A5: Relational Schema, validation and schema refinement

SegFault is a collaborative platform for programmers to learn, discuss different approaches, present ideas and share knowledge in a Q&A style. To this end, the following sections provide detailed insight into the project's relational schema, domains, functional dependencies and schema validation.

1. Relational Schema

> The Relational Schema includes the relation schemas, attributes, domains, primary keys, foreign keys and other integrity rules: UNIQUE, DEFAULT, NOT NULL, CHECK. Relation schemas are specified in the compact notation:

Relation reference	Relation Compact Notation
R01	Category(**ID**, name NN, description, numPosts NN CK numPosts >= 0)
R02	QuestionCategory(**questionID** → Question, **categoryID** → Category)
R03	Question(**ID** → Commentable, title NN, correctAnswer → Answer UK)
R04	Answer(**ID** → Commentable, questionID → Question NN)
R05	Commentable(**ID** → Message)
R06	Comment(**ID** → Message, commentableID → Commentable NN)
R07	Message(**ID**, creationDate NN DF Today, score NN DF 0, numReports NN DF 0, isBanned NN DF False)
R08	MessageContent(**ID**, content NN, messageID → Message NN)
R09	TimeStamp(**messageContentID** → MessageContent, userID → User NN, timeStamp NN DF Today)
R10	ModEdition(**messageContentID** → MessageContent, moderatorID → Moderator NN, timeStamp NN DF Today)
R11	Vote(**messageID** → Message, **userID** → User, positive NN)
R12	User(**ID**, userName UK NN, email UK NN, passwordHash NN, bio, reputation NN)
R13	Moderator(**ID** → User)
R14	Notification(**ID**, description NN, date NN, read NN, userID → User)
R15	CommentableNotification(**ID** → Notification, commentableID → Commentable NN)
R16	BadgeNotification(**ID** → Notification, badgeID → Badge NN)

Relation reference	Relation Compact Notation
R17	BadgeAttainment(**userID** →User, **badgeID** → Badge, attainmentDate NN)
R18	Badge(**ID**, description NN)
R19	ModeratorBadge(**ID** → Badge)
R20	TrustedBadge(**ID** → Badge)

2. Domains

The specification of additional domains can also be made in a compact form, using the notation:

Domain Name	Domain Specification
Today	DATE DEFAULT CURRENT_DATE

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. Should it be necessary, in case the scheme is not in the Boyce–Codd Normal Form (BCNF), the relational schema is refined using normalization.

Table R01	(Category)
Keys: {ID}	
Functional Dependencies	
FD0101	{ID} → {attribute, name, description, numPosts}
Normal Form	BCNF

Table R02	(QuestionCategory)
Keys: {questionID, categoryID}	
Functional Dependencies	
(none)	
Normal Form	BCNF

Table R03	(Question)
Keys: {ID}	

Table R03	(Question)
Functional Dependencies	
FD0301	{ID} → {title, correctAnswer, numPosts}
FD0302	{correctAnswer} → {ID, title, numPosts}
Normal Form	BCNF

Table R04	(Answer)
Keys: {ID}	
Functional Dependencies	
FD0401	{ID} → {questionID}
Normal Form	BCNF

Table R05	(Commentable)
Keys: {ID}	
Functional Dependencies	
(none)	
Normal Form	BCNF

Table R06	(Comment)
Keys: {ID}	
Functional Dependencies	
FD0601	{ID} → {commentableID}
Normal Form	BCNF

Table R07	(Message)
Keys: {ID}	
Functional Dependencies	
FD0701	{ID} → {creationDate, score, numReports, isBanned}
Normal Form	BCNF

Table R08	(MessageContent)
Keys: {ID}	
Functional Dependencies	
FD0801	{ID} → {content, messageID}
FD0802	{messageID} → {content, ID}

Table R08	(MessageContent)
Normal Form	BCNF

Table R09	(TimeStamp)
Keys: {messageContentID}	
Functional Dependencies	
FD0901	{messageContentID} → {userID, timestamp}
Normal Form	BCNF

Table R10	(ModEdition)
Keys: {messageContentID}	
Functional Dependencies	
FD1001	{messageContentID} → {moderatorID, timestamp}
Normal Form	BCNF

Table R11	(Vote)
Keys: {messageID, userID}	
Functional Dependencies	
FD1101	{messageID, userID} → {positive}
Normal Form	BCNF

Table R12	(User)
Keys: {ID}	
Functional Dependencies	
FD1201	{ID} → {username, email, passwordHash, bio, reputation}
FD1202	{username} → {ID, email, passwordHash, bio, reputation}
FD1203	{email} → {username, ID, passwordHash, bio, reputation}
Normal Form	BCNF

Table R13	(Moderator)
Keys: {ID}	
Functional Dependencies	
(none)	
Normal Form	BCNF

Table R14	(Notification)
Keys: {ID}	
Functional Dependencies	
FD1401	{ID} → {description, date, read, userID}
Normal Form	BCNF

Table R15	(CommentableNotification)
Keys: {ID}	
Functional Dependencies	
FD1501	{ID} → {commentableID}
Normal Form	BCNF

Table R16	(BadgeNotification)
Keys: {ID}	
Functional Dependencies	
FD1601	{ID} → {badgeID}
Normal Form	BCNF

Table R17	(BadgeAttainment)
Keys: {userID, badgeID}	
Functional Dependencies	
FD1701	{userID, badgeID} → {attainmentDate}
Normal Form	BCNF

Table R18	(Badge)
Keys: {ID}	
Functional Dependencies	
FD1801	{ID} → {description}
Normal Form	BCNF

Table R19	(ModeratorBadge)
Keys: {ID}	
Functional Dependencies	
(none)	
Normal Form	BCNF

Table R20	(TrustedBadge)
Keys: {ID}	
Functional Dependencies	
(none)	
Normal Form	BCNF

If necessary, description of the changes necessary to convert the schema to BCNF. Justification of the BCNF.

4. SQL Code

```
-- Tables
DROP TABLE IF EXISTS Category CASCADE;
DROP TABLE IF EXISTS QuestionCategory CASCADE;
DROP TABLE IF EXISTS Question CASCADE;
DROP TABLE IF EXISTS Answer CASCADE;
DROP TABLE IF EXISTS Commentable CASCADE;
DROP TABLE IF EXISTS Comment CASCADE;
DROP TABLE IF EXISTS Message CASCADE;
DROP TABLE IF EXISTS MessageContent CASCADE;
DROP TABLE IF EXISTS "TimeStamp" CASCADE;
DROP TABLE IF EXISTS ModEdition CASCADE;
DROP TABLE IF EXISTS Vote CASCADE;
DROP TABLE IF EXISTS "User" CASCADE;
DROP TABLE IF EXISTS Moderator CASCADE;
DROP TABLE IF EXISTS Notification CASCADE;
DROP TABLE IF EXISTS CommentableNotification CASCADE;
DROP TABLE IF EXISTS BadgeNotification CASCADE;
DROP TABLE IF EXISTS BadgeAttainment CASCADE;
DROP TABLE IF EXISTS Badge CASCADE;
DROP TABLE IF EXISTS ModeratorBadge CASCADE;
DROP TABLE IF EXISTS TrustedBadge CASCADE;

CREATE TABLE Category (
    id SERIAL,
    name TEXT NOT NULL,
    description TEXT,
    num_posts INTEGER DEFAULT 0 NOT NULL
);

CREATE TABLE QuestionCategory (
    question_id BIGINT NOT NULL,
    category_id INTEGER NOT NULL
);
```

```sql
CREATE TABLE Question (
    id BIGINT NOT NULL,
    title TEXT NOT NULL,
    correct_answer BIGINT
);

CREATE TABLE Answer (
    id BIGINT NOT NULL,
    question_id BIGINT NOT NULL
);

CREATE TABLE Commentable (
    id BIGINT NOT NULL
);

CREATE TABLE Comment (
    id BIGINT NOT NULL,
    commentable_id BIGINT NOT NULL
);

CREATE TABLE Message (
    id BIGSERIAL,
    creation_date TIMESTAMP WITH TIME ZONE DEFAULT now() NOT NULL,
    score INTEGER DEFAULT 0 NOT NULL,
    num_reports SMALLINT DEFAULT 0 NOT NULL,
    is_banned BOOLEAN DEFAULT FALSE
);

CREATE TABLE MessageContent (
    id BIGSERIAL,
    content TEXT NOT NULL,
    message_id BIGINT
);

CREATE TABLE "TimeStamp" (
    message_content_id BIGINT NOT NULL,
    user_id BIGINT NOT NULL,
    creation_time TIMESTAMP WITH TIME ZONE DEFAULT now() NOT NULL
);

CREATE TABLE ModeratorEdition (
    message_content_id BIGINT NOT NULL,
    moderator_id BIGINT NOT NULL,
    creation_time TIMESTAMP WITH TIME ZONE DEFAULT now() NOT NULL
);
```

```
CREATE TABLE Vote (
    message_id BIGINT NOT NULL,
    user_id BIGINT NOT NULL,
    positive BOOLEAN NOT NULL
);

CREATE TABLE "User" (
    id BIGSERIAL,
    username TEXT NOT NULL,
    email TEXT NOT NULL,
    password_hash TEXT NOT NULL,
    biography TEXT,
    reputation SMALLINT NOT NULL
);

CREATE TABLE Moderator (
    id BIGINT NOT NULL
);

CREATE TABLE Notification (
    id BIGSERIAL,
    description TEXT NOT NULL,
    "date" TIMESTAMP WITH TIME ZONE DEFAULT now() NOT NULL,
    read BOOLEAN NOT NULL,
    user_id BIGINT NOT NULL
);

CREATE TABLE CommentableNotification (
    id BIGINT NOT NULL,
    commentable_id BIGINT NOT NULL
);

CREATE TABLE BadgeNotification (
    id BIGINT NOT NULL,
    badge_id BIGINT NOT NULL
);

CREATE TABLE BadgeAttainment (
    user_id BIGINT NOT NULL,
    badge_id SMALLINT NOT NULL,
    attainment_date TIMESTAMP WITH TIME ZONE DEFAULT now() NOT NULL
);

CREATE TABLE Badge (
    id SERIAL,
    description TEXT NOT NULL
);

CREATE TABLE ModeratorBadge (
    id INTEGER NOT NULL
);

CREATE TABLE TrustedBadge (
    id INTEGER NOT NULL
);


-- Primary Keys
ALTER TABLE ONLY Category
  ADD CONSTRAINT category_pkey PRIMARY KEY (id);

ALTER TABLE ONLY QuestionCategory
  ADD CONSTRAINT question_category_pkey PRIMARY KEY (question_id, category_id);

ALTER TABLE ONLY Question
  ADD CONSTRAINT question_pkey PRIMARY KEY (id);

ALTER TABLE ONLY Answer
  ADD CONSTRAINT answer_pkey PRIMARY KEY (id);

ALTER TABLE ONLY Commentable
  ADD CONSTRAINT commentable_pkey PRIMARY KEY (id);

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

ALTER TABLE ONLY Message
  ADD CONSTRAINT message_pkey PRIMARY KEY (id);

ALTER TABLE ONLY MessageContent
  ADD CONSTRAINT message_content_pkey PRIMARY KEY (id);

ALTER TABLE ONLY "TimeStamp"
  ADD CONSTRAINT timestamp_pkey PRIMARY KEY (message_content_id);

ALTER TABLE ONLY ModeratorEdition
  ADD CONSTRAINT moderator_edition_pkey PRIMARY KEY (message_content_id);

ALTER TABLE ONLY Vote
  ADD CONSTRAINT vote_pkey PRIMARY KEY (message_id, user_id);
```

```sql
ALTER TABLE ONLY "User"
  ADD CONSTRAINT user_pkey PRIMARY KEY (id);

ALTER TABLE ONLY Moderator
  ADD CONSTRAINT moderator_pkey PRIMARY KEY (id);

ALTER TABLE ONLY Notification
  ADD CONSTRAINT notification_pkey PRIMARY KEY (id);

ALTER TABLE ONLY CommentableNotification
  ADD CONSTRAINT commentable_notification_pkey PRIMARY KEY (id);

ALTER TABLE ONLY BadgeNotification
  ADD CONSTRAINT badge_notification_pkey PRIMARY KEY (id);

ALTER TABLE ONLY BadgeAttainment
  ADD CONSTRAINT badge_attainment_pkey PRIMARY KEY (user_id, badge_id);

ALTER TABLE ONLY Badge
  ADD CONSTRAINT badge_pkey PRIMARY KEY (id);

ALTER TABLE ONLY ModeratorBadge
  ADD CONSTRAINT moderator_badge_pkey PRIMARY KEY (id);

ALTER TABLE ONLY TrustedBadge
  ADD CONSTRAINT trusted_badge_pkey PRIMARY KEY (id);


-- Unique
ALTER TABLE ONLY Question
  ADD CONSTRAINT correct_answer_key UNIQUE (correct_answer);

ALTER TABLE ONLY "User"
  ADD CONSTRAINT user_email_key UNIQUE (email);

ALTER TABLE ONLY "User"
  ADD CONSTRAINT username_key UNIQUE (username);


-- Foreign Keys
ALTER TABLE ONLY QuestionCategory
  ADD CONSTRAINT question_category_question_fkey FOREIGN KEY (question_id) REFERENCES Questi

ALTER TABLE ONLY QuestionCategory
  ADD CONSTRAINT question_category_category_fkey FOREIGN KEY (category_id) REFERENCES Catego
```

```
ALTER TABLE ONLY Question
  ADD CONSTRAINT question_fkey FOREIGN KEY (id) REFERENCES Commentable(id) ON UPDATE CASCADE

ALTER TABLE ONLY Question
  ADD CONSTRAINT question_correct_fkey FOREIGN KEY (correct_answer) REFERENCES Answer(id) ON

ALTER TABLE ONLY Answer
  ADD CONSTRAINT answer_fkey FOREIGN KEY (id) REFERENCES Commentable(id) ON UPDATE CASCADE;

ALTER TABLE ONLY Answer
  ADD CONSTRAINT answer_question_fkey FOREIGN KEY (question_id) REFERENCES Question(id) ON U

ALTER TABLE ONLY Commentable
  ADD CONSTRAINT commentable_message_fkey FOREIGN KEY (id) REFERENCES Message(id) ON UPDATE

ALTER TABLE ONLY Comment
  ADD CONSTRAINT comment_fkey FOREIGN KEY (id) REFERENCES Message(id) ON UPDATE CASCADE;

ALTER TABLE ONLY Comment
  ADD CONSTRAINT comment_commentable_fkey FOREIGN KEY (commentable_id) REFERENCES Commentabl

ALTER TABLE ONLY MessageContent
  ADD CONSTRAINT message_content_message_fkey FOREIGN KEY (message_id) REFERENCES Message(i

ALTER TABLE ONLY "TimeStamp"
  ADD CONSTRAINT time_stamp_message_content_fkey FOREIGN KEY (message_content_id) REFERENCES

ALTER TABLE ONLY "TimeStamp"
  ADD CONSTRAINT time_stamp_user_fkey FOREIGN KEY (user_id) REFERENCES "User"(id) ON UPDATE

ALTER TABLE ONLY ModeratorEdition
  ADD CONSTRAINT moderator_edition_message_content_fkey FOREIGN KEY (message_content_id) REF

ALTER TABLE ONLY ModeratorEdition
  ADD CONSTRAINT moderator_edition_user_fkey FOREIGN KEY (moderator_id) REFERENCES Moderator

ALTER TABLE ONLY Vote
  ADD CONSTRAINT vote_message_fkey FOREIGN KEY (message_id) REFERENCES Message(id) ON UPDATE

ALTER TABLE ONLY Vote
  ADD CONSTRAINT vote_user_fkey FOREIGN KEY (user_id) REFERENCES "User"(id) ON UPDATE CASCAD

ALTER TABLE ONLY Moderator
  ADD CONSTRAINT moderator_user_fkey FOREIGN KEY (id) REFERENCES "User"(id) ON UPDATE CASCAD

ALTER TABLE ONLY Notification
```

```
  ADD CONSTRAINT notification_user_fkey FOREIGN KEY (user_id) REFERENCES "User"(id) ON UPDAT

ALTER TABLE ONLY CommentableNotification
  ADD CONSTRAINT commentable_notification_fkey FOREIGN KEY (id) REFERENCES Notification(id)

ALTER TABLE ONLY CommentableNotification
  ADD CONSTRAINT commentable_notification_commentable_fkey FOREIGN KEY (id) REFERENCES Comme

ALTER TABLE ONLY BadgeNotification
  ADD CONSTRAINT badge_notification_fkey FOREIGN KEY (id) REFERENCES Notification(id) ON UPD

ALTER TABLE ONLY BadgeNotification
  ADD CONSTRAINT badge_notification_badge_fkey FOREIGN KEY (badge_id) REFERENCES Badge(id) O

ALTER TABLE ONLY BadgeAttainment
  ADD CONSTRAINT badge_attainment_user_fkey FOREIGN KEY (user_id) REFERENCES "User"(id) ON U

ALTER TABLE ONLY BadgeAttainment
  ADD CONSTRAINT badge_attainment_badge_fkey FOREIGN KEY (badge_id) REFERENCES Badge(id) ON

ALTER TABLE ONLY ModeratorBadge
  ADD CONSTRAINT moderator_badge_fkey FOREIGN KEY (id) REFERENCES Badge(id) ON UPDATE CASCAD

ALTER TABLE ONLY TrustedBadge
  ADD CONSTRAINT trusted_badge_fkey FOREIGN KEY (id) REFERENCES Badge(id) ON UPDATE CASCADE;
```

Revision history

GROUP1763, 15/03/2018

André Cruz, up201503776@fe.up.pt
Daniel Marques, up201503822@fe.up.pt
Edgar Carneiro, up201503784@fe.up.pt
João Carvalho, up201504875@fe.up.pt