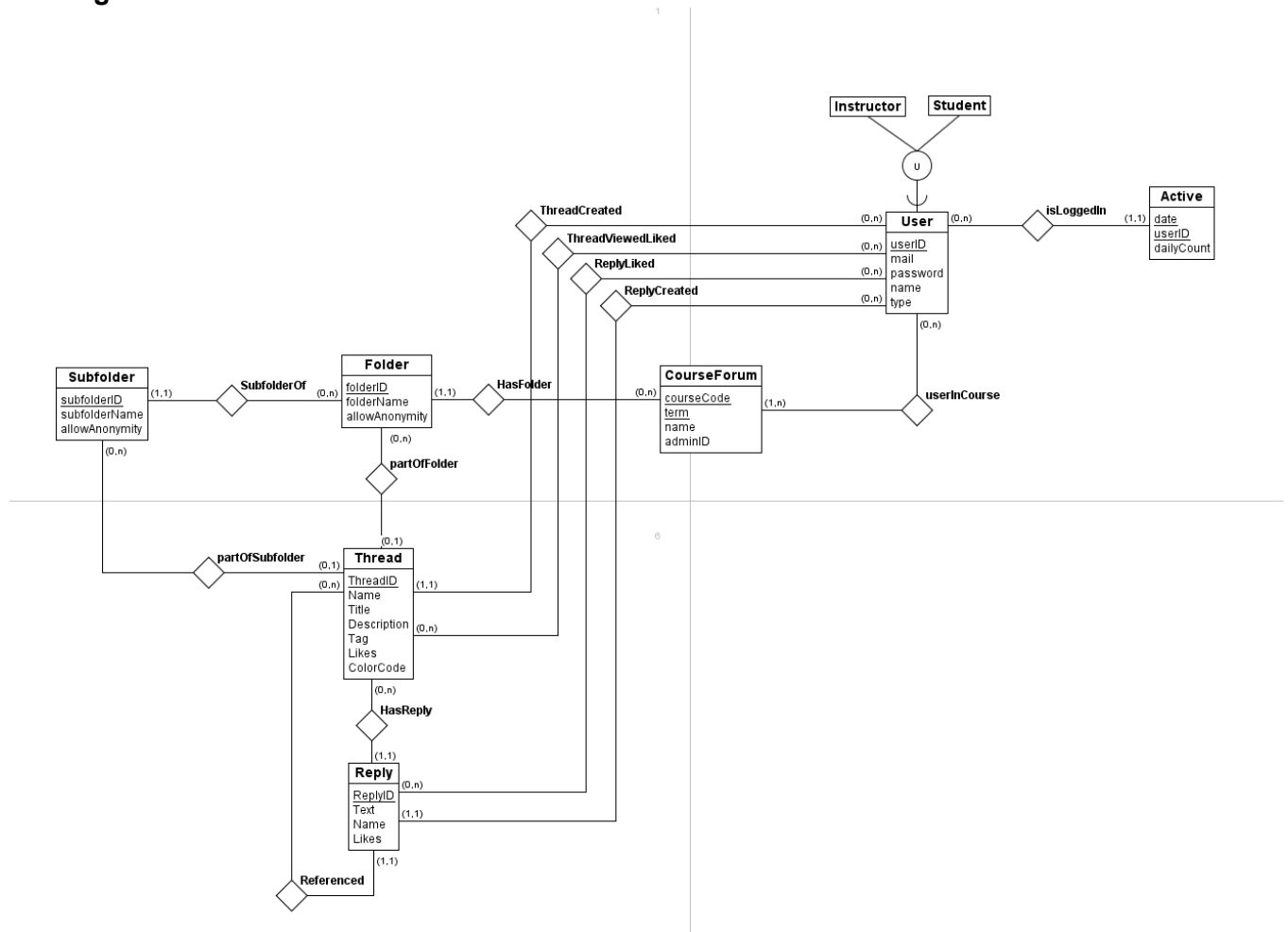


TDT4145 Datamodellering og databasesystemer
Prosjekt levering 1
Erik Galler, Johannes Finne Reinseth, Nils-Martin Hillesund

a)
ER-diagram



Antagelser:

Type-attributtet i User kan enten være Instructor eller Student, hvor Instructor involverer professor, vitass, undass, studass, osv. Vi har valgt å lage en kategorisering hvor User kun kan være enten Instructor eller Student.

Active-entiteten og relasjonen mellom User og Active sørger for å holde oversikt over hvor mange ganger en User har logget inn, noe som vil hjelpe oss med å finne statistikk på brukeraktivitet.

Instruktøren som lager et CourseForum vil bli admin og muligheter til å kontrollere og håndtere forumet. UserID til instruktøren som lager forumet vil bli lagret som en attributt i

CourseForum. Andre instruktører kan bli invitert til forumet, men har ikke samme tilgang som admin. Studenter kan også bli invitert til CourseForum og har heller ikke samme tilgang som admin eller instruktør.

CourseForum kan inneholde flere foldere (Folder) og hver folder kan inneholder flere subfoldere (Subfolder). Vi antar at subfoldere kan ikke ha subfoldere under seg selv. En folder eller en subfolder kan bli opprettet, redigert eller slettet kun av en Instructor.

Både en folder og en subfolder kan inneholde flere threads (Thread), men en thread er enten en del av en subfolder eller en folder. Svar (Reply) på threads må være koblet til en spesifikk thread. Et svar kan ha en referanse (via Referenced) til en annen thread. FolderID, subfolderID, ThreadID og ReplyID er tilfeldig genererte sekvenser som er unike for alle folders/subfolder og threads/replies i alle kurs.

User kan lage en thread, og det vil også bli registrert om en User ser på en thread og liker en thread. På samme måte kan en User lage et svar på en thread, samt like et svar. Et svar kan ikke ha svar under seg igjen, på samme måte som en subfolder ikke kan ha en subfolder under seg. Vi antar at en thread kun kan få svar, hvor ytterligere kommentarer og oppfølgingsspørsmål tilsvare svar. Vi skiller dermed ikke på svar, kommentarer og oppfølgingsspørsmål.

b)

Relasjonstabeller

user(userID, mail, password, name, type)

- Type vil enten være Instruktur eller Student

userID -> mail

userID -> password

userID -> name

userID -> type

1NF: Alle attributter er atomiske. ✓

2NF: Kandidatnøkkelen er userID, og alle resterende attributter (ikke-nøkkel-attributter) er direkte avhengig av den. ✓

3NF/BCNF: userID er en primærnøkkel ✓

4NF: Tabellen har ingen MVDer ✓

active(date, userID, dailyCount)

- userID er fremmednøkkel mot user-tabellen, kan ikke være NULL

date, userID -> dailyCount

1NF: Alle attributter er atomiske. ✓

2NF: Kandidatnøkkelen er date og userID, og alle resterende attributter (ikke-nøkkel-attributter) er direkte avhengig av dem. ✓

3NF/BCNF: date, userID er en primærnøkkel ✓

4NF: Tabellen har ingen MVDer ✓

userInCourse(userID, courseCode, term, permissionType)

- userID er fremmednøkkel mot user-tabellen, kan ikke være NULL.
- courseCode, term er fremmednøkkel mot courseForum-tabellen, kan ikke være NULL.

userID, courseCode, term -> permissionType

1NF: Alle attributter er atomiske. ✓

2NF: Kandidatnøkkelen er userID, courseCode og term, og alle resterende attributter (ikke-nøkkel-attributter) er direkte avhengig av dem. ✓

3NF/BCNF: userID, courseCode, term er en primærnøkkel ✓

4NF: Tabellen har ingen MVDer ✓

courseForum(courseCode, term, courseName, adminID)

courseCode, term -> adminID

courseCode, term -> courseName

1NF: Alle attributter er atomiske. ✓

2NF: Kandidatnøkkelen er courseCode og term, og alle resterende attributter (ikke-nøkkel-attributter) er direkte avhengig av dem. ✓

3NF/BCNF: courseCode, term er en primærnøkkel ✓

4NF: Tabellen har ingen MVDer ✓

folder(folderID, allowAnonymity, folderName courseCode, term)

- courseCode, term er fremmednøkkel mot courseForum-tabellen, kan ikke være NULL.

folderID-> allowAnonymity

folderID-> folderName

folderID-> courseCode, term

1NF: Alle attributter er atomiske. ✓

2NF: Kandidatnøkkelen er folderID, og alle resterende attributter (ikke-nøkkel-attributter) er direkte avhengig av dem. ✓

3NF/BCNF: folderID er en primærnøkkel ✓

4NF: Tabellen har ingen MVDer ✓

subfolder(subfolderID, allowAnonymity, subfolderName, folderID)

- folderID er fremmednøkkel til folder-tabellen, kan ikke være NULL.

subfolderID -> allowAnonymity

subfolderID -> subfolderName

subfolderID -> folderID

1NF: Alle attributter er atomiske. ✓

2NF: Kandidatnøkkelen er subfolderID og alle resterende attributter (ikke-nøkkel-attributter) er direkte avhengig av dem. ✓

3NF/BCNF: subfolderID er en primærnøkkel ✓

4NF: Tabellen har ingen MVDer ✓

thread(threadID, name, title, description, tag, likes, colorCode, folderID, subfolderID, createdBy)

- folderID er fremmednøkkel mot folder-tabellen.
- subfolderID er fremmednøkkel mot subfolder-tabellen.
- createdBy er fremmednøkkel mot user-tabellen, kan ikke være NULL.

threadID -> name

threadID -> title

threadID -> description

threadID -> tag

threadID -> likes

threadID -> colorCode

threadID -> folderID

threadID -> subfolderID

threadID -> createdBy

1NF: Alle attributter er atomiske. ✓

2NF: Kandidatnøkkelen er threadID og alle resterende attributter (ikke-nøkkel-attributter) er direkte avhengig av dem. ✓

3NF/BCNF: threadID er en primærnøkkel ✓

4NF: Tabellen har ingen MVDer ✓

threadViewedLiked(userID, threadID, viewed, liked)

- userID er fremmednøkkel mot user-tabellen, kan ikke være NULL.
- threadID er fremmednøkkel mot thread-tabellen, kan ikke være NULL.

userID, threadID -> viewed

userID, threadID -> liked

1NF: Alle attributter er atomiske. ✓

2NF: Kandidatnøkkelen er userID og threadID, og alle resterende attributter (ikke-nøkkel-attributter) er direkte avhengig av dem. ✓

3NF/BCNF: userID, threadID er en primærnøkkel ✓

4NF: Tabellen har ingen MVDer ✓

reply(replyID, text, name, likes, threadID, createdBy, referencedThreadID)

- threadID er fremmednøkkel mot thread-tabellen, kan ikke være NULL.
- createdBy er fremmednøkkel mot user-tabellen, kan ikke være NULL.
- referencedThreadID er fremmednøkkel mot thread-tabellen, kan ikke være NULL.

Refererer til en thread med svaret på den originale threaden.

replyID -> text

replyID -> name

replyID -> likes

replyID -> threadID

replyID -> createdBy

replyID -> referencedThreadID

1NF: Alle attributter er atomiske. ✓

2NF: Kandidatnøkkelen er replyID og alle resterende attributter (ikke-nøkkel-attributter) er direkte avhengig av dem. ✓

3NF/BCNF: replyID er en primærnøkkel ✓

4NF: Tabellen har ingen MVDer ✓

replyLiked(userID, replyID, liked)

- userID er fremmednøkkel mot user-tabellen, kan ikke være NULL.
- replyID er fremmednøkkel mot reply-tabellen, kan ikke være NULL.

userID, replyID -> liked

1NF: Alle attributter er atomiske. ✓

2NF: Kandidatnøkkelen er userID og replyID og alle resterende attributter (ikke-nøkkel-attributter) er direkte avhengig av dem. ✓

3NF/BCNF: userID, replyID er en primærnøkkel ✓

4NF: Tabellen har ingen MVDer ✓

c)

1. A student logs into the system, i.e., check user name and password (you do not need to encrypt/decrypt passwords). This should have e-mail and password as input, and these should match this info in the database.

Når en bruker logger inn, skal input-en til brukeren matche unikt med en tilsvarende kombinasjon av email og password i user relasjons tabellen. Blir det ingen match, blir ikke brukeren logget inn, og får heller ikke tilgang på sin userID.

2. A student makes a post belonging to the folder “Exam” and tagged with “Question”. Input to the use case should be a post and the texts “Exam” and “Question”.

Etter at brukeren har fått logg inn, vil studenten få en oversikt over alle kursene han/hun har tilgang til. Brukeren navigerer seg til kurset og får en ny oversikt over alle mapper tilgjengelig og velger deretter “Exam” mappen. Der har man muligheten til å trykke seg inn på subfolders, opprettede posts eller lage en ny post. Studenten velger å opprette en ny post, og velger “Question” som tag til posten.

3. An instructor replies to a post belonging to the folder “Exam”. The input to this is the id of the post replied to. This could be the post created in use case 2.

En instruktør har mulighet til å se alle poster i folderen “Exam”. Innenfor “Exam” vil alle titlene til hver post dukke opp, som da er identifisert med threadID. Instruktøren velger en post og får muligheten til å like eller svar på posten, han velger å svare.

4. A student searches for posts with a specific keyword “WAL”. The return value of this should be a list of ids of posts matching the keyword.

Alle threads har en tittel og beskrivelse. En student kan dermed søke etter ord i postene og vil få en respons med alle poster med match.

5. An instructor views statistics for users and how many post they have read and how many they have created. These should be sorted on highest read posting numbers. The output is “user name, number of posts read, number of posts created”. You don’t need to order by posts created, but the number should be displayed. The result should also include users which have not read or created posts.

Hver gang en bruker logger seg inn vil det registreres antall ganger personen har logget inn den dagen. En instruktør vil da kunne se hvilke brukere som har logget seg inn flest ganger, siden hans ID gir han tilgang til informasjonen. Det vil registreres når en bruker oppretter en post, ser en post, liker en post, svarer på en post og liker et svar i en post. En instruktør kan dermed få oversikt over hvilke brukere som er mest aktive i kurset. Siden vi har informasjon om alle brukere i databasen vil vi også kunne vise alle brukere som ikke har laget eller likt en post.