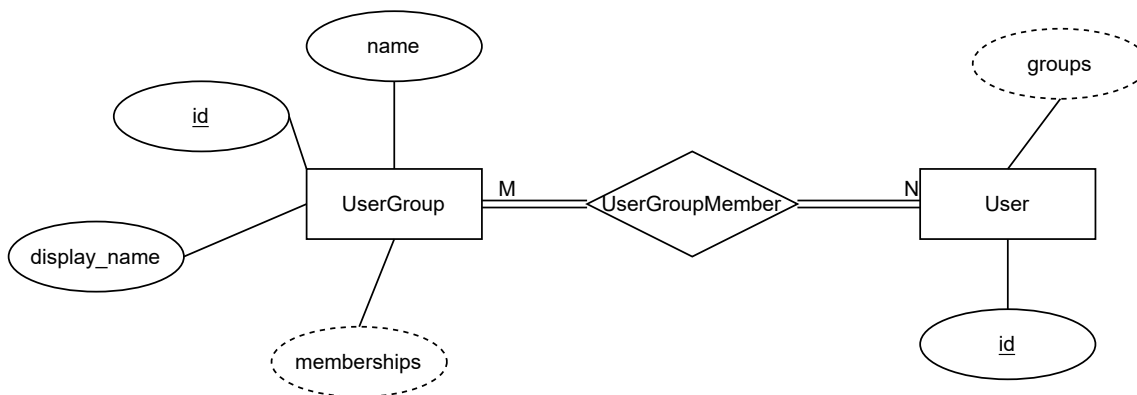


Tietokannan suunnittelu viestilistoille

Muistutus ER-kaavioiden toiminnasta: ITKA204-kurssi

Kaaviotesti

Osa `UserGroup` ja `User` taulusta ja niiden suhteesta esitettynä ER-kaaviona



Ryhmän ajatuksia

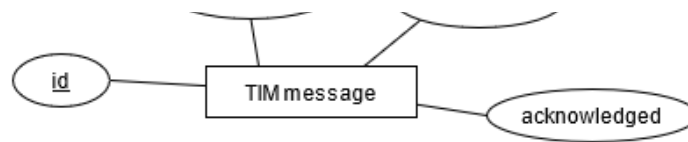
Vastaanottaja voi olla myös ryhmä.
TIM-käyttäjillä on oma 1 hlö ryhmä,
joten kaikki viestit voisi kohdentaa jollekin
ryhmälle eikä käyttäjälle.

ts. `group_id` on vastaanottajan id

tarvitaanko lähettäjä?
lähetysaikaa?

`group_id`

`contents`



TIMMESSAGE(id, group_id, contents, acknowledged)

USERGROUP(id, name, display_name, created, modified)

Viesti-taulu:

- id
- teksti
- lähettäjän id
- "millä sivuilla näkyy"
- "kenellä näkyy" (tim-ryhmän id?)
- kuitattu

Viestilista:

- Viestilistan nimi
- id
- spostilistan id?
-

Viestilistan jäsenet

- listan id
- jäsenen/usergroupin id

Kysymyksiä

Viestitaulun millä sivuilla näkyy, onko yhdessä taulussa näkyvyys-sarakkeessa lista sivujen id:stä, jokaisesta näkyvästä sivusta oma rivi tauluun vai erillinen näkyvyystaulu?

- Moniarvoisen attribuutin transformointi ITKA204

Jos viestilistan asetuksista tehdään taulu, onko yksi "iso" taulu parempi vai monta pienempää taulua

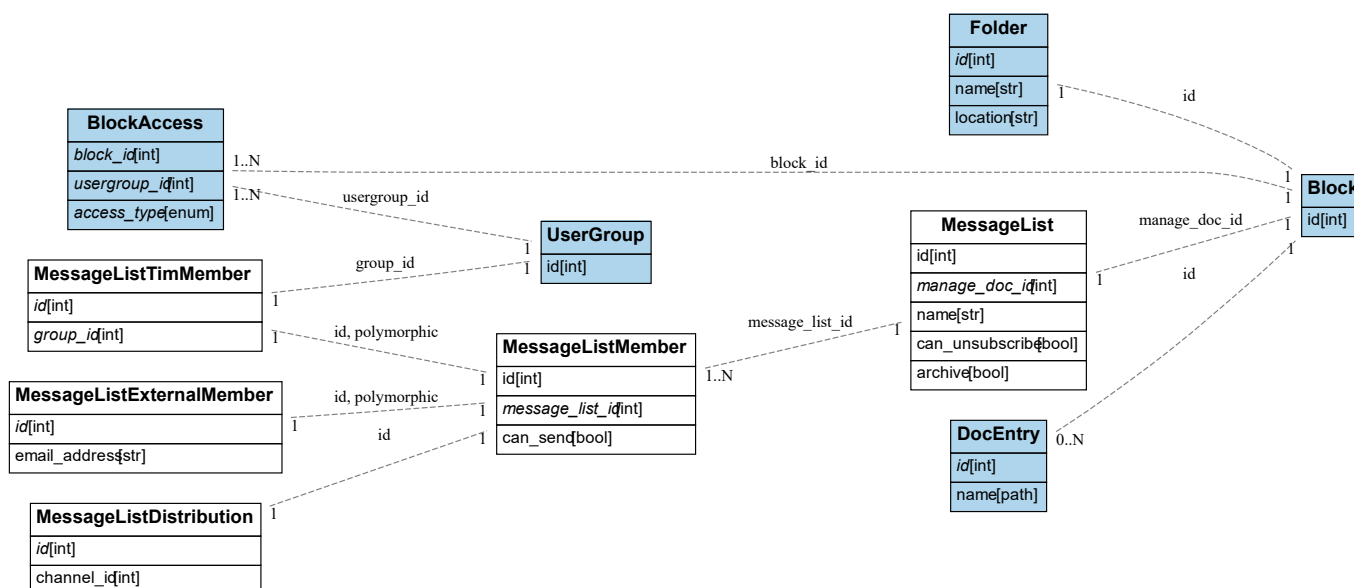
- Ylläpidon/kehityksen kannalta?
- TIMin suorituskyvyn kannalta?

Taulusuunnitelma

Selitystä

- Sinisellä väritetyt taulut ovat TIMissä jo olevat taulut
- Valkoisella väritetyt ovat uudet lisättävät taulut
- Alle viivatut attribuutit ovat osaa pääavainta (TODO: TIM näköjään sanitoi tarvittavan attribuutin pois toistaiseksi)
- *kursiivilla* merkityt attribuutit ovat ulkoiset avaimet
- Viivat kuvaavat suhdetta taulujen välillä. Viivojen kohdalla lukee, mikä ulkoinen avain muodostaa suhteen
- Viivojen päässä olevat merkinnät kuvaavat kardinaliteettisuhdetta

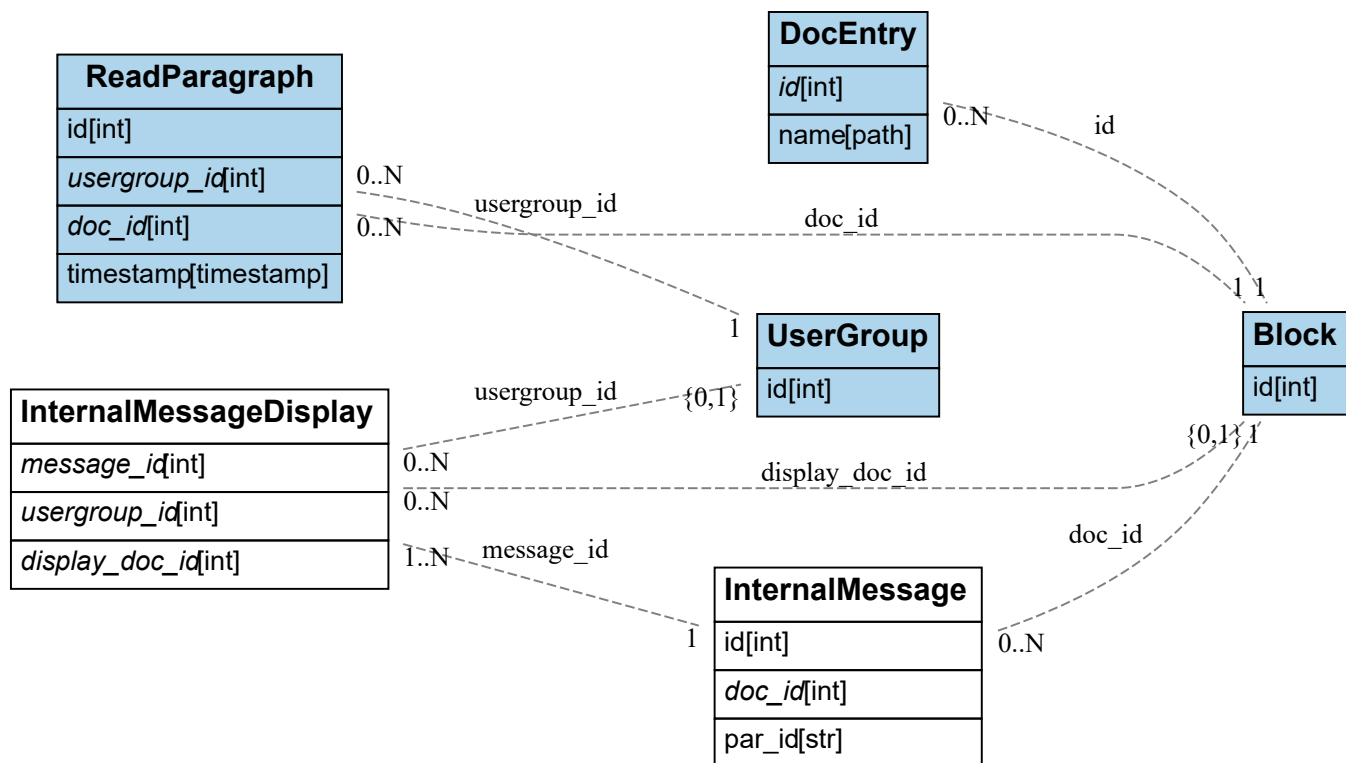
Viestiryhmät



Huomioita:

- Viestilistaan kuuluu myös kansio (taulussa `Folder`), johon laitetaan listan viestit dokumentteina
- `DocEntry` sisältää dokumenttiin liittyvää tietoa. `DocEntry:n id` vastaa `Block:n id:tä`.

Sisäinen viestijärjestelmä



Huomioita

- Nimityksinä käytetään `InternalMessage`, sillä `Message`-taulu on jo olemassa TIMissa
- Sisäinen viestijärjestelmä on myös eräs viestikanava
- Sisäinen viesti määräytyy dokumentin ja sen sisällä olevan tekstilohkon perusteella.
- Ketju muodostuu, kun dokumenttiin laitetaan uusia lohkoja
 - Tämä pitää vielä miettiä, ehkä parempi olisi lisätä tietokantaan suoraan viite edelliseen viestiin
- Sisäiseen viestiin liittyy kansio(t), mutta niihin pääsee käsiksi polun kautta

Alkuperäinen taulusuunnitelma

Syntaksi:

```

Taulunnimi
-----
sarakkeen_nimi: tyyppi [-> MihinTauluunViittaa]
    
```

Taulut

```
MessageList
```

```
-----
```

```
id: int
block_id: int -> Block(id)
name: str
can_unsubscribe: bool
archive: bool
```

```
MessageListMember
```

```
-----
```

```
id: int
message_list_id: int -> MessageList(id)
can_send: bool
```

```
MessageListTimMember
```

```
-----
```

```
id: int -> MessageListMember(id)
group_id: int -> UserGroup(id)
```

```
MessageListExternalMember
```

```
-----
```

```
id: int -> MessageListMember(id)
email_address: str
```

```
MessageListDistribution
```

```
-----
```

```
id: int -> MessageListMember(id)
channel_id: int/enum # email, TIM, Discord, Slack, ...
```

```
Useraccount-taulussa prefs:
```

```
{..., "default_message_channels": [1, 2], ...} # käyttäjäkohtaiset oletusviestikanavat
```

```
Message
```

```
-----
```

```
id: int
content: str
sender_id: int -> UserGroup(id)
```

```
MessageDisplay
```

```
-----
```

```
message_id: int -> Message(id)
usergroup_id: int -> UserGroup(id) # jos null, näytetään kaikille
block_id: int -> Block(id) # missä dokumentissa/kansiossa viesti näkyy (jos null, niin k
aikkialla)
```

```
MessageAcknowledgement
-----
message_id: int -> Message(id)
user_id: int -> User(id)
acked_on: timestamp # onko viesti kuitattu
```

Taulujen MessageListMember, MessageListTimMember, ja MessageListExternalMember osalta ks.

- <https://docs.sqlalchemy.org/en/14/orm/inheritance.html#joined-table-inheritance>
- https://docs.sqlalchemy.org/en/14/orm/inheritance_loading.html#loading-joined-inheritance