



Università degli Studi di Salerno



Dipartimento di Ingegneria dell'Informazione ed Elettrica e
Matematica Applicata

Corso di Laurea in Ingegneria Informatica

Basi di Dati 2022/2023

Canale A-H

Project Work

**Traccia N. 2 - Gestione dei dati sui post degli
influencer su LinkedIn**

Gruppo n. **09 – AH**

WP	Cognome e Nome	Matricola	e-mail	Responsabile
1	Attianese Carmine	0612705049	c.attianese13@studenti.unisa.it	
2	De Bartolomeis Mattia	0612705104	m.debartolomeis1@studenti.unisa.it	
3	Cerino Marco	0612704997	m.cerino11@studenti.unisa.it	X

4	De Angelis Christian	0612705050	c.deangelis29@studenti.unisa.it	
---	----------------------	------------	--	--

Anno accademico 2022-2023

Sommario

1.	Descrizione della realtà di interesse	3
1.1.	Analisi della realtà di interesse	3
2.	Analisi delle specifiche	6
2.1.	Glossario dei termini.....	6
2.2.	Strutturazione dei requisiti in frasi.....	6
2.2.1.	Frasi di carattere generale	6
2.2.2.	Frasi relative a post.....	6
2.2.3.	Frasi relative a utente	7
2.2.4.	Frasi relative a tipi specifici di utente	7
2.2.5.	Frasi relative a hashtag	7
2.2.6.	Frasi relative a panoramica.....	7
2.2.7.	Frasi relative a media	7
2.3.	Identificazione delle operazioni principali	7
3.	Progettazione Concettuale.....	8
3.1.	Schema Concettuale	8
3.1.1.	Note sullo schema E-R	9
3.2.	Design Pattern	9
3.2.1.	Pattern Reificazione di Attributo di Entità	9
3.2.2.	Pattern Reificazione di Relazione Binaria	9
3.2.3.	Pattern Reificazione di Relazione Ricorsiva	10
3.2.4.	Pattern Storizzazione di entità	10
3.2.5.	Pattern Reificazione di attributo di entità	11
3.2.6.	Pattern Reificazione di attributo di entità	11
3.2.7.	Pattern Reificazione di attributo di entità	11
3.3.	Dizionario dei Dati	12
3.4.	Regole Aziendali	14
4.	Progettazione Logica	16
4.1.	Ristrutturazione Schema Concettuale	16
4.1.1.	Analisi delle Prestazioni	16
4.2.	Analisi delle ridondanze.....	17
4.2.1.	Analisi della ridondanza 1: Numero post.....	17
4.2.2.	Analisi della ridondanza 2: Numero hashtag	19
4.3.	Eliminazione delle generalizzazioni	21
4.3.1.	Generalizzazione <i>Utente LinkedIn</i>	21
4.3.2.	Generalizzazione <i>Following</i>	22
4.4.	Partizionamento/Accorpamento Entità e Associazioni	22
4.5.	Scelta degli identificatori principali	23

4.6.	Schema ristrutturato finale	24
4.7.	Schema logico	25
4.8.	Documentazione dello schema logico	25
5.	Normalizzazione	26
6.	Script Creazione e Popolamento Database	30
7.	Query SQL	43
7.1.	Query con operatore di aggregazione e join: Numero di visualizzazioni per ogni tipo di media	43
7.2.	Query nidificata complessa: Visualizzazioni totali per influencer con almeno un post contenente un'immagine	43
7.3.	Query insiemistica: Influencer con hashtag, link specifici e più di 10k follower.....	44
7.4.	Eventuali Altre query	44
7.4.1.	Influencer con tutti i post con reazioni > 500	44
7.4.2.	Post più influente per somma di reazioni e commenti	45
7.4.3.	Influencer con tutti i post con reazioni > 500	45
8.	Viste	47
8.1.	Vista Numero post ed hashtag di uno specifico utente	47
8.1.1.	Query con Vista: Selezione di uno specifico utente secondo determinate condizioni	47
9.	Trigger	48
9.1.	Trigger inizializzazione: Gestione del vincolo “un hashtag deve essere contenuto almeno in una tabella Caratterizzato”	48
9.2.	Trigger per vincoli aziendali	48
9.2.1.	Trigger1: Controllo numero post di un influencer (RV12 e RV15)	48
9.2.2.	Trigger2: Voti di un sondaggio(RV8)	50
9.2.3.	Trigger3: Aggiornamento numero hashtag.....	50

1. Descrizione della realtà di interesse

Titolo: **Gestione dei dati sui post degli influencer su LinkedIn**

Si vuole realizzare un database per una società che analizza dati dei social network, di cui si vogliono rappresentare le informazioni relative ai post e ai profili LinkedIn degli influencer. La base di dati è in grado di memorizzare informazioni che possono essere caricate sia manualmente sia da un dataset esistente. Il dataset di riferimento "LinkedIn Influencers' Data" è realizzato da Shreya Sajal, studentessa presso IIT Guwahati, e Vineet Agarwal, ricercatore presso Cranfield University.

In particolare:

- L'utente (circa 500.000), caratterizzato da un codice identificativo, nell'ambito di LinkedIn ha un profilo, di cui si vuole memorizzare il nome, il cognome, la città, la regione e il paese dove è ubicato, il numero di follower, una descrizione del profilo, il numero di connessioni, il numero di utenti seguiti e l'istituzione che comprende il nome del luogo di lavoro e le proprie occupazioni. Ogni utente può essere un influencer, di cui si vuole memorizzare il numero di post pubblicati. **Per ogni utente si vuole tener traccia di tutti gli altri utenti seguiti al variare del tempo.**
- **Gli utenti di LinkedIn pubblicano dei post periodicamente sul loro profilo.** Per ogni post (circa 50000), caratterizzato da un codice univoco, si vuole memorizzare il numero di hashtag utilizzati nel post, la somma del numero di follower di tutti gli hashtag presenti all'interno del post, il tempo trascorso dalla pubblicazione del post alla data di riferimento (data di aggiornamento del dataset), i link all'interno del post (esclusi gli hashtag e i media) ed il contenuto testuale del post (dato opzionale) esclusi gli hashtag.
- **Ad ogni post può essere associato un media.** Per ogni media, caratterizzato da un url, si vuole memorizzare il tipo (immagini, video, sondaggi, articoli o documento) (dato opzionale).
- **In ogni post possono essere presenti uno o più hashtag.** Ogni hashtag è caratterizzato da un nome o dal relativo link.
- **Per ogni post dell'influencer si analizza una panoramica.** Per ogni panoramica, caratterizzata dal relativo post, si vuole memorizzare il numero di reazioni, di commenti, di visualizzazioni (dato opzionale), di voti (in caso di sondaggi) associati al post.

1.1. Analisi della realtà di interesse

L'obiettivo del progetto è quello di creare una base di dati che sia modellata sul dataset di riferimento (LinkedIn Influencers' Data) i quali dati sono raccolti tramite procedure di scraping utilizzando la libreria di Python BeautifulSoup ed il tool Selenium. Il dataset (presente al seguente link: <https://www.kaggle.com/datasets/shreyasajal/linkedin-influencers-data>) contiene 34011 post ed è composto da 19 colonne quali: il numero seriale che identifica i post numerandoli consecutivamente da 0(#), il nome ed il cognome dell'influencer(**name**), l'istituzione che comprende il nome del luogo di lavoro e le proprie occupazioni(**headline**), la locazione dell'influencer fornita su LinkedIn(**location**), il numero di follower di ogni influencer(**#followers**), il numero di connessioni di ogni profilo dell'influencer(**connections**), una descrizione dell'influencer

utilizzata come biografia del profilo(**about**), il tempo trascorso dal caricamento del post fino ad una data di riferimento(attualmente 15 Febbraio 2021)(**time_spent**), il contenuto del post(**content**), i link contenuti all' interno del post ed il link del post stesso(**content_links**), il tipo di media usati all' interno del post(come immagini, video e sondaggi)(**media_type**), l' url relativo al contenuto multimediale presente all' interno del post(**media_url**),il numero di hashtag usati all' interno del post(**# num_hashtags**), la somma del numero di follower di tutti gli hashtag presenti all' interno del post(**# hashtag_followers**), la lista di hashtag citati all' interno del post composta dal nome di ogni hashtag ed il relativo link(**hashtags**), il numero di reazioni del post(**# reactions**), il numero di commenti sotto al post(**# comments**), il numero di visualizzazioni del post(**views**) ed il numero di voti al post nel caso sia un sondaggio(**# votes**).

Nel dataset non vengono considerati i dati relativi agli utenti di LinkedIn che non sono influencer. Tuttavia, è di interesse tener traccia di tale insieme per una successiva analisi più dettagliata. Si precisa che i dati relativi agli utenti che non sono influencer sono raccolti analizzando i reali profili che seguono, o sono seguiti, dagli influencer presenti nel dataset di riferimento. Tali dati verranno caricati manualmente nel database.

Ogni influencer considerato all' interno del dataset ha almeno un post pubblicato, il quale può contenere testo e/o media. All' interno del dataset vengono rispettati i seguenti vincoli imposti da LinkedIn: ogni post ha un limite massimo di 1300 caratteri,30 hashtag ed un solo tipo di media. Ogni media può essere un articolo, un'immagine, un video, un sondaggio o un documento. L'url che caratterizza il media può avere al massimo 1000 caratteri. Il profilo di ogni utente di LinkedIn è strutturato in modo da avere un massimo di 20 caratteri per il nome, 40 caratteri per il cognome,220 caratteri per l'intestazione (120 per occupazione e 100 per nome del luogo di lavoro) e 2000 caratteri per la descrizione del profilo. Non vi è un numero minimo/massimo per il numero di follower di ogni profilo e per il numero di reazioni, commenti, visualizzazioni e voti (nel caso di sondaggio) relativo al singolo post dell'influencer. Tali caratteristiche sono mostrate dalla panoramica associata al post. Per le connessioni relative al singolo profilo dell'utente, superiori a 500, non è importante conoscerne il numero preciso.

Inoltre, per ogni utente linkedin, si può avere come informazione solo la città di ubicazione, solo lo stato, solo regione-città, tutti e tre insieme oppure nessuno dei tre. È importante sottolineare che, per regioni, si intendono anche gli stati americani e tutte le zone con la suddivisione geografica simile ad essa.

Si precisa che il numero di follower si riferisce al numero totale di follower del profilo dell'utente, tuttavia potrebbero non essere analizzati tutti i seguaci di quest'ultimo.

È richiesto che il database venga gestito da PostgreSQL (requisito non funzionale).

In particolare, l'attenzione del progetto è rivolta ai post pubblicati dagli influencer, con l'obiettivo di fornire dettagli esaustivi sulle informazioni relative ai post. Ciò comporta la memorizzazione dei link corrispondenti, del contenuto testuale del post, del numero di hashtag utilizzati, del numero di follower degli hashtag contenuti e dei tipi di media associati. Si precisa che il contenuto testuale del post non contiene gli hashtag al suo interno. Tutti gli url considerati devono necessariamente iniziare con 'http'.

Un altro aspetto di interesse sono le panoramiche sui post degli influencer, per le quali si memorizza il numero di commenti, di reazioni, di visualizzazioni e di voti in caso di sondaggio. Un fattore secondario è l'analisi del profilo dell'influencer in base ai follower così da capire la tipologia dei propri seguaci.

Inoltre, il tempo trascorso di tutti i post è calcolato rispetto ad una data di riferimento, ovvero la data di aggiornamento del dataset (il dataset in esame è attualmente aggiornato al 15 febbraio 2021). Ciò implica che ci sono post precedenti (caricati dal dataset) e successivi alla data all' interno del nostro database (caricati dall' utente). Il tempo trascorso può essere utile per risalire alla data di creazione del post.

Si precisa che nel database i soli autori dei post sono gli influencer, e quindi seppur tutti gli utenti siano in grado di pubblicare i post su LinkedIn, tale aspetto viene tralasciato. All' interno del database non è di interesse tener traccia degli influencer che non sono più tali o di un utente che diventa influencer.

Non è di interesse tener traccia delle occupazioni passate dell'utente ma solo di quelle attuali. **Non vengono distinte città che hanno lo stesso nome.**

Per una futura implementazione per analizzare al meglio i post di un influencer si potrebbe pensare all'aggiunta del numero di condivisioni del post. Un altro aspetto di interesse futuro potrebbe essere considerare i post caricati da tutti gli utenti LinkedIn senza trattare esclusivamente quelli degli influencer.

2. Analisi delle specifiche

Workpackage	Task	Responsabile
WP0	Analisi delle specifiche	Intero Gruppo

2.1. Glossario dei termini

	Termine	Descrizione	Sinonimi	Collegamenti
1	Post	Qualsiasi pubblicazione di un utente sul proprio profilo	-	Utente, Media, Hashtag, Panoramica
2	Utente	Persona registrata su LinkedIn avente un proprio profilo, contenente le proprie informazioni personali	-	Influencer, Post
3	Influencer	Un utente che ha raggiunto un alto livello di visibilità ed influenza sulla piattaforma LinkedIn mediante la pubblicazione regolare di post.	-	Utente
4	Media	Contenuti come video e immagini all'interno del post	-	Post
5	Panoramica Post	Resoconto di like, visualizzazioni, commenti e voti, in caso di sondaggi, associati ad un singolo post	-	Utente, Post
6	Hashtag	Parola o frase, preceduta dal simbolo #, che viene utilizzato per categorizzare i post in modo da renderli facilmente ricercabili e raggiungibili da altri utenti interessate agli stessi argomenti.	-	Post

Tabella 1. Glossario dei Termini

2.2. Strutturazione dei requisiti in frasi

2.2.1. Frasi di carattere generale

Si vuole realizzare un database per una società che analizza dati dei social network, di cui vogliamo rappresentare le informazioni relative ai post e ai profili LinkedIn degli influencer. La base di dati è in grado di memorizzare informazioni che possono essere caricate sia manualmente che dal dataset esistente.

2.2.2. Frasi relative a post

Per ogni post (circa 50000), caratterizzato da un codice univoco, si vuole memorizzare il numero di hashtag utilizzati nel post, la somma del numero di follower di tutti gli hashtag presenti all'interno del post, il tempo trascorso dalla pubblicazione del post alla data di riferimento (data di aggiornamento del dataset), i link all'interno del post (esclusi gli hashtag) ed il contenuto testuale del post (dato opzionale).

2.2.3. Frasi relative a utente

L'utente (circa 500000), caratterizzato da un codice identificativo, nell'ambito di LinkedIn ha un profilo, di cui si vuole memorizzare il nome, il cognome, la città, la regione e il paese dove è ubicato, il numero di follower, una descrizione del profilo, il numero di connessioni, il numero di utenti seguiti e l'istituzione che comprende il nome del luogo di lavoro e le proprie occupazioni. Ogni utente può essere un influencer, di cui si vuole memorizzare il numero di post pubblicati.

2.2.4. Frasi relative a tipi specifici di utente

Ogni utente può essere un influencer, di cui si vuole memorizzare il numero di post pubblicati. **Per ogni utente si vuole tener traccia di tutti gli altri utenti seguiti al variare del tempo.**

2.2.5. Frasi relative a hashtag

Ogni hashtag è caratterizzato da un nome o dal relativo link.

2.2.6. Frasi relative a panoramica

Per ogni post dell'influencer si analizza una panoramica. Per ogni panoramica, caratterizzata dal relativo post, si vuole memorizzare il numero di reazioni, di commenti, di visualizzazioni (dato opzionale), di voti (in caso di sondaggi) associati al post.

2.2.7. Frasi relative a media

Per ogni media, caratterizzato da un url, si vuole memorizzare il tipo (immagini, video, sondaggi, articoli o documento) (dato opzionale).

2.3. Identificazione delle operazioni principali

Operazione 1: inserisci i dati relativi ad un post dell'influencer con la relativa panoramica ed i relativi hashtag utilizzati (in media 1500 volte al giorno)

Operazione 2: inserisci i dati relativi al profilo dell'utente (in media 15000 volte al giorno);

Operazione 3: Trovare le informazioni degli utenti che seguono un determinato influencer (in media 300 volte al giorno);

Operazione 4: stampa la panoramica per un singolo post di un influencer (in media 1500 volte al giorno);

Operazione 5: aggiorna i dati relativi ai profili degli utenti, ai post degli influencer e alle relative panoramiche rispetto alla data di riferimento (in media 1 volta al mese);

Operazione 6: identifica il post più influente basandosi sul numero totale di dati presenti nella rispettiva panoramica (in media 1 volta al mese Batch)

Operazione 7: Stampa il numero di hashtag all'interno e di hashtag follower di ogni post di un utente (in media 600 volte al giorno);

3. Progettazione Concettuale

Workpackage	Task	Responsabile
WP1	Progettazione Concettuale	Attianese Carmine

3.1. Schema Concettuale

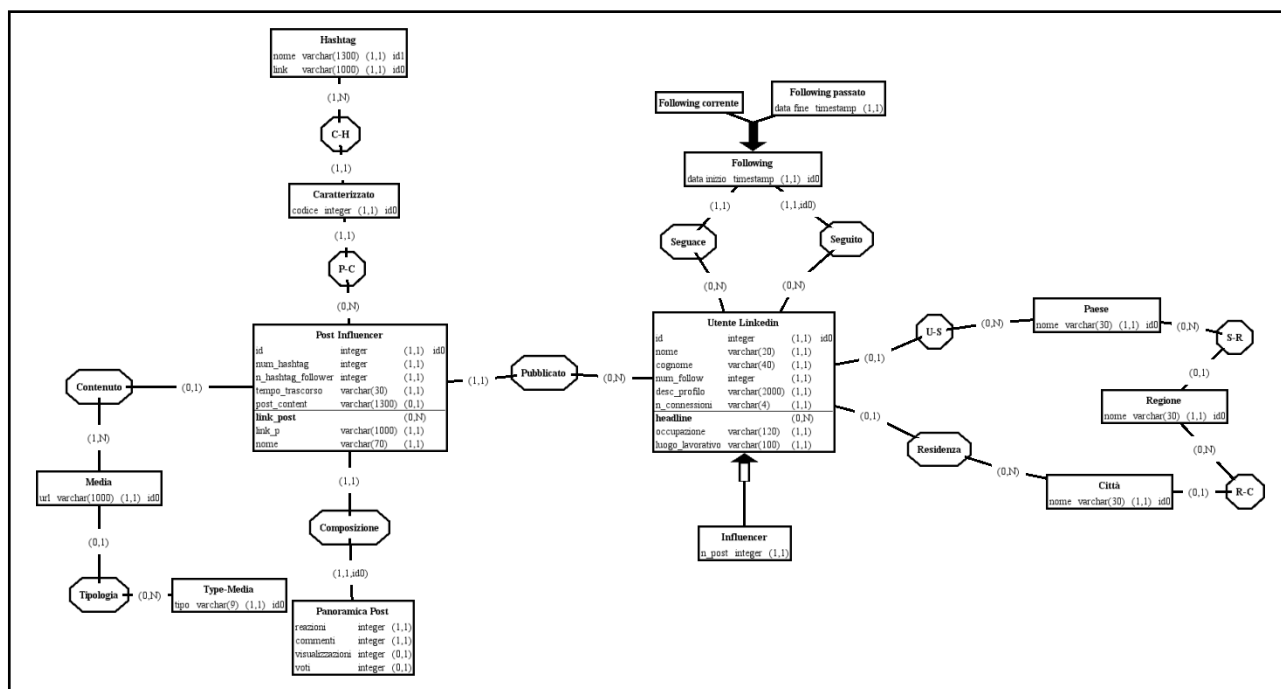


Figura 1. Schema E-R

3.1.1. Note sullo schema E-R

Per la fase di progettazione concettuale è stata impiegata la strategia mista. Quest'ultima combina i vantaggi della strategia top-down e bottom-up. Vi è una suddivisione dei requisiti in diverse componenti e la creazione di uno schema scheletro che contiene, a livello astratto, i concetti principali del progetto. Da qui, partendo da un concetto cardine, si effettuano diversi raffinamenti espandendosi a macchia d'olio.

3.2. Design Pattern

3.2.1. Pattern Reificazione di Attributo di Entità

L'obiettivo è la classificazione di tutti i tipi di media contenuti all'interno del post. La separazione dell'informazione sui diversi tipi di media in un'entità distinta semplifica la gestione e la definizione di regole aziendali per la classificazione dei media.

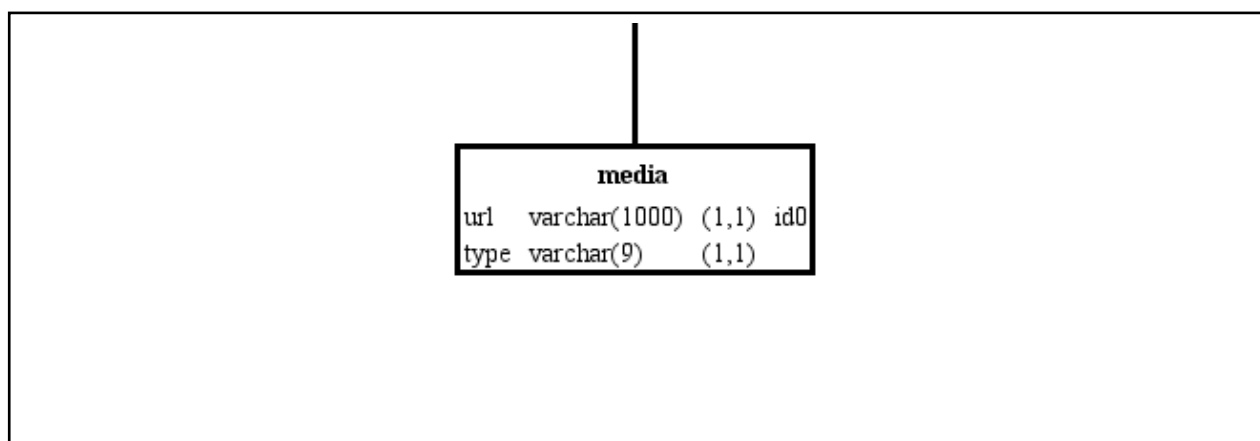


Figura 2. Schema precedente all'applicazione del Pattern REIFICAZIONE DI ATTRIBUTO DI ENTITA.

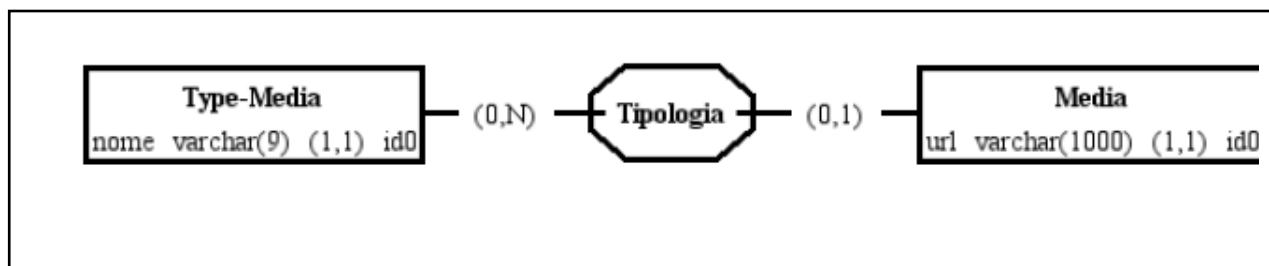


Figura 3. Schema successiva all'applicazione del Pattern REIFICAZIONE DI ATTRIBUTO DI ENTITA.

Tale opzione può essere applicata anche senza l'utilizzo del pattern rappresentando il tipo di media direttamente come un concetto a sé, ovvero come un'entità.

3.2.2. Pattern Reificazione di Relazione Binaria

Il seguente pattern viene applicato poiché nella realtà un post può contenere lo stesso hashtag più volte, tuttavia il nostro schema non permette di rappresentare tale situazione.

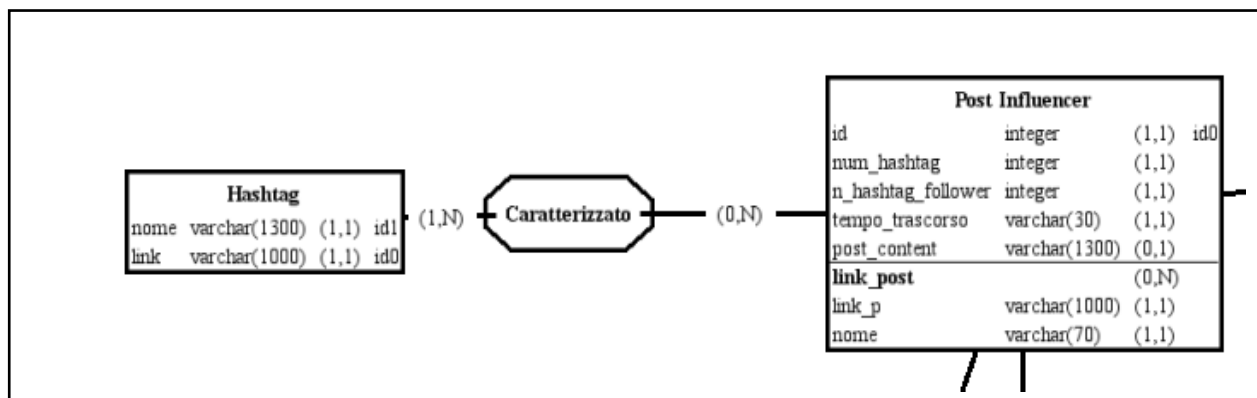


Figura 4. Schema precedente all'applicazione del Pattern REIFICAZIONE RELAZIONE BINARIA.

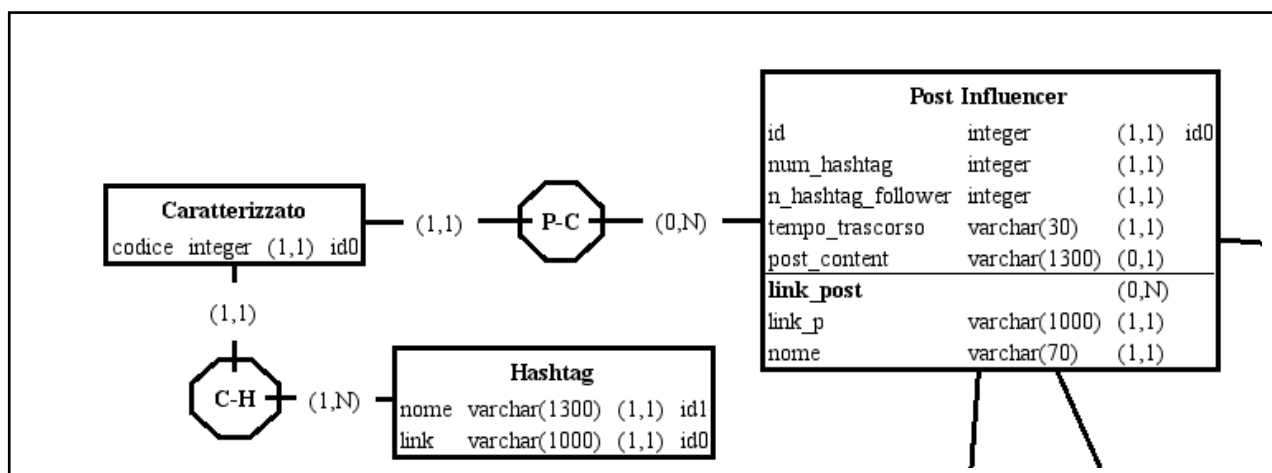


Figura 5. Schema successiva all'applicazione del Pattern REIFICAZIONE RELAZIONE BINARIA.

In questo caso l'associazione diviene un'entità a sé legata alle due entità (post influencer ed hashtag) mediante due associazioni. Le cardinalità delle associazioni tra le diverse entità dipendono dalle cardinalità nello schema iniziale.

3.2.3. Pattern Reificazione di Relazione Ricorsiva

Il seguente pattern viene applicato poiché lo schema iniziale non permette di rappresentare la situazione in cui un utente può seguire un altro utente più volte (ad esempio un utente smette di seguire un altro utente e successivamente lo risegue).

Nota:

Dato che il pattern è stato applicato in precedenza al pattern di storicizzazione, le immagini sono inserite direttamente nella sezione successiva.

3.2.4. Pattern Storicizzazione di entità

Il seguente pattern viene applicato poiché nella realtà è richiesto per ogni utente di tener traccia di tutti gli altri utenti seguiti al variare del tempo.

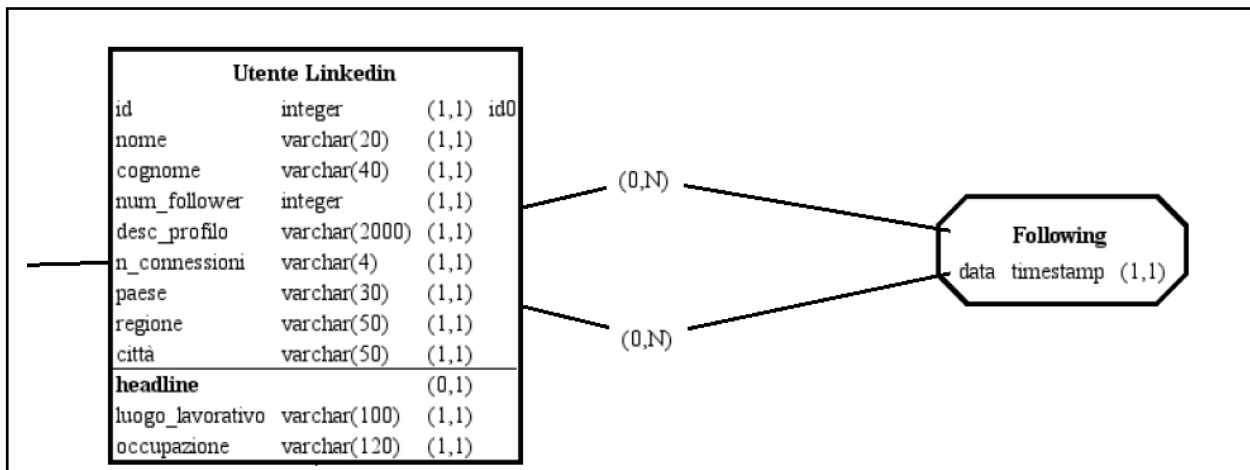


Figura 6. Schema precedente all'applicazione dei Pattern REIFICAZIONE RELAZIONE RICORSIVA e STORICIZZAZIONE

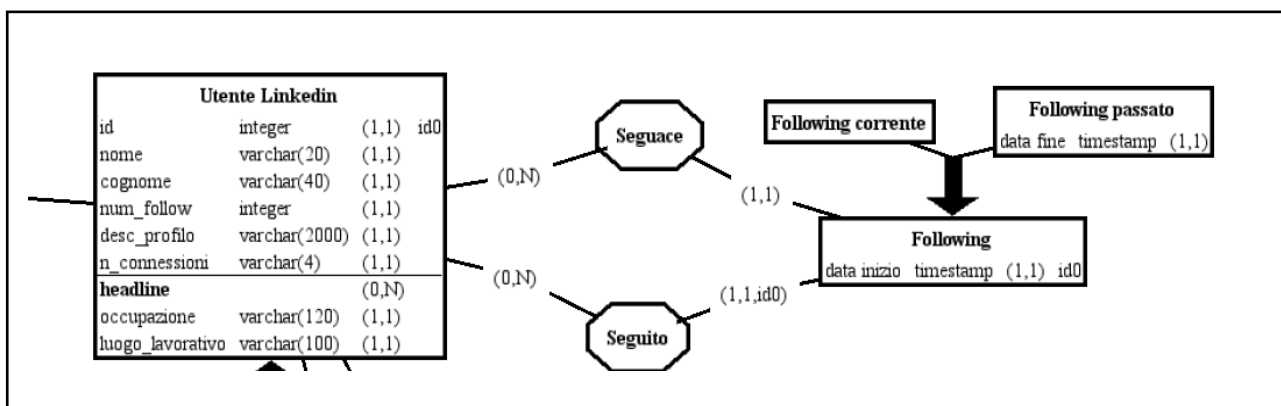


Figura 7. Schema successiva all'applicazione dei Pattern REIFICAZIONE RELAZIONE RICORSIVA e STORICIZZAZIONE

3.2.5. Pattern Reificazione di attributo di entità

L'obiettivo è la classificazione delle diverse città. La separazione dell'informazione in un'entità distinta semplifica la gestione e la definizione di regole aziendali per la classificazione delle città. Tale opzione può essere applicata anche senza l'utilizzo del pattern rappresentando la città direttamente come un concetto a sé, ovvero come un'entità.

3.2.6. Pattern Reificazione di attributo di entità

L'obiettivo è la classificazione delle diverse regioni. La separazione dell'informazione in un'entità distinta semplifica la gestione e la definizione di regole aziendali per la classificazione delle regioni. Tale opzione può essere applicata anche senza l'utilizzo del pattern rappresentando la regione direttamente come un concetto a sé, ovvero come un'entità.

3.2.7. Pattern Reificazione di attributo di entità

L'obiettivo è la classificazione dei diversi paesi. La separazione dell'informazione in un'entità distinta semplifica la gestione e la definizione di regole aziendali per la classificazione dei paesi. Si precisa che, per i paesi europei, tale entità tiene in considerazioni i diversi stati mentre, per paesi extra europei (come ad esempio l'America), tale entità identifica i diversi paesi.

Tale opzione può essere applicata anche senza l'utilizzo del pattern rappresentando il paese direttamente come un concetto a sé, ovvero come un'entità.

3.3. Dizionario dei Dati

Entità	Descrizione	Attributi	Identificatore
Post	Qualsiasi pubblicazione di un influencer sul proprio profilo	Id, numero di hashtag, follower hashtag, tempo trascorso, contenuto post, link del post (nome profilo, link)	Id
Utente LinkedIn	Persona registrata su LinkedIn avente un proprio profilo, contenente le proprie informazioni personali	id, nome, cognome, numero follower, descrizione del profilo, numero di connessioni, headline (occupazione e luogo_lavorativo)	id
Influencer	Un utente che ha raggiunto un alto livello di visibilità ed influenza sulla piattaforma LinkedIn mediante la pubblicazione regolare di post.	Attributi ereditati da utente, numero post	id
Media	Contenuti come video e immagini all'interno del post	url	url
Panoramica Post	Resoconto di like, visualizzazioni, commenti e voti, in caso di sondaggi, associati ad un singolo post	Reazioni, commenti, visualizzazioni, voti	Post influencer
Hashtag	Parola o frase, preceduta dal simbolo #, che viene utilizzato per categorizzare i post in modo da renderli facilmente ricercabili e raggiungibili da altri	Nome, link	Nome o link

	utenti interessate agli stessi argomenti		
Type-Media	Indica i diversi tipi di media (reificazione di attributo di entità)	tipo	tipo
Caratterizzato	Permette di avere più volte lo stesso hashtag all' interno di un post (reificazione di relazione binaria)	codice	codice
Following	Tiene traccia del legame di following (reificazione relazione ricorsiva)	Data inizio	Data inizio
Following Corrente	Tiene traccia del legame di following corrente (storicizzazione di entità)	Attributi ereditati da Following	Data inizio
Following Passato	Tiene traccia del legame di following passato (storicizzazione di entità)	Attributi ereditati da Following, data fine	Data inizio
Città	Indica le diverse città (reificazione attributo di entità)	Nome	Nome
Regione	Indica le diverse regioni (reificazione attributo di entità)	Nome	Nome
Paese	Indica i diversi paesi (reificazione attributo di entità)	Nome	Nome

Tabella 2. Dizionario dei dati – Entità

Relazioni	Descrizione	Entità Coinvolte	Attributi
Pubblicato	Associa un utente ad un post dell'influencer	Influencer (1, N), Post (1,1)	
Contenuto	Associa un post ad un media	Post (0, 1), Media (1, N)	
Composizione	Associa un post ad una panoramica post	Post (1,1), Panoramica Post (1,1)	
Tipologia	Associa un media ad un tipo di media	Media (0,1), Type-Media (0, N)	
P-C	Associa un post a caratterizzato	Post Influencer (0, N), Caratterizzato (1,1)	

C-H	Associa caratterizzato ad hashtag	Caratterizzato (1,1), Hashtag (1, N)	
Seguace	Associa un utente al following	Utente LinkedIn (0, N), Following (1,1)	
Seguito	Associa un utente al following	Utente LinkedIn (0, N), Following (1,1)	
U-S	Associa un utente ad un paese	Utente LinkedIn (0,1), Paese (0, N)	
S-R	Associa un paese ad una regione	Paese (0, N), Regione (0,1)	
R-C	Associa una regione ad una città	Regione (0, N), Città (0,1)	
Residenza	Associa un utente ad una città	Città (0, N), Utente LinkedIn (0,1)	

Tabella 3. Dizionario dei dati - Relazioni

Attributo	Descrizione	Entità coinvolta	Cardinalità
Tempo trascorso	Indica il tempo trascorso dalla pubblicazione di un post alla data di riferimento	Post	(1,1)

Workpackage	Task	Responsabile
WP4	Regole Aziendali	De Angelis Christian

3.4. Regole Aziendali

Regole di Vincolo
<p>(RV1) Un post dell'influencer deve avere al massimo 1300 caratteri.</p> <p>(RV2) Il numero di hashtag contenuti in un post deve essere compreso tra 0 e 30(estremi inclusi).</p> <p>(RV3) Il numero di follower degli hashtag contenuti all'interno di un post deve essere ≥ 0.</p> <p>(RV4) Il numero di commenti della panoramica di un post deve essere ≥ 0.</p> <p>(RV5) Il numero di visualizzazioni della panoramica di un post deve essere ≥ 0.</p> <p>(RV6) Il numero di reazioni della panoramica di un post deve essere ≥ 0.</p> <p>(RV7) Il numero di voti della panoramica di un post deve essere ≥ 0.</p> <p>(RV8) Un post che non contiene un sondaggio non deve indicare il numero di voti.</p> <p>(RV9) Il numero di follower di un profilo dell'utente deve essere ≥ 0.</p> <p>(RV10) Il numero di caratteri del nome del profilo di un utente deve essere compreso tra 0 e 20(estremi inclusi).</p> <p>(RV11) Il numero di caratteri del cognome del profilo di un utente deve essere compreso tra 0 e 40(estremi inclusi).</p>

<p>(RV12) Quando il numero di post di un influencer è esattamente 1, viene inviato un messaggio di avviso. Inoltre, se il numero di post di un influencer supera 10, viene inviato un messaggio di avviso per segnalare che la media del post di quell'utente sta superando 5(media prefissata).</p> <p>(RV13) Il numero di caratteri della descrizione del profilo di un utente deve essere compreso tra 0 e 2000(estremi inclusi).</p> <p>(RV14) Le connessioni del profilo di un utente superiori a 500 devono essere indicate con 500+.</p> <p>(RV15) Il numero di post del profilo di un influencer deve essere >0.</p> <p>(RV16) Il nome di un hashtag deve iniziare con il carattere '#'. (RV17) Il tempo trascorso dalla pubblicazione di un post alla data di aggiornamento del dataset deve seguire il formato di LinkedIn.</p> <p>(RV18) Il numero di post pubblicato dagli influencer deve essere ≥ 1.</p> <p>(RV19) Un utente non influencer deve avere 0 post pubblicati.</p> <p>(RV20) Il luogo lavorativo non deve essere presente se non è presente l'occupazione.</p> <p>(RV21) L' url deve iniziare con http:// oppure https://.</p>

Tabella 4. Regole di vincolo

Regole di derivazione
<p>(RD1) Il numero di hashtag si ottiene sommando gli hashtag contenuti in un post.</p> <p>(RD2) Il numero di post di un influencer si ottiene sommando tutti i post pubblicati.</p>

Tabella 5. Regole di derivazione

Note:

È richiesto che il database venga gestito da PostgreSQL (requisito non funzionale).

I post caricati dal dataset sono precedenti rispetto alla data di riferimento, quindi il tempo trascorso contiene la parola chiave "ago", mentre i post caricati manualmente successivi alla data di riferimento contengono la parola chiave "after" (RV17).

Analisi di qualità

Durante la progettazione concettuale sono state verificate più volte le seguenti proprietà che uno schema concettuale di buona qualità deve possedere:

Correttezza verificata mediante ispezione, confrontando i concetti presenti nello schema con le specifiche e con le definizioni dei costrutti del modello concettuale usato;

Completezza verificata analizzando che tutte le specifiche sui dati siano rappresentate nello schema e che tutte le operazioni presenti nelle specifiche siano raggiungibili navigando attraverso lo schema;

Leggibilità verificando che i requisiti siano rappresentati in modo naturale e facilmente comprensibile;

Assenza di minimalità in quanto esistono delle ridondanze all' interno dello schema.

All' interno dello schema concettuale, vi sono i diversi dati ridondanti:

numero di hashtag contenuti in un post;

numero di post pubblicati da un influencer;

4. Progettazione Logica

Workpackage	Task	Responsabile
WP2	Progettazione Logica	De Bartolomeis Mattia

4.1. Ristrutturazione Schema Concettuale

4.1.1. Analisi delle Prestazioni

4.1.1.1. Tavola dei volumi

Concetto	Tipo	Volume
Utente LinkedIn	E	500.000
Influencer	E	10.000
Città	E	10.000
Regione/Stato	E	1.000
Paese	E	195
Impiego	E	5.000
Following	E	50.000.000
Following corrente	E	40.000.000
Following passato	E	10.000.000
Post influencer	E	50.000
Caratterizzato	E	1.000
Hashtag	E	1.000
Panoramica post	E	50.000
Media	E	30.000
Type-media	E	5
Residenza	R	200.000
U-S	R	250.000
S-R	R	50.000
R-C	R	100.000
lavoro	R	5.000
seguito	R	50.000.000
seguace	R	50.000.000
pubblicato	R	50.000
composizione	R	50.000
P-C	R	1.000
C-H	R	1.000
contenuto	R	40.000
tipologia	R	25.000

Tabella 6. Tavola dei volumi

4.1.1.2. Tavola delle operazioni

Viene di seguito riportata la tavola delle operazioni senza modifiche rispetto alle operazioni inserite nel WP0.

Operazione	Tipo	Frequenza
Operazione 1: inserisci i dati relativi ad un post dell'influencer con la	I	1500 al giorno

relativa panoramica ed i relativi hashtag utilizzati		
Operazione 2: inserisci i dati relativi al profilo dell'utente	I	1500 al giorno
Operazione 3: trovare le informazioni degli utenti che seguono un determinato influencer	I	300 al giorno
Operazione 4: stampa la panoramica per un singolo post di un influencer	I	1500 al giorno
Operazione 5: aggiorna i dati relativi ai profili degli utenti, ai post degli influencer e alle relative panoramiche rispetto alla data di riferimento	I	1 al mese
Operazione 6: identifica il post più influente basandosi sulla somma di reazioni e commenti presenti nella rispettiva panoramica	B	1 al mese
Operazione 7: stampa il numero di hashtag e di hashtag follower all'interno di ogni post di un utente	I	600 al giorno

Tabella 7. Tavola delle operazioni

4.2. Analisi delle ridondanze

[Indicare le ridondanze presenti e che saranno analizzate. Si ricorda di non eccedere con i dati ridondanti nel modello concettuale ma di considerare solo le ridondanze ritenute utili in fase di progettazione concettuale. Per ogni ridondanza effettuare l'analisi delle operazioni.]

- **Ridondanza 1: numero di hashtag (Post influencer).** Il numero di hashtag presenti in un post si ottiene contando il numero di occorrenze dell'associazione "P-C" (oppure dell'associazione C-H o dell'entità caratterizzato) a cui partecipa il post dell'influencer.
TIPO: Attributo Derivabile da conteggio di occorrenze.
- **Ridondanza 2: numero post (INFLUENCER).** Il numero di post pubblicati dagli influencer si ottiene contando il numero di occorrenze dell'associazione "pubblica" di utente.
TIPO: Attributo Derivabile da conteggio di occorrenze

4.2.1. Analisi della ridondanza 1: Numero post

[Analizzare il costo delle operazioni che coinvolgono la ridondanza (SOLO QUELLE CHE COINVOLGONO LA RIDONDANZA) in presenza e assenza dell'attributo.]

- **Operazione 1: inserisci i dati relativi ad un post dell'influencer con la relativa panoramica ed i relativi hashtag utilizzati**
- *L'operazione 1 richiede un accesso in scrittura all'entità 'Post' (per memorizzare un nuovo post), 3 accessi in scrittura alla relazione P-C (per ogni post in media vengono memorizzati 3 hashtag) e di conseguenza 3 accessi in scrittura all'entità 'Caratterizzato', 3 accessi in*

scrittura alla relazione 'C-H' e 3 accessi in scrittura all'entità 'Hashtag'. Successivamente vi è 1 accesso in scrittura alla relazione 'composizione' (per memorizzare una nuova coppia post influencer- panoramica post), 1 accesso in scrittura all'entità 'panoramica post' (per memorizzare la panoramica di un post), 1 accesso in scrittura alla relazione 'pubblicato' (per memorizzare la coppia post-influencer), 1 accesso in lettura all'entità 'Utente LinkedIn' (per cercare l'influencer di interesse), 1 accesso in lettura (per cercare l'influencer di interesse) ed 1 in scrittura (per incrementare di uno il numero di post dell'occorrenza dell'influencer). La differenza rispetto all'operazione senza ridondanze è che in quest'ultima non sono presenti accessi all'entità 'Utente LinkedIn' ed 'Influencer'. Analizziamo in media 5 post per ogni influencer

Con ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Post Influencer	E	1	S
P-C	R	3	S
Caratterizzato	E	3	S
C-H	R	3	S
hashtag	E	3	S
Composizione	R	1	S
Panoramica post	E	1	S
Pubblicato	R	1	S
Utente LinkedIn	E	1	L
Influencer	E	1	L
Influencer	E	1	S

Senza ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Post Influencer	E	1	S
P-C	R	3	S
Caratterizzato	E	3	S
C-H	R	3	S
hashtag	E	3	S
Composizione	R	1	S
Panoramica post	E	1	S
Pubblicato	R	1	S

Operazione 2: inserisci i dati relativi al profilo dell'utente (in media 15000 volte al giorno);

In questo caso andiamo a considerare l'inserimento dei soli influencer, in quanto solo in tale situazione viene coinvolta la ridondanza.

Con ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Utente LinkedIn	E	1	S
Influencer	E	1	S

Senza ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Utente LinkedIn	E	1	S
Influencer	E	1	S

- **Operazione 5: aggiorna i dati relativi ai profili degli utenti, ai post degli influencer e alle relative panoramiche rispetto alla data di riferimento**

Tale operazione non viene considerata in quanto si tratta di un'operazione poco frequente.

4.2.1.1. Valutazione della ridondanza 1

Commentare l'analisi e indicare la decisione presa considerando il costo delle operazioni. Ad esempio:

Dopo aver analizzato le operazioni che coinvolgono la ridondanza si osserva che, con il carico considerato:

- In presenza di ridondanza il costo delle operazioni è di circa $(1500 \cdot 36(\text{Operazione 1}) + 15000 \cdot 4(\text{Operazione 2})) = 114000$ accessi giornalieri.
- L'occupazione di memoria è di circa 40.000 (assumendo che sono necessari 4 byte per ogni occorrenza di influencer per memorizzare il numero di post pubblicati) byte.
- In assenza di ridondanza il costo delle operazioni è di 108000 accessi giornalieri.

Pertanto, si decide di togliere la ridondanza in quanto aumenta il numero di accessi.

4.2.2. Analisi della ridondanza 2: Numero hashtag

- **Operazione 1: inserisci i dati relativi ad un post dell'influencer con la relativa panoramica ed i relativi hashtag utilizzati**

L'operazione 1 richiede un accesso in scrittura all'entità 'Post' (per memorizzare un nuovo post), 3 accessi in scrittura alla relazione P-C (per ogni post in media vengono memorizzati 3 hashtag) e di conseguenza 3 accessi in scrittura all'entità 'Caratterizzato', 3 accessi in scrittura alla relazione 'C-H' e 3 accessi in scrittura all'entità 'Hashtag'. Successivamente vi sono 3 accessi in lettura alla relazione 'P-C' per calcolare quanti hashtag sono contenuti nel post di interesse ed 1 accesso in scrittura per l'incremento del numero di hashtag del post considerato. Inoltre vi è 1 accesso in scrittura alla relazione 'composizione' (per memorizzare una nuova coppia post influencer- panoramica post), 1 accesso in scrittura all'entità 'panoramica post' (per memorizzare la panoramica di un post), 1 accesso in scrittura alla relazione 'pubblicato' (per memorizzare la coppia post-influencer)

Per ogni post in media si considerano 3 hashtag utilizzati.

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Post Influencer	E	1	S
P-C	R	3	S
Caratterizzato	R	3	S
C-H	R	3	S
Hashtag	R	3	S
P-C	R	3	L
Post Influencer	E	1	S
Pubblicato	R	1	S
Composizione	R	1	S
Panoramica post	E	1	S

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Post Influencer	E	1	S
Pubblicato	R	1	S
P-C	R	3	S
Caratterizzato	R	3	S
C-H	R	3	S
Hashtag	R	3	S
Composizione	R	1	S
Panoramica post	E	1	S

- **Operazione 7: Stampa il numero di hashtag all' interno e di hashtag follower di ogni post di un utente**

Per ogni influencer si analizzano 5 post, si ipotizza che ogni post ha 5 hashtag

Con ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Utente LinkedIn	E	1	L
Pubblicato	R	5	L
Post Influencer	E	5	L

Senza ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Utente LinkedIn	E	1	L
Pubblicato	R	5	L
Post Influencer	E	5	L
P-C	R	15	L

4.2.2.1. Valutazione della ridondanza 2

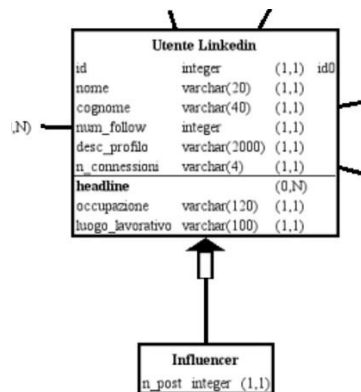
Dopo aver analizzato le operazioni che coinvolgono la ridondanza si osserva che, con il carico considerato:

- In presenza di ridondanza il costo delle operazioni è di circa $(1500 \cdot 37(\text{Operazione 1}) + 300 \cdot 11(\text{Operazione 7}))$ 58.800 accessi giornalieri.
- L'occupazione di memoria è di circa 200.000 (assumendo che sono necessari 4 byte per ogni occorrenza di post per memorizzare il numero di hashtag presenti nel post) byte.
- In assenza di ridondanza il costo delle operazioni è di 63600 accessi giornalieri.

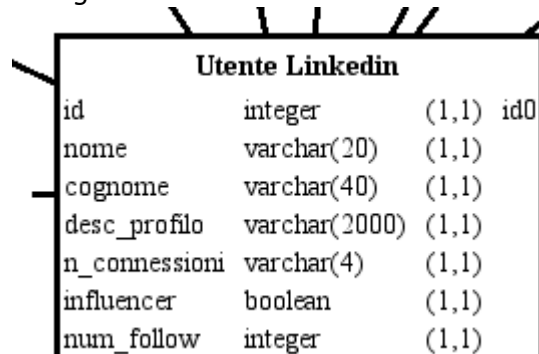
Pertanto, si decide di mantenere la ridondanza in quanto riduce il numero di accessi.

4.3. Eliminazione delle generalizzazioni

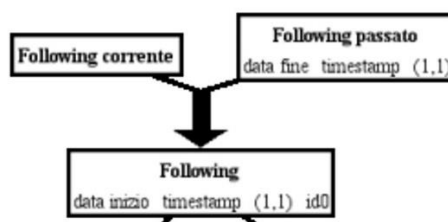
4.3.1. Generalizzazione *Utente LinkedIn*



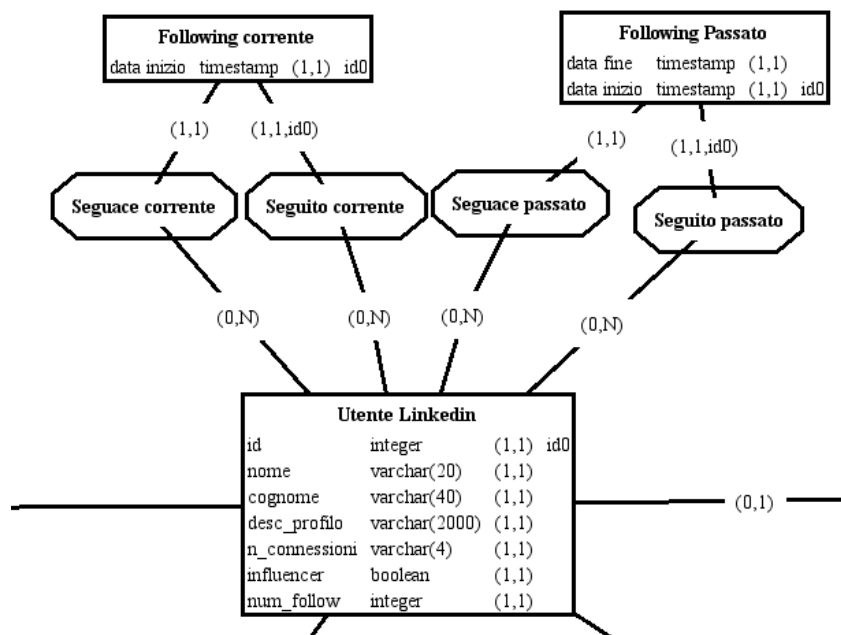
In questo caso abbiamo una generalizzazione parziale (ed ovviamente esclusiva). L'entità 'Influencer' viene eliminata e all'entità 'Utente LinkedIn' (entità padre) viene aggiunto un ulteriore attributo ('tipo') che serve a distinguere l'occorrenza di 'Utente LinkedIn', ovvero a discriminare se è o meno un influencer. La scelta risulta conveniente in quanto la maggior parte delle operazioni non presentano differenze tra le occorrenze dell' 'Utente LinkedIn' e di 'Influencer', inoltre l'entità 'Influencer' non presenta attributi aggiuntivi rispetto all'entità padre, dunque dopo aver accorpato l'entità figlia nell'entità padre, l'occorrenza 'Utente LinkedIn' (che non è un influencer) non presenterà attributi nulli derivati da 'Influencer'. Infine, la scelta ci assicura un numero minore di accessi in quanto le occorrenze e gli attributi non sono distribuiti in diverse entità.



4.3.2. Generalizzazione Following



In questo caso abbiamo una generalizzazione totale (ed ovviamente esclusiva). L'entità genitore 'Following' viene eliminata e, per la proprietà dell'ereditarietà, il suo identificatore, i suoi attributi e le associazioni a cui tale proprietà partecipa, vengono aggiunti alle entità figlie 'Following Corrente' e 'Following Passato'. Tale scelta viene adottata in quanto vi sono operazioni che fanno riferimento solo ad occorrenze di 'Following Passato' o di 'Following Corrente', ovvero vi è una distinzione tra le due entità. Il vantaggio è un risparmio di memoria in quanto non vi sono degli attributi nulli, a differenza di come poteva essere se si fosse utilizzata la stessa strategia adottata per la generalizzazione 'Utente LinkedIn'. Un ulteriore vantaggio è quello di non visitare prima l'occorrenza dell'entità padre per accedere poi ad una delle due entità figlie.



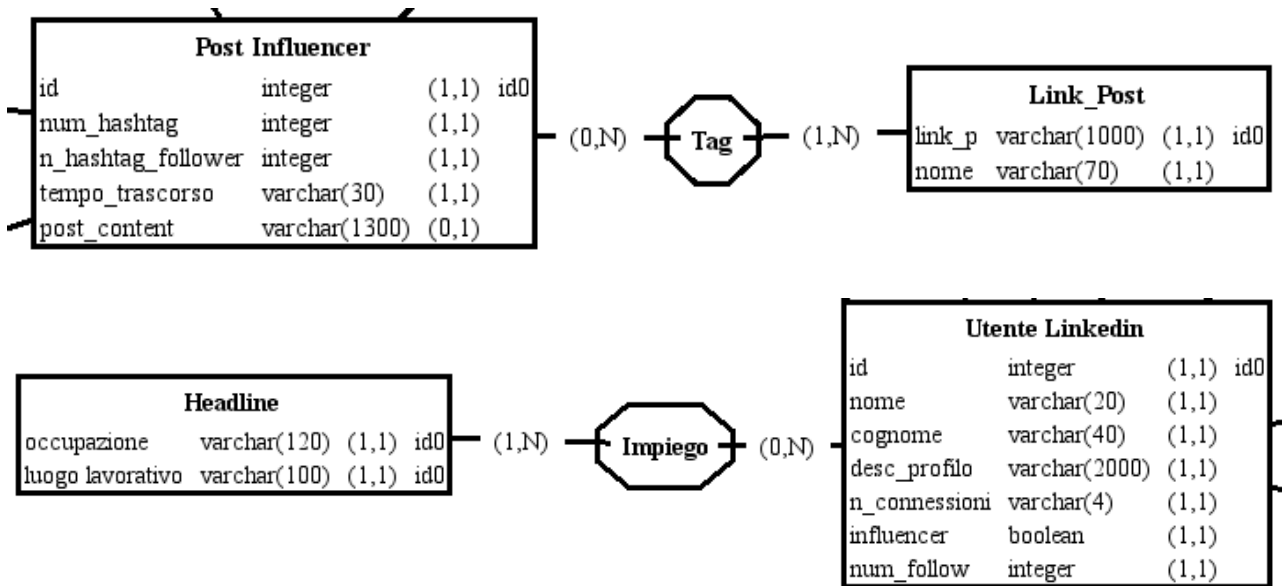
4.4. Partizionamento/Accorpamento Entità e Associazioni

In fase di ristrutturazione non sono stati effettuati accorpamenti/partizioni di entità o associazioni in quanto non ritenuti efficienti ai fini dell'ottimizzazione del progetto.

Inserire qui anche l'eventuale eliminazione di attributi multivalore.

All'interno dell'entità 'Post Influencer' vi è l'attributo 'link_post' che è compost e multivalore. Si crea dunque un'entità 'Link Post' con gli attributi 'link' e 'nome' che è in associazione con l'entità 'Post Influencer'.

Il secondo attributo multivalore è il 'luogo lavorativo' all'interno dell'entità 'Impiego'. In questo caso si crea una nuova entità 'Luogo lavorativo' in associazione con l'entità 'Impiego'.



4.5. Scelta degli identificatori principali

Indicare le scelte degli identificatori principali, in presenza di più identificatori o rispetto all'introduzione di nuovi identificatori. Motivare la scelta (ad esempio per avere un identificatore più semplice, etc.)

I sistemi di gestione di basi di dati richiedono generalmente di specificare una chiave primaria sulla quale vengono costruite automaticamente delle strutture ausiliarie, dette indici, per il reperimento efficiente di dati. Nel caso in cui vi sono entità con più identificatori, bisogna decidere quale di questi verrà utilizzato come chiave primaria a seconda di diversi criteri quali assenza di opzionalità, semplicità ed utilizzo nelle operazioni più frequenti o importanti.

Le seguenti entità hanno un unico identificativo, dunque verrà scelto quest'ultimo come identificatore principale. A sostegno di ciò tali identificativi non sono opzionali, sono utilizzati nelle operazioni più frequenti ed importanti ed inoltre tali identificativi sono composti da pochi attributi. Abbiamo:

Entità: Type-media – Identificatore principale: nome;

Entità: Media – Identificatore principale: url;

Entità: Post Influencer – Identificatore principale: id e Utente LinkedIn(esterno);

Entità: Caratterizzato – Identificatore principale: codice;

Entità: Panoramica Post – Identificatore principale: post influencer(esterno);

Entità: Link Post – Identificatore principale: link;

Entità: Utente LinkedIn – Identificatore principale: id;

Entità: Following Corrente – Identificatore principale: data inizio e Utente LinkedIn(esterno);

Entità: Following Passato – Identificatore principale: data inizio e Utente LinkedIn(esterno);

Entità: Paese – Identificatore principale: nome;

Entità: Regione – Identificatore principale: nome;

Entità: Città – Identificatore principale: nome;

Entità: Lavoro – Identificatore principale: codice;

Entità: Impiego – Identificatore principale: occupazione;

Entità: Luogo Lavorativo – Identificatore principale: nome;

Inoltre, per l'entità 'hashtag' vi sono due identificativi, l'attributo 'nome' e l'attributo 'link'. La scelta per l'identificatore principale ricade sull' attributo nome in quanto ha una dimensione minore rispetto al link ed inoltre il nome dell'hashtag viene utilizzato dalle operazioni più frequenti.

4.6. Schema ristrutturato finale

Inserire lo schema ristrutturato finale

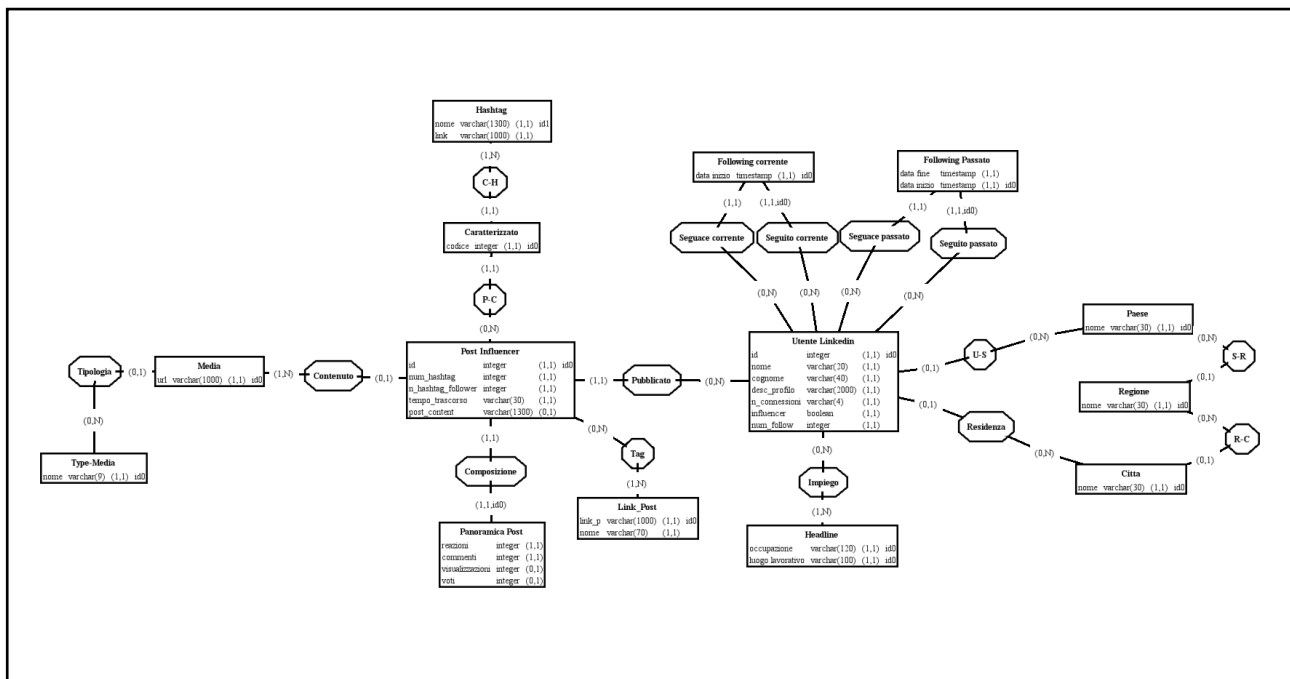


Figura 4. Schema ER Ristrutturato

4.7. Schema logico

MEDIA (url, type-media)

TYPE MEDIA (nome)

POST (id, utente, num_hashtag, n_hashtag_follower, tempo trascorso, contenuto post, media)

PANORAMICA POST (post, reazioni, commenti, visualizzazioni, voti)

CARATTERIZZATO (codice, hashtag, post)

HASHTAG (nome, link)

TAG (id post, link_post)

LINK_POST (link, nome)

UTENTE LINKEDIN (id, nome, cognome, desc_profilo, n_conessioni, num_follower, influencer)

FOLLOWING CORRENTE (seguito, data inizio, seguace)

FOLLOWING PASSATO (data inizio, seguito, data fine, seguace)

IMPIEGO (id utente, luogo lavorativo headline, occupazione headline)

HEADLINE (occupazione, luogo lavorativo)

U-S (utente, paese)

RESIDENZA (utente, città)

PAESE (nome)

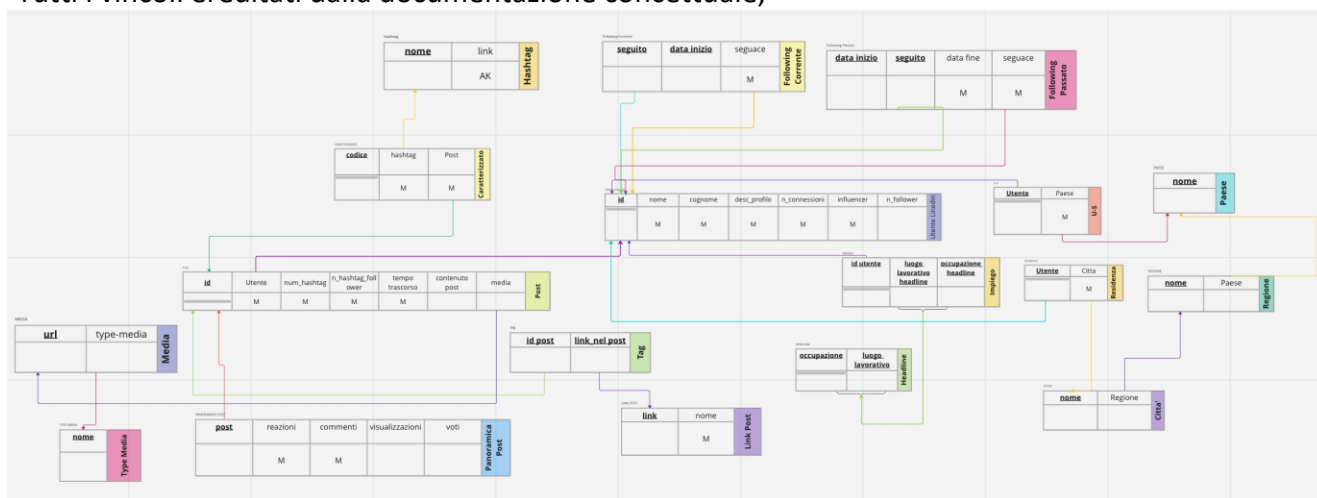
REGIONE (nome, paese)

CITTA' (nome, regione)

4.8. Documentazione dello schema logico

Vincoli logici:

- Un media deve essere contenuto almeno in un post;
- Un hashtag deve essere contenuto almeno in una tabella Caratterizzato;
- Un link post deve avere almeno un tag;
- Una headline deve avere almeno un impiego;
- Numero hashtag è ridondante;
- Tutti i vincoli ereditati dalla documentazione concettuale;



5. Normalizzazione

Workpackage	Task	Responsabile
WP3	Normalizzazione	Cerino Marco

Verifica della forma normale, identificando ed analizzando le dipendenze funzionali.

Se necessario e se possibile, effettuare le decomposizioni per normalizzare.

*Si precisa che tutte le tabelle sono garantite di essere almeno in prima forma normale. Questo perché durante la fase di ristrutturazione della progettazione logica, svolta precedentemente, vengono eliminati gli attributi composti, multivalore o composti e multivalore. Pertanto, il dominio degli attributi è limitato solo a valori atomici. Detto questo, **per verificare la seconda forma normale, si presume che il requisito della prima forma normale sia già soddisfatto**. Inoltre, si specifica che, per la verifica delle forme normali delle relazioni, si considerano solo dipendenze funzionali non banali, complete ed espresse nella forma minima. Per quanto riguarda la terza forma normale o la forma Boyce and Codd, il requisito in comune è che la relazione sia già in seconda forma normale. Una volta che questo requisito è verificato, non sarà necessario ripeterlo per determinare una delle due forme (forma normale o Boyce and Codd) della relazione di interesse.*

Type-media:

Ha un singolo attributo come chiave, dunque è sicuramente in seconda forma normale. Inoltre, avendo un singolo attributo e non essendoci dipendenze funzionali rilevanti, è in forma Boyce and Codd.

Media:

Ha un singolo attributo come chiave, dunque è sicuramente in seconda forma normale. Ha un 'unica dipendenza funzionale:

url -> type media;

Essendo url la chiave, possiamo affermare che tale relazione è in forma di Boyce and Codd.

Hashtag:

Ha un singolo attributo come chiave, dunque è sicuramente in seconda forma normale. Ha due dipendenze funzionali:

nome -> link;

link -> nome;

Dato che in tali dipendenze funzionali a sinistra vi è sempre una superchiave(nel nostro caso chiave) o una chiave candidata la relazione è in forma di Boyce and Codd.

Caratterizzato:

Ha un singolo attributo come chiave(codice), dunque è sicuramente in seconda forma normale. Inoltre, è in forma di Boyce and Codd in quanto abbiamo le due sole dipendenze funzionali:

codice -> hashtag;(sicuramente perché codice chiave);

codice -> post;(sicuramente perché codice chiave);

dove a sinistra abbiamo sempre la superchiave(nel nostro caso la chiave).

Nella nostra realtà, un post può contenere più volte lo stesso hashtag, dunque:

post, hashtag non implica codice;

Post:

Ha un singolo attributo come chiave(id), dunque sicuramente è in seconda forma normale.

Inoltre, è in forma di Boyce and Codd in quanto abbiamo le seguenti dipendenze funzionali:

id -> utente;

id -> num_hashtag;

id -> n_hashtag_follower;

id -> tempo trascorso;

id -> contenuto post;

id -> media;

dove a sinistra abbiamo sempre la superchiave(nel nostro caso la chiave).

Nella nostra realtà, il contenuto del post non contiene gli hashtag, dunque:

contenuto post non implica numero hashtag;

contenuto post non implica n_hashtag_follower;

In tal caso non vi è rischio che la relazione non sia in terza forma normale.

Tag:

è sicuramente in seconda forma normale in quanto non ha attributi non primi.

Inoltre, non avendo dipendenze funzionali non banali ed essendo in prima forma normale, tale relazione è in forma di Boyce and Codd.

Link Post:

è sicuramente in seconda forma normale in quanto ha un singolo attributo come chiave(link).

Inoltre, la relazione è in forma di Boyce and Codd in quanto sussiste solo la seguente dipendenza funzionale:

link -> nome

dove a sinistra vi è la superchiave (nel nostro caso la chiave).

Non sussiste la dipendenza funzionale inversa in quanto, per diversi profili, possiamo avere lo stesso nome.

Panoramica post:

Ha un singolo attributo come chiave(post), dunque sicuramente è in seconda forma normale.

Inoltre, è in forma di Boyce and Codd in quanto abbiamo le seguenti dipendenze funzionali:

post -> reazioni;

post -> commenti;

post -> visualizzazioni;

post -> voti;

dove a sinistra abbiamo sempre la superchiave (nel nostro caso la chiave).

Following corrente:

è sicuramente in seconda forma normale dato che, ogni attributo non primo, dipende funzionalmente in maniera piena dalla chiave:

seguito, data inizio -> seguace;

dove seguito e data inizio sono la chiave.

Inoltre:

seguito, seguace -> data inizio;

Dunque, essendo che, nelle dipendenze funzionali non banali, piene e minimali a sinistra abbiamo sempre la chiave oppure a destra abbiamo un attributo primo, ed essendo in seconda forma normale, possiamo dire che tale relazione è in terza forma normale.

Si precisa che:

seguace, data inizio non implica seguito

in quanto si considera il caso in cui, uno stesso profilo utente, presente su più dispositivi, nello stesso momento segua più utenti.

Following passato:

è sicuramente in seconda forma normale dato che, ogni attributo non primo, dipende funzionalmente in maniera piena dalla chiave:

data inizio, seguito -> data fine;

data inizio, seguito -> seguace;

Inoltre:

seguito, seguace, data fine -> data inizio (in quanto nello stesso istante, è impensabile che un utente segua una persona, la smetta di seguire, la risegua e la smetta di seguire nuovamente).

Dunque, essendo che, nelle dipendenze funzionali non banali, piene e minimali a sinistra abbiamo sempre la chiave oppure a destra abbiamo un attributo primo, ed essendo in seconda forma normale, possiamo dire che tale relazione è in terza forma normale.

Si precisa che:

Data inizio, data fine, seguace non implica seguito

In quanto consideriamo l'ipotesi che un utente possa smettere di seguire nello stesso istante più utenti da dispositivi diversi.

Utente LinkedIn:

Ha un singolo attributo come chiave(id), dunque sicuramente è in seconda forma normale.

Inoltre, è in forma di Boyce and Codd in quanto abbiamo le seguenti dipendenze funzionali:

id -> nome;

id -> cognome;

id -> desc_profilo;

id -> n_conessioni;

id -> influencer;

id -> n_follower

dove a sinistra abbiamo sempre e solo la superchiave(nel nostro caso la chiave).

Si precisa che diversi utenti possono avere stesso nome e cognome, quindi non sussistono altre dipendenze funzionali rilevanti.

Impiego:

è sicuramente in seconda forma normale in quanto non vi sono attributi non primi. Dunque non vi sono dipendenze funzionali che trattano quest'ultimi.

Dato che non sussistono dipendenze funzionali non banali, allora tale relazione risulta sicuramente in forma di Boyce and Codd.

Si precisa che un utente può avere più impieghi, in particolare un caso particolare può essere che ricopre la stessa posizione ma in aziende diverse.

Headline:

è sicuramente in seconda forma normale in quanto non vi sono attributi non primi. Dunque, non vi sono dipendenze funzionali che trattano quest' ultimi.

Dato che non sussistono dipendenze funzionali non banali, allora tale relazione risulta sicuramente in forma di Boyce and Codd.

U-S:

Ha un singolo attributo come chiave (Utente), dunque sicuramente è in seconda forma normale.

È anche in forma di Boyce and Codd in quanto vi è un'unica dipendenza funzionale:

utente -> paese

dove utente è superchiave(nel nostro caso chiave).

Paese:

Ha un singolo attributo come chiave(nome), dunque sicuramente è in seconda forma normale.

È anche in forma di Boyce and Codd in quanto non sussistono dipendenze funzionali rilevanti.

Residenza:

Ha un singolo attributo come chiave(utente), dunque sicuramente è in seconda forma normale.

È anche in forma di Boyce and Codd in quanto vi è un'unica dipendenza funzionale:

utente -> città

dove utente è superchiave(nel nostro caso una chiave).

Regione:

Ha un singolo attributo come chiave(nome), dunque sicuramente è in seconda forma normale.

È anche in forma di Boyce and Codd in quanto vi è un'unica dipendenza funzionale:

nome -> paese

dove nome è superchiave(nel nostro caso una chiave).

Città:

Ha un singolo attributo come chiave(nome), dunque sicuramente è in seconda forma normale.

È anche in forma di Boyce and Codd in quanto vi è un'unica dipendenza funzionale:

nome -> regione

dove nome è superchiave(nel nostro caso una chiave).

Dopo un'attenta analisi, si può affermare che tutte le relazioni di interesse sono almeno in terza forma normale e che la maggior parte soddisfano la forma di Boyce and Codd. Questo indica che lo schema è già ben strutturato dal punto di vista della normalizzazione e che ogni relazione presenta concetti omogenei. Non è necessario quindi procedere con ulteriori processi di decomposizione per migliorare la struttura delle relazioni. Questo è un risultato positivo, poiché indica che le relazioni sono coerenti e non presentano anomalie e ridondanze indesiderate.

6. Script Creazione e Popolamento Database

Workpackage	Task	Responsabile
WP2	SQL: Script creazione e popolamento	De Bartolomeis Mattia

Inserire lo script di creazione del database e lo script per il popolamento con dati di esempio (può essere anche un solo script che contiene sia creazione che popolamento) del database progettato. Gli script devono essere incollati qui (come testo o come immagine) e poi inseriti nella consegna finale come file di testo .sql

Note: Per la gestione dei vincoli dovuti dalle cardinalità (1, n) e [(0,1) o (1,1)] tra le relazioni, viene implementato il vincolo referenziale come deferred e creato un trigger. Si precisa che, per scopo esemplificativo, è stato creato il trigger solo per la gestione dei vincoli sulle relazioni hashtag e caratterizzato. L'inserimento prevede la creazione di una transazione che coinvolge le due tabelle legate dal vincolo, tuttavia SQL considera come un'unica transazione l'intero script.

```
DROP TABLE IF EXISTS type_media CASCADE;
DROP TABLE IF EXISTS media CASCADE;
DROP TABLE IF EXISTS hashtag CASCADE;
DROP TABLE IF EXISTS caratterizzato CASCADE;
DROP TABLE IF EXISTS post CASCADE;
DROP TABLE IF EXISTS panoramica_post CASCADE;
DROP TABLE IF EXISTS following_corrente CASCADE;
DROP TABLE IF EXISTS utente_linkedin CASCADE;
DROP TABLE IF EXISTS following_passato CASCADE;
DROP TABLE IF EXISTS link_post CASCADE;
DROP TABLE IF EXISTS tag CASCADE;
DROP TABLE IF EXISTS headline CASCADE;
DROP TABLE IF EXISTS impiego CASCADE;
DROP TABLE IF EXISTS paese CASCADE;
DROP TABLE IF EXISTS regione CASCADE;
DROP TABLE IF EXISTS citta CASCADE;
DROP TABLE IF EXISTS u_s CASCADE;
DROP TABLE IF EXISTS residenza CASCADE;
DROP DOMAIN IF EXISTS Domain_num_hash;
DROP DOMAIN IF EXISTS Domain_positive;
DROP DOMAIN IF EXISTS Domain_hashtag;
DROP DOMAIN IF EXISTS Domain_date;
DROP DOMAIN IF EXISTS Domain_link;
```

```
Create domain Domain_num_hash AS Integer
CHECK(value >= 0 and value <= 30);
```

```
Create domain Domain_hashtag AS VARCHAR(100)
CHECK(value like '%#');
```

```
Create domain Domain_positive AS Integer
Check(value >= 0);
```

```
Create domain Domain_date AS VARCHAR(30)
Check(value like '%ago' or value like '%after');
```

```
Create domain Domain_link AS VARCHAR(1000)
Check(value like 'http://%' or value like 'https://%');
```

```
create table type_media(
    nome varchar(30) primary key
);
create table media(
    url Domain_link primary key,
    type_media varchar(30),
    constraint fk_type foreign key (type_media) references type_media(nome) on update
restrict on delete restrict
);
```

```
create table hashtag(
    nome Domain_hashtag primary key,
    link Domain_link unique not null
);
```

```
CREATE TABLE utente_linkedin(
    id integer primary key,
    nome varchar(20) not null ,
    cognome varchar(40) not null,
    desc_profilo varchar(2000) not null,
    n_conessioni varchar(30) not null,
    num_follower Domain_positive not null,
    influencer boolean not null
);
```

```
create table post(
    id integer primary key,
    utente integer not null,
    num_hashtag Domain_num_hash not null,
    n_hashtag_follower Domain_positive not null ,
    tempo_trascorso Domain_date not null,
    contenuto_post varchar(1300),
    media Domain_link,
    constraint fk_media foreign key (media) references media(url) on update cascade on delete
cascade deferrable initially deferred,
    constraint fk_utente_linkedin foreign key (utente) references utente_linkedin(id) on update
cascade on delete cascade
);
```



```
create table caratterizzato(  
    codice integer primary key,  
    hashtag varchar(30) not null,  
    post integer not null,  
    constraint fk_post foreign key (post) references post(id) on update cascade on delete  
cascade,  
    constraint fk_hash foreign key(hashtag) references hashtag(nome) on update cascade on  
delete cascade deferrable initially deferred  
);
```

```
create table panoramica_post(  
    post integer primary key,  
    reazioni Domain_positive not null,  
    commenti Domain_positive not null,  
    visualizzazioni Domain_positive,  
    voti Domain_positive,  
    constraint fk_pan_post foreign key (post) references post(id) on update cascade on delete  
cascade  
);
```

```
CREATE TABLE link_post(  
    link_p Domain_link primary key,  
    nome varchar(30) not null  
);
```

```
create table tag(  
    post integer,  
    link_nel_post varchar(1000),  
    Constraint fk_id foreign key (post) references post(id) on update cascade on delete cascade,  
    Constraint fk_link FOREIGN KEY (link_nel_post) references link_post(link_p) on update  
cascade on delete cascade deferrable initially deferred,  
    PRIMARY KEY(post,link_nel_post)  
);
```

```
create table following_corrente(  
    seguito integer,  
    data_inizio timestamp,  
    seguace integer not null,  
    primary key(seguito,data_inizio),  
    constraint fk_seguito FOREIGN KEY (seguito) references utente_linkedin(id) on update  
cascade on delete cascade,  
    constraint fk_seguace FOREIGN KEY (seguace) references utente_linkedin(id) on update  
cascade on delete cascade  
);
```

```
CREATE TABLE following_passato (  
    data_inizio timestamp,  
    seguito INTEGER,  
    data_fine timestamp,
```

```
    seguace INTEGER not null,
    PRIMARY KEY ( data_inizio, seguito ),
    constraint fk_seguito_pas FOREIGN KEY (seguito) references utente_linkedin(id) on update
cascade on delete cascade,
    constraint fk_seguace_pas FOREIGN KEY (seguace) references utente_linkedin(id) on
update cascade on delete cascade
);
CREATE TABLE headline(
    luogo_lavorativo VARCHAR(30),
    occupazione VARCHAR(30),
    PRIMARY KEY (luogo_lavorativo, occupazione)
);

create table impiego(
    id_utente INTEGER,
    luogo_lavorativo_headline VARCHAR(30),
    occupazione_headline VARCHAR(30),
    PRIMARY KEY (id_utente, luogo_lavorativo_headline, occupazione_headline),
    Constraint fk_headline FOREIGN KEY (luogo_lavorativo_headline, occupazione_headline)
references headline(luogo_lavorativo, occupazione) on update cascade on delete cascade
deferrable initially deferred
);
CREATE TABLE paese(
    nome VARCHAR(30) PRIMARY KEY
);
CREATE TABLE regione(
    nome VARCHAR(30) PRIMARY KEY,
    paese VARCHAR(30),
    constraint fk_paese_regione foreign key(paese) references paese(nome) on update cascade
on delete set null
);

CREATE TABLE citta(
    nome VARCHAR(30) PRIMARY KEY,
    regione VARCHAR(30),
    constraint fk_regione foreign key(regione) references regione(nome) on update cascade on
delete set null
);

CREATE TABLE u_s(
    utente INTEGER PRIMARY KEY,
    paese VARCHAR(30) not null,
    constraint fk_utente_stato foreign key(utente) references utente_linkedin(id) on update
cascade on delete cascade,
    constraint fk_paese foreign key(paese) references paese(nome) on update cascade on
delete restrict
);
```

```
CREATE TABLE residenza(  
    utente INTEGER PRIMARY KEY,  
    citta VARCHAR(30) not null,  
    constraint fk_utente_residenza foreign key(utente) references utente_linkedin(id) on  
update cascade on delete cascade,  
    constraint fk_citta foreign key(citta) references citta(nome) on update cascade on delete  
restrict  
);
```

```
--trigger per vincolo da hashtag a caratterizzatoo  
CREATE OR REPLACE FUNCTION AtLeastONE() returns trigger as $$  
BEGIN  
if (exists (select nome from hashtag  
where nome not in (select hashtag from caratterizzatoo))) then  
RAISE EXCEPTION 'ERRORE 1';  
end if;  
return NEW;  
END $$ LANGUAGE plpgsql;
```

```
create trigger AtLeastONE  
after insert on hashtag  
for each row execute procedure AtLeastONE();
```

```
create trigger AtLeastONE  
after delete on caratterizzatoo  
for each row execute procedure AtLeastONE();
```

--trigger 1

```
CREATE OR REPLACE FUNCTION update_panoramica_voti() RETURNS TRIGGER AS $$  
BEGIN  
    IF NEW.voti IS NOT NULL THEN  
        -- Controlla se il media associato al post non è un "sondaggio"  
        IF (SELECT type_media FROM media WHERE url = (SELECT media FROM post WHERE id =  
NEW.post)) <> 'sondaggio' THEN  
            RAISE EXCEPTION 'Se il post non ha come media un sondaggio, i voti devono essere NULL.';  
        END IF;  
    END IF;  
  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_update_panoramica_voti  
AFTER INSERT OR UPDATE ON panoramica_post  
FOR EACH ROW
```

```
EXECUTE FUNCTION update_panoramica_voti();
```

--trigger dato ridondante

```
CREATE OR REPLACE FUNCTION caratterizzato() RETURNS TRIGGER AS $$
```

```
begin
```

```
if (new is not null) then--insert o update
```

```
    update post set num_hashtag = 1 where id = new.post;
```

```
end if;
```

```
if (old is not null) then
```

```
    update post set num_hashtag = 1 where id = old.post;
```

```
end if;
```

```
return null;
```

```
End;$$ language plpgsql;
```

create trigger caratterizzato

after insert or delete or update of hashtag on caratterizzato

for each row

execute procedure caratterizzato();

--trigger dato ridondante parte 2

```
CREATE OR REPLACE FUNCTION protegginumerohashtag() RETURNS TRIGGER AS $$
```

```
begin
```

```
    select count(*) into new.num_hashtag
```

```
    from caratterizzato where post = new.id;
```

```
    return new;
```

```
End;$$ LANGUAGE plpgsql;
```

create trigger protegginumerohashtag

before insert or update of num_hashtag on post

for each row

execute procedure protegginumerohashtag();

--INSERIMENTO UTENTE DAL DATASET

```
INSERT INTO utente_linkedin (id, nome, cognome, desc_profilo, n_connessioni, num_follower,
influencer) VALUES (6, 'Nicholas', 'Wyman', 'Nicholas Wyman for the past 25 years has shone a
light on the mismatch between employers seeking qualified staff yet the hordes of people who
remain unemployed. Nicks advocated for and created solutions on workforce issues of the 21st
Century. He is an international expert, particularly zeroing in on Apprenticeship and CTE education
and training models in the US, UK, Germany, Switzerland and Australia. He taps out opinion pieces
for Forbes, Quartz and Fortune, makes appearances on National Public Radio, he has notched top
education writer on LinkedIn globally and recognised for his leadership in education in Boss
Magazine awards. He wrote the international award-winning book, Job U. Nick holds an MBA, has
```

studied at Harvard Business School and the Kennedy School of Government plus is a Winston Churchill Fellow. He's a third-generation writer, who started his own career earning a trade in commercial cookery, was awarded Australian Apprentice of the Year in 1988 and captaining Australia's gold-medal-winning Culinary Youth Team before becoming a chef specializing in fish at the London Ritz Hotel.', '500+', 6484, true);

```
INSERT INTO utente_linkedin (id, nome, cognome, desc_profilo, n_conessioni, num_follower, influencer) VALUES (1, 'Mario', 'Rossi', 'Sviluppatore software appassionato di tecnologia.', '500+', 20000, true);
```

```
INSERT INTO utente_linkedin (id, nome, cognome, desc_profilo, n_conessioni, num_follower, influencer) VALUES (2, 'Laura', 'Verdi', 'Esperta di marketing digitale con focus su social media.', '200', 2000, true);
```

```
INSERT INTO utente_linkedin (id, nome, cognome, desc_profilo, n_conessioni, num_follower, influencer) VALUES (3, 'Luca', 'Bianchi', 'Appassionato di fotografia e video editing.', '500+', 500, true);
```

```
INSERT INTO utente_linkedin (id, nome, cognome, desc_profilo, n_conessioni, num_follower, influencer) VALUES (4, 'Giulia', 'Gialli', 'Globetrotter e autrice di articoli di viaggio.', '300', 100, true);
```

```
INSERT INTO utente_linkedin (id, nome, cognome, desc_profilo, n_conessioni, num_follower, influencer) VALUES (5, 'Andrea', 'Neri', 'Esperto di finanza e consulente aziendale.', '500+', 3000, true);
```

--INSERIMENTO POST DAL DATASET

```
INSERT INTO post (id, utente, num_hashtag, n_hashtag_follower, tempo_trascorso, contenuto_post, media) VALUES (7, 6, 4, 0, '1 day ago', 'Robert Lerman writes that achieving a healthy future of work requires employees to build skills that help them attain productive and rewarding careers. He notes - "one of the most cost-effective ways to do this is through apprenticeship, which helps workers master occupations and gain professional identity and pride". Couldnt agree more! #workbasedlearning #USA #apprenticeship Read the article on #UrbanWire Urban Institute ', 'https://www.urban.org/urban-wire/its-time-modernize-american-apprenticeship-system');
```

```
INSERT INTO post (id, utente, num_hashtag, n_hashtag_follower, tempo_trascorso, contenuto_post, media) VALUES (1, 1, 0, 500, '2 ore ago', 'Nuovo progetto di sviluppo software!', 'http://www.example.com/image1.jpg');
```

```
INSERT INTO post (id, utente, num_hashtag, n_hashtag_follower, tempo_trascorso, contenuto_post, media) VALUES (2, 2, 0, 1000, '3 giorni ago', 'Strategie di marketing per il 2023.', 'http://www.example.com/video1.mp4');
```

```
INSERT INTO post (id, utente, num_hashtag, n_hashtag_follower, tempo_trascorso,
contenuto_post, media) VALUES (3, 3, 0, 200, '1 settimana ago', 'Consigli per la fotografia
notturna.', 'http://www.example.com/image1.jpg');
INSERT INTO post (id, utente, num_hashtag, n_hashtag_follower, tempo_trascorso,
contenuto_post, media) VALUES (4, 4, 0, 50, '2 settimane ago', 'Viaggio avventuroso in Sud
America.', 'http://www.example.com/article1.html');
INSERT INTO post (id, utente, num_hashtag, n_hashtag_follower, tempo_trascorso,
contenuto_post, media) VALUES (5, 5, 0, 1500, '1 mese ago', 'Strategie di investimento nel mercato
azionario.', 'http://www.example.com/poll1');
INSERT INTO post (id, utente, num_hashtag, n_hashtag_follower, tempo_trascorso,
contenuto_post, media) VALUES (6, 4, 0, 50, '2 settimane ago', 'Viaggio avventuroso in Sud
America.', 'http://www.example.com/article1.html');
```

```
INSERT INTO post (id, utente, num_hashtag, n_hashtag_follower, tempo_trascorso,
contenuto_post, media) VALUES (9, 5, 0, 300, '2 ore ago', 'Nuovo progetto di sviluppo software!',
'http://www.example.com/image1.jpg');
INSERT INTO post (id, utente, num_hashtag, n_hashtag_follower, tempo_trascorso,
contenuto_post, media) VALUES (8, 5, 0, 1000, '3 giorni ago', 'Strategie di marketing per il 2023.',
'http://www.example.com/video1.mp4');
```

```
INSERT INTO type_media (nome) VALUES ('immagine');
INSERT INTO type_media (nome) VALUES ('video');
INSERT INTO type_media (nome) VALUES ('documento');
INSERT INTO type_media (nome) VALUES ('articolo');
INSERT INTO type_media (nome) VALUES ('sondaggio');
```

--PER IL POPOLAMENTO DEL DATABASE, VERRANNO INSERITI DATI MANUALMENTE DA ANALISI
EFFETTUATE SU LINKEDIN E VERRA' INSERITO UN INFLUENCER CONSIDERATO NEL DATASET DI
RIFERIMENTO. (PRIMA RIGA DEL DATASET)

--INSERIMENTO MEDIA DAL DATASET

```
INSERT INTO media (url, type_media) VALUES ('https://www.urban.org/urban-wire/its-time-
modernize-american-apprenticeship-system', 'articolo');
```

```
INSERT INTO media (url, type_media) VALUES ('http://www.example.com/image1.jpg',
'immagine');
INSERT INTO media (url, type_media) VALUES ('http://www.example.com/video1.mp4', 'video');
INSERT INTO media (url, type_media) VALUES ('http://www.example.com/document1.pdf',
'documento');
INSERT INTO media (url, type_media) VALUES ('http://www.example.com/article1.html', 'articolo');
INSERT INTO media (url, type_media) VALUES ('http://www.example.com/poll1', 'sondaggio');
```

--INSERIMENTO CARATTERIZZATO DAL DATASET

INSERT INTO caratterizzato (codice, hashtag, post) VALUES (6, '#workbasedlearning', 7);

INSERT INTO caratterizzato (codice, hashtag, post) VALUES (7, '#USA', 7);

INSERT INTO caratterizzato (codice, hashtag, post) VALUES (8, '#apprenticeship', 7);

INSERT INTO caratterizzato (codice, hashtag, post) VALUES (9, '#UrbanWire', 7);

INSERT INTO caratterizzato (codice, hashtag, post) VALUES (1, '#technology', 1);

INSERT INTO caratterizzato (codice, hashtag, post) VALUES (2, '#marketing', 2);

INSERT INTO caratterizzato (codice, hashtag, post) VALUES (3, '#programming', 3);

INSERT INTO caratterizzato (codice, hashtag, post) VALUES (4, '#socialmedia', 4);

INSERT INTO caratterizzato (codice, hashtag, post) VALUES (5, '#business', 5);

--INSERIMENTO HASHTAG DAL DATASET

INSERT INTO hashtag (nome, link) VALUES ('#workbasedlearning',

'https://www.linkedin.com/feed/hashtag/?keywords=workbasedlearning&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A6765387069389967360');

INSERT INTO hashtag (nome, link) VALUES ('#USA',

'https://www.linkedin.com/feed/hashtag/?keywords=usa&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A6765387069389967360');

INSERT INTO hashtag (nome, link) VALUES ('#apprenticeship',

'https://www.linkedin.com/feed/hashtag/?keywords=apprenticeship&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A6765387069389967360');

INSERT INTO hashtag (nome, link) VALUES ('#UrbanWire',

'https://www.linkedin.com/feed/hashtag/?keywords=urbanwire&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A6765387069389967360

');

INSERT INTO hashtag (nome, link) VALUES ('#technology',

'http://www.example.com/hashtags/technology');

INSERT INTO hashtag (nome, link) VALUES ('#programming',

'http://www.example.com/hashtags/programming');

INSERT INTO hashtag (nome, link) VALUES ('#business',

'http://www.example.com/hashtags/business');

INSERT INTO hashtag (nome, link) VALUES ('#socialmedia',

'http://www.example.com/hashtags/socialmedia');

INSERT INTO hashtag (nome, link) VALUES ('#marketing',

'http://www.example.com/hashtags/marketing');

--INSERIMENTO PANORAMICA DAL DATASET

INSERT INTO panoramica_post (post, reazioni, commenti, visualizzazioni, voti) VALUES (7, 12, 1, null, null);

```
INSERT INTO panoramica_post (post, reazioni, commenti, visualizzazioni, voti) VALUES (1, 600, 50, 1000, null);
INSERT INTO panoramica_post (post, reazioni, commenti, visualizzazioni, voti) VALUES (2, 200, 100, 2000, null);
INSERT INTO panoramica_post (post, reazioni, commenti, visualizzazioni, voti) VALUES (3, 700, 20, 500, null);
INSERT INTO panoramica_post (post, reazioni, commenti, visualizzazioni, voti) VALUES (4, 30, 10, 300, null);
INSERT INTO panoramica_post (post, reazioni, commenti, visualizzazioni, voti) VALUES (5, 20, 700, 5000, 50);
```

```
INSERT INTO panoramica_post (post, reazioni, commenti, visualizzazioni, voti) VALUES (9, 20, 700, 5000, null);
INSERT INTO panoramica_post (post, reazioni, commenti, visualizzazioni, voti) VALUES (8, 20, 700, 200, null);
```

--INSERIMENTO TAG DAL DATASET

```
INSERT INTO tag (post, link_nel_post) VALUES (7, 'https://www.linkedin.com/in/ACoAAACy1HkBviRGLfLG__Jk8FRH2JY2rGg3nTU');
INSERT INTO tag (post, link_nel_post) VALUES (7, 'https://www.linkedin.com/company/urban-institute/');
INSERT INTO tag (post, link_nel_post) VALUES (1, 'http://www.example.com/post1');
INSERT INTO tag (post, link_nel_post) VALUES (2, 'http://www.example.com/post2');
INSERT INTO tag (post, link_nel_post) VALUES (3, 'http://www.example.com/post3');
INSERT INTO tag (post, link_nel_post) VALUES (4, 'http://www.example.com/post4');
INSERT INTO tag (post, link_nel_post) VALUES (5, 'http://www.example.com/post5');
```

--INSERIMENTO LINK POST DAL DATASET

```
INSERT INTO link_post (link_p, nome) VALUES ('https://www.linkedin.com/in/ACoAAACy1HkBviRGLfLG__Jk8FRH2JY2rGg3nTU', 'Robert Lerman');
INSERT INTO link_post (link_p, nome) VALUES ('https://www.linkedin.com/company/urban-institute/', 'Urban Institute');
```

```
INSERT INTO link_post (link_p, nome) VALUES ('http://www.example.com/post1', 'Articolo interessante');
INSERT INTO link_post (link_p, nome) VALUES ('http://www.example.com/post2', 'Video tutorial');
INSERT INTO link_post (link_p, nome) VALUES ('http://www.example.com/post3', 'Risultati del sondaggio');
INSERT INTO link_post (link_p, nome) VALUES ('http://www.example.com/post4', 'Ultimo aggiornamento');
```



```
INSERT INTO link_post (link_p, nome) VALUES ('http://www.example.com/post5', 'Guida completa');
```

```
--INSERIMENTO FOLLOWING_CORRENTE DAL DATASET  
--VUOTO
```

```
INSERT INTO following_corrente (seguito, data_inizio, seguace) VALUES (2, CURRENT_TIMESTAMP, 1);  
INSERT INTO following_corrente (seguito, data_inizio, seguace) VALUES (3, CURRENT_TIMESTAMP, 1);  
INSERT INTO following_corrente (seguito, data_inizio, seguace) VALUES (4, CURRENT_TIMESTAMP, 1);  
INSERT INTO following_corrente (seguito, data_inizio, seguace) VALUES (5, CURRENT_TIMESTAMP, 1);  
INSERT INTO following_corrente (seguito, data_inizio, seguace) VALUES (1, CURRENT_TIMESTAMP, 2);
```

```
--INSERIMENTO FOLLOWING_PASSATO DAL DATASET  
--VUOTO
```

```
INSERT INTO following_passato (data_inizio, seguito, data_fine, seguace) VALUES (CURRENT_TIMESTAMP, 1, '2023-05-01', 2);  
INSERT INTO following_passato (data_inizio, seguito, data_fine, seguace) VALUES (CURRENT_TIMESTAMP, 2, '2023-05-15', 3);  
INSERT INTO following_passato (data_inizio, seguito, data_fine, seguace) VALUES (CURRENT_TIMESTAMP, 3, '2023-05-10', 4);  
INSERT INTO following_passato (data_inizio, seguito, data_fine, seguace) VALUES (CURRENT_TIMESTAMP, 4, '2023-05-20', 5);  
INSERT INTO following_passato (data_inizio, seguito, data_fine, seguace) VALUES (CURRENT_TIMESTAMP, 5, '2023-05-05', 1);
```

```
--INSERIMENTO IMPIEGO DAL DATASET
```

```
INSERT INTO impiego (id_utente, luogo_lavorativo_headline, occupazione_headline) VALUES (6, 'IWSI Group', 'CEO');
```

```
INSERT INTO impiego (id_utente, luogo_lavorativo_headline, occupazione_headline) VALUES (1, 'Google', 'Software Engineer');  
INSERT INTO impiego (id_utente, luogo_lavorativo_headline, occupazione_headline) VALUES (2, 'Facebook', 'Digital Marketing Specialist');  
INSERT INTO impiego (id_utente, luogo_lavorativo_headline, occupazione_headline) VALUES (3, 'Microsoft', 'Data Analyst');
```

INSERT INTO impiego (id_utente, luogo_lavorativo_headline, occupazione_headline) VALUES (4, 'Amazon', 'Product Manager');

INSERT INTO impiego (id_utente, luogo_lavorativo_headline, occupazione_headline) VALUES (5, 'Apple', 'UX/UI Designer');

--INSERIMENTO HEADLINE DAL DATASET

INSERT INTO headline (luogo_lavorativo, occupazione) VALUES ('IWSI Group', 'CEO');

INSERT INTO headline (luogo_lavorativo, occupazione) VALUES ('Google', 'Software Engineer');

INSERT INTO headline (luogo_lavorativo, occupazione) VALUES ('Facebook', 'Digital Marketing Specialist');

INSERT INTO headline (luogo_lavorativo, occupazione) VALUES ('Microsoft', 'Data Analyst');

INSERT INTO headline (luogo_lavorativo, occupazione) VALUES ('Amazon', 'Product Manager');

INSERT INTO headline (luogo_lavorativo, occupazione) VALUES ('Apple', 'UX/UI Designer');

--INSERIMENTO PAESE/REGIONE/CITTA'/U-S/RESIDENZA DAL DATASET

--VUOTO PER L'UTENTE CONSIDERATO

INSERT INTO paese (nome) VALUES ('Italia');

INSERT INTO paese (nome) VALUES ('Stati Uniti');

INSERT INTO paese (nome) VALUES ('Regno Unito');

INSERT INTO paese (nome) VALUES ('Germania');

INSERT INTO paese (nome) VALUES ('Francia');

INSERT INTO regione (nome, paese) VALUES ('Lombardia', 'Italia');

INSERT INTO regione (nome, paese) VALUES ('California', 'Stati Uniti');

INSERT INTO regione (nome, paese) VALUES ('Londra', 'Regno Unito');

INSERT INTO regione (nome, paese) VALUES ('Baviera', 'Germania');

INSERT INTO regione (nome, paese) VALUES ('Île-de-France', 'Francia');

INSERT INTO citta (nome, regione) VALUES ('Milano', 'Lombardia');

INSERT INTO citta (nome, regione) VALUES ('Los Angeles', 'California');

INSERT INTO citta (nome, regione) VALUES ('Londra', 'Londra');

INSERT INTO citta (nome, regione) VALUES ('Monaco di Baviera', 'Baviera');

INSERT INTO citta (nome, regione) VALUES ('Parigi', 'Île-de-France');

INSERT INTO u_s (utente, paese) VALUES (1, 'Italia');

INSERT INTO u_s (utente, paese) VALUES (2, 'Stati Uniti');

INSERT INTO u_s (utente, paese) VALUES (3, 'Regno Unito');

INSERT INTO u_s (utente, paese) VALUES (4, 'Germania');

INSERT INTO u_s (utente, paese) VALUES (5, 'Francia');

INSERT INTO residenza (utente, citta) VALUES (1, 'Milano');

INSERT INTO residenza (utente, citta) VALUES (2, 'Los Angeles');

```
INSERT INTO residenza (utente, citta) VALUES (3, 'Londra');  
INSERT INTO residenza (utente, citta) VALUES (4, 'Monaco di Baviera');  
INSERT INTO residenza (utente, citta) VALUES (5, 'Parigi');
```

7. Query SQL

Workpackage	Task	Responsabile
WP3	SQL: Query	Cerino Marco

7.1. Query con operatore di aggregazione e join: Numero di visualizzazioni per ogni tipo di media

Tale query permette di trovare il numero totale di visualizzazioni per ogni tipo di media (immagini, video, sondaggi, articoli o documenti) pubblicato dagli influencer.

Si parte da un inner join tra la tabella media e la tabella post sull'uguaglianza degli attributi url (di media) e media (di post). Inoltre, vi è un ulteriore inner join tra la tabella precedentemente prodotta e la tabella panoramica post sull'uguaglianza degli attributi id (di post) e post (di panoramica post).

In seguito, l'operatore group by permette di effettuare il raggruppamento sui diversi tipi di media così da ottenere, mediante la select, per ogni tipo diverso di media il numero di visualizzazioni totali.

```
SELECT m.type_media, SUM(p_p.visualizzazioni) AS total_views
FROM media m
JOIN post p ON m.url = p.media
JOIN panoramica_post p_p ON p.id = p_p.post
GROUP BY m.type_media;
```

7.2. Query nidificata complessa: Visualizzazioni totali per influencer con almeno un post contenente un'immagine

Inserire una breve descrizione della query.

Trova gli influencer che hanno pubblicato almeno un post contenente un'immagine e il numero totale di visualizzazioni per tutti i loro post. Mostra il nome, il cognome e il numero totale di visualizzazioni.

Per ogni riga della tabella utente, si verifica se la tabella nidificata ritorna almeno una riga mediante l'operatore exists, permettendo di selezionare diverse righe della tabella utente. In particolar modo la query nidificata permette di capire se quell'utente ha pubblicato almeno un post contenente il media 'immagine' tramite l'inner join tra le tabelle post e media (sull'uguaglianza degli attributi media e url, rispettivamente di post e media) e la condizione espressa nel where. Inoltre, per ottenere il numero di visualizzazioni, è stata effettuata una query nidificata nella select che restituisce la somma del numero di visualizzazioni di tutti i post di quell'utente (tale query è effettuata un numero pari di volte agli utenti selezionati mediante l'exists nella clausola where).

```
SELECT u.nome, u.cognome, (
  SELECT SUM(p_p.visualizzazioni)
  FROM post p
  JOIN panoramica_post p_p ON p.id = p_p.post
  WHERE p.utente = u.id
) AS total_views
FROM utente_linkedin u
WHERE EXISTS (
  SELECT *
  FROM post p
  JOIN media m on (p.media = m.url)
```

```
WHERE p.utente = u.id and m.type_media = 'immagine'
)
ORDER BY u.cognome,u.nome;
```

7.3. Query insiemistica: Influencer con hashtag, link specifici e più di 10k follower

Trovare gli influencer che hanno pubblicato almeno un post con un determinato hashtag e almeno un post con un determinato link (si tengono in considerazione anche i post contenenti sia quell'hashtag sia quel link) con almeno 10k follower.

Per raggiungere questo obiettivo, la query utilizza una struttura nidificata. La parte interna della query è composta da due sottoquery. La prima sottoquery seleziona gli identificatori degli utenti che hanno pubblicato post contenenti l'hashtag "#technology". La seconda sottoquery seleziona gli identificatori degli utenti che hanno pubblicato post contenenti il link con il nome "Articolo interessante". Con l'operatore insiemistico di intersezione si verifica se l'identificatore dell'utente è presente in entrambi i risultati delle sottoquery. Inoltre, la query principale esterna verifica che l'utente abbia un numero di follower maggiore di 10.000. Se un utente soddisfa tutti questi criteri, il suo nome e cognome saranno restituiti come risultato della query.

```
SELECT ul.nome, ul.cognome
FROM utente_linkedin ul
WHERE ul.id IN (
  SELECT p.utente
  FROM post p
  INNER JOIN caratterizzato c ON p.id = c.post
  WHERE c.hashtag = '#technology'
  INTERSECT

  SELECT p.utente
  FROM post p
  INNER JOIN tag t ON p.id = t.post
  INNER JOIN link_post lp ON t.link_nel_post = lp.link_p
  WHERE lp.nome like 'Articolo interessante'
)
AND ul.num_follower > 10000;
```

7.4. Eventuali Altre query

7.4.1. Influencer con tutti i post con reazioni > 500

Inserire una breve descrizione della query.

*Trovami gli influencer che hanno **tutti** i post con un numero di reazioni > 500.*

Tale query è una nidificata complessa in quanto, per ogni utente della tabella utente_linkedin, si verifica, mediante la query nidificata, se esiste almeno un suo post con un numero di reazioni <= 500. In questo caso, mediante l'operatore not exists presente nella clausola where, non viene preso in considerazione mentre, per il caso opposto, viene restituito il nome ed il cognome di ogni utente che ha tutti i post con un numero di reazioni > 500.

La query utilizza una struttura nidificata complessa per eseguire questa verifica per ciascun utente nella tabella `utente_linkedin`. Nella query esterna, si selezionano il nome e il cognome degli utenti dalla tabella `utente_linkedin`. La parte nidificata della query è una sottoquery che verifica se esiste almeno un post dell'utente con un numero di reazioni inferiore o uguale a 500. Se tale condizione è verificata, l'operatore "NOT EXISTS" nella clausola WHERE garantisce che quell'utente non verrà incluso nel risultato finale.

Questa query complessa permette di individuare gli influencer che hanno mantenuto una buona popolarità con un alto numero di reazioni per tutti i loro post su LinkedIn.

```
SELECT u.nome, u.cognome
FROM utente_linkedin u
WHERE NOT EXISTS (
  SELECT *
  FROM post p join panoramica_post p_p on (p.id = p_p.post)
  WHERE p.utente = u.id AND p_p.reazioni <= 500
);
```

7.4.2. Post più influente per somma di reazioni e commenti

Identifica il post più influente basandosi sulla somma di reazioni e commenti presenti nella rispettiva panoramica (operazione 6 della tavola delle operazioni). Nel caso vi siano post con somma uguali devono essere identificati.

La query seleziona l'identificatore del post, il contenuto del post, il numero di reazioni, il numero di commenti e la somma di reazioni e commenti come "somma". La tabella "post" viene unita alla tabella "panoramica_post" utilizzando l'identificatore del post come chiave di collegamento. Nella clausola WHERE, la query confronta la somma delle reazioni e dei commenti di ciascun post con il valore massimo ottenuto dalla sottoquery nidificata. La sottoquery restituisce il valore massimo della somma di reazioni e commenti dalla tabella "panoramica_post".

```
SELECT p.id, p.contenuto_post, pp.reazioni, pp.commenti, (pp.reazioni + pp.commenti) AS
somma
FROM post p
JOIN panoramica_post pp ON p.id = pp.post
WHERE (pp.reazioni + pp.commenti) = (
  SELECT MAX(reazioni + commenti)
  FROM panoramica_post
)
```

7.4.3. Influencer con tutti i post con reazioni > 500

Query alternativa a: Trovami gli influencer che hanno **tutti** i post con un numero di reazioni > 500.

```
SELECT u.id, u.nome, u.cognome
FROM utente_linkedin u
EXCEPT (
```

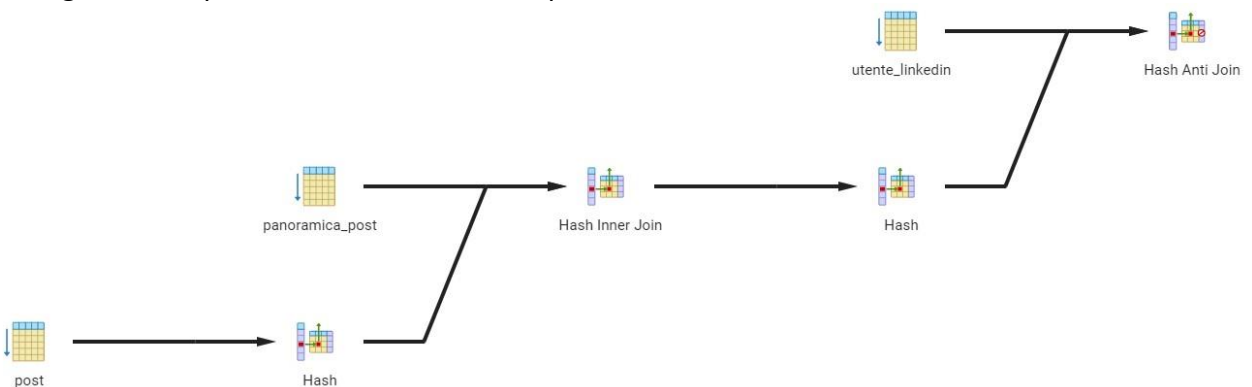
```

SELECT u.id,u.nome, u.cognome
FROM utente_linkedin u
JOIN post p ON u.id = p.utente
JOIN panoramica_post p_p ON p.id = p_p.post
WHERE p_p.reazioni <= 500
);

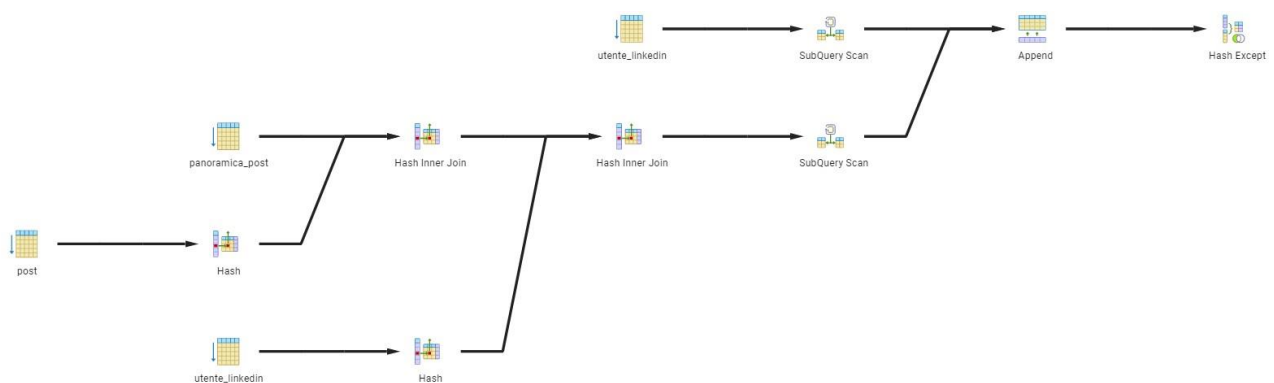
```

ANALISI CONFRONTO tra 7.4.1. e 7.4.3.

La differenza è che nella prima query alla tabella utente linkedin si accede solo una volta (sfruttando il meccanismo delle query annidate complesse) mentre, nella seconda query, accediamo alla tabella utente linkedin due volte. A supporto di tale osservazione vi è la seguente analisi grafica dei piani di accessi delle due operazioni:



(Query 7.4.1.)



(Query 7.4.3.)

Si precisa che, a livello fisico, vi sono dei meccanismi di ottimizzazione delle query che riducono le differenze tra le operazioni, rendendole più o meno simili tra loro.

Si mostrano i diversi tempi di esecuzioni delle query (simili tra loro):

21	Planning Time: 0.167 ms
22	Execution Time: 0.062 ms

(Query 7.4.1.)

31	Planning Time: 0.272 ms
32	Execution Time: 0.137 ms

(Query 7.4.3.)

8. Viste

Workpackage	Task	Responsabile
WP4	Viste	De Angelis Christian

8.1. Vista *Numero post ed hashtag di uno specifico utente*

La vista "utenti_post_hashtag" fornisce un'analisi aggregata degli utenti di LinkedIn, concentrandosi sul numero di post pubblicati da ciascun utente e sul numero totale di hashtag associati a tali post. La vista combina informazioni provenienti da diverse tabelle del database. In particolare, utilizza la tabella "utente_linkedin" per ottenere i dettagli degli utenti, la tabella "post" per ottenere informazioni sui post pubblicati e la tabella "caratterizzato" per contare il numero di hashtag associati a ciascun post. Attraverso JOIN e GROUP BY, la vista aggrega i dati in base all'ID dell'utente, al nome e al cognome, calcolando il numero totale di post e il numero totale di hashtag per ciascun utente.

In sostanza, la vista "utenti_post_hashtag" semplifica l'accesso ai dati aggregati sugli utenti di LinkedIn, consentendo di ottenere rapidamente informazioni sul numero di post e hashtag di un utente specifico. Si precisa che tale vista non è modificabile in quanto non è definita su un'unica tabella.

```
CREATE VIEW utenti_post_hashtag AS
SELECT ul.id AS utente_id, ul.nome, ul.cognome, COUNT(p.id) AS num_post, SUM(p.num_hashtag)
AS num_hashtag_totali
FROM utente_linkedin ul
INNER JOIN post p ON ul.id = p.utente
GROUP BY ul.id, ul.nome, ul.cognome;
```

8.1.1. Query con Vista: Selezione di uno specifico utente secondo determinate condizioni

```
SELECT utente_id, nome, cognome, num_hashtag_totali
FROM utenti_post_hashtag
WHERE num_post > 5 AND num_hashtag_totali > (
    SELECT AVG(num_hashtag_totali)
    FROM utenti_post_hashtag)
```


9. Trigger

9.1. Trigger inizializzazione: Gestione del vincolo “un hashtag deve essere contenuto almeno in una tabella Caratterizzato”

Workpackage	Task	Responsabile
WP1	Trigger inizializzazione/popoloamento database	Attianese Carmine

Vengono definiti due trigger utilizzando la funzione di trigger "AtLeastONE". Il primo trigger viene attivato dopo l'inserimento di una riga nella tabella "hashtag". Il secondo trigger viene attivato dopo l'eliminazione di una riga dalla tabella "caratterizzato". Entrambi i trigger eseguiranno la funzione di trigger "AtLeastONE".

La funzione di trigger verifica se esistono valori nella tabella "hashtag" che non sono presenti nella tabella "caratterizzato". Se tale condizione è verificata, viene generata un'eccezione con il messaggio "ERRORE 1" che non permette l'inserimento o l'eliminazione.

```
CREATE OR REPLACE FUNCTION AtLeastONE() returns trigger as $$
BEGIN
if (exists (select nome from hashtag
where nome not in (select hashtag from caratterizzato))) then
RAISE EXCEPTION 'ERRORE 1';
end if;
return NEW;
END $$ LANGUAGE plpgsql;
```

```
create trigger AtLeastONE
after insert on hashtag
for each row execute procedure AtLeastONE();
```

```
create trigger AtLeastONE
after delete on caratterizzato
for each row execute procedure AtLeastONE();
```

9.2. Trigger per vincoli aziendali

Workpackage	Task	Responsabile
WP4	Trigger per vincoli aziendali	De Angelis Christian

9.2.1. Trigger1: Controllo numero post di un influencer (RV12 e RV15)

La funzione del trigger è quella di associare un insieme di azioni ad un evento specifico che si verifica su una tabella del database. In questo caso, ci sono tre trigger definiti nel codice fornito.

Il trigger si attiva dopo ogni inserimento o eliminazione di una riga nella tabella "post"

Sia dopo una cancellazione "OLD IS NOT NULL" che dopo un inserimento "NEW IS NOT NULL" la funzione associata al trigger conta il numero di post associato all'id dell'utente. Se durante l'inserimento il numero di post diventa maggiore di 10, seguendo la regola aziendale, invia un messaggio d'avviso in quanto il numero medio di post supera quello prefissato, tuttavia non blocca l'inserimento. Se durante una cancellazione di una riga il numero di post diventa uguale ad 1 invia un messaggio d'avviso mentre se diventa uguale a 0 lancia un'eccezione che blocca la cancellazione in quanto un influencer deve avere almeno un post.

```
--Trigger
CREATE OR REPLACE FUNCTION lowest_views_post()
RETURNS TRIGGER AS $$
DECLARE
post_count integer;
lowest_views_post1 integer :=-1;
BEGIN

    IF (NEW IS NOT NULL) THEN
        SELECT COUNT(*) INTO post_count
        FROM post
        WHERE utente = NEW.utente;

        IF post_count > 10 THEN

            -- Notifica il post con il numero di visualizzazioni più basso
            RAISE NOTICE'ATTENZIONE:L' utente id:% ha raggiunto 10 post',NEW.utente;
        END IF;
    END IF;
    IF (OLD IS NOT NULL) THEN
        SELECT COUNT(*) INTO post_count
        FROM post
        WHERE utente = OLD.utente;
        IF (post_count = 1) THEN
            RAISE NOTICE'ATTENZIONE: numero post uguale ad uno per tale influencer' ;
        ELSIF (post_count = 0) THEN
            RAISE EXCEPTION'influencer non può avere 0 post';
        END IF;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER check_influencer_post_count
AFTER INSERT OR DELETE ON post
```

```
FOR EACH ROW  
EXECUTE FUNCTION lowest_views_post();
```

9.2.2. Trigger2: Voti di un sondaggio(RV8)

Il trigger viene attivato dopo che si verifica un'operazione di inserimento o aggiornamento sulla tabella "panoramica_post" e si attiva per ogni riga su cui è eseguita l'operazione.

La funzione associata controlla se il campo "voti" del record appena inserito o aggiornato non è nullo. Se il campo "voti" non è nullo, la funzione verifica se il tipo di media associato al post (recuperato dalla tabella "media" usando l'URL ottenuto dalla tabella "post") non è "sondaggio". Se il tipo di media non è "sondaggio", viene generata un'eccezione con il messaggio "Se il post non ha come media un sondaggio, i voti devono essere NULL" che blocca l'inserimento. Se invece le condizioni sono verificate viene permesso l'inserimento.

```
CREATE OR REPLACE FUNCTION update_panoramica_voti() RETURNS TRIGGER AS $$  
BEGIN  
  IF NEW.voti IS NOT NULL THEN  
    -- Controlla se il media associato al post non è un "sondaggio"  
    IF (SELECT type_media FROM media WHERE url = (SELECT media FROM post WHERE id =  
NEW.post)) <> 'sondaggio' THEN  
      RAISE EXCEPTION 'Se il post non ha come media un sondaggio, i voti devono essere NULL.';  
    END IF;  
  END IF;  
  
  RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_update_panoramica_voti  
AFTER INSERT OR UPDATE ON panoramica_post  
FOR EACH ROW  
EXECUTE FUNCTION update_panoramica_voti();
```

9.2.3. Trigger3: Aggiornamento numero hashtag

Il trigger "caratterizzato" è associato all'evento di inserimento, eliminazione o aggiornamento della colonna "hashtag" nella tabella "caratterizzato" e si attiva per ogni riga su cui è eseguita l'operazione. La funzione associata al trigger è quella di aggiornare il valore della colonna "num_hashtag" nella tabella "post" a 1, sia per la riga appena inserita o aggiornata ("if new is not null"), sia per la riga eliminata ("if old is not null"), in modo da richiamare il trigger "protegginumerohashtag".

Il secondo trigger è chiamato "protegginumerohashtag" ed è associato all'evento di inserimento o aggiornamento della colonna "num_hashtag" nella tabella "post". La funzione associata al trigger è quella di eseguire una query che conta il numero di righe nella tabella "caratterizzato" dove il campo

"post" corrisponde all'id della riga appena inserita o aggiornata ("new"). Il risultato di questa query viene inserito nel campo "num_hashtag" della riga interessata. Tale trigger si attiva sia nella fase di inizializzazione sia successivamente per rimanere coerente il dato ridondante.

```
CREATE OR REPLACE FUNCTION caratterizzato() RETURNS TRIGGER AS $$  
begin  
if (new is not null) then--insert o update  
    update post set num_hashtag = 1 where id = new.post;  
end if;  
if (old is not null) then  
    update post set num_hashtag = 1 where id = old.post;  
end if;  
return null;  
End;$$ language plpgsql;
```

```
create trigger caratterizzato  
after insert or delete or update of hashtag on caratterizzato  
for each row  
execute procedure caratterizzato();
```

--trigger dato ridondante parte 2

```
CREATE OR REPLACE FUNCTION protegginumerohashtag() RETURNS TRIGGER AS $$  
begin  
    select count(*) into new.num_hashtag  
    from caratterizzato where post = new.id;  
    return new;  
End;$$ LANGUAGE plpgsql;
```

```
create trigger protegginumerohashtag  
before insert or update of num_hashtag on post  
for each row  
execute procedure protegginumerohashtag();
```