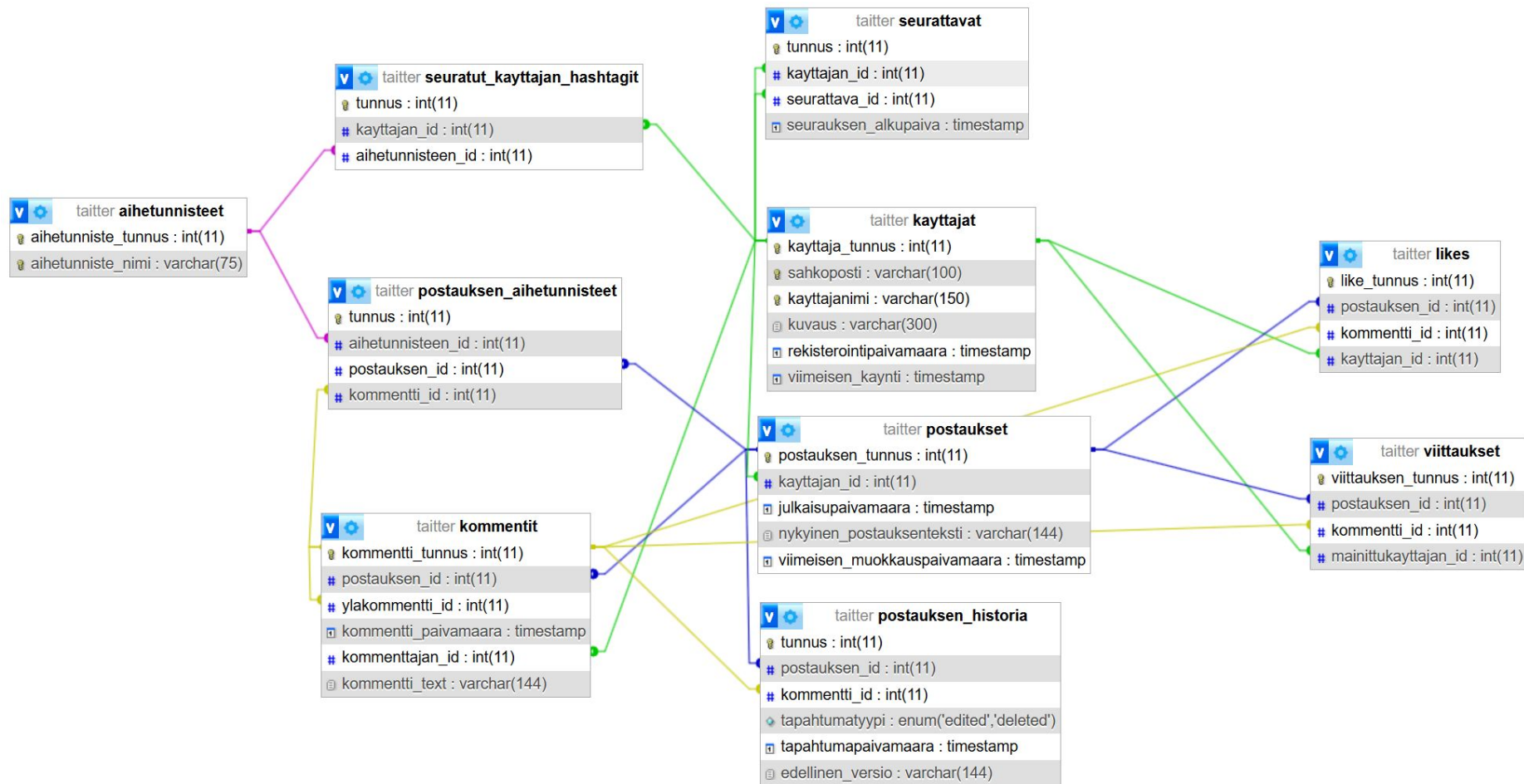


MySQL Queries

TAltter

Maaria, Mariia, Markus, Riku, Yuliia

taitter



1) Find how many users a specific user is following

kayttajan_tunnus = 517;

```
SELECT kayttajat.kayttajanimi, COUNT(seurattavat.seurattava_id) AS followings
FROM kayttajat
LEFT JOIN seurattavat ON seurattavat.kayttajan_id = kayttajat.kayttaja_tunnus
WHERE kayttajat.kayttaja_tunnus = 517;
```

2) Find how many followers a specific user has

kayttajan_tunnus = 517;

```
SELECT kayttajat.kayttajanimi, COUNT(seurattavat.kayttajan_id) AS followers
FROM kayttajat
LEFT JOIN seurattavat ON seurattavat.seurattava_id = kayttajat.kayttaja_tunnus
WHERE kayttajat.kayttaja_tunnus = 517;
```

3) For each user, find how many users are following them

```
SELECT kayttajat.kayttajanimi, COUNT(seurattavat.seurattava_id) AS  
followings  
FROM kayttajat  
LEFT JOIN seurattavat ON seurattavat.kayttajan_id =  
kayttajat.kayttaja_tunnus  
GROUP BY kayttajat.kayttajanimi;
```

4) For each user, find how many followers they have

```
SELECT kayttajat.kayttajanimi, COUNT(seurattavat.kayttajan_id) AS followers
FROM kayttajat
LEFT JOIN seurattavat ON seurattavat.seurattava_id = kayttajat.kayttaja_tunnus
GROUP BY kayttajat.kayttajanimi;
```

5) Find how many instances of mutual followings exist (where UserA follows UserB and UserB follows UserA).

1

```
SELECT COUNT(*) AS mutual_follows
FROM seurattavat AS seurattavatA, seurattavat AS seurattavatB
WHERE seurattavatA.kayttajan_id = seurattavatB.seurattava_id
AND seurattavatA.seurattava_id = seurattavatB.kayttajan_id
AND seurattavatA.kayttajan_id < seurattavatA.seurattava_id;
```

Warning ⚠: We actually need the statement

```
AND seurattavatA.kayttajan_id < seurattavatA.seurattava_id
```

because the query will return pairs of [kayttaja_id, seurattava_id] and [seurattava_id, kayttaja_id]

5) Find how many instances of mutual followings exist (where UserA follows UserB and UserB follows UserA).
Continuation

2

To make it more easily understandable we will display all the fields also from the same query changing only the select line of code.

```
SELECT seurattavatA.*  
FROM seurattavat AS seurattavatA, seurattavat AS seurattavatB  
WHERE seurattavatA.kayttajan_id = seurattavatB.seurattava_id  
AND seurattavatA.seurattava_id = seurattavatB.kayttajan_id  
AND seurattavatA.kayttajan_id < seurattavatA.seurattava_id;
```

The result is:

tunnus	kayttajan_id	seurattava_id	seurauksen_alkupaiva
1	112	1009	2026-01-07 16:55:06
309	1	50	2025-12-12 21:44:38

5) Find how many instances of mutual followings exist (where UserA follows UserB and UserB follows UserA).
Continuation

Without the last check:

```
SELECT seurattavatA.*  
FROM seurattavat AS seurattavatA, seurattavat AS seurattavatB  
WHERE seurattavatA.kayttajan_id = seurattavatB.seurattava_id  
AND seurattavatA.seurattava_id = seurattavatB.kayttajan_id;
```

The result is:

tunnus	kayttajan_id	seurattava_id	seurauksen_alkupaiva
1	112	1009	2026-01-07 16:55:06
309	1	50	2025-12-12 21:44:38
310	50	1	2025-12-12 21:44:38
311	1009	112	2025-12-12 21:46:54

But we only want to count the mutual follows. That's why we need the last check.

6) Find how many posts a specific user has

kayttajan_id = 10;

```
SELECT COUNT(*) AS total_posts  
FROM postaukset  
WHERE postaukset.kayttajan_id = 10;
```

7) Found how many posts each user has (fields to be displayed: user's id, user's name, total posts)

```
SELECT
    kayttajat.kayttaja_tunnus,
    kayttajat.kayttajanimi,
    COUNT(postaukset.kayttajan_id) AS total_posts
FROM kayttajat
LEFT JOIN postaukset ON postaukset.kayttajan_id = kayttajat.kayttaja_tunnus
GROUP BY kayttajat.kayttaja_tunnus, kayttajat.kayttajanimi;
```

8) Find who has the maximum number of posts (this is an extension of query 7).

As you can see, the query became more complex. We are using nested selects.

```
SELECT kayttajat.kayttaja_tunnus, kayttajat.kayttajanimi, COUNT(postaukset.kayttajan_id) AS total_posts
FROM kayttajat
LEFT JOIN postaukset ON postaukset.kayttajan_id = kayttajat.kayttaja_tunnus
GROUP BY kayttajat.kayttaja_tunnus, kayttajat.kayttajanimi
HAVING total_posts = (SELECT MAX(total_posts_per_user.total_posts2)
                      FROM (
                        SELECT COUNT(postaukset.kayttajan_id) AS total_posts2
                        FROM kayttajat
                        LEFT JOIN postaukset ON postaukset.kayttajan_id = kayttajat.kayttaja_tunnus
                        GROUP BY kayttajat.kayttaja_tunnus
                        ) AS total_posts_per_user
                      );
```

9) For a specific user, display their posts

user_id = 4

```
SELECT postaukset.postauksen_tunnus ,  
postaukset.nykyinen_postauksenteksti  
FROM postaukset  
WHERE postaukset.kayttajan_id = 4;
```

10) For a specific user and one of their posts, find how many comments that post has.

user_id = 4, post = 10

```
SELECT COUNT(kommentit.kommentti_tunnus)
FROM kommentit
INNER JOIN postaukset ON postaukset.postauksen_tunnus = kommentit.postauksen_id
WHERE postaukset.kayttajan_id = 4
AND postaukset.postauksen_tunnus = 10;
```

11) For a specific user and one of their posts, display its comments.

user_id = 4, post = 10

```
SELECT kommentit.kommentti_tunnus, kommentit.kommentti_text
FROM kommentit
INNER JOIN postaukset ON postaukset.postauksen_tunnus =
kommentit.postauksen_id
WHERE postaukset.kayttajan_id = 4
AND postaukset.postauksen_tunnus = 10;
```

13) Display posts where they have more than 1 tags

```
SELECT postaukset.postauksen_tunnus, postaukset.nykyinen_postauksenteksti
FROM postaukset
INNER JOIN postauksen_aihetunnisteet ON postauksen_aihetunnisteet.postauksen_id
= postaukset.postauksen_tunnus
GROUP BY postaukset.postauksen_tunnus, postaukset.nykyinen_postauksenteksti
HAVING COUNT(postauksen_aihetunnisteet.aihetunnisteen_id) > 1;
```


13) For a specific post, display its tags

post_id = 44

```
SELECT aihetunnisteet.aihetunniste_tunnus, aihetunnisteet.aihetunniste_nimi
FROM postauksen_aihetunnisteet
INNER JOIN postaukset ON postaukset.postauksen_tunnus =
postauksen_aihetunnisteet.postauksen_id
INNER JOIN aihetunnisteet ON aihetunnisteet.aihetunniste_tunnus =
postauksen_aihetunnisteet.aihetunniste_id
WHERE postaukset.postauksen_tunnus = 44;
```

14) For a specific post, display which comment has more than 1 tags

post_id = 44

```
SELECT kommentit.kommentti_tunnus, kommentit.kommentti_text
FROM kommentit
INNER JOIN postauksen_aihetunnisteet ON
postauksen_aihetunnisteet.kommentti_id = kommentit.kommentti_tunnus
WHERE kommentit.postauksen_id = 44
GROUP BY kommentit.kommentti_tunnus, kommentit.kommentti_text
HAVING COUNT(postauksen_aihetunnisteet.aihetunnisteen_id) > 1;
```

15) For a specific comment from a specific post, display its tags

post_id = 44, comment_id = 312

```
SELECT aihetunnisteet.aihetunniste_tunnus, aihetunnisteet.aihetunniste_nimi
FROM postauksen_aihetunnisteet
INNER JOIN aihetunnisteet ON postauksen_aihetunnisteet.aihetunniste_id =
aihetunnisteet.aihetunniste_tunnus
INNER JOIN kommentit ON postauksen_aihetunnisteet.komentti_id =
kommentit.komentti_tunnus
WHERE kommentit.postauksen_id = 44 AND kommentit.komentti_tunnus = 312;
```

16) Display how many likes a specific post has

post_id = 44

```
SELECT COUNT(likes.postauksen_id) AS likes
FROM likes
WHERE likes.postauksen_id = 44;
```

17) For a specific post, display who pressed like

post_id = 44

```
SELECT kayttajat.kayttaja_tunnus, kayttajat.kayttajanimi
FROM likes
INNER JOIN kayttajat ON likes.kayttajan_id = kayttajat.kayttaja_tunnus
WHERE likes.postauksen_id = 44;
```

18) For a specific post, count its mentions

post_id = 44

```
SELECT COUNT(viittaukset.postauksen_id) AS mentions
FROM viittaukset
WHERE viittaukset.postauksen_id = 44;
```

19) For a specific post, display who is mentioned

post_id = 44

```
SELECT kayttajat.kayttaja_tunnus, kayttajat.kayttajanimi
FROM viittaukset
INNER JOIN kayttajat ON kayttajat.kayttaja_tunnus =
viittaukset.mainittukayttajan_id
WHERE viittaukset.postauksen_id = 44;
```

20) Display how many likes a specific comment from a specific post has

post_id = 44, comment_id = 169

```
SELECT COUNT(likes.kommentti_id) as likes
FROM likes
INNER JOIN kommentit ON likes.kommentti_id = kommentit.kommentti_tunnus
WHERE kommentit.postauksen_id = 44 and likes.kommentti_id = 169;
```


21) Display who pressed like in specific comment in a specific post

post_id = 44, comment_id = 169

```
SELECT kayttajat.kayttaja_tunnus, kayttajat.kayttajanimi
FROM likes
INNER JOIN kommentit ON likes.kommentti_id = kommentit.kommentti_tunnus
INNER JOIN kayttajat ON likes.kayttajan_id = kayttajat.kayttaja_tunnus
WHERE kommentit.postauksen_id = 44 and likes.kommentti_id = 169;
```

22) For a specific comment from a specific post, display who is mentioned

post_id = 44, comment_id = 169

```
SELECT kayttajat.kayttaja_tunnus, kayttajat.kayttajanimi
FROM viittaukset
INNER JOIN kayttajat ON kayttajat.kayttaja_tunnus =
viittaukset.mainittukayttajan_id
INNER JOIN kommentit ON kommentit.kommentti_tunnus = viittaukset.kommentti_id
WHERE kommentit.postauksen_id = 44 AND viittaukset.kommentti_id = 169;
```

23) Display which tags a specific user is following

user_id = 1011

```
SELECT aihetunnisteet.aihetunniste_tunnus, aihetunnisteet.aihetunniste_nimi
FROM seuratus_kayttajan_hashtagit
INNER JOIN aihetunnisteet ON seuratus_kayttajan_hashtagit.aihetunniste_id =
aihetunnisteet.aihetunniste_tunnus
INNER JOIN kayttajat ON seuratus_kayttajan_hashtagit.kayttajan_id =
kayttajat.kayttaja_tunnus
WHERE kayttajat.kayttaja_tunnus = 1011;
```

24) Display how many tags a specific user is following

user_id = 1011

```
SELECT COUNT(seuratut_kayttajan_hashtagit.aihetunnisteen_id) AS total_tags
FROM seuratut_kayttajan_hashtagit
WHERE seuratut_kayttajan_hashtagit.kayttajan_id = 1011;
```

Conclusion:

This database supports both simple and complex queries to retrieve the necessary information. All required tables have been properly created to ensure efficient data management and querying.
