

Database Project

CS4416

Database Management | 09/04/2017

Matthew Murphy – 15147193

Daniel Keeley - 15189082

Shíofra Keogh - 15145247

Eva Finn - 15172899

# 1.

Matthew Murphy – Worked mainly on stored procedures and database structure.

Daniel Keeley – Worked on project report, database structure and populating the tables with data.

Shiofra Keogh – Worked on database structure, triggers and ensuring all tables were in 3NF.

Eva Finn – Worked on database structure design and writing SQL queries.

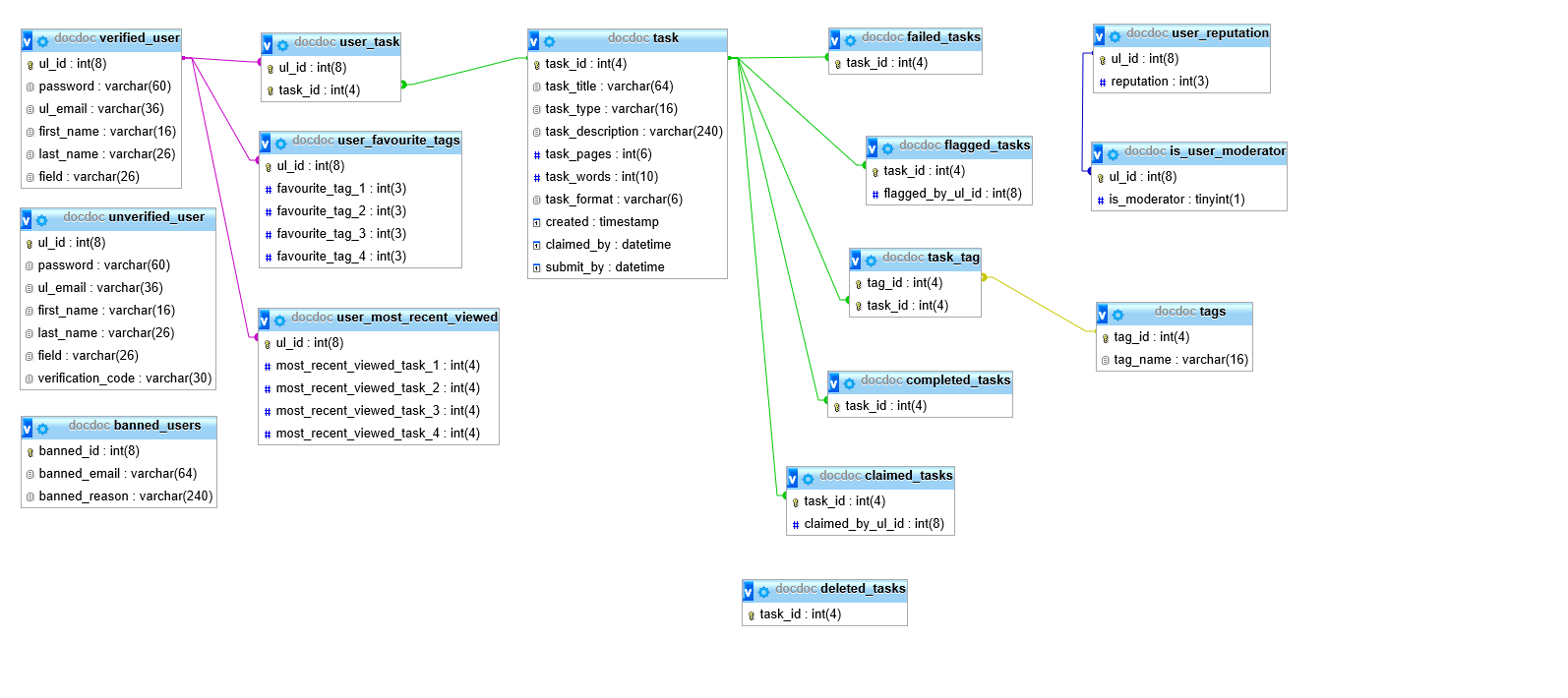
## 2.

Our database is based on our website we made for our software development module. The database stores information for a website called DocDoc which is designed to allow students and peers to proofread and review each other’s work such as papers, thesis’s etc.

Our database implements tables intended for online use such as verified\_user and unverified\_user depending on whether they have verified their email or not. Banned\_users store all the ul\_id for banned users so that banned users cannot make a new account with the banned id. Tags are an important aspect of the website. Tasks tables are used to store the information about a user’s tasks and are connected to the user and claimant. Each task can have up to 4 tags which allow users to easily search for specific topics they’re interested in. Tags also allowed us to generate personalized task feeds for each user.

We used a junction table to connect various tables together such as unverified\_user and verified\_user relates to task through user\_task. We chose this structure to ensure a normalized design. Flagged\_tasks will allow users to flag inappropriate tasks and let them be reviewed by mods. Flagged\_tasks are kept in a separate table for mods to verify if the task is offensive and either re-instate the task or permanently remove it.

DocDoc Entity-Relationship Diagram:



3.

Key:

|  |  |
| --- | --- |
|  | Primary Key |
|  | Foreign Key |
|  | Primary and Foreign Key |

unverified user

|  |  |  |  |
| --- | --- | --- | --- |
| ul\_id | password | ul\_email | first\_name |
| 15189082 | Consuelo20 | 15189082@studentmail.ul.ie | Danilo |

|  |  |  |
| --- | --- | --- |
| last\_name | field | verification\_code |
| Montoya | Aeronautical | 34256 |

verified\_user

|  |  |  |  |
| --- | --- | --- | --- |
| ul\_id | password | ul\_email | first\_name |
| 15112345 | ColombiaNo1 | 15112345@studentmail.ul.ie | Daniela |

|  |  |
| --- | --- |
| last\_name | field |
| Sosa | Engineering |

banned\_user

|  |  |  |
| --- | --- | --- |
| banned\_id | banned\_email | banned\_reason |
| 15112345 | 15112345@studentmail.ul.ie | Spam |

task

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| task\_id | task\_title | task\_type | task\_description | task\_pages |
| 1 | Aeronautical Engineering | Thesis | A thesis on how planes can… | 400 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| task\_words | task\_format | created | claimed\_by | submit\_by |
| 3450 | .pdf | 10/04/2017 | 15/05/2017 | 25/05/2017 |

claimed\_tasks

|  |  |
| --- | --- |
| task\_id | claimed\_by\_ul\_id |
| 2 | 15167524 |

deleted\_tasks

|  |
| --- |
| task\_id |
| 3 |

flagged\_task

|  |  |
| --- | --- |
| task\_id | flagged\_by\_ul\_id |
| 4 | 15162543 |

user\_task

|  |  |
| --- | --- |
| user\_id | task\_id |
| 15189082 | 4 |

user\_favourite\_tasks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ul\_id | favourite\_tag\_1 | favourite\_tag\_2 | favourite\_tag\_3 | favourite\_tag\_4 |
| 15134562 | 1 | 4 | 6 | 10 |

user\_most\_recent\_viewed

|  |  |  |
| --- | --- | --- |
| ul\_id | most\_recent\_viewed\_task\_1 | most\_recent\_viewed\_task\_2 |
| 15134562 | 1 | 4 |

|  |  |
| --- | --- |
| most\_recent\_viewed\_task\_1 | most\_recent\_viewed\_task\_1 |
| 6 | 3 |

user\_reputation

|  |  |
| --- | --- |
| ul\_id | reputation |
| 15116254 | 35 |

is\_user\_moderator

|  |  |
| --- | --- |
| ul\_id | is\_moderator |
| 15172514 | 0 |

task\_tag

|  |  |
| --- | --- |
| task\_id | tag\_id |
| 5 | 1 |

tags

|  |  |
| --- | --- |
| tag\_id | tag\_name |
| 3 | Java |

4.

List of functional dependencies:

1. unverifies\_user: ul\_id, ul\_email -> password, first\_name, last\_name, field, verification\_code.
2. verified\_user: ul\_id, ul\_email -> password, first\_name, last\_name, field.
3. banned\_users: banned\_id, banned\_email -> banned\_reason.
4. task: task\_id -> task\_title, task\_type, task\_description, task\_pages, task\_words, task\_format, created, clamied\_by, submit\_by.
5. claimed\_task: task\_id -> claimed\_by\_ul\_id.
6. completed\_task: task\_id.
7. deleted\_task: task\_id.
8. flagged\_task: task\_id.
9. user\_task: ul\_id -> task\_id.
10. user\_reputation: ul\_id -> reputation.
11. task\_tag: task\_id -> tag\_id.
12. tags: tag\_id -> tag\_name.
13. is\_user\_moderator: ul\_id -> is\_moderator.
14. user\_favourite\_tags: ul\_id -> favourite\_tag\_1, favourite\_tag\_2, favourite\_tag\_3, favourite\_tag\_4.
15. user\_most\_recent\_viewed: ul\_id -> most\_recent\_viewed\_task\_1, most\_recent\_viewed\_task\_2, most\_recent\_viewed\_task\_3, most\_recent\_viewed\_task\_4.
16. failed\_task: task\_id.

5.

Proof that each table is in 3NF:

1st NF All tables fulfill these conditions:

1. There are no duplicated rows in the table.

2. Each cell is single-valued (i.e., there are no repeating groups or arrays).

3. Entries in a column (attribute, field) are of the same kind.

2nd NF All tables fulfill these conditions:

1. All non-key attributes depend on all the key.

3rd NF All tables fulfill these condition:

1. All tables have no transitive dependencies.

6.

Justification for the usefulness of the queries and views proposed in part B within a scenario for possible use of the database within a software system.

The query where we select the first 10 the tasks with more than 5 pages. It is a good way to reward the first 10 tasks that were created due to the auto increment system we have for task id. Them having more than 5 pages makes sure they are authentic tasks.

In the more advanced query where we select all student who have written more than 2 theses’ before 2017-04-10, we can identify the more intellectual users since writing 2 theses’ is quite an accomplishment.

7.

Analysis of the speed of your queries and justification for the indexes proposed in part B.

Since the first query is only using one table, there should be no delay in retrieving the data.

However both the second and third queries contain subqueries which will slow them down. The third query will be slower than the second as it is dealing with more data.

idx\_student - will quickly find the students details on their first and last name. Since this table will contain a lot of date the index will improve query performance.

idx\_login - will retrieve the log in details for each registered user. Since ul\_id is a primary key, this makes it unique and will better the performance.

idx\_task - will retrieve both the title and the description of the task. This index will also improve query performance similar to idx\_student.

idx\_task\_type - will point to the type of each of the tasks.

8.

Justification for the necessity of the triggers and stored procedures/functions proposed in part B within a scenario for possible use of the database within a software system.

Triggers

Remove\_banned\_user – This will automatically remove a banned user form verified users, when the user is added to banned\_users.

After\_delete\_task – This will automatically move a deleted task to deleted tasks.

Procedures

TaskPagesGreaterThen – This is used to check every task that page count is greater than a given number. This is handy when wanting to minimize the number of tasks shown.

VerifyUser – This is used to move a user from unverified\_users to verified\_user when they correctly input a user\_id.