

Università Politecnica delle Marche



Facoltà di Ingegneria

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

Progettazione e implementazione di un database NoSQL per il social network Instagram

Angelini Giordano, Di Silvestre Cristian, Scuriatti Mattia

Indice

ANALISI DEI REQUISITI	2
REQUISITI ESPRESSI IN LINGUAGGIO NATURALE	2
GLOSSARIO DEI TERMINI	3
PROGETTAZIONE CONCETTUALE	5
TAVOLA DEI VOLUMI	6
TAVOLA DELLE OPERAZIONI	7
ANALISI DELLE RIDONDANZE	9
PARTIZIONAMENTI ORIZZONTALI.....	13
AGGREGATI.....	14
TRACCIAMENTO DEI CONFINI DI AGGREGAZIONE	14
PROGETTAZIONE DEGLI AGGREGATI.....	24
TABELLE DI ANALISI DEGLI AGGREGATI	24
RAPPRESENTAZIONE E PARTITIONING DEGLI AGGREGATI	45
NOAM	45
ACCESS PATTERN	49
TRADUZIONE DELLO SCHEMA NOAM	54
COLLECTION STORY	54
COLLECTION USER.....	55
COLLECTION DIRECT	55
COLLECTION THREAD	56
COLLECTION POST	57
COLLECTION HASHTAG.....	57
COLLECTION COMMENT	58
COLLECTION CHAT.....	58
IMPLEMENTAZIONE	59
QUERY.....	59

Analisi dei requisiti

Il progetto descritto si propone di realizzare una base di dati non relazionale che soddisfi i requisiti di un applicativo che simuli il comportamento di *Instagram*.

Instagram è un social media statunitense che permette agli utenti di scattare foto, applicarvi filtri e condividerle via Internet. Nel 2012 Facebook Inc. la acquistò per 1 miliardo di dollari. L'applicazione è stata lanciata il 6 ottobre 2010. Inizialmente era pensata per la sola condivisione di foto in formato 1:1 (quadrato), per poi aggiungere una serie di funzionalità che l'hanno portata ad essere uno tra i 10 social più utilizzati al mondo.

Requisiti espressi in linguaggio naturale

Ogni persona fisica può creare un **account utente** tramite il quale iniziare a utilizzare l'applicazione. Tale registrazione avviene fornendo alcune informazioni anagrafiche dell'utente al quale verranno associate come credenziali l'email e la password fornite nella registrazione.

Può anche decidere di modificare i dettagli dell'account come lo **username**, il **nome** o la **biografia**.

Ogni utente può, a sua volta, seguire (**follow**) o meno (unfollow) gli altri utenti sulla piattaforma, con la possibilità di **bloccare** (o sbloccare) particolari account, ad esempio se non si vuole interagire con questi ultimi.

Ogni utente può pubblicare media quali **post** e **storie** con i quali possono interagire gli altri utenti tramite:

- **like**, da poter aggiungere o rimuovere;
- **commenti**, da poter aggiungere, rimuovere e modificare;
- **reaction** (solo per le storie).

Ogni post, storia o commento può contenere uno o più **hashtag**, che viene spesso utilizzato per enfatizzare parole all'interno dei media e come funzione di ricerca, ad esempio per trovare tutti i media contenenti un certo hashtag.

Allo stesso modo, ogni post, storia o commento può contenere il **tag** di uno o più utenti dell'applicazione, utilizzato per menzionare uno user specifico nella rete del social media. Quest'ultimo è uno degli eventi che generano la ricezione di una notifica (interna all'applicazione) da parte dell'utente, eventi analoghi sono:

- like a un proprio post;
- follow da parte di un utente;
- commento a un proprio post;
- ricezione di un **direct**.

Quest'ultimo è un messaggio (di testo o multimediale) scambiato in privato tra due o più utenti. Le **chat** sono lo spazio privato in cui scambiarsi i direct.

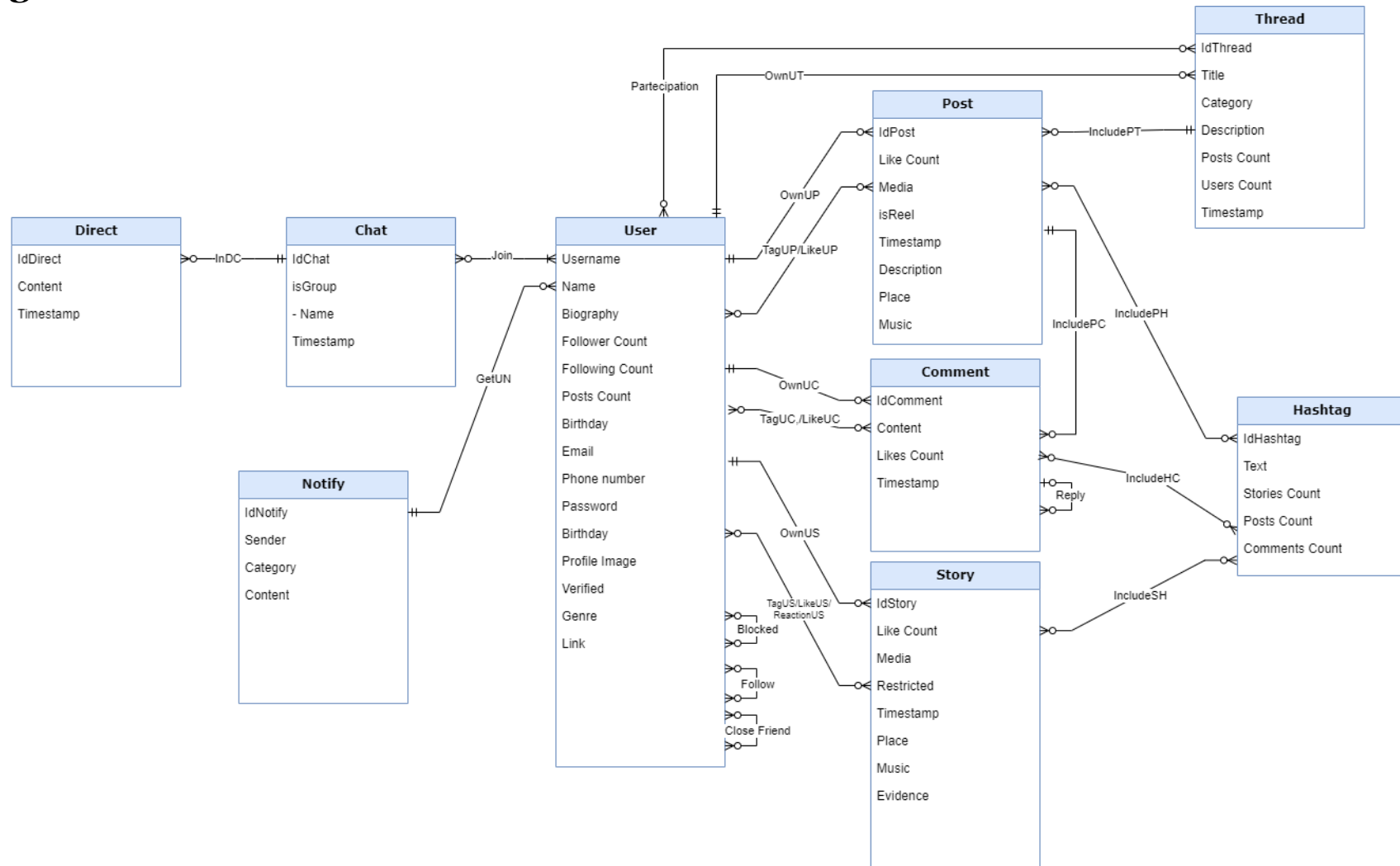
Una funzionalità aggiuntiva del lavoro svolto è quella dei **thread**, "contenitori" di post riguardanti un medesimo argomento. I thread sono creati da un utente che li amministra e ogni altro user può pubblicare post associati al thread, in modo che compaiano al suo interno.

Glossario dei termini

Termine	Descrizione	Sinonimo
Account Utente	Un profilo digitale associato ad una persona fisica nel momento in cui si registra all'interno del social. Nel dettaglio in un Account sono contenute un insieme di funzionalità e informazioni attribuite ad ogni persona.	Profilo, User, Iscritto, Account
Username	Stringa identificativa associata ad ogni utente (viene creata dall'utente nella fase di registrazione)	Nome utente
Nome	Nome e cognome (anagrafico) dell'utente	Name
Biografia	Stringa in cui un utente può scrivere un qualsiasi testo (visualizzata nel profilo)	Biography, Bio
Follow	Utente che decide di seguire un altro Utente e di riceverne, quindi, contenuti e aggiornamenti. La stessa operazione può essere effettuata da un Utente verso un Thread.	Seguire
Bloccare	Operazione che nega ad un utente di cercare il profilo di chi lo ha bloccato, non potendo quindi vedere i suoi post e le sue storie, né tanto meno di scrivergli messaggi.	Block
Post	Contenitore per file multimediali come foto o video. È pubblicato da un utente e presenta una descrizione testuale e informazioni opzionali come la posizione. Possono essere anche di tipo "reel", ovvero video nel formato 9:16 (verticale).	
Storie	Contenuti temporanei (video, foto o testi) che restano visibili per 24 ore dalla pubblicazione sul profilo Instagram.	Story
Like	Operazione che indica il gradimento di un Utente ad un Post, una Storie e/o un Commento.	Mi piace
Commenti	Un'ulteriore operazione per interagire con un contenuto del social (Storie e Post). Si tratta di una stringa che l'autore inserisce in relazione ad un contenuto.	Comment
Reaction	Interazione tra un Utente ed una Story, permettendo ai follower, gli utenti che seguono il proprietario della Story, di esprimersi verso la Story inviando un emoji (tramite click) tra quelle disponibili. Un emoji è un'icona di piccole dimensioni che rappresenta un'emozione, un simbolo o un oggetto.	Reazione

Hashtag	Parola chiave preceduta dal simbolo # che viene utilizzata per enfatizzare il contenuto di un post, una storia o un commento. Vengono salvati nella descrizione dei post, nel campo testuale dei commenti o all'interno della storia come elemento aggiuntivo al media stesso.	
Tag	Nome di uno user preceduto dal simbolo @, utilizzato per menzionare un altro user all'interno della piattaforma. Quest'ultimo riceverà una notifica che lo informa della menzione. È utile per condividere in maniera rapida contenuti sul social.	Menzione, riferimento
Direct	Messaggio di testo o multimediale scambiato in privato da due o più utenti. Instagram dispone, attraverso i direct, di un ottimo impianto di messaggistica online.	Messaggio
Chat	Contenitore per i direct, definisce quali utenti possono scambiarsi messaggi in privato e confina questi ultimi in uno spazio dedicato.	Conversazione
Thread	Contenitore per post riguardanti lo stesso argomento, ogni utente può crearne uno assegnandogli un titolo e una descrizione. Ogni utente può seguire uno o più thread e associarvi i post che pubblica.	Argomento, topic

Progettazione concettuale



In figura è visibile lo schema concettuale realizzato sulla base dell'analisi dei requisiti riportata in precedenza. Rispetto alla progettazione concettuale per database di tipo relazionale, in cui si utilizza il modello ER, in questo caso si è deciso di utilizzare il modello UML poiché la progettazione del database è parte dell'applicazione, la quale solitamente viene a sua volta progettata tramite un paradigma ad oggetti.

Al fine di attuare una ristrutturazione dello schema concettuale sono state intraprese due operazioni:

- Analisi delle ridondanze;
- Attuazione di partizionamenti orizzontali.

Prima di procedere con la descrizione delle due operazioni di cui sopra è necessario innanzitutto definire la tavola dei volumi e la tavola delle operazioni.

Tavola dei volumi

Di seguito sono riportate le tavole dei volumi divise per entità e relazioni. Le tavole fanno riferimento ad un periodo di attività del social pari a 12 anni, dunque, trattandosi di volumi molto grandi è stata adottata una notazione esponenziale per favorire la leggibilità.

Concetto	Tipo	Volume
User	E	2,00E+09
Comment	E	1,40E+11
Direct	E	1,40E+13
Post	E	7,00E+10
Hashtag	E	5,00E+10
Story	E	3,50E+12
Notify	E	6,00E+10
Thread	E	2,80E+06
Chat	E	6,00E+10

Concetto	Tipo	Volume
Join	R	1,50E+11
InDC	R	1,80E+14
InUN	R	6,00E+10
Participation	R	1,40E+10
OwnUT	R	2,80E+06
OwnUP	R	7,00E+10
OwnUS	R	3,50E+12
OwnUC	R	1,40E+11

IncludeTP	R	1,40E+09
IncludeHP	R	1,05E+12
IncludePC	R	1,40E+11
IncludeCH	R	7,00E+10
IncludeSH	R	1,75E+13
TagUS	R	3,50E+12
TagUD	R	7,00E+11
TagUP	R	7,00E+10
TagUC	R	2,80E+10
LikeUS	R	1,75E+13
LikeUP	R	3,50E+12
LikeUC	R	2,80E+11
ReactionUS	R	1,75E+13
Blocked	R	4,00E+09
Follow	R	4,00E+12
CloseFriend	R	1,00E+11
Reply	R	1,40E+11
LikeUC	R	2,80E+10

Tavola delle operazioni

La seguente tabella mostra le operazioni provviste per il sistema in analisi.

Operazione	Descrizione	Frequenza / giorno
Op.1	Memorizza nuovo User	5,00E+05
Op.2	Memorizza nuovo Comment	3,00E+07
Op.3	Memorizza nuovo Direct	4,00E+09
Op.4	Memorizza nuovo Post	1,60E+07
Op.5	Memorizza nuovo Hashtag	1,10E+07
Op.6	Memorizza nuova Story	1,40E+09
Op.7	Memorizza nuova Notify	1,40E+07
Op.8	Memorizza nuovo Thread	1,50E+03
Op.9	Memorizza nuovo Chat	1,60E+07
Op.10	Aggiornamento User	2,20E+07
Op.11	Aggiornamento Chat	4,00E+10

Op.12	Aggiornamento Thread	1,40E+07
Op.13	Aggiornamento Comment	1,20E+08
Op.14	Aggiornamento Post	2,90E+09
Op.15	Aggiornamento Story	5,00E+10
Op.16	Memorizzo nuovo Hashtag incluso in un Post	2,40E+08
Op.17	Memorizzo nuovo Hashtag incluso in un Comment	1,60E+07
Op.18	User partecipa ad un nuovo Thread	8,00E+06
Op.19	User aggiunge Like ad un Post	8,00E+08
Op.20	Tag di uno User su un Post	1,60E+07
Op.21	User aggiunge Like ad una Story	7,00E+09
Op.22	Tag di uno User su una Story	1,40E+09
Op.23	Memorizza nuovo Hashtag incluso in una Story	7,00E+09
Op.24	Reaction di una Story	7,00E+09
Op.25	Tag in un Comment	6,00E+06
Op.26	Risposta ad un Comment	3,00E+07
Op.27	Blocco di uno User	9,00E+05
Op.28	Follow di uno User	9,00E+08
Op.29	Aggiunta di uno User nei Close Friends	2,30E+07
Op.30	Aggiunta di un Post in un Thread	8,00E+05
Op.31	Visualizza User	4,00E+09
Op.32	Visualizza User che seguono	2,00E+09
Op.33	Visualizza User seguiti	2,00E+09
Op.34	Visualizza Post	3,00E+10
Op.35	Visualizza ultimi Posts di uno User	4,00E+09
Op.36	Visualizza tutti i Posts di uno User	2,00E+09
Op.38	Visualizza ultimi Posts di uno Thread	6,00E+09
Op.39	Visualizza tutti i Posts di uno Thread	3,00E+09
Op.40	Visualizza Chats di uno User	4,00E+09
Op.41	Visualizza ultimi Directs di una Chat	3,00E+09
Op.42	Visualizza tutti i Directs di una Chat	1,00E+09
Op.43	Visualizza tutte le Notifies di uno User	6,00E+09
Op.44	Visualizza tutti i Comments di un Post	9,00E+09
Op.45	Visualizza Comments con più Like di un Post	2,90E+10
Op.46	Visualizza Comments in risposta ad un Comment	2,60E+09

Op.47	Visualizza Posts con un certo Hashtag	2,00E+09
Op.49	Visualizza Stories con un certo Hashtag	2,00E+09
Op.50	Visualizza le Stories di uno Users	4,00E+10
Op.51	Visualizza archivio delle Stories	2,00E+08
Op.52	Visualizza Stories in evidenza	4,00E+09
Op.53	Visualizza User taggati su un Story	1,00E+09
Op.54	Visualizza Reaction su una Story	2,00E+09
Op.55	Visualizza User che hanno messo like ad una Story	6,00E+09
Op.56	Visualizza User che partecipano ad un Thread	4,00E+08
Op.57	Visualizza Thread posseduti da uno User	2,00E+08
Op.58	Visualizza Thread a cui partecipa uno User	2,00E+08
Op.59	Visualizza Thread	8,00E+08
Op.60	User aggiunge like ad un Comment	8,00E+07
Op.61	Visualizza Comment	6,00E+10

Analisi delle ridondanze

L'intera analisi si basa sullo stabilire l'effettiva convenienza ottenuta dall'inserimento, in alcune entità, di campi deducibili da altre informazioni. Al fine di valutare i costi delle operazioni che riguardano le varie ridondanze, si consideri che le operazioni di scrittura (S) hanno un costo doppio rispetto a quelle di lettura (L).

Nelle tabelle che seguono, per favorire la leggibilità delle stesse, si adotta una notazione esponenziale; si tenga in considerazione, inoltre, che la frequenza indicata è giornaliera.

Per lo User sono stati valutati gli attributi Follow count, Following count, Post count: tre informazioni visualizzate ogni qual volta si accede al profilo di uno user.

Follow Count/Following count				
Numero medio di seguiti		2.000		
Numero medio di seguaci		2.000		
Operazione	Descrizione	Frequenza	Con ridondanza	Senza ridondanza
Op.31	Visualizza user	1,46E+12	1L*1,46E+12	(1L+1L*2.000)*1,46E+12
			1,46E+12	2,92E+15
Op.28	Follow di uno user	9,00E+08	2S*9,00E+08	1S*9,00E+08
			3,60E+09	3,60E+09
		Totale	1,46E+12	2,92E+15

Post count				
Numero medio di post per user	35			
Numero medio di user per post	1			
Operazione	Descrizione	Frequenza	Con ridondanza	Senza ridondanza
Op.4	Memorizza nuovo post	5,00E+09	3S*5,00E+09	2S*5,00E+09
			3,00E+10	2,00E+10
Op.31	Visualizza user	1,46E+12	1L*1,46E+12	(1L+35L)*1,46E+12
			1,46E+12	5,26E+13
		Totale	1,49E+12	5,26E+13

Per quanto riguarda i Post si valuta l'attributo Like count.

Like count				
Numero medio di like per post	50			
Numero medio di post a cui utente mette like	1.750			
Operazione	Descrizione	Frequenza	Con ridondanza	Senza ridondanza
Op.34	Visualizza post	1,10E+13	1L*1,10E+13	(1L+1L*50)*1,10E+13
			1,10E+13	5,61E+14
Op.19	User aggiunge Like ad un Post	2,90E+11	2S*2,90E+11	1S*2,90E+11
			1,16E+12	5,80E+11
		Totale	1,22E+13	5,62E+14

Nelle Stories si valuta l'attributo Like count

Like count				
Numero medio di like per Stories	5			
Numero medio di Stories a cui User mette like	8.750			
Operazione	Descrizione	Frequenza	Con ridondanza	Senza ridondanza
Op.50	Visualizza Stroies di uno user	1,46E+13	1L*1,46E+13	(1L+1L*5)*1,46E+13
			1,46E+13	8,76E+13
Op.21	User aggiunge Like ad una Stories	2,50E+12	2S*2,50E+12	1S*2,50E+12
			1,00E+13	5,00E+12
		Totale	2,46E+13	9,26E+13

Si prosegue con la valutazione dell'attributo Like count in Comment.

Like count				
Numero medio di like per Comment	2			
Numero medio di Comment a cui User mette like	140			
Operazione	Descrizione	Frequenza	Con ridondanza	Senza ridondanza
Op.44	Visualizza tutti i Comments di un Post	3,20E+12	1L*3,20E+12	(1L+1L*2)*3,20E+12
			3,20E+12	9,60E+12
Op.60	User aggiunge Like ad una Comment	8,00E+07	2S*8,00E+07	1S*8,00E+07
			3,20E+08	1,60E+08
		Totale	3,20E+12	9,60E+12

Si valutano, infine, per i Thread gli attributi Post count e User count,

Post count				
Numero medio di Post per Thread	500			
Numero medio di Thread in cui è incluso un Post	0,02			
Operazione	Descrizione	Frequenza	Con ridondanza	Senza ridondanza
Op.37	Visualizza Thread	6,00E+09	1L*6,00E+09	(1L+1L*500)*6,00E+09
			6,00E+09	3,01E+12
Op.30	Aggiunta di un Post in un Thread	8,00E+05	2S*8,00E+05	1S*8,00E+05
			3,20E+06	1,60E+06
		Totale	6,00E+09	3,01E+12

User count				
Numero medio di User che partecipano a un Thread	5.000			
Numero medio di Thread a cui partecipa uno User	7			
Operazione	Descrizione	Frequenza	Con ridondanza	Senza ridondanza
Op.37	Visualizza Thread	6,00E+09	1L*6,00E+09	(1L+1L*5000)*6,00E+09
			6,00E+09	3,00E+13
Op.18	User partecipa ad un nuovo Thread	8,00E+06	2S*8,00E+06	1S*8,00E+06
			3,20E+07	1,60E+07
		Totale	6,03E+09	3,00E+13

Come si può notare in tutte le analisi effettuate il risultato è il medesimo: conviene mantenere gli attributi ridondanti. La spiegazione di tale risultato si può trovare nel fatto che l'applicazione, di cui si sta eseguendo l'analisi, prevede per le operazioni di lettura una frequenza notevolmente maggiore rispetto alle operazioni di scrittura. Viene da sé, quindi, considerare come vantaggiosa la presenza di ridondanze nel sistema.

Partizionamenti orizzontali

Nel caso in cui il volume dei dati incorporati ha una tendenza "non limitato", le dimensioni della memoria necessaria crescono, così come il numero di blocchi da recuperare dalla memoria, in maniera non prevedibile. Una possibile soluzione sta nel partizionamento orizzontale di relazione, definendo un aggregato con la sola relazione dei più recenti. In questo modo si ha il vantaggio di poter ragionare sugli aggregati sulle nuove relazioni ottenute. Difatti, in NoSQL si vorrebbe avere un aggregato limitato in dimensione per evitare problemi di memoria per diminuire, di conseguenza, i tempi di accesso.

Le relazioni che sono state individuate per effettuare su di esse un partizionamento orizzontale, riportate in rosso nello schema ristrutturato, sono le seguenti:

- *InDC*: relazione che associa i Direct alla relativa Chat;
- *IncludePC*: relazione che associa i Comment al Post;
- *OwnUP*: relazione che definisce lo User proprietario di un Post;
- *OwnUS*: relazione che definisce lo User proprietario di una Story;
- *IncludePT*: relazione che associa i Post ad un Thread.

Le relazioni partizionate sono riportate di seguito; la logica con cui è stata divisa la relazione dipende dal contesto applicativo.

- *LatestInDC*: contiene gli ultimi 40 Direct nella Chat;
- *MostLikedPC*: contiene i 15 commenti più apprezzati del post;
- *LatestOwnUP*: contiene gli ultimi 5 Post di uno User;
- *OwnUS*: contiene le storie visibili di un profilo (mediamente 50), tutte le altre sono spostate nella relazione *ExpiredOwnUS*;
- *LatestOwnUT*: contiene gli ultimi 50 Post di un Thread.

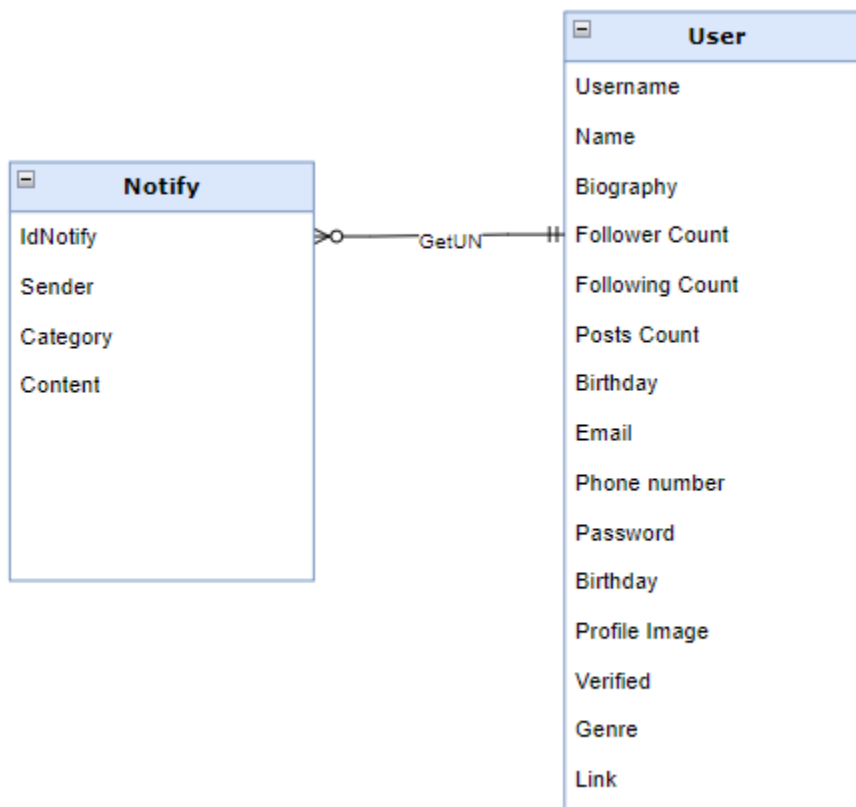
Aggregati

Tracciamento dei confini di aggregazione

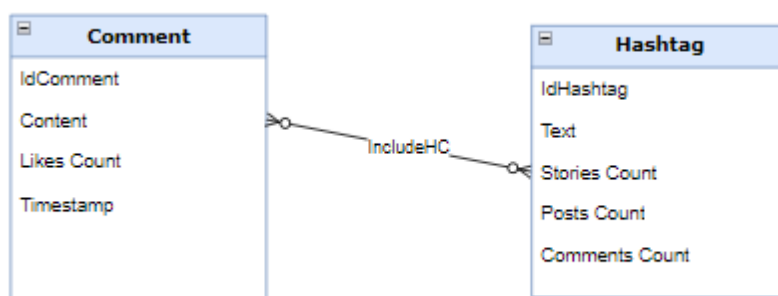
Di seguito viene riportato lo schema ristrutturato, esso è stato diviso in sezioni per garantire una maggiore leggibilità. Ogni figura riguarda un singolo confine di aggregazione, esse mantengono lo stesso ordine con cui sono riportate le tabelle di progettazione dell'aggregato visibili nella sezione seguente.

Non è riduttivo riportare i soli confini di aggregazione poiché tutte le classi rientrano in almeno un confine e, di conseguenza, rappresentando le sole classi protagoniste di processi di aggregazione si procede con la rappresentazione dell'intero contesto applicativo.

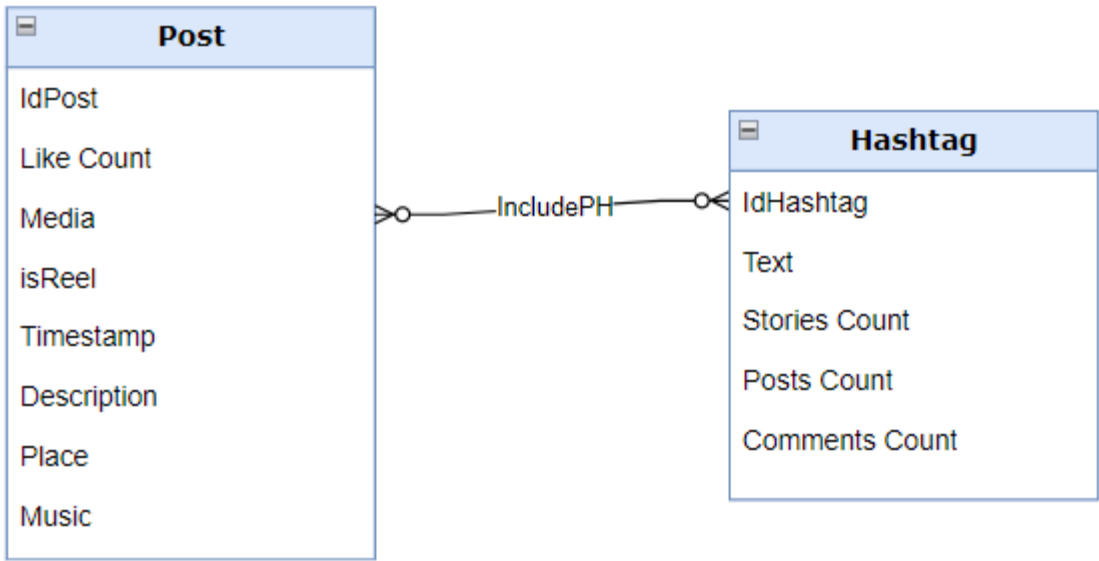
User - GetUN – Notify



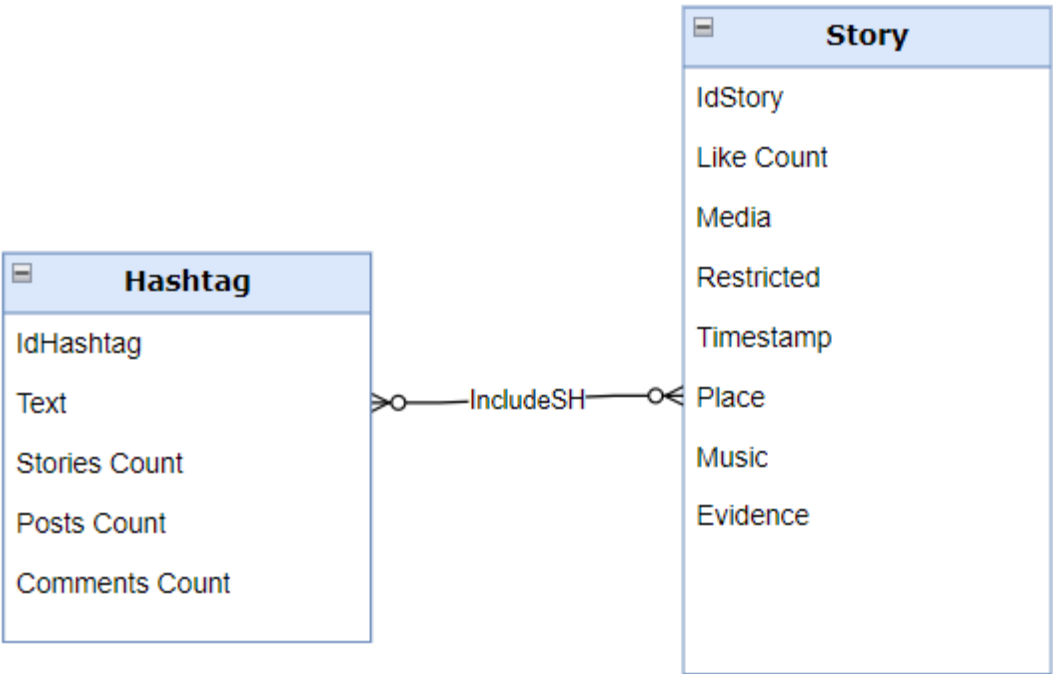
Hashtag - IncludeHC – Comment



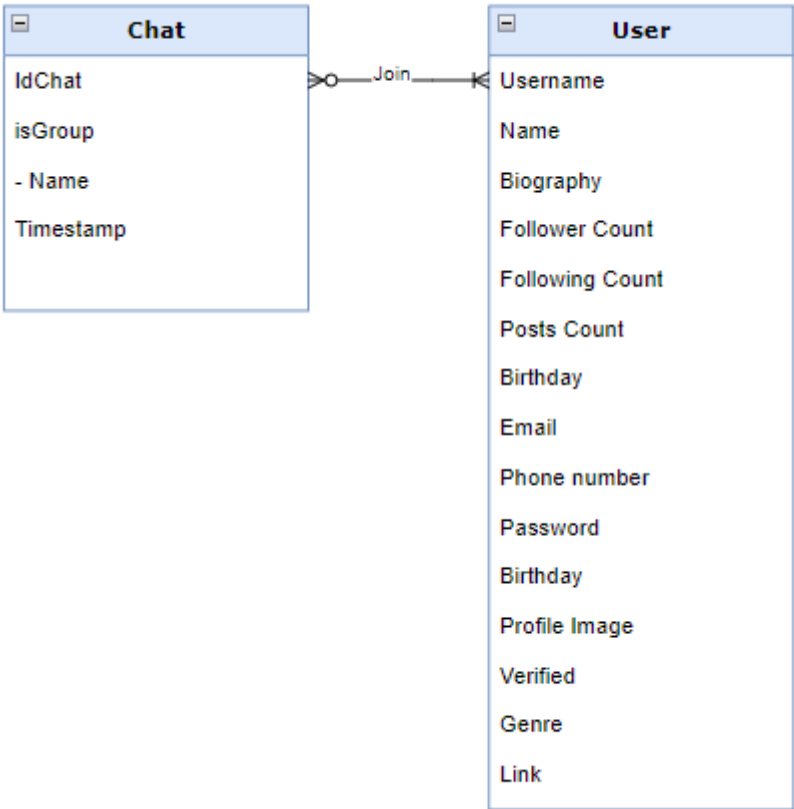
Hashtag - IncludePH – Post



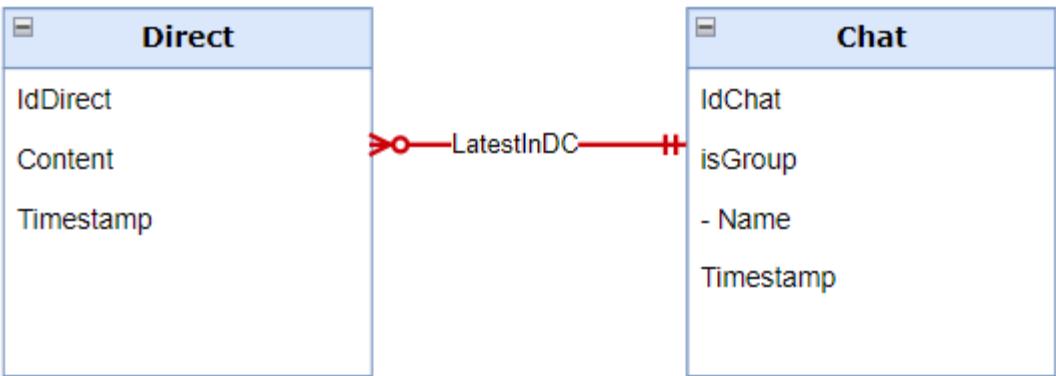
Hashtag - IncludeSH – Story



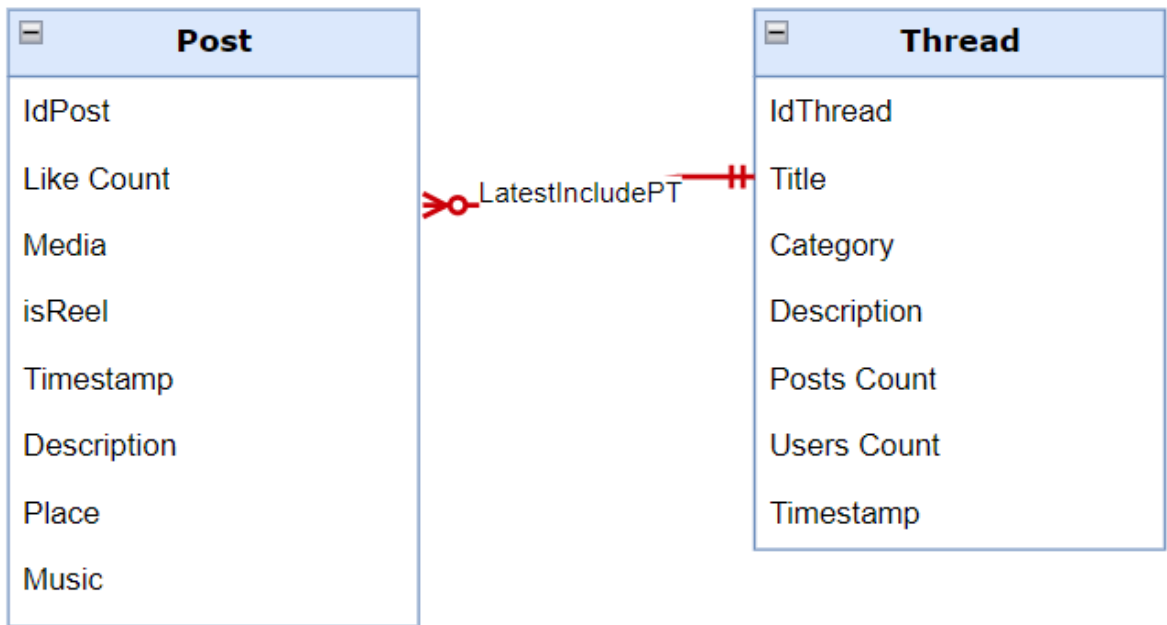
User - Join – Chat



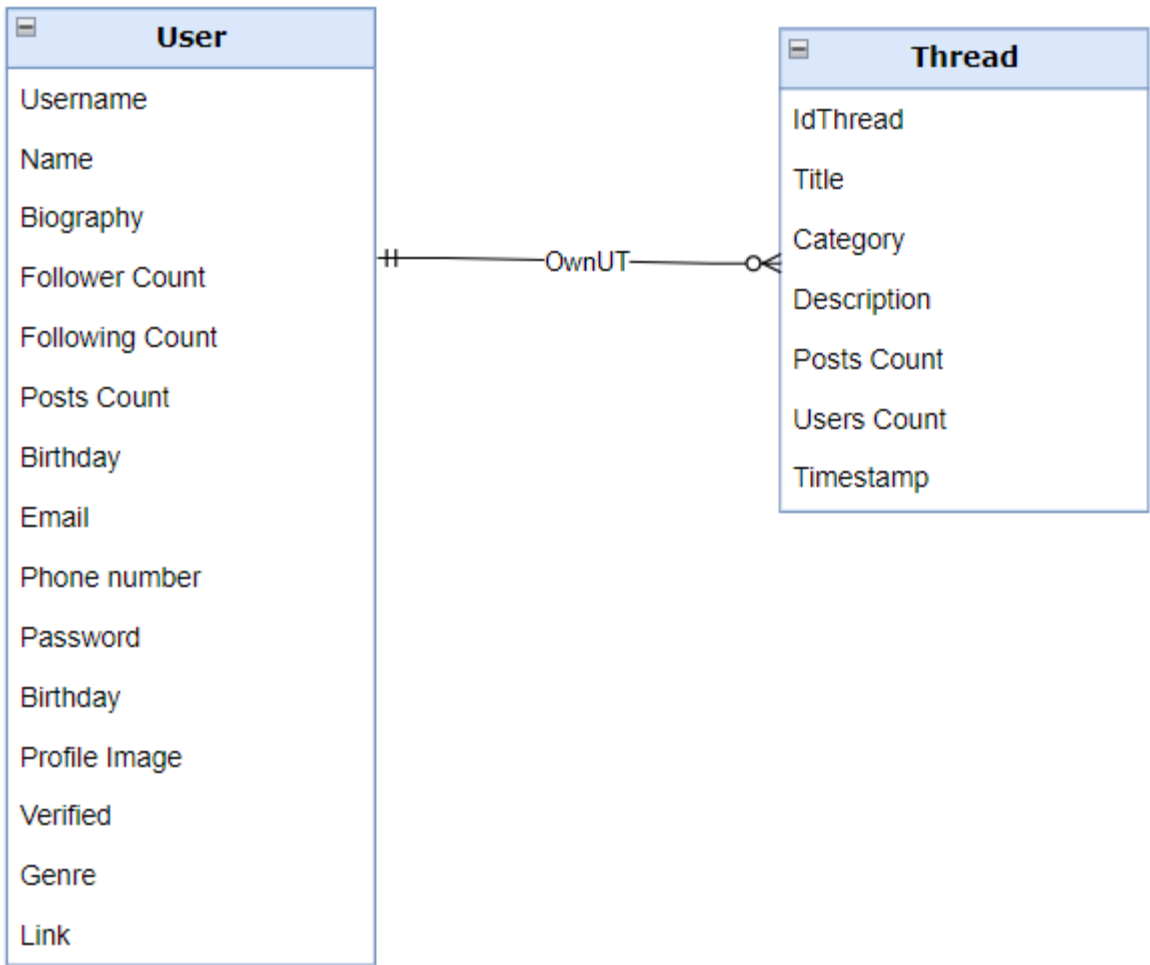
Direct - LatestInDC – Chat



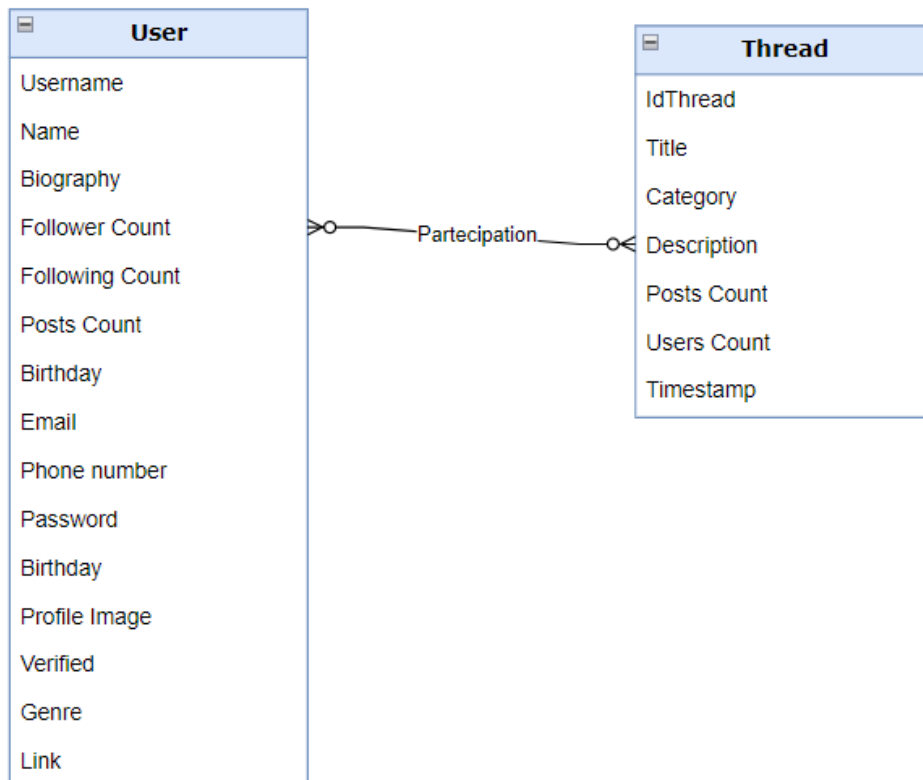
Thread - LatestIncludePT – Post



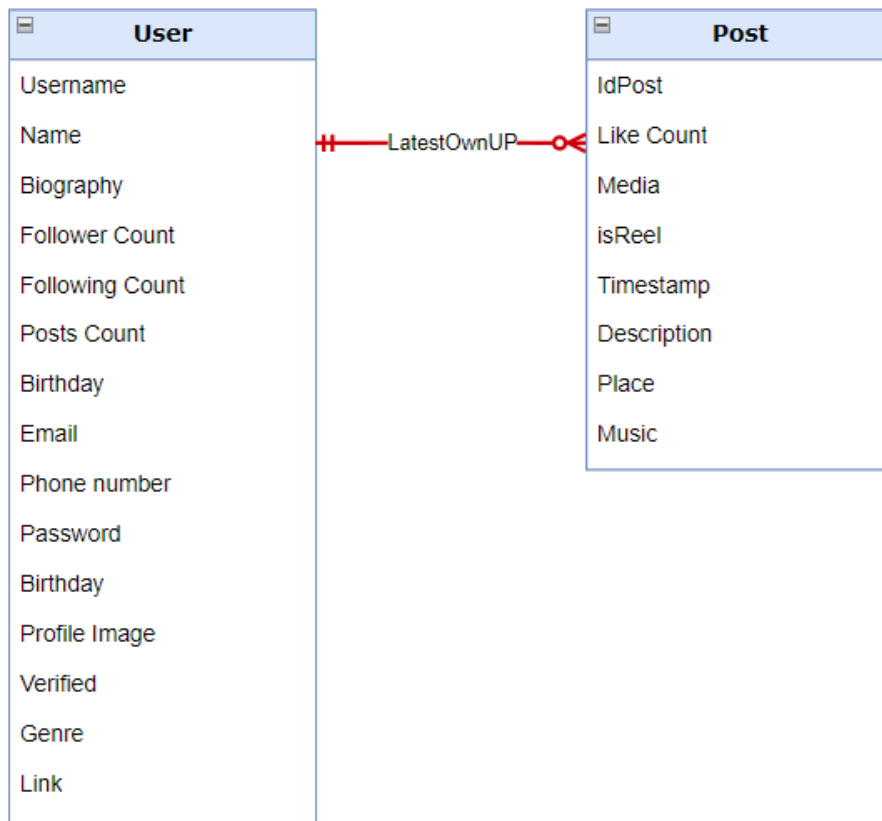
User - OwnUT – Thread



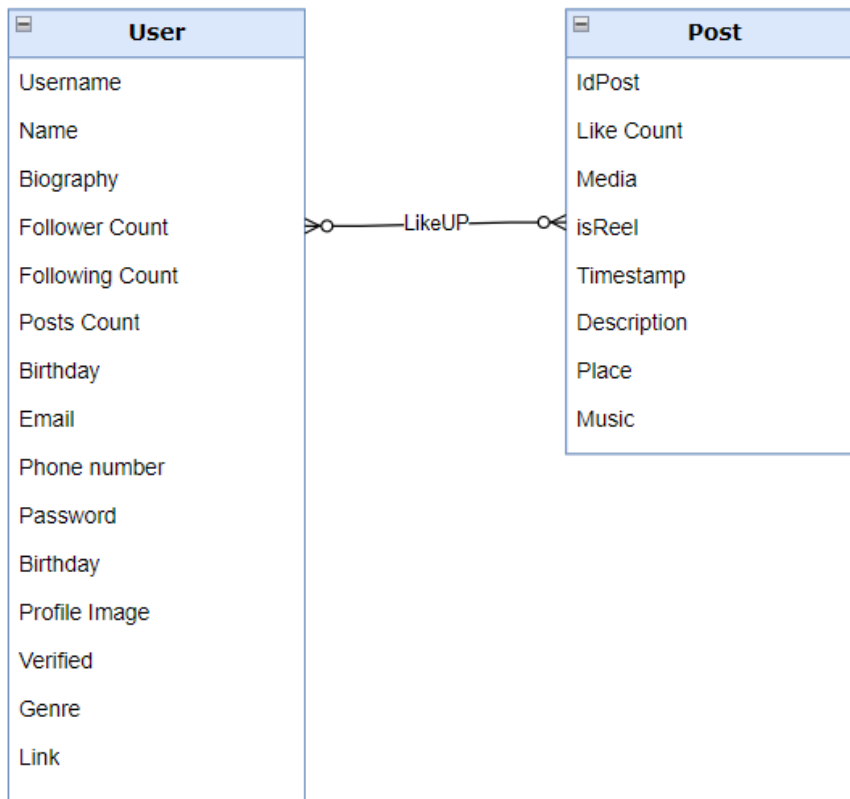
User - Participation – Thread



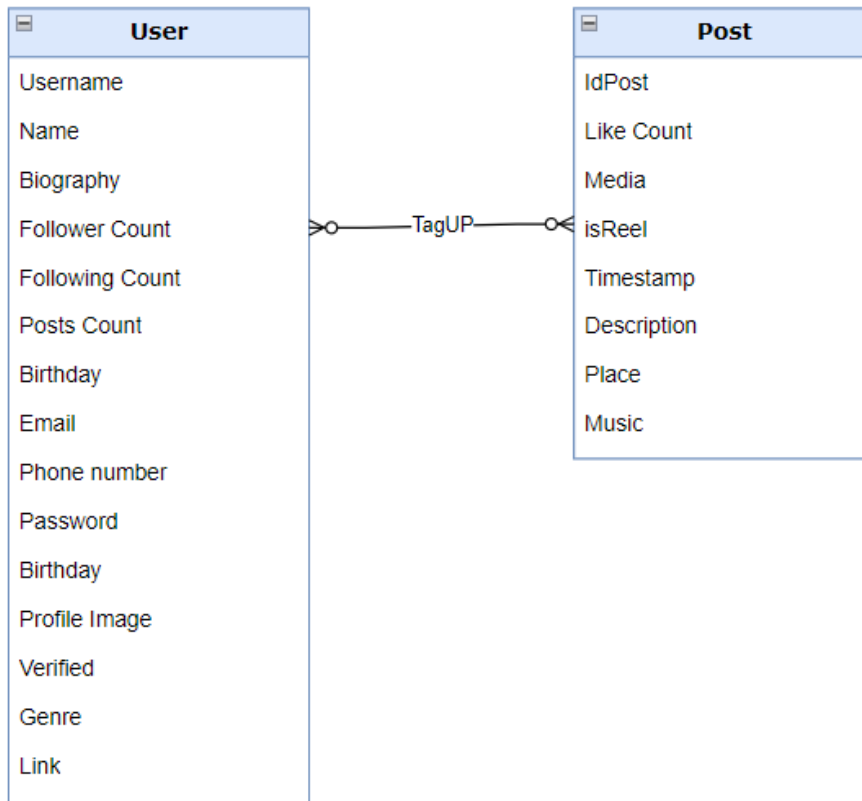
User - LatestOwnUP – Post



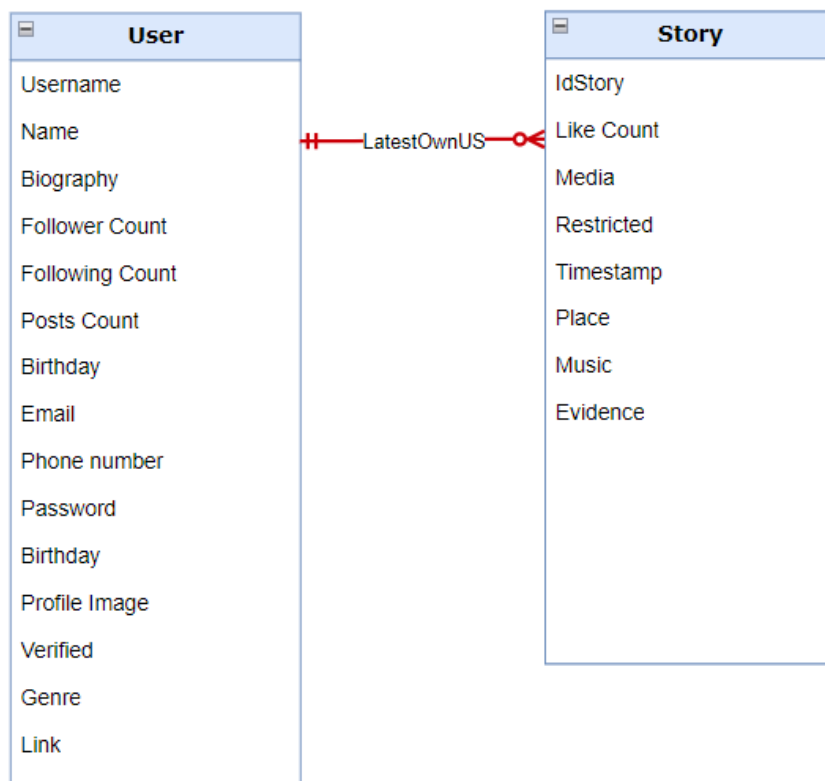
User - LikeUP – Post



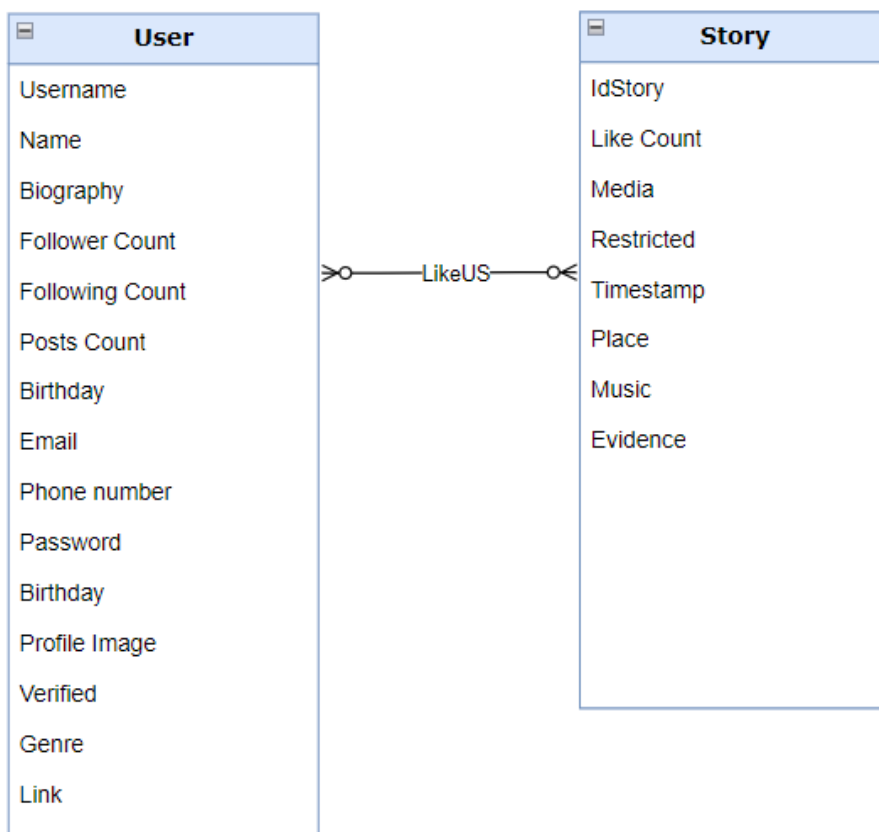
User - TagUP – Post



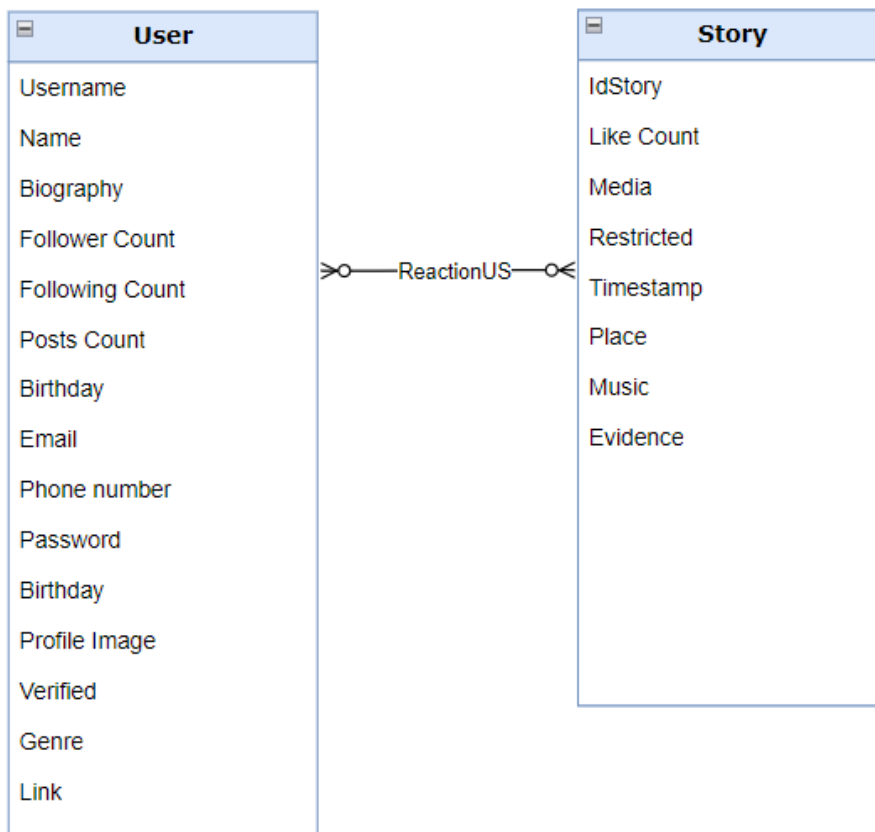
User - LatestOwnUS – Story



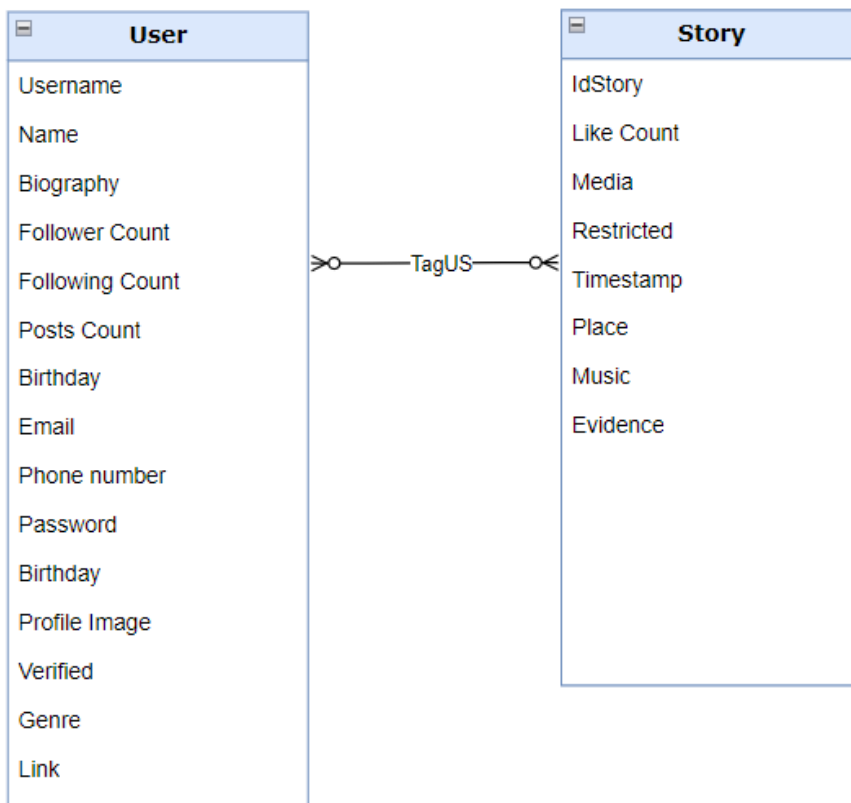
User - LikeUS – Story



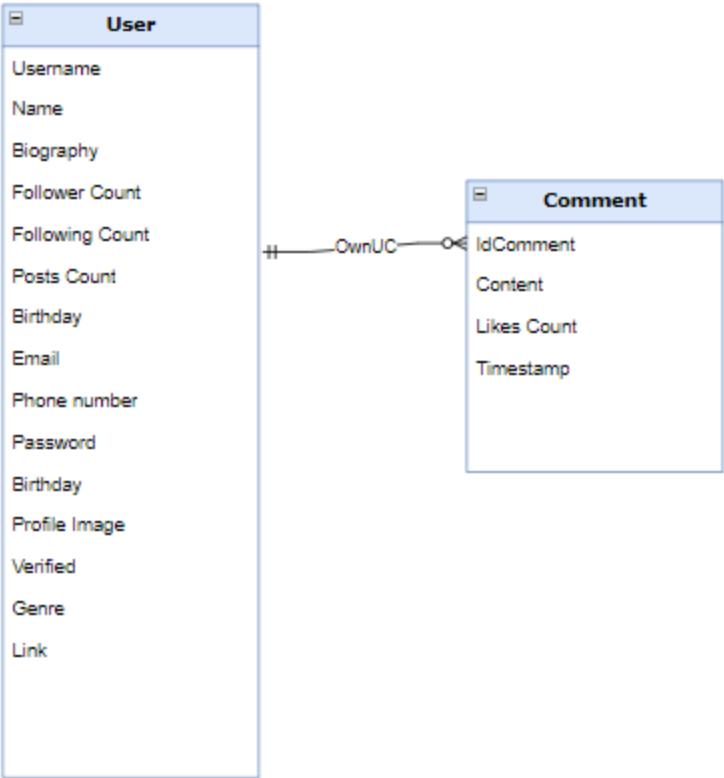
User - ReactionUS – Story



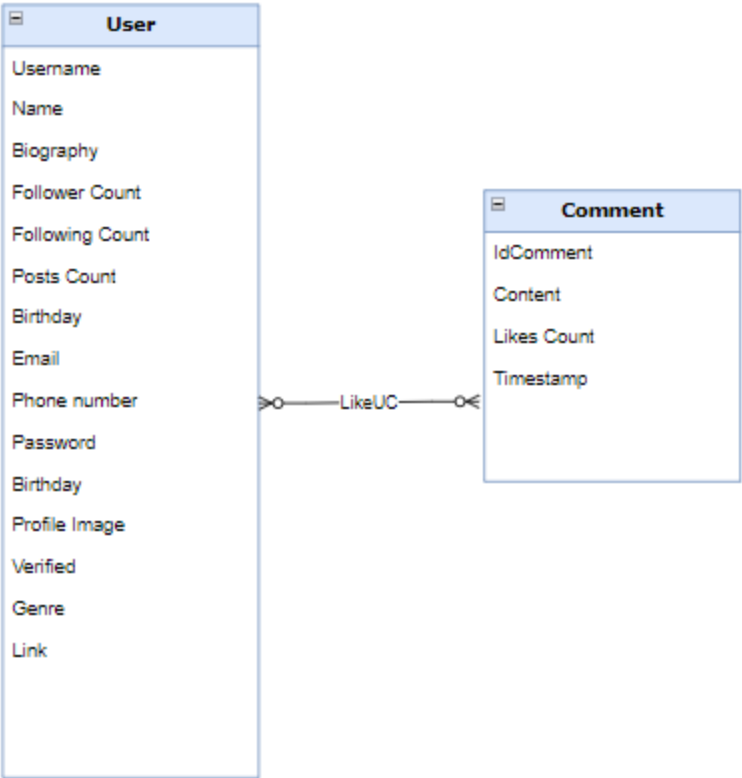
User - TagUS – Story



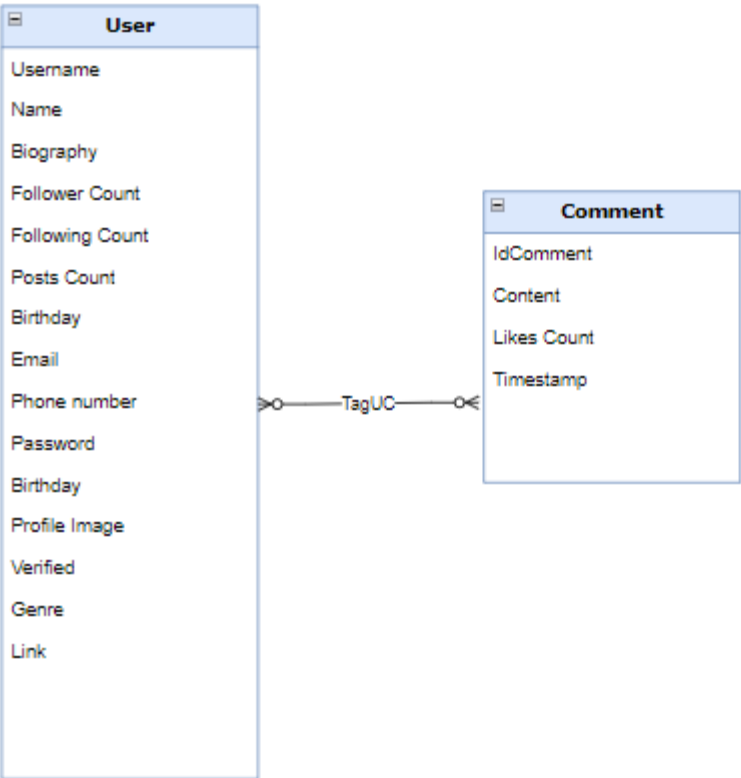
User - OwnUC – Comment



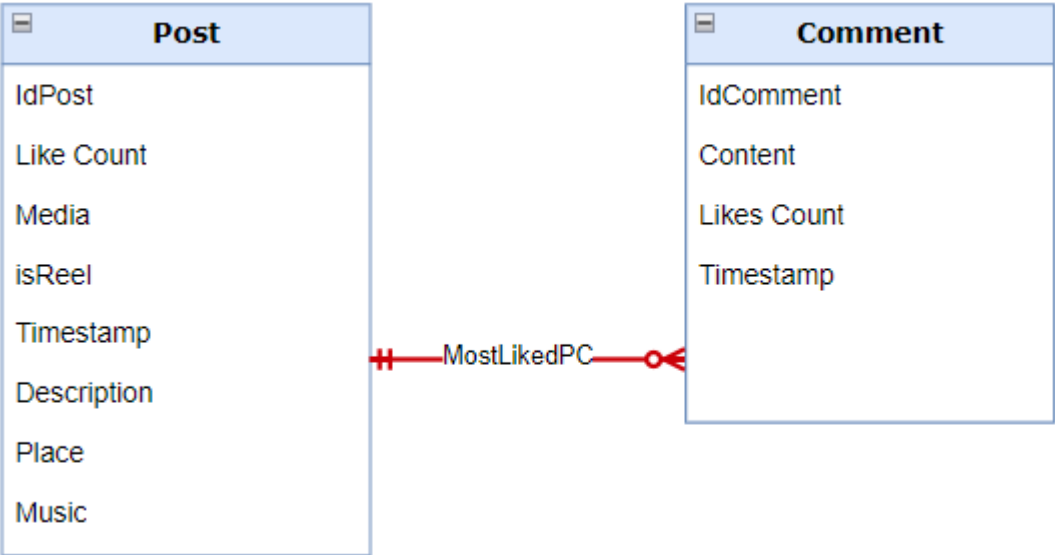
User - LikeUC – Comment



User - TagUc – Comment



Post - MostLikedPC – Comment



Progettazione degli aggregati

Tenendo in considerazione che ogni aggregato dovrebbe rappresentare un concetto ben definito all'interno del contesto applicativo e che dovrebbe essere grande abbastanza da includere i dati richieste dalle operazioni rilevanti, sono stati definiti i confini già riportati in precedenza.

Per ognuno di essi, segue un'analisi volta a valutare quale forma di aggregazione sia la migliore, quindi si procede con la definizione dell'entità chiave dell'aggregato valutando anche la possibilità di inserire dei collegamenti senza dover aggregare l'intero oggetto dentro l'entità chiave.

Nel caso in cui la relazione protagonista del confine di aggregazione abbia cardinalità 1-N e l'entità dal lato N della relationship sia debole, si valuta il solo verso di aggregazione naturale. Si definisce debole un'entità i cui dati solitamente non vengono acceduti direttamente, ma ha senso di esistere solo in relazione alla relativa entità forte.

Nei casi in cui l'entità Hashtag sia presente nel confine dell'aggregato, non si valuta la considerazione di Hashtag come entità chiave in quanto le istanze degli oggetti che dovrebbero essere aggregati in essa crescono in maniera incontrollata.

Si consideri l'assunzione della disponibilità di implementare degli accessi indicizzati rispetto alle chiavi. I calcoli che riguardano situazioni di collegamento sono stati effettuati considerando ciò. Anticipando una scelta descritta in seguito nella relazione il database sarà realizzato come document store in MongoDB, di conseguenza tale assunzione verrà attuata in fase implementativa.

Tabelle di analisi degli aggregati

In capo ad ogni analisi viene riportato il nome della relazione protagonista dell'aggregazione. In verde viene evidenziato la soluzione più conveniente valutando costi, occupazione memoria, struttura degli aggregati.

User - GetUN - Notify

N° medio Notify per User	30				
N° medio User per Notify	1				
Op/tipo aggregato	di	Frequenza giornaliera	Descrizione	Notify in User	K(Notify) in User
Op.7		1,40E+07	Memorizza una nuova Notify	1S*14.000.000	2S*14.000.000
				2,80E+07	5,60E+07
Op.43		6,00E+09	Visualizza tutte le Notify di uno User	1L*6.000.000.000	(1L+30L)*6.000.000.000
				6,00E+09	1,86E+11
			Totale	6,03E+09	1,86E+11

Hashtag - IncludeHC - Comment

N° medio hashtag per comment	0,5				
N° medio comment per hashtag	1,4				
Op/tipo aggregato	di	Frequenza giornaliera	Descrizione	Hashtag in Comment	K(Hashtag) in Comment
Op.2		3,00E+07	Memorizza nuovo Comment	1S*30.000.000	1S*30.000.000
				6,00E+07	6,00E+07
Op.13		1,20E+08	Aggiornamento Comment	1S*30.000.000	1S*30.000.000
				2,40E+08	2,40E+08

Op.17	1,60E+07	Memorizza nuovo Hashtag incluso in un Comment	1S*1,4*16.000.000	(1S+1S*1,4)*16.000.000
			4,48E+07	7,68E+07
Op.61	6,00E+10	Visualizza Comment	1L*60.000.000.000	1L*60.000.000.000
			6,00E+10	6,00E+10
		Totale	3,45E+08	3,77E+08

Hashtag - IncludeHP - Post

N° medio hashtag per post	15				
N° medio post per hashtag	21				
Op/tipo aggregato	di	Frequenza giornaliera	Descrizione	Hashtag in Post	K(Hashtag) in Post
Op.4		1,60E+07	Memorizza nuovo Post	1S*16.000.000	1S*16.000.000
				3,20E+07	3,20E+07
Op.16		2,40E+08	Memorizza nuovo Hashtag incluso in un Post	1S*21*240.000.000	(1S+1S*21)*240.000.000
				1,01E+10	1,06E+10
Op.14		2,90E+09	Aggiornamento Post	1S*2.900.000.000	1S*2.900.000.000
				5,80E+09	5,80E+09
Op.47		2,00E+09	Visualizza Posts con un certo Hashtag	1L*70.000.000.000*2.000.000.000	15L*2.000.000.000
				1,40E+20	3,00E+10
			Totale	1,40E+20	4,64E+10

Hashtag - IncludeHS - Story

N° medio hashtag per story	5			
N° medio story per hashtag	350			
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Hashtag in Story	K(Hashtag) in Story
Op.6	1,60E+07	Memorizza nuova Story	1S*16.000.000	1S*16.000.000
			3,20E+07	3,20E+07
Op.15	5,00E+10	Aggiornamento Story	1S*50.000.000.000	1S*50.000.000.000
			1,00E+11	1,00E+11
Op.23	7,00E+09	Memorizza nuovo Hashtag incluso in una Story	1S*350*7.000.000.000	(1S+1S*350)*.7000.000.000
			4,90E+12	4,91E+12
Op.49	2,00E+09	Visualizza Stories con un certo Hashtag	1L*3.500.000.000.000*2.000.000.000	15L*2.000.000.000
			7,00E+21	1,05E+11
		Totale	7,00E+21	5,12E+12

User - Join - Chat

N° medio chat in uno user	75						
N° medio user in chat	2,5						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Chat in User	User in Chat	K(Chat) in User	K(User) in Chat	K(User) in Chat e K(Chat) in User
Op.9	1,60E+07	Memorizza nuova Chat	2,5S*16.000.000	1S*16.000.000	(1S+2,5S)*16.000.000	(1S)*16.000.000	(1S+2,5S)*16.000.000
			8,00E+07	3,20E+07	1,12E+08	3,20E+07	1,12E+08
Op.10	2,20E+07	Aggiornamento User	1S*22.000.000	(1L*60.000.000.000+75S)*22.000.000	1S*22.000.000	1S*22.000.000	1S*22.000.000
			4,40E+07	1,32E+18	8,80E+07	4,40E+07	8,80E+07
Op.11	4,00E+10	Aggiornamento Chat	(1L*2.000.000.000+1S*2,5)*40.000.000.000	1S*40.000.000.000	1S*40.000.000.000	1S*40.000.000.000	1S*40.000.000.000
			8,00E+19	8,00E+10	8,00E+10	8,00E+10	8,00E+10
Op.34	4,00E+09	Visualizza Chats di uno User	1L*4.000.000.000	1L*75*4.000.000.000	(1L+75L)*4.000.000.000	1L*75*4.000.000.000	1L*75*4.000.000.000
			4,00E+09	3,00E+11	3,04E+11	3,00E+11	3,00E+11
		Totale	8,00E+19	1,32E+18	3,84E+11	3,80E+11	3,80E+11

Direct - LatestInDC - Chat

N° medio ultimi direct in una chat	28			
N° medio chat in ultimi direct	1			
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Direct in Chat	K(Direct) in Chat
Op.41	3,00E+09	Visualizza ultimi Directs di una Chat	1L*3.000.000.000	(1L + 28L)* 3.000.000.000
			3,00E+09	8,70E+10
Op.3	4,00E+09	Memorizza nuovo Direct	(2S+1L)*4.000.000.000	(2S+1L)*4.000.000.000
			2,00E+10	2,00E+10
		Totale	2,30E+10	1,07E+11

Thread - LeatestIncludePT - Post

N° medio di ultimi post in un thread	9						
N° medio di thread in cui un post è tra i più recenti	0,5						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Post in Thread	K(Post) in Thread	Thread in Post	K(Thread) in Post	K(Thread) in Post e K(Post) in Thread
Op.38	6,00E+09	Visualizza Ultimi Post di un Thread	1L*6.000.000.000	(1L + 9L)*6.000.000.000	1L*0,5*6.000.000.000	(1L*9+1L)*6.000.000.000	(1L + 9L)*6.000.000.000
			6,00E+09	6,00E+10	3,00E+09	6,00E+10	6,00E+10
Op.30	8,00E+05	Aggiunta di un Post in un Thread	(2S+1L)*800.000	(2S+1L)*800.000	1S*800.000	1S*800.000	1S*800.000
			3,00E+10	3,00E+10	1,60E+06	1,60E+06	1,60E+06
Op.14	2,90E+09	Aggiornamento Post	0,5S*2.900.000.000	1S	1S*2.900.000.000	(1S)*800.001	(1S)*800.002
			2,90E+09	5,80E+09	5,80E+09	5,80E+09	5,80E+09
Op.12	1,40E+07	Aggiornamento Thread	1S*14.000.000	1S*14.000.000	(1L*70.000.000.000+1S)*14.000.000	1S*14.000.000	1S*14.000.000
			2,80E+07	2,80E+07	9,80E+17	2,80E+07	2,80E+07
		Totale	3,89E+10	9,58E+10	9,80E+17	6,58E+10	6,58E+10

User - OwnUT - Thread

N° medio thread posseduti da uno user	0,0014						
N° medio user che possiedono un thread	1						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Thread in User	K(Thread) in User	User in Thread	K(User) in Thread	K(User) in Thread e K(Thread) in User
Op.8	1,50E+03	Memorizza nuovo Thread	1S*0,0014*1500	(1S+1S*0,0014)*1500	1S*1.500	1S*1.500	(1S+1S*0,0014)*1500
			4,20E+00	3,00E+03	3,00E+03	3,00E+03	3,00E+03
Op.12	1,40E+07	Aggiornament o Thread	(1L*2.000.000.000+1S*1)*14.000.000	1S*14.000.000	1S*14.000	1S*14.000	1S*14.000
			2,80E+16	2,80E+07	2,80E+07	2,80E+07	2,80E+07
Op.10	2,20E+07	Aggiornament o User	1S*22.000.000	1S*22.000.000	(1S*1+1L*2.800.000)*22.000.000	1S*22.000.000	1S*22.000.000
			4,40E+07	4,40E+07	4,40E+07	4,40E+07	4,40E+07
Op.31	4,00E+09	Visualizza User	1L*4.000.000.000	1L*4.000.000.000	1L*2.800.000*4.000.000.000	1L*4.000.000.000	1L*4.000.000.000
			4,00E+09	4,00E+09	1,12E+16	4,00E+09	4,00E+09
Op.57	2,00E+08	Visualizza Thread posseduti da uno User	1L*200.000.000	(1L+0,0014*1L)*200.000.000	1L*1*200.000.000	1*1L*200.000.000	1*1L*200.000.000
			2,00E+08	2,80E+05	2,00E+08	2,00E+08	2,00E+08

Op.59	2,92E+11	Visualizza Thread	1L*2.000.000.000*292.000.000.000	1L*292.000.000.000	1L*292.000.000.000	1L*292.000.000.000	1L*292.000.000.000
			5,84E+20	2,92E+11	2,92E+11	2,92E+11	2,92E+11
		Totale	5,84E+20	2,96E+11	1,12E+16	2,96E+11	2,96E+11

User - Partecipazione - Thread

N° medio User che partecipano ad un Thread	5000						
N° medio Thread in cui partecipa uno User	7						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Thread in User (thread partecipati)	User in Thread	K(Thread) in User	K(User) in Thread	K(User) in Thread e K(Thread) in User
Op.10	2,20E+07	Aggiornamento User	1S*22.000.000	(1S*7+1L*2.800.000)*22.000.000	1S*22.000.000	1S*22.000.000	1S*22.000.000
			4,40E+07	6,16E+13	4,40E+07	4,40E+07	4,40E+07
Op.12	1,40E+07	Aggiornamento Thread	(1L*2.000.000.000+1S*5000)*14.000.000	1S*14.000.000	1S*14.000.000	1S*14.000.000	1S*14.000.000
			2,80E+16	2,80E+07	2,80E+07	2,80E+07	2,80E+07

Op.18	8,00E+06	User partecipa ad un nuovo Thread	1S*8.000.000	1S*8.000.000	1S*8.000.000	1S*8.000.000	1S*8.000.000
			1,60E+07	4,00E+07	1,60E+07	4,00E+07	4,00E+07
Op.56	4,00E+08	Visualizza User che partecipano ad un Thread	1L*5000*400.000.000	1L*400.000.000	(1L*5000+1L*5000)*400.000.000	(1L+5000L)*400.000.000	(1L+5000L)*400.000.000
			2,00E+12	4,00E+08	4,00E+12	2,00E+12	2,00E+12
Op.58	2,00E+08	Visualizza Thread a cui partecipa uno User	1L*200.000.000	7L*200.000.000	(1L+7L)*200.000.000	(1L*7)*200.000.000	(1L*7)*200.000.000
		Totale	2,80E+16	6,16E+13	4,00E+12	2,00E+12	2,00E+12

User - LatestOwnUP - Post

N° medio Post per user	3			
N° medio user per post	1			
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Post in User	K(Post) in User
Op.4	1,60E+07	Memorizza un nuovo Post	$(2S+1L)*16.000.000$	$2S*1*16.000.000$
			8,00E+07	6,40E+07
Op.14	2,90E+09	Aggiornamento post	$(1L*2.000.000.000+1S)*2.900.000.000$	$1S*2.900.000.000$
			5,80E+18	5,80E+09
Op.35	4,00E+09	Visualizza ultimi post user	$1L*4.000.000.000$	$4L*4.000.000.000$
			4,00E+09	1,60E+10
		Totale	5,80E+18	2,19E+10

User - LikeUP - Post

N° medio post a cui uno user mette like	1750						
N° medio di like per post	50						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Post in User	K(Post) in User	User in Post	K(User) in Post	K(Post) in User e K(User) in Post
Op.10	4,57E+05	Aggiornamen to user	1S*457.000	1S*457.000	(1L*70.000.000.000 +1750*1S)*457.000	1S*457.000	1S*457.000
			9,14E+05	9,14E+05	3,20E+16	9,14E+05	9,14E+05
Op.14	2,90E+09	Aggiornamen to post	(1L*2.000.000. 000+50*1S)*2. 900.000.000	1S*2.900.000.000	1S*2.900.000.000	1S*2.900.000.000	1S*2.900.000.000
			5,80E+18	5,80E+09	5,80E+09	5,80E+09	5,80E+09
Op.19	8,00E+08	User aggiunge Like ad un Post	50*1S*800.00 0.000	1S*800.000.000	1S*800.000.000	1S*800.000.000	2S*800.000.000
			8,00E+10	1,60E+09	1,60E+09	1,60E+09	3,20E+09
Op.34	3,00E+10	Visualizza post	50L*30.000.00 0.000	50L*30.000.000.0 00	1L*30.000.000.000	1L*30.000.000.000	1L*30.000.000.000
			1,50E+12	1,50E+12	3,00E+10	3,00E+10	3,00E+10
		Totale	5,80E+18	1,51E+12	3,20E+16	3,74E+10	3,90E+10

User - TagUP - Post

N° medio post a cui uno user è taggato	35						
N° medio di tag per post	1						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Post in User	K(Post) in User	User in Post	K(User) in Post	K(Post) in User e K(User) in Post
Op.10	4,57E+05	Aggiornamen to user	1S*457.000	1S*457.000	(1L*70.000.000.000+3 5*1S)*457.000	1S*457.000	1S*457.000
			9,14E+05	9,14E+05	3,20E+16	9,14E+05	9,14E+05
Op.14	2,90E+09	Aggiornamen to post	(1L*2.000.00 0.000+1S)*2. 900.000.000	1S*2.900.000.000	1S*2.900.000.000	1S*2.900.000.000	1S*2.900.000.000
			5,80E+18	5,80E+09	5,80E+09	5,80E+09	5,80E+09
Op.20	1,60E+07	Tag di uno user su un post	1S*16.000.0 00	1S*16.000.000	1S*16.000.000	1S*16.000.000	2S*16.000.000
			3,20E+07	3,20E+07	3,20E+07	3,20E+07	3,20E+07
Op.34	3,00E+10	Visualizza post	1L*30.000.00 0.000	(1L+35L)*30.000. 000.000	1L*35*30.000.000.000	35L*30.000.000.000	(1L+35L)*30.000.000.000
			3,00E+10	1,08E+12	1,05E+12	1,05E+12	1,08E+12
		Totale	5,80E+18	1,09E+12	3,20E+16	1,06E+12	1,09E+12

User - LatestOwnUS - Story

N° medio ultime Story possedute da uno user	20 (di cui 5 in evidenza)			
N° medio user che possiedono una Story	1			
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Story in User	K(Story) in User
Op.6	1,40E+09	Memorizza nuova Story	$(2S+1L)*1.400.000.000$	$(2S+1L)*1.400.000.000$
			7,00E+09	7,00E+09
Op.15	5,00E+10	Aggiornamento Story	$(1L*2.000.000.000 + 1S*1)*50.000.000.000$	$1S*50.000.000.000$
			1,00E+20	1,00E+11
Op.10	2,20E+07	Aggiornamento User	$1S*22.000.000$	$1S*2.000.000$
			4,40E+07	4,40E+07
Op.50	4,00E+10	Visualizza le Story di uno User	$1L*40.000.000.000$	$(1L+20L)*40.000.000.000$
			4,00E+10	8,40E+11
Op.52	4,00E+09	Visualizza Story in evidenza	$1L*40.000.000.000$	$(1L+5L)*40.000.000.000$
			4,00E+09	2,40E+10
		Totale	1,00E+20	9,71E+11

User - LikeUS - Story

N° medio User a cui piace una Story	5						
N° medio Story piaciute ad un User	8750						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Story in User (Story piaciute)	User in Story	K(Story) in User	K(User) in Story	K(User) in Story e K(Story) in User
Op.15	5,00E+10	Aggiornamento Story	$(1L*2.000.000.000+1S*5)*50.000.000.000$	$1S*50.000.000.000$	$1S*50.000.000.000$	$1S*50.000.000.000$	$1S*50.000.000.000$
			1,00E+20	1,00E+11	1,00E+11	1,00E+11	1,00E+11
Op.10	2,20E+07	Aggiornamento User	$1S*22.000.000$	$(1L*3.500.000.000.000+1S*8750)*22.000.000$	$1S*22.000.000$	$1S*22.000.000$	$1S*22.000.000$
			4,40E+07	7,70E+19	4,40E+07	4,40E+07	4,40E+07
Op.21	7,00E+09	User aggiunge Like ad una Story	$5S*7.000.000.000$	$1S*7.000.000.000$	$5S*7.000.000.000$	$1S*7.000.000.000$	$1S*7.000.000.000$
			7,00E+10	1,40E+10	7,00E+10	1,40E+10	1,40E+10
Op.55	6,00E+09	Visualizza User che hanno messo like ad una Story	$5L*6.000.000.000$	$1L*6.000.000.000$	$(5L+5L)*6.000.000.000$	$(1L+5L)*6.000.000.000$	$(1L+5L)*6.000.000.000$
			3,00E+10	6,00E+09	6,00E+10	3,60E+10	3,60E+10
		Totale	1,00E+20	7,70E+19	2,30E+11	1,50E+11	1,50E+11

User - ReactionUS - Story

N° medio User che hanno reagito ad una Story	5						
N° medio Story piaciute ad un User	8750						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Story in User (Story con reazione)	User in Story	K(Story) in User	K(User) in Story	K(User) in Story e K(Story) in User
Op.15	5,00E+10	Aggiornamento Story	$(1L*2.000.000.000+1S*5)*50.000.000.000$	$1S*50.000.000.000$	$1S*50.000.000.000$	$1S*50.000.000.000$	$1S*50.000.000.000$
			1,00E+20	1,00E+11	1,00E+11	1,00E+11	1,00E+11
Op.10	2,20E+07	Aggiornamento User	$1S*22.000.000$	$(1L*3.500.000.000.000+1S*8750)*22.000.000$	$1S*22.000.000$	$1S*22.000.000$	$1S*22.000.000$
			4,40E+07	7,70E+19	4,40E+07	4,40E+07	4,40E+07
Op.24	7,00E+09	Reaction di una Story	$(2S+1L)*7.000.000.000$	$1S*7.000.000.000$	$1S*7.000.000.000$	$1S*7.000.000.000$	$1S*7.000.000.000$
			3,50E+10	1,40E+10	1,40E+10	1,40E+10	1,40E+10
Op.54	2,00E+09	Visualizza tutte le Reaction su una Story	$5L*2.000.000.000$	$1L*2.000.000.000$	$(5L+5L)*2.000.000.000$	$(1L+5L)*2.000.000.000$	$(1L+5L)*2.000.000.000$
			1,00E+10	2,00E+09	2,00E+10	1,20E+10	1,20E+10
		Totale	1,00E+20	7,70E+19	1,34E+11	1,26E+11	1,26E+11

User - TagUS - Story

N° medio User taggati in una Story	1						
N° medio Story in cui è taggato un User	1750						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Story in User (Story taggate)	User in Story (User taggati)	K(Story) in User	K(User) in Story	K(User) in Story e K(Story) in User
Op.15	5,00E+10	Aggiornamento Story	$(1L \cdot 2.000.000.000 + 1S \cdot 1) \cdot 50.000.000.000$	$1S \cdot 50.000.000.000$	$1S \cdot 50.000.000.000$	$1S \cdot 50.000.000.000$	$1S \cdot 50.000.000.000$
			1,00E+20	1,00E+11	1,00E+11	1,00E+11	1,00E+11
Op.10	2,20E+07	Aggiornamento User	$1S \cdot 22.000.000$	$(1L \cdot 3.500.000.000 + 1S \cdot 1750) \cdot 22.000.000$	$1S \cdot 22.000.000$	$1S \cdot 22.000.000$	$1S \cdot 22.000.000$
			4,40E+07	7,70E+19	4,40E+07	4,40E+07	4,40E+07
Op.22	1,40E+09	Tag di un User su una Story	$1S \cdot 1 \cdot 1.400.000.000$	$1S \cdot 1.400.000.000$	$1S \cdot 1 \cdot 1.400.000.000$	$1S \cdot 1.400.000.000$	$1S \cdot 1 \cdot 1.400.000.000$
			2,80E+09	2,80E+09	2,80E+09	2,80E+09	2,80E+09
Op.53	1,00E+09	Visualizza User taggati su una Story	$1L \cdot 1 \cdot 1.000.000.000$	$1L \cdot 1.000.000.000$	$(1L + 1L \cdot 1) \cdot 1.000.000.000$	$(1L + 1L) \cdot 1.000.000.000$	$(1L + 1L) \cdot 1.000.000.000$
			1,00E+09	1,00E+09	2,00E+09	2,00E+09	2,00E+09
		Totale	1,00E+20	7,70E+19	1,05E+11	1,05E+11	1,05E+11

User - OwnUC - Comment

N° medio commenti posseduti da uno user	70						
N° medio user che possiedono un commento	1						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Comment in User	K(Comment) in User	User in Comment	K(User) in Comment	K(User) in Comment e K(Comment) in User
Op.2	3,20E+07	Memorizza nuovo Comment	1S*1*32.000.000	2S*1*32.000.000	1S*32.000.000	1S*32.000.000	(2S)*32.000.000
			6,40E+07	1,28E+08	6,40E+07	6,40E+07	1,28E+08
Op.13	1,15E+08	Aggiornamen to Comment	1S*1*32.000.000	1S*1*32.000.001	1S*1*32.000.002	1S*1*32.000.003	1S*1*32.000.004
			6,40E+07	6,40E+07	6,40E+07	6,40E+07	6,40E+07
Op.61	6,00E+10	Visualizza Comment	1L*2.000.000.000.000*60.000.000.000	1L*60.000.000.000	1L*60.000.000.000	1L*60.000.000.000	1L*60.000.000.000
			1,20E+23	6,00E+10	6,00E+10	6,00E+10	6,00E+10
Op.31	4,00E+09	Visualizza User	1L*4.000.000.000	1L*4.000.000.000	1L*140.000.000.000*4.000.000	1L*4.000.000.000	1L*4.000.000.000
			4,00E+09	4,00E+09	5,60E+20	4,00E+09	4,00E+09
Op.10	2,20E+07	Aggiornamen to User	1S*22.000.000	1S*22.000.000	(1L*140.000.000.000+1S)*70*22.000.000	1S*22.000.000	1S*22.000.000
			4,40E+07	4,40E+07	2,16E+20	4,40E+07	4,40E+07
		Totale	1,20E+23	6,42E+10	7,76E+20	6,42E+10	6,42E+10

User - LikeUC - Comment

N° medio commenti in cui uno user ha messo like	14						
N° medio di Like in un commento	0,2						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Comment (commenti piaciuti) in User	K(Comment) in User	User in Comment	K(User) in Comment	K(User) in Comment e K(Comment) in User
Op.2	3,20E+07	Memorizza nuovo Comment	$0,25 \cdot 1S \cdot 32.000.000$	$(1S + 0,25S) \cdot 32.000.000$	$1S \cdot 32.000.000$	$1S \cdot 32.000.000$	$(1S + 0,25S) \cdot 32.000.000$
			1,60E+07	8,00E+07	6,40E+07	6,40E+07	8,00E+07
Op.10	2,20E+07	Aggiornamen to User	$1S \cdot 22.000.000$	$1S \cdot 22.000.000$	$(1L \cdot 140.000.000.000 + 14 \cdot 1S) \cdot 22.000.000$	$1S \cdot 22.000.000$	$1S \cdot 22.000.000$
			4,40E+07	4,40E+07	3,08E+18	4,40E+07	4,40E+07
Op.13	1,15E+08	Aggiornamen to Comment	$0,25 \cdot 1S \cdot 115.000.000$	$1S \cdot 115.000.000$	$1S \cdot 115.000.000$	$1S \cdot 115.000.000$	$1S \cdot 115.000.000$
			5,75E+07	2,30E+08	2,30E+08	2,30E+08	2,30E+08
Op.60	8,00E+07	User aggiunge like ad un Comment	$1S \cdot 80.000.000$	$2S \cdot 80.000.000$	$1S \cdot 80.000.000$	$1S \cdot 80.000.000$	$2S \cdot 80.000.000$
			1,60E+08	3,20E+08	1,60E+08	1,60E+08	3,20E+08
Op.61	6,00E+10	Visualizza Comment	$1L \cdot 0,2 \cdot 2.000.000.000.000$	$1L \cdot 60.000.000.000$	$1L \cdot 60.000.000.000$	$1L \cdot 60.000.000.000$	$1L \cdot 60.000.000.000$
			2,40E+22	6,00E+10	6,00E+10	6,00E+10	6,00E+10
		Totale	2,40E+22	6,07E+10	3,08E+18	6,05E+10	6,07E+10

User - TagUC - Comment

N° medio commenti in cui uno user è taggato	14						
N° medio di tag in un commento	0,2						
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Comment in User	K(Comment) in User	User in Comment	K(User) in Comment	K(User) in Comment e K(Comment) in User
Op.2	3,20E+07	Memorizza nuovo Comment	0,25*1S*32.000.000	(1S+0,25S)*32.000.000	1S*32.000.000	1S*32.000.000	(1S+0,25S)*32.000.000
			1,60E+07	8,00E+07	6,40E+07	6,40E+07	8,00E+07
Op.10	2,20E+07	Aggiornament o User	1S*22.000.000	1S*22.000.000	(1L*140.000.000.000+14*1S)*22.000.000	1S*22.000.000	1S*22.000.000
			4,40E+07	4,40E+07	3,08E+18	4,40E+07	4,40E+07
Op.13	1,15E+08	Aggiornament o Comment	0,25*1S*115.000.000	1S*115.000.000	1S*115.000.000	1S*115.000.000	1S*115.000.000
			5,75E+07	2,30E+08	2,30E+08	2,30E+08	2,30E+08
Op.61	6,00E+10	Visualizza Comment	1L*0,2*2.000.000.000*60.000.000.000	1L*60.000.000.000	1L*60.000.000.000	1L*60.000.000.000	1L*60.000.000.000
			2,40E+22	6,00E+10	6,00E+10	6,00E+10	6,00E+10
		Totale	2,40E+22	6,04E+10	3,08E+18	6,03E+10	6,04E+10

Post - MostLikedPC - Comment

N° medio Post per un Comment che ha più like	1			
N° medio Comment con più mi piace nel Post	10			
Op/tipo di aggregato	Frequenza giornaliera	Descrizione	Comment in Post	K(Comment) in Post
Op.45	9,00E+09	Visualizza Comments con più like in un Post	1L*9.000.000.000	(1L + 10L)*9.000.000.000
			9,00E+09	9,90E+10
Op.13	1,20E+08	Aggiorna Comment	1S*120.000.000	(1L+1S)*120.000.000
Op.26	3,00E+07	Risposta ad un Comment	1S*30.000.000	(1L+1S)*30.000.000
		Totale	9,24E+09	9,94E+10

Come è possibile notare osservando le tabelle appena riportate, non vi è una crescita notevole nei costi dovuti ai collegamenti bidirezionali, difatti, il contesto applicativo non prevede operazioni che portino al cambiamento delle chiavi dei singoli oggetti (o per lo meno hanno una frequenza notevolmente inferiore alle altre operazioni).

Con l'ottenimento dei risultati appena descritti si può considerare come conclusa la progettazione degli aggregati, si può procedere dunque con la relativa rappresentazione.

Rappresentazione e Partitioning degli Aggregati

NoAM

Una volta conclusa la progettazione degli aggregati, di questi ne è stata definita una rappresentazione astratta. Il modello utilizzato in questa fase prende il nome di NoAM (NoSQL Abstract Data Model), il quale ha la peculiarità di essere "system-independent" ossia non è legato a nessun modello NoSQL. Per definizione un database modellato in NoAM è un insieme di collezioni, ognuna delle quali ha un nome distinto. Una collezione è un insieme di blocchi (identificati da una chiave di blocco): un insieme non vuoto di entries. Ogni entry è una coppia (k, val), dove "k" è la chiave della entry e "val" è il suo valore che può essere semplice o complesso. Nella pratica la rappresentazione segue questo schema:

- Lo schema dell'aggregato (entità con il confine) definisce una collezione NoAM che deriva il nome dall'entità radice dell'aggregato;
- All'interno della collezione ogni istanza dell'aggregato è definita da un blocco in cui la chiave sarà l'identificatore dell'entità radice. Le entries saranno un insieme di attributi dell'entità radice e di quella annidata. Vi sono diverse alternative per l'organizzazione del blocco (esempio: EAO ed ETF)

Al fine di individuare la migliore alternativa per organizzare un blocco è stata effettuata la fase di "aggregate partitioning". In sintesi, questa procedura considera i pattern di accesso, relativamente alle operazioni, per individuare la modalità più conveniente. Un "access path" viene definito come segue: *access path ap=p1.p2...pn* dove *pi* rappresenta il valore di una componente strutturata che costituisce parte del blocco. Definiti tali pattern, per definire cosa mantenere in una singola entry oppure no vengono mutuati gli stessi principi della progettazione logica:

- Due o più elementi dovrebbero appartenere alla stessa entry se frequentemente acceduti insieme;
- Due o più elementi dovrebbero appartenere a entry separate se tipicamente acceduti separatamente.

Per il database in questione, il risultato delle fasi di rappresentazione e "partitioning" sono le collezioni NoAM riportate nelle sezioni di seguito.

Thread

Nella collezione seguente è stata definita una entry “€” con gli attributi dell’entità radice che vengono acceduti insieme. Sono state definite altre due entries singole: “PostsCount” e “UsersCount”; sono sempre attributi dell’entità Thread ma essendo acceduti/modificati singolarmente vengono mantenuti in entries differenti. Infine, vi sono: “LatestIncludePt”, “Partecipation” e “IncludePt” le quali conterranno nel primo caso l’aggregato vero e proprio degli ultimi Post inclusi (part. orizzontale) e negli ultimi due le chiavi (collegamenti) rispettivamente degli User che partecipano al Thread e di tutti i Post inclusi (strategia ETF).

idThread	€	{Title: string, Category: string, Description: string, Timestamp: timestamp}
	OwnUT	K(User)
	PostsCount	int
	UsersCount	int
	LatestIncludePt	{Post, ... ,Post}
	Partecipation	{K(User), ... , K(User)}
	IncludePt	{K(Post), ... , K(Post)}

User

Nella collezione seguente è stata definita una entry “€” con gli attributi dell’entità radice che vengono acceduti insieme. L’entry “OwnUT” esprime la chiave dello User proprietario del Thread. Sono state definite altre tre enties singole per lo stesso principio descritto sopra: “PostsCount”, “FollowerCount” e “FollowingCount”. Successivamente vengono definite tante entries quanti gli aggregati, in tutti i casi si tratta di collegamenti (strategia ETF). Questo vale anche per gli aggregati definiti sulle relazioni ricorsive “Blocked”, “Follow” e “Close Friend”.

UserName	€	{Name: string, Biography: string, Birthday: date, Email: string, PhoneNumber: string, Password: string, ProfileImage: string, Verified: boolean, Genre: string, Link: string}
	PostsCount	int
	FollowerCount	int
	FollowingCount	int
	OwnUS	{K(Story), ... , K(Story)}
	ExpiredOwnUS	{K(Story), ... , K(Story)}
	OwnUT	{K(Thread), ... , K(Thread)}
	LatestOwnUP	{K(Post), ... , K(Post)}
	OwnUP	{K(Post), ... , K(Post)}
	OwnUC	{K(Comment), ... , K(Comment)}
	Notify	{{Sender: K(User), Category: string, Content: string}, ... , {Sender: K(User), Category: string, Content: string}}
	Blocked	{K(User), ... , K(User)}
	Follow	{K(User), ... , K(User)}
	Close Friend	{K(User), ... , K(User)}

Chat

Nella collezione seguente è stata definita una entry “€” con tutti gli attributi dell’entità radice, essendo che in Chat non erano presenti attributi acceduti/modificati singolarmente dalle operazioni. Le ultime due entries sono gli aggregati “Join” (collegamenti) e “LatestInDC” (part. orizzontale) (strategia ETF).

idChat	€	{IsGroup: bool, Name: string, Timestamp: timestamp}
	Join	{K(User), ... , K(User)}
	LatestInDC	{Direct, ... , Direct}

Direct

Nella collezione seguente è stata definita una entry “€” con tutti gli attributi dell’entità Direct e un aggregato “InDC” contenente il riferimento alla Chat in cui è presente il Direct (collegamento).

idDirect	€	{Content: string, Timestamp: timestamp}
	InDC	K(Chat)

Story

Nella collezione seguente è stata definita una entry “€” con tutti gli attributi dell’entità eccetto “LikesCount” e “LikeUS” (acceduti/modificati singolarmente). L’entry “OwnUS” contiene la chiave dello User proprietario della Story. Gli aggregati “LikeUS”, “TagUS”, “ReactionUS” e “IncludeSH” sono dei collegamenti ognuno contenente delle liste di entità.

idStory	€	{Media: string, Restricted: bool, Timestamp: timestamp, Place: string, Music: string, Evidence: bool}
	OwnUS	K(User)
	LikesCount	int
	LikeUS	{K(User), ... , K(User)}
	TagUS	{K(User), ... , K(User)}
	ReactionUS	{K(User), ... , K(User)}
	IncludeSH	{K(Hashtag), ... , K(Hashtag)}

Hashtag:

Nella collezione seguente è stata definita una entry per ogni attributo dell’entità Hashtag, che però non vengono mai usati contemporaneamente. Nella collezione non sono presenti aggregati.

idHashtag	Text	string
	PostsCount	int
	CommentsCount	int
	StoriesCount	int

Post

Nella collezione seguente è stata definita una entry “€” con tutti gli attributi dell’entità ad eccezione di “LikesCount”. L’entry “OwnUP” contiene la chiave dello User proprietario del Post. L’unico aggregato “vero e proprio” è “MostLiked” che include la lista dei Comment con più like (partiz. orizzontale) mentre gli altri aggregati sono dei collegamenti ad altre entità.

idPost	€	{Media: string, IsReel: bool, Timestamp: timestamp, Description: string, Place: string, Music: string}
	OwnUP	K(User)
	LikesCount	int
	MostLiked	{Comment, ... , Comment}
	IncludePH	{K(Hashtag), ... , K(Hashtag)}
	LikeUP	{K(User), ... , K(User)}
	TagUP	{K(User), ... , K(User)}

Comment

Nella collezione seguente è stata definita una entry “€” con tutti gli attributi dell’entità eccetto “LikesCounts”. L’entry “OwnUC” contiene la chiave dello User proprietario del Comment. Gli aggregati sono esclusivamente dei collegamenti (anche per la relazione ricorsiva “Reply”).

idComment	€	{Content: string, Timestamp: timestamp}
	OwnUC	K(User)
	LikeUC	{K(User), ... , K(User)}
	TagUC	{K(User), ... , K(User)}
	IncludePC	K(Post)
	LikesCount	int
	IncludeCH	{K(Hashtag), ... , K(Hashtag)}
	Reply	{K(Comment), ... , K(Comment)}

Access Pattern

Come anticipato per arrivare a questo risultato sono stati studiati preliminarmente gli access pattern relativi alle operazioni. Di seguito sono stati riportati tali pattern raggruppati per entità radice:

User

K(Story) in User [OwnUS]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.6	Memorizza nuova Story	User.OwnUS
Op.10	Aggiornamento User	User
Op.50	Visualizza le Story di uno User	User.OwnUS
Op.52	Visualizza Story in evidenza	User.OwnUS.Evidence

K(Post) in User [LatestOwnUP]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.4	Memorizza un nuovo Post	User.LatestOwnUP
Op.35	Visualizza ultimi Post di uno User	User.LatestOwnUP

K(Thread) in User [OwnUT]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.8	Memorizza nuovo Thread	User.OwnUT
OpOp.31	Visualizza User	User
Op.57	Visualizza Thread posseduti da uno User	User.OwnUT

Notify in User [GetUN]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.7	Memorizza una nuova Notify	User.GetUN
Op.43	Visualizza tutte le Notify di uno User	User.GetUN

Altre operazioni inerenti all'entità radice sono:

Op/tipo di aggregato	Descrizione	Access Pattern
Op.1	Memorizza nuovo User	User
Op.27	Blocco di uno User	User.Blocked
Op.28	Follow di uno User	User.Follow
Op.29	Aggiunta di uno User nei Close Friends	User.Close Friend

Op.32	Visualizza User che seguono	User.Follow
Op.33	Visualizza User seguiti	User.Follow
Op.36	Visualizza tutti i Post di uno User	User.OwnUP

Thread

Post in Thread [LatestIncludePT]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.38	Visualizza Ultimi Post di un Thread	Thread.LatestIncludePT
Op.30	Aggiunta di un Post in un Thread	Thread.Post
Op.14	Aggiornamento Post	Thread.Post

K(User) in Thread [Partecipation]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.12	Aggiornamento Thread	Thread
Op.18	User partecipa ad un nuovo Thread	Thread.partecipation
Op.56	Visualizza User che partecipano ad un Thread	Thread.partecipation
Op.58	Visualizza Thread a cui partecipa uno User	Thread.partecipation

Altre operazioni inerenti all'entità radice sono:

Op/tipo di aggregato	Descrizione	Access Pattern
Op.8	Memorizza nuovo Thread	Thread
Op.59	Visualizza Thread	Thread
Op.39	Visualizza tutti i Posts di uno Thread	Thread.IncludePT
Op.57	Visualizza Thread posseduti da uno User	Thread

Chat

K(User) in Chat [Join]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.9	Memorizza nuova Chat	Chat
Op.11	Aggiornamento Chat	Chat
Op.34	Visualizza Chats di uno User	Chat.Join

Direct in Chat [LatestInDC]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.36	Visualizza ultimi Directs di una Chat	Chat.LatestInDC
Op.3	Memorizza nuovo Direct	Chat.Direct

Story

K(User) in Story [TagUS]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.15	Aggiornamento Story	Story
Op.22	Tag di un User su una Story	Story.TagUS
Op.53	Visualizza User taggati su una Story	Story.TagUS

K(User) in Story [LikeUS]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.15	Aggiornamento Story	Story
Op.21	User aggiunge Like ad una Story	Story.LikeUS
Op.55	Visualizza User che hanno messo like ad una Story	Story.LikeUS

K(User) in Story [ReactionUS]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.15	Aggiornamento Story	Story
Op.24	Reaction di una Story	Story.ReactionUS
Op.54	Visualizza Reaction su una Story	Story.ReactionUS

K(Hashtag) in Story [IncludeSH]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.6	Memorizza nuova Story	Story
Op.15	Aggiornamento Story	Story
Op.23	Memorizza nuovo Hashtag incluso in una Story	Story.IncludeSH
Op.49	Visualizza Stories con un certo Hashtag	Story.IncludeSH

Post

K(Hashtag) in Post [Include]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.4	Memorizza nuovo Post	Post
Op.16	Memorizza nuovo Hashtag incluso in un Post	Post.Include
Op.14	Aggiornamento Post	Post
Op.47	Visualizza Posts con un certo Hashtag	Post.Include

K(User) in Post (LikeUP)		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.14	Aggiornamento post	Post
Op.19	User aggiunge Like ad un Post	Post.LikeUP
Op.34	Visualizza post	Post

K(User) in Post (TagUP)		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.14	Aggiornamento post	Post
Op.20	Tag di uno user su un post	Post.TagUP
Op.34	Visualizza post	Post

Comment in Post [MostLiked]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.45	Visualizza Comments con più like in un Post	Post.MostLiked
Op.13	Aggiorna Comment	Post.MostLiked

Altre operazioni inerenti all'entità radice sono:

Op/tipo di aggregato	Descrizione	Access Pattern
Op.39	Visualizza tutti i Posts di uno Thread	Post

Comment

K(User) in Comment [OwnUC]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.2	Memorizza nuovo Comment	Comment
Op.13	Aggiornamento Comment	Comment
Op.61	Visualizza Comment	Comment

K(User) in Comment [TagUC]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.2	Memorizza nuovo Comment	Comment
Op.13	Aggiornamento Comment	Comment
Op.61	Visualizza Comment	Comment

Hashtag in Comment [IncludeCH]		
Op/tipo di aggregato	Descrizione	Access Pattern
Op.2	Memorizza nuovo Comment	Comment
Op.13	Aggiornamento Comment	Comment
Op.17	Memorizza nuovo Hashtag incluso in un Comment	Comment.IncludeCH
Op.61	Visualizza Comment	Comment

Altre operazioni inerenti all'entità radice sono:

Op/tipo di aggregato	Descrizione	Access Pattern
Op.26	Risposta ad un Comment	Comment.Reply
Op.46	Visualizza Comments in risposta ad un Comment	Comment.Reply

Hashtag

Op/tipo di aggregato	Descrizione	Access Pattern
Op.5	Memorizza nuovo Hashtag	Hashtag
Op.16	Memorizzo nuovo Hashtag incluso in un Post	Hashtag
Op.17	Memorizzo nuovo Hashtag incluso in un Comment	Hashtag
Op.23	Memorizza nuovo Hashtag incluso in una Story	Hashtag

Direct

Op/tipo di aggregato	Descrizione	Access Pattern
Op. 42	Visualizza tutti i Directs di una Chat	Direct.InDC

Le ultime due rappresentazioni descrivono due aggregati composti da una singola entità, rispettivamente Hashtag e Direct.

Non è presente un aggregato che prevede Notify come entità radice in quanto essa è riportata all'interno dell'aggregato User.

Traduzione dello schema NoAM

La traduzione dallo schema NoAM al document store su cui si basa MongoDB avviene nella modalità seguente:

- le collezioni in NoAM vengono mappate in collezioni di MongoDB;
- ogni blocco NoAM viene mappato in un documento di MongoDB;
- ogni entry del documento viene definita tramite una serializzazione in JSON/BSON del valore complesso o semplice della entry NoAM

Le collezioni ottenute da questo processo di traduzione sono quelle riportate di seguito.

Collection Story

```
{
  "idStory": "S1",
  "detailsS": {
    "media": "url",
    "restricted": false,
    "timestamp": 1694267071,
    "place": "url",
    "music": "url",
    "evidence": true
  },
  "ownUS": "U1",
  "likesCount": 2,
  "likeUS": ["U2", "U3"],
  "tagUS": ["U2"],
  "reactionUS": ["U3"],
  "includeSH": ["H1", "H2"]
}
```

Collection User

```
{
  "username": "U1",
  "detailsU": {
    "name": "Mario Rossi",
    "biography": "Studente Ingegneria Informatica e Automazione - UnivPM",
    "birthday": "01/01/1998",
    "email": "mario.rossi@email.com",
    "phoneNumber": "+39 3330947592",
    "password": "password123",
    "profileImage": "url",
    "verified": false,
    "genre": "male",
    "link": "url"
  },
  "postsCount": 4,
  "followerCount": 123,
  "followingCount": 123,
  "latestOwnUS": ["S1"],
  "ownUS": ["S1", "S2"],
  "ownUT": ["T1"],
  "latestOwnUP": ["P1", "P2", "P3"],
  "ownUP": ["P4"],
  "notify": [
    {
      "sender": "U2",
      "category": "likeP",
      "content": "A U2 piace il tuo post"
    },
    {
      "sender": "U2",
      "category": "likeS",
      "content": "A U2 piace la tua storia"
    },
    {
      "sender": "U2",
      "category": "follow",
      "content": "U2 ha iniziato a seguirti"
    }
  ],
  "blocked": ["U4"],
  "follow": ["U2", "U3"],
  "closeFriend": ["U2"]
}
```

Collection Direct

```
{
  "idDirect": "D1",
  "detailsD": {
    "content": "Ciao, come state?",
    "timestamp": 1694267071
  },
  "inDC": "C1"
}
```


Collection Thread

```
{
  "idThread": "T1",
  "detailsT": {
    "title": "NoSQL",
    "category": "tech",
    "description": "Approfondimento NoSQL",
    "timestamp": 1694267071
  },
  "ownUT": "U1",
  "postscountT": "gold",
  "usersCountT": 1234,
  "partecipation": ["U1", "U2", "U3"],
  "includePt": ["P1", "P2"],
  "latestIncludePT": [
    {
      "idPost": "P1",
      "detailsP": {
        "media": "url",
        "isReel": false,
        "timestamp": 1694267071,
        "description": "Questo è un bel post",
        "place": "url",
        "music": "url"
      },
      "ownUP": "U1",
      "likesCountP": 2,
      "includePH": ["H1", "H2", "H3"],
      "likeUP": ["P2", "P3"],
      "tagUP": ["P2"],
      "mostLiked": [
        {
          "idComment": "C1",
          "detailsC": {
            "content": "Bel post!",
            "timestamp": 1694267071
          },
          "ownUC": "U2",
          "includePC": "P1",
          "likesCountC": 1234,
          "includeCH": ["H1", "H2"],
          "reply": ["C2"]
        },
        {
          "idComment": "C2",
          "detailsC": {
            "content": "Grazie",
            "timestamp": 1694267071
          },
          "ownUC": "U1",
          "includePC": "P1",
          "likesCountC": 123,
          "includeCH": [],
          "reply": []
        }
      ]
    }
  ]
}
```

Collection Post

```
{
  "idPost": "P1",
  "detailsP": {
    "media": "url",
    "isReel": false,
    "timestamp": 1694267071,
    "description": "Questo è un bel post",
    "place": "url",
    "music": "url"
  },
  "ownUP": "U1",
  "likesCountP": 2,
  "includePH": ["H1", "H2", "H3"],
  "likeUP": ["P2", "P3"],
  "tagUP": ["P2"],
  "mostLiked": [
    {
      "idComment": "C1",
      "detailsC": {
        "content": "Bel post!",
        "timestamp": 1694267071
      },
      "ownUC": "U2",
      "includePC": "P1",
      "likesCountC": 1234,
      "includeCH": ["H1", "H2"],
      "reply": ["C2"]
    },
    {
      "idComment": "C2",
      "detailsC": {
        "content": "Grazie",
        "timestamp": 1694267071
      },
      "ownUC": "U1",
      "includePC": "P1",
      "likesCountC": 123,
      "includeCH": [],
      "reply": []
    }
  ]
}
```

Collection Hashtag

```
{
  "idHashtag": "H1",
  "text": "nosql",
  "postsCountH": 1234,
  "commentsCountH": 1234,
  "storiesCountH": 1234
}
```

Collection Comment

```
{
  "idComment": "C1",
  "detailsC": {
    "content": "Bel post!",
    "timestamp": 1694267071
  },
  "ownUC": "U2",
  "tagUC": [],
  "likeUC": ["U1"],
  "includePC": "P1",
  "likesCountC": 1,
  "includeCH": ["H1", "H2"],
  "reply": ["C2"]
}
```

Collection Chat

```
{
  "idChat": "C1",
  "detailsC": {
    "isGroup": true,
    "name": "Gruppo studio NoSQL",
    "timestamp": 1694267071
  },
  "join": ["U1", "U2", "U3"],
  "latestInDC": [
    {
      "idDirect": "D1",
      "detailsD": {
        "content": "Ciao, come state?",
        "timestamp": 1694267071
      },
      "inDC": "C1"
    },
    {
      "idDirect": "D2",
      "detailsD": {
        "content": "Alla grande!",
        "timestamp": 1694267071
      },
      "inDC": "C1"
    },
    {
      "idDirect": "D3",
      "detailsD": {
        "content": "Bene!",
        "timestamp": 1694267071
      },
      "inDC": "C1"
    }
  ]
}
```

Implementazione

La base di dati oggetto di tale elaborato, a seguito della progettazione appena descritta, è stata implementata all'interno del DBMS non relazionale, basato sul modello NoSQL document store, MongoDB.

Attraverso MongoDB Compass è stata instaurata una connessione verso un la porta 27017 del localhost su cui è in esecuzione il servizio di MongoDBServer. Fatto ciò, è stato creato un nuovo database denominato "Instagram", all'interno del quale, con la funziona "create Collection", sono state implementate le collezioni descritte nel capitolo precedente.

Al fine di popolare il suddetto database, è stata sviluppato uno script in Python in grado di creare dei documenti JSON che garantiscono un buon livello di coerenza rispetto ai dati in essi contenuti. Lo script appena citato è disponibile nel repository GitHub https://github.com/giordanoangelini/InstagramNoSQL_NGDB_2023.

Query

Implementato e popolato il database, sono state condotte le query associate alle operazioni più interessanti tra quelle riportate nella "Tabella delle Operazioni".

Di seguito sono visibili i risultati delle singoli query, associati alla descrizione dell'operazione stessa.

User aggiunge Like ad un post (Op. 19)

```
db.Post.updateOne(  
  {  
    idPost:"U0P0"  
  },  
  {  
    $inc: {  
      likesCountP: 1  
    },  
    $push: {  
      likeUP:"U1"  
    }  
  }  
)
```

In figura è riportata la conferma dell'esecuzione della query appena decritta. Nello specifico si aggiorna la lista degli User che hanno aggiunto Like al Post e si incrementa il likeCount. Come è visibile, viene aggiornato l'unico post corrispondente all'identificativo indicato nella query.

```
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

Visualizzazione di tutti gli User taggati in una Story (Op. 53)

```
db.Story.aggregate([
  { '$match': { idStory: 'U0S7' } },
  {
    '$lookup': {
      from: 'User',
      localField: 'tagUS',
      foreignField: 'username',
      as: 'result'
    }
  },
  { '$project': { result: 1, idStory: 1 } }
])
```

Nella figura seguente sono riportati i risultati della query, per garantire una maggiore flessibilità non sono stati espansi i singoli array.

```
_id: ObjectId('6500650b2d6a9eb140a47c0d')
idStory: "U0S7"
▼ result: Array
  ▼ 0: Object
    _id: ObjectId('650065032d6a9eb140a47bb3')
    username: "U18"
    ▶ detailsU: Object
    postsCount: 5
    followerCount: 44
    followingCount: 16
    ▶ latestOwnUS: Array
    ▶ ownUS: Array
    ▶ ownUT: Array
    ▶ latestOwnUP: Array
    ▶ ownUP: Array
    ▶ notify: Array
    ▶ blocked: Array
    ▶ follow: Array
    ▶ closeFriend: Array
  ▼ 1: Object
    _id: ObjectId('650065032d6a9eb140a47bfe')
    username: "U93"
    ▶ detailsU: Object
    postsCount: 7
    followerCount: 47
    followingCount: 47
    ▶ latestOwnUS: Array
    ▶ ownUS: Array
    ▶ ownUT: Array
    ▶ latestOwnUP: Array
    ▶ ownUP: Array
    ▶ notify: Array
    ▶ blocked: Array
    ▶ follow: Array
    ▶ closeFriend: Array
```

Visualizza tutti i Direct di una Chat

```
db.Chat.aggregate([
  [
    { '$match': { idChat: 'C1' } },
    {
      '$lookup': {
        from: 'Direct',
        localField: 'idChat',
        foreignField: 'inDC',
        as: 'direct_of_chat'
      }
    }
  ]
])
```

Per visualizzare tutti i Direct occorre effettuare la lettura di latestInDc per ottenere gli ultimi messaggi e in seguito di tutti i Direct con il collegamento alla Chat. Di seguito il risultato ottenuto eseguendo la query.

```
_id: ObjectId('6501ccf1e8a8a2be52f0cc00')
idChat: "C1"
▸ detailsC: Object
▸ join: Array
▾ latestInDC: Array
  ▸ 0: Object
  ▸ 1: Object
  ▸ 2: Object
  ▸ 3: Object
  ▸ 4: Object
  ▸ 5: Object
  ▸ 6: Object
  ▸ 7: Object
▸ direct_of_chat: Array
```

Inserimento di un Post in un Thread

Nel caso in cui la lista di “latestIncludePT”, avente gli ultimi Post associati al Thread, abbia raggiunto il numero massimo di Post, è necessario rimuovere l’ultimo Post dalla lista creando un nuovo oggetto nella collection Post; infine occorre aggiungere il nuovo Post.

Rimozione ultimo Post e aggiunta nuovo Post.

```
db.Thread.updateOne(  
  {  
    idThread: "T0"  
  },  
  {  
    $pop: {  
      latestIncludePT: -1  
    },  
    $push: {  
      latestIncludePT: {<ObjectPost>}  
    }  
  }  
)
```

In figura si vede il risultato di questa prima query.

```
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

Recupero dell’oggetto di cui si è fatta la post e creazione nuovo documento

```
db.post.insertOne(  
  db.thread.findOne(  
    {  
      idThread: "T0"  
    }  
  )  
  ["latestIncludePT"].slice(-1)[0]  
)
```

Il risultato di questa seconda query è il seguente.

```
{  
  acknowledged: true,  
  insertedId: ObjectId("6501e57ce341c52d40cad417")  
}
```