

Relazione progetto basi di dati

Michele Ceccacci, Luca Patrignani

June 26, 2023

Contents

1	Analisi dei requisiti	3
1.1	Intervista	3
1.2	Definizioni	4
1.2.1	Operazioni utente	4
2	Progettazione Concettuale	6
2.1	Utenti	6
2.2	Chat e membri della chat	6
2.2.1	Uscita dalla chat	6
2.3	Regione	7
2.4	Password	7
2.5	Contenuto (post e commenti)	7
2.6	Schema concettuale finale	9
3	Progettazione logica	10
3.1	Stima del volume dei dati	10
3.2	Tabelle degli accessi	11
3.2.1	Creare un nuovo utente	11
3.2.2	Cambiare password	11
3.2.3	Autenticazione dell'utente all'interno del social network tramite username e password	12
3.2.4	Seguire un utente	12
3.2.5	Smettere di seguire un utente	13
3.2.6	Creare una nuova chat	14
3.2.7	Postare e commentare	14
3.2.8	Reagire con like/dislike a post e commenti	15
3.2.9	Leggere un post	15
3.2.10	Scorrere il post pubblicati dagli utenti seguiti (feed)	15
3.2.11	Vedere il profilo di un utente	16
3.2.12	Vedere i contenuti pubblicati da un utente	16
3.2.13	Scrivere un messaggio nella chat	17

3.2.14	Uscire volontariamente dalla chat	17
3.2.15	Elencare i membri attuali della chat	18
3.2.16	Leggere i messaggi della chat	18
3.2.17	Aggiungere nuovi membri alla chat	18
3.2.18	Dare diritti amministratore ad un utente in una chat .	19
3.2.19	Cacciare un utente da una chat	19
3.2.20	Ricostruire ricorsivamente le regioni superiori di una data regione	20
3.3	Raffinamento dello schema	21
3.3.1	Eliminazione di gerarchie	21
3.3.2	Numero di like netto dei contenuti	21
3.3.3	Numero di follower di un utente	23
3.3.4	Reputazione di un utente	25
3.4	Traduzione di entità e associazioni in relazioni	26
3.5	Schema relazionale finale	28
3.6	Traduzione delle operazioni in query SQL	29
3.6.1	Creazione tabelle	29
3.6.2	Operazioni utente	34
4	Progettazione dell'applicazione	37
4.1	Descrizione dell'architettura dell'applicazione e alcuni screen- shot dell'interfaccia utente	37

Chapter 1

Analisi dei requisiti

1.1 Intervista

Vogliamo sviluppare un social network puramente testuale. Gli **utenti** possono iscriversi e disiscriversi. Al momento dell'iscrizione l'utente inserisce un username identificativo e opzionalmente le sue generalità (nome, cognome, data di nascita, proprio domicilio). Inoltre decide una password che utilizzerà per autenticarsi al social network. L'utente può cambiare la sua password ma non può riutilizzare una password precedente. Ogni utente può pubblicare dei **post**. Un post consiste in un titolo, un testo scritto, opzionalmente accompagnato da una locazione decisa dall'autore (cioè l'utente che ha scritto il post). I post possono essere commentati dagli utenti. Ogni utente può mettere **like** e dislike a post e commenti. Ricevendo reazioni ai propri post e commenti gli utenti si costruiscono una reputazione, cioè un indice frutto della somma algebrica tra like e dislike. Gli utenti inoltre possono **seguire** altri utenti e smettere di seguirli. Sarà possibile avere uno storico di queste operazioni. Il social fornirà ad ogni utente un feed, cioè una sequenza di post pubblicati dagli utenti da lui seguiti. Gli utenti possono chattare con altri utenti. Le **chat** si compongono di messaggi inviati dai membri di quella chat. Ogni **messaggio** ha un suo autore, un timestamp di invio, un testo ed può citare un post o commento. Ogni messaggio inoltre può essere in risposta ad un altro messaggio di quella chat. Ogni utente può creare una **chat**. Ogni chat ha uno o più membri amministratori. Gli **amministratori** possono far entrare nuovi membri nelle chat di cui sono amministratori. Gli amministratori hanno diritto di cacciare degli utenti dalla chat, eventualmente comunicando una motivazione. Gli utenti possono uscire volontariamente dalle chat a cui appartengono. Gli utenti che escono volontariamente dalla chat possono lasciare una motivazione del loro gesto.

1.2 Definizioni

- Utente: un iscritto al nostro social network.
- Post: un testo scritto da un utente
- Autore di un post: l'utente che ha scritto il post
- Commento: un post che fa riferimento ad un altro post (detto post padre)
- Like: è una segnalazione di gradimento ad un post
- Dislike: l'opposto di un like
- Reputazione di un utente: la differenza tra like e dislike complessivi dei post scritti dall'utente
- Messaggio: un testo scritto da un utente con destinatario una chat
- Chat: una sequenza di messaggi
- Membro di una chat: un utente che fa parte della chat
- Regione: un luogo geografico
- Amministratore di una chat: un utente che può aggiungere e rimuovere utenti dalla chat
- Creatore di una chat: l'utente che ha creato la chat, in altre parole il primo membro di una chat (nonchè il primo amministratore)

1.2.1 Operazioni utente

- Creare un nuovo utente
- Cambiare password utente
- Seguire utente
- Mandare un messaggio
- Postare e commentare
- Like/dislike
- Creare un gruppo

- Aggiungere utente al gruppo
- Vedere tutti i post in ordine cronologico (feed)
- Vedere utenti che hanno messo like a un post
- Leggere messaggi di una chat
- Login utente
- Aggiungere location a tabella location
- Vedere profilo utente (location, reputazione, post ...)
- Dare diritti amministratore ad un utente in un gruppo

Chapter 2

Progettazione Concettuale

2.1 Utenti

L'entità utente ha come unico identificatore lo username. Per permettere agli utenti di seguire altri utenti abbiamo creato una associazione ad anello SEGUIRE. Però per permettere la creazione di uno storico dei seguiti abbiamo reificato questa associazione. Ogni qual volta l'utente entra nel social network viene creata una istanza di STORICO_ACCESSO.

2.2 Chat e membri della chat

Quando un utente entra a fare parte di una chat viene creata una istanza di MEMBRO, identificata dalla tripla (utente, chat, data di entrata). Rimane inespresso il vincolo secondo cui un utente non far parte contemporaneamente della stessa chat. Quando un membro esce dalla chat la sua istanza non viene eliminata ma semplicemente viene creata una istanza di USCITA, la quale verrà referenziata dal membro uscito. Modelliamo i membri amministratori come sottotipi di membri. Ogni membro ha un membro amministratore che lo ha aggiunto alla chat tranne il creatore della chat. Non riusciamo ad esprimere tramite E/R il vincolo secondo cui gli amministratori possono aggiungere utenti alle chat di cui amministratori.

2.2.1 Uscita dalla chat

L'uscita volontaria dalla chat è rappresentata dal sottotipo di USCITA chiamato VOLONTARIA. Quando invece un amministratore caccia un membro allora si crea una istanza di tipo ESILIO, la quale referencia l'amministratore responsabile della cacciata.

2.3 Regione

Una regione può far parte di una ed una sola regione più grande, mentre una regione può contenere delle regioni più piccole. Esprimiamo questi vincoli con una relazione ad anello. Si viene perciò a formare un albero (o una foresta di alberi) che esprimono le gerarchie delle regioni geografiche. Lasciamo però inesperto il vincolo secondo cui non si debbano formare dei cicli all'interno di questa foresta, vincolo che abbiamo intenzione di imporre a livello applicativo.

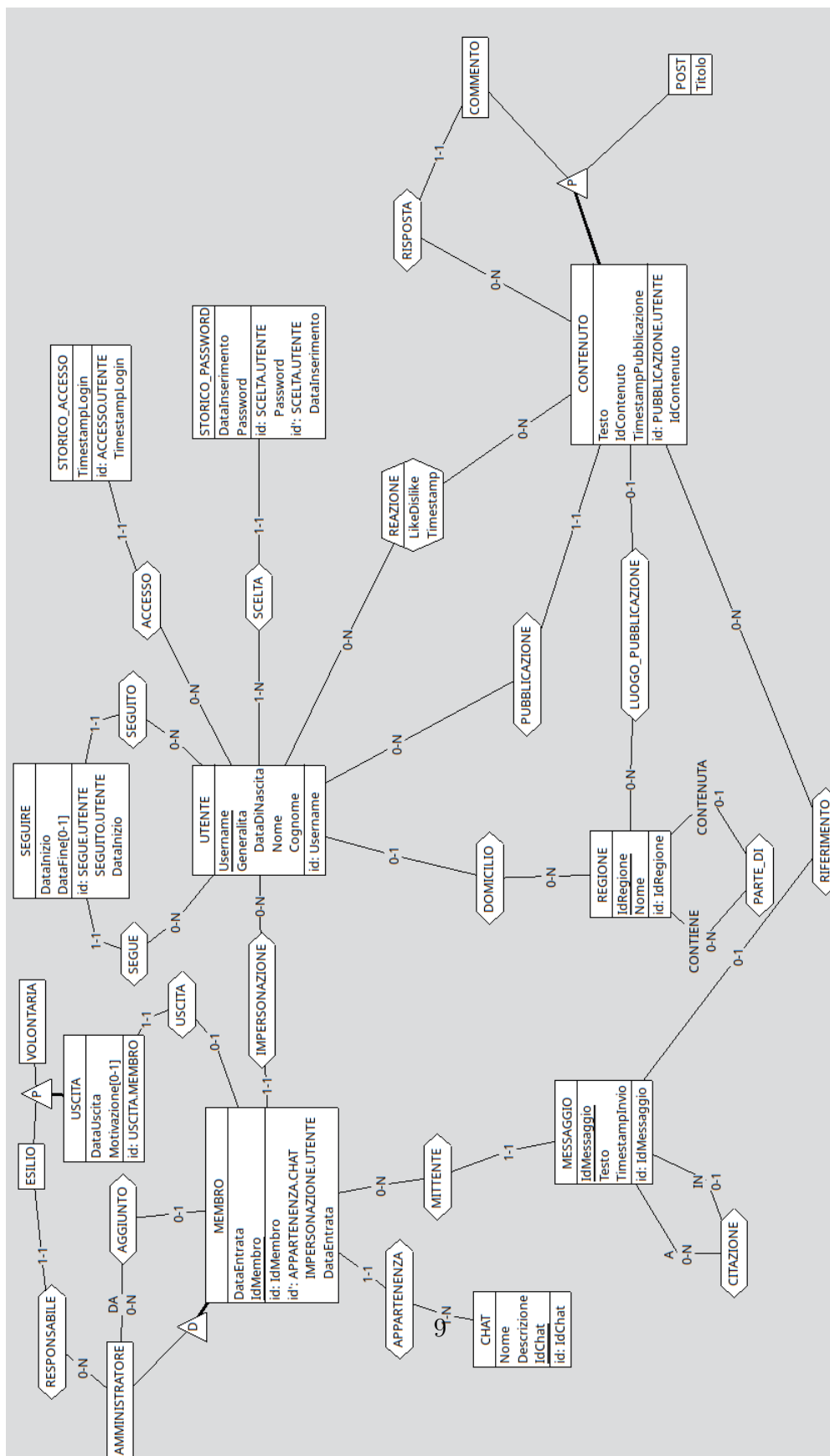
2.4 Password

Ogni utente ha uno storico di password che ci permette di impedire all'utente di riutilizzare password passate. Questo vincolo è espresso imponendo come identificatore della entità `STORICO_PASSWORD` la tupla `(Password, UTENTE)`. Imponiamo inoltre il vincolo secondo cui un utente non può avere contemporaneamente due password creando l'identificatore `(Password, DataInserimento)`.

2.5 Contenuto (post e commenti)

Abbiamo modellizzato post e commenti in una gerarchia totale ed esclusiva dove i post veri e propri (cioè quelli pubblicati **non** in risposta a qualche altro contenuto) sono istanze della sottoclasse `POST`, mentre i commenti (cioè quei contenuti in risposta ad altri contenuti) sono rappresentati dall'entità figlia `COMMENTO`. Questa entità è legata a `CONTENUTO` dalla relazione `RISPOSTA`, che indica a quale contenuto il commento fa riferimento. Per modellizzare la possibilità da parte degli utenti di mettere like e dislike ai contenuti abbiamo creato la relazione `REAZIONE` il cui attributo `LikeDislike` indica se la reazione è un like o un dislike. L'attributo `Timestamp` indica il momento temporale in cui l'utente ha espresso la sua reazione riguardo al contenuto.

2.6 Schema concettuale finale



Chapter 3

Progettazione logica

3.1 Stima del volume dei dati

- UTENTE, E, 1000
- SCELTA, R, 5000
- STORICO_PASSWORD, E, 5000
- SEGUIRE, R, 100.000
- SEGUE, E, 100.000
- SEGUIRE E, 100.000
- MEMBRO, E, 20000
- AMMINISTRATORE, E, 4000
- PUBBLICAZIONE, R, 20000
- POST, E, 20000
- COMMENTO, E, 200.000
- REAZIONE, R, 2.000.000
- IMPERSONAZIONE, R, 20000
- APPARTENENZA, R, 20000
- CHAT, E, 2000
- AGGIUNTO, R, 20000

- MESSAGGIO, E, 300000
- MITTENTE, R, 300000
- CITAZIONE, R, 10000
- DOMICILIO, R, 100
- REGIONE, E, 150.000.000
- PARTE_DI, R, 150.000.000

3.2 Tabelle degli accessi

3.2.1 Creare un nuovo utente

Descrizione Aggiungere alla tabella degli utenti una nuova riga e assegnargli una password

Frequenza 10 / g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	W
SCELTA	A	1	W
STORICO_PASSWORD	E	1	W
<i>Gerarchia regioni 3.2.20</i>		14	
<i>TOTALE</i>		20	

3.2.2 Cambiare password

Descrizione Aggiungere alla tabella dello storico password una nuova riga.

Frequenza 1 / 60 g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
SCELTA	A	1	W
STORICO_PASSWORD	E	1	W
<i>TOTALE</i>		5	

3.2.3 Autenticazione dell'utente all'interno del social network tramite username e password

Descrizione Si controlla se esiste istanza di STORICO_PASSWORD contenente l'username e la password inserite

Frequenza 10 / g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
SCELTA	A	1	R
STORICO_PASSWORD	E	5	R
ACCESSO	A	1	W
STORICO_ACCESSO	E	1	W
TOTALE		11	

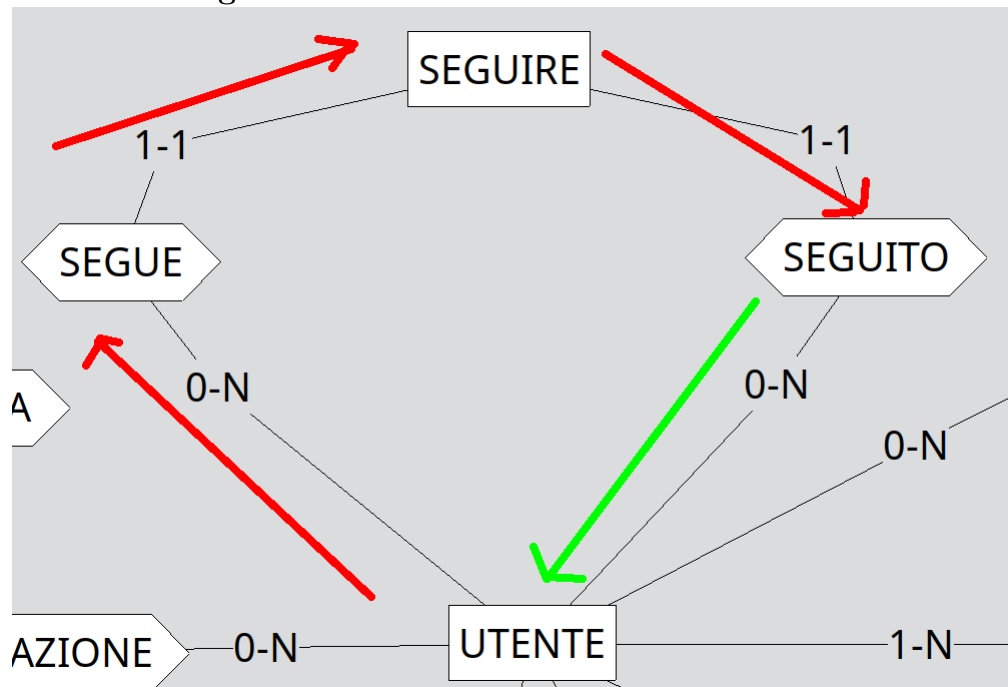
3.2.4 Seguire un utente

Descrizione Si aggiunge una istanza alla tabella SEGUIRE avente come chiave la tupla (utente seguace, utente seguito)

Frequenza 2 / g

Tavola degli accessi Trattata nell'analisi delle ridondanze.

Schema di navigazione



3.2.5 Smettere di seguire un utente

Descrizione Si accede all'istanza di **SEGUIRE** identificata dalla tupla (utente seguace, utente seguito) e si aggiunge assegna al campo DataFine la data attuale.

Tavola degli accessi Trattata nell'analisi delle ridondanze.

Frequenza 1 / 10 g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
SEGUE	A	1	R
SEGUIRE	E	1	W
UTENTE	E	1	R
TOTALE		5	

3.2.6 Creare una nuova chat

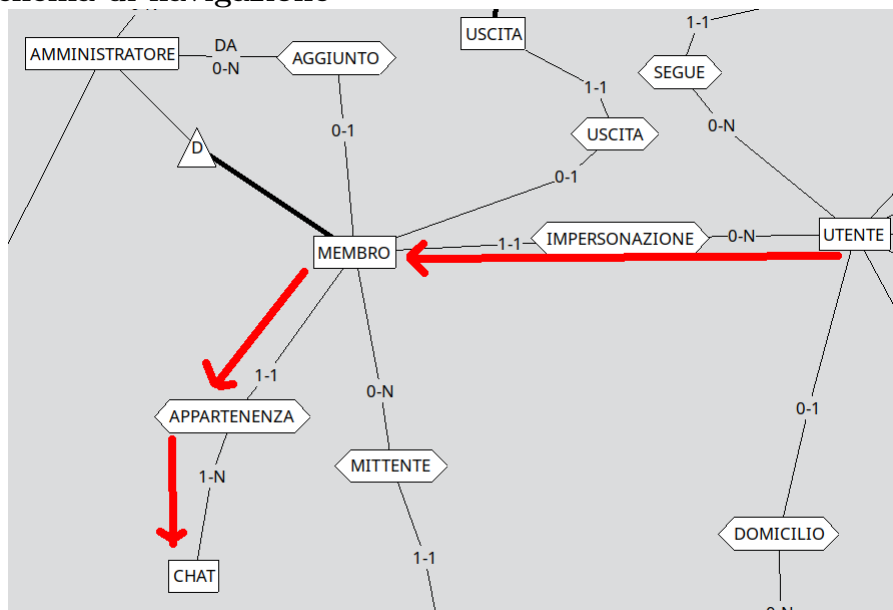
Descrizione Viene creato una nuova istanza di CHAT e un primo membro amministratore della chat.

Frequenza 1 / 20 g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
IMPERSONAZIONE	A	1	W
AMMINISTRATORE	E	1	W
APPARTENENZA	A	1	W
CHAT	E	1	W
TOTALE		9	

Schema di navigazione



3.2.7 Postare e commentare

Descrizione Aggiunge una nuova riga alla tabella CONTENUTO.

Frequenza 10 / g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
CONTENUTO	E	1	W
PUBBLICAZIONE	A	1	W
UTENTE	E	1	R
LUOGO_PUBBLICAZIONE	A	1	W
REGIONE	E	1	R
RISPOSTA	A	1	W
CONTENUTO	E	1	R
TOTALE		11	

3.2.8 Reagire con like/dislike a post e commenti

Descrizione Si aggiunge una istanza alla tabella reazione con identificatore la tupla (utente, contenuto) e nel campo LikeDislike +1 se è un like, -1 se è un dislike.

Frequenza 80 / g

Tavola degli accessi Trattata nell'analisi delle ridondanze.

3.2.9 Leggere un post

Descrizione Si mostrano il contenuto, l'ora di pubblicazione, l'autore, i commenti, il numero di like netto (cioè la somma algebrica dei like e dislike), se presente il luogo di pubblicazione.

Frequenza 100 / g

Tavola degli accessi Trattata nell'analisi delle ridondanze.

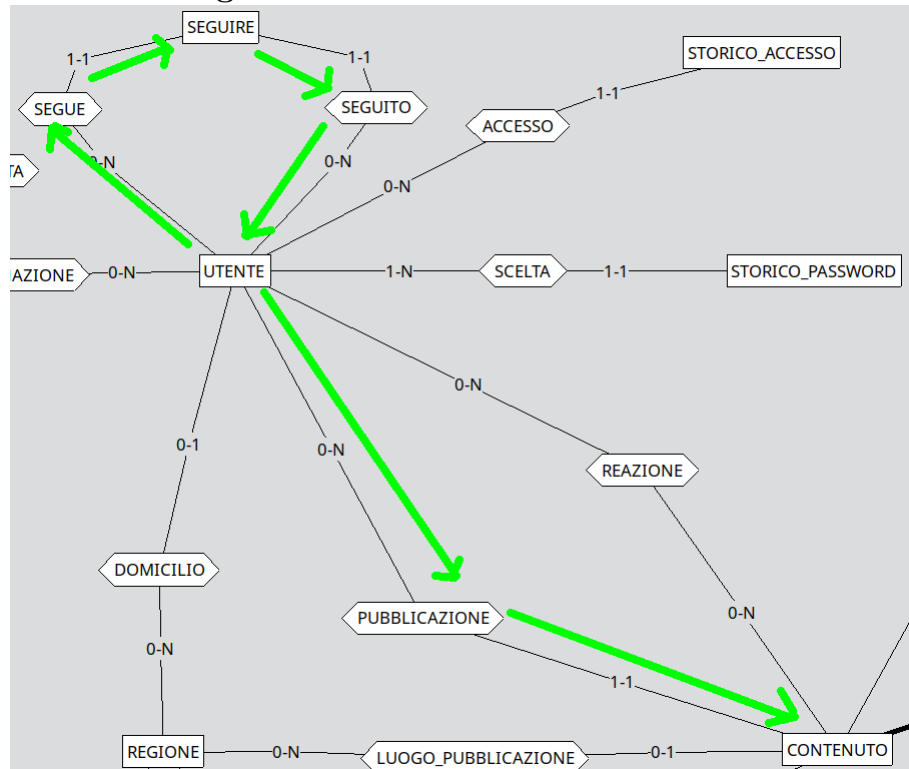
3.2.10 Scorrere il post pubblicati dagli utenti seguiti (feed)

Descrizione Questa operazione mostra in ordine cronologico i titoli, il numero di like netto e l'autore dei post pubblicati dagli utenti seguiti.

Frequenza 40 / g

Tavola degli accessi Trattata nell'analisi delle ridondanze.

Schema di navigazione



3.2.11 Vedere il profilo di un utente

Descrizione Si mostrano nome, cognome, data di nascita, location e la sua reputazione.

Frequenza 5 / g

Tavola degli accessi Trattata nell'analisi delle ridondanze.

3.2.12 Vedere i contenuti pubblicati da un utente

Descrizione Si mostrano tutti i contenuti che hanno come autore l'utente considerato

Frequenza 1 / g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
PUBBLICAZIONE	A	20	R
CONTENUTO	E	20	R
TOTALE		41	

3.2.13 Scrivere un messaggio nella chat

Descrizione Si aggiunge una istanza nella tabella MESSAGGIO con una referenza esterna al membro autore.

Frequenza 50 / g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
MEMBRO	E	1	R
MITTENTE	A	1	W
MESSAGGIO	E	1	W
CITAZIONE	A	1	W
MESSAGGIO	E	1	R
RIFERIMENTO	A	1	W
CONTENUTO	E	1	R
TOTALE		11	

3.2.14 Uscire volontariamente dalla chat

Descrizione Nell'istanza del membro che sta uscendo si aggiunge una referenza ad una nuova istanza USCITA di tipo VOLONTARIA.

Frequenza 1 / 100 g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
MEMBRO	E	1	W
USCITA	A	1	W
VOLONTARIA	E	1	W
TOTALE		6	

3.2.15 Elencare i membri attuali della chat

Descrizione Vengono elencati gli username, ruolo e data di entrata degli utenti membri della chat.

Frequenza 1 / g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
CHAT	E	1	R
APPARTENENZA	A	20	R
MEMBRO	E	20	R
TOTALE		41	

3.2.16 Leggere i messaggi della chat

Descrizione Ordinati dal più recente al più vecchio vengono mostrati il contenuto dei messaggi, il mittente, un eventuale messaggio citato e un eventuale contenuto in riferimento. Verranno ignorati i messaggi eliminati.

Frequenza 90 / g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
CHAT	E	1	R
APPARTENENZA	A	20	R
MEMBRO	E	20	R
MITTENTE	A	300	R
MESSAGGIO	E	300	R
CITAZIONE	A	300	R
MESSAGGIO	E	300	R
RIFERIMENTO	A	300	R
CONTENUTO	E	300	R
TOTALE		1841	

3.2.17 Aggiungere nuovi membri alla chat

Descrizione Viene creata una nuova istanza nella tabella MEMBRO che fa riferimento all'utente aggiunto, alla chat in cui è stato inserito, la data di entrata e il membro amministratore che lo ha aggiunto.

Frequenza 1 / g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
AMMINISTRATORE	E	1	R
AGGIUNTO	A	1	W
MEMBRO	E	1	W
IMPERSONAZIONE	A	1	W
UTENTE	E	1	R
APPARTENENZA	A	1	W
CHAT	E	1	R
TOTALE		11	

3.2.18 Dare diritti amministratore ad un utente in una chat

Descrizione L'operazione viene svolta come una uscita volontaria da parte del membro candidato a diventare amministratore e il suo reinserimento come membro amministratore.

Frequenza 1 / 4 g

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
MEMBRO	E	1	W
USCITA	A	1	W
USCITA	E	1	W
<i>TOTALE USCITA VOLONTARIA</i>		<i>6</i>	
AMMINISTRATORE	E	1	R
AGGIUNTO	A	1	W
AMMINISTRATORE	E	1	W
IMPERSONAZIONE	A	1	W
APPARTENENZA	A	1	W
TOTALE		15	

3.2.19 Cacciare un utente da una chat

Descrizione Similmente a come avviene l'uscita volontaria, l'istanza del membro cacciato referenzierà una istanza della tabella USCITA del tipo ES-ILIO.

Frequenza 1 / 10 g

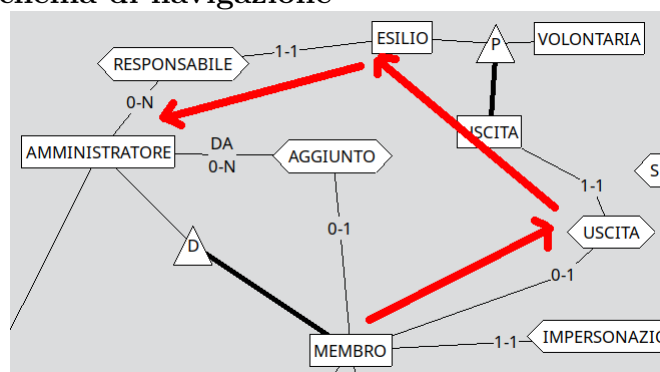
Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
MEMBRO	E	1	W
USCITA	A	1	W
ESILIO	E	1	W
RESPONSABILE	A	1	W
AMMINISTRATORE	E	1	R
TOTALE		9	

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
MESSAGGIO	E	1	R
AMMINISTRATORE	E	1	R
TOTALE		3	

Schema di navigazione



3.2.20 Ricostruire ricorsivamente le regioni superiori di una data regione

Descrizione Si percorre ricorsivamente il cammino che parte dalla regione data fino alla radice dell'albero attraversando tutte le regioni padre della regione data.

Frequenza 1 / 100 g

NOTA Nel calcolo degli accessi abbiamo posto come ipotesi che l'altezza massima di questo albero sia 7.

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
REGIONE	E	7	R
PARTE_DI	A	7	R
TOTALE		14	

3.3 Raffinamento dello schema

3.3.1 Eliminazione di gerarchie

Nel nostro social network, un contenuto può essere sia un post che un messaggio. Dato che i messaggi non hanno un titolo, mentre un post ha necessariamente un titolo, abbiamo optato definirli attraverso la presenza questo attributo, usando un collasso verso l'alto. Un utente può essere un amministratore o no in una chat, e anche in questo caso abbiamo optato per un collasso verso l'alto. Infatti, abbiamo inserito creato una table amministratori, e se il membro presente in questa tabella allora quel membro è anche un amministratore. Questo ci permette di usare il vincolo referenziale per garantire che le operazioni che sono di esclusiva competenza degli amministratori siano svolte solo da loro.

3.3.2 Numero di like netto dei contenuti

Consideriamo se è il caso di aggiungere un campo ridondante nell'entità CONTENUTO che calcola il numero netto di like riguardanti quel contenuto. Le operazioni interessate da questa ridondanza sono le operazioni 3.2.8, 3.2.9, 3.2.10.

Caso senza ridondanza

Operazione 3.2.8

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
REAZIONE	A	1	W
CONTENUTO	E	1	R
<i>TOTALE</i>		4	

Operazione 3.2.9

Concetto	Costrutto	Accessi	Tipo
CONTENUTO	E	1	R
REAZIONE	A	200	R
PUBBLICAZIONE	A	1	R
UTENTE	E	1	R
LUOGO_PUBBLICAZIONE	A	1	R
REGIONE	E	7	R
PARTE_DI	E	7	R
RISPOSTA	A	10	R
COMMENTO	E	10	R
REAZIONE	A	2000	R
TOTALE		2238	

Operazione 3.2.10

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
SEGUE	A	100	R
SEGUIRE	E	100	R
SEGUITO	A	100	R
UTENTE	E	100	R
PUBBLICAZIONE	A	2000	R
CONTENUTO	E	2000	R
REAZIONE	A	400000	R
TOTALE		404401	

Il costo giornaliero di queste 3 operazioni in totale:

$$\frac{80}{g} \times 4R + \frac{100}{g} \times 2.238R + \frac{40}{g} \times 404.401R = 16.400.160 \frac{R}{g} \quad (3.1)$$

Caso con ridondanza**Operazione 3.2.8**

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
REAZIONE	A	1	W
CONTENUTO	E	2	W
<i>TOTALE</i>		5	

Operazione 3.2.9

Concetto	Costrutto	Accessi	Tipo
CONTENUTO	E	1	R
PUBBLICAZIONE	A	1	R
UTENTE	E	1	R
LUOGO_PUBBLICAZIONE	A	1	R
REGIONE	E	7	R
PARTE_DI	E	7	R
RISPOSTA	A	10	R
COMMENTO	E	10	R
TOTALE		38	

Operazione 3.2.10

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
SEGUE	A	100	R
SEGUIRE	E	100	R
SEGUITO	A	100	R
UTENTE	E	100	R
PUBBLICAZIONE	A	2000	R
CONTENUTO	E	2000	R
TOTALE		4401	

Il costo giornaliero di queste 3 operazioni in totale:

$$\frac{80}{g} \times 5R + \frac{100}{g} \times 38R + \frac{40}{g} \times 4.401R = 180.440 \frac{R}{g} \quad (3.2)$$

Concludiamo che dobbiamo preservare la ridondanza.

3.3.3 Numero di follower di un utente

Consideriamo la possibilità di avere un campo ridondante nel profilo utente con il numero di utenti che seguono questo utente. Le operazioni interessate sono 3.2.4, 3.2.5 e 3.2.11.

Caso senza ridondanza

Operazione 3.2.4

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
SEGUE	A	1	W
SEGUIRE	E	1	W
UTENTE	E	1	R
TOTALE		6	

Operazione 3.2.5

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
SEGUE	A	1	R
SEGUIRE	E	1	W
UTENTE	E	1	R
TOTALE		5	

Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
SEGUE	A	100	R
TOTALE		101	

Il costo giornaliero di queste 3 operazioni in totale:

$$\frac{2}{g} \times 6R + \frac{1}{10g} \times 5R + \frac{5}{g} \times 101R = 517,5 \frac{R}{g} \quad (3.3)$$

Caso con ridondanza

Operazione 3.2.4

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
SEGUE	A	1	W
SEGUIRE	E	1	W
UTENTE	E	1	W
TOTALE		7	

Operazione 3.2.5

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
SEGUE	A	1	R
SEGUIRE	E	1	W
UTENTE	E	1	W
TOTALE		6	

Operazione 3.2.11

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
TOTALE		1	

Il costo giornaliero di queste 3 operazioni in totale:

$$\frac{2}{g} \times 7R + \frac{1}{10g} \times 6R + \frac{5}{g} \times 1R = 19,6 \frac{R}{g} \quad (3.4)$$

Alla luce di ciò abbiamo scelto di preservare la ridondanza.

3.3.4 Reputazione di un utente

Consideriamo la possibilità di aggiungere un campo ridondante che tenga conto della reputazione attuale di un utente. Ricordiamo che la reputazione di un utente è la somma algebrica dei like dati ai contenuti pubblicati da quel utente. Le operazioni interessate dalla ridondanza sono 3.2.8 e 3.2.11.

Caso senza ridondanza**Operazione 3.2.8**

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
REAZIONE	A	1	W
CONTENUTO	E	1	W
<i>TOTALE</i>		5	

Operazione 3.2.11

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
REAZIONE	A	2000	R
TOTALE		2001	

Quindi il costo totale giornaliero è:

$$\frac{80}{g} \times 5R + \frac{5}{g} \times 2001R = 10405 \frac{R}{g} \quad (3.5)$$

Caso con ridondanza

Operazione 3.2.8

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
REAZIONE	A	1	W
CONTENUTO	E	1	W
PUBBLICAZIONE	A	1	W
UTENTE	E	1	W
<i>TOTALE</i>		9	

Operazione 3.2.11

Concetto	Costrutto	Accessi	Tipo
UTENTE	E	1	R
TOTALE		1	

Quindi il costo totale giornaliero è:

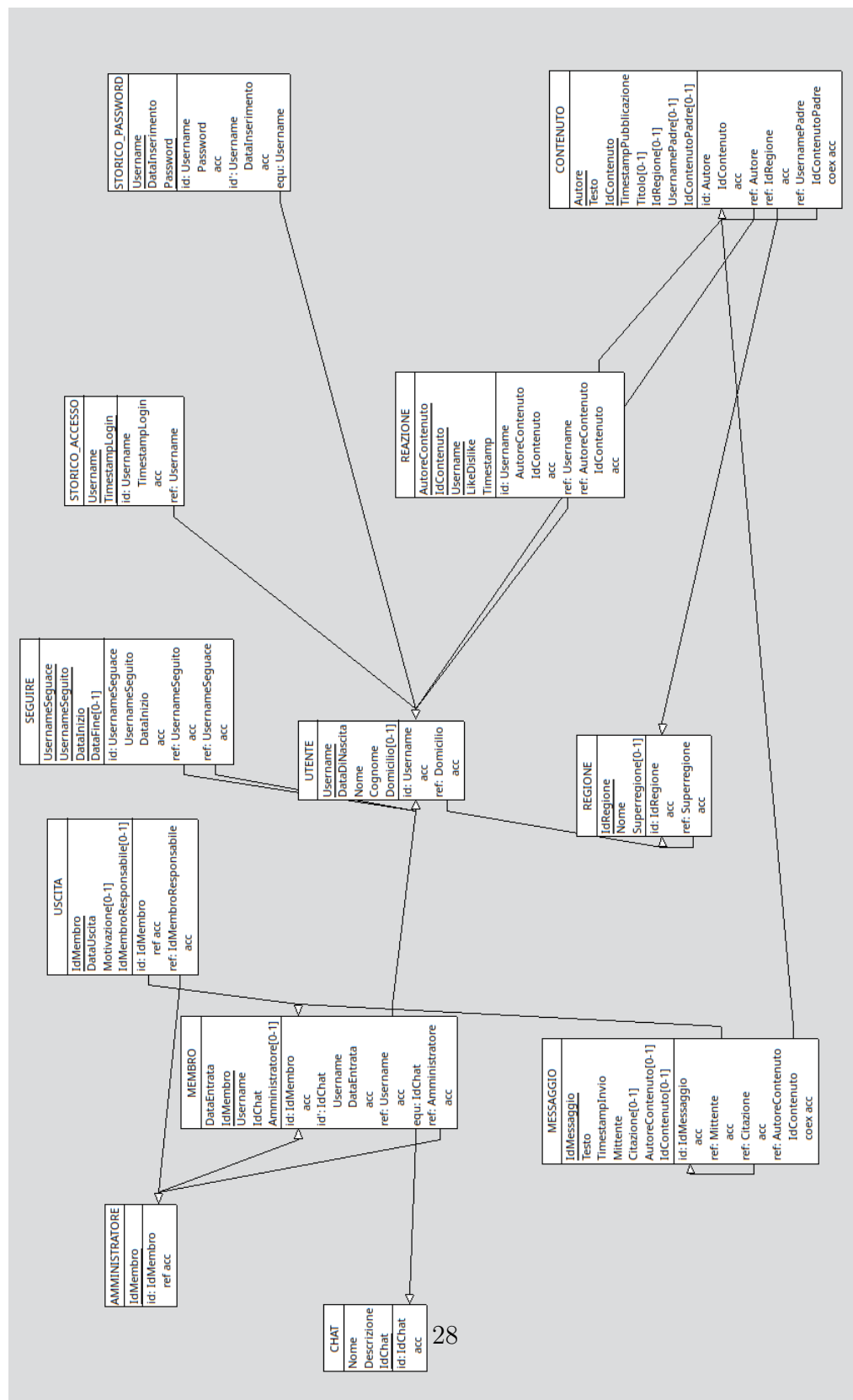
$$\frac{80}{g} \times 9R + \frac{5}{g} \times 1R = 725 \frac{R}{g} \quad (3.6)$$

In luce di ciò abbiamo deciso di mantenere la ridondanza.

3.4 Traduzione di entità e associazioni in relazioni

Abbiamo ovviamente trasformato tutte le entità del modello concettuale in relazioni, mentre per quanto riguardava le associazioni nessuna di loro è stata trasformata tranne REAZIONE, cioè l'unica che aveva come cardinalità (0,N), (0, N), Tutte le altre sono del tipo uno a molti, perciò abbiamo deciso di tradurle in attributi delle entità dal lato dell'uno. L'assenza di molte associazioni molti a molti non ci stupisce perchè molte di queste relazioni sono state reificate per supportare il sviluppo temporale di queste associazioni. Un esempio di queste è l'entità SEGUIRE, la quale era in prima analisi una associazione, ma successivamente ci siamo posti il problema di avere tutta la storia di come gli utenti inizino a seguirsi, smettono e ricominciano.

3.5 Schema relazionale finale



3.6 Traduzione delle operazioni in query SQL

3.6.1 Creazione tabelle

```
— Database Section
— _____
```

```
— Tables Section
— _____
```

```
create table AMMINISTRATORE (
    IdMembro integer not null,
    constraint FKAMMINISTRATORE_ID primary
    key (IdMembro));

create table CHAT (
    Nome varchar(100) not null,
    Descrizione varchar(5000) not null,
    IdChat serial not null,
    constraint ID_CHAT_ID primary key (IdChat));

create table CONTENUTO (
    Autore varchar(20) not null,
    Testo varchar(1000) not null,
    IdContenuto serial not null,
    TimestampPubblicazione date not null,
    Titolo varchar(30),
    IdRegione integer,
    UsernamePadre varchar(20),
    IdContenutoPadre integer,
    LikeDelta integer not null,
    constraint ID_CONTENUTO primary key
    (Autore, IdContenuto));

create table MEMBRO (
    DataEntrata date not null,
    IdMembro serial not null,
    Username varchar(20) not null,
    IdChat integer not null,
```

```

        Amministratore integer ,
        constraint ID_MEMBRO primary key (IdMembro),
        constraint SID_MEMBRO unique
        (IdChat, Username, DataEntrata));

create table MESSAGGIO (
    IdMessaggio serial not null ,
    Testo varchar(1000) not null ,
    TimestampInvio date not null ,
    Mittente integer not null ,
    Citazione integer ,
    AutoreContenuto varchar(20),
    IdContenuto integer ,
    AmministratoreEliminatore integer ,
    constraint ID_MESSAGGIO primary key
    (IdMessaggio));

create table REAZIONE (
    AutoreContenuto varchar(20) not null ,
    IdContenuto integer not null ,
    Username varchar(20) not null ,
    LikeDislike integer not null ,
    Timestamp timestamp not null ,
    constraint ID_REAZIONE primary key
    (Username, AutoreContenuto, IdContenuto));

create table REGIONE (
    IdRegione serial not null ,
    Nome varchar(50) not null ,
    Superregione integer ,
    constraint ID_REGIONE primary key (IdRegione));

ALTER sequence REGIONE_IdRegione_seq
    MINVALUE 0 RESTART WITH 0;

create table SEGUIRE (
    UsernameSeguace varchar(20) not null ,
    UsernameSeguito varchar(20) not null ,
    DataInizio date not null ,
    DataFine date ,
    constraint ID_SEGUIRE primary key

```

```

        (UsernameSeguace, UsernameSeguito, DataInizio),
        constraint DataInizioMinDataFine CHECK
        (DataInizio < DataFine),
        constraint SeguireNonSeStessi CHECK
        (UsernameSeguace <> UsernameSeguito));

create table STORICO_ACCESO (
    Username varchar(20) not null,
    TimestampLogin timestamp not null,
    constraint ID_STORICO_ACCESO
    primary key (Username, TimestampLogin));

create table STORICO_PASSWORD (
    Username varchar(20) not null,
    DataInserimento timestamp not null,
    Password varchar(20) not null,
    constraint SID_STORICO_PASSWORD unique
    (Username, DataInserimento),
    constraint ID_STORICO_PASSWORD
    primary key (Username, Password));

create table USCITA (
    IdMembro integer not null,
    DataUscita date not null,
    Motivazione varchar(500),
    IdMembroResponsabile integer,
    constraint FKUSCITA_ID primary key (IdMembro));

create table UTENTE (
    Username varchar(20) not null,
    DataDiNascita date,
    Nome varchar(20),
    Cognome varchar(20),
    Domicilio integer,
    NumeroSeguaci integer not null,
    constraint ID_UTENTE_ID primary key (Username));

```

— Constraints Section

— _____


```

alter table AMMINISTRATORE add constraint
FKAMMINISTRATORE_FK
    foreign key (IdMembro)
    references MEMBRO;

alter table CONTENUTO add constraint FKPUBBLICAZIONE
    foreign key (Autore)
    references UTENTE;

alter table CONTENUTO add constraint
FKLUOGO_PUBBLICAZIONE_FK
    foreign key (IdRegione)
    references REGIONE;

alter table CONTENUTO add constraint FKRISPOSTA_FK
    foreign key (UsernamePadre, IdContenutoPadre)
    references CONTENUTO;

alter table CONTENUTO add constraint FKRISPOSTA_CHK
    check((UsernamePadre is not null and
           IdContenutoPadre is not null)
           or (UsernamePadre is null and
              IdContenutoPadre is null));

alter table MEMBRO add constraint FKIMPERSONAZIONE_FK
    foreign key (Username)
    references UTENTE;

alter table MEMBRO add constraint FKAPPARTENENZA
    foreign key (IdChat)
    references CHAT;

alter table MEMBRO add constraint FKAGGIUNTO_FK
    foreign key (Amministratore)
    references AMMINISTRATORE;

alter table MESSAGGIO add constraint FKMITTENTE_FK
    foreign key (Mittente)
    references MEMBRO;

alter table MESSAGGIO add constraint FKCITAZIONE_FK

```

```

foreign key (Citazione)
references MESSAGGIO;

alter table MESSAGGIO add constraint FKRIFERIMENTO_FK
foreign key (AutoreContenuto, IdContenuto)
references CONTENUTO;

alter table MESSAGGIO add constraint
FKRIFERIMENTO_CHK
check((AutoreContenuto is not null and
      IdContenuto is not null)
      or (AutoreContenuto is null
          and IdContenuto is null));

alter table MESSAGGIO
add constraint FKELIMINAZIONE_FK
foreign key (AmministratoreEliminatore)
references AMMINISTRATORE;

alter table REAZIONE add constraint FKREA_UTE
foreign key (Username)
references UTENTE;

alter table REAZIONE add constraint FKREA_CON_FK
foreign key (AutoreContenuto, IdContenuto)
references CONTENUTO;

alter table REGIONE add constraint FKPARTE_DI_FK
foreign key (Superregione)
references REGIONE;

alter table SEGUIRE add constraint FKSEGUITO_FK
foreign key (UsernameSeguito)
references UTENTE;

alter table SEGUIRE add constraint FKSEGUE_FK
foreign key (UsernameSeguace)
references UTENTE;

alter table STORICO_ACCESO add constraint FKACCESSO
foreign key (Username)

```

```

references UTENTE;

alter table STORICO_PASSWORD add constraint FKSCelta
foreign key (Username)
references UTENTE;

alter table USCITA add constraint FKUSCITA_FK
foreign key (IdMembro)
references MEMBRO;

alter table USCITA add constraint FKRESPONSABILE_FK
foreign key (IdMembroResponsabile)
references AMMINISTRATORE;

alter table UTENTE add constraint FKDOMICILIO_FK
foreign key (Domicilio)
references REGIONE;

```

3.6.2 Operazioni utente

Creare nuovo utente

```

— name: InsertUser :exec
INSERT INTO UTENTE (Username,
DataDiNascita, Nome, Cognome, Domicilio, NumeroSeguaci)
VALUES ($1,$2,$3, $4, $5, 0);

```

Cambiare password utente

```

— name: InsertPassword :exec
INSERT INTO STORICO_PASSWORD (Username, Password,
DataInserimento) VALUES ($1, $2, $3);

```

Seguire utente

```

— name: InsertFollower :exec
INSERT INTO SEGUIRE
(UsernameSeguace, UsernameSeguito, DataInizio, DataFine)
VALUES ($1, $2, $3, $4);

```

Mandare un messaggio

```

— name: InsertMessage :exec
INSERT INTO MESSAGGIO
(Testo, TimestampInvio, Mittente) VALUES ($1, $2, $3);

```

Postare e commentare

```
— name: InsertComment :exec
INSERT INTO CONTENUTO (
  Autore, Testo, TimestampPubblicazione,
  IdRegione, UsernamePadre, IdContenutoPadre, LikeDelta
) VALUES ($1, $2, $3, $4, $5, $6, 0);
```

Like

```
— name: PutLike :exec
INSERT INTO REAZIONE (
  autorecontenuto, IdContenuto, username,
  likedislike, timestamp)
VALUES ($1, $2, $3, $4, $5)
ON CONFLICT (autorecontenuto, IdContenuto,
  username) DO UPDATE SET
  likedislike = excluded.likedislike,
  timestamp = excluded.timestamp;
```

Aggiungere utente al gruppo (non operazione utente?)

```
— name: InsertMember :one
INSERT INTO MEMBRO (
  DataEntrata, Username, IdChat, Amministratore
) VALUES ( $1, $2, $3, $4 )
RETURNING IdMembro;
```

Vedere tutti i post in ordine cronologico (feed)

```
— name: GetFullFeed :many
SELECT titolo, autore, IdContenuto,
  TimestampPubblicazione, Testo FROM SEGUIRE
JOIN CONTENUTO ON
CONTENUTO.Autore = SEGUIRE.usernameSeguito
WHERE SEGUIRE.usernameseguace = ($1) AND
  datafine IS NULL AND IdContenutoPadre IS NULL;
```

Leggere messaggi di una chat

```
— name: GetChatMessages :many
SELECT testo, TimestampInvio, username
FROM MESSAGGIO JOIN MEMBRO ON
MEMBRO.IdMembro = MESSAGGIO.Mittente
WHERE MEMBRO.idChat = $1;
```

Login utente

```

— name: Authenticate :one
SELECT COUNT(SUBQUERY.DataInserimento)
FROM (
    SELECT MIN(SP.DataInserimento) AS DataInserimento
    FROM STORICO_PASSWORD SP
    WHERE SP.Username = $1 AND SP.Password = $2
) AS SUBQUERY;

Vedere profilo utente Dare diritti amministratore ad utente

— name: InsertAdmin :exec
INSERT INTO AMMINISTRATORE (
    IdMembro
) VALUES ( $1 );

```

Chapter 4

Progettazione dell'applicazione

4.1 Descrizione dell'architettura dell'applicazione e alcuni screenshot dell'interfaccia utente

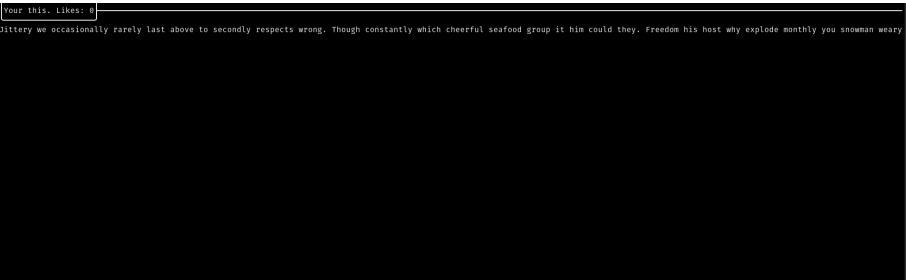
Abbiamo progettato l'applicazione cercando di tenere divisi le responsabilità di view da quelle del model. Perciò abbiamo sviluppato una struct `Model` che ha dei metodi che permettono di eseguire tutte le operazioni sul database. La view invece usa le funzionalità del model e grazie alla funzionalità di Go di implementare implicitamente le interfacce permettiamo ai vari componenti della view di dichiarare delle loro piccole interfacce private che esplicassero solo le funzionalità del model di cui hanno bisogno, usando il concetto di interfacce implicite di go. Abbiamo introdotto dei check di base nella view (ad esempio non è possibile inserire password vuote nella view di registrazione), ma abbiamo preferito il più possibile di inserire direttamente questi constraint a livello di schema.

Registrazione:

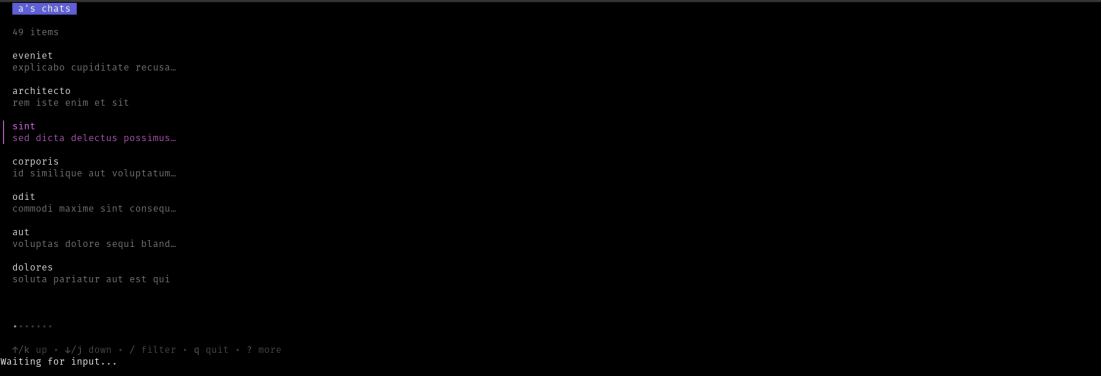
```
Press Enter to submit
> Username
> Password
> Name
> Surname
> Birthdate

Press ctrl+l to toggle location selection
Waiting for input...
```

Visualizzazione di un post:



Selezione chat:



Chat singola:

