A5: Relational schema, validation and schema refinement

Our project, answerly, is a web application for collaborative questions and answers.

This artifact contains the Relational Schema obtained by mapping from the Conceptual Data Model. The Relational Schema includes the relation schema, attributes, domains, primary keys, foreign keys and other integrity rules: UNIQUE, DEFAULT, NOT NULL, CHECK

1. Relational Schema

Relation Reference	Relation Compact Notation
R01	user(**id**, first_name *NN*, last_name *NN*, email *UK NN*, bio, username *UK NN*, password *NN*, score *NN DF 0*)
R02	label(**id**, name *NN*)
R03	notification(**id**, content *NN*, date *DF Now NN*, viewed *DF False NN*, user_id → user *NN*)
R04	user_management(**id**, status *NN DF user CK status IN UserStatus*, date_last_changed *NN DF Now*, user_id → user *UK NN*)
R05	vote(**id**, vote *NN*, user_id → user *NN*, question_id → question, answer_id → answer *CK question_id = NN XOR answer_id = NN*)
R06	question(**id**, user_id → user *NN*, title *NN*, description *NN*, nr_likes *NN DF 0*, nr_dislikes *NN DF 0 CK nr_likes >= 0 AND nr_dislikes >= 0*, question_date *NN DF Now*)
R07	answer(**id**, user_id → user *NN*, question_id → question *NN*, answer_date *NN DF Now*, content *NN*, nr_likes *NN DF 0*, nr_dislikes *NN DF 0 CK nr_likes >= 0 AND nr_dislikes >= 0*, marked_answer *NN DF FALSE*)
R08	comment(**id**, user_id → user *NN*, question_id→question, answer_id→answer *CK question_id = NN XOR answer_id = NN*, content *NN*, comment_date *NN DF Now*)
R09	report(**id**, reporter_id→user *NN*, user_id→user, question_id→question, answer_id→answer, comment_id→comment, description *CK user_id = NN XOR question_id = NN XOR answer_id = NN XOR comment_id = NN*)
R10	report_status(**id**, report_id → report *ODC NN*, state *NN DF unresolved CK state IN ReportStates*, comment, responsible_user → user *NN ODC*)
R11	question_following(**user_id** → user, **question_id** → question)
R12	label_following(**user_id** → user, **label_id** → label)
R13	question_label(**question_id** → question, **label_id** → label)

- UK means UNIQUE KEY

- NN means NOT NULL
- DF means DEFAULT
- CK means CHECK
- ODC means ON DELETED CASCADE

2. Domains

Specification of additional domains:

Domain Name	Domain Specification
Now	DATE DEFAULT CURRENT_TIMESTAMP_
ReportStates	ENUM ('unresolved', 'reviewing', 'resolved')
UserStatus	ENUM ('user', 'moderator', 'administrator', 'banned')

3. Functional Dependencies and schema validation

In the following tables, all relations are in the Boyce-Codd Normal Form, since for each non trivial functional dependency A → B, A is a (super)key of the relation.

Table R01 (user)	
Keys: {id}, {username}, {email}	
Functional Dependencies	
FD0101	{id} → {first_name, last_name, email, bio, username, password, score}
FD0102	{username} → {id, first_name, last_name, email, bio, password, score}
FD0103	{email} → {id, first_name, last_name, bio, username, password, score}
NORMAL FORM	BCNF

Table R02 (label)	
Keys: {id}	
Functional Dependencies	
FD0201	{id} → {name}
NORMAL FORM	BCNF

Table R03 (notification)	
Keys: {id}	
Functional Dependencies	
FD0301	{id} → {content, date, viewed, user_id}
NORMAL FORM	BCNF

Table R04 (user_management)	
Keys: {id}	
Functional Dependencies	
FD0401	{id} → {status, date_last_changed, user_id}
NORMAL FORM	BCNF

Table R05 (vote)	
Keys: {id}	
Functional Dependencies	
FD0501	{id} → {vote, user_id, question_id, answer_id}
NORMAL FORM	BCNF

Table R06 (question)	
Keys: {id}	
Functional Dependencies	
FD0601	{id} → {user_id, title, description, nr_likes, nr_dislikes, question_date}
NORMAL FORM	BCNF

Table R07 (answer)	
Keys: {id}	
Functional Dependencies	
FD0701	{id} → {user_id, question_id, answer_date, content, nr_likes, nr_dislikes, marked_answer}
NORMAL FORM	BCNF

Table R08 (comment)	
Keys: {id}	
Functional Dependencies	
FD0801	{id} → {user_id, question_id, answer_id, content, comment_date}
NORMAL FORM	BCNF

Table R09 (report)
Keys: {id}

Table R09 (report)	
Functional Dependencies	
FD0901	{id} → {reporter_id, user_id, question_id, answer_id, comment_id, description}
NORMAL FORM	BCNF

Table R10 (report_status)	
Keys: {id}	
Functional Dependencies	
FD1001	{id} → {report_id, state, comment, responsible_user}
NORMAL FORM	BCNF

Table R11 (question_following)	
Keys: {user_id, question_id}	
Functional Dependencies	
(none)	
NORMAL FORM	BCNF

Table R12 (label_following)	
Keys: {user_id, label_id}	
Functional Dependencies	
(none)	
NORMAL FORM	BCNF

Table R13 (question_label)	
Keys: {question_id, label_id}	
Functional Dependencies	
(none)	
NORMAL FORM	BCNF

4. SQL Code

```
-----------------------------------------
-- Drop old schmema
```

```
-----------------------------------------

DROP TABLE IF EXISTS "user" CASCADE;
DROP TABLE IF EXISTS label CASCADE;
DROP TABLE IF EXISTS notification CASCADE;
DROP TABLE IF EXISTS user_management CASCADE;
DROP TABLE IF EXISTS question CASCADE;
DROP TABLE IF EXISTS answer CASCADE;
DROP TABLE IF EXISTS comment CASCADE;
DROP TABLE IF EXISTS vote CASCADE;
DROP TABLE IF EXISTS report CASCADE;
DROP TABLE IF EXISTS report_status CASCADE;
DROP TABLE IF EXISTS question_following CASCADE;
DROP TABLE IF EXISTS label_following CASCADE;
DROP TABLE IF EXISTS question_label CASCADE;

-----------------------------------------
-- Tables
-----------------------------------------

-- Table: user
CREATE TABLE "user" (
    id                SERIAL          PRIMARY KEY,
    first_name        TEXT            NOT NULL,
    last_name         TEXT            NOT NULL,
    email             TEXT            NOT NULL UNIQUE,
    bio               TEXT,
    username          TEXT            NOT NULL UNIQUE,
    password          TEXT            NOT NULL,
    score             INTEGER         NOT NULL DEFAULT 0
);

-- Table: label
CREATE TABLE label (
    id                SERIAL          PRIMARY KEY,
    name              TEXT            NOT NULL
);

-- Table: notification
CREATE TABLE notification (
    id                SERIAL          PRIMARY KEY,
    content           TEXT            NOT NULL,
    date              DATE            DEFAULT 'Now' NOT NULL,
    viewed            BOOLEAN         DEFAULT FALSE NOT NULL,
    user_id           INTEGER         REFERENCES "user" (id) NOT NULL
);

-- Table: user_management
CREATE TABLE user_management (
    id                    SERIAL          PRIMARY KEY,
    status                TEXT            DEFAULT 'user' NOT NULL,
    date_last_changed     DATE            DEFAULT 'Now' NOT NULL,
    user_id               INTEGER         REFERENCES "user" (id) NOT NULL
UNIQUE,
    CHECK (
        status = 'user' OR status = 'moderator' OR status = 'administrator'
OR status = 'banned'
    )
);

-- Table: question
CREATE TABLE question (
    id                SERIAL            PRIMARY KEY,
    user_id           INTEGER           REFERENCES "user" (id) NOT NULL,
    title             TEXT              NOT NULL,
    description       TEXT              NOT NULL,
    nr_likes          INTEGER           DEFAULT 0 NOT NULL,
    nr_dislikes       INTEGER           DEFAULT 0 NOT NULL,
    question_date     DATE              DEFAULT 'Now' NOT NULL,
    CHECK (
        nr_likes >= 0 AND nr_dislikes >= 0
    )
);

-- Table: answer
CREATE TABLE answer (
    id                 SERIAL            PRIMARY KEY,
    user_id            INTEGER           REFERENCES "user" (id) NOT NULL,
    question_id        INTEGER           REFERENCES "question" (id) ON DELETE
CASCADE NOT NULL,
    answer_date        DATE              DEFAULT 'Now' NOT NULL,
    content            TEXT              NOT NULL,
    nr_likes           INTEGER           DEFAULT 0 NOT NULL,
    nr_dislikes        INTEGER           DEFAULT 0 NOT NULL,
    marked_answer      BOOLEAN           DEFAULT FALSE NOT NULL,
    CHECK (
        nr_likes >= 0 AND nr_dislikes >= 0
    )
);

-- Table: comment
CREATE TABLE comment (
    id                 SERIAL            PRIMARY KEY,
    user_id            INTEGER           REFERENCES "user" (id) NOT NULL,
    question_id        INTEGER           REFERENCES "question" (id) ON DELETE
CASCADE,
    answer_id          INTEGER           REFERENCES "answer" (id) ON DELETE
CASCADE,
    content            TEXT              NOT NULL,
    comment_date       DATE              DEFAULT 'Now' NOT NULL,
    CHECK (
        (question_id IS NOT NULL AND answer_id IS NULL) OR
        (question_id IS NULL AND answer_id IS NOT NULL)
    )
);

-- Table: vote
CREATE TABLE vote (
    id                 SERIAL          PRIMARY KEY,
    "vote"             BOOLEAN         NOT NULL,
    user_id            INTEGER         REFERENCES "user" (id) NOT NULL,
    question_id        INTEGER         REFERENCES "question" (id) ON DELETE
CASCADE,
    answer_id          INTEGER         REFERENCES "answer" (id) ON DELETE
CASCADE,
    CHECK (
        (question_id IS NOT NULL AND answer_id IS NULL) OR
        (question_id IS NULL AND answer_id IS NOT NULL)
    )
);

-- Table: report
CREATE TABLE report (
    id                 SERIAL          PRIMARY KEY,
    reporter_id        INTEGER         REFERENCES "user" (id) NOT NULL,
    user_id            INTEGER         REFERENCES "user" (id),
    question_id        INTEGER         REFERENCES "question" (id) ON DELETE
CASCADE,
    answer_id          INTEGER         REFERENCES "answer" (id) ON DELETE
CASCADE,
    comment_id         INTEGER         REFERENCES "comment" (id) ON DELETE
CASCADE,
    report_date        DATE            DEFAULT 'Now' NOT NULL,
    description        TEXT            NOT NULL,
    CHECK(
        (user_id IS NOT NULL AND question_id IS NULL AND answer_id IS NULL
AND comment_id IS NULL) OR
        (user_id IS NULL AND question_id IS NOT NULL AND answer_id IS NULL
AND comment_id IS NULL) OR
        (user_id IS NULL AND question_id IS NULL AND answer_id IS NOT NULL
AND comment_id IS NULL) OR
        (user_id IS NULL AND question_id IS NULL AND answer_id IS NULL AND
comment_id IS NOT NULL)
    )
);

-- Table: report_status
CREATE TABLE report_status (
    id                 SERIAL          PRIMARY KEY,
    report_id          INTEGER         REFERENCES "report" (id) ON DELETE
CASCADE NOT NULL,
    state              TEXT            DEFAULT 'unresolved' NOT NULL,
    comment            TEXT,
    responsible_user INTEGER           REFERENCES "user" (id) ON DELETE
CASCADE NOT NULL,
    CHECK (
        state = 'unresolved' OR state = 'reviewing' OR state = 'resolved'
    )
);

-- Table: question_following
CREATE TABLE question_following (
    user_id           INTEGER          REFERENCES "user" (id) NOT NULL,
    question_id       INTEGER          REFERENCES "question" (id) ON DELETE
CASCADE NOT NULL,
    PRIMARY KEY (user_id, question_id)
);

-- Table: label_following
CREATE TABLE label_following (
    user_id           INTEGER          REFERENCES "user" (id) NOT NULL,
    label_id          INTEGER          REFERENCES "label" (id) NOT NULL,
    PRIMARY KEY (user_id, label_id)
);

-- Table: question_label
CREATE TABLE question_label (
    question_id       INTEGER          REFERENCES "question" (id) ON DELETE
CASCADE NOT NULL,
    label_id          INTEGER          REFERENCES "label" (id) NOT NULL,
    PRIMARY KEY (question_id, label_id)
);
```

Revision history

1. First submission (23/03/2020).
2. Deleted Administrator and Moderator tables. Changed all id's to "id" and other minor changes (28/03/2020).
3. Tested sql code, fixed some errors (29/03/2020).
4. Changed some relations (09/04/2020).

GROUP2064, 09/04/2020

- [Editor] Antonio Pedro Reis Ribeiro Sousa Dantas, up201703878@fe.up.pt
- Eduardo João Santana Macedo, up201703658@fe.up.pt
- Nuno Miguel Teixeira Cardoso, up201706162@fe.up.pt
- Paulo Roberto Dias Mourato, up201705616@fe.up.pt