Project Management

A6: Integrity constraints. Indexes, triggers, user functions and database populated with data

This artefact contains the physical schema of the database, the identification and characterisation of the indexes, the support of data integrity rules with triggers and the definition of the database user-defined functions. This artefact also contains the database's workload as well as the complete database creation script, including all SQL necessary to define all integrity constraints, indexes and triggers.

1. Database workload

1.1 Estimate of tuples

Relation reference	Relation Name	Order of magnitude	Estimated growth
R01	user	Tens	Units per day
R02	project	Hundreds	Units per day
R03	sprint	Hundreds	Dozens per day
R04	task	Tens	Thousands per day
R05	thread	Hundreds	Dozens per day
R06	comment	Tens	Thousands per day
R07	category	Hundreds	Dozens per year
R08	project_members	Hundreds	Units per day
R09	administrator	Hundreds	Units per day
R10	report	Hundreds	Units per day
R11	notification	tens	Thousands per day
R12	invite	tens	Units per day
R13	taskStateRecord	tens	Thousands per day
R14	sprintStateRecord	tens	Units per week
R15	project_categories	hundreds	Dozens per day

1.2 Most frequent queries

Query reference	SELECT01
Query description	Check if User exists and if password is correct
Query frequency	Hundreds per day
SELECT * FROM "user" WHERE username = $username AND password = $password;	

Query reference	SELECT02
Query description	Lists User Reports
Query frequency	Units per day
SELECT * FROM Report WHERE reportType = 'UserReported';	

Query reference	SELECT03
Query description	Lists Comment Reports
Query frequency	Units per day
SELECT * FROM Report WHERE reportType = 'CommentReported';	

Query reference	SELECT04
Query description	List all Threads of one specific Project
Query frequency	Dozens per day
SELECT thread.name, "user".username, thread.date FROM thread, "user" WHERE thread.project_id = $project_id AND "user".id = thread.user_creator_id LIMIT 20 OFFSET $n;	

Query reference	SELECT05
Query description	Get thread information (title, user who created, description)
Query frequency	Dozens per day
SELECT thread.name, thread.description, "user".username, "user".image, thread.date FROM thread, "user" WHERE thread.id = $thread_id AND "user".id = thread.user_creator_id;	

Query reference	SELECT06
Query description	List all comments of a specific thread
Query frequency	Hundreds per day

```
SELECT comment.content, comment.date, "user".username, "user".image
FROM comment, "user"
WHERE comment.thread_id = $thread_id AND comment.user_id = "user".id
LIMIT 10 OFFSET $n;
```

Query reference	SELECT07
Query description	list all notifications of a specific client
Query frequency	Hundreds per day

```
SELECT * from Notification WHERE user_id = $user_id;
```

Query reference	SELECT08
Query description	List all projects of a specific user, with their info, nº of members and nº of sprints and role
Query frequency	Dozens per day

```
SELECT project.name, project.description, project_members.isCoordinator,
num.num_members, sprints.sprints_num
FROM "user", project_members, project
INNER JOIN
(SELECT project_id, COUNT(project_id) AS num_members
FROM project_members GROUP BY project_members.project_id) num
ON project.id = num.project_id
INNER JOIN
(SELECT project_id, COUNT(*) AS sprints_num FROM sprint
GROUP BY project_id) sprints
ON project.id = sprints.project_id
WHERE "user".username = $username AND project_members.user_id = "user".id
AND project_members.project_id = project.id AND num.project_id = project.id
LIMIT 5 OFFSET $n;
```

Query reference	SELECT09
Query description	Visitor version of a profile page, with all the projects of the user without the number of sprints
Query frequency	Dozens per day

```
SELECT project.name, project.description, project_members.isCoordinator,
num.num_members
```

```
FROM "user", project_members, project
INNER JOIN
(SELECT project_id, COUNT(project_id) AS num_members
FROM project_members GROUP BY project_members.project_id) num
ON project.id = num.project_id
WHERE "user".username = $username AND project_members.user_id = "user".id
AND project_members.project_id = project.id AND num.project_id = project.id
LIMIT 5 OFFSET $n;
```

Query reference	SELECT10
Query description	Get user information (email, name, image)
Query frequency	Hundreds per day

```
SELECT * FROM "user" WHERE username = $username;
```

Query reference	SELECT11
Query description	Get the number of tasks completed by a user in the current week
Query frequency	Hundreds per day

```
SELECT COUNT(*) FROM task_state_record
WHERE user_completed_id = $user_id AND state = 'Completed'
AND (SELECT extract('week' FROM task_state_record.date)) = (select extract('week' from current_date));
```

Query reference	SELECT12
Query description	Get the number of tasks completed by a user in the current month
Query frequency	Hundreds per day

```
SELECT COUNT(*) FROM task_state_record
WHERE user_completed_id = $user_id AND state = 'Completed'
AND (SELECT extract('month' FROM task_state_record.date)) = (select extract('month' from current_date));
```

Query reference	SELECT13
Query description	Get the number of sprints the user contributed to (by being assigned or completed them)
Query frequency	Hundreds per day

```
SELECT COUNT(task.sprint_id) FROM task_state_record, task
WHERE task_state_record.user_completed_id = $user_id
AND task_state_record.state != 'Created'
AND task_state_record.task_id = task.id;
```

Query reference	SELECT14
Query description	Search user projects, by role
Query frequency	Dozens per day

```
SELECT project.name, project.description, project_members.isCoordinator,
num.num_members, sprints.sprints_num
FROM "user", project_members, project
INNER JOIN
(SELECT project_id, COUNT(project_id) AS num_members
FROM project_members GROUP BY project_members.project_id) num
ON project.id = num.project_id
INNER JOIN
(SELECT project_id, COUNT(*) AS sprints_num FROM sprint
GROUP BY project_id) sprints
ON project.id = sprints.project_id
WHERE "user".id = $user_id AND project_members.user_id = "user".id
AND project_members.project_id = project.id AND num.project_id = project.id
AND project_members.isCoordinator = $isCoordinator
LIMIT 5 OFFSET $n;
```

Query reference	SELECT15
Query description	Search user projects, by project name or description
Query frequency	Dozens per day

```
SELECT project.name, project.description, project_members.isCoordinator,
num.num_members, sprints.sprints_num
FROM "user", project_members, project
INNER JOIN
(SELECT project_id, COUNT(project_id) AS num_members
FROM project_members GROUP BY project_members.project_id) num
ON project.id = num.project_id
INNER JOIN
(SELECT project_id, COUNT(*) AS sprints_num FROM sprint
```

```
GROUP BY project_id) sprints
ON project.id = sprints.project_id
WHERE "user".id = $user_id AND project_members.user_id = "user".id
AND project_members.project_id = project.id AND num.project_id = project.id
AND (project.name LIKE '%$search%' OR project.description LIKE '%$search%')
LIMIT 5 OFFSET $n;
```

Query reference	SELECT16
Query description	Search user projects, by category
Query frequency	Dozens per day

```
SELECT project.name, project.description, project_members.isCoordinator,
num.num_members, sprints.sprints_num
FROM "user", project_members, project_categories, category, project
INNER JOIN
(SELECT project_id, COUNT(project_id) AS num_members
FROM project_members GROUP BY project_members.project_id) num
ON project.id = num.project_id
INNER JOIN
(SELECT project_id, COUNT(*) AS sprints_num FROM sprint
GROUP BY project_id) sprints
ON project.id = sprints.project_id
WHERE "user".id = $user_id AND project_members.user_id = "user".id
AND project_members.project_id = project.id AND num.project_id = project.id
AND category.name LIKE '%$search%' AND category.id = project_categories.category_id
AND project_categories.project_id = project.id
LIMIT 5 OFFSET $n;
```

Query reference	SELECT17
Query description	show project info and categories
Query frequency	Dozens per day

```
SELECT name, description FROM project WHERE id = $project_id;

SELECT category.name FROM category, project_categories
WHERE category.id = project_categories.category_id AND project_categories.project_id =
$project_id;
```

Query reference	SELECT18
Query description	Search a team member or coordinator (Project Settings)
Query frequency	Dozens per week

```
SELECT "user".username, "user".image, project_members.isCoordinator
FROM project_members, "user"
WHERE project_members.project_id = $project_id AND project_members.user_id = "user".id
AND "user".username LIKE '%$search%'
LIMIT 10 OFFSET $n;
```

Query reference	SELECT19
Query description	Search users to invite (optional)
Query frequency	Dozens per week

```
SELECT "user".username
FROM "user"
WHERE "user".username LIKE '%$search%';
```

Query reference	SELECT20
Query description	List all requests to participate in a specific project
Query frequency	Dozens per week

```
SELECT "user".username FROM invite, "user"
WHERE project_id = $project_id AND invite.user_who_invited_id IS NULL
AND invite.user_invited_id = "user".id
LIMIT 10 OFFSET $n;
```

Query reference	SELECT21
Query description	list members of one project
Query frequency	Dozens per week

```
SELECT "user".username FROM invite, "user"
WHERE project_id = $project_id AND invite.user_who_invited_id IS NULL
AND invite.user_invited_id = "user".id
LIMIT 10 OFFSET $n;
```

Query reference	SELECT22

Query description	list all tasks of one specific project (info and state)
Query frequency	Hundreds per day

```
SELECT task.name, task_state_record.state FROM task, task_state_record
WHERE task.project_id = $project_id AND task_state_record.task_id = task.id
AND task_state_record.state =
(SELECT "state" FROM task_state_record WHERE task_id = task.id GROUP BY "state", date
ORDER BY date DESC LIMIT 1)
GROUP BY task.name, task_state_record.state;
```

Query reference	SELECT23
Query description	Get users assigned to a specific task
Query frequency	Dozens per week

```
SELECT "user".username, "user".image FROM task, "user", task_state_record
WHERE task.id = $task_id AND task.id = task_state_record.task_id AND
task_state_record.state = 'Assigned'
AND task_state_record.user_completed_id = "user".id;
```

Query reference	SELECT24
Query description	List all comments of a task
Query frequency	Dozens per day

```
SELECT "comment".content, "comment".date, "user".username, "user".image
FROM "comment", task, "user"
WHERE task.id = $task_id AND "comment".task_id = task.id AND "comment".user_id =
"user".id;
```

Query reference	SELECT25
Query description	List all sprints of one specific project, and their state
Query frequency	Hundreds per day

```
SELECT sprint.id, sprint.name, sprint.deadline, sprint_state_record.state FROM sprint,
sprint_state_record
WHERE sprint.project_id = $project_id AND sprint_state_record.sprint_id = sprint.id
AND sprint_state_record.state =
(SELECT "state" FROM sprint_state_record WHERE sprint_id = sprint.id
GROUP BY "state", date ORDER BY date DESC LIMIT 1)
```

```
GROUP BY sprint.id, sprint.name, sprint.deadline, sprint_state_record.state
ORDER BY deadline ASC;
```

Query reference	SELECT26
Query description	List all tasks of a sprint
Query frequency	Hundreds per day

```
SELECT task.name, task_state_record.state FROM task, task_state_record
WHERE task.sprint_id = $sprint_id AND task_state_record.task_id = task.id
AND task_state_record.state = (SELECT "state" FROM task_state_record WHERE task_id =
task.id
GROUP BY "state", date ORDER BY date DESC LIMIT 1)
GROUP BY task.name, task_state_record.state;
```

Query reference	SELECT27
Query description	Get information about one specific task
Query frequency	Hundreds per day

```
SELECT name, description, effort
FROM task
WHERE task.id = $task_id;
```

Query reference	SELECT28
Query description	Search project by category
Query frequency	Dozens per day

```
SELECT project.name, project.description, category.name FROM project, category,
project_categories
WHERE category.name LIKE '%$search%' AND category.id = project_categories.category_id
AND project_categories.project_id = project.id AND isPublic = TRUE
ORDER BY project.name
LIMIT 10 OFFSET $n;
```

Query reference	SELECT29
Query description	Search project by name or description
Query frequency	Dozens per day

```
SELECT "name", description FROM project
WHERE "name" LIKE '%$search%' OR description LIKE '%$search%'
AND isPublic = TRUE
ORDER BY name
LIMIT 10 OFFSET $n;
```

Query reference	SELECT30
Query description	Number of tasks completed in a project (statistics)
Query frequency	Dozens per day

```
SELECT COUNT(*)
FROM task, task_state_record
WHERE task.project_id = $project_id
AND task_state_record.task_id = task.id
AND task_state_record.state = 'Completed';
```

Query reference	SELECT31
Query description	Number of sprints completed in a project (statistics)
Query frequency	Dozens per day

```
SELECT COUNT(*)
FROM sprint, sprint_state_record
WHERE sprint.project_id = $project_id
```

```
AND sprint_state_record.sprint_id = sprint.id
AND sprint_state_record.state = 'Completed';
```

Query reference	SELECT32
Query description	Top 3 contributors in a project (statistics)
Query frequency	Dozens per day

```
SELECT "user".username, "user".image, COUNT(*) AS num
FROM "user", task_state_record, task
WHERE task.project_id = $project_id
AND task_state_record.task_id = task.id
AND "user".id = task_state_record.user_completed_id
AND task_state_record.state = 'Completed'
GROUP BY "user".username, "user".image
ORDER BY num DESC LIMIT 3;
```

Query reference	SELECT33
Query description	Number of tasks completed this month in a project, by each day (statistics)
Query frequency	Dozens per day

```
SELECT COUNT(*), date_part('day',date) AS day
FROM task_state_record, task
WHERE task.project_id = $project_id AND task_state_record.task_id = task.id
AND task_state_record.state = 'Completed' AND date_part('month',date) =
date_part('month',now())
GROUP BY day;
```

Query reference	SELECT34
Query description	Number of sprints completed this year in a project, by each month (statistics)
Query frequency	Dozens per day

```
SELECT COUNT(*), date_part('month',date) AS month
FROM sprint_state_record, sprint
WHERE sprint.project_id = $project_id AND sprint_state_record.sprint_id = sprint.id
AND sprint_state_record.state = 'Completed' AND date_part('year',date) =
date_part('year',now())
```

GROUP BY month;

1.3 Most frequent modifications

Query reference	INSERT01
Query description	Create new User
Query frequency	Dozens per day
INSERT INTO User (name, username, email, image, password) VALUES ($name, $username, $email, $image, $password);	

Query reference	INSERT02
Query description	Create new Sprint
Query frequency	
INSERT INTO Sprint (name, deadline, effort, project_id, user_creator_id) VALUES ($name, $deadline, $effort, $project_id, $user_creator_id);	

Query reference	INSERT03
Query description	Create new Task of Project
Query frequency	Hundreds per day
INSERT INTO Task (name, description, effort, project_id) VALUES ($name, $description, $effort, $project_id);	

Query reference	INSERT04
Query description	Create new Task of Sprint
Query frequency	Hundreds per day
INSERT INTO Task (name, description, effort, project_id, sprint_id) VALUES ($name, $description, $effort, $project_id, $sprint_id);	

Query reference	INSERT05
Query description	Create new TaskStateRecord
Query frequency	Hundreds per day
INSERT INTO TaskStateRecord (state, user_completed_id, task_id)	

```
VALUES ($state, $user_completed_id, $task_id);
```

Query reference	INSERT06
Query description	Create new SprintStateRecord
Query frequency	Dozens per day

```
INSERT INTO SprintStateRecord (state, user_completed_id, sprint_id)
VALUES ($state, $user_completed_id, $sprint_id);
```

Query reference	INSERT07
Query description	Create new Thread
Query frequency	Dozens per day

```
INSERT INTO Thread (name, description, project_id, user_creator_id)
VALUES ($name, $description, $project_id, $user_creator_id);
```

Query reference	INSERT08
Query description	Create new Project
Query frequency	Dozens per day

```
INSERT INTO Project (name, description, isPublic)
VALUES ($name, $description, $isPublic);
```

Query reference	INSERT09
Query description	Create new Comment in a Task
Query frequency	Dozens per day

```
INSERT INTO Comment (content, user_id, task_id)
VALUES ($content, $user_id, $task_id);
```

Query reference	INSERT10
Query description	Create new Comment in a Thread
Query frequency	Dozens per day

```
INSERT INTO Comment (content, user_id, thread_id)
VALUES ($content, $user_id, $thread_id);
```

Query reference	INSERT11
Query description	Create new Category
Query frequency	Units per month
INSERT INTO Category (name) VALUES ($name);	

Query reference	UPDATE01
Query description	Update Project info
Query frequency	Units per month
UPDATE Project SET name = $name, description = $description, isPublic = $isPublic WHERE id = $project_id;	

Query reference	UPDATE02
Query description	Update User info
Query frequency	Dozens per month
UPDATE User SET name = $name, username = $username, email = $email, password = $password, image = $image WHERE id = $user_id;	

Query reference	UPDATE03
Query description	Update Task info
Query frequency	Units per day
UPDATE Task SET name = $name, description = $description, effort = $effort WHERE id = $task_id;	

Query reference	UPDATE04
Query description	Update Sprint info
Query frequency	Units per week
UPDATE Sprint SET name = $name, deadline = $deadline, effort = $effort WHERE id = $sprint_id;	

Query reference	UPDATE05
Query description	Update Comment info
Query frequency	Dozens per week
UPDATE Comment SET content = $content WHERE id = $comment_id;	

Query reference	DELETE01
Query description	Remove User from Project
Query frequency	Units per month
DELETE FROM Project_members WHERE user_id = (SELECT id FROM User WHERE username = $username);	

Query reference	DELETE02
Query description	Remove Comment from Thread or Task
Query frequency	Units per month
DELETE FROM Comment WHERE comment_id = $comment_id;	

2. Proposed Indexes

2.1 Performance indexes

Index Reference	IDX01
Related queries	SELECT01, SELECT08, SELECT09, SELECT10
Index relation	“user”
Index attribute	username
Index type	Hash
Cardinality	High
Clustering	No
Justification	Queries above have to be executed fast because they are executed many times; not a good candidate for clustering

```
CREATE INDEX username_user
ON public."user" USING hash
(username COLLATE pg_catalog."default")
TABLESPACE pg_default;
```

Index Reference	IDX02
Related queries	SELECT22, SELECT30, SELECT32, SELECT33
Index relation	project
Index attribute	id
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	There is a big amount of data in this table, as well as a considerable number of queries accessing to it (SELECT22 being the most common), therefore needs to be fast; cardinality is medium, therefore it can be clustered

```
CREATE INDEX task_project
ON public.task USING btree
(project_id)
TABLESPACE pg_default;
```

Index Reference	IDX03
Related queries	SELECT26
Index relation	task
Index attribute	sprint_id
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	There is a big amount of data in this table, and the query can have a big frequency, therefore needs to be fast; cardinality is medium, therefore it can be clustered

```
CREATE INDEX task_sprint
ON public.task USING btree
(sprint_id)
TABLESPACE pg_default;
```

Index Reference	IDX04
Related queries	SELECT06, SELECT24
Index relation	comment
Index attribute	user_id
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	Tables with big amount of data; queries with high values of frequency, therefore needs to be fast; cardinality is medium, therefore it can be clustered

```
CREATE INDEX comment_creator
ON public.comment USING btree
(user_id)
TABLESPACE pg_default;
```

Index Reference	IDX05
Related queries	SELECT06
Index relation	comment
Index attribute	task_id
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	Tables with big amount of data; queries with medium/lower values of frequency, needs to be relatively fast; cardinality is medium, therefore it can be clustered

```
CREATE INDEX comment_thread
ON public.comment USING btree
(task_id)
TABLESPACE pg_default;
```

Index Reference	IDX06
Related queries	SELECT24
Index relation	comment
Index attribute	task_id
Index type	B-tree

Cardinality	Medium
Clustering	Yes
Justification	Tables with big amount of data; queries with medium values of frequency, needs to be relatively fast; cardinality is medium, therefore it can be clustered

```
CREATE INDEX comment_task
ON public.comment USING btree
(task_id)
TABLESPACE pg_default;
```

Index Reference	IDX07
Related queries	SELECT25
Index relation	sprint
Index attribute	deadline
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	There is a big amount of data in this table and the queries accessing to it require sorting, therefore needs to be fast; cardinality is medium, therefore it can be clustered

```
CREATE INDEX sprint_deadline
ON public.sprint USING btree
(deadline)
TABLESPACE pg_default;
```

Index Reference	IDX08
Related queries	SELECT08, SELECT14, SELECT15, SELECT16, SELECT25, SELECT31, SELECT34
Index relation	sprint
Index attribute	project_id
Index type	btree
Cardinality	medium
Clustering	Yes
Justification	There are a lot of data in this table, as well as a considerable number of queries accessing to it , therefore needs to be fast; cardinality is

	medium, therefore it can be clustered
CREATE INDEX sprint_project ON public.sprint USING btree (project_id) TABLESPACE pg_default;	

Index Reference	IDX09
Related queries	SELECT07
Index relation	notification
Index attribute	user_id
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	Table with considerable amounts of data; queries have higher frequency, therefore needs to be fast; clustering is advisable
CREATE INDEX notification_user ON public.notification USING btree (user_id) TABLESPACE pg_default;	

Index Reference	IDX10
Related queries	SELECT25, SELECT31, SELECT34
Index relation	sprint_state_record
Index attribute	state
Index type	B-tree
Cardinality	Low
Clustering	Yes
Justification	Table with huge amounts of data, with high frequency queries accessing to it; Needs to be fast; Cardinality is low, so clustering is advisable
CREATE INDEX sprint_record_state ON public.sprint_state_record USING btree (state COLLATE pg_catalog."default") TABLESPACE pg_default;	

Index Reference	IDX11
Related queries	SELECT34
Index relation	sprint_state_record
Index attribute	date
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	Table with huge amounts of data, with medium frequency queries accessing to it; Needs to be relatively fast; Cardinality is medium, so clustering is advisable

```
CREATE INDEX sprint_state_record_date
ON public.sprint_state_record USING btree
(date)
TABLESPACE pg_default;
```

Index Reference	IDX12
Related queries	SELECT25, SELECT31, SELECT34
Index relation	sprint_state_record
Index attribute	sprint_id
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	Table with huge amounts of data, with high frequency queries accessing to it (specially SELECT25); Needs to be fast; Clustering is advisable because cardinality is medium

```
CREATE INDEX sprint_state_sprint
ON public.sprint_state_record USING btree
(sprint_id)
TABLESPACE pg_default;
```

Index Reference	IDX13
Related queries	SELECT11, SELECT12, SELECT13, SELECT22, SELECT23, SELECT26, SELECT30, SELECT32, SELECT33
Index relation	task_state_record
Index attribute	state

Index type	B-tree
Cardinality	Low
Clustering	Yes
Justification	Table with huge amounts of data, with high frequency queries accessing to it; Needs to be fast; Cardinality is low, so clustering is advisable

```
CREATE INDEX task_record_state
ON public.task_state_record USING btree
(state COLLATE pg_catalog."default")
TABLESPACE pg_default;
```

Index Reference	IDX14
Related queries	SELECT11, SELECT12, SELECT33
Index relation	task_state_record
Index attribute	date
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	Table with huge amounts of data, with medium frequency queries accessing to it; Needs to be relatively fast; Cardinality is medium, so clustering is advisable

```
CREATE INDEX task_state_record_date
ON public.task_state_record USING btree
(date)
TABLESPACE pg_default;
```

Index Reference	IDX15
Related queries	SELECT13, SELECT22, SELECT23, SELECT26, SELECT30, SELECT32, SELECT33
Index relation	task_state_record
Index attribute	task_id
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	Table with huge amounts of data, with high frequency queries

	accessing to it; Needs to be fast; Clustering is advisable
CREATE INDEX task_state_task ON public.task_state_record USING btree (task_id) TABLESPACE pg_default;	

Index Reference	IDX16
Related queries	SELECT11, SELECT12, SELECT13, SELECT23, SELECT32
Index relation	task_state_record
Index attribute	user_completed_id
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	Table with huge amounts of data, with high frequency queries accessing to it (specially query SELECT23); Needs to be fast; Clustering is advisable
CREATE INDEX task_state_user ON public.task_state_record USING btree (user_completed_id) TABLESPACE pg_default;	

Index Reference	IDX17
Related queries	SELECT04
Index relation	thread
Index attribute	project_id
Index type	B-tree
Cardinality	Medium
Clustering	Yes
Justification	Table with considerable amount of data; queries with medium/low frequencies; needs to be relatively fast; clustering is possible because cardinality is medium
CREATE INDEX thread_project ON public.thread USING btree (project_id) TABLESPACE pg_default;	

2.2 Full-text Search Indices

Index Reference	IDX18
Related queries	SELECT15, SELECT29
Index relation	project
Index attribute	description
Index type	GiST
Clustering	No
Justification	Improves performance for full text searches; Use of GiST because it is dynamic data

```
CREATE INDEX project_text_search_description
ON public.project USING gist
(to_tsvector('english'::regconfig, description))
TABLESPACE pg_default;
```

Index Reference	IDX19
Index relation	SELECT15, SELECT29
Index attribute	project
Index type	name
Clustering	No
Justification	Improves performance for full text searches; Use of GiST because it is dynamic data

```
CREATE INDEX project_text_search_name
ON public.project USING gist
(to_tsvector('english'::regconfig, name))
TABLESPACE pg_default;
```

3. Triggers

Trigger Reference	TRIGGER01
Trigger Description	After creating a comment, creates a notification

```
DROP FUNCTION IF EXISTS add_notification_comment();
CREATE FUNCTION add_notification_comment() RETURNS TRIGGER AS
$BODY$
DECLARE
```

```
user_thread_id "thread".user_creator_id%TYPE;
BEGIN
IF(NEW.thread_id != NULL) THEN
SELECT thread.user_creator_id INTO user_thread_id
FROM thread WHERE thread.id = NEW.thread_id;
INSERT INTO notification (date,user_id,project_id,comment_id,user_action_id)
VALUES (NEW.date,user_thread_id,NULL,NEW.id,NULL);
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS add_notification_comment ON comment;
CREATE TRIGGER add_notification_comment
AFTER INSERT ON comment
FOR EACH ROW
EXECUTE PROCEDURE add_notification_comment();
```

Trigger Reference	TRIGGER02
Trigger Description	After creating an invite, creates a notification

```
DROP FUNCTION IF EXISTS add_notification_invite();
CREATE FUNCTION add_notification_invite() RETURNS TRIGGER AS
$BODY$
BEGIN
IF(NEW.user_who_invited_id != NULL)
THEN INSERT INTO notification (date,notification_type,user_id,project_id,user_action_id)
VALUES (now(),'invite',NEW.user_invited_id,NEW.project_id,NEW.user_who_invited_id);
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS add_notification_invite ON invite;
CREATE TRIGGER add_notification_invite
AFTER INSERT ON invite
FOR EACH ROW
EXECUTE PROCEDURE add_notification_invite();
```

Trigger Reference	TRIGGER03
Trigger Description	After a user is promoted, creates a notification

```
DROP FUNCTION IF EXISTS add_notification_promotion();
CREATE FUNCTION add_notification_promotion() RETURNS TRIGGER AS
$BODY$
BEGIN
IF(OLD.role != NEW.role) THEN
INSERT INTO Notification (date,user_id,project_id,comment_id,user_action_id)
VALUES (now(),NEW.user_id,NEW.project_id,NULL,NULL);
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS add_notification_promotion ON project_members;
CREATE TRIGGER add_notification_promotion
AFTER UPDATE ON project_members
FOR EACH ROW
EXECUTE PROCEDURE add_notification_promotion();
```

Trigger Reference	TRIGGER04
Trigger Description	After a user is removed from a project, creates a notification

```
DROP FUNCTION IF EXISTS add_notification_remove();
CREATE FUNCTION add_notification_remove() RETURNS TRIGGER AS
$BODY$
BEGIN
INSERT INTO Notification (date,user_id,project_id,comment_id,user_action_id)
VALUES (now(),OLD.user_id,OLD.project_id,NULL,NULL);
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS add_notification_remove ON project_members;
CREATE TRIGGER add_notification_remove
AFTER DELETE ON project_members
FOR EACH ROW
```

```
EXECUTE PROCEDURE add_notification_remove();
```

Trigger Reference	TRIGGER05
Trigger Description	If a user is already a member of a project, throws an exception

```
DROP FUNCTION IF EXISTS check_user_member();
CREATE FUNCTION check_user_member() RETURNS TRIGGER AS
$BODY$
BEGIN
IF (EXISTS(SELECT * FROM project_members WHERE project_id =
NEW.project_id AND user_id = NEW.user_id))
THEN RAISE EXCEPTION 'This member is already in the project.';
END IF;
RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS check_user_member ON project_members;
CREATE TRIGGER check_user_member
BEFORE INSERT ON project_members
FOR EACH ROW
EXECUTE PROCEDURE check_user_member();
```

Trigger Reference	TRIGGER06
Trigger Description	When a Comment is reported, creates a notification

```
DROP FUNCTION IF EXISTS add_notification_report();
CREATE FUNCTION add_notification_report() RETURNS TRIGGER AS
$BODY$
BEGIN
IF(NEW.type = 'commentreported')
THEN INSERT INTO "notification" (date,notification_type,user_id,comment_id)
VALUES (now(),'commentreported',NEW.user_id,NEW.comment_reported_id);
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;
```

```
DROP TRIGGER IF EXISTS add_notification_report ON report;
CREATE TRIGGER add_notification_report
AFTER INSERT ON report
FOR EACH ROW
EXECUTE PROCEDURE add_notification_report();
```

Trigger Reference	TRIGGER07
Trigger Description	When a sprint is created,inserts a tuple to the SprintStateRecord with the information that it was created

```
DROP FUNCTION IF EXISTS sprint_created();
CREATE FUNCTION sprint_created() RETURNS TRIGGER AS
$BODY$
BEGIN
INSERT INTO sprint_state_record (date,state,sprint_id)
VALUES (now(), 'Created', NEW.id);
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS sprint_created ON task;
CREATE TRIGGER sprint_created
AFTER INSERT ON sprint
FOR EACH ROW
EXECUTE PROCEDURE sprint_created();
```

Trigger Reference	TRIGGER08
Trigger Description	Change state of sprint if a sprint's deadline is updated as the sprint was outdated

```
DROP FUNCTION IF EXISTS update_sprint_state_deadline();
CREATE FUNCTION update_sprint_state_deadline() RETURNS TRIGGER AS
$BODY$
BEGIN
IF(NEW.deadline <> OLD.deadline) THEN
IF((SELECT state FROM sprint_state_record WHERE NEW.id = sprint_state_record.sprint_id
ORDER BY date LIMIT 1) = 'Outdated')
THEN
INSERT INTO sprint_state_record(date,state,sprint_id)
```

```
VALUES (now(), 'Created', NEW.id);
END IF;
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS update_sprint_state_deadline ON task;
CREATE TRIGGER update_sprint_state_deadline
AFTER UPDATE ON sprint
FOR EACH ROW
EXECUTE PROCEDURE update_sprint_state_deadline();
```

Trigger Reference	TRIGGER09
Trigger Description	Check if a sprint is completed and revoke that if a task is added to it

```
DROP FUNCTION IF EXISTS change_sprint_state();
CREATE FUNCTION change_sprint_state() RETURNS TRIGGER AS
$BODY$
DECLARE
temprow RECORD;
BEGIN
IF(NEW.sprint_id <> NULL) THEN
FOR temprow IN
SELECT state FROM sprint_state_record WHERE NEW.sprint_id = sprint_state_record.sprint_id
ORDER BY date
LOOP
IF(temprow.state = 'Created' OR temprow.state = 'Outdated') THEN
RETURN NULL;
ELSE
IF(temprow.state = 'Completed') THEN
INSERT INTO sprint_state_record (date,state,sprint_id)
VALUES (now(), 'Created', NEW.sprint_id);
END IF;
END IF;
END LOOP;
END IF;
RETURN NULL;
END
```

```
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS change_sprint_state ON task;
CREATE TRIGGER change_sprint_state
AFTER INSERT ON task
FOR EACH ROW
EXECUTE PROCEDURE change_sprint_state();
```

Trigger Reference	TRIGGER10
Trigger Description	Checks if the sum of task's efforts exceeds the sprint's effort

```
DROP FUNCTION IF EXISTS check_effort();
CREATE FUNCTION check_effort() RETURNS TRIGGER AS
$BODY$
BEGIN
IF ((SELECT SUM(effort) FROM task WHERE NEW.sprint_id = task.sprint_id) >
(SELECT effort FROM sprint WHERE id = NEW.sprint_id))
THEN RAISE EXCEPTION 'This task exceeds the limit effort of the sprint.';
END IF;
RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS check_effort ON task;
CREATE TRIGGER check_effort
BEFORE INSERT OR UPDATE ON task
FOR EACH ROW
EXECUTE PROCEDURE check_effort();
```

Trigger Reference	TRIGGER11
Trigger Description	When a task is created,inserts a tuple to the TaskStateRecord with the information that it was created

```
DROP FUNCTION IF EXISTS task_created();
CREATE FUNCTION task_created() RETURNS TRIGGER AS
$BODY$
BEGIN
INSERT INTO task_state_record (date,state,user_completed_id,task_id)
```

```
VALUES (now(), 'Created', NULL, NEW.id);
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS task_created ON task;
CREATE TRIGGER task_created
AFTER INSERT ON task
FOR EACH ROW
EXECUTE PROCEDURE task_created();
```

Trigger Reference	TRIGGER12
Trigger Description	Completes a sprint when its tasks are completed (adds a tuple to sprint_state_record with state completed)

```
DROP FUNCTION IF EXISTS check_completed_sprint();
CREATE FUNCTION check_completed_sprint() RETURNS TRIGGER AS
$BODY$
DECLARE
sprint_id "task".sprint_id%TYPE;
BEGIN
SELECT task.sprint_id INTO sprint_id FROM task WHERE task.id = NEW.task_id;

IF (NEW.state = 'Completed') THEN
IF(
(SELECT COUNT(*) FROM task WHERE task.sprint_id =
(SELECT task.sprint_id FROM task WHERE task.id = NEW.task_id)) -- tasks of respective sprint
=
(SELECT COUNT(*) FROM task_state_record t, task a
WHERE t.state = 'Completed' AND a.sprint_id =
(SELECT task.sprint_id FROM task WHERE task.id = NEW.task_id))
)
THEN
INSERT INTO sprint_state_record (id, date, state, sprint_id) VALUES (DEFAULT, now(), 'Completed',
sprint_id);
END IF;
END IF;
RETURN NULL;
END;
$BODY$
```

```
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS check_completed_sprint ON task_state_record;
CREATE TRIGGER check_completed_sprint
AFTER INSERT ON task_state_record
FOR EACH ROW
EXECUTE PROCEDURE check_completed_sprint();
```

Trigger Reference	TRIGGER13
Trigger Description	Change sprint Status to outdated when the deadline is passed (adds a tuple to sprint_state_record with the state as Outdated) - Triggered using cron

```
DROP FUNCTION IF EXISTS check_deadline();
CREATE FUNCTION check_deadline() RETURNS TRIGGER AS
$BODY$
DECLARE
  deadline "sprint"%rowtype;
BEGIN
FOR deadline IN
SELECT deadline, id FROM sprint
LOOP
   IF (deadline < now()) THEN
   INSERT INTO sprint_state_record (date, state, sprint_id) VALUES (now(), 'Outdated', id);
         END IF;
END LOOP;
END;
$BODY$
LANGUAGE plpgsql;
```

4. SQL Code

```
CREATE TABLE "user" (
id SERIAL NOT NULL,
name text NOT NULL,
username text NOT NULL,
email text NOT NULL,
image text,
password text NOT NULL
);

CREATE TABLE project (
	id SERIAL NOT NULL,
	name text NOT NULL,
	description text NOT NULL,
	isPublic boolean NOT NULL
);

CREATE TABLE sprint (
	id SERIAL NOT NULL,
	name text NOT NULL,
	deadline TIMESTAMP WITH TIME zone NOT NULL,
	effort INTEGER NOT NULL,
	project_id INTEGER NOT NULL,
	user_creator_id INTEGER NOT NULL,
	CONSTRAINT deadline CHECK (deadline > now())
);

CREATE TABLE task (
	id SERIAL NOT NULL,
	name text NOT NULL,
	description text,
	effort INTEGER NOT NULL,
	project_id INTEGER NOT NULL,
	sprint_id INTEGER
);

CREATE TABLE thread (
	id SERIAL NOT NULL,
	name text NOT NULL,
	description text,
	date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
	project_id INTEGER NOT NULL,
	user_creator_id INTEGER NOT NULL
);

CREATE TABLE comment (
	id SERIAL NOT NULL,
	content text NOT NULL,
	date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
	user_id INTEGER NOT NULL,
	task_id INTEGER,
	thread_id INTEGER,
	CONSTRAINT belongs CHECK ((task_id != NULL AND thread_id = NULL) OR (task_id =
NULL AND thread_id != NULL))
);

CREATE TABLE category (
	id SERIAL NOT NULL,
	name text NOT NULL
```

```
);

CREATE TABLE project_members (
        user_id INTEGER NOT NULL,
        project_id INTEGER NOT NULL,
        isCoordinator boolean NOT NULL,
        date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL
);

CREATE TABLE administrator (
        id SERIAL NOT NULL,
        username text NOT NULL,
        password text NOT NULL
);

CREATE TABLE report (
        id SERIAL NOT NULL,
        date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
        type text NOT NULL,
        summary text NOT NULL,
        user_id INTEGER NOT NULL,
        comment_reported_id INTEGER,
        user_reported_id INTEGER,
        CONSTRAINT reportType CHECK ((type = ANY(ARRAY['commentReported'::text,
                                                'userReported'::text]))),
        CONSTRAINT typeConstraint CHECK ((type = 'commentReported' AND
        comment_reported_id != NULL AND user_reported_id = NULL) OR
        (type = 'userReported' AND user_reported_id != NULL AND comment_reported_id =
        NULL))
);

CREATE TABLE notification (
        id SERIAL NOT NULL,
        date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
        notification_type text NOT NULL,
        user_id INTEGER NOT NULL,
        project_id INTEGER,
        comment_id INTEGER,
        user_action_id INTEGER,
        CONSTRAINT notificationType CHECK ((notification_type = ANY(ARRAY['comment'::text,
        'commentReported'::text, 'Promotion'::text, 'RemovedFromproject'::text, 'invite'::text,
                'Request'::text]))),
        CONSTRAINT notificationConstraint CHECK ((notification_type = 'comment' AND
        comment_id != NULL) OR
        (notification_type = 'commentReported' AND comment_id != NULL) OR
        (notification_type = 'Promotion' AND project_id != NULL) OR
        (notification_type = 'RemovedFromProject' AND project_id != NULL) OR
        (notification_type = 'Invite' AND project_id != NULL AND user_action_id != NULL) OR
        (notification_type = 'Request' AND project_id != NULL))
);

CREATE TABLE invite (
        id SERIAL NOT NULL,
        date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
        user_invited_id INTEGER NOT NULL,
        project_id INTEGER NOT NULL,
        user_who_invited_id INTEGER
);
```

```
CREATE TABLE task_state_record(
        id SERIAL NOT NULL,
        date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
        state text NOT NULL,
        user_completed_id INTEGER,
        task_id INTEGER NOT NULL,
        CONSTRAINT state CHECK ((state = ANY(ARRAY['Completed'::text, 'Assigned'::text,
        'Unnassigned'::text, 'Created'::text])))
);

CREATE TABLE sprint_state_record(
        id SERIAL NOT NULL,
        date TIMESTAMP WITH TIME zone DEFAULT now() NOT NULL,
        state text NOT NULL,
        sprint_id INTEGER NOT NULL,
        CONSTRAINT state CHECK ((state = ANY(ARRAY['Completed'::text, 'Outdated'::text,
                                    'Created'::text]))))
);

CREATE TABLE project_categories (
        project_id INTEGER NOT NULL,
        category_id INTEGER NOT NULL
);

/* Primary Keys and Uniques*/

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

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

ALTER TABLE ONLY "user"
  ADD CONSTRAINT user_username_key UNIQUE (username);

ALTER TABLE ONLY project
ADD CONSTRAINT project_pkey PRIMARY KEY (id);

ALTER TABLE ONLY sprint
ADD CONSTRAINT sprint_pkey PRIMARY KEY (id);

ALTER TABLE ONLY task
ADD CONSTRAINT task_pkey PRIMARY KEY (id);

ALTER TABLE ONLY thread
ADD CONSTRAINT thread_pkey PRIMARY KEY (id);

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

ALTER TABLE ONLY category
ADD CONSTRAINT category_pkey PRIMARY KEY (id);

ALTER TABLE ONLY project_members
ADD CONSTRAINT project_members_pkey PRIMARY KEY (user_id, project_id);

ALTER TABLE ONLY administrator
ADD CONSTRAINT administrator_pkey PRIMARY KEY (id);

ALTER TABLE ONLY report
ADD CONSTRAINT report_pkey PRIMARY KEY (id);
```

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

ALTER TABLE ONLY invite
ADD CONSTRAINT invite_pkey PRIMARY KEY (id);

ALTER TABLE ONLY task_state_record
ADD CONSTRAINT task_state_record_pkey PRIMARY KEY (id);

ALTER TABLE ONLY sprint_state_record
ADD CONSTRAINT sprint_state_record_pkey PRIMARY KEY (id);

ALTER TABLE ONLY project_categories
ADD CONSTRAINT project_categories_pkey PRIMARY KEY (project_id, category_id);

/* Foreign Keys */

ALTER TABLE ONLY sprint
ADD CONSTRAINT task_id_project_fkey FOREIGN KEY (project_id) REFERENCES project(id) ON
UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY sprint
ADD CONSTRAINT sprint_id_user_creator_fkey FOREIGN KEY (user_creator_id) REFERENCES
"user"(id) ON UPDATE CASCADE;

ALTER TABLE ONLY task
ADD CONSTRAINT task_id_user_project_fkey FOREIGN KEY (project_id) REFERENCES project(id)
ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY task
ADD CONSTRAINT task_id_user_sprint_fkey FOREIGN KEY (sprint_id) REFERENCES sprint(id) ON
UPDATE CASCADE;

ALTER TABLE ONLY thread
ADD CONSTRAINT thread_id_project_fkey FOREIGN KEY (project_id) REFERENCES project(id) ON
UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY thread
ADD CONSTRAINT thread_id_user_creator_fkey FOREIGN KEY (user_creator_id) REFERENCES
"user"(id) ON UPDATE CASCADE;

ALTER TABLE ONLY comment
ADD CONSTRAINT comment_id_user_fkey FOREIGN KEY (user_id) REFERENCES "user"(id) ON
UPDATE CASCADE;

ALTER TABLE ONLY comment
ADD CONSTRAINT comment_id_task_fkey FOREIGN KEY (task_id) REFERENCES task(id) ON
UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY comment
ADD CONSTRAINT comment_id_thread_fkey FOREIGN KEY (thread_id) REFERENCES thread(id)
ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY project_members
ADD CONSTRAINT members_id_user_fkey FOREIGN KEY (user_id) REFERENCES "user"(id) ON
UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY project_members
ADD CONSTRAINT members_id_project_fkey FOREIGN KEY (project_id) REFERENCES project(id)
ON UPDATE CASCADE ON DELETE CASCADE;
```

```
ALTER TABLE ONLY report
ADD CONSTRAINT report_id_user_fkey FOREIGN KEY (user_id) REFERENCES "user"(id) ON
UPDATE CASCADE;

ALTER TABLE ONLY report
ADD CONSTRAINT report_id_comment_reported_fkey FOREIGN KEY (comment_reported_id)
REFERENCES comment(id) ON UPDATE CASCADE;

ALTER TABLE ONLY report
ADD CONSTRAINT report_id_user_reported_fkey FOREIGN KEY (user_reported_id) REFERENCES
"user"(id) ON UPDATE CASCADE;

ALTER TABLE ONLY notification
ADD CONSTRAINT notification_id_user_fkey FOREIGN KEY (user_id) REFERENCES "user"(id) ON
UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY notification
ADD CONSTRAINT notification_id_project_fkey FOREIGN KEY (project_id) REFERENCES
project(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY notification
ADD CONSTRAINT notification_id_comment_fkey FOREIGN KEY (comment_id) REFERENCES
comment(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY notification
ADD CONSTRAINT notification_id_user_action_fkey FOREIGN KEY (user_action_id) REFERENCES
"user"(id) ON UPDATE CASCADE;

ALTER TABLE ONLY invite
ADD CONSTRAINT invite_id_user_fkey FOREIGN KEY (user_invited_id) REFERENCES "user"(id)
ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY invite
ADD CONSTRAINT invite_id_user_who_invited_fkey FOREIGN KEY (user_who_invited_id)
REFERENCES "user"(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY invite
ADD CONSTRAINT invite_id_project_fkey FOREIGN KEY (project_id) REFERENCES project(id) ON
UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY task_state_record
ADD CONSTRAINT task_state_record_id_user_fkey FOREIGN KEY (user_completed_id)
REFERENCES "user"(id) ON UPDATE CASCADE;

ALTER TABLE ONLY task_state_record
ADD CONSTRAINT task_state_record_id_task_fkey FOREIGN KEY (task_id) REFERENCES task(id)
ON UPDATE CASCADE;

ALTER TABLE ONLY sprint_state_record
ADD CONSTRAINT sprint_state_record_id_sprint_fkey FOREIGN KEY (sprint_id) REFERENCES
sprint(id) ON UPDATE CASCADE;

ALTER TABLE ONLY project_categories
ADD CONSTRAINT project_categories_id_project_fkey FOREIGN KEY (project_id) REFERENCES
project(id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY project_categories
ADD CONSTRAINT project_categories_id_category_fkey FOREIGN KEY (category_id)
REFERENCES category(id) ON UPDATE CASCADE ON DELETE CASCADE;
```

```
/* TRIGGERS */
-- Checks if the effort of a sprint tasks exceeds the sprint effort
DROP FUNCTION IF EXISTS check_effort();
CREATE FUNCTION check_effort() RETURNS TRIGGER AS
$BODY$
BEGIN
IF ((SELECT SUM(effort) FROM task WHERE NEW.sprint_id = task.sprint_id) >
(SELECT effort FROM sprint WHERE id = NEW.sprint_id))
THEN RAISE EXCEPTION 'This task exceeds the limit effort of the sprint.';
END IF;
RETURN NEW;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS check_effort ON task;
CREATE TRIGGER check_effort
BEFORE INSERT OR UPDATE ON task
FOR EACH ROW
EXECUTE PROCEDURE check_effort();


-- Create a notification when a report is created
DROP FUNCTION IF EXISTS add_notification_report();
CREATE FUNCTION add_notification_report() RETURNS TRIGGER AS
$BODY$
BEGIN
IF(NEW.type = 'commentreported')
THEN INSERT INTO "notification" (date,notification_type,user_id,comment_id)
VALUES (now(),'commentreported',NEW.user_id,NEW.comment_reported_id);
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS add_notification_report ON report;
CREATE TRIGGER add_notification_report
AFTER INSERT ON report
FOR EACH ROW
EXECUTE PROCEDURE add_notification_report();

-- Create a notification when an invite is created (not a request)
DROP FUNCTION IF EXISTS add_notification_invite();
CREATE FUNCTION add_notification_invite() RETURNS TRIGGER AS
$BODY$
BEGIN
IF(NEW.user_who_invited_id != NULL)
THEN INSERT INTO notification (date,notification_type,user_id,project_id,user_action_id)
VALUES (now(),'invite',NEW.user_invited_id,NEW.project_id,NEW.user_who_invited_id);
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS add_notification_invite ON invite;
CREATE TRIGGER add_notification_invite
AFTER INSERT ON invite
FOR EACH ROW
EXECUTE PROCEDURE add_notification_invite();
```

```
-- Change sprint Status when the deadline is passed
DROP FUNCTION IF EXISTS check_deadline();
CREATE FUNCTION check_deadline() RETURNS TRIGGER AS
$BODY$
DECLARE
  deadline "sprint"%rowtype;
BEGIN
FOR deadline IN
SELECT deadline, id FROM sprint
LOOP
   IF (deadline < now()) THEN
   INSERT INTO sprint_state_record (date, state, sprint_id) VALUES (now(), 'Outdated', id);
         END IF;
END LOOP;
END;
$BODY$
LANGUAGE plpgsql;

-- Complete a sprint when its tasks are completed
DROP FUNCTION IF EXISTS check_completed_sprint();
CREATE FUNCTION check_completed_sprint() RETURNS TRIGGER AS
$BODY$
DECLARE
sprint_id "task".sprint_id%TYPE;
BEGIN
SELECT task.sprint_id INTO sprint_id FROM task WHERE task.id = NEW.task_id;

IF (NEW.state = 'Completed') THEN
IF(
(SELECT COUNT(*) FROM task WHERE task.sprint_id =
(SELECT task.sprint_id FROM task WHERE task.id = NEW.task_id)) -- tasks of respective sprint
=
(SELECT COUNT(*) FROM task_state_record t, task a
WHERE t.state = 'Completed' AND a.sprint_id =
(SELECT task.sprint_id FROM task WHERE task.id = NEW.task_id))
)
THEN
INSERT INTO sprint_state_record (id, date, state, sprint_id) VALUES (DEFAULT, now(),
'Completed', sprint_id);
END IF;
END IF;
RETURN NULL;
END;
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS check_completed_sprint ON task_state_record;
CREATE TRIGGER check_completed_sprint
AFTER INSERT ON task_state_record
FOR EACH ROW
EXECUTE PROCEDURE check_completed_sprint();

-- Check if a user invited isn't already on the project
DROP FUNCTION IF EXISTS check_user_member();
CREATE FUNCTION check_user_member() RETURNS TRIGGER AS
$BODY$
BEGIN
IF (EXISTS(SELECT * FROM project_members WHERE project_id =
NEW.project_id AND user_id = NEW.user_id))
THEN RAISE EXCEPTION 'This member is already in the project.';
END IF;
RETURN NEW;
```

```
END;
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS check_user_member ON project_members;
CREATE TRIGGER check_user_member
BEFORE INSERT ON project_members
FOR EACH ROW
EXECUTE PROCEDURE check_user_member();

-- Insert into Notification of the user who wrote the thread, when a comment is made on it
DROP FUNCTION IF EXISTS add_notification_comment();
CREATE FUNCTION add_notification_comment() RETURNS TRIGGER AS
$BODY$
DECLARE
user_thread_id "thread".user_creator_id%TYPE;
BEGIN
IF(NEW.thread_id != NULL) THEN
SELECT thread.user_creator_id INTO user_thread_id
FROM thread WHERE thread.id = NEW.thread_id;
INSERT INTO notification (date,user_id,project_id,comment_id,user_action_id)
VALUES (NEW.date,user_thread_id,NULL,NEW.id,NULL);
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS add_notification_comment ON comment;
CREATE TRIGGER add_notification_comment
AFTER INSERT ON comment
FOR EACH ROW
EXECUTE PROCEDURE add_notification_comment();

-- Insert into Notification if user as been updated to Coordenator
DROP FUNCTION IF EXISTS add_notification_promotion();
CREATE FUNCTION add_notification_promotion() RETURNS TRIGGER AS
$BODY$
BEGIN
IF(OLD.role != NEW.role) THEN
INSERT INTO Notification (date,user_id,project_id,comment_id,user_action_id)
VALUES (now(),NEW.user_id,NEW.project_id,NULL,NULL);
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS add_notification_promotion ON project_members;
CREATE TRIGGER add_notification_promotion
AFTER UPDATE ON project_members
FOR EACH ROW
EXECUTE PROCEDURE add_notification_promotion();

-- Insert into Notification if user as been expelled from a project
DROP FUNCTION IF EXISTS add_notification_remove();
CREATE FUNCTION add_notification_remove() RETURNS TRIGGER AS
$BODY$
BEGIN
INSERT INTO Notification (date,user_id,project_id,comment_id,user_action_id)
VALUES (now(),OLD.user_id,OLD.project_id,NULL,NULL);
RETURN NULL;
```

```
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS add_notification_remove ON project_members;
CREATE TRIGGER add_notification_remove
AFTER DELETE ON project_members
FOR EACH ROW
EXECUTE PROCEDURE add_notification_remove();


-- Check if a sprint is completed and revoke that if a task is added to it
DROP FUNCTION IF EXISTS change_sprint_state();
CREATE FUNCTION change_sprint_state() RETURNS TRIGGER AS
$BODY$
DECLARE
temprow RECORD;
BEGIN
IF(NEW.sprint_id <> NULL) THEN
FOR temprow IN
SELECT state FROM sprint_state_record WHERE NEW.sprint_id = sprint_state_record.sprint_id
ORDER BY date
LOOP
IF(temprow.state = 'Created' OR temprow.state = 'Outdated') THEN
RETURN NULL;
ELSE
IF(temprow.state = 'Completed') THEN
INSERT INTO sprint_state_record (date,state,sprint_id)
VALUES (now(), 'Created', NEW.sprint_id);
END IF;
END IF;
END LOOP;
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS change_sprint_state ON task;
CREATE TRIGGER change_sprint_state
AFTER INSERT ON task
FOR EACH ROW
EXECUTE PROCEDURE change_sprint_state();

-- Change state of sprint if a sprint's deadline is updated
DROP FUNCTION IF EXISTS update_sprint_state_deadline();
CREATE FUNCTION update_sprint_state_deadline() RETURNS TRIGGER AS
$BODY$
BEGIN
IF(NEW.deadline <> OLD.deadline) THEN
IF((SELECT state FROM sprint_state_record WHERE NEW.id = sprint_state_record.sprint_id
ORDER BY date LIMIT 1) = 'Outdated')
THEN
INSERT INTO sprint_state_record (date,state,sprint_id)
VALUES (now(), 'Created', NEW.id);
END IF;
END IF;
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;
```

```
DROP TRIGGER IF EXISTS update_sprint_state_deadline ON task;
CREATE TRIGGER update_sprint_state_deadline
AFTER UPDATE ON sprint
FOR EACH ROW
EXECUTE PROCEDURE update_sprint_state_deadline();

-- When a task is created, add an entry to the task_state_record
DROP FUNCTION IF EXISTS task_created();
CREATE FUNCTION task_created() RETURNS TRIGGER AS
$BODY$
BEGIN
INSERT INTO task_state_record (date,state,user_completed_id,task_id)
VALUES (now(), 'Created', NULL, NEW.id);
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS task_created ON task;
CREATE TRIGGER task_created
AFTER INSERT ON task
FOR EACH ROW
EXECUTE PROCEDURE task_created();

-- When a sprint is created, add an entry to the sprint_state_record
DROP FUNCTION IF EXISTS sprint_created();
CREATE FUNCTION sprint_created() RETURNS TRIGGER AS
$BODY$
BEGIN
INSERT INTO sprint_state_record (date,state,sprint_id)
VALUES (now(), 'Created', NEW.id);
RETURN NULL;
END
$BODY$
LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS sprint_created ON task;
CREATE TRIGGER sprint_created
AFTER INSERT ON sprint
FOR EACH ROW
EXECUTE PROCEDURE sprint_created();

/* INDEXES */

CREATE INDEX username_user ON "user" USING hash(username);

CREATE INDEX task_project ON task USING btree(project_id);
CREATE INDEX task_sprint ON task USING btree(sprint_id);

CREATE INDEX sprint_project ON sprint USING btree(project_id);

CREATE INDEX comment_creator ON comment USING btree(user_id);
CREATE INDEX comment_task ON comment USING btree(task_id);
CREATE INDEX comment_thread ON comment USING btree(thread_id);

CREATE INDEX thread_project ON thread USING btree(project_id);

CREATE INDEX notification_user ON notification USING btree(user_id);

CREATE INDEX task_state_task ON task_state_record USING btree(task_id);
CREATE INDEX task_state_user ON task_state_record USING btree(user_completed_id);
CREATE INDEX task_record_state ON task_state_record USING btree(state);
```

```
CREATE INDEX sprint_state_sprint ON sprint_state_record USING btree(sprint_id);
CREATE INDEX sprint_record_state ON sprint_state_record USING btree(state);

CREATE INDEX task_state_record_date ON task_state_record USING btree(date);
CREATE INDEX sprint_state_record_date ON sprint_state_record USING btree(date);
CREATE INDEX sprint_deadline ON sprint USING btree(deadline);

CREATE INDEX project_text_search_name ON project USING GIST(to_tsvector('english',name));
CREATE INDEX project_text_search_description ON project USING
GIST(to_tsvector('english',description));

/* INSERTS */

INSERT INTO administrator (id, username, password) VALUES (1, 'admin', '1234Admin-');

INSERT INTO "user" (id,name,username,email,image,password) VALUES (1,'Pedro
Reis','partelinuxes','pedroreis@gmail.com',NULL,'eunaverdadegostodewindows');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (2,'Ana
Margarida','PortugueseCountryFan','just2playgg@gmail.com',NULL,'asdasdparecemeseguro123'
);
INSERT INTO "user" (id,name,username,email,image,password) VALUES (3,'Luis
Correia','luigi_darkside','luigi_mei<3@gmail.com',NULL,'passwordprofissional123');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (4,'Vicente
Espinha','vespinha','sdds_do_liedson@gmail.com','http://i.dailymail.co.uk/i/pix/2008/04/01/articl
e-1004361-00A0672B00000578-20_468x321_popup.jpg','queroverosportingcampeao');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (5,'Marco
Silva','Marcus_97','marcus_silva_97@gmail.com',NULL,'1234Marcus-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (6,'André
Ribeiro','programmer_rib','andre_ribeiro@gmail.com',NULL,'1234Andre-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (7,'Diana
Salgado','dianne_sal','diana_salgado_2@hotmail.com',NULL,'1234Diana-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (8,'Andrew
Tanenbaum','minix_lover','tanenbaum@gmail.com','https://pt.wikipedia.org/wiki/Andrew_Stuart
_Tanenbaum','1234Andrew-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (9,'Linus
Torvalds','linux_lover_52','linus_torvalds@gmail.com','https://en.wikipedia.org/wiki/Linus_Torval
ds','1234Linus-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (10,'Susana
Torres','susana_torres_92','susana_torres_92@gmail.com',NULL,'1234Susana-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (11,'Diogo
Mateus','diogo_76','diogo.mateus@gmail.com',NULL,'1234Diogo-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (12,'Adelino
Bastos','adele_boy_67','adelino.bastos@gmail.com',NULL,'1234Adelino-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (13,'Analisa
Correia','anacruza_dacapo','analisa_correia.93@gmail.com',NULL,'1234Analisa-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (14,'Madalena
Soares','madalena_muffin','madalena_muffin@gmail.com',NULL,'1234Madalena-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (15,'Pedro
Batista','batista_89','pedro.batista@gmail.com',NULL,'1234Pedro-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (16,'Raul
Vidal','sejam_felizes','raul.vidal@gmail.com',NULL,'1234Vidal-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (17,'Elliot
Alderson','Mr_Robot','im_not_mr_robot@gmail.com','https://shiiftyshift.deviantart.com/art/Hack
erman-643435212','1234Elliot-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (18,'Felix
Kjellberg','Pewdiepie','meme_review@gmail.com','https://gfycat.com/gifs/detail/hilariouseagerar
mednylonshrimp','1234Pewdiepie-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (19,'Helga
Smith','helga_93','helga_legit@gmail.com',NULL,'1234Helga-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (20,'Jeff
Sessions','my_name_jeff','my_name_jeff@gmail.com',NULL,'1234Jeff-');
```

```
INSERT INTO "user" (id,name,username,email,image,password) VALUES (21,'Diogo Dores','d_pain','fortnite@gmail.com',NULL,'fortniteisluv');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (22,'Ventura Pereira','fcparasempre','vp@gmail.com',NULL,'mourossucc');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (23,'Miguel Mano','blunky','fantano<3@gmail.com',NULL,'headwasclean');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (24,'Ze Borges','nbamaster','shaquille@gmail.com',NULL,'memes');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (25,'Joao Seixas','seixano','seixo@gmail.com',NULL,'-seixas-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (26,'Joao Conde','saltyaf','lol@gmail.com',NULL,'iactuallyhatelol');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (27,'Sofia Silva','aesthetic','blogger@gmail.com',NULL,'sofs');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (28,'Lil Pump','eskeeetit','gucci_gang@gmail.com',NULL,'fantanolovesme');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (29,'Jaden Smith','eyes','im_an_icon@gmail.com',NULL,'seriously_im_an_icon-');
INSERT INTO "user" (id,name,username,email,image,password) VALUES (30,'Mac DeMarco','pepperoni_playboy','blueboy@gmail.com',NULL,'freaking-');

INSERT INTO project (id,name,description,isPublic) VALUES (1,'Education Through the Web','A web page, made specifically to support students on their quest to learn more efficiently.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (2,'Cryptocurrency applied to auction houses','The rise of cryptocurrency demands that such a profitable business such as online auction houses remain up to date with technology.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (3,'Character design pipeline for a videogame','The struggle of designing a character for 3d in Blender', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (4,'LBAW project','Developing a entire web site in just a couple of months!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (6,'Secure platform for wire transfers','Platform that is totally secure for wire transfers between accounts in the same or different banks. The main focus is security.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (7,'Sigarra Website','New Sigarra Website, one more sensible and with more usability. Also with a mobile site that makes sense.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (8,'Backup Program using distributed systems','System that uses several servers to backup files through the network.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (9,'Open Source Half Life 3','The making of a dream, the highly requested Half Life 3. Will it achieve the high expectations?', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (10,'The Elder Scrolls V : Skyrim Mods','Because we just don''t have anything to do, and Bethesda doesn''t stop the exploitation of this game', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (11,'Security Course','Help us build this course on security with your knowledge, so we can inform and teach this important subject to everyone who wants to learn', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (12,'Minix 4','Minix is the best OS ever. This project will build a more complete and secure version, with new features!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (13,'8-bit Rick Roll', 'A way better version of Never Gonna Give You Up by Rick Astley.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (14,'Price Comparator', 'Developing a platform that provides you the best price from stores around you.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (15,'Minix GUI', 'Crating a graphical interface for the best operating system ever!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (16,'Paragon 2.0', 'Rebuilding the
```

```
well-know MOBA Paragon, ditched by Epic Games.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (17,'CGRA Submarine', 'Building a moving submarine able to shoot torpedos, using WebGL.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (18,'To-do List', 'A simple to-do list, for the everyday life. Project developed in NodeJS.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (19,'NECGM website', 'Rebuilding a new website for the NECGM from FEUP, now in NodeJS. New features will be added.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (20,'Fallout New Vegas VR', 'The best fallout game, now in VR. In development by Obsidian Entertainment.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (21,'Heart-rate app', 'Developing an app that allows you to measure your heart-rate, and analyse the data.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (22,'Fakecloud','A platform for music sharing ,similar to Soundcloud, but only covers are allowed.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (23,'FCPStream', 'Creating a streaming platform broadcasting matches and info about Futebol Clube do Porto.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (24,'Krita Mobile', 'An open-source mobile port of the designing tool, Krita.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (25,'Musebot', 'Open-source project for a music streaming bot for Discord.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (26,'Fashion platform', 'A platform to help you find the best deals in fashion retail.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (27,'NodeJS guide', 'Building a complete guide for NodeJS enthusiasts.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (28,'WebGL Water Asset', 'Open-source asset that emulates water physics', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (29,'Civ V Mods', 'Open-source mods for Civilization V, bringing some exciting features to the game.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (30,'Facial expression analyser','Software that allows to figure the mood of a person based on facial expressions.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (31,'Daily meme bot','A web bot that posts a meme everyday for your amusement', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (32,'Sports stats app','Developing an app that delivers the most reliable data about statitics in every sport', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (33,'Meme reviewer platform','A website to review the freshest memes and rate them!', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (34,'NoFakeNews', 'A web platform that delivers the most unbiased and reliable news', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (35,'PReis','Developing a high-performance linux distro that aims to be beautiful. Can also be used for gaming!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (36,'Pizza4All','An app created to increase the number of propotionally sliced pizzas', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (37,'Samurai Souls','Dark souls...but with samurais!!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (38,'Rambox', 'An app that has all your message apps.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (39,'Vinyl Finder', 'A platform for finding the best deals in vinyl records.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (40,'LaterX','A open-source laTEX editor. Made for students in a rush!', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (41,'Education Through the Web','A web page, made specifically to support students on their quest to learn more efficiently.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (42,'Cryptocurrency applied to auction houses','The rise of cryptocurrency demands that such a profitable business such as online auction houses remain up to date with technology.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (43,'Character design pipeline for a videogame','The struggle of designing a character for 3d in Blender', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (44,'LBAW Project','Developing a entire web site in just a couple of months!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (45,'CGRA Submarine', 'Building a moving submarine able.', FALSE);
```

```
INSERT INTO project (id,name,description,isPublic) VALUES (46,'Secure platform for wire transfers','Platform that is totally secure for wire transfers between accounts in the same or different banks. The main focus is security.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (47,'Sigarra Website','New Sigarra Website, one more sensible and with more usability. Also with a mobile site that makes sense.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (48,'Backup Program using distributed systems','System that uses several servers to backup files through the network.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (49,'Open Source Half Life 3','The making of a dream, the highly requested Half Life 3. Will it achieve the high expectations?', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (50,'The Elder Scrolls V : Skyrim Mods','Because we just don''t have anything to do, and Bethesda doesn''t stop the exploitation of this game', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (51,'Security Course','Help us build this course on security with your knowledge, so we can inform and teach this important subject to everyone who wants to learn', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (52,'Minix 4','Minix is the best OS ever. This project will build a more complete and secure version, with new features!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (53,'8-bit Rick Roll', 'A way better version of Never Gonna Give You Up by Rick Astley.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (54,'Price Comparator', 'Developing a platform that provides you the best price from stores around you.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (55,'Minix GUI', 'Crating a graphical interface for the best operating system ever!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (56,'Paragon 2.0', 'Rebuilding the well-know MOBA Paragon, ditched by Epic Games.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (57,'CGRA Submarine', 'Building a moving submarine able to shoot torpedos, using WebGL.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (58,'To-do List', 'A simple to-do list, for the everyday life. Project developed in NodeJS.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (59,'NECGM website', 'Rebuilding a new website for the NECGM from FEUP, now in NodeJS. New features will be added.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (60,'Fallout New Vegas VR', 'The best fallout game, now in VR. In development by Obsidian Entertainment.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (61,'Heart-rate app', 'Developing an app that allows you to measure your heart-rate, and analyse the data.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (62,'Fakecloud','A platform for music sharing ,similar to Soundcloud, but only covers are allowed.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (63,'FCPStream', 'Creating a streaming platform broadcasting matches and info about Futebol Clube do Porto.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (64,'Krita Mobile', 'An open-source mobile port of the designing tool, Krita.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (65,'Musebot', 'Open-source project for a music streaming bot for Discord.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (66,'Fashion platform', 'A platform to help you find the best deals in fashion retail.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (67,'NodeJS guide', 'Building a complete guide for NodeJS enthusiasts.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (68,'WebGL Water Asset', 'Open-source asset that emulates water physics', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (69,'Civ V Mods', 'Open-source mods for Civilization V, bringing some exciting features to the game.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (70,'Facial expression analyser','Software that allows to figure the mood of a person based on facial expressions.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (71,'Daily meme bot','A web bot that posts a meme everyday for your amusement', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (72,'Sports stats app','Developing an app that delivers the most reliable data about statitics in every sport', FALSE);
```

```
INSERT INTO project (id,name,description,isPublic) VALUES (73,'Meme reviewer platform','A website to review the freshest memes and rate them!', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (74,'NoFakeNews', 'A web platform that delivers the most unbiased and reliable news', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (75,'PReis','Developing a high-performance linux distro that aims to be beautiful. Can also be used for gaming!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (76,'Pizza4All','An app created to increase the number of propotionally sliced pizzas', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (77,'Samurai Souls','Dark souls...but with samurais!!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (78,'Rambox', 'An app that has all your message apps.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (79,'Vinyl Finder', 'A platform for finding the best deals in vinyl records.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (80,'LaterX','A open-source laTEX editor. Made for students in a rush!', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (81,'Education Through the Web','A web page, made specifically to support students on their quest to learn more efficiently.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (82,'Cryptocurrency applied to auction houses','The rise of cryptocurrency demands that such a profitable business such as online auction houses remain up to date with technology.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (83,'Character design pipeline for a videogame','The struggle of designing a character for 3d in Blender', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (84,'LBAW Project','Developing a entire web site in just a couple of months!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (85,'Security Course','Help us build this course on security with your knowledge, so we can inform and teach this important subject to everyone who wants to learn', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (86,'Secure platform for wire transfers','Platform that is totally secure for wire transfers between accounts in the same or different banks. The main focus is security.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (87,'CGRA Submarine', 'Building a moving submarine able to shoot torpedos, using WebGL.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (88,'Sigarra Website','New Sigarra Website, one more sensible and with more usability. Also with a mobile site that makes sense.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (89,'Backup Program using distributed systems','System that uses several servers to backup files through the network.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (90,'Open Source Half Life 3','The making of a dream, the highly requested Half Life 3. Will it achieve the high expectations?', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (91,'The Elder Scrolls V : Skyrim Mods','Because we just don''t have anything to do, and Bethesda doesn''t stop the exploitation of this game', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (92,'Minix 4','Minix is the best OS ever. This project will build a more complete and secure version, with new features!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (93,'8-bit Rick Roll', 'A way better version of Never Gonna Give You Up by Rick Astley.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (94,'Price Comparator', 'Developing a platform that provides you the best price from stores around you.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (95,'Minix GUI', 'Crating a graphical interface for the best operating system ever!', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (96,'Paragon 2.0', 'Rebuilding the well-know MOBA Paragon, ditched by Epic Games.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (97,'CGRA Submarine', 'Building a moving submarine able to shoot torpedos, using WebGL.', FALSE);
INSERT INTO project (id,name,description,isPublic) VALUES (98,'To-do List', 'A simple to-do list, for the everyday life. Project developed in NodeJS.', TRUE);
INSERT INTO project (id,name,description,isPublic) VALUES (99,'NECGM website', 'Rebuilding a new website for the NECGM from FEUP, now in NodeJS. New features will be added.', FALSE);
```

```
INSERT INTO project (id,name,description,isPublic) VALUES (100,'Fallout New Vegas VR', 'The best fallout game, now in VR. In development by Obsidian Entertainment.', FALSE);

INSERT INTO category (id, name) VALUES (1,'Entertainment');
INSERT INTO category (id, name) VALUES (3,'Productivity');
INSERT INTO category (id, name) VALUES (4,'Software');
INSERT INTO category (id, name) VALUES (5,'Application');
INSERT INTO category (id, name) VALUES (6,'Education');
INSERT INTO category (id, name) VALUES (7,'Business');
INSERT INTO category (id, name) VALUES (8,'Web');
INSERT INTO category (id, name) VALUES (9,'Game');
INSERT INTO category (id, name) VALUES (10,'Open Source');
INSERT INTO category (id, name) VALUES (11,'Graphic');
INSERT INTO category (id, name) VALUES (12,'Design');
INSERT INTO category (id, name) VALUES (13,'Sports');
INSERT INTO category (id, name) VALUES (14,'Music');
INSERT INTO category (id, name) VALUES (15,'Financial');
INSERT INTO category (id, name) VALUES (16,'Medical');
INSERT INTO category (id, name) VALUES (17,'Information');
INSERT INTO category (id, name) VALUES (18,'Lifestyle');
INSERT INTO category (id, name) VALUES (19,'Shopping');
INSERT INTO category (id, name) VALUES (20,'Social Networking');
INSERT INTO category (id, name) VALUES (21,'Multimedia');

INSERT INTO project_categories (project_id, category_id) VALUES (1,4);
INSERT INTO project_categories (project_id, category_id) VALUES (1,6);
INSERT INTO project_categories (project_id, category_id) VALUES (2,7);
INSERT INTO project_categories (project_id, category_id) VALUES (2,4);
INSERT INTO project_categories (project_id, category_id) VALUES (3,1);
INSERT INTO project_categories (project_id, category_id) VALUES (3,3);
INSERT INTO project_categories (project_id, category_id) VALUES (4,6);
INSERT INTO project_categories (project_id, category_id) VALUES (6,7);
INSERT INTO project_categories (project_id, category_id) VALUES (7,6);
INSERT INTO project_categories (project_id, category_id) VALUES (6,4);
INSERT INTO project_categories (project_id, category_id) VALUES (7,8);
INSERT INTO project_categories (project_id, category_id) VALUES (8,4);
INSERT INTO project_categories (project_id, category_id) VALUES (9,9);
INSERT INTO project_categories (project_id, category_id) VALUES (9,1);
INSERT INTO project_categories (project_id, category_id) VALUES (10,9);
INSERT INTO project_categories (project_id, category_id) VALUES (10,1);
INSERT INTO project_categories (project_id, category_id) VALUES (9,10);
INSERT INTO project_categories (project_id, category_id) VALUES (11,6);
INSERT INTO project_categories (project_id, category_id) VALUES (12,4);
INSERT INTO project_categories (project_id, category_id) VALUES (13,14);
INSERT INTO project_categories (project_id, category_id) VALUES (14,4);
INSERT INTO project_categories (project_id, category_id) VALUES (15,6);
INSERT INTO project_categories (project_id, category_id) VALUES (16,7);
INSERT INTO project_categories (project_id, category_id) VALUES (17,4);
INSERT INTO project_categories (project_id, category_id) VALUES (18,1);
INSERT INTO project_categories (project_id, category_id) VALUES (19,3);
INSERT INTO project_categories (project_id, category_id) VALUES (20,6);
INSERT INTO project_categories (project_id, category_id) VALUES (21,7);
INSERT INTO project_categories (project_id, category_id) VALUES (22,6);
INSERT INTO project_categories (project_id, category_id) VALUES (23,4);
INSERT INTO project_categories (project_id, category_id) VALUES (24,8);
INSERT INTO project_categories (project_id, category_id) VALUES (25,4);
INSERT INTO project_categories (project_id, category_id) VALUES (26,9);
INSERT INTO project_categories (project_id, category_id) VALUES (27,1);
INSERT INTO project_categories (project_id, category_id) VALUES (28,9);
INSERT INTO project_categories (project_id, category_id) VALUES (29,1);
INSERT INTO project_categories (project_id, category_id) VALUES (30,10);
INSERT INTO project_categories (project_id, category_id) VALUES (31,6);
```

```
INSERT INTO project_categories (project_id, category_id) VALUES (32,4);
INSERT INTO project_categories (project_id, category_id) VALUES (32,14);
INSERT INTO project_categories (project_id, category_id) VALUES (33,4);
INSERT INTO project_categories (project_id, category_id) VALUES (33,14);
INSERT INTO project_categories (project_id, category_id) VALUES (34,4);
INSERT INTO project_categories (project_id, category_id) VALUES (35,6);
INSERT INTO project_categories (project_id, category_id) VALUES (36,7);
INSERT INTO project_categories (project_id, category_id) VALUES (37,4);
INSERT INTO project_categories (project_id, category_id) VALUES (38,1);
INSERT INTO project_categories (project_id, category_id) VALUES (39,3);
INSERT INTO project_categories (project_id, category_id) VALUES (40,6);
INSERT INTO project_categories (project_id, category_id) VALUES (41,7);
INSERT INTO project_categories (project_id, category_id) VALUES (42,6);
INSERT INTO project_categories (project_id, category_id) VALUES (43,4);
INSERT INTO project_categories (project_id, category_id) VALUES (44,8);
INSERT INTO project_categories (project_id, category_id) VALUES (45,4);
INSERT INTO project_categories (project_id, category_id) VALUES (46,9);
INSERT INTO project_categories (project_id, category_id) VALUES (47,1);
INSERT INTO project_categories (project_id, category_id) VALUES (48,9);
INSERT INTO project_categories (project_id, category_id) VALUES (49,1);
INSERT INTO project_categories (project_id, category_id) VALUES (50,10);
INSERT INTO project_categories (project_id, category_id) VALUES (51,6);
INSERT INTO project_categories (project_id, category_id) VALUES (52,4);
INSERT INTO project_categories (project_id, category_id) VALUES (52,14);
INSERT INTO project_categories (project_id, category_id) VALUES (53,4);
INSERT INTO project_categories (project_id, category_id) VALUES (53,14);
INSERT INTO project_categories (project_id, category_id) VALUES (54,4);
INSERT INTO project_categories (project_id, category_id) VALUES (55,6);
INSERT INTO project_categories (project_id, category_id) VALUES (56,7);
INSERT INTO project_categories (project_id, category_id) VALUES (57,4);
INSERT INTO project_categories (project_id, category_id) VALUES (58,1);
INSERT INTO project_categories (project_id, category_id) VALUES (59,3);
INSERT INTO project_categories (project_id, category_id) VALUES (60,6);
INSERT INTO project_categories (project_id, category_id) VALUES (61,7);
INSERT INTO project_categories (project_id, category_id) VALUES (62,6);
INSERT INTO project_categories (project_id, category_id) VALUES (63,4);
INSERT INTO project_categories (project_id, category_id) VALUES (64,8);
INSERT INTO project_categories (project_id, category_id) VALUES (65,4);
INSERT INTO project_categories (project_id, category_id) VALUES (66,9);
INSERT INTO project_categories (project_id, category_id) VALUES (67,1);
INSERT INTO project_categories (project_id, category_id) VALUES (68,9);
INSERT INTO project_categories (project_id, category_id) VALUES (69,1);
INSERT INTO project_categories (project_id, category_id) VALUES (60,10);
INSERT INTO project_categories (project_id, category_id) VALUES (61,6);
INSERT INTO project_categories (project_id, category_id) VALUES (62,4);
INSERT INTO project_categories (project_id, category_id) VALUES (62,14);
INSERT INTO project_categories (project_id, category_id) VALUES (63,14);
INSERT INTO project_categories (project_id, category_id) VALUES (64,4);
INSERT INTO project_categories (project_id, category_id) VALUES (65,6);
INSERT INTO project_categories (project_id, category_id) VALUES (66,7);
INSERT INTO project_categories (project_id, category_id) VALUES (67,4);
INSERT INTO project_categories (project_id, category_id) VALUES (68,11);
INSERT INTO project_categories (project_id, category_id) VALUES (69,3);
INSERT INTO project_categories (project_id, category_id) VALUES (70,6);
INSERT INTO project_categories (project_id, category_id) VALUES (71,17);
INSERT INTO project_categories (project_id, category_id) VALUES (72,14);
INSERT INTO project_categories (project_id, category_id) VALUES (73,4);
INSERT INTO project_categories (project_id, category_id) VALUES (74,8);
INSERT INTO project_categories (project_id, category_id) VALUES (75,4);
INSERT INTO project_categories (project_id, category_id) VALUES (76,9);
INSERT INTO project_categories (project_id, category_id) VALUES (77,1);
INSERT INTO project_categories (project_id, category_id) VALUES (78,9);
```

```
INSERT INTO project_categories (project_id, category_id) VALUES (79,1);
INSERT INTO project_categories (project_id, category_id) VALUES (80,10);
INSERT INTO project_categories (project_id, category_id) VALUES (81,6);
INSERT INTO project_categories (project_id, category_id) VALUES (82,16);
INSERT INTO project_categories (project_id, category_id) VALUES (82,14);
INSERT INTO project_categories (project_id, category_id) VALUES (83,4);
INSERT INTO project_categories (project_id, category_id) VALUES (83,14);
INSERT INTO project_categories (project_id, category_id) VALUES (84,4);
INSERT INTO project_categories (project_id, category_id) VALUES (85,6);
INSERT INTO project_categories (project_id, category_id) VALUES (86,7);
INSERT INTO project_categories (project_id, category_id) VALUES (87,4);
INSERT INTO project_categories (project_id, category_id) VALUES (88,1);
INSERT INTO project_categories (project_id, category_id) VALUES (89,16);
INSERT INTO project_categories (project_id, category_id) VALUES (90,6);
INSERT INTO project_categories (project_id, category_id) VALUES (91,7);
INSERT INTO project_categories (project_id, category_id) VALUES (92,6);
INSERT INTO project_categories (project_id, category_id) VALUES (93,4);
INSERT INTO project_categories (project_id, category_id) VALUES (94,18);
INSERT INTO project_categories (project_id, category_id) VALUES (95,4);
INSERT INTO project_categories (project_id, category_id) VALUES (96,21);
INSERT INTO project_categories (project_id, category_id) VALUES (97,1);
INSERT INTO project_categories (project_id, category_id) VALUES (98,9);
INSERT INTO project_categories (project_id, category_id) VALUES (99,1);
INSERT INTO project_categories (project_id, category_id) VALUES (100,10);

INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (1, now(), 1,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (2, now(), 1,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (2, now(), 2,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (3, now(), 3,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (4, now(), 3,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (1, now(), 4,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (2, now(), 4,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (3, now(), 4,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (4, now(), 4,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (8, now(), 12,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (9, now(), 12,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (7, now(), 12,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 7,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (16, now(), 12,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (12, now(), 6,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 6,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (17, now(), 11,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (19, now(), 11,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (18, now(), 9,
```

```
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (1, now(), 9,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (2, now(), 11,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (11, now(), 8,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (20, now(), 10,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (8, now(), 9,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (15, now(), 7,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (5, now(), 12,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (13, now(), 12,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (10, now(), 6,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (9, now(), 6,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (12, now(), 7,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (11, now(), 7,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (14, now(), 8,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (15, now(), 8,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (17, now(), 10,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (18, now(), 10,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (3, now(), 10,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 12,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (14, now(), 12,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (17, now(), 2,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 2,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (11, now(), 3,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (1, now(), 3,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (7, now(), 4,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (18, now(), 4,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (16, now(), 2,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (1, now(), 14,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (2, now(), 15,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (2, now(), 16,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (3, now(), 17,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (4, now(), 18,
```

```
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (1, now(), 19,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (2, now(), 20,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (3, now(), 21,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (4, now(), 22,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (8, now(), 23,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (9, now(), 24,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (7, now(), 25,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 26,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (16, now(), 27,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (12, now(), 28,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 29,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (17, now(), 30,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (19, now(), 31,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (18, now(), 32,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (1, now(), 33,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (2, now(), 34,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (11, now(), 35,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (20, now(), 36,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (8, now(), 37,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (15, now(), 38,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (5, now(), 39,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (13, now(), 40,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (10, now(), 41,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (9, now(), 42,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (12, now(), 43,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (11, now(), 44,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (14, now(), 45,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (15, now(), 46,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (17, now(), 47,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (18, now(), 48,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (3, now(), 49,
```

```
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 50,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (14, now(), 51,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (17, now(), 52,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 53,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (11, now(), 54,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (1, now(), 55,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (7, now(),56,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (18, now(), 58,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (16, now(), 59,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (16, now(), 60,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (3, now(), 61,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (4, now(), 62,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (8, now(), 63,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (9, now(), 64,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (7, now(), 65,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 66,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (16, now(),67,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (12, now(), 68,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 69,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (17, now(), 70,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (19, now(), 71,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (18, now(), 72,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (1, now(), 73,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (2, now(), 74,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (11, now(), 75,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (20, now(), 76,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (8, now(), 77,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (15, now(), 78,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (5, now(), 79,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (13, now(), 80,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (10, now(), 81,
```

```
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (9, now(), 82,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (12, now(), 83,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (11, now(), 84,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (14, now(), 85,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (15, now(), 86,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (17, now(), 87,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (18, now(), 88,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (3, now(), 89,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 90,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (14, now(), 91,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (17, now(), 92,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (6, now(), 93,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (11, now(), 94,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (1, now(), 95,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (7, now(),96,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (18, now(), 98,
FALSE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (16, now(), 99,
TRUE);
INSERT INTO project_members (user_id, date, project_id, isCoordinator) VALUES (16, now(),
100, TRUE);

INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (1,'Could
there be a section about Programming?','I think we are focusing more on mathematics and
programming is being left out. It is an interesting subject and very useful these days!',
now(),1,2);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (2,'I think I
broke the project....oopsie!','Ah...guys, it ain''t working! Could someone fix this
please!?\n*screeching*', now(),4,18);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (3,'Another
game with a game with a female lead character....boring!','Guys, come on! Not again! I know it
is a trend, but why not vary and make, for example, a game with several principal characters,
where you can play with different characters, both in gender but also in race. This game could
do it, the story allows it!', now(),3,4);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (4,'I don''t
know...will this really work?','Will it be really possible to make this game? It is HL3 and, well, is
open source. By the way, isn''t it kinda illegal? Doesn''t Valve has the rights to this?\nJust
saying...', now(),9,18);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (5,'I have a
great idea!','Lets make the character like Geralt of Witcher 3 and the dragons will be Roach!
Ah, hilarious!\nMy name''s Jeff!', now(),10,20);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (6,'Did you
know?','Linux is kinda based on Minix...well not really, but first I wanted to improve Minix
features but Andrew didn''t wanted me to, so I based some of Linux in Minix... but I changed
lots of things, of course!', now(),12,9);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (7,'I believe
```

```
the mock ups are kinda ugly...','We should do it again',now(),4,18);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (8,'I have a great idea!','Let''s make the character like Geralt of Witcher 3 and the dragons will be Roach! Ah, hilarious!\nMy name''s Jeff!', now(),10,20);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (9,'Did you know?','Linux is kinda based on Minix...well not really, but first I wanted to improve Minix features but Andrew didn''t wanted me to, so I based some of Linux in Minix... but I changed lots of things, of course!', now(),12,9);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (10,'Witcher 3 quest!','Could someone give some hints about where i can find cedaline in witcher 3?', now(),90,2);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (11,'Could there be a section about Programming?','I think we are focusing more on mathematics and programming is being left out. It is an interesting subject and very useful these days!', now(),1,2);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (12,'I think I broke the project....oopsie!','Ah...guys, it ain''t working! Could someone fix this please!?\n*screeching*', now(),4,18);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (13,'Another game with a game with a female lead character....boring!','Guys, come on! Not again! I know it is a trend, but why not vary and make, for example, a game with several principal characters, where you can play with different characters, both in gender but also in race. This game could do it, the story allows it!', now(),3,4);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (14,'I don''t know...will this really work?','Will it be really possible to make this game? It is HL3 and, well, is open source. By the way, isn''t it kinda illegal? Doesn''t Valve has the rights to this?\nJust saying...', now(),9,18);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (15,'I have a great idea!','Let''s make the character like Geralt of Witcher 3 and the dragons will be Roach! Ah, hilarious!\nMy name''s Jeff!', now(),10,20);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (16,'Did you know? ','Linux is kinda based on Minix...well not really, but first I wanted to improve Minix features but Andrew didn''t wanted me to, so I based some of Linux in Minix... but I changed lots of things, of course!', now(),12,9);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (17,'Witcher 3 quest!','Could someone give some hints about where i can find cedaline in witcher 3?', now(),33,3);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (18,'I have a great idea!','Let''s make the character like Geralt of Witcher 3 and the dragons will be Roach! Ah, hilarious!\nMy name''s Jeff!', now(),10,20);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (19,'Did you know? ','Linux is kinda based on Minix...well not really, but first I wanted to improve Minix features but Andrew didn''t wanted me to, so I based some of Linux in Minix... but I changed lots of things, of course!', now(),12,9);
INSERT INTO thread (id,name,description,date,project_id,user_creator_id) VALUES (20,'Witcher 3 quest!','Could someone give some hints about where i can find cedaline in witcher 3?', now(),78,1);

INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (1,'Mock-Ups','2018-05-20 00:00:00+01',1,2,5);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (2,'Database structure','2018-05-20 00:00:00+01',1,2,3);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (3,'Website','2018-04-20 12:00:00+01',2,16,5);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (4,'Build Security','2018-05-20 08:00:00+01',2,16,5);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (5,'Draw Mock-up','2018-04-12 23:59:00+01',3,11,3);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (6,'Design with blender','2018-04-20 22:59:00+01',3,1,7);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (7,'Database','2018-04-20 23:00:00+01',4,3,7);
```

```
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (8,'Make Website','2018-05-21 23:00:00+01',4,2,10);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (9,'Mobile App','2018-05-20 23:00:00+01',6,6,10);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (10,'Security Verifications','2018-05-25 23:00:00+01',6,6,8);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (11,'Mock-Ups','2018-05-20 23:00:00+01',7,6,7);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (12,'Security','2018-05-30 23:00:00+01',7,6,7);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (13,'Client RMI','2018-04-29 23:00:00+01',8,11,7);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (14,'Communications between servers','2018-05-02 23:00:00+01',8,11,8);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (15,'Write history','2018-05-20 23:00:00+01',9,1,6);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (16,'Draw characters','2018-05-20 00:00:00+01',9,1,8);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (17,'Decide Improvements','2018-04-18 23:00:00+01',10,18,5);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (18,'Make models 3D','2018-04-30 23:00:00+01',10,17,10);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (19,'Design Course Program','2018-04-18 23:00:00+01',11,17,3);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (20,'Introduction','2018-04-22 23:00:00+01',11,17,5);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (21,'Decide Improvements','2018-04-18 23:00:00+01',12,8,3);
INSERT INTO sprint (id,name,deadline,project_id,user_creator_id,effort) VALUES (22,'Kernel','2018-04-30 23:00:00+01',12,8,20);

INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (1,'Index Page','Make a responsive mock up of the index page, with tonalities of blue and gold. Images will be added next',1,1,1);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (2,'Video Page','Responsive page to allocate many videos',2,1,1);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (3,'Basic database','Solid structure of basic database to support video',1,1,2);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (4,'Security','Implement mechanism to prevent SQL Injections',2,1,2);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (5,'Database','Solid and secure database',2,2,3);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (6,'Transfer Page','',2,2,3);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (7,'Cross-Site Scripting Security','Implement mechanism to prevent XSS',2,2,4);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (8,'Cross-Site Request Forgery','Implement mechanism to prevent CSRF',2,2,4);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (9,'Make principal character','Female, long dark hair, blue jeans and flannel shirt, nerdy look',1,3,5);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (10,'Villain character','Guy, normal person, glasses and with a trustworthy expression',1,3,5);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (11,'Sidekick character','Flashy character, guy, always smiling and with a funny haircut and style.',1,3,5);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (12,'Basic design','',3,3,6);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (13,'Animations','Walking, jumping, rolling',4,3,6);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (14,'Populate','At least 25 tasks',3,4,7);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (15,'Make queries','To
```

```
all the tables',2,4,7);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (16,'Triggers','',1,4,7);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (17,'project
Page','Use AJAX to switch between the possible pages of the project page.\nMake animations
fluid and natural.',6,4,8);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (18,'Resolve bug on
the forum page','CSS and Javascript bug, doesn''t show information about the date because it is
cut off, and the date is wrongly calculated',2,4,8);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (19,'Put in Google
Play','Share the application in Google Play',1,6,9);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (20,'Connect with
several banks','Get agreements with several banks to access to their platform.',2,6,9);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (21,'Security','Make
the mobile app secure',4,6,9);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (22,'Hire company
specialized in security','',2,6,10);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (23,'Design Index
Page','Make a pleasant and informative index page',4,7,11);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (24,'Make responsive
to mobile devices','',3,7,11);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (25,'XXS
security','',2,7,12);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (26,'CSRF
Security','',2,7,12);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (27,'SQL Injections
Verification','Very important verification!',2,7,12);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (28,'Make reference
to the registry','Don''t forget to use the right instructions,
here:\nhttps://docs.oracle.com/javase/tutorial/rmi/client.html',2,8,13);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (29,'Code','',4,8,13);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (30,'Create multicast
channels to every socket used','Don''t forget to join by group and use different IPs to each
socket',2,8,14);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (31,'Concurrent
Mechanism','Don''t forget to check the replicationDegree and send only to that number of
servers. Check if the stored messages are received, and in their correct number.\nAlso, it has to
be possible to process several requests at once!\nUse threads and/or threadPools!',5,8,14);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (32,'Main Quest','It
has to start where the previous one has ended',2,9,15);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (33,'Write 3 side-
quests','Have to be at least 45min long',3,9,15);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (34,'Principal
Character - Gordon Freeman','Keep it close to the original one',2,9,16);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (35,'G-Man','Keep it
mysterious',2,9,16);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (36,'Current Meme
incorporation','What meme to use in this mod?',1,10,17);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (37,'Decision to make
this a serious or a stupid mod','',2,10,17);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (38,'Chicken
Model','Yap, a chicken model, we are going with that',2,10,18);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (39,'Decide number
of chapters','',1,11,19);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (40,'Pen Testing?','Is it
possible to make a chapter about this one, and an extensive one?',1,11,19);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (41,'Introduce
yourself and the course','Explain who you are, what you do for a living and your
motivations.\nExplain what are the objectives of the course, the resources needed and the
degree of difficulty.',1,11,20);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (42,'Course
mapping','Explain the different topics that will be covered, as well as their
importance.',1,11,20);
```

```
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (43,'Write in the comments bellow your opinion','',1,12,21);
INSERT INTO task (id,name,description,effort,project_id,sprint_id) VALUES (44,'Rewrite function about sound drivers','This function contains a bug with specific sound cards',4,12,22);

INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (1,'There will be a part of the website that will focus totally on Programming but, for now, it is more imperative that we finish the Mathematics chapters.',now(),1,NULL,1);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (2,'Ah, I didn''t know! Thank you!',now(),2,NULL,1);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (3,'Oh man, not again! I will see what is broke then, but please say something before you go there. I don''t know what you do, but you have a knack for breaking websites!',now(),3,NULL,2);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (4,'I agree with him, it makes total sense! It is a history similar to Doctor Who, we have the material to make it like it.',now(),3,NULL,3);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (5,'Okay, we will see about it! For now keep working on the character chosen, and we will see about changing the history.\n\nThank you for the suggestion!',now(),11,NULL,3);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (6,'Well, we won''t gain money from this, so I guess it is legal...ah, right?',now(),1,NULL,4);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (7,'In my opinion, that is an awful idea. It doesn''t make any sense whatsoever!',now(),17,NULL,5);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (8,'I kinda like it!',now(),18,NULL,5);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (9,'Why are you always telling this story? Everyone knows it!',now(),8,NULL,6);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (10,'It is an interesting fact',now(),9,NULL,6);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (11,'Is it possible for someone to give more detailed points about this one?',now(),3,10,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (12,'The point is for the villain to be like a normal person, like a friendly neighbor or a friendly coworker',now(),11,10,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (13,'Is it only related to checking if a member is a coordinator or team member when doing some type of action?',now(),7,16,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (14,'Not only, but also checking if a value of effort on a sprint is exceeded by its tasks.',now(),2,16,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (15,'Well, of course it would be this',now(),3,38,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (16,'I would like to do this one',now(),2,7,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (17,'I think this one isn''t really a Javascript bug but a PHP error...there isn''t any way of showing the date in the php file',now(),7,18,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (18,'I can do this one, I have experience with security. It helps to save money',now(),10,22,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (19,'Isn''t there an easier way of doing this? It is a lot of work',now(),18,14,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (20,'I don''t think so, this is the only way',now(),3,14,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (21,'It would help if there were more than one person doing this',now(),2,14,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (22,'I''ll help has well!',now(),1,14,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (23,'Can it use glasses or some kind of googles?',now(),4,11,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (24,'Sure, any suggestion can be done',now(),11,11,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (25,'I''ve done this, but it isn''t working. Can someone help?',now(),14,30,NULL);
```

```sql
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (26,'I think I know how...I think you are forgetting to create e InetAddress and are passing only a string with the address.',now(),15,30,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (27,'You''re right, thanks!',now(),14,30,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (28,'Yes! I think that kind of content would be very important! If there isn''t any problem, I would like to do it',now(),19,40,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (29,'Of course!',now(),17,40,NULL);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (30,'Oh, shut up, you know nothing!',now(),4,NULL,3);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (31,'Jeeeesss, you are so dumb!',now(),7,NULL,2);
INSERT INTO comment (id,content,date,user_id,task_id,thread_id) VALUES (32,'SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM SPAM',now(),18,NULL,4);

INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',1);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',2);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',3);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',4);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Completed',5);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',6);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',7);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',8);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',9);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',10);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',11);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',12);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Completed',13);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',14);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',15);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',16);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',17);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',18);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',19);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',20);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Completed',21);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',22);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Created',21);
INSERT INTO sprint_state_record (id,date,state,sprint_id) VALUES (DEFAULT,now(),'Completed',21);

INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES (DEFAULT,now(),'Created',2,1);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES (DEFAULT,now(),'Created',2,2);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES (DEFAULT,now(),'Created',2,3);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES (DEFAULT,now(),'Created',2,4);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES (DEFAULT,now(),'Created',16,5);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
```

```
(DEFAULT,now(),'Created',16,6);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',16,7);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',16,8);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',11,9);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',1,10);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',11,11);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',1,12);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',11,13);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',3,14);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',4,15);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',2,16);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',1,17);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',2,18);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',6,19);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',6,20);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',6,21);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',6,22);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',6,23);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',6,24);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',6,25);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',6,26);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',6,27);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',11,28);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',11,29);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',11,30);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',11,31);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',1,32);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',1,33);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',1,34);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',1,35);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',17,36);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
```

```
(DEFAULT,now(),'Created',18,37);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',18,38);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',17,39);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',17,40);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',17,41);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',17,42);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',8,43);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',8,44);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',1,1);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',1,3);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',2,6);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',3,9);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',4,9);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',3,10);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Unnassigned',3,10);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',4,10);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Completed',3,9);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Completed',4,10);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',4,11);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Completed',4,11);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',7,14);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Completed',15,16);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',14,28);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Completed',14,28);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',15,29);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Assigned',14,29);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Completed',14,29);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Completed',8,43);
```

```
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Created',9,43);
INSERT INTO task_state_record (id,date,state,user_completed_id,task_id) VALUES
(DEFAULT,now(),'Completed',9,43);

INSERT INTO report (id,date,summary,user_id,type,comment_reported_id,user_reported_id)
VALUES (DEFAULT, now(),'It is an offensive comment and totally out of
place',3,'commentReported',30,NULL);
INSERT INTO report (id,date,summary,user_id,type,comment_reported_id,user_reported_id)
VALUES (DEFAULT,now(),'comment extremely offensive',18,'commentReported',32,NULL);
INSERT INTO report (id,date,summary,user_id,type,comment_reported_id,user_reported_id)
VALUES (DEFAULT,now(),'It''s clearly spam, and it should be removed, as well as the
user',8,'commentReported',31,NULL);
INSERT INTO report (id,date,summary,user_id,type,comment_reported_id,user_reported_id)
VALUES (DEFAULT,now(),'Clearly a Nazi, it''s what J.K.Rowling and the WSJ
says...',20,'userReported',NULL,18);

INSERT INTO invite (id,date,user_invited_id,project_id,user_who_invited_id) VALUES
(DEFAULT,now(),7,9,1);
INSERT INTO invite (id,date,user_invited_id,project_id,user_who_invited_id) VALUES
(DEFAULT,now(),4,12,NULL);
INSERT INTO invite (id,date,user_invited_id,project_id,user_who_invited_id) VALUES
(DEFAULT,now(),4,11,NULL);
INSERT INTO invite (id,date,user_invited_id,project_id,user_who_invited_id) VALUES
(DEFAULT,now(),6,8,11);
```

Link to SQL Script

Grupo 1717, 3/4/2018

Ana Margarida Oliveira Pinheiro da Silva, up201505505@fe.up.pt

Luís Miguel Cardoso Lopes Correia, up201503342@fe.up.pt

Pedro Daniel dos Santos Reis, up201506046@fe.up.pt

Vicente Fernandes Ramada Caldeira Espinha, up201503764@fe.up.pt