

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 \rightarrow Question, categoryID \rightarrow Category)
R03	Question(ID \rightarrow Commentable, title NN, correctAnswer \rightarrow Answer UK)
R04	Answer(ID \rightarrow Commentable, questionID \rightarrow Question NN)
R05	Commentable(ID \rightarrow Message)
R06	Comment(ID \rightarrow Message, commentableID \rightarrow 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 \rightarrow Message NN)
R09	TimeStamp(messageContentID \rightarrow MessageContent, userID \rightarrow User NN, timeStamp NN DF Today)
R10	Vote(messageID \rightarrow Message, userID \rightarrow User, positive NN)
R11	User(ID , userName UK NN, email UK NN, passwordHash NN, bio, reputation NN)
R12	Moderator(ID \rightarrow User)
R13	Notification(ID , description NN, date NN, read NN, userID \rightarrow User)
R14	CommentableNotification(ID \rightarrow Notification, commentableID \rightarrow Commentable NN)
R15	BadgeNotification(ID \rightarrow Notification, badgeID \rightarrow Badge NN)
R16	BadgeAttainment(userID \rightarrow User, badgeID \rightarrow Badge, attainmentDate NN)

Relation reference	Relation Compact Notation
R17	Badge(ID , description NN)
R18	ModeratorBadge(ID \rightarrow Badge)
R19	TrustedBadge(ID \rightarrow 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} \rightarrow {attribute, name, description, numPosts}
Normal Form	BCNF

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

Table R03	(Question)
Keys: {id}	
Functional Dependencies	
FD0301	{id} \rightarrow {commentableID, title, correctAnswer, numPosts}

Table R03	(Question)
FD0302	$\{\text{commentableID}\} \rightarrow \{\text{id}, \text{title}, \text{correctAnswer}, \text{numPosts}\}$
FD0303	$\{\text{correctAnswer}\} \rightarrow \{\text{id}, \text{title}, \text{commentableID}, \text{numPosts}\}$
Normal Form	BCNF

Table R04	(Answer)
Keys: {id}	
Functional Dependencies	
FD0401	$\{\text{id}\} \rightarrow \{\text{commentableID}, \text{questionID}\}$
FD0402	$\{\text{commentableID}\} \rightarrow \{\text{id}, \text{questionID}\}$
Normal Form	BCNF

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

Table R06	(Comment)
Keys: {id}	
Functional Dependencies	
FD0601	$\{\text{id}\} \rightarrow \{\text{messageID}, \text{commentableID}\}$
FD0602	$\{\text{messageID}\} \rightarrow \{\text{id}, \text{commentableID}\}$
Normal Form	BCNF

Table R07	(Message)
Keys: {id}	
Functional Dependencies	
FD0701	$\{\text{id}\} \rightarrow \{\text{creationDate}, \text{score}, \text{numReports}, \text{isBanned}\}$
Normal Form	BCNF

Table R08	(MessageContent)
Keys: {id}	
Functional Dependencies	
FD0801	$\{\text{id}\} \rightarrow \{\text{content}, \text{messageID}\}$
FD0802	$\{\text{messageID}\} \rightarrow \{\text{content}, \text{id}\}$

Table R08	(MessageContent)
Normal Form	BCNF

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

Table R10	(Vote)
Keys: {messageID, userID}	
Functional Dependencies	
FD1001	{messageID, userID} → {positive}
Normal Form	BCNF

Table R11	(User)
Keys: {id}	
Functional Dependencies	
FD1101	{id} → {username, email, passwordHash, bio, reputation}
FD1102	{username} → {id, email, passwordHash, bio, reputation}
FD1103	{email} → {username, id, passwordHash, bio, reputation}
Normal Form	BCNF

Table R12	(Moderator)
Keys: {userID}	
Functional Dependencies	
(none)	
Normal Form	BCNF

Table R13	(Notification)
Keys: {id}	
Functional Dependencies	
FD1301	{id} → {description, date, read, userID}
Normal Form	BCNF

Table R14	(CommentableNotification)
Keys: {id}	
Functional Dependencies	
FD1401	$\{id\} \rightarrow \{notificationID, commentableID\}$
FD1402	$\{notificationID\} \rightarrow \{id, commentableID\}$
Normal Form	BCNF

Table R15	(BadgeNotification)
Keys: {id}	
Functional Dependencies	
FD1501	$\{id\} \rightarrow \{notificationID, badgeID\}$
FD1502	$\{notificationID\} \rightarrow \{id, badgeID\}$
Normal Form	BCNF

Table R16	(BadgeAttainment)
Keys: {userID, badgeID}	
Functional Dependencies	
FD1601	$\{userID, badgeID\} \rightarrow \{attainmentDate\}$
Normal Form	BCNF

Table R17	(Badge)
Keys: {id}	
Functional Dependencies	
FD1701	$\{id\} \rightarrow \{description\}$
Normal Form	BCNF

Table R18	(ModeratorBadge)
Keys: {badgeID}	
Functional Dependencies	
(none)	
Normal Form	BCNF

Table R19	(TrustedBadge)
Keys: {badgeID}	
Functional Dependencies	
(none)	

Table R19	(TrustedBadge)
Normal Form	BCNF

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

4. SQL Code

```
-- Tables

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
);

CREATE TABLE Question (
    id BIGSERIAL,
    commentable_id BIGINT NOT NULL,
    title TEXT NOT NULL,
    correct_answer BIGINT
);

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

CREATE TABLE Commentable (
    message_id BIGINT NOT NULL
);

CREATE TABLE Comment (
    id BIGSERIAL,
    message_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 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 (
    user_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 BIGSERIAL,
    notification_id BIGINT NOT NULL,
    commentable_id BIGINT NOT NULL
);

CREATE TABLE BadgeNotification (
    id BIGSERIAL,
    notification_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 (
    badge_id INTEGER NOT NULL
);

CREATE TABLE TrustedBadge (
    badge_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 (message_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 Vote
    ADD CONSTRAINT vote_pkey PRIMARY KEY (message_id, user_id);

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

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

ALTER TABLE ONLY TrustedBadge
    ADD CONSTRAINT trusted_badge_pkey PRIMARY KEY (badge_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 Question(id) ON UPDATE CASCADE;

ALTER TABLE ONLY QuestionCategory
    ADD CONSTRAINT question_category_category_fkey FOREIGN KEY (category_id) REFERENCES Category(id) ON UPDATE CASCADE;

ALTER TABLE ONLY Question
    ADD CONSTRAINT question_commentable_fkey FOREIGN KEY (commentable_id) REFERENCES Commentable(id) ON UPDATE CASCADE;

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

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

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

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

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

ALTER TABLE ONLY Comment
    ADD CONSTRAINT comment_commentable_fkey FOREIGN KEY (commentable_id) REFERENCES Commentable(id) ON UPDATE CASCADE;

ALTER TABLE ONLY MessageContent
    ADD CONSTRAINT message_content_message_fkey FOREIGN KEY (message_id) REFERENCES Message(id) ON UPDATE CASCADE;

ALTER TABLE ONLY "TimeStamp"
    ADD CONSTRAINT time_stamp_message_content_fkey FOREIGN KEY (message_content_id) REFERENCES MessageContent(id) ON UPDATE CASCADE;

```

```

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

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

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

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

ALTER TABLE ONLY Notification
ADD CONSTRAINT notification_user_fkey FOREIGN KEY (user_id) REFERENCES "User"(id) ON UPDATE CASCADE;

ALTER TABLE ONLY CommentableNotification
ADD CONSTRAINT commentable_notification_fkey FOREIGN KEY (notification_id) REFERENCES Notification(id) ON UPDATE CASCADE;

ALTER TABLE ONLY CommentableNotification
ADD CONSTRAINT commentable_notification_commentable_fkey FOREIGN KEY (commentable_id) REFERENCES Commentable(id) ON UPDATE CASCADE;

ALTER TABLE ONLY BadgeNotification
ADD CONSTRAINT badge_notification_fkey FOREIGN KEY (notification_id) REFERENCES Notification(id) ON UPDATE CASCADE;

ALTER TABLE ONLY BadgeNotification
ADD CONSTRAINT badge_notification_badge_fkey FOREIGN KEY (badge_id) REFERENCES Badge(id) ON UPDATE CASCADE;

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

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

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

ALTER TABLE ONLY TrustedBadge
ADD CONSTRAINT trusted_badge_fkey FOREIGN KEY (badge_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