A6: Indexes, triggers, user functions, transactions and population

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

This artifact contains the indexes, triggers, user functions, transactions and population specifications of the database.

1. Database Workload

1.1. Tuple Estimation

Relation reference	Relation Name	Order of magnitude	Estimated growth
R01	user	thousands	hundreds per day
R02	label	dozens	units per day
R03	notification	tens of thousands	hundreds per day
R04	user_management	thousands	hundreds per day
R05	vote	hundreds of thousands	thousands per day
R06	question	thousands	hundreds per day
R07	answer	thousands	hundreds per day
R08	comment	hundreds	dozens per day
R09	report	hundreds	dozens per day
R010	report_status	hundreds	dozens per day
R011	question_following	tens of thousands	hundreds per day
R012	label_following	tens of thousands	hundreds per day
R013	question_label	tens of thousands	hundreds per day

1.2. Frequent Queries

> Most important queries (SELECT) and their frequency.

Query	SELECT01
Description	Gets questions with most likes for guests homepage
Frequency	thousands per day

SQL code

```
 SELECT question.title, question.description, "user".username, question.nr_likes, question.nr_dislikes, question.question_date
FROM question JOIN "user" ON question.user_id = "user".id
ORDER BY question.nr_likes desc
LIMIT 30
OFFSET $offset;
```

Query	SELECT02
Description	Gets most recent questions
Frequency	thousands per day

SQL code

```
 SELECT * FROM question
ORDER BY question.question_date desc
LIMIT 30
OFFSET $offset;
```

Query	SELECT03
Description	Gets most scored answers
Frequency	hundreds per day

SQL code

```
 SELECT * FROM answer
ORDER BY nr_likes desc
LIMIT 30
OFFSET $offset;
```

Query	SELECT04
Description	Gets most recent answers
Frequency	hundreds per day

SQL code

```
 SELECT * FROM answer
ORDER BY answer_date desc
LIMIT 30
OFFSET $offset
```

Query	SELECT05
Description	Gets most recent comments
Frequency	hundreds per day

SQL code

```
 SELECT * FROM comment
ORDER BY comment.comment_date desc
LIMIT 30
OFFSET $offset
```

Query	SELECT06
Description	Gets most popular labels
Frequency	hundreds per day

SQL code

```
 SELECT "label".name, count(*) as nr
FROM "label" join question_label ON "label".id = question_label.label_id
JOIN question ON question.id = question_label.question_id
group by "label".id
order by nr desc
limit 5;
```

Query	SELECT07
Description	Gets most popular questions within label
Frequency	hundreds per day

SQL code

```
 SELECT question.* , label.name
FROM label join question_label ON label.id = question_label.label_id
JOIN question ON question.id = question_label.question_id
where label.name = $label
order by question.nr_likes desc
LIMIT 30
OFFSET = $offset;
```

Query	SELECT08
Description	Gets most recent questions within label
Frequency	hundreds per day

SQL code

```
 SELECT question.*
FROM label join question_label ON label.id = question_label.label_id
JOIN question ON question.id = question_label.question_id
where label.name = $label
order by question.answer_date desc
LIMIT 20;
```

Query	SELECT09
Description	Gets all user's question, score for profile
Frequency	hundreds per day

SQL code

```
 SELECT question.title, (question.nr_likes - question.nr_dislikes) as score
from question
where question.user_id = 2;
```

Query	SELECT10
Description	Gets all notifications order by date of a given user
Frequency	hundreds per day

SQL code

```
 SELECT notification.*
FROM "user" u join notification ON u.id = notification.user_id
WHERE u.id = 4
ORDER BY notification.date desc
LIMIT 5
OFFSET $offset;
```

Query	SELECT11

Query	SELECT11
Description	Gets user info
Frequency	hundreds per day

SQL code

```
 SELECT first_name, last_name, bio, score
FROM "user" u
WHERE u.id = 4
```

Query	SELECT12
Description	Gets a label with string
Frequency	dozens per day

SQL code

```
 SELECT id, title
FROM label
WHERE name LIKE '%$string%'
ORDER BY notification.date desc;
```

Query	SELECT13
Description	Gets a question with string
Frequency	dozens per day

SQL code

```
 SELECT id, title,
FROM question
WHERE description LIKE '%$string%'
ORDER BY notification.date desc;
```

Query	SELECT14
Description	Gets questions reports
Frequency	dozens per day

SQL code

```
 SELECT report.*, u.username
FROM report join "user" u on u.id = report.reporter_id
WHERE answer_id IS NULL
AND comment_id IS NULL
and user_id is NULL;
```

Query	SELECT15
Description	Gets answers reports
Frequency	dozens per day

SQL code

```
 SELECT report.*, u.username
FROM report join "user" u on u.id = report.reporter_id
WHERE question_id IS NULL
AND comment_id IS NULL
and user_id is NULL;
```

Query	SELECT16
Description	Gets comments reports
Frequency	dozens per day

SQL code

```
 SELECT report.*, u.username
FROM report join "user" u on u.id = report.reporter_id
WHERE answer_id IS NULL
AND question_id IS NULL
and user_id is NULL;
```

Query	SELECT17
Description	Gets user reports
Frequency	dozens per day

SQL code

```
 SELECT report.*, u.username
FROM report join "user" u on u.id = report.reporter_id
WHERE user_id is NOT NULL;
```

Query	SELECT18
Description	Gets users with most scores
Frequency	dozens per day

SQL code

```
 SELECT *
FROM "user" u join user_management on u.id = user_management.user_id
where user_management.status = 'user'
ORDER BY score desc
LIMIT 30
```

Query	SELECT19
Description	Gets moderators with most scores
Frequency	dozens per day

SQL code

```
SELECT *
FROM "user" u join user_management on u.id = user_management.user_id
where user_management.status = 'moderator'
ORDER BY score desc
LIMIT 30;
```

Query	SELECT20
Description	Get question info
Frequency	dozens per day

SQL code

```
SELECT *
FROM question WHERE id = $id;
```

Query	SELECT21
Description	Get comment info
Frequency	dozens per day

SQL code

```
SELECT *
FROM comment WHERE id = $id;
```

Query	SELECT22
Description	Get a user's score
Frequency	dozens per day

SQL code

```
SELECT score
FROM "user" u WHERE u.id = $id;
```

Query	SELECT23
Description	Gets the count of answers in a question
Frequency	dozens per day

SQL code

```
SELECT count(*) as nº
FROM answer
WHERE question_id = $id;
```

Query	SELECT24
Description	Gets the most recent questions that the user is following for homepage
Frequency	dozens per day

SQL code

```
select question.*
from question join question_following
on question.id = question_following.question_id
where question_following.user_id = $id
limit 5;
```

Query	SELECT25
Description	Gets users with most reports
Frequency	dozens per day

SQL code

```
SELECT u.username, count(*) as reports
FROM report
JOIN "user" u ON u.id = report.user_id
GROUP BY u.id
ORDER BY reports desc;
```

1.3. Frequent Updates

> This section features the most important updates and their respective convenience.

| **Query** | INSERT01 | | --- | --- | | **Description** | Adding a new question | | **Frequency** | hundreds per day | **SQL code**

```
INSERT INTO questions (user_id, title, description)
 VALUES ($user_id, $title, $description);
```

| **Query** | INSERT02 | | --- | --- | | **Description** | Adding a new answer | | **Frequency** | hundreds per day | **SQL code**

```
INSERT INTO answer (user_id, question_id, content)
 VALUES ($user_id, $question_id, $content);
```

| **Query** | INSERT03 | | --- | --- | | **Description** | Adding a new coment | | **Frequency** | dozens per day | **SQL code**

```
INSERT INTO comment (user_id, question_id, answer_id, content)
 VALUES ($user_id, $question_id, $answer_id $content);
```

| **Query** | INSERT04 | | --- | --- | | **Description** | Adding a new user | | **Frequency** | dozens per day | **SQL code**

```
INSERT INTO "user" (first_name, last_name, email, bio, username, password, bio)
 VALUES ($first_name, $last_name, $email, $bio, $username, $password, $bio);
```

| **Query** | UPDATE01 | | --- | --- | | **Description** | Updating user info | | **Frequency** | dozens per day | **SQL code**

```
UPDATE "user"
 SET username = $username, bio = $bio, first_name = $first_name, last_name = $last_name
 VALUES id = $id;
```

| **Query** | UPDATE02 | | --- | --- | | **Description** | Change user status | | **Frequency** | dozens per day | **SQL code**

```
UPDATE user_management
 SET status = $status
 VALUES user_id = $user_idid;
```

| **Query** | UPDATE03 | | --- | --- | | **Description** | Edit question | | **Frequency** | dozens per day | **SQL code**

```
UPDATE question
 SET title = $title, description = $description
 VALUES id = $id;
```

| **Query** | UPDATE04 | | --- | --- | | **Description** | Edit question | | **Frequency** | dozens per day | **SQL code**

```
UPDATE answer
 SET content = $content
 VALUES id = $id;
```

| **Query** | UPDATE05 | | --- | --- | | **Description** | Edit question | | **Frequency** | dozens per day | **SQL code**

```
UPDATE comment
 SET content = $content
 VALUES id = $id;
```

2. Proposed Indices

2.1. Performance Indices

> Indices proposed to improve performance of the identified queries. B-Tree is used when we want to grab chunks of data, whilst Hash is used to grab specific database elements.

Index	**IDX01**
Related queries	SELECT01, SELECT07, SELECT23
Relation	question
Attribute	nr_likes
Type	B-tree
Cardinality	high
Clustering	No
Justification	To allow searching questions ordered by score faster.

```
CREATE INDEX question_score ON question USING btree(nr_likes)
```

Index	**IDX02**
Related queries	SELECT02, SELECT08
Relation	question
Attribute	question_date
Type	B-tree
Cardinality	high
Clustering	No
Justification	To allow searching questions ordered by date faster.

```
CREATE INDEX question_date ON question USING btree(question_date)
```

Index	**IDX03**
Related queries	SELECT03
Relation	answer
Attribute	question_id, nr_likes

Index	IDX03
Type	B-tree
Cardinality	medium
Clustering	No
Justification	To allow searching answers to a specific question ordered by score faster.

```
CREATE INDEX answer_score ON answer USING btree(question_id, nr_likes)
```

Index	IDX04
Related queries	SELECT04
Relation	answer
Attribute	question_id, answer_date
Type	B-tree
Cardinality	medium
Clustering	No
Justification	To allow searching answers to a specific question ordered by date faster.

```
CREATE INDEX answer_date ON answer USING btree(question_id, answer_date)
```

Index	IDX05
Related queries	SELECT05
Relation	comment
Attribute	question_id, comment_date
Type	B-tree
Cardinality	medium
Clustering	No
Justification	To allow searching comments to a specific question ordered by date faster.

```
CREATE INDEX comment_date ON comment USING btree(question_id, comment_date)
```

Index	IDX06
Related queries	SELECT06
Relation	following
Attribute	label_id
Type	B-tree
Cardinality	medium

Index	IDX06
Clustering	No
Justification	To allow searching for the most popular labels faster.

```
CREATE INDEX label_popularity ON following USING hash(label_id)
```

Index	IDX07
Related queries	SELECT09
Relation	question
Attribute	user_id
Type	B-tree
Cardinality	medium
Clustering	No
Justification	To allow searching for a user's number of questions faster.

```
CREATE INDEX question_user ON question USING btree(user_id)
```

Index	IDX08
Related queries	SELECT09
Relation	answer
Attribute	user_id
Type	B-tree
Cardinality	medium
Clustering	No
Justification	To allow searching for a user's number of answers faster.

```
CREATE INDEX answer_user ON answer USING btree(user_id)
```

Index	IDX09
Related queries	SELECT10
Relation	notification
Attribute	user_id, date
Type	B-tree
Cardinality	medium
Clustering	No
Justification	To allow searching for a user's number of notifications ordered by date faster.

```
CREATE INDEX notification_user_date ON notification USING btree(user_id, date)
```

Index	IDX10
Related queries	SELECT11
Relation	user
Attribute	email
Type	Hash
Cardinality	medium
Clustering	No
Justification	To allow retrieving user information faster, based on the email

```
CREATE INDEX user_email ON "user" USING hash(email)
```

Index	IDX11
Related queries	SELECT17
Relation	report
Attribute	user_id
Type	B-tree
Cardinality	low
Clustering	No
Justification	To allow searching for the number of reports a user has received

```
CREATE INDEX report_user ON report USING btree(user_id)
```

Index	IDX12
Related queries	SELECT18, SELECT19
Relation	user
Attribute	score
Type	B-tree
Cardinality	low
Clustering	No
Justification	To allow searching for users and moderators ordered by score faster

```
CREATE INDEX user_score ON "user" USING btree(score)
```

2.2. Full-text Search Indices

> GIN is used for infrequently updated tables, leading to faster lookups. GiST is better for dynamic data.

Index	IDX13
Related queries	SELECT12
Relation	label
Attribute	name
Type	GIN
Cardinality	high
Clustering	No
Justification	To enhance performance of full text searches on label names

```
CREATE INDEX label_name ON label USING gin(to_tsvector('english', name));
```

Index	IDX14
Related queries	SELECT13
Relation	question
Attribute	title
Type	GiST
Cardinality	high
Clustering	No
Justification	To enhance performance of full text searches on question titles

```
CREATE INDEX question_title ON question USING gist(to_tsvector('english', title), to_tsvector('english', description))
```

3. Triggers

Trigger	TRIGGER01
Description	When a user votes on a question, the number of likes or dislikes of that question needs to be updated, as well as the owner's score.
SQL code	

```
CREATE OR REPLACE FUNCTION update_score_question() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT vote.id FROM vote WHERE NEW."vote" = FALSE) THEN
        UPDATE question
        SET nr_dislikes = nr_dislikes+1
        WHERE NEW.question_id = id;
        UPDATE "user"
        SET score = score-1
        FROM question
        WHERE NEW.question_id = question.id AND question.user_id = "user".id;
    ELSE IF EXISTS (SELECT vote.id FROM vote WHERE NEW."vote" = TRUE) THEN
        UPDATE question
        SET nr_likes = nr_likes+1
        WHERE NEW.question_id = id;
        UPDATE "user"
        SET score = score+1
        FROM question
        WHERE NEW.question_id = question.id AND question.user_id = "user".id;
    END IF;
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER update_score_question
    AFTER INSERT OR UPDATE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE update_score_question();
```

Trigger	TRIGGER02
Description	When a user votes on an answer, the number of likes or dislikes of that answer needs to be updated, as well as the owner's score.
SQL code	

```
 CREATE OR REPLACE FUNCTION update_score_answer() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT vote.id FROM vote WHERE NEW."vote" = FALSE) THEN
        UPDATE answer
        SET nr_dislikes = nr_dislikes+1
        WHERE NEW.answer_id = id;
        UPDATE "user"
        SET score = score-1
        FROM answer
        WHERE NEW.answer_id = answer.id AND answer.user_id = "user".id;
    ELSE IF EXISTS (SELECT vote.id FROM vote WHERE NEW."vote" = TRUE) THEN
        UPDATE answer
        SET nr_likes = nr_likes+1
        WHERE NEW.answer_id = id;
        UPDATE "user"
        SET score = score+1
        FROM answer
        WHERE NEW.answer_id = answer.id AND answer.user_id = "user".id;
    END IF;
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER update_score_answer
    AFTER INSERT OR UPDATE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE update_score_answer();
```

Trigger	TRIGGER03
Description	Users cannot vote on their own questions.
SQL code	

```
 CREATE OR REPLACE FUNCTION vote_own_question() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT question.user_id
               FROM question
               WHERE NEW.user_id = question.user_id AND NEW.question_id = question.id) THEN
        RAISE EXCEPTION 'A user cannot vote on his own question';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER vote_own_question
    BEFORE INSERT OR UPDATE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE vote_own_question();
```

Trigger	TRIGGER04
Description	Users cannot vote on their own answers.
SQL code	

```
 CREATE OR REPLACE FUNCTION vote_own_answer() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT answer.user_id
               FROM answer
               WHERE NEW.user_id = answer.user_id AND NEW.answer_id = answer.id) THEN
        RAISE EXCEPTION 'A user cannot vote on his own answer';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER vote_own_answer
    BEFORE INSERT OR UPDATE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE vote_own_answer();
```

Trigger	TRIGGER05
Description	The date of an answer must be greater than or equal to the date of the respective question.
SQL code	

```
 CREATE OR REPLACE FUNCTION answer_date() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT question.question_date
               FROM question
               WHERE NEW.question_id = question.id AND NEW.answer_date < question.question_date) THEN
        RAISE EXCEPTION 'The date of an answer cannot be earlier than the date of its question';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER answer_date
    BEFORE INSERT OR UPDATE ON answer
    FOR EACH ROW
    EXECUTE PROCEDURE answer_date();
```

Trigger	TRIGGER06
Description	The date of a comment must be greater than or equal to the date of the respective answer.
SQL code	

```
CREATE OR REPLACE FUNCTION comment_date_answer() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT answer.answer_date
               FROM answer
               WHERE NEW.answer_id = answer.id AND NEW.comment_date < answer.answer_date) THEN
        RAISE EXCEPTION 'The date of a comment cannot be earlier than the date of its answer';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER comment_date_answer
    BEFORE INSERT OR UPDATE ON comment
    FOR EACH ROW
    EXECUTE PROCEDURE comment_date_answer();
```

Trigger	TRIGGER07
Description	The date of a comment must be greater than or equal to the date of the respective question.
SQL code	

```
CREATE OR REPLACE FUNCTION comment_date_question() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT question.question_date
               FROM question
               WHERE NEW.question_id = question.id AND NEW.comment_date < question.question_date) THEN
        RAISE EXCEPTION 'The date of a comment cannot be earlier than the date of its question';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER comment_date_question
    BEFORE INSERT OR UPDATE ON comment
    FOR EACH ROW
    EXECUTE PROCEDURE comment_date_question();
```

Trigger	TRIGGER08
Description	An answer or question can only be voted once by the same user.
SQL code	

```
CREATE OR REPLACE FUNCTION vote_once() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT * FROM vote
               WHERE ((NEW.user_id = vote.user_id AND NEW.question_id = vote.question_id) OR
                      (NEW.user_id = vote.user_id AND NEW.answer_id = vote.answer_id))) THEN
        RAISE EXCEPTION 'An element can only be voted once by the same user';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER vote_once
    BEFORE INSERT ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE vote_once();
```

Trigger	TRIGGER09
Description	A report can only be handled by an administrator or moderator.
SQL code	

```
CREATE OR REPLACE FUNCTION report_status_responsible() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF NOT EXISTS (SELECT user_management.user_id
                   FROM user_management
                   WHERE ((user_management.status = 'moderator' OR user_management.status = 'administrator')
                          AND
                          (user_management.user_id = NEW.responsible_user))) THEN
            RAISE EXCEPTION 'A report can only be handled by an administrator or moderator';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER report_status_responsible
    BEFORE INSERT OR UPDATE ON report_status
    FOR EACH ROW
    EXECUTE PROCEDURE report_status_responsible();
```

Trigger	TRIGGER10
Description	There should be only one marked correct answer per question.
SQL code	

```
 CREATE OR REPLACE FUNCTION marked_answer() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT marked_answer FROM answer WHERE answer.id = NEW.id AND NEW.marked_answer = TRUE) THEN
    UPDATE answer
    SET marked_answer = FALSE
    WHERE (answer.question_id = NEW.question_id AND answer.id != NEW.id);
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER marked_answer
    AFTER UPDATE OF marked_answer ON answer
    FOR EACH ROW
    EXECUTE PROCEDURE marked_answer();
```

Trigger	TRIGGER11
Description	Updates the question score when a vote is removed from the table. (Useful for the "vote" feature)
SQL code	

```
 CREATE OR REPLACE FUNCTION update_score_question_delete() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT vote.id FROM vote WHERE OLD."vote" = FALSE) THEN
        UPDATE question
        SET nr_dislikes = nr_dislikes-1
        WHERE OLD.question_id = id;
        UPDATE "user"
        SET score = score+1
        FROM question
        WHERE OLD.question_id = question.id AND question.user_id = "user".id;
    ELSE IF EXISTS (SELECT vote.id FROM vote WHERE OLD."vote" = TRUE) THEN
        UPDATE question
        SET nr_likes = nr_likes-1
        WHERE OLD.question_id = id;
        UPDATE "user"
        SET score = score-1
        FROM question
        WHERE OLD.question_id = question.id AND question.user_id = "user".id;
    END IF;
    END IF;
    RETURN OLD;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER update_score_question_delete
    AFTER DELETE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE update_score_question_delete();
```

Trigger	TRIGGER12
Description	Updates the answer score when a vote is removed from the table. (Useful for the "vote" feature)
SQL code	

```
 CREATE OR REPLACE FUNCTION update_score_answer_delete() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT vote.id FROM vote WHERE OLD."vote" = FALSE) THEN
        UPDATE answer
        SET nr_dislikes = nr_dislikes-1
        WHERE OLD.answer_id = id;
        UPDATE "user"
        SET score = score+1
        FROM answer
        WHERE OLD.answer_id = answer.id AND answer.user_id = "user".id;
    ELSE IF EXISTS (SELECT vote.id FROM vote WHERE OLD."vote" = TRUE) THEN
        UPDATE answer
        SET nr_likes = nr_likes-1
        WHERE OLD.answer_id = id;
        UPDATE "user"
        SET score = score-1
        FROM answer
        WHERE OLD.answer_id = answer.id AND answer.user_id = "user".id;
    END IF;
    END IF;
    RETURN OLD;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER update_score_answer_delete
    AFTER DELETE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE update_score_answer_delete();
```

4. Transactions

| T01 | Insert new user | | -------------- | ---------------------------------- | | Justification | When we add a new user, we need to garantee that both inserts on table user and table user_management are consistently sucessful, so that we use a transaction. We want to ensure that the user_management get the user foreign key correspondent to the primary key of the new user inserted. The READ UNCOMMITTED Isolation Level is the lowest, but it ensures that everything goes well. | | Isolation level | READ UNCOMMITTED | **SQL code**

```
 CREATE OR REPLACE FUNCTION add_user_management(first_name TEXT, last_name TEXT, email TEXT, bio TEXT, username TEXT, password TEXT) R
BEGIN
    INSERT INTO "user" (first_name, last_name, email, bio, username, password)
        VALUES (first_name, last_name, email, bio, username, password);
    INSERT INTO user_management (user_id)
        VALUES (currval('user_id_seq'));
END;
$$ LANGUAGE plpgsql;

BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
    SELECT add_user_management($first_name, $last_name, $email, $bio, $username, $password);
COMMIT;
```

| T02 | Insert new report | | -------------- | ---------------------------------- | | Justification | When we add a new report, we need to garantee that both inserts on table report and table report_status are consistently sucessful, so that we use a transaction. We want to ensure that the report_status get the report foreign key correspondent to the primary key of the new report inserted. This READ UNCOMMITTED Isolation Level is the lowest, but it ensures that everything goes well. | | Isolation level | READ UNCOMMITTED | **SQL code**

```
 CREATE OR REPLACE FUNCTION add_report_status(reporter_id INT, user_id INT, description TEXT, responsible_user INT) RETURNS void AS $$
BEGIN
    INSERT INTO report (reporter_id, user_id, description)
        VALUES (reporter_id, user_id, description);

    INSERT INTO report_status (report_id, responsible_user)
        VALUES (currval('report_id_seq'), responsible_user);
END;
$$ LANGUAGE plpgsql;

BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
    SELECT  add_report_status($reporter_id, $user_id, $description, $responsible_user);
COMMIT;
```

| T03 | Insert new notification upon getting a new answer to a question| | -------------- | ---------------------------------- | | Justification | When we add a new answer, we need to garantee that both inserts on table answer and table notification are consistently sucessful, so that we use a transaction. We want to ensure that the notification gets a foreign key to the question owner user, and the content on the notification shows the user that answered the question. This READ UNCOMMITTED Isolation Level is the lowest, but it ensures that everything goes well. | | Isolation level | READ UNCOMMITTED | **SQL code**

```
 CREATE OR REPLACE FUNCTION answered_question_notif(answer_user_id INT, question_user_id INT, question_id INT, answer_content TEXT) RE
BEGIN
    INSERT INTO answer (user_id, question_id, content)
        VALUES (answer_user_id, question_id, answer_content);

    INSERT INTO notification (content, user_id)
        VALUES (CONCAT((SELECT username AS responsible FROM "user" WHERE id = answer_user_id), ' answered your question!'), question_
END;
$$ LANGUAGE plpgsql;

BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
    SELECT answered_question_notif($answer_user_id, $question_user_id, $question_id, $answer_content);
COMMIT;
```

| T04 | Insert new notification upon getting a new comment to a question| | -------------- | ---------------------------------- | | Justification | When we add a new comment to a question, we need to garantee that both inserts on table comment and table notification are consistently sucessful, so that we use a transaction. We want to ensure that the notification gets a foreign key to the question owner user, and the content on the notification shows the user that commented the question. This READ UNCOMMITTED Isolation Level is the lowest, but it ensures that everything goes well. | | Isolation level | READ UNCOMMITTED | **SQL code**

```
 CREATE OR REPLACE FUNCTION commented_question_notif(comment_user_id INT, question_user_id INT, question_id INT, comment_content TEXT)
BEGIN
    INSERT INTO comment (user_id, question_id, content)
        VALUES (comment_user_id, question_user_id, comment_content);

    INSERT INTO notification (content, user_id)
        VALUES (CONCAT((SELECT username AS responsible FROM "user" WHERE id=comment_user_id), ' commented your question!'), question_
END;
$$ LANGUAGE plpgsql;

BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
    SELECT commented_question_notif($comment_user_id, $question_user_id, $question_id, $comment_content);
COMMIT;
```

| T05 | Insert new notification upon getting a new comment to an answer| | -------------- | ---------------------------------- | | Justification | When we add a new comment to an answer, we need to garantee that both inserts on table comment and table notification are consistently sucessful, so that we use a transaction. We want to ensure that the notification gets a foreign key to the answer owner user, and the content on the notification shows the user that commented the answer. This READ UNCOMMITTED Isolation Level is the lowest, but it ensures that everything goes well. | | Isolation level | READ UNCOMMITTED | **SQL code**

```
CREATE OR REPLACE FUNCTION commented_answer_notif(comment_user_id INT, answer_user_id INT, answer_id INT, comment_content TEXT) RETUR
BEGIN
    INSERT INTO comment (user_id, answer_id, content)
        VALUES (comment_user_id, answer_user_id, comment_content);

    INSERT INTO notification (content, user_id)
        VALUES (CONCAT((SELECT username AS responsible FROM "user" WHERE id=comment_user_id), ' commented your answer!'), answer_user
END;
$$ LANGUAGE plpgsql;

BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
    SELECT commented_answer_notif($comment_user_id, $answer_user_id, $answer_id, $comment_content);
COMMIT;
```

5. SQL Code

5.1. Database schema

```
-----------------------------------------
-- 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)
);

-----------------------------------------
-- INDEXES
-----------------------------------------

CREATE INDEX question_score ON question USING btree(nr_likes);
CREATE INDEX question_date ON question USING btree(question_date);
CREATE INDEX answer_score ON answer USING btree(question_id, nr_likes);
CREATE INDEX answer_date ON answer USING btree(question_id, answer_date);
CREATE INDEX comment_date ON comment USING btree(question_id, comment_date);
CREATE INDEX label_popularity ON label_following USING btree(label_id);
CREATE INDEX question_user ON question USING btree(user_id);
CREATE INDEX answer_user ON answer USING btree(user_id);
CREATE INDEX notification_user_date ON notification USING btree(user_id, date);
CREATE INDEX user_username ON "user" USING hash(username);
CREATE INDEX report_user ON report USING btree(user_id);
CREATE INDEX user_score ON "user" USING btree(score);

CREATE INDEX label_name ON label USING gin(to_tsvector('english', name));
CREATE INDEX question_title ON question USING gist(to_tsvector('english', title));

-----------------------------------------
-- TRIGGERS and UDFs
-----------------------------------------
--Trigger 1
CREATE OR REPLACE FUNCTION update_score_question() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT vote.id FROM vote WHERE NEW."vote" = FALSE) THEN
        UPDATE question
        SET nr_dislikes = nr_dislikes+1
        WHERE NEW.question_id = id;
        UPDATE "user"
        SET score = score-1
        FROM question
        WHERE NEW.question_id = question.id AND question.user_id = "user".id;
    ELSE IF EXISTS (SELECT vote.id FROM vote WHERE NEW."vote" = TRUE) THEN
        UPDATE question
        SET nr_likes = nr_likes+1
        WHERE NEW.question_id = id;
        UPDATE "user"
        SET score = score+1
        FROM question
        WHERE NEW.question_id = question.id AND question.user_id = "user".id;
    END IF;
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER update_score_question
    AFTER INSERT OR UPDATE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE update_score_question();


--Trigger 2
CREATE OR REPLACE FUNCTION update_score_answer() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT vote.id FROM vote WHERE NEW."vote" = FALSE) THEN
        UPDATE answer
        SET nr_dislikes = nr_dislikes+1
        WHERE NEW.answer_id = id;
        UPDATE "user"
        SET score = score-1
        FROM answer
        WHERE NEW.answer_id = answer.id AND answer.user_id = "user".id;
    ELSE IF EXISTS (SELECT vote.id FROM vote WHERE NEW."vote" = TRUE) THEN
        UPDATE answer
        SET nr_likes = nr_likes+1
        WHERE NEW.answer_id = id;
        UPDATE "user"
        SET score = score+1
        FROM answer
        WHERE NEW.answer_id = answer.id AND answer.user_id = "user".id;
    END IF;
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER update_score_answer
    AFTER INSERT OR UPDATE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE update_score_answer();


--Trigger 3
CREATE OR REPLACE FUNCTION vote_own_question() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT question.user_id
               FROM question
               WHERE NEW.user_id = question.user_id AND NEW.question_id = question.id) THEN
        RAISE EXCEPTION 'A user cannot vote on his own question';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER vote_own_question
    BEFORE INSERT OR UPDATE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE vote_own_question();


--Trigger 4
CREATE OR REPLACE FUNCTION vote_own_answer() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT answer.user_id
               FROM answer
               WHERE NEW.user_id = answer.user_id AND NEW.answer_id = answer.id) THEN
        RAISE EXCEPTION 'A user cannot vote on his own answer';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER vote_own_answer
    BEFORE INSERT OR UPDATE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE vote_own_answer();


--Trigger 5
CREATE OR REPLACE FUNCTION answer_date() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT question.question_date
               FROM question
               WHERE NEW.question_id = question.id AND NEW.answer_date < question.question_date) THEN
        RAISE EXCEPTION 'The date of an answer cannot be earlier than the date of its question';
    END IF;
    RETURN NEW;
END
$BODY$
```

```
LANGUAGE plpgsql;

CREATE TRIGGER answer_date
    BEFORE INSERT OR UPDATE ON answer
    FOR EACH ROW
    EXECUTE PROCEDURE answer_date();


--Trigger 6
CREATE OR REPLACE FUNCTION comment_date_answer() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT answer.answer_date
               FROM answer
               WHERE NEW.answer_id = answer.id AND NEW.comment_date < answer.answer_date) THEN
        RAISE EXCEPTION 'The date of a comment cannot be earlier than the date of its answer';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER comment_date_answer
    BEFORE INSERT OR UPDATE ON comment
    FOR EACH ROW
    EXECUTE PROCEDURE comment_date_answer();


--Trigger 7
CREATE OR REPLACE FUNCTION comment_date_question() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT question.question_date
               FROM question
               WHERE NEW.question_id = question.id AND NEW.comment_date < question.question_date) THEN
        RAISE EXCEPTION 'The date of a comment cannot be earlier than the date of its question';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER comment_date_question
    BEFORE INSERT OR UPDATE ON comment
    FOR EACH ROW
    EXECUTE PROCEDURE comment_date_question();


--Trigger 8
CREATE OR REPLACE FUNCTION vote_once() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT * FROM vote
               WHERE ((NEW.user_id = vote.user_id AND NEW.question_id = vote.question_id) OR
                      (NEW.user_id = vote.user_id AND NEW.answer_id = vote.answer_id))) THEN
        RAISE EXCEPTION 'An element can only be voted once by the same user';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER vote_once
    BEFORE INSERT ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE vote_once();
    EXECUTE PROCEDURE vote_once();


--Trigger 9
CREATE OR REPLACE FUNCTION report_status_responsible() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF NOT EXISTS (SELECT user_management.user_id
                   FROM user_management
                   WHERE ((user_management.status = 'moderator' OR user_management.status = 'administrator')
                          AND
                          (user_management.user_id = NEW.responsible_user))) THEN
            RAISE EXCEPTION 'A report can only be handled by an administrator or moderator';
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER report_status_responsible
    BEFORE INSERT OR UPDATE ON report_status
    FOR EACH ROW
    EXECUTE PROCEDURE report_status_responsible();


--Trigger 10
CREATE OR REPLACE FUNCTION marked_answer() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT marked_answer FROM answer WHERE answer.id = NEW.id AND NEW.marked_answer = TRUE) THEN
    UPDATE answer
    SET marked_answer = FALSE
    WHERE (answer.question_id = NEW.question_id AND answer.id != NEW.id);
    END IF;
    RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER marked_answer
    AFTER UPDATE OF marked_answer ON answer
    FOR EACH ROW
    EXECUTE PROCEDURE marked_answer();

--Trigger 11
CREATE OR REPLACE FUNCTION update_score_question_delete() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT vote.id FROM vote WHERE OLD."vote" = FALSE) THEN
        UPDATE question
        SET nr_dislikes = nr_dislikes-1
        WHERE OLD.question_id = id;
        UPDATE "user"
        SET score = score+1
        FROM question
        WHERE OLD.question_id = question.id AND question.user_id = "user".id;
    ELSE IF EXISTS (SELECT vote.id FROM vote WHERE OLD."vote" = TRUE) THEN
        UPDATE question
        SET nr_likes = nr_likes-1
        WHERE OLD.question_id = id;
        UPDATE "user"
        SET score = score-1
        FROM question
        WHERE OLD.question_id = question.id AND question.user_id = "user".id;
    END IF;
    END IF;
    RETURN OLD;
    RETURN OLD;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER update_score_question_delete
    AFTER DELETE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE update_score_question_delete();


--Trigger12
CREATE OR REPLACE FUNCTION update_score_answer_delete() RETURNS TRIGGER AS
$BODY$
BEGIN
    IF EXISTS (SELECT vote.id FROM vote WHERE OLD."vote" = FALSE) THEN
        UPDATE answer
        SET nr_dislikes = nr_dislikes-1
        WHERE OLD.answer_id = id;
        UPDATE "user"
        SET score = score+1
        FROM answer
        WHERE OLD.answer_id = answer.id AND answer.user_id = "user".id;
    ELSE IF EXISTS (SELECT vote.id FROM vote WHERE OLD."vote" = TRUE) THEN
        UPDATE answer
        SET nr_likes = nr_likes-1
        WHERE OLD.answer_id = id;
        UPDATE "user"
        SET score = score-1
        FROM answer
        WHERE OLD.answer_id = answer.id AND answer.user_id = "user".id;
    END IF;
    END IF;
    RETURN OLD;
END
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER update_score_answer_delete
    AFTER DELETE ON vote
    FOR EACH ROW
    EXECUTE PROCEDURE update_score_answer_delete();

-----------------------------------------
-- TRANSACTIONS FUNCTIONS
-----------------------------------------
--Transaction 1
CREATE OR REPLACE FUNCTION add_user_management(first_name TEXT, last_name TEXT, email TEXT, bio TEXT, username TEXT, password TEXT) R
BEGIN
    INSERT INTO "user" (first_name, last_name, email, bio, username, password)
        VALUES (first_name, last_name, email, bio, username, password);
    INSERT INTO user_management (user_id)
        VALUES (currval('user_id_seq'));
END;
$$ LANGUAGE plpgsql;

--Transaction 2
CREATE OR REPLACE FUNCTION add_report_status(reporter_id INT, user_id INT, description TEXT, responsible_user INT) RETURNS void AS $$
BEGIN
    INSERT INTO report (reporter_id, user_id, description)
        VALUES (reporter_id, user_id, description);

    INSERT INTO report_status (report_id, responsible_user)
        VALUES (currval('report_id_seq'), responsible_user);
END;
$$ LANGUAGE plpgsql;
```

```
--Transaction 3
CREATE OR REPLACE FUNCTION answered_question_notif(answer_user_id INT, question_user_id INT, question_id INT, answer_content TEXT) RE
BEGIN
    INSERT INTO answer (user_id, question_id, content)
        VALUES (answer_user_id, question_id, answer_content);

    INSERT INTO notification (content, user_id)
        VALUES (CONCAT((SELECT username AS responsible FROM "user" WHERE id = answer_user_id), ' answered your question!'), question_
END;
$$ LANGUAGE plpgsql;

--Transaction 4
CREATE OR REPLACE FUNCTION commented_question_notif(comment_user_id INT, question_user_id INT, question_id INT, comment_content TEXT)
BEGIN
    INSERT INTO comment (user_id, question_id, content)
        VALUES (comment_user_id, question_user_id, comment_content);

    INSERT INTO notification (content, user_id)
        VALUES (CONCAT((SELECT username AS responsible FROM "user" WHERE id=comment_user_id), ' commented your question!'), question_
END;
$$ LANGUAGE plpgsql;

--Transaction 5
CREATE OR REPLACE FUNCTION commented_answer_notif(comment_user_id INT, answer_user_id INT, answer_id INT, comment_content TEXT) RETUR
BEGIN
    INSERT INTO comment (user_id, answer_id, content)
        VALUES (comment_user_id, answer_user_id, comment_content);

    INSERT INTO notification (content, user_id)
        VALUES (CONCAT((SELECT username AS responsible FROM "user" WHERE id=comment_user_id), ' commented your answer!'), answer_user
END;
$$ LANGUAGE plpgsql;

-----------------------------------------
-- end
-----------------------------------------
```

5.2. Database population

```
 --user
--password for user 2: 1234
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Nuno', 'Cardoso', 'nmtc01@hotmail.com', 'UP student', 'nmtc01', '5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef7
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Pedro', 'Dantas', 'pedrodantas@hotmail.com', 'UP student', 'pedrodantas', '$2y$10$HfzIhGCCaxqyaIdGgjARSuOKAcm1Uy82YfLuNa
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Eduardo', 'Macedo', 'edumacedo@gmail.com', 'UP student', 'edu1234', '54ada40a5452d89d06fdfc1f3ac106a8ee360a6437cb3749ac1
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Roberto', 'Mourato', 'robmoura@hotmail.com', 'UP student', 'bob56moura', '75748e28843cab0cc7ca4ba8dcdc11552ea7bb3b652a3c
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Lurdes', 'Cardoso', 'lurdes01@hotmail.com', 'babysitter', 'lurdes01', 'e8253a1cf0d0d37032c0665de988369972f1b8502025aae5c
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Paula', 'Tavares', 'paultavares@hotmail.com', 'hairdresser', 'paula64', '39427764c90328dc57389a2adb9202ae18cb3c38f27f5e7
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Paulo', 'Cardoso', 'pcardoso@gmail.com', 'caixa', 'pcardoso', '359b5c9cef644b8cf40dd4c4b046bb69602efe18f2ffb75350d3d15dc
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Juliana', 'Pinto', 'jupinto@gmail.com', 'beautician', 'juju', 'b04a508509a4b3cd0aa7a5a5824c3bce43e8512572cbc35506a255f6a
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Carolina', 'Santos', 'carol@gmail.com', 'Up student', 'carolas', 'b04a508509a4b3cd0aa7a5a5824c3bce43e8512572cbc35506a255
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Clarinda', 'Teixeira', 'cteixeira@gmail.com', 'secretary', 'lindas', 'f15c4ab57663235d2212e96c280feae3e73ece417252d82c87
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Teresa', 'Teixeira', 'teresateixeira@gmail.com', 'seamstress', 'costureirinha', '327e51644afc5c688575ae43f30293513b6ed0d
insert into "user" (first_name, last_name, email, bio, username, password)
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Susana', 'Silva', 'ssilva@gmail.com', 'translator', 'silva77', 'd7aceb7abf5cf0b582ed8d10f324bb35495611edf04a8d30d699e21f
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Margarida', 'Pereira', 'mpereira@gmail.com', 'doctor', 'per77', 'b427c3dec9045990213ccc3dbffbdf189e75c7ce18e1a86ed7c4fe9
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Elisa', 'Neves', 'eneves@gmail.com', 'receptionist', 'neves02', '724bc1231a18dc91fb24605bcd7c542cdf176d7d26958416c08aca2
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Ana', 'Aires', 'aaires@gmail.com', 'psychiatrist', 'aires01', '33b1d188f33b0b088783aec571d122c38b060bb550c114f642191aa08
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Arlete', 'Araujo', 'arlete@gmail.com', 'urologist', 'araujo34', 'e98a98f373df403810bd4e569273e55351adede3f82461dbbc266af
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Aurora', 'Pinto', 'apinto@gmail.com', 'florist', 'flo55', '7cf60a4b93054d6ed588dddfc392903c18ed03161d70a9f161a62ad37bac9
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Albino', 'Costa', 'alcosta@gmail.com', 'chmist', 'ch10', '4e73b8eacfd31c5e8b51eb897694758b59d8d562ed684ee18783873ebe9677
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Aberto', 'Ferreira', 'ferreira01@gmail.com', 'librarian', 'afer5', '106dde628c0da1d7d96851ca4110e68d31513bec50a905597a74
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('António', 'Carvalho', 'antcarvalho@gmail.com', 'barber', 'bar01', '8287464bb9bba50593d53d95ced168b43db7db958d1263544a6d6
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Cândido', 'Faria', 'cfaria@gmail.com', 'carpenter', 'carp5', '365e72262bed1b6afb91ff1c7107b12bd7206d1f7a8c0c70217c57af54
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Cosme', 'Dias', 'cdias@gmail.com', 'adminstrator', 'adm11', '9911486dbd7cbe092287bb63e1ed44fee854d215eff4979b9b9ea837890
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Delfim', 'Mendes', 'dmendes@gmail.com', 'bricklayer', 'bri02', 'f28f28bbe8e16e9c6541f5e9c018d2060aaf5d0557202b916a0954b6
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Carlos', 'Freitas', 'freitas01@gmail.com', 'biomedical', 'frei70', 'a03d4e5546dea2b8a82c13e06542f6d454040c84a5d5caa5f106
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Domingos', 'Figueiredo', 'figueiredo1@gmail.com', 'mechanical', 'fig11', '6f490a2fe7dbe7cfcf08b48446db534791813b952b28f7
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('David', 'Faria', 'davfaria@gmail.com', 'magician', 'magic', '3be7a505483c0050243c5cbad4700da13925aa4137a55e9e33efd8bc4d0
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Emanuel', 'Fonseca', 'fonsec11@gmail.com', 'musician', 'fonseca22', 'd6518aa213ea4939fde8bd047ca8fb775cac36bf95d26f28cf2
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Eduardo', 'Santos', 'edsantos@gmail.com', 'oncologist', 'edu70', '4a53d7664413d4e5eefff5ad99e59ac1657b1cbdd6f1acbeea2a5f
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Carlos', 'Cruz', 'cruz@gmail.com', 'fisheman', 'cruz24', '2faecef2c88d125f3ba7dcac070b493f99d701ff43070c298cfa20a9f1459f
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Teodora', 'Vieira', 'vieira@gmail.com', 'cattle breeder', 'vieira01', 'e4a80fee9a4d8f87b0a5a408103ad723e2f83db981cdb10ef
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Silvia', 'Rodrigues', 'silrodrigues@gmail.com', 'librarian', 'sil14', 'f820f43abb76bc0ff9a3a1b5ddbea160efa28921723b601ca
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Pedro', 'Dantas', 'dantas@hotmail.com', 'UP student', 'dantas', 'ed5ae5a93a86c36e004d985cd801b3390dc91e4dda5c6d59b421c9f
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Eduardo', 'Macedo', 'macedo@gmail.com', 'UP student', 'edu1', '54ada40a5452d89d06fdfc1f3ac106a8ee360a6437cb3749ac1b5263b
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Roberto', 'Mourato', 'bmoura@hotmail.com', 'UP student', '56moura', '75748e28843cab0cc7ca4ba8dcdc11552ea7bb3b652a3cbd26c
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Ana', 'Aires', 'aires@gmail.com', 'psychiatrist', 'res01', '33b1d188f33b0b088783aec571d122c38b060bb550c114f642191aa089c2
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Arlete', 'Araujo', 'lete@gmail.com', 'urologist', 'araujo', 'e98a98f373df403810bd4e569273e55351adede3f82461dbbc266affedd
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Aurora', 'Pinto', 'pinto@gmail.com', 'florist', 'flo', '7cf60a4b93054d6ed588dddfc392903c18ed03161d70a9f161a62ad37bac9806
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Arlete', 'Araujo', 'arlete1@gmail.com', 'urologist', 'ar34', 'e98a98f373df403810bd4e569273e55351adede3f82461dbbc266affed
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Aurora', 'Pinto', 'aurora@gmail.com', 'florist', 'flo2', '7cf60a4b93054d6ed588dddfc392903c18ed03161d70a9f161a62ad37bac98
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Paula', 'Tavares', 'paul@hotmail.com', 'hairdresser', 'paula', '39427764c90328dc57389a2adb9202ae18cb3c38f27f5e7d3a19e5d2
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Paulo', 'Cardoso', 'paulo@gmail.com', 'caixa', 'cardoso', '359b5c9cef644b8cf40dd4c4b046bb69602efe18f2ffb75350d3d15dcff27
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Juliana', 'Pinto', 'juliana@gmail.com', 'beautician', 'pinto', 'b04a508509a4b3cd0aa7a5a5824c3bce43e8512572cbc35506a255f6
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Emanuel', 'Fonseca', 'fonseca@gmail.com', 'musician', 'f22', 'd6518aa213ea4939fde8bd047ca8fb775cac36bf95d26f28cf2c77f2eb
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Eduardo', 'Santos', 'eduardo@gmail.com', 'oncologist', 'edu', '4a53d7664413d4e5eefff5ad99e59ac1657b1cbdd6f1acbeea2a5f683
insert into "user" (first_name, last_name, email, bio, username, password)
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Maria João', 'Gonçalves', 'joao@gmail.com', 'jeweler', 'joao1', '6354f4d43066bed4fe6ad569ee99bf27c759c1e90b91ee5aef80b92
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Rui', 'Brás', 'ruibras@gmail.com', 'lawer', 'lawer', '00efdad92afe16836c0a2fccd7c4ac73e65f678691a98d031f74cc7032f00b3f')
    insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Leonor', 'Pinto', 'leonorpinto@gmail.com', 'dentist', 'leonor', '7a06c3703cc09df6281b90b8011a154f4e73a918bfe4305cfbd7138
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Luís', 'Ferreira', 'ferreira@gmail.com', 'magician', 'mag', 'f8685538bda2d6dc0d6d013011da6c986f1646c66f1b7a09035fc199d77
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Rui', 'Brás', 'bras67@gmail.com', 'lawer', 'rui01', '00efdad92afe16836c0a2fccd7c4ac73e65f678691a98d031f74cc7032f00b3f');
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Carlos', 'Cruz', '20cruz@gmail.com', 'fisheman', 'cruz', '2faecef2c88d125f3ba7dcac070b493f99d701ff43070c298cfa20a9f1459f
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Teodora', 'Vieira', 'teodora@gmail.com', 'cattle breeder', 'tvieira', 'e4a80fee9a4d8f87b0a5a408103ad723e2f83db981cdb10ef
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Silvia', 'Rodrigues', 'silvia@gmail.com', 'librarian', 'silr', 'f820f43abb76bc0ff9a3a1b5ddbea160efa28921723b601caf746533
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('amilcar', 'Dantas', 'dantas2@hotmail.com', 'make-up artist', 'artist', 'ed5ae5a93a86c36e004d985cd801b3390dc91e4dda5c6d59
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Laurinda', 'Marques', 'marques@gmail.com', 'gynaecologist', 'gy20', '54ada40a5452d89d06fdfc1f3ac106a8ee360a6437cb3749ac1
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Albino', 'Costa', 'costa@gmail.com', 'chmist', 'costa10', '4e73b8eacfd31c5e8b51eb897694758b59d8d562ed684ee18783873ebe967
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Raul', 'Ferreira', 'raul01@gmail.com', 'librarian', 'ra20', '106dde628c0da1d7d96851ca4110e68d31513bec50a905597a74feafe45
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('António', 'Carvalho', 'carvalho@gmail.com', 'barber', 'carv01', '8287464bb9bba50593d53d95ced168b43db7db958d1263544a6d6e3
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Cândido', 'Faria', 'faria@gmail.com', 'carpenter', 'faria20', '365e72262bed1b6afb91ff1c7107b12bd7206d1f7a8c0c70217c57af5
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Cosme', 'Dias', 'dias@gmail.com', 'adminstrator', 'cosm01', '9911486dbd7cbe092287bb63e1ed44fee854d215eff4979b9b9ea837890
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Juliana', 'Pinto', 'pinto4@gmail.com', 'beautician', 'juli', 'b04a508509a4b3cd0aa7a5a5824c3bce43e8512572cbc35506a255f6a5
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Emanuel', 'Fonseca', 'efonseca@gmail.com', 'musician', 'efonseca', 'd6518aa213ea4939fde8bd047ca8fb775cac36bf95d26f28cf2c
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Francisco', 'Noval', 'xicotruques@gmail.com', 'Handball coach, 20y', 'truques5', '172ba889dbec45e652e5ce30bd7a9ae195879c
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Gonçalo', 'Nogueira', 'salinhonogueira@gmail.com', 'Try a little harder to be a little better', 'salinho5', '172ba889dbe
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Joao', 'Torgal', 'joaotorgal@gmail.com', '21y, bl4z3 it', 'torgal5', '172ba889dbec45e652e5ce30bd7a9ae195879ccfba508a8897
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Luis', 'Fernandes', 'luisfernandes@gmail.com', 'Engineering student, NEEGIUM', 'luisfernandes5', '172ba889dbec45e652e5ce
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Raquel', 'Fonseca', 'carlafonseca@gmail.com', '21y, FEUP, Portugal', 'raquelfonseca5', '172ba889dbec45e652e5ce30bd7a9ae1
insert into "user" (first_name, last_name, email, bio, username, password)
    values ('Vitor', 'Goncalves', 'vitorturrinheira@gmail.com', '20y, FEUP, Pardelhas', 'vitorturrinheira6', 'ed5ae5a93a86c36e004d985

--label
insert into label (name) values ('environment');
insert into label (name) values ('accounting');
insert into label (name) values ('economics');
insert into label (name) values ('companies');
insert into label (name) values ('psychology');
insert into label (name) values ('engineering');
insert into label (name) values ('energy');
insert into label (name) values ('agriculture');
insert into label (name) values ('education');
insert into label (name) values ('sustainable');
insert into label (name) values ('development');
insert into label (name) values ('society');
insert into label (name) values ('architecture');
insert into label (name) values ('ecology');
insert into label (name) values ('culture');
insert into label (name) values ('science');
insert into label (name) values ('consumerism');
insert into label (name) values ('recycling');
insert into label (name) values ('virus');
insert into label (name) values ('sports');
insert into label (name) values ('philosophy');
insert into label (name) values ('music');
insert into label (name) values ('science');
insert into label (name) values ('cinema');
insert into label (name) values ('photography');
insert into label (name) values ('computers');
insert into label (name) values ('programming');
insert into label (name) values ('tv');
insert into label (name) values ('gaming');
insert into label (name) values ('macOS');
insert into label (name) values ('windows');
insert into label (name) values ('linux');
insert into label (name) values ('geography');
insert into label (name) values ('coronavirus');
insert into label (name) values ('jazz');
insert into label (name) values ('politics');
insert into label (name) values ('football');
insert into label (name) values ('cr7');
insert into label (name) values ('health');
insert into label (name) values ('technology');
insert into label (name) values ('psychology');
insert into label (name) values ('java');
insert into label (name) values ('c++');
insert into label (name) values ('python');
insert into label (name) values ('hockey');
insert into label (name) values ('basketball');
insert into label (name) values ('rock');
insert into label (name) values ('pop');
insert into label (name) values ('english');

--notification
insert into notification (content, date, viewed, user_id) values ('A user has answered a question of yours', '2019-12-1', FALSE, 20);
insert into notification (content, date, viewed, user_id) values ('A user has answered a question of yours', '2019-12-2', FALSE, 19);
insert into notification (content, date, viewed, user_id) values ('A user has answered a question you follow', '2019-12-3', FALSE, 18
insert into notification (content, date, viewed, user_id) values ('A user has answered a question you follow', '2019-12-4', FALSE, 17
insert into notification (content, date, viewed, user_id) values ('A user has answered a question of yours', '2019-12-5', TRUE, 16);
insert into notification (content, date, viewed, user_id) values ('A user has answered a question you follow', '2019-12-6', TRUE, 15)
insert into notification (content, date, viewed, user_id) values ('A user has answered a question you follow', '2019-12-7', FALSE, 14
insert into notification (content, date, viewed, user_id) values ('A user has answered a question you follow', '2019-12-8', TRUE, 13)
insert into notification (content, date, viewed, user_id) values ('A user has answered a question of yours', '2019-12-9', TRUE, 12);
insert into notification (content, date, viewed, user_id) values ('A user has answered a question of yours', '2019-12-10', FALSE, 11)
insert into notification (content, date, viewed, user_id) values ('A user has answered a question of yours', '2019-12-11', TRUE, 10);
insert into notification (content, date, viewed, user_id) values ('A user has answered a question of yours', '2019-12-12', TRUE, 9);
insert into notification (content, date, viewed, user_id) values ('A user has answered a question you follow', '2019-12-13', FALSE, 8
insert into notification (content, date, viewed, user_id) values ('A user has answered a question you follow', '2019-12-1', TRUE, 7);
insert into notification (content, date, viewed, user_id) values ('A user has answered a question of yours', '2019-12-1', TRUE, 6);
insert into notification (content, date, viewed, user_id) values ('A user has answered a question of yours', '2019-12-1', FALSE, 5);
insert into notification (content, date, viewed, user_id) values ('A user has answered a question you follow', '2019-12-1', TRUE, 4);
insert into notification (content, date, viewed, user_id) values ('A user has answered a question of yours', '2019-12-1', TRUE, 3);
insert into notification (content, date, viewed, user_id) values ('A user has answered a question you follow', '2019-12-1', FALSE, 2)
insert into notification (content, date, viewed, user_id) values ('A user has answered a question you follow', '2019-12-1', TRUE, 1);

--user_management
insert into user_management (status, date_last_changed, user_id) values ('administrator', '2020-03-01', 1);
insert into user_management (status, date_last_changed, user_id) values ('administrator', '2020-03-01', 2);
insert into user_management (status, date_last_changed, user_id) values ('administrator', '2020-03-01', 3);
insert into user_management (status, date_last_changed, user_id) values ('administrator', '2020-03-01', 4);
insert into user_management (status, date_last_changed, user_id) values ('moderator', '2019-04-10', 5);
insert into user_management (status, date_last_changed, user_id) values ('moderator', '2019-04-10', 6);
insert into user_management (status, date_last_changed, user_id) values ('moderator', '2019-04-10', 7);
insert into user_management (status, date_last_changed, user_id) values ('moderator', '2019-04-10', 8);
insert into user_management (status, date_last_changed, user_id) values ('moderator', '2019-04-10', 9);
insert into user_management (status, date_last_changed, user_id) values ('moderator', '2019-04-10', 10);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 11);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 12);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 13);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 14);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 15);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 16);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 17);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 18);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-03-10', 19);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 20);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 21);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 22);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 23);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 24);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 25);
insert into user_management (status, date_last_changed, user_id) values ('user', '2018-04-10', 26);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 27);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 28);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 29);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 30);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 31);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-02-10', 32);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 33);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 34);
insert into user_management (status, date_last_changed, user_id) values ('user', '2017-04-10', 35);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-02-14', 36);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 37);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 38);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-01', 39);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 40);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 41);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 42);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 43);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 44);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 45);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 46);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 47);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-02-09', 48);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 49);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 50);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 51);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 52);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 53);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 54);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 55);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 56);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 57);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 58);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 59);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 60);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 61);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-02-01', 62);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 63);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 64);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 65);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 66);
insert into user_management (status, date_last_changed, user_id) values ('user', '2019-04-10', 67);

--question
insert into question (user_id, title, description, nr_likes, nr_dislikes, question_date)
    values (4, 'Name some great method actors', 'What are some fascinating examples of how an actor got into character for a movie ro
insert into question (user_id, title, description, nr_likes, nr_dislikes, question_date)
    values (2, 'Are intelligent and wise people the same?', 'What is the difference between an intelligent person and a wise person?'
insert into question (user_id, title, description, nr_likes, nr_dislikes, question_date)
    values (1, 'Who are the biggest jerks in music?', 'Can you name some musicians that are particularly obnoxious? Who are the bigge
insert into question (user_id, title, description, nr_likes, nr_dislikes, question_date)
    values (3, 'Is there a difference between every day and everyday?', 'This an english language question that has always tormented
insert into question (user_id, title, description, nr_likes, nr_dislikes, question_date)
    values (6, 'Will science reconcile with religion in the future?', 'Just wondering if I could ever pair the certainty of science w
insert into question (user_id, title, description, nr_likes, nr_dislikes, question_date)
    values (7, 'Can someone explain to me the AirPod hype?', 'Why are Apples AirPods so hyped up when fake Airpods work just the same
insert into question (user_id, title, description, nr_likes, nr_dislikes, question_date)
    values (5, 'Running a program from within java code', 'What is the simplest way to call a program from with a piece of Java code?
insert into question (user_id, title, description, nr_likes, nr_dislikes, question_date)
    values (8, 'What is the best TV series and why?', 'I would like to really get into a tv show, but I only want to watch the best o
insert into question (user_id, title, description, question_date)
    values (10, 'What country’s flag is the most controversial?', 'This may seem odd, but I would like to hear the internets opinion.
insert into question (user_id, title, description, question_date)
    values (9, 'Who is handling the coronavirus outbreak better, the United States or Canada?', 'Im a proud American and Ill be damne
insert into question (user_id, title, description, question_date)
    values (40, 'Paging a collection with LINQ', 'How do you page through a collection in LINQ given that you have a startIndex and a
insert into question (user_id, title, description, question_date)
    values (45, 'Microsoft office 2007 file type, mime types and identifying characters', 'Where can I find a list of all of the MIME
insert into question (user_id, title, description, question_date)
    values (46, 'Internationalization in your projects', 'How have you implemented Internationalization (i18n) in actual projects you
insert into question (user_id, title, description, question_date)
    values (47, 'English (language)', 'In the UK, what is the correct way to order food, coffee, etc."can have a latte" or " I" have 
insert into question (user_id, title, description, question_date)
    values (50, 'How do restaurants cook pasta so fast?', 'Can please someone explain me how can restaurants cook all types of food s
insert into question (user_id, title, description, question_date)
    values (60, 'British Royalty', 'Who is allowed to call the Bristish Queen by her first name?', '2019-04-12');
insert into question (user_id, title, description, question_date)
    values (42, 'Why do many Italian pizzas have such large crusts even though it is not healthy and has no taste?', 'I could really 
insert into question (user_id, title, description, question_date)
    values (30, 'Express How You Feel', 'Why is Italy still getting thousands of coronavirus cases per day if they are locked down fo
insert into question (user_id, title, description, question_date)
    values (20, 'Covid-19 in Italy', 'Why is the coronavirus so bad in Italy?', '2019-03-15');
insert into question (user_id, title, description, question_date)
    values (41, 'Studying in Japan', 'What schools accepted/rejected you (April 2020)?', '2019-09-10');
insert into question (user_id, title, description, question_date)
    values (25, 'New York architecture', 'Why are New York brownstones elevated so high above the street?', '2019-03-15');
insert into question (user_id, title, description, question_date)
    values (42, 'Photography and Lady Gaga', 'What is an interesting photo of Lady Gaga in concert?', '2018-01-10');
insert into question (user_id, title, description, question_date)
    values (26, 'Professionals Cooking', 'Why professional chefs always seem to use an excessive amount of butter and oil? Surely the
insert into question (user_id, title, description, question_date)
    values (43, 'Queen and Freddie Mercury', 'How did Freddie Mercury manage to sound so powerful and stable in all parts of his voca
insert into question (user_id, title, description, question_date)
    values (44, 'What is the oldest age a K-pop idol ever debuted?', 'I really want to know this.',  '2019-12-15');
insert into question (user_id, title, description, question_date)
    values (45, 'Difference between Windows laptops e MacBooks?', 'Why cannot Windows laptops have a similar build quality to MacBook
insert into question (user_id, title, description, question_date)
    values (46, 'History', 'What was the most clichéd moment in history?', '2018-11-10');
insert into question (user_id, title, description, question_date)
    values (47, 'Tell me your opinion', 'What is a historical fact that would sadden most people if they found out?', '2019-12-12');
insert into question (user_id, title, description, question_date)
    values (49, 'History', 'Who are some historical figures wrongly portrayed in popular culture?', '2019-09-01');
insert into question (user_id, title, description, question_date)
    values (59, 'Popular cliches in movies', 'What is a movie cliche you never get tired of seeing?', '2019-08-01');

--answer
insert into answer (user_id, question_id, answer_date, content, nr_likes, nr_dislikes, marked_answer)
    values (1, 1, '2019-12-13', 'Undoubtedly the worst form of over-the-top method acting is the kind that a) makes you look like an 
insert into answer (user_id, question_id, answer_date, content, nr_likes, nr_dislikes, marked_answer)
    values (3, 2, '2019-12-01', 'An intelligent person knows so much that it knows nothing, leaves you hanging upside down, mouthing 
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (4, 3, '2020-01-12', 'There are lots of jerks in music. People like Gene Simmons and Morrissey are well-known jerks.', FAL
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (2, 4, '2019-05-01', 'I’m glad you realize there is a difference, because many people don’t. When you do something daily, 
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (5, 5, '2020-02-05', 'There never was a breach. BOTH two sides of the SAME coin. Science asks "how", religion asks "why".'
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (6, 6, '2019-10-01', 'First of all, Apple AirPods are not hype, they are the real deal with near unparalleled Bluetooth pa
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (7, 7, '2020-11-05', 'Take a look at Process and Runtime classes. Keep in mind that what you are trying to accomplish is p
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (9, 8, '2020-03-26', 'BREAKING BAD! This is the best TV show on earth. It builds up slowly but methodically to expand over
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (8, 9, '2019-11-06', 'Japans rising sun flag. The Rising Sun Flag is the military flag of Japan. It was used as the ensign
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (10, 10, '2020-03-30', 'it would depend upon your definition of better, is it better to have a low figure because you have
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (11, 11, '2019-06-04', 'A few months back I wrote a blog post about Fluent Interfaces and LINQ which used an Extension Met
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (20, 11, '2019-06-05', 'It is very simple with the Skip and Take extension methods.', FALSE);
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (22, 12, '2019-06-14', 'Office 2007 MIME Types for IIS', FALSE);
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (22, 13, '2019-06-03', 'Character Sets: Unicode is great, but you cant get away with ignoring other character sets. The de
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (23, 13, '2019-06-04', 'Having a PHP and MySQL Application that works well with German and French, but now needs to suppor
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (31, 14, '2020-02-03', 'The former is a little more polite, although “could I have a latte, please?” would be better.', FA
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (48, 15, '2020-02-03', 'I’m not proud to know this. And I would not recommend it to my worse enemy. We boiled the pasta in
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (1, 16, '2019-06-12', 'It is well known that President Nelson Mandela always called Her Majesty by her first name, Elizabe
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (43, 17, '2019-06-14', 'Let’s think about this a little bit… The crust isn’t healthy? Magically, the amount of crust that
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (2, 19, '2019-06-14', 'Italians may have drawn the shortest straw. There are solid reasons to understand why the coronavir
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (2, 21, '2019-04-01', 'Knowledgeable on all aspects of the residential and commercial real estate industry in New York Cit
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (13, 22, '2018-01-10', 'Especially if you haven’t been to one of her shows, you might find this very interesting.', FALSE)
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (65, 23, '2019-03-15', 'Oh, it gets worse. A friend of mine is married to a chef at a highly regarded hotel in Manchester.
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (41, 24, '2019-06-14', 'Wish he was around to tell us all about it. So do scientists who study the human voice. Several st
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (63, 26, '2019-12-15', 'I recently got a new job and due to software requirements, my work computer was forced to be a Win
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (44, 28, '2019-12-31', 'That is a really good question. I hope someone asks.', FALSE);
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (34, 29, '2019-12-03', 'That is a really good question. I hope someone asks.', FALSE);
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (31, 30, '2019-12-03', 'Bar silence- The protagonist walks into a bar and all of the patrons stop talking and look at him
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (34, 30, '2019-08-03', 'That is a really good question. I hope someone asks.', FALSE);
insert into answer (user_id, question_id, answer_date, content, marked_answer)
    values (39, 30, '2019-12-03', 'I dont remember right now.', FALSE);


--comment
insert into comment (user_id, question_id, answer_id, comment_date, content)
    values (3, 1, NULL, '2019-12-07', 'I believe it would also be interesting if people commented on the worse method actors as well.
insert into comment (user_id, question_id, answer_id, comment_date, content)
    values (4, 10, NULL, '2020-03-21', 'You should not compare the health strength of two countries during these hard times! Dislike!
insert into comment (user_id, question_id, answer_id, comment_date, content)
    values (1, NULL, 28, '2020-03-27', 'I agree! Awesome dialogue, great choice of actors and awesome camera shots. The perfect drama
insert into comment (user_id, question_id, answer_id, comment_date, content)
    values (6, 9, NULL, '2020-03-28', 'Well, that is a rather wird question, but in a good way though.');
insert into comment (user_id, question_id, answer_id, comment_date, content)
    values (9, NULL, 2, '2020-03-28', 'Impecable! What a good answer. Like!');
insert into comment (user_id, question_id, answer_id, comment_date, content)
    values (43, 13, NULL, '2019-06-03', 'Good point, and its not just the order. You might have trouble with grammatical agreements b
insert into comment (user_id, question_id, answer_id, comment_date, content)
    values (47, NULL, 16, '2020-02-03', 'Oh, thank you, thank you for pointing out that horror which so aggravates me “Can I get a la
insert into comment (user_id, question_id, answer_id, comment_date, content)
    values (9, 23, NULL, '2020-03-28', 'Salt and acid are powerful flavour enhancers. You don’t taste the salt nor the acid until you
insert into comment (user_id, question_id, answer_id, comment_date, content)
    values (1, NULL, 26, '2020-03-11', 'It’s like the design team found a Macbook Pro and decided to mess it’s best features up and a
insert into comment (user_id, question_id, answer_id, comment_date, content)
    values (30, NULL, 2, '2020-04-02', 'Thank you for such a good question. I was thinking on the same thing.');

--vote
--question 1
insert into vote ("vote", user_id, question_id) values (TRUE, 1, 1);
insert into vote ("vote", user_id, question_id) values (TRUE, 2, 1);
insert into vote ("vote", user_id, question_id) values (TRUE, 3, 1);
insert into vote ("vote", user_id, question_id) values (TRUE, 5, 1);
insert into vote ("vote", user_id, question_id) values (TRUE, 6, 1);
insert into vote ("vote", user_id, question_id) values (TRUE, 7, 1);
insert into vote ("vote", user_id, question_id) values (TRUE, 8, 1);
insert into vote ("vote", user_id, question_id) values (FALSE, 9, 1);
insert into vote ("vote", user_id, question_id) values (FALSE, 10, 1);
--question 2
insert into vote ("vote", user_id, question_id) values (TRUE, 1, 2);
insert into vote ("vote", user_id, question_id) values (TRUE, 3, 2);
insert into vote ("vote", user_id, question_id) values (TRUE, 4, 2);
insert into vote ("vote", user_id, question_id) values (TRUE, 5, 2);
insert into vote ("vote", user_id, question_id) values (TRUE, 6, 2);
insert into vote ("vote", user_id, question_id) values (TRUE, 7, 2);
insert into vote ("vote", user_id, question_id) values (TRUE, 8, 2);
insert into vote ("vote", user_id, question_id) values (TRUE, 9, 2);
insert into vote ("vote", user_id, question_id) values (TRUE, 10, 2);
insert into vote ("vote", user_id, question_id) values (TRUE, 11, 2);
insert into vote ("vote", user_id, question_id) values (FALSE, 12, 2);
insert into vote ("vote", user_id, question_id) values (FALSE, 13, 2);
insert into vote ("vote", user_id, question_id) values (FALSE, 14, 2);
insert into vote ("vote", user_id, question_id) values (FALSE, 15, 2);
insert into vote ("vote", user_id, question_id) values (FALSE, 16, 2);
--question 3
insert into vote ("vote", user_id, question_id) values (TRUE, 2, 3);
insert into vote ("vote", user_id, question_id) values (TRUE, 3, 3);
insert into vote ("vote", user_id, question_id) values (TRUE, 4, 3);
insert into vote ("vote", user_id, question_id) values (TRUE, 5, 3);
insert into vote ("vote", user_id, question_id) values (TRUE, 6, 3);
insert into vote ("vote", user_id, question_id) values (TRUE, 7, 3);
insert into vote ("vote", user_id, question_id) values (TRUE, 8, 3);
insert into vote ("vote", user_id, question_id) values (TRUE, 9, 3);
insert into vote ("vote", user_id, question_id) values (TRUE, 10, 3);
insert into vote ("vote", user_id, question_id) values (TRUE, 11, 3);
insert into vote ("vote", user_id, question_id) values (FALSE, 12, 3);
insert into vote ("vote", user_id, question_id) values (FALSE, 13, 3);
insert into vote ("vote", user_id, question_id) values (FALSE, 14, 3);
insert into vote ("vote", user_id, question_id) values (FALSE, 15, 3);
insert into vote ("vote", user_id, question_id) values (FALSE, 16, 3);
--question 4
insert into vote ("vote", user_id, question_id) values (TRUE, 1, 4);
insert into vote ("vote", user_id, question_id) values (TRUE, 2, 4);
insert into vote ("vote", user_id, question_id) values (TRUE, 4, 4);
insert into vote ("vote", user_id, question_id) values (TRUE, 5, 4);
insert into vote ("vote", user_id, question_id) values (TRUE, 6, 4);
insert into vote ("vote", user_id, question_id) values (TRUE, 7, 4);
insert into vote ("vote", user_id, question_id) values (TRUE, 8, 4);
insert into vote ("vote", user_id, question_id) values (TRUE, 9, 4);
insert into vote ("vote", user_id, question_id) values (TRUE, 10, 4);
insert into vote ("vote", user_id, question_id) values (TRUE, 11, 4);
insert into vote ("vote", user_id, question_id) values (FALSE, 12, 4);
insert into vote ("vote", user_id, question_id) values (FALSE, 13, 4);
insert into vote ("vote", user_id, question_id) values (FALSE, 14, 4);
insert into vote ("vote", user_id, question_id) values (FALSE, 15, 4);
insert into vote ("vote", user_id, question_id) values (FALSE, 16, 4);
insert into vote ("vote", user_id, question_id) values (FALSE, 17, 4);
insert into vote ("vote", user_id, question_id) values (FALSE, 18, 4);
insert into vote ("vote", user_id, question_id) values (FALSE, 19, 4);
insert into vote ("vote", user_id, question_id) values (FALSE, 20, 4);
insert into vote ("vote", user_id, question_id) values (FALSE, 21, 4);
insert into vote ("vote", user_id, question_id) values (FALSE, 22, 4);
--question 5
insert into vote ("vote", user_id, question_id) values (TRUE, 1, 5);
insert into vote ("vote", user_id, question_id) values (TRUE, 2, 5);
insert into vote ("vote", user_id, question_id) values (FALSE, 3, 5);
insert into vote ("vote", user_id, question_id) values (FALSE, 4, 5);
insert into vote ("vote", user_id, question_id) values (FALSE, 5, 5);
--question 6
insert into vote ("vote", user_id, question_id) values (TRUE, 1, 6);
insert into vote ("vote", user_id, question_id) values (TRUE, 2, 6);
insert into vote ("vote", user_id, question_id) values (TRUE, 3, 6);
insert into vote ("vote", user_id, question_id) values (FALSE, 4, 6);
insert into vote ("vote", user_id, question_id) values (FALSE, 5, 6);
--question 7
insert into vote ("vote", user_id, question_id) values (TRUE, 1, 7);
insert into vote ("vote", user_id, question_id) values (TRUE, 2, 7);
insert into vote ("vote", user_id, question_id) values (TRUE, 3, 7);
insert into vote ("vote", user_id, question_id) values (TRUE, 4, 7);
insert into vote ("vote", user_id, question_id) values (TRUE, 6, 7);
insert into vote ("vote", user_id, question_id) values (TRUE, 7, 7);
insert into vote ("vote", user_id, question_id) values (TRUE, 8, 7);
insert into vote ("vote", user_id, question_id) values (FALSE, 9, 7);
insert into vote ("vote", user_id, question_id) values (FALSE, 10, 7);
insert into vote ("vote", user_id, question_id) values (FALSE, 11, 7);
insert into vote ("vote", user_id, question_id) values (FALSE, 12, 7);
insert into vote ("vote", user_id, question_id) values (FALSE, 13, 7);
--question 8
insert into vote ("vote", user_id, question_id) values (TRUE, 1, 8);
insert into vote ("vote", user_id, question_id) values (TRUE, 2, 8);
insert into vote ("vote", user_id, question_id) values (TRUE, 3, 8);
insert into vote ("vote", user_id, question_id) values (TRUE, 4, 8);
insert into vote ("vote", user_id, question_id) values (TRUE, 5, 8);
insert into vote ("vote", user_id, question_id) values (TRUE, 6, 8);
insert into vote ("vote", user_id, question_id) values (TRUE, 7, 8);
insert into vote ("vote", user_id, question_id) values (TRUE, 9, 8);
insert into vote ("vote", user_id, question_id) values (TRUE, 10, 8);
insert into vote ("vote", user_id, question_id) values (TRUE, 11, 8);
insert into vote ("vote", user_id, question_id) values (FALSE, 12, 8);
insert into vote ("vote", user_id, question_id) values (FALSE, 13, 8);
insert into vote ("vote", user_id, question_id) values (FALSE, 14, 8);
insert into vote ("vote", user_id, question_id) values (FALSE, 15, 8);
insert into vote ("vote", user_id, question_id) values (FALSE, 16, 8);
--answer 1
insert into vote ("vote", user_id, answer_id) values (TRUE, 2, 1);
insert into vote ("vote", user_id, answer_id) values (TRUE, 3, 1);
insert into vote ("vote", user_id, answer_id) values (TRUE, 4, 1);
insert into vote ("vote", user_id, answer_id) values (FALSE, 5, 1);
insert into vote ("vote", user_id, answer_id) values (FALSE, 6, 1);
--answer 2
insert into vote ("vote", user_id, answer_id) values (TRUE, 1, 2);
insert into vote ("vote", user_id, answer_id) values (TRUE, 2, 2);
insert into vote ("vote", user_id, answer_id) values (TRUE, 4, 2);
insert into vote ("vote", user_id, answer_id) values (TRUE, 5, 2);
insert into vote ("vote", user_id, answer_id) values (TRUE, 6, 2);
insert into vote ("vote", user_id, answer_id) values (TRUE, 7, 2);
insert into vote ("vote", user_id, answer_id) values (TRUE, 8, 2);
insert into vote ("vote", user_id, answer_id) values (FALSE, 9, 2);
insert into vote ("vote", user_id, answer_id) values (FALSE, 10, 2);
insert into vote ("vote", user_id, answer_id) values (FALSE, 11, 2);
insert into vote ("vote", user_id, answer_id) values (FALSE, 12, 2);

--report
insert into report (reporter_id, user_id, question_id, answer_id, comment_id, report_date, description)
insert into report (reporter_id, user_id, question_id, answer_id, comment_id, report_date, description)
    values (5, 10, NULL, NULL, NULL, '2020-04-10', 'This user did not respect the privacy policy of the site');
insert into report (reporter_id, user_id, question_id, answer_id, comment_id, report_date, description)
    values (20, NULL, 3, NULL, NULL, '2020-04-10', 'This question is disrespectful');
insert into report (reporter_id, user_id, question_id, answer_id, comment_id, report_date, description)
    values (21, NULL, NULL, 9, NULL, '2020-01-12', 'This answer is racist');
insert into report (reporter_id, user_id, question_id, answer_id, comment_id, report_date, description)
    values (22, NULL, NULL, NULL, 9, '2020-03-11', 'This comment is totally unhelpful');

--report_status
insert into report_status (report_id, state, comment, responsible_user)
    values (1, 'unresolved', 'Waiting for review', 1);
insert into report_status (report_id, state, comment, responsible_user)
    values (2, 'unresolved', 'Waiting for review', 2);
insert into report_status (report_id, state, comment, responsible_user)
    values (3, 'reviewing', 'Working on it', 5);
insert into report_status (report_id, state, comment, responsible_user)
    values (4, 'resolved', 'Finished review, fixed problem', 6);

--question_following
insert into question_following (user_id, question_id) values (1, 2);
insert into question_following (user_id, question_id) values (2, 1);
insert into question_following (user_id, question_id) values (3, 4);
insert into question_following (user_id, question_id) values (4, 3);
insert into question_following (user_id, question_id) values (5, 6);
insert into question_following (user_id, question_id) values (6, 5);
insert into question_following (user_id, question_id) values (7, 8);
insert into question_following (user_id, question_id) values (8, 7);
insert into question_following (user_id, question_id) values (9, 10);
insert into question_following (user_id, question_id) values (10, 9);
insert into question_following (user_id, question_id) values (11, 12);
insert into question_following (user_id, question_id) values (12, 13);
insert into question_following (user_id, question_id) values (13, 14);
insert into question_following (user_id, question_id) values (14, 15);
insert into question_following (user_id, question_id) values (15, 16);
insert into question_following (user_id, question_id) values (16, 17);
insert into question_following (user_id, question_id) values (17, 18);
insert into question_following (user_id, question_id) values (18, 19);
insert into question_following (user_id, question_id) values (19, 20);
insert into question_following (user_id, question_id) values (20, 21);
insert into question_following (user_id, question_id) values (21, 22);
insert into question_following (user_id, question_id) values (22, 23);
insert into question_following (user_id, question_id) values (23, 24);
insert into question_following (user_id, question_id) values (24, 25);
insert into question_following (user_id, question_id) values (25, 26);
insert into question_following (user_id, question_id) values (26, 27);
insert into question_following (user_id, question_id) values (27, 28);
insert into question_following (user_id, question_id) values (28, 29);
insert into question_following (user_id, question_id) values (29, 30);
insert into question_following (user_id, question_id) values (30, 21);
insert into question_following (user_id, question_id) values (31, 22);
insert into question_following (user_id, question_id) values (33, 24);
insert into question_following (user_id, question_id) values (34, 25);
insert into question_following (user_id, question_id) values (35, 26);
insert into question_following (user_id, question_id) values (36, 27);
insert into question_following (user_id, question_id) values (37, 28);
insert into question_following (user_id, question_id) values (38, 29);
insert into question_following (user_id, question_id) values (39, 30);
insert into question_following (user_id, question_id) values (40, 21);
insert into question_following (user_id, question_id) values (41, 22);
insert into question_following (user_id, question_id) values (42, 23);
insert into question_following (user_id, question_id) values (43, 24);
insert into question_following (user_id, question_id) values (44, 25);
insert into question_following (user_id, question_id) values (45, 26);
insert into question_following (user_id, question_id) values (46, 27);
insert into question_following (user_id, question_id) values (47, 28);
insert into question_following (user_id, question_id) values (48, 29);
insert into question_following (user_id, question_id) values (48, 29);
insert into question_following (user_id, question_id) values (49, 30);
insert into question_following (user_id, question_id) values (50, 11);
insert into question_following (user_id, question_id) values (51, 12);
insert into question_following (user_id, question_id) values (52, 13);
insert into question_following (user_id, question_id) values (53, 14);
insert into question_following (user_id, question_id) values (54, 15);
insert into question_following (user_id, question_id) values (55, 16);
insert into question_following (user_id, question_id) values (56, 17);
insert into question_following (user_id, question_id) values (57, 18);
insert into question_following (user_id, question_id) values (58, 19);
insert into question_following (user_id, question_id) values (59, 20);
insert into question_following (user_id, question_id) values (60, 1);
insert into question_following (user_id, question_id) values (20, 1);
insert into question_following (user_id, question_id) values (22, 3);
insert into question_following (user_id, question_id) values (6, 7);
insert into question_following (user_id, question_id) values (7, 20);
insert into question_following (user_id, question_id) values (8, 23);
insert into question_following (user_id, question_id) values (9, 11);

--label_following
insert into label_following (user_id, label_id) values (1, 3);
insert into label_following (user_id, label_id) values (1, 5);
insert into label_following (user_id, label_id) values (1, 7);
insert into label_following (user_id, label_id) values (2, 7);
insert into label_following (user_id, label_id) values (2, 9);
insert into label_following (user_id, label_id) values (2, 11);
insert into label_following (user_id, label_id) values (3, 1);
insert into label_following (user_id, label_id) values (3, 2);
insert into label_following (user_id, label_id) values (3, 4);
insert into label_following (user_id, label_id) values (3, 6);
insert into label_following (user_id, label_id) values (4, 8);
insert into label_following (user_id, label_id) values (4, 10);
insert into label_following (user_id, label_id) values (4, 12);
insert into label_following (user_id, label_id) values (5, 13);
insert into label_following (user_id, label_id) values (5, 14);
insert into label_following (user_id, label_id) values (5, 15);
insert into label_following (user_id, label_id) values (6, 16);
insert into label_following (user_id, label_id) values (6, 17);
insert into label_following (user_id, label_id) values (6, 18);
insert into label_following (user_id, label_id) values (7, 19);
insert into label_following (user_id, label_id) values (7, 20);
insert into label_following (user_id, label_id) values (8, 21);
insert into label_following (user_id, label_id) values (8, 22);
insert into label_following (user_id, label_id) values (8, 23);
insert into label_following (user_id, label_id) values (9, 24);
insert into label_following (user_id, label_id) values (9, 25);
insert into label_following (user_id, label_id) values (9, 26);
insert into label_following (user_id, label_id) values (9, 27);
insert into label_following (user_id, label_id) values (9, 28);
insert into label_following (user_id, label_id) values (10, 29);
insert into label_following (user_id, label_id) values (10, 30);
insert into label_following (user_id, label_id) values (11, 3);
insert into label_following (user_id, label_id) values (11, 5);
insert into label_following (user_id, label_id) values (11, 7);
insert into label_following (user_id, label_id) values (12, 7);
insert into label_following (user_id, label_id) values (12, 9);
insert into label_following (user_id, label_id) values (12, 11);
insert into label_following (user_id, label_id) values (13, 1);
insert into label_following (user_id, label_id) values (13, 2);
insert into label_following (user_id, label_id) values (13, 4);
insert into label_following (user_id, label_id) values (13, 6);
insert into label_following (user_id, label_id) values (14, 8);
insert into label_following (user_id, label_id) values (14, 10);
insert into label_following (user_id, label_id) values (14, 12);
insert into label_following (user_id, label_id) values (15, 13);
insert into label_following (user_id, label_id) values (15, 14);
insert into label_following (user_id, label_id) values (15, 14);
insert into label_following (user_id, label_id) values (15, 15);
insert into label_following (user_id, label_id) values (16, 16);
insert into label_following (user_id, label_id) values (16, 17);
insert into label_following (user_id, label_id) values (16, 18);
insert into label_following (user_id, label_id) values (17, 19);
insert into label_following (user_id, label_id) values (17, 44);
insert into label_following (user_id, label_id) values (18, 21);
insert into label_following (user_id, label_id) values (18, 22);
insert into label_following (user_id, label_id) values (18, 23);
insert into label_following (user_id, label_id) values (19, 24);
insert into label_following (user_id, label_id) values (19, 25);
insert into label_following (user_id, label_id) values (19, 26);
insert into label_following (user_id, label_id) values (19, 27);
insert into label_following (user_id, label_id) values (19, 28);
insert into label_following (user_id, label_id) values (20, 29);
insert into label_following (user_id, label_id) values (20, 30);
insert into label_following (user_id, label_id) values (21, 3);
insert into label_following (user_id, label_id) values (21, 5);
insert into label_following (user_id, label_id) values (21, 7);
insert into label_following (user_id, label_id) values (22, 7);
insert into label_following (user_id, label_id) values (22, 9);
insert into label_following (user_id, label_id) values (22, 11);
insert into label_following (user_id, label_id) values (23, 1);
insert into label_following (user_id, label_id) values (23, 2);
insert into label_following (user_id, label_id) values (23, 4);
insert into label_following (user_id, label_id) values (23, 6);
insert into label_following (user_id, label_id) values (24, 45);
insert into label_following (user_id, label_id) values (24, 10);
insert into label_following (user_id, label_id) values (24, 12);
insert into label_following (user_id, label_id) values (25, 13);
insert into label_following (user_id, label_id) values (25, 14);
insert into label_following (user_id, label_id) values (25, 15);
insert into label_following (user_id, label_id) values (26, 16);
insert into label_following (user_id, label_id) values (26, 17);
insert into label_following (user_id, label_id) values (26, 18);
insert into label_following (user_id, label_id) values (27, 19);
insert into label_following (user_id, label_id) values (27, 48);
insert into label_following (user_id, label_id) values (28, 21);
insert into label_following (user_id, label_id) values (28, 22);
insert into label_following (user_id, label_id) values (28, 23);
insert into label_following (user_id, label_id) values (29, 24);
insert into label_following (user_id, label_id) values (29, 25);
insert into label_following (user_id, label_id) values (29, 26);
insert into label_following (user_id, label_id) values (29, 27);
insert into label_following (user_id, label_id) values (29, 28);
insert into label_following (user_id, label_id) values (30, 29);
insert into label_following (user_id, label_id) values (30, 30);
insert into label_following (user_id, label_id) values (31, 3);
insert into label_following (user_id, label_id) values (31, 5);
insert into label_following (user_id, label_id) values (31, 7);
insert into label_following (user_id, label_id) values (32, 7);
insert into label_following (user_id, label_id) values (32, 9);
insert into label_following (user_id, label_id) values (32, 11);
insert into label_following (user_id, label_id) values (33, 1);
insert into label_following (user_id, label_id) values (33, 2);
insert into label_following (user_id, label_id) values (33, 4);
insert into label_following (user_id, label_id) values (33, 6);
insert into label_following (user_id, label_id) values (34, 8);
insert into label_following (user_id, label_id) values (34, 10);
insert into label_following (user_id, label_id) values (34, 12);
insert into label_following (user_id, label_id) values (35, 13);
insert into label_following (user_id, label_id) values (35, 14);
insert into label_following (user_id, label_id) values (35, 15);
insert into label_following (user_id, label_id) values (36, 48);
insert into label_following (user_id, label_id) values (36, 17);
insert into label_following (user_id, label_id) values (36, 18);
insert into label_following (user_id, label_id) values (36, 18);
insert into label_following (user_id, label_id) values (37, 19);
insert into label_following (user_id, label_id) values (37, 20);
insert into label_following (user_id, label_id) values (38, 21);
insert into label_following (user_id, label_id) values (38, 22);
insert into label_following (user_id, label_id) values (38, 23);
insert into label_following (user_id, label_id) values (39, 24);
insert into label_following (user_id, label_id) values (39, 25);
insert into label_following (user_id, label_id) values (39, 26);
insert into label_following (user_id, label_id) values (39, 27);
insert into label_following (user_id, label_id) values (39, 28);
insert into label_following (user_id, label_id) values (40, 29);
insert into label_following (user_id, label_id) values (40, 30);
insert into label_following (user_id, label_id) values (41, 3);
insert into label_following (user_id, label_id) values (41, 5);
insert into label_following (user_id, label_id) values (41, 7);
insert into label_following (user_id, label_id) values (42, 7);
insert into label_following (user_id, label_id) values (42, 9);
insert into label_following (user_id, label_id) values (42, 36);
insert into label_following (user_id, label_id) values (43, 1);
insert into label_following (user_id, label_id) values (43, 2);
insert into label_following (user_id, label_id) values (43, 4);
insert into label_following (user_id, label_id) values (43, 6);
insert into label_following (user_id, label_id) values (44, 8);
insert into label_following (user_id, label_id) values (44, 10);
insert into label_following (user_id, label_id) values (44, 12);
insert into label_following (user_id, label_id) values (45, 13);
insert into label_following (user_id, label_id) values (45, 14);
insert into label_following (user_id, label_id) values (45, 15);
insert into label_following (user_id, label_id) values (46, 16);
insert into label_following (user_id, label_id) values (46, 17);
insert into label_following (user_id, label_id) values (46, 37);
insert into label_following (user_id, label_id) values (46, 18);
insert into label_following (user_id, label_id) values (47, 19);
insert into label_following (user_id, label_id) values (47, 20);
insert into label_following (user_id, label_id) values (48, 21);
insert into label_following (user_id, label_id) values (48, 22);
insert into label_following (user_id, label_id) values (48, 23);
insert into label_following (user_id, label_id) values (49, 24);
insert into label_following (user_id, label_id) values (49, 25);
insert into label_following (user_id, label_id) values (49, 26);
insert into label_following (user_id, label_id) values (49, 27);
insert into label_following (user_id, label_id) values (49, 28);
insert into label_following (user_id, label_id) values (50, 29);
insert into label_following (user_id, label_id) values (50, 30);
insert into label_following (user_id, label_id) values (51, 3);
insert into label_following (user_id, label_id) values (51, 5);
insert into label_following (user_id, label_id) values (51, 7);
insert into label_following (user_id, label_id) values (52, 7);
insert into label_following (user_id, label_id) values (52, 9);
insert into label_following (user_id, label_id) values (52, 11);
insert into label_following (user_id, label_id) values (53, 1);
insert into label_following (user_id, label_id) values (53, 2);
insert into label_following (user_id, label_id) values (53, 4);
insert into label_following (user_id, label_id) values (53, 6);
insert into label_following (user_id, label_id) values (54, 8);
insert into label_following (user_id, label_id) values (54, 10);
insert into label_following (user_id, label_id) values (54, 12);
insert into label_following (user_id, label_id) values (55, 13);
insert into label_following (user_id, label_id) values (55, 38);
insert into label_following (user_id, label_id) values (55, 15);
insert into label_following (user_id, label_id) values (56, 16);
insert into label_following (user_id, label_id) values (56, 17);
insert into label_following (user_id, label_id) values (56, 18);
insert into label_following (user_id, label_id) values (57, 19);
insert into label_following (user_id, label_id) values (57, 20);
insert into label_following (user_id, label_id) values (58, 21);
insert into label_following (user_id, label_id) values (58, 22);
insert into label_following (user_id, label_id) values (58, 40);
insert into label_following (user_id, label_id) values (59, 24);
insert into label_following (user_id, label_id) values (59, 25);
insert into label_following (user_id, label_id) values (59, 26);
insert into label_following (user_id, label_id) values (59, 27);
insert into label_following (user_id, label_id) values (59, 28);
insert into label_following (user_id, label_id) values (60, 29);
insert into label_following (user_id, label_id) values (60, 30);
insert into label_following (user_id, label_id) values (61, 3);
insert into label_following (user_id, label_id) values (61, 5);
insert into label_following (user_id, label_id) values (61, 7);
insert into label_following (user_id, label_id) values (62, 7);
insert into label_following (user_id, label_id) values (62, 9);
insert into label_following (user_id, label_id) values (62, 11);
insert into label_following (user_id, label_id) values (63, 1);
insert into label_following (user_id, label_id) values (63, 2);
insert into label_following (user_id, label_id) values (63, 4);
insert into label_following (user_id, label_id) values (63, 6);
insert into label_following (user_id, label_id) values (64, 8);
insert into label_following (user_id, label_id) values (64, 10);
insert into label_following (user_id, label_id) values (64, 12);
insert into label_following (user_id, label_id) values (65, 13);
insert into label_following (user_id, label_id) values (65, 14);
insert into label_following (user_id, label_id) values (65, 15);
insert into label_following (user_id, label_id) values (66, 16);
insert into label_following (user_id, label_id) values (66, 17);
insert into label_following (user_id, label_id) values (66, 41);
insert into label_following (user_id, label_id) values (66, 42);
insert into label_following (user_id, label_id) values (67, 43);
insert into label_following (user_id, label_id) values (67, 44);

--question_label
insert into question_label (question_id, label_id) values (1, 24);
insert into question_label (question_id, label_id) values (2, 16);
insert into question_label (question_id, label_id) values (3, 22);
insert into question_label (question_id, label_id) values (4, 49);
insert into question_label (question_id, label_id) values (5, 23);
insert into question_label (question_id, label_id) values (6, 26);
insert into question_label (question_id, label_id) values (7, 27);
insert into question_label (question_id, label_id) values (8, 28);
insert into question_label (question_id, label_id) values (9, 15);
insert into question_label (question_id, label_id) values (10, 34);
insert into question_label (question_id, label_id) values (11, 42);
insert into question_label (question_id, label_id) values (12, 26);
insert into question_label (question_id, label_id) values (13, 4);
insert into question_label (question_id, label_id) values (14, 49);
insert into question_label (question_id, label_id) values (15, 12);
insert into question_label (question_id, label_id) values (16, 12);
insert into question_label (question_id, label_id) values (17, 12);
insert into question_label (question_id, label_id) values (18, 19);
insert into question_label (question_id, label_id) values (19, 16);
insert into question_label (question_id, label_id) values (20, 9);
insert into question_label (question_id, label_id) values (21, 15);
insert into question_label (question_id, label_id) values (22, 25);
insert into question_label (question_id, label_id) values (23, 17);
insert into question_label (question_id, label_id) values (24, 22);
insert into question_label (question_id, label_id) values (25, 15);
insert into question_label (question_id, label_id) values (26, 49);
insert into question_label (question_id, label_id) values (27, 15);
insert into question_label (question_id, label_id) values (28, 5);
insert into question_label (question_id, label_id) values (29, 15);
insert into question_label (question_id, label_id) values (30, 24);
```

Revision history

1. First submission (30/03/2020).
2. Improved populate.sql (10/04/2020).
3. Fixed minor issues on transactions, like avoiding Isolation Level Repeatable Read using Read Uncommitted, and added transactions for notifications (11/04/2020).
4. Fixed minor issues on queries (12/04/2020).
5. Added 2 new triggers (11/05/2020).

GROUP2064, 12/04/2020

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