Social Computing Homework Coursebook

Instructions
Please fill in each exercise and submit the entire document as a PDF on Moodle before the section's respective deadline. Keep working on the whole document so that for the last submission you submit a completely filled in template. You may not change previous sections in subsequent submissions. Some sections require you to work on an existing software project, which you have to fork on GitHub.com, or clone and create your repository. Provide the URL of your public fork or repository of this project below.

Fill in each answer to a homework task to the textbox underneath. Use as much space as you wish. Do not provide long code snippets or other irrelevant information.

Restrictions
You may use AI tools for language styling or only. Usage of any AI tools to answer questions, inspire creative solutions or write code is strictly forbidden. Group work and sharing solutions is strictly prohibited. Any suspected cases of misconduct will be referred to the Education Dean. If you are not sure whether you are in violation of course-specific restrictions or the university's code of conduct, please ask the Lecturer or a TA.

Name

Vu Truong

Student ID

2506393

Student Email

Vu.Truong@student.oulu.fi

GitHub Repository URL

Write your answer here…

AI Use Disclaimer

Explain in detail in what parts and how AI was used for any of the work above. Fill it out and update after each homework submission, even if you did not use AI at all.

Your answers to homework tasks should not include AI-generated code or text.

I did not use AI

Task 1 (due 22.9.2025 23:59) 15 points

Exercise 1.1 Reading the dataset: Load the database and for each table, print and inspect the available columns and the number of rows. Explain below how you loaded the database. For each table, describe all columns (name, purpose, type, example of contents). You may use SQL and/or Python to perform this task. (3 points)

```python
# Explanations for the work are being added as comments
import sqlite3
import pandas as pd

# Current db file location (same location as the code file)
dbfile = 'database.sqlite'
# Establish a connection to the db
conn = sqlite3.connect(dbfile)

# Read all table names -> turn it to a dataframe
tablenames_df = pd.read_sql_query("SELECT name FROM sqlite_master WHERE
type='table';", conn)

# Convert df to a list
tables = tablenames_df['name'].tolist()

for table in tables:
    print(f"Table: {table}")
    df = pd.read_sql_query(f"SELECT * FROM {table};", conn)
    # Inspect the table
    print(f"Number of rows: {len(df)}")
    print(f"Available columns: {df.columns.tolist()}")
    # Get metadata
    col = pd.read_sql_query(f"PRAGMA table_info({table});", conn)
    for idx, row in col.iterrows():
        print(f"Name: {row['name']}")
        print(f"Type: {row['type']}")
        # Hardcoded purpose as metadata is not available in db
        print(f"Purpose: -")
        print(f"Example: {df[row['name']].head(1).values[0]}")
```

Exercise 1.2 Lurkers: How many users are there on the platform who have not interacted with posts or posted any content yet (but may have followed other users)? Answer and explain your queries/calculations below. You may use SQL and/or Python to perform this task. (3 points)

```python
# Explanations for the work are being added as comments
import sqlite3
import pandas as pd

# Current db file location
dbfile = 'database.sqlite'
# Establish a connection to the db
conn = sqlite3.connect(dbfile)

try:
    # Check for users who not exist in posts, comments, and reactions table
using subqueries
    lurkers = pd.read_sql_query("""
    SELECT
        id
    FROM users
    WHERE id NOT IN (SELECT user_id FROM posts)
    AND id NOT IN (SELECT user_id FROM comments)
    AND id NOT IN (SELECT user_id FROM reactions);
    """, conn)
    # print("Lurkers: ")
    # print(lurkers)
    print("The number of people who have not interacted at all: ",
len(lurkers))
except Exception as e:
    print(f"Error: {e}")
```

Exercise 1.3 Influencers: In the history of the platform, who are the 5 users with the most engagement on their posts? Describe how you measure engagement. Answer and explain your queries/calculations below. You may use SQL and/or Python to perform this task. (4 points)

```python
# Explanations for the work are being added as comments
import sqlite3
import pandas as pd

# Current db file location
dbfile = 'database.sqlite'
# Establish a connection to the db
conn = sqlite3.connect(dbfile)

"""
```

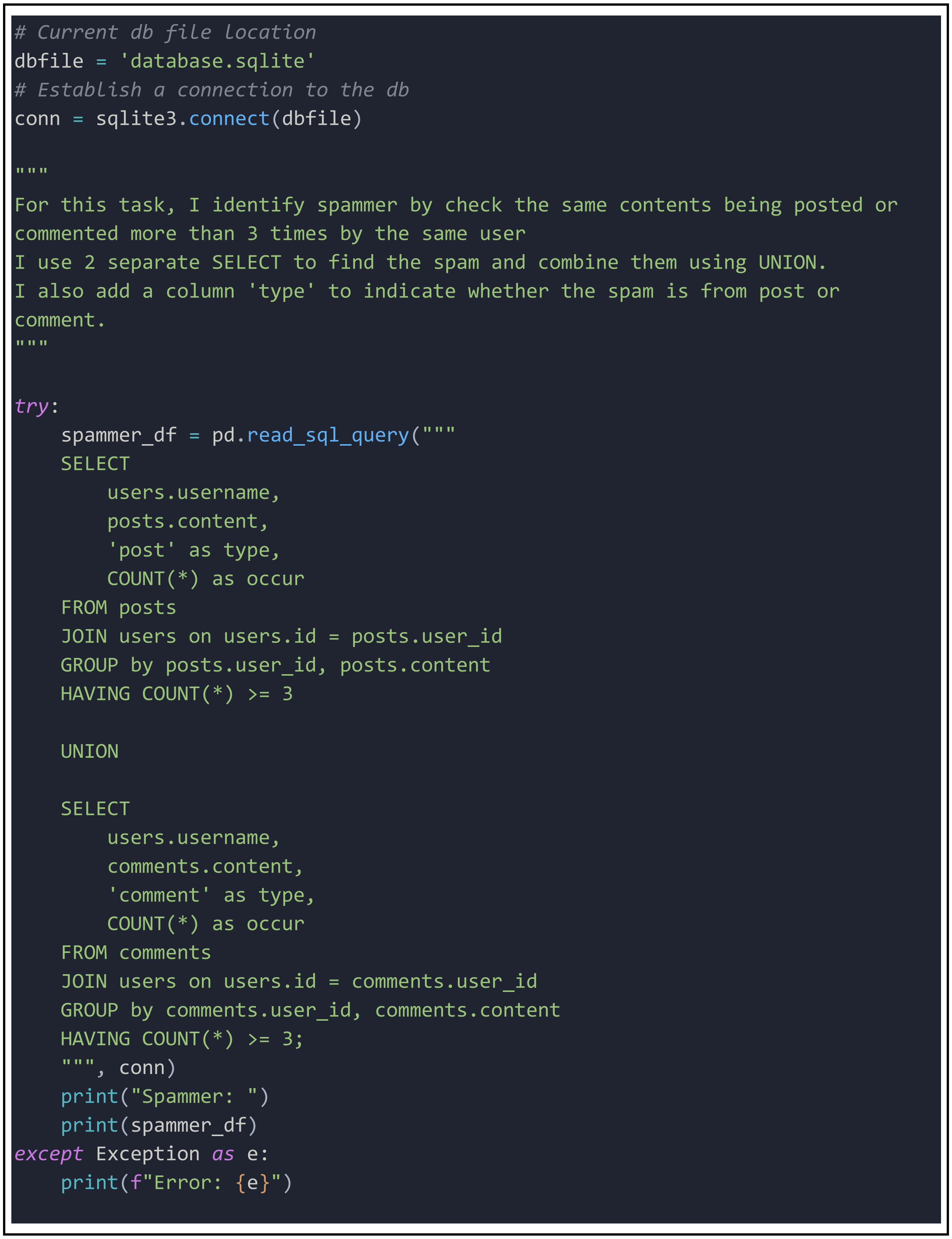
```
To find top 5 influencers, I count the number of reactions and comments on each
user's posts.
First, I JOIN the posts table with the users table to get the username (the
author).
Then, I LEFT JOIN the reactions and comments tables to count the number of
reactions and comments for each posts.
Finally, I group the results by username and order them by the total number of
reactions and comments in descending order, limiting the results to the top 5.

By using DISTINCT in the COUNT, I ensure that each reaction and is counted only
once, because when joining multiple tables, there can be duplicate rows for the
same reaction and comment, resulting in same count value for these columns.
"""

try:
    influencer_df = pd.read_sql_query("""
    SELECT
        users.id,
        users.username,
        COUNT(DISTINCT reactions.id) as Reactions,
        COUNT(DISTINCT comments.id) AS Comments
    FROM posts
    JOIN users on users.id = posts.user_id
    LEFT JOIN reactions on posts.id = reactions.post_id
    LEFT JOIN comments ON posts.id = comments.post_id
    GROUP by users.username
    ORDER BY (COUNT(DISTINCT reactions.id) + COUNT(DISTINCT comments.id)) DESC
    LIMIT 5;
    """, conn)
    print("Top 5 influencers: ")
    print(influencer_df)
except Exception as e:
    print(f"Error: {e}")
```

Exercise 1.4 Spammers: Identify users who have shared the same text in posts or comments at least 3 times over and over again (in all their history, not just the last 3 contributions). Answer and explain your queries/calculations below. You may use SQL and/or Python to perform this task. (5 points)

```
# Explanations for the work are being added as comments
import sqlite3
import pandas as pd
```



```python
# Current db file location
dbfile = 'database.sqlite'
# Establish a connection to the db
conn = sqlite3.connect(dbfile)

"""
For this task, I identify spammer by check the same contents being posted or
commented more than 3 times by the same user
I use 2 separate SELECT to find the spam and combine them using UNION.
I also add a column 'type' to indicate whether the spam is from post or
comment.
"""

try:
    spammer_df = pd.read_sql_query("""
    SELECT
        users.username,
        posts.content,
        'post' as type,
        COUNT(*) as occur
    FROM posts
    JOIN users on users.id = posts.user_id
    GROUP by posts.user_id, posts.content
    HAVING COUNT(*) >= 3

    UNION

    SELECT
        users.username,
        comments.content,
        'comment' as type,
        COUNT(*) as occur
    FROM comments
    JOIN users on users.id = comments.user_id
    GROUP by comments.user_id, comments.content
    HAVING COUNT(*) >= 3;
    """, conn)
    print("Spammer: ")
    print(spammer_df)
except Exception as e:
    print(f"Error: {e}")
```

Task 2 (due 29.9.2025 23:59) 15 points

Exercise 2.1 Growth: This year, we are renting 16 servers to run our social media platform. They are soon at 100% capacity, so we need to rent more servers. We would like to rent enough to last for 3 more years without upgrades, plus 20% capacity for redundancy. We need an estimate of how many servers we need to start renting based on past growth trends. Plot the trend on a graph using Python and include it below. Answer and explain your queries/calculations below. You may use SQL and/or Python to perform this task. (Note that the dataset may not end in the current year, please assume that the last data marks today's date) (3 points)

Write your answer here…

Exercise 2.2 Virality: Identify the 3 most viral posts in the history of the platform. Select and justify a specific metric or requirements for a post to be considered viral. Answer and explain your queries/calculations below. You may use SQL and/or Python to perform this task. (4 points)

Write your answer here…

Exercise 2.3 Content Lifecycle: What is the average time between the publishing of a post and the first engagement it receives? What is the average time between the publishing of a post and the last engagement it receives? Answer and explain your queries/calculations below. You may use SQL and/or Python to perform this task. (4 points)

Write your answer here…

Exercise 2.4 Connections: Identify the top 3 user pairs who engage with each other's content the most. Define and describe your metric for engagement. Answer and explain your queries/calculations below. You may use SQL and/or Python to perform this task. (4 points)

Write your answer here…

Task 3 (due 19.10.2025 23:59) 15 points

Exercise 3.1 Censorship: implement the `moderate_content` function that automatically detects and censors inappropriate user posts on the platform. Your function should take a post, comment or user introduction as input and apply censorship rules to either clean or remove content, and supply a risk score that corresponds to the number and weight of violations in the content (note the risk classification thresholds in the code). The exact rules are detailed on the Rules page. Think of and implement one more moderation measure you think is important to keep the platform safe. Include and explain your implementation below. (5 points)

Write your answer here…

Exercise 3.2 User risk analysis: Assign risk scores to each user by implementing the `user_risk_analysis` function. This function returns a risk score for a given user based on rules presented on the Rules page. Identify the top 5 highest risk users. Think of and implement one more risk prediction measure you think is important to keep the platform safe. Answer and explain your queries/calculations below. (5 points)

Write your answer here…

Exercise 3.3 Recommendation Algorithm: Implement the `recommend` function. Identify a suitable, simple recommendation algorithm that will recommend 5 relevant posts on the "Recommended" tab based on the posts the user reacted to positively and the users they followed. (5 points)

Write your answer here…

Task 4 (due 27.10.2025 23:59) 20 points

Exercise 4.1 Topics: Identify the 10 most popular topics discussed on our platform. Use Latent Dirichlet Allocation (LDA) with the `gensim` library. Answer and explain your queries/calculations below. You may use SQL and/or Python to perform this task. 5 points)

Write your answer here…

Exercise 4.2 Sentiment: Perform sentiment analysis on posts and comments. What is the overall tone of the platform? How does sentiment vary across user posts discussing different topics identified in Exercise 3? Please use VADER (`nltk.sentiment`) for this analysis. Answer and explain your queries/calculations below. You may use SQL and/or Python to perform this task. (5 points)

Write your answer here…

Exercise 4.3 Learning from others' mistakes: Find two social platforms similar to Mini Social that have been under fire for an engineering, design or operation error that severely affected a large group of users. Describe how we can learn from their mistakes and draft up a plan about how Mini Social can be improved learning from their mistakes. You do not need to write code in this exercise unless your plan includes a specific change to an algorithm or function. (5 points)

Write your answer here…

Exercise 4.4 Design and implement a new social feature in Mini Social. For example, a user reputation scoring system, a reporting system, a feature to find related content to a post, new post modalities such as polls or reposts. Your change must include a UI improvement or addition. Do not implement non-social, technical features, such as resource optimization, security improvements or style changes. Document the design and implementation process of your addition here. You must also demonstrate a fully functional feature in a maximum 2-minute video recording uploaded to Moodle. (5 points)

Write your answer here…