username, password}
<b>Forma Normal</b>	BCNF

<b>Table R15 (friendRequest)</b>	
<b>Keys:</b> {id}	
<b>Dependências Funcionais</b>	
FD1501	{id} → {answer, sender_id, receiver_id}
<b>Forma Normal</b>	BCNF

<b>Table R16 (friendActivity)</b>	
<b>Keys:</b> {id}	
<b>Dependências Funcionais</b>	
FD1601	{id} → {sender_id, receiver_id, event_id}
<b>Forma Normal</b>	BCNF

<b>Table R17 (eventInvite)</b>	
<b>Keys:</b> {id}	
<b>Dependências Funcionais</b>	
FD1701	{id} → {answer, event_id, owner_id, receiver_id }
<b>Forma Normal</b>	BCNF

## 4. SQL Código

### [Sql code](#)

```
--Types Enums

DROP TYPE IF EXISTS category_type CASCADE;
CREATE TYPE category_type AS ENUM(
    'Music',
    'Sports',
    'Entertainment',
    'Educational',
    'Business',
    'Other'
);

DROP TYPE IF EXISTS event_type CASCADE;
CREATE TYPE event_type AS ENUM(
    'Public',
    'Private'
);

--Tables

DROP TABLE IF EXISTS admin CASCADE;
CREATE TABLE admin (
    id SERIAL NOT NULL,
    username text NOT NULL,
    password text NOT NULL,
    email text NOT NULL
);

DROP TABLE IF EXISTS city CASCADE;
CREATE TABLE city (
    id SERIAL NOT NULL,
    name text NOT NULL,
    country_id INTEGER NOT NULL
);

DROP TABLE IF EXISTS country CASCADE;
CREATE TABLE country (
    id SERIAL NOT NULL,
    name text NOT NULL
);

DROP TABLE IF EXISTS done CASCADE;
CREATE TABLE done (
    event_id INTEGER NOT NULL,
    rating INTEGER NOT NULL,
    CONSTRAINT rating_ck CHECK (((rating > 0) AND (rating <= 5)))
);
```

```
DROP TABLE IF EXISTS event CASCADE;
CREATE TABLE event (
    id SERIAL NOT NULL,
    name text NOT NULL,
    date TIMESTAMP WITH TIME zone NOT NULL,
    description text NOT NULL,
    owner_id INTEGER NOT NULL,
    localization_id INTEGER NOT NULL,
    image_id INTEGER NOT NULL,
    type event_type NOT NULL,
    category category_type NOT NULL,
    CONSTRAINT date_ck CHECK ((date > now()))
);
```

```
DROP TABLE IF EXISTS eventInvite CASCADE;
CREATE TABLE eventInvite (
    id SERIAL NOT NULL,
    answer text NOT NULL,
    event_id INTEGER NOT NULL,
    owner_id INTEGER NOT NULL,
    receiver_id INTEGER NOT NULL
);
```

```
DROP TABLE IF EXISTS friendActivity CASCADE;
CREATE TABLE friendActivity (
    id SERIAL NOT NULL,
    sender_id INTEGER NOT NULL,
    receiver_id INTEGER NOT NULL,
    event_id INTEGER NOT NULL
);
```

```
DROP TABLE IF EXISTS friendRequest CASCADE;
CREATE TABLE friendRequest (
    id SERIAL NOT NULL,
    answer text NOT NULL,
    sender_id INTEGER NOT NULL,
    receiver_id INTEGER NOT NULL,
    event_id INTEGER NOT NULL
);
```

```
DROP TABLE IF EXISTS image CASCADE;
CREATE TABLE image (
    id SERIAL NOT NULL,
    path text NOT NULL
);
```

```
DROP TABLE IF EXISTS localization CASCADE;
CREATE TABLE localization (
```

```
    id SERIAL NOT NULL,
    name text NOT NULL,
    address text NOT NULL,
    latitude FLOAT,
    longitude FLOAT,
    city_id INTEGER NOT NULL
);

DROP TABLE IF EXISTS notDone CASCADE;
CREATE TABLE notDone (
    event_id INTEGER NOT NULL
);

DROP TABLE IF EXISTS option CASCADE;
CREATE TABLE option (
    id SERIAL NOT NULL,
    description text NOT NULL,
    poll_id INTEGER NOT NULL
);

DROP TABLE IF EXISTS owner CASCADE;
CREATE TABLE owner (
    user_id INTEGER NOT NULL,
    event_id INTEGER NOT NULL
);

DROP TABLE IF EXISTS participant CASCADE;
CREATE TABLE participant (
    user_id INTEGER NOT NULL,
    event_id INTEGER NOT NULL
);

DROP TABLE IF EXISTS poll CASCADE;
CREATE TABLE poll (
    id SERIAL NOT NULL,
    post_id INTEGER NOT NULL
);

DROP TABLE IF EXISTS post CASCADE;
CREATE TABLE post (
    id SERIAL NOT NULL,
    description text NOT NULL,
    date TIMESTAMP WITH TIME zone NOT NULL,
    event_id INTEGER NOT NULL,
    user_id INTEGER NOT NULL,
    image_id INTEGER NOT NULL
);

DROP TABLE IF EXISTS "user" CASCADE;
```

```
CREATE TABLE "user" (  
    id SERIAL NOT NULL,  
    username text NOT NULL,  
    password text NOT NULL,  
    email text NOT NULL,  
    registDate TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,  
    firstName text NOT NULL,  
    lastName text NOT NULL,  
    imagePath text NOT NULL,  
    city_id INTEGER NOT NULL  
);
```

-- Primary Keys and Uniques

```
ALTER TABLE ONLY admin  
    ADD CONSTRAINT admin_pk PRIMARY KEY (id);
```

```
ALTER TABLE ONLY admin  
    ADD CONSTRAINT admin_email_uk UNIQUE (email);
```

```
ALTER TABLE ONLY city  
    ADD CONSTRAINT city_pk PRIMARY KEY (id);
```

```
ALTER TABLE ONLY city  
    ADD CONSTRAINT city_name_uk UNIQUE (name);
```

```
ALTER TABLE ONLY country  
    ADD CONSTRAINT country_pk PRIMARY KEY (id);
```

```
ALTER TABLE ONLY country  
    ADD CONSTRAINT country_name_uk UNIQUE (name);
```

```
ALTER TABLE ONLY done  
    ADD CONSTRAINT done_pk PRIMARY KEY (event_id);
```

```
ALTER TABLE ONLY event  
    ADD CONSTRAINT event_pk PRIMARY KEY (id);
```

```
ALTER TABLE ONLY eventInvite  
    ADD CONSTRAINT eventInvite_pk PRIMARY KEY (id);
```

```
ALTER TABLE ONLY friendActivity  
    ADD CONSTRAINT friendActivity_pk PRIMARY KEY (id);
```

```
ALTER TABLE ONLY friendRequest  
    ADD CONSTRAINT friendRequest_pk PRIMARY KEY (id);
```

```
ALTER TABLE ONLY image  
    ADD CONSTRAINT image_pk PRIMARY KEY (id);
```



```

ALTER TABLE ONLY image
    ADD CONSTRAINT image_path_uk UNIQUE (path);

ALTER TABLE ONLY localization
    ADD CONSTRAINT localization_pk PRIMARY KEY (id);

ALTER TABLE ONLY notDone
    ADD CONSTRAINT notDone_pk PRIMARY KEY (event_id);

ALTER TABLE ONLY option
    ADD CONSTRAINT option_pk PRIMARY KEY (id);

ALTER TABLE ONLY owner
    ADD CONSTRAINT owner_pk PRIMARY KEY (user_id, event_id);

ALTER TABLE ONLY participant
    ADD CONSTRAINT participant_pk PRIMARY KEY (user_id, event_id);

ALTER TABLE ONLY poll
    ADD CONSTRAINT poll_pk PRIMARY KEY (id);

ALTER TABLE ONLY post
    ADD CONSTRAINT post_pk PRIMARY KEY (id);

ALTER TABLE ONLY "user"
    ADD CONSTRAINT user_pk PRIMARY KEY (id);

ALTER TABLE ONLY "user"
    ADD CONSTRAINT user_name_uk UNIQUE (username);

ALTER TABLE ONLY "user"
    ADD CONSTRAINT user_email_uk UNIQUE (email);

-- Foreign Keys

ALTER TABLE ONLY city
    ADD CONSTRAINT city_country_id_fk FOREIGN KEY (country_id) REFERENCES
country(id) ON DELETE SET NULL;

ALTER TABLE ONLY done
    ADD CONSTRAINT done_event_id_fk FOREIGN KEY (event_id) REFERENCES
event(id) ON UPDATE CASCADE;

ALTER TABLE ONLY event
    ADD CONSTRAINT event_owner_id_fk FOREIGN KEY (owner_id) REFERENCES
"user"(id) ON UPDATE CASCADE;

ALTER TABLE ONLY event

```

```
ADD CONSTRAINT event_localization_id_fk FOREIGN KEY (localization_id)
REFERENCES localization(id) ON DELETE SET NULL;

ALTER TABLE ONLY event
ADD CONSTRAINT event_image_id_fk FOREIGN KEY (image_id) REFERENCES
image(id) ON DELETE SET NULL;

ALTER TABLE ONLY eventInvite
ADD CONSTRAINT eventInvite_event_id_fk FOREIGN KEY (event_id)
REFERENCES notDone(event_id) ON DELETE CASCADE;

ALTER TABLE ONLY eventInvite
ADD CONSTRAINT eventInvite_owner_id_fk FOREIGN KEY (owner_id)
REFERENCES "user"(id) ON DELETE SET NULL;

ALTER TABLE ONLY eventInvite
ADD CONSTRAINT eventInvite_receiver_id_fk FOREIGN KEY (receiver_id)
REFERENCES "user"(id) ON DELETE SET NULL;

ALTER TABLE ONLY friendActivity
ADD CONSTRAINT friendActivity_sender_id_fk FOREIGN KEY (sender_id)
REFERENCES "user"(id) ON DELETE SET NULL;

ALTER TABLE ONLY friendActivity
ADD CONSTRAINT friendActivity_receiver_id_fk FOREIGN KEY
(receiver_id) REFERENCES "user"(id) ON DELETE SET NULL;

ALTER TABLE ONLY friendActivity
ADD CONSTRAINT friendActivity_event_id_fk FOREIGN KEY (event_id)
REFERENCES event(id) ON DELETE CASCADE;

ALTER TABLE ONLY friendRequest
ADD CONSTRAINT friendRequest_sender_id_fk FOREIGN KEY (sender_id)
REFERENCES "user"(id) ON DELETE CASCADE;

ALTER TABLE ONLY friendRequest
ADD CONSTRAINT friendRequest_receiver_id_fk FOREIGN KEY (receiver_id)
REFERENCES "user"(id) ON DELETE CASCADE;

ALTER TABLE ONLY localization
ADD CONSTRAINT localization_city_id_fk FOREIGN KEY (city_id)
REFERENCES city(id) ON DELETE SET NULL;

ALTER TABLE ONLY notDone
ADD CONSTRAINT notDone_event_id_fk FOREIGN KEY (event_id) REFERENCES
event(id) ON UPDATE CASCADE;

ALTER TABLE ONLY option
```

```
    ADD CONSTRAINT option_poll_id_fk FOREIGN KEY (poll_id) REFERENCES
poll(id) ON DELETE CASCADE;

ALTER TABLE ONLY owner
    ADD CONSTRAINT owner_user_id_fk FOREIGN KEY (user_id) REFERENCES
"user"(id) ON UPDATE CASCADE;

ALTER TABLE ONLY owner
    ADD CONSTRAINT owner_event_id_fk FOREIGN KEY (event_id) REFERENCES
event(id) ON DELETE SET NULL;

ALTER TABLE ONLY participant
    ADD CONSTRAINT participant_user_id_fk FOREIGN KEY (user_id)
REFERENCES "user"(id) ON UPDATE CASCADE;

ALTER TABLE ONLY participant
    ADD CONSTRAINT participant_event_id_fk FOREIGN KEY (event_id)
REFERENCES event(id) ON DELETE SET NULL;

ALTER TABLE ONLY poll
    ADD CONSTRAINT poll_post_id_fk FOREIGN KEY (post_id) REFERENCES
post(id) ON DELETE CASCADE;

ALTER TABLE ONLY post
    ADD CONSTRAINT post_event_id_fk FOREIGN KEY (event_id) REFERENCES
event(id) ON DELETE CASCADE;

ALTER TABLE ONLY post
    ADD CONSTRAINT post_user_id_fk FOREIGN KEY (user_id) REFERENCES
"user"(id) ON DELETE CASCADE;

ALTER TABLE ONLY post
    ADD CONSTRAINT post_image_id_fk FOREIGN KEY (image_id) REFERENCES
image(id) ON DELETE SET NULL;

ALTER TABLE ONLY "user"
    ADD CONSTRAINT user_city_id_fk FOREIGN KEY (city_id) REFERENCES
city(id) ON DELETE SET NULL;
```

**GRUPO1765, 21/03/2018**

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