UNIVERSITÀ POLITECNICA DELLE MARCHE

DIPARTIMENTO DI INGEGNERIA DELL' INFORMAZIONE

Corso di Laurea Magistrale in

Ingegneria Informatica e dell'Automazione

PROGETTAZIONE ED IMPLEMENTAZIONE DI UNA BASE DI DATI NOSQL PER L'APPLICAZIONE MULTIMEDIALE SNAPCHAT

Autori:

Baio Antonio *1108645*

Dediu Razvan Alexandru *1108807*

Gatti Giada *1108648*

A.A. 2022/2023

Indice

Progettazione concettuale e logica

La progettazione concettuale di un database consiste nella costruzione di uno schema E-R in grado di descrivere le specifiche sui dati di una applicazione. Lo sviluppo di uno schema concettuale può essere fatto sia per database relazionali (SQL) che per database "di nuova generazione" come noSQL. Infatti, il modello concettuale non dipende dal modello dati utilizzato.

1.1 Analisi dei requisiti

L'obiettivo del progetto è sviluppare un database noSQL, utilizzando il DBMS non relazionale MongoDB, che possa essere la base per un possibile sviluppo del noto social network Snapchat.

1.1.1 Requisiti espressi in linguaggio naturale

L'applicazione permette di creare un account tramite l'utilizzo di un numero di telefono valido, una e-mail non precedentemente utilizzata e uno username univoco non ancora presente nel database.
Dopo aver inserito una coppia di credenziali valida (e-mail e password), all'utente viene proposta oltre che la creazione di un avatar, anche la creazione una lista di contatti iscritti a Snapchat e presenti nella rubrica dell'utente.
Durante la creazione dell'avatar, l'utente può scegliere tra diverse configurazioni dell'aspetto e dell'abbigliamento da assegnargli.
A seguito della registrazione, l'utente può:

- Aggiungere alla lista dei propri amici anche persone non presenti nei contatti personali. Infatti, vengono mostrati dei suggerimenti in base agli amici già presenti nella lista stessa.
- Creare uno *Snap*, ovvero uno scatto o un video da pubblicare. Lo Snap rimane visibile per 24 ore a partire dalla pubblicazione.
- Visualizzare l'archivio degli Snap pubblicati precedentemente e scaduti.
- Visualizzare su una mappa (sezione *Map*) luoghi di interesse, amici o Snap creati nelle vicinanze e lungo tutto il territorio italiano.
- Iniziare conversazioni con utenti (sezione *Chat*). Per iniziare una conversazione è necessario effettuare la richiesta di amicizia.
- Visualizzare le *Stories* sia degli utenti più influenti sulla piattaforma, che della propria lista di amici (nella sezione *Stories*).

Inoltre è possibile mettere *like* alle storie degli utenti e riceverne per le proprie stories.
L'inserimento della posizione dell'utente sulla mappa è consentito solo dopo l'approvazione dell'utente stesso. Inoltre è possibile attivare le seguenti modalità:

- *Ghost Mode*, che permette di nascondere la propria posizione a tutti o solo ad un numero ristretto di amici.
- *Hide my live location*, che permette di nascondere la propria posizione in tempo reale a tutti gli amici.

Inoltre la mappa può essere visualizzata attraverso due grafiche differenti: modalità *Hotspot* o modalità *Satellite*.

1.1.2 Glossario dei termini

Termine	**Descrizione**	**Sinonimo**	**Collegamento**
Snap	Scatto o video effettuato dall'utente. Viene chiamato anche "storia".	Story	Stories, Spotlight
Snapchat	Applicazione multimediale e social network.	-	Applicazione
Stories	Insieme di snap effettuati dall'utente	Snaps	Pubblicazione, Snap, Spotlight
Database	Collezione di dati organizzati	Storage, Banca dati	-
Chat	Collegamento con un amico per effettuare uno scambio di messaggi.	Messaggi	-
Spotlight	Insieme di stories in riproduzione casuale.	-	Stories, Snap
Map	Mappa su cui vengono mostrati luoghi, amici e snap passati.	-	Ghost Mode, Satellite, Hotspot, HMLL
Amici	Lista di persone a cui l'utente ha deciso di richiedere il collegamento.	(Lista di) Chat	Chat
E-mail	E-mail con cui l'utente si registra.	-	Account, Password
Password	Stringa con cui l'utente può accedere al proprio account.	Key	Account, E-mail
Account	Profilo dell'utente, creato dopo la registrazione.	Profilo	E-mail, Password, Registrazione

Registrazione	Form con cui l'utente crea un nuovo account	-	Account
Like	Reazione di un utente allo snap di un altro utente	-	Account, Snap
Pubblicazione	Si intende la condivisione sulla piattaforma di uno snap	-	Snap, Stories
Sounds	Effetto che permette di aggiungere musica su uno snap	Effetto	Snap, Stories
Green Screen	Effetto che permette di modificare lo sfondo di uno snap	Effetto	Snap, Stories
Dual Camera	Effetto che permette di scattare uno snap con la fotocamera anteriore e posteriore. Permette di avere entrambe le immagini in una unica story	Effetto	Snap, Stories
Filters	Modifica del volto dell'utente	-	Snap, Stories
Ghost Mode	Modalità in cui l'utente restringe la visibilità della sua posizione a un certo numero di contatti	Nascondi posizione	Map,Account
Satellite	Specifica grafica con cui viene mostrata la mappa	-	Map, Account, Snap
Hotspot	Ulteriore grafica con cui può essere visualizzata la mappa	-	Map, Account, Snap
Hide my live location (HMLL)	Modalità con cui viene nascosta la posizone in tempo reale dell'utente	Nascondi posizione RT	Map, Account

1.2 Modello E-R

1.2.1 Notazione

Uno schema entità-relazione è composto da 3 costrutti fondamentali: entità, relazioni, attributi. Queste componenti possono essere rappresentate attraverso schemi con grafiche diverse; La notazione utilizzata per la realizzazione del nostro schema è chiamata *Crow's Foot*: utilizza dei rettangoli per indicare le entità che contengono gli attributi. La principale differenza tra le varie notazioni è la grafica con cui vengono rappresentate le relazioni e le cardinalità: di seguito viene mostrato uno schema riassuntivo della notazione utilizzata.

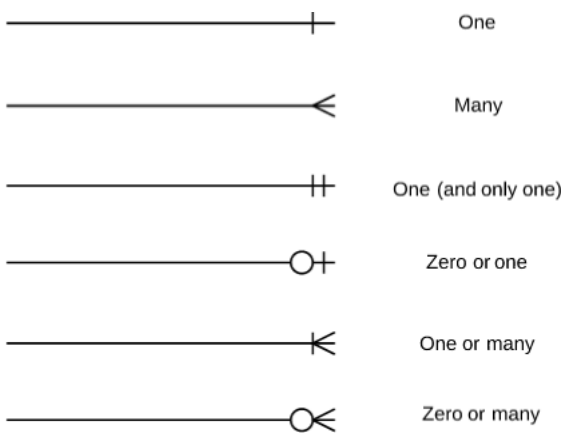

Figura 1.1: relazioni e cardinalità con crow's foot

1.2.2 Schema E-R

Per comprendere lo schema E-R è importante chiarire alcune funzionalità dell'applicazione in questione.

- Ogni utente può attivare e disattivare un account business, che permette all'utente di mostrare sulla mappa la propria attività commerciale, di ottenere like e di mostrare l'indirizzo e i dettagli del business.

- Per ogni chat possono essere generati diversi stickers: questi vengono aggiornati ogni volta che avviene un aggiornamento all'intyerno della chat (come l'invio di un messaggio). La generazione di stickers coinvolge anche l'avatar generato dall'utente durante la creazione del profilo.

- Per Snap si intende uno scatto o un video che l'utente decide di pubblicare. Con l'attributo *Location* in Snap si intende la posizione in cui è stato effettuato tale scatto.

- Per MapAsset si intende la configurazione che l'utente ha deciso di avere sulla mappa e l'insieme di Snap che verrano mostrati per ogni utente. Ad ogni utente viene associato un MapAsset e possono essere presenti più Snap da mostrare sulla mappa.
  Il valore della posizione dell'utente business o dello Snap dipende dal valore dell'attributo *statusGps*, che rappresenta il consenso dato dall'utente alla geolocalizzazione.

- È possibile avere un solo avatar per utente e questo può essere modificato durante l'utilizzo dell'app.

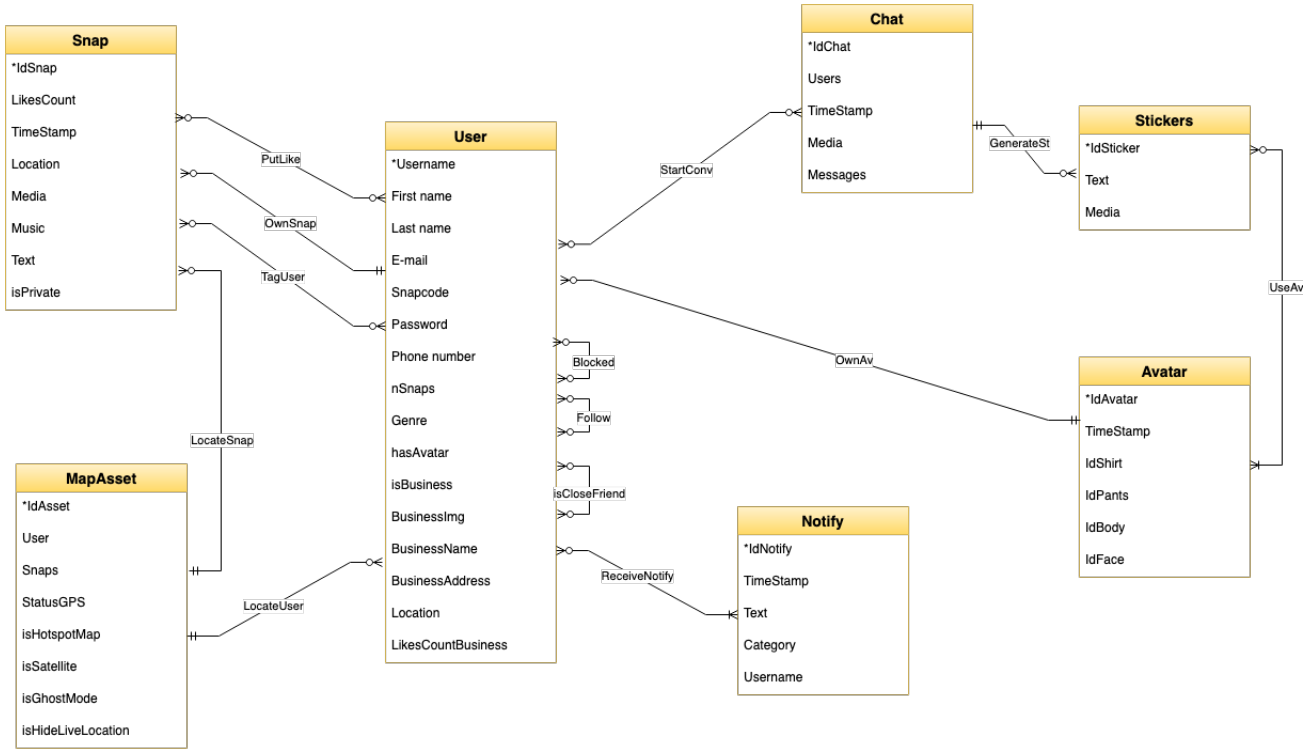

Figura 1.2: Modello E-R

1.3 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 network pari a 12 anni. Trattandosi di volumi molto grandi è stata adottata una notazione esponenziale per favorire la leggibilità.

Concetto	**Tipo**	**Volume**
User	E	5,00E+08
Avatar	E	5,00E+08
Snap	E	1,00E+09
Notify	E	1,00E+09
Chat	E	1,50E+09
Stickers	E	3,50E+09
MapAsset	E	7,50E+08

Concetto	**Tipo**	**Volume**
PutLike	R	1,50E+10
OwnSnap	R	5,00E+08
ReceiveNotify	R	2,00E+09
OwnAv	R	5,00E+08
UseAv	R	5,00E+08
StartConv	R	1,50E+09
GenerateSt	R	4,50E+09
Blocked	R	1,50E+09
Follow	R	1,00E+10
isCloseFriend	R	2,00E+10
LocateUser	R	5,00E+08
TagUser	R	2,50E+08
LocateSnap	R	1,00E+05

1.4 Tavola delle operazioni

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

Operazione	**Descrizione**	**Frequenza** [1]
Operazione 1	Memorizza nuovo user	1,20E+05
Operazione 2	Memorizza nuovo Snap	2,30E+05
Operazione 3	Memorizza nuova notifica	1,00E+08
Operazione 4	Memorizza nuova chat	1,00E+07
Operazione 5	Memorizza nuovo asset sulla mappa	3,50E+05
Operazione 6	Aggiornamento profilo user	2,00E+07
Operazione 7	Aggiornamento chat	2,50E+09
Operazione 8	Aggiornamento avatar	2,50E+05
Operazione 9	Aggiornamento Snap	1,00E+09
Operazione 10	Aggiornamento asset della mappa	1,00E+03
Operazione 11	Visualizza profilo di uno user	2,10E+08
Operazione 12	Visualizza notifiche di uno user	3,50E+05
Operazione 13	Visualizza lista di amici di uno user	2,10E+08
Operazione 14	Visualizza specifica chat di uno user	2,10E+08
Operazione 15	Visualizza intera lista di chat di uno user	1,20E+06
Operazione 16	Visualizza messaggi di una chat	2,00E+08
Operazione 17	Visualizza immagini inviate in una chat	1,50E+05

Operazione 18	Visualizza specifico snap di uno user	5,00E+08
Operazione 19	Visualizza ultimi 10 snap di uno user	1,00E+06
Operazione 20	Visualizza ultimi 5 memories (snap) di uno user sulla mappa	2,50E+05
Operazione 21	Visualizza intero archivio snap di uno user	1,20E+06
Operazione 22	Visualizza musica presente in uno snap	1,50E+05
Operazione 23	Visualizza posizione di uno snap	3,00E+06
Operazione 24	Visualizza user taggati in uno snap	4,60E+05
Operazione 25	Visualizza dettagli avatar	1,20E+05
Operazione 26	Aggiorna mappa in modalità spotlight	6,00E+08
Operazione 27	Aggiorna mappa in modalità satellite	5,00E+08
Operazione 28	Aggiorna lista memories di uno user sulla mappa	4,00E+06
Operazione 29	Aggiorna mappa in modalità ghostmode	2,50E+04
Operazione 30	Aggiorna mappa in modalità HLL	2,50E+04
Operazione 31	Aggiorna mappa in modalità hotspot	2,50E+04
Operazione 32	Aggiorna consenso geolocalizzazione	1,00E+05

Operazione 33	Visualizza like ai luoghi presenti sulla mappa (user business)	2,00E+09
Operazione 34	Visualizza luoghi (user business) sulla mappa	3,00E+08
Operazione 35	Visualizza utenti ghostmode	3,00E+05
Operazione 36	User aggiunge like ad uno snap	4,00E+08
Operazione 37	Inserisci tag di uno user su uno snap	3,50E+08
Operazione 38	Blocco di uno user	1,00E+09
Operazione 39	Aggiungi agli amici uno user	2,00E+09
Operazione 40	Aggiungi uno user agli amici stretti	1,00E+09
Operazione 41	Aggiungi snap nella lista di memories	1,00E+09
Operazione 42	Memorizza avatar	1,15E+05
Operazione 43	Memorizza nuovo sticker	2,00E+09
Operazione 44	Aggiornamento sticker	2,50E+09
Operazione 45	Visualizza sticker di una chat	2,00E+09

[1]si intende la frequenza/giorno

1.5 Analisi delle ridondanze

Per ridondanza si intende un dato che può essere derivato, attraverso una serie di operazioni, da altri dati. Gli attributi considerati possono far parte di una stessa entità (o relazioni) o possono essere di entità (o relazioni) diverse. Il costo delle operazioni di scrittura (**S**) viene considerato doppio rispetto al costo delle operazioni di lettura (**L**).

Si valuta l'attributo ***LikesCount*** presente nell'entità **Snap**.
Per valutare l'attributo si cosiderano i seguenti valori:

- Numero medio di like per snap: **15**
- Numero medio di snap a cui gli utenti mettono like: **3**

Operazione	**Frequenza** [1]	**Senza ridondanza**	**Con ridondanza**
Visualizza specifico snap di uno user [Op.18]	5,00E+08	(**1L**+**1L***15**)*5,00E+08	**1L***5,00E+08
Costo op. 18		8,00E+09	5,00E+08
User aggiunge like ad uno snap [Op.36]	4,00E+08	**1S***4,00E+08	(**1L**+**2S**)*4,00E+08
Costo op. 36		8,00E+08	2,00E+09
Costo totale		8,80E+09	2,50E+09

Si valuta l'attributo ***LikesCountBusiness*** presente nell'entità **MapAsset**.
Per valutare l'attributo si cosiderano i seguenti valori:

- Numero medio di like per luogo : **5**
- Numero medio di luoghi a cui gli utenti mettono like: **1**

Operazione	**Frequenza** [1]	**Senza ridondanza**	**Con ridondanza**

Visualizza like ai luoghi presenti sulla mappa (user business) [Op.33]	2,00E+09	(**1L+1L*5**)*2,00E+09	**1L***2,00E+09
Costo totale		1,20E+10	2,00E+09

Si valuta l'attributo ***nSnaps*** presente nell'entità **User**.

Per valutare l'attributo si cosiderano i seguenti valori:

- Numero medio di snap creati da un utente: **3**
- Numero medio di utenti per snap: **1**

Operazione	**Frequenza** [1]	**Senza ridondanza**	**Con ridondanza**
Memorizza nuovo snap [Op.2]	2,30E+05	(**1L** + **2S**)*2,30E+05	**1S***2,30E+05
Costo op. 2		1,15E+06	4,60E+05
Visualizza intero archivio snap di uno user [Op.21]	1,20E+06	(**1L+1L*3**)*1,20E+06	**1L***1,20E+06
Costo op. 21		4,80E+06	1,20E+06
Costo totale		5,95E+06	1,86E+06

[1] si intende la frequenza/giorno

1.6 Partizionamenti

1.6.1 Partizionamenti orizzontali

Per evitare l'utilizzo di un ampio numero di risorse, è possibile effettuare una ristrutturazione dello schema.
In particolare, viene applicato un partizionamento orizzontale alle relazioni che possono crescere in modo illimitato; si considerano le seguenti relazioni su cui applicare il partizionamento:

- User - **OwnSnap** - Snap
- Chat - **GenerateSt** - Sticker
- MapAsset - **LocateSnap** - Snap

L'obiettivo è creare una nuova relazione che indichi solo i dati più recenti.
La ristrutturazione porterà dunque ai seguenti risultati:

- User - **LatestOwnSnap** - Snap
- Chat - **LatestGenSticker** - Stickers
- MapAsset - **LatestSnap** - Snap

Le relazioni indicate sopra si riferiscono alle ultime 10 storie per utente per la prima relazione, agli ultimi 15 stickers generati per chat per la seconda, e agli ultimi 5 snap per l'ultima.
Le nuove relazioni verranno utilizzate nel capitolo successivo per la progettazione degli aggregati.

Progettazione degli aggregati

2.1 Definizione dei confini degli aggregati

Nel contesto di un database NoSQL come MongoDB, i "borders" o confini degli aggregati definiscono le limitazioni logiche che circondano un insieme di dati correlati. Gli aggregati raggruppano documenti con significato comune, consentendo di accedere e manipolare dati correlati in modo coeso. MongoDB utilizza chiavi di partizionamento e transazioni distribuite per garantire che gli aggregati siano isolati e consistente al loro interno, garantendo coerenza e atomicità per le operazioni che coinvolgono gli stessi dati all'interno di un aggregato.

2.2 Analisi degli aggregati

Stickers - UseAv - Avatar

- Numero medio di sticker per avatar: **3**
- Numero medio di avatar per sticker: **1**

Op./tipo di aggregato	**Frequenza**	**Stickers in Avatar**	**K(Stickers) in Avatar**
Memorizza nuovo Sticker [Op.43]	2,00E+09	**1S***2,00E+09	**2S***2,00E+09
Costo op. 43		4,00E+09	8,00E+09
Aggiornamento Sticker [Op.44]	2,50E+09	**1L*5,00E+08** + **1S***2,50E+09	**1S***2,50E+09
Costo op. 44		1,25E+18	5,00E+09
Costo totale		1,25E+18	1,30E+10

MapAsset - LatestSnap - Snap

- Numero medio di snap per asset presente sulla mappa: **5**
- Numero medio di Asset per Snap: **1**

Op./tipo di aggregato	**Frequenza**	**Snap in MapAsset**	**K(Snap) in MapAsset**
Visualizza ultimi 5 memories (snap) di uno user sulla mappa [Op.20]	2,50E+05	**1L*5** *2,50E+05	**1L+1L*5** *2,50E+05
Costo op. 20		1,25E+06	1,50E+06
Aggiornamento snap [Op.9]	1,00E+09	(**1L*7,50E+08**+ **1S*1**)* 1,00E+09	**1S***1,00E+09
Costo op.9		7,50E+17	2,00E+09
Aggiorna lista memories di uno user sulla mappa [Op. 28]	4,00E+06	(**1L*7,50E+08**)+ (**1S*1**) *4,00E+06	**2S***2,00E+09
Costo op. 28		7,50E+17	8,00E+09
Costo totale		1,50E+18	9,00E+09

Chat - LatestGenSticker - Stickers

- Numero medio di stickers per chat: **3**
- Numero medio di chat per stickers: **1**

Op./tipo di aggregato	**Frequenza**	**Stickers in Chat**	**K(Stickers) in Chat**
Visualizza specifica chat di uno user [Op.14]	2,10E+08	**1L***2,10E+08	**1L***2,10E+08
Costo op. 14		2,10E+08	2,10E+08
Visualizza stickers di una chat [Op.45]	2,00E+09	(**1L*3**)*2,00E+09	(**1L**+**1L*3**) *2,00E+09
Costo op.45		6,00E+09	8,00E+09
Aggiornamento chat [Op.7]	2,50E+09	**2S***2,50E+09	**2S***2,50E+09
Costo op. 7		1,00E+10	1,00E+10
Memorizza nuova chat [Op.4]	1,00E+07	**1S** *1,00E+07	**1S** *1,00E+07
Costo op.4		2,00E+07	2,00E+07
Costo totale		1,60E+10	1,80E+10

User - ReceiveNotify - Notify

- Numero medio Notify per User: **2**
- Numero medio User per Notify: **1**

Op./tipo di aggregato	**Frequenza**	**Notify in User**	**K(Notify) in User**
Memorizza nuova notifica[Op. 3]	1,00E+08	**1S***1,00E+08	**2S***1,00E+08
Costo op. 3		2,00E+08	4,00E+08
Visualizza notifiche di uno user[Op.12]	3,50E+05	**1L***3,50E+05	(**1L**+**2L**)*3,50E+05
Costo op. 12		3,50E+05	10,5E+05
Costo totale		2,00E+08	4,00E+08

User - LocateUser - MapAsset

- Numero medio Asset per User: **1**
- Numero medio User per Asset: **1**

Op./tipo di aggregato	**Frequenza**	**Map in User**	**K(Map) in User**	**User in Map**	**K(User) in Map**	**K(Map) in User e K(User) in Map**
Aggiorna asset della mappa [Op. 10]	1,00E+03	(**1L**+**1S**)*0.67 *1,00E+03	(**1L**+**1S**) *1,00E+03	**1S***1,00E+03	**1S***1,00E+03	(**1L**+**1S**) *1,00E+03
Costo op. 10		2,00E+03	3,00E+03	2,00E+03	2,00E+03	3,00E+03
Visualizza luoghi (user business) sulla mappa [Op.34]	3,00E+08	(**1L***5E+08) *3E+08	(**1L***5E+08) *3E+08	**1L** *3,00E+08	**1L***3,00E+08	(**1L***5E+08) *3E+08
Costo op. 34		15,00E+16	15,00E+16	3,00E+08	3,00E+08	15,00E+16

Visualizza like ai luoghi presenti sulla mappa (user business) [Op. 33]	2,00E+09	(**1L***5E+08) *2,00E+09	(**1L***5E+08) *2,00E+098	**1L** *2,00E+09	**1L***2,00E+09	(**1L***5E+08) *2,00E+09
Costo op. 33		15,00E+16	15,00E+16	3,00E+08	3,00E+08	15,00E+16
Costo totale		15,00E+16	15E+16	3,00E+08	3,00E+08	15E+16

User - Create - Avatar

- Numero medio Avatar per User: **1**
- Numero medio User per Avatar: **1**

Op./tipo di aggregato	**Frequenza**	**Avatar in User**	**K(Avatar) in User**
Aggiornamento avatar[Op. 8]	2,50E+05	**1S***2,50E+05	**2S***2,50E+05
Costo op. 8		5,00E+05	10,00E+05
Visualizza dettagli avatar[Op. 25]	1,20E+05	**1L***1,20E+05	**1L***1,20E+05
Costo op. 25		1,20E+05	1,20E+05
Creazione avatar[Op. 32]	1,15E+05	**2S***1,15E+05	**2S***1,15E+05
Costo op. 32		4,6E+05	4,6E+05
Costo totale		10,00E+05	16,00E+05

User - PutLike - Snap

- Numero medio di Snap a cui un utente mette like: **30**
- Numero medio di User che mettono like ad uno Snap: **15**

Op./tipo di aggregato	**Frequenza**	**Snap in User**	**K(Snap) in User**	**User in Snap**	**K(User) in Snap**	**K(User) in Snap e K(Snap) in User**
Aggiornamento profilo user [Op.6]	2,00E+07	**1S***2,00E+07	**1S***2,00E+07	(**1L*5,00E+08**)+ (**1S*30**)*2,00E+07	**1S***2,00E+07	**1S***2,00E+07
Costo op.6		4,00E+07	4,00E+07	6,00E+17	4,00E+07	a
User aggiunge like ad uno Snap [Op.36]	4,00E+08	(**1S*15**) *4,00E+08	**1S***4,00E+08	**1S***4,00E+08	**1S***4,00E+08	**1S***4,00E+08
Costo op.36		1,20E+10	8,00E+08	8,00E+08	8,00E+08	

Visualizza specifico Snap di un utente [Op.18]	5,00E+08	(**1L*15**) *5,00E+08	(**1L*15**) *5,00E+08	**1L** *5,00E+08	**1L** *5,00E+08	**1L+1L*15**
Costo op. 18		7,50E+09	7,50E+09	5,00E+08	5,00E+08	a
Aggiornamento Snap [Op.9]	1,00E+09	(**1L*1,00E+09**+ **1S*15**) *1,00E+09	(**1S**)*1,00E+09	**1S***1,00E+09	**1S***1,00E+09	**1S***1,00E+09
Costo op. 9		4,50E+10	1,00E+09	1,00E+09	1,00E+09	
Costo totale		6,50E+10	9,50E+09	6,00E+17	2,30E+09	

User - TagUser - Snap

- Numero medio di Snap a cui un utente mette like: **40**
- Numero medio di User che mettono like ad uno Snap: **1**

Op./tipo di aggregato	**Frequenza**	**Snap in User**	**K(Snap) in User**	**User in Snap**	**K(User) in Snap**	**K(User) in Snap e K(Snap) in User**
Aggiornamento profilo user [Op.6]	2,00E+07	**1S***2,00E+07	**1S***2,00E+07	(**1L*5,00E+08**)+ (**1S*40**)*2,00E+07	**1S***2,00E+07	**1S***2,00E+07
Costo op.6		4,00E+07	4,00E+07	6,00E+17	4,00E+07	a
Visualizza user taggati su uno snap [Op. 24]	4,60E+05	(**1L*1**) *4,60E+05	(**1L+1L*1**) *4,60E+05	**1L** *4,60E+05	**1L+1L** *4,60E+05	**1L+1L** *4,60E+05
Costo op.24		1,20E+10	8,00E+08	8,00E+08	8,00E+08	

Inserisci tag di uno user su uno snap [Op.37]	3,50E+08	**1S*1**)*3,50E+08	**1S***3,50E+08	**1S*1** *3,50E+08	**1S** *3,50E+08	**1S*1** *3,50E+08
Costo op. 37		7,50E+09	7,50E+09	5,00E+08	5,00E+08	a
Aggiornamento Snap [Op.9]	1,00E+09	(**1L*1,00E+09**+ **1S*1**) *1,00E+09	(**1S**)*1,00E+09	**1S***1,00E+09	**1S***1,00E+09	**1S***1,00E+09
Costo op.9		4,50E+10	1,00E+09	1,00E+09	1,00E+09	
Costo totale		6,50E+10	9,50E+09	6,00E+17	2,30E+09	

User - LatestOwnSnap - Snap

- Numero medio di User che possiedono uno Snap: **0,5**
- Numero medio di ultimi Snap posseduti da uno User: **2**
- Numero medio di Snap archiviati da uno User: **0,1**

Op./tipo di aggregato	**Frequenza**	**Snap in User**	**K(Snap) in User**
Aggiornamento profilo user [Op.6]	2,00E+07	(**1S**)*2,00E+07	(**1S**)*2,00E+07
Costo op.6		4,00E+07	4,00E+07
Memorizza nuovo Snap [Op.2]	2,30E+05	**1S***2,30E+05	(2S+1L)*2,30E+05
Costo op.2		4,60E+05	1,15E+06
Visualizza specifico Snap di un utente [Op.18]	5,00E+08	(**1L**)*5,00E+08	(**1L*2*2**)*5,00E+08
Costo op. 18		5,00E+08	2,00E+09
Visualizza intero archivio snap di uno user [Op. 21]	1,20E+06	(**1L*0,1**)*1,20E+06	(**2L*0,1**)*1,20E+06
Costo op.21		1,20E+05	2,40E+05
Aggiornamento Snap [Op.9]	1,00E+09	(**1L*5,00E+08** +**1S*1**) *1,00E+09	(**1S**)*1,00E+09
Costo op.9		5,00E+17	2,00E+09
Costo totale		5,00E+17	1,00E+09

Rappresentazione astratta degli aggregati

3.1 Modello NoAM

In questa fase viene utilizzato un modello di progettazione chiamato NoAM (noSQL Abstract Model), come modello intermedio tra gli aggregati e la base di dati NoSQL.
Il modello NoAM è *system-independent*, ovvero è indipendente dal tipo di sistema NoSQL utilizzato.
È composto da elementi chiamati *collections*, ognuna delle quali può contenere uno o più blocchi, a loro volta composti da un insieme non vuoto di *entries*. Ogni entry è una coppia chiave-valore, dove la chiave viene utilizzata per accedere al valore (semplice o composto) dell'entry stessa.
La rappresentazione degli aggregati può essere effettuata seguendo diverse strategie; per il nostro progetto è stata scelta la strategia **ETF**, che permette di rappresentare gli aggregati con multiple entries. Inoltre, per ogni campo non innestato dell' aggregato si definisce una entry che ha come chiave il nome del campo, e come valore il valore del campo.

User

La collection User è composta dalle seguenti entries:

- Username: rappresenta il nome dell'account scelto dall'utente in fase di registrazione.
- *Anagrafica*: composta da tutti gli attributi che descrivono l'utente e che vengono acceduti insieme.
- *BusinessInfo*: composta da tutti gli attributi che descrivono un profilo *business*. I dati vengono acceduti insieme.
- *hasAvatar*: valore booleano che indica se l'utente ha creato un avatar oppure no. A seconda del valore, il campo "Avatar" sarà popolato o meno.
- *Avatar*: composta da tutti gli attributi che descrivono l'avatar di un utente. Anche in questo caso gli attributi vengono acceduti insieme. L'entità Avatar presente nella progettazione concettuale è stata aggregata all'entità User, così da poter ridurre il numero di operazioni come osservato nell'analisi al capitolo precedente.
- *Notify*: L'entità Notify è stata accorpata all'entità User, come consigliato dai risultati dell'analisi degli aggregati. È composta da attributi che vengono acceduti insieme.
- *OwnSnaps* e *LatestOwnSnaps*: entry ottenute con il partizionamento effettuato nei capitoli precedenti.
- *Friends, CloseFriends, Blocked*: Indicano la lista di amici, di amici stretti e degli utenti bloccati di uno user.
- *nSnaps*: indica il numero di snaps creati da un utente.

Username	**Anagrafica**	{ **FirstName**: string, **LastName**: string, **Email**: string, **SnapCode**: int, **Password**: h(string), **Genre**: string, **PhoneNumber**: int, **Bday**: date}
	BusinessInfo	{ **isBusiness**: bool, **BImg**: string, **BName**: string, **BAddress**: string, **BLocation:** string, **BLikesCount**: int }
	nSnaps	int
	ownSnaps	{ **K(snap)**, **K(snap)**, ..., **K(snap)** }
	LatestOwnSnaps	{ **K(snap)**, **K(snap)**, ..., **K(snap)** }
	Notify	[{ **Sender**: K(user), **idNotify**: string, **Cat**: string, **Text**: string }, ..., { **Sender**: K(user), **idNotify**: string, **Cat**: string, **Text**: string }]
	Friends	{ **K(user)**, **K(user)**, ..., **K(user)**}
	CloseFriend	{ **K(user)**, **K(user)**, ..., **K(user)**}
	Blocked	{ **K(user)**, **K(user)**, ..., **K(user)**}
	hasAvatar	bool
	Avatar	{ **idFace**: string, **idBody**: string, **idShirt**: string, **idPants**: string, **TimeStamp**: string }

Map

La collection Map è formata da due valori composti. Per Snaps, si intende la lista (un array) di storie che hanno informazioni circa il luogo in cui quella storia è stata scattata. In ghostMode vengono invece riportati, se la funzione è attiva (bool = True), la lista di amici a cui nascondere la propria posizione.

IdAsset		
	User	**K(user)**
	Snaps	{ **K(snap)**, ..., **K(snap)** }
	isHotSpot	bool
	isSatellite	bool
	GhostMode	{ **isGhostMode**: bool, { **K(user)**, ..., **K(user)** } }
	isHideLiveLocation	bool
	StatusGps	bool

Sticker

La collection sticker contiene gli attributi per identificare il testo presente nello sticker; il *media*, ovvero l'immagine utilizzata per generare lo sticker; l'identificativo dell'avatar presente nella entità radice user e il *TimeStamp* che indica la data di creazione dello sticker.

IdSticker	**Text**	string
	media	string
	User	**K(user)**
	TimeStamp	string

Chat

In questa collection sono presenti i *messaggi* scambiati dai singoli all'interno di una chat. Ogni messaggio ha come identificativo la chiave dell'utente che lo ha inviato, il contenuto testuale e il timestamp relativo all'invio. Inoltre è presente un valore composto *media*, che indica i documenti o le immagini inviati in una chat.

IdChat	**Media**	{ **Docs**: {string, ..., string}, **Pics**: {string, ..., string} }
	Users	{ **K(user)**, **K(user)** }
	Messages	{ { **K(user)**, **Text**: string, **TS**: string, { **K(user)**, **Text**: string, **TS**: string } ..., { **K(user)**, **Text**: string, **TS**: string } }
	Stickers	{ **K(stickers)**, ..., **K(sticker)** }
	TimeStamp	string

Snap

La collection Snap rappresenta la singola storia pubblicata da un utente: gli attributi composti *Media* e *UsersLikes* indicano rispettivamente i testi, le immagini e la musica inseriti in uno snap e gli utenti che hanno messo like a quella storia. Inoltre, in TagUsers vengono riportati gli utenti taggati in quella storia.

IdSnap	**Media**	{ **Text**: string, **Pic**: string, **Music**: string }
	OwnSnap	**K(user)**
	UsersLikes	{ **K(user)**, **K(user)**, ..., **K(user)** }
	TagUsers	{ **K(user)**, **K(user)**, ..., **K(user)** }
	Location	string
	LikesCount	int
	IsPrivate	bool
	TimeStamp	string

Partizionamento degli aggregati

4.1 Pattern di accesso

Nella fase di *partizionamento degli aggregati* si vanno a considerare i **pattern di accesso**, definiti come una sequenza ap = p1, p2....pn, dove ogni pi identfica il valore di una componente strutturata. Nello sviluppo, si considerano le seguenti linee guida:

- Se un aggregato è di piccola dimensione e tutti o quasi i suoi dati sono modificati insieme, allora dovrebbe essere rappresentato da una singola entry.
- Al contrario, un aggregato dovrebbe essere partizionato in più entry se la sua dimensione è grande e le operazioni accedono o modificano di frequente solo una specifica porzione dell'aggregato.

User

Op./tipo di aggregato	**Access Pattern**
Visualizza user [Op.]	User
Memorizza nuovo user [Op.1]	User

Memorizza nuova notifica [Op.3]	User.Notify
Aggiornamento user [Op.5]	User
Visualizza profilo di uno user [Op.]	User.Anagrafica
Visualizza lista amici di uno user [Op.]	User.Friends
Visualizza notifiche di uno user [Op.]	User.Notify
Visualizza avatar [Op.]	User.Avatar
Creazione avatar [Op.]	User.Avatar
Follow di uno user [Op.]	User.Friends
Blocco di uno user [Op.]	User.Blocked
Aggiunta di uno user negli amici stretti [Op.]	User.CloseFriend
Aggiornamento avatar [Op.]	User.Avatar
Memorizza nuovo snap [Op.]	User.OwnSnap
Visualizza specifico snap di un utente [Op.]	User.OwnSnap
Visualizza ultimi 10 snap di un utente [Op.]	User.LatestOwnSnap
Visualizza intero archivio snap di uno user [Op.]	User.OwnSnap

Snap

Op./tipo di aggregato	**Access Pattern**
Memorizza nuovo snap [Op.]	Snap
Aggiornamento snap [Op.]	Snap
User aggiunge like ad uno snap [Op.]	Snap.LikesCount
Visualizza musica presente in uno snap [Op.]	Snap.Media.Music
Aggiorna lo snap a modalità privata [Op.]	Snap.IsPrivate
Visualizza posizione di uno snap [Op.]	Snap.Location
Visualizza user taggati su uno snap [Op.]	Snap.TagUsers
Aggiungi tag di uno user su uno snap[Op.]	Snap.TagUsers

Chat

Op./tipo di aggregato	**Access Pattern**
Memorizza nuova chat [Op.]	Chat
Visualizza lista chat di uno user [Op.]	Chat.Users
Visualizza messaggi di una chat [Op.]	Chat.Messages

Aggiornamento chat [Op.]	Chat
Visualizza messaggi di una chat [Op.]	Chat.Messages
Visualizza immagini inviare in una chat [Op.]	Chat.Media.Pics
Visualizza stickers di una chat [Op.]	Chat.Stickers

MapAsset

Op./tipo di aggregato	**Access Pattern**
Memorizza nuovo asset sulla mappa [Op.]	MapAsset
Aggiorna asset della mappa [Op.]	MapAsset
Aggiorna mappa in modalità spotlight [Op.]	Map.IsSpotlight
Aggiorna mappa con modalità satellite [Op.]	MapAsset.IsSatellite
Aggiorna mappa in modalità ghost mode [Op.]	MapAsset.GhostMode.IsGhostMode
Visualizza utenti ghost mode [Op.]	MapAsset.GhostMode
Aggiorna mappa in modalità HLL [Op.]	MapAsset.IsHideLiveLocation
Aggiorna mappa in modalità hotspot [Op.]	MapAsset.IsHostspot

Aggiorna consenso geolocalizzazione [Op.]	MapAsset.StatusGps
Visualizza utenti sulla mappa [Op.]	MapAsset.User
Visualizza ultimi 5 memories (snap) di uno user sulla mappa [Op.]	MapAsset.LatestSnaps
Aggiorna lista memories di uno user sulla mappa [Op.]	MapAsset.Snaps
Aggiungi snap nella lista di memories [Op.]	MapAsset.LatestSnaps

Sticker

Op./tipo di aggregato	**Access Pattern**
Memorizza nuovo sticker [Op.]	Sticker
Aggiornamento sticker [Op.]	Sticker

Traduzione del modello NoAM

5.1 Traduzione delle collection NoAM

In questo capitolo, le collection precedentemente rappresentate tramite modello NoAM vengono tradotte in collection del database: ogni blocco viene mappato in un documento, ed ogni entry viene serializzata in formato JSON.

Collection Snap

```
{
    "idSnap": "User0Snap0",
    "media": {
        "text": "Ape",
        "pic": "",
        "music": ""
    },
    "ownSnaps": "User0",
    "UserLikes": [
        "User78",
        "User60",
        "User9",
        "User45",
        ...
```

```
        ],
        "LikesCount": 77,
        "TagUsers": [
            "User36",
            "User42",
            "User69",
            ...
        ],
        "Location": "",
        "isPrivate": false,
        "Timestamp": 1699288283.644067
    },
    ...
```

Collection User


```
{
    "username": "User0",
    "Anagrafica": {
        "FirstName": "Juliana",
        "LastName": "Mckenzie",
        "email": "Juliana.Mckenzie.0@email.com",
        "snapCode": "29811247",
        "password": "password_Juliana",
        "genre": "male",
        "phoneNumber": "111111112",
        "birthday": "15/07/1973"
    },
    "BusinessInfo": {
        "isBusiness": false,
        "BusinessImage": "",
        "BusinessName": "",
        "BusinessAddress": "",
        "BusinessLocation": ""
    },
    "nSnaps": 26,
    "ownSnaps": [
        "User0Snap0",
        "User0Snap1",
        "User0Snap2",
        "User0Snap3",
        ...
    ],
    "LatestOwnSnaps": [
        ...
```

```
        "User0Snap23",
        "User0Snap24",
        "User0Snap25"
    ],
    "hasAvatar": false,
    "Avatar": {
        "idFace": "",
        "idBody": "",
        "idShirt": "",
        "idPants": "",
        "Timestamp": ""
    },
    "Notify": [
        {
            "Sender": "User54",
            "idNotify": "Notify1Snap",
            "Cat": "Snap",
            "Text": "New Snap"
        },
        ...
        {
            "Sender": "User97",
            "idNotify": "Notify5Message",
            "Cat": "Message",
            "Text": "New Message"
        }
    ],
    "Friends": [
        "User21",
        "User34",
        "User8",
```

```
            ...
        ],
        "CloseFriends": [
            "User21",
            "User29",
            ...
        ],
        "Blocked": [
            "User50",
            "User14",
            ...
        ]
    },
    {
        "username": "User1",
        "Anagrafica": {
            "FirstName": "Cecilia",
            "LastName": "Tyler",
            "email": "Cecilia.Tyler.1@email.com",
            "snapCode": "44338509",
            "password": "password_Cecilia",
            "genre": "female",
            "phoneNumber": "999999995",
            "birthday": "05/06/1993"
        },
        "BusinessInfo": {
            "isBusiness": true,
            "BusinessImage": "",
            "BusinessName": "UserBusiness1",
            "BusinessAddress": "Via della Libert\u00e0, n. 58",
            "BusinessLocation": "(-22.9068, -43.1729)"
```

```
},
"nSnaps": 21,
"ownSnaps": [
    "User1Snap0",
    "User1Snap1",
    "User1Snap2",
    "User1Snap3",
    ...
],
"LatestOwnSnaps": [
    ...
    "User1Snap19",
    "User1Snap20"
],
"hasAvatar": true,
"Avatar": {
    "idFace": "31134",
    "idBody": "69303",
    "idShirt": "83318",
    "idPants": "89253",
    "Timestamp": 1699288283.631531
},
"Notify": [
    {
        "Sender": "User71",
        "idNotify": "Notify6Request",
        "Cat": "Request",
        "Text": "New Request"
    },
    ...
    {
```

```
                "Sender": "User40",
                "idNotify": "Notify0Request",
                "Cat": "Request",
                "Text": "New Request"
            }
        ],
        "Friends": [
            "User91",
            "User58",
            "User2",
            ...
        ],
        "CloseFriends": [
            ...
            "User84",
            "User14",
            "User68"
        ],
        "Blocked": [
            "User0",
            "User29",
            ...
        ]
    }
```

Collection Chat


```
{
  "idChat": "C1",
  "Media": {
        "Pics": [],
        "Docs": [
          "doc0",
          "doc1",
          ...
        ]
  },
  "Users": [
    "User75",
    "User95"
  ],
  "Messages": [
    {
      "User": "User95",
      "Text": "Neve cade oggi",
      "Timestamp": 1699301792.177046
    },
    ...
    {
      "User": "User75",
      "Text": "Risate sono contagiose sempre",
      "Timestamp": 1699301792.177055
    }
  ],
  "Stickers": [
    {
      "idSticker": "Sticker20",
```

```
        "text": "Porta",
        "media": "animatedSticker",
        "idAvatar": "User95",
        "Timestamp": 1699301792.193243
      },
      ...
    ],
    "Timestamp": 1699301792.177084
  }
```

Collection Sticker


```
{
        "idSticker": "Sticker0",
        "text": "Giardino",
        "media": "staticSticker",
        "idAvatar": "User88",
        "Timestamp": 1699288283.782431
    },
    ...
    {
        "idSticker": "Sticker8",
        "text": "Specchio",
        "media": "staticSticker",
        "idAvatar": "User92",
        "Timestamp": 1699288283.782466
    }
```

Collection Map


```
{
        "idAsset": "570571",
        "User": "User0",
        "Snaps": [
            "User0Snap4",
            "User0Snap6",
            "User0Snap9",
            "User0Snap10",
            ...
        ],
        "StatusGPS": true,
        "isHotspot": false,
        "isSatellite": true,
        "isHideLiveLocation": true,
        "GhostMode": {
            "isGhostMode": false,
            "GhostModeFriends": []
        }
    },
    {
        "idAsset": "565432",
        "User": "User1",
        "Snaps": [
            "User1Snap0",
            "User1Snap1",
            "User1Snap5",
            "User1Snap6",
            ...
        ],
        "StatusGPS": true,
```

```
        "isHotspot": true,
        "isSatellite": true,
        "isHideLiveLocation": false,
        "GhostMode": {
            "isGhostMode": true,
            "GhostModeFriends": [
                "User58",
                "User16"
            ]
        }
    }
```

Implementazione e query

6.1 Implementazione del database

Una volta creato il database, questo è stato popolato con dati prodotti da uno script in Python. I dati sono fittizi e sono generati in maniera random. Lo script Python, le immagini di tutti gli schemi e la documentazione si trovano all'interno della repository **GitHub**.
Le query degli screenshot che seguiranno in questo paragrafo sono state eseguite attraverso la shell di **MongoDB Compass**.
Prima di poter utilizzare le query è necessario inserire il comando *use snapchat* per posizionarsi sul giusto database.

6.2 Query in MongoDB

Per la creazione della base di dati è stato utilizzato un servizio chiamato **MongoDB Atlas**: quest'ultimo permette di effettuare il deploy di un database in cloud, in modo da permettere di avere un ambiente di lavoro condiviso e sempre online. Per l'interfacciamento con il database è stato invece utilizzato **MongoDB Compass**, altro strumento fornisce un'interfaccia grafica attraverso la quale effettuare il deploy e realizzare query sul database.

Il database, come visto precedentemente, è composto da cinque collection: user, map, snap, chat, sticker. Sono state realizzate le query per le operazioni più interessanti fra quelle presenti nella tabella delle operazioni.
Le operazioni selezionate sono:

1. Visualizza intero archivio snap di uno user
2. User aggiunge like ad uno snap
3. Visualizza user taggati su uno snap
4. Visualizza ultimi 5 memories di uno user sulla mappa
5. Visualizza like ai luoghi presenti sulla mappa (user business)
6. Aggiungi snap nella lista di memories

Visualizza intero archivio snap di uno user

Con la seguente query vengono cercati tutti gli snap di un utente e vergono ordinati per timestamp, ovvero partendo dal più recente.

```
db.snap.aggregate([
  { $match: { ownSnaps: "user0" } },
  {
    $sort: { timestamp: 1 }
  },
  {
    $project: {
      _id: 0,
      idSnap: 1,
      ownSnaps: 1,
      timestamp: 1
    }
  }
]);
```

Di seguito è riportato il risultato ottenuto applicando la query.

User aggiunge like ad uno snap

La query è composta dall'operatore ***$inc*** per incrementare il numero dei likes e l'operatore ***$push*** per inserire la chiave dell'utente nella struttura dati che memorizza gli utenti che hanno messo like.

```
db.snap.updateOne(
  { idSnap: "User0Snap0" },
  {
    $inc: { LikesCount: 1 },
    $push: { UserLikes: "User17" }
  }
)
```

Di seguito è riportato il risultato ottenuto applicando la query.

```
> db.snap.updateOne(
    { idSnap: "User0Snap0" },
    {
      $inc: { LikesCount: 1 },
      $push: { UserLikes: "User17" }
    }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 0
  }
```

Visualizza user taggati su uno snap

Nella seguente query si effettua un collegamento tra le collection user e snap, per ottenere gli utenti taggati sugli snap (da collection snap) e i dati relativi agli stessi utenti (da collection user).

Di seguito è presente l'implementazione della query per un determinato snap.

```
db.snap.aggregate([
  { $match: { idSnap: "user0Snap8" } },
  {
    $lookup: {
      from: "user",
      localField: "tagUsers",
      foreignField: "username",
      as: "risultati"
    }
  },
  {
    $project: {
      _id: 0,
      idSnap: 1,
      "risultati.username": 1,
      "risultati.anagrafica.firstName": 1,
```

```
      "risultati.anagrafica.lastName": 1
    }
  }
]);
```

Di seguito è riportato il risultato ottenuto applicando la query.

Visualizza ultimi 5 memories (snap) di uno user sulla mappa

La seguente query viene utilizzata per ottenere gli ultimi 5 snap di un utente inseriti nella mappa. Con l'operatore ***$match*** si definisce l'utente proprietario degli snap e si verifica che l'array non sia vuoto.

Viene utilizzato l'operatore ***$unwind*** per scomporre l'array Snaps e trattare ogni elemento come un documento separato.

L'operatore ***$lookup*** è utilizzato per trovare i documenti corrispondenti in base all'attributo idSnap nei documenti della collezione map. I risultati vengono proiettati nell'array snapData.

```
db.map.aggregate([
  {
    $match: {
      user: "user1",
      snaps: { $ne: [] }
    }
  },
  {
    $unwind: "$snaps"
  },
  {
    $lookup: {
      from: "snap",
      localField: "snaps",
      foreignField: "idSnap",
      as: "snapData"
    }
  },
  {
    $unwind: "$snapData"
  },
  {
    $project: {
      "snapData.idSnap": 1,
      "snapData.ownSnaps": 1,
      "snapData.Timestamp": 1,
      "_id": 0
    }
```

```
  },
  {
    $sort: {
      "snapData.Timestamp": -1
    },
  },
  {
    $limit: 5
  }
])
```

Di seguito è riportato il risultato ottenuto applicando la query.

Visualizza like ai luoghi presenti sulla mappa (user business)

La query seguente viene utilizzata per mostrare i luoghi presenti sulla mappa ed i like di tali luoghi.

Inizialmente si verifica che l'attributo isBusiness sia *true* in quanto solo gli account business possono essere mostrati sulla mappa, dopo vengono mostrati gli attributi che mostrano il numero dei likes e il nome del luogo.

```
db.user.aggregate([
  {
    $match: {
      "businessInfo.isBusiness": true
    }
  },
  {
    $project: {
      _id: 0,
      username: 1,
      bLikesCount: "$businessInfo.bLikesCount",
      bName: "$businessInfo.bName"
    }
  }
])
```

Di seguito è riportato il risultato ottenuto applicando la query.

Aggiungi snap nella lista di memories

La query seguente viene utilizza per aggiungere uno snap nella lista di memories mostrate sulla mappa. Si verifica che lo snap non sia già presente nella lista, che l'attributo statusGps sia a *true* e che l'attributo location non sia una stringa vuota.

```
db.map.updateOne(
  {
    user: "user0",
    snaps: { $ne: "user0Snap7" },
    location: { $ne: "" },
    statusGps: true
```

```
  },
  {
    $push: {
      snaps: "user0Snap7"
    }
  }
);
```

Di seguito è riportato il risultato ottenuto applicando la query.