**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(id, username **UK NN**, password **NN**, email **UK NN**, regist\_date **NN DF** Today, first\_name **NN**, last\_name **NN**, image\_path, city\_id → City **NN**) |
| R02 | Event(id, name **NN**, date **NN** date > Today, description **NN**, owner\_id → User **NN**, localization\_id → Localization **NN**, image\_id → Image,  type **NN CK** type IN Types\_of\_event, category **NN CK** category IN Categories) |
| R03 | Done(event\_id → Event, rating **NN CK** rating > 0 AND rating < = 5) |
| R04 | Not\_done(event\_id → Event) |
| R05 | Paticipant(id, (user\_id → User**,** event\_id → Event) **UK** **NN**) |
| R06 | Owner(id (user\_id → User**,** event\_id → Event) **UK NN**) |
| R07 | Image(id, path **NN UK**) |
| R08 | Localization(id, name, address **NN**, latitude, longitude, city\_id → City **NN**) |
| R09 | City(id, name **UK NN**, country\_id → Country **NN**) |
| R10 | Country(id, name **UK NN**) |
| R11 | Post(id, description **NN**, date **NN DF** Today, event\_id → Event **NN**,  user\_id → User **NN**, image\_id → Image) |
| R12 | Poll(id, post\_id → Post **NN**) |
| R13 | Option(id, description **NN**, poll\_id → Poll **NN**) |
| R14 | Admin(id, username **UK NN**, password **NN**, email **UK NN**) |
| R15 | Friend\_request(id, answer, sender\_id → User **NN,** receiver\_id → User **NN**) |
| R16 | Friend\_activity(id, sender\_id → User **NN,** receiver\_id → User **NN**,  event\_id → Event **NN**) |
| R17 | Event\_invite(id, answer, event\_id → Not\_done **NN**, owner\_id → User **NN**,  receiver\_id → User **NN**) |

**2. Domínios**

|  |  |
| --- | --- |
| Today | DATE DEFAULT CURRENT\_DATE |
| Types\_of\_event | ENUM(‘Public’, Private’) |
| Categories | ENUM(‘Music’, ‘Sports’, ‘Entertainment’, ‘Educational’, ‘Business’, ‘Other’) |

**3. Dependências funcionais e validação do esquema**

|  |  |
| --- | --- |
| **Table R01** (User) | |
| **Keys:** {id}, {username}, {email} | |
| **Dependências Funcionais** | |
| FD0101 | {id} → {username, email, password, regist\_date, first\_name, last\_name, image, city} |
| FD0102 | {username} → {id, email, password, regist\_date, first\_name, last\_name, image, city} |
| FD0103 | {email} → {id, username, password, regist\_date, first\_name, last\_name, image, city} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R02** (Event) | |
| **Keys:** {id} | |
| **Dependências Funcionais** | |
| FD0201 | {id} → {name, date, type, category, description, localization, owner, image\_id} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R03** (Done) | |
| **Keys:** {event\_id} | |
| **Dependências Funcionais** | |
| FD0301 | {event\_id} → {rating} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R04** (Not\_done) | |
| **Keys:** {event\_id} | |
| **Dependências Funcionais** | |
| (none) | |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R05** (Participant) | |
| **Keys:** {user\_id, event\_id} | |
| **Dependências Funcionais** | |
| (none) | |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R06** (Owner) | |
| **Keys:** {user\_id, event\_id} | |
| **Dependências Funcionais** | |
| (none) | |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R07** (Image) | |
| **Keys:** {id} | |
| **Dependências Funcionais** | |
| FD0701 | {id} → {path} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R08** (Localization) | |
| **Keys:** {id} | |
| **Dependências Funcionais** | |
| FD0801 | {id} → {name, address, latitude, longitude, city\_id} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R09** (City) | |
| **Keys:** {id} | |
| **Dependências Funcionais** | |
| FD0901 | {id} → {name, country\_id} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R10** (Country) | |
| **Keys:** {id} | |
| **Dependências Funcionais** | |
| FD1001 | {id} → {name} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R11** (Post) | |
| **Keys:** {id} | |
| **Dependências Funcionais** | |
| FD1101 | {id} → {description, date, event, user\_id, image} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R12** (Poll) | |
| **Keys:** {id} | |
| **Dependências Funcionais** | |
| FD1201 | {id} → {post} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R13** (Option) | |
| **Keys:** {id} | |
| **Dependências Funcionais** | |
| FD1301 | {id} → {description, poll} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R14** (Admin) | |
| **Keys:** {id}, {username}, {email} | |
| **Dependências Funcionais** | |
| FD1401 | {id} → {username, email, password} |
| FD1402 | {username} → {id, email, password} |
| FD1403 | {email} → {id, username, password} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R15** (Friend\_request) | |
| **Keys:** {id} | |
| **Dependências Funcionais** | |
| FD1501 | {id} → {answer, sender\_id, receiver\_id} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R16** (Friend\_activity) | |
| **Keys:** {id} | |
| **Dependências Funcionais** | |
| FD1601 | {id} → {sender\_id, receiver\_id, event\_id} |
| **Forma Normal** | BCNF |

|  |  |
| --- | --- |
| **Table R17** (Event\_invite) | |
| **Keys:** {id} | |
| **Dependências Funcionais** | |
| FD1701 | {id} → {answer, event\_id, owner\_id, receiver\_id } |
| **Forma Normal** | BCNF |

**4. SQL Código**

[Sql code](https://github.com/Quaresma1997/lbaw1765/blob/gh-pages/lbaw1765_db.sql)

--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)