

Esercitazione di laboratorio Corso di Basi di Dati

A.A. 2016/2017

Docente: Marco Mesiti

Titolo esercitazione:

Cinguettio

Gruppo composto da :

Francesco Urso - francesco.urso1@studenti.unimi.it [referente]

Ismail donà - ismail.dona@studenti.unimi.it

Data consegna: 17/07/2017

Il progetto è già stato consegnato in precedenza? no

1. Progettazione concettuale

SPECIFICHE RISTRUTTURATE

Utente

Un utente per poter usufruire dei servizi di cinguettio si deve registrare al sistema inserendo obbligatoriamente:

- Nickname → {**nickname**}* (1,1)
- Password → {**password**}* (1,1)
- Un'email che rappresenta il "nome utente" → {**email_utente**}* (1,1)

*{**email_utente**} identifica univocamente un utente e ha una lunghezza massima di 70 caratteri ed ha il seguente formato '<utente>@<provider>.<dominio>' e può contenere lettere numeri e i seguenti caratteri [- _ .] è case insensitive.

*{**password**} va inserita in fase di registrazione 2 volte per controllo e ha una lunghezza tra 1 e 50 caratteri e contiene solo caratteri alfanumerici e i seguenti caratteri [- _ * ! .]

*{**nickname**} è univoco e ha una lunghezza tra 1 e 20 caratteri contiene solo caratteri alfanumerici e i seguenti caratteri [' " - _ * ! .] e spazi

Una volta verificata la correttezza di {**email_utente**} e {**password**} l'utente è "loggato" al sistema.

L'utente può quindi completare il suo profilo aggiungendo informazioni personali quali:

- Nome → {**nome**}* (0,1)
- Cognome → {**cognome**}* (0,1)
- La data di nascita → {**data_nascita**}* (0,1)
- Il luogo di nascita → {**nazione_nascita**}* (0,1)
- La regione di nascita → {**regione_nascita**}* (0,1)
- Il sesso → {**sesso**}* (0,1)
- Gli hobbies → {**tipo_hobby**}* (0, 10)
- La città in cui attualmente abita → {**citta_residenza**}* (0,1)

*{**nome**} ha una lunghezza massima di 20 caratteri e può contenere solo caratteri alfabetici il carattere di spazio e il carattere di apostrofo.

*{**cognome**} ha una lunghezza massima di 20 caratteri e contiene solo caratteri alfabetici il carattere di spazio e il carattere di apostrofo.

*{**data_nascita**} deve essere minore o uguale alla data corrente nel momento in cui viene inserita.

*{**nazione_nascita**} ha una lunghezza massima di 50 caratteri e contiene solo caratteri alfabetici il carattere di spazio e il carattere di apostrofo.

*{**regione_nascita**} ha una lunghezza massima di 50 caratteri e contiene solo caratteri alfabetici il carattere di spazio e il carattere di apostrofo.

*{**sesso**} ha alternativamente uno tra i seguenti valori:

- 'm' → per un utente di sesso maschile
- 'f' → per un utente di sesso femminile

Per quanto riguarda gli hobby l'utente può scegliere solo tra un insieme di 9 hobby.

*{tipo_hobby} → può assumere i seguenti valori:

- 'sport'
- 'musica'
- 'arte'
- 'tecnologia'
- 'film e serie tv'
- 'fotografia'
- 'moda e design'
- 'videogames'
- 'viaggiare'

L'utente può inserire infine un hobby aggiuntivo tramite una form testuale che non superi i 20 caratteri la quale contiene solo caratteri alfabetici e il carattere di spazio e il carattere di apostrofo. Se l'hobby inserito dall'utente è uguale ad uno degli hobby inseriti in precedenza questo viene semplicemente ignorato. Un utente può quindi scegliere da 0 ad un massimo di 10 hobby.

*{citta_residenza} ha una lunghezza massima di 50 caratteri e contiene solo caratteri alfabetici il carattere di spazio e il carattere di apostrofo

Ogni utente ha un'immagine profilo con le seguenti caratteristiche:

- Il nome dell'immagine → {img_profilo}* (1,1)
- Il percorso sul file system → {path_img_profilo}* (1, 1)

*{img_profilo} è costituito dalla stringa

- 'profilo_{nickname}.<formato>'

dove {nickname} è il nickname dell'utente.

Il valore di default è 'default.png'

*{path_img_profilo} è costituito dalla stringa

- 'img/utenti/profilo/profilo_{nickname}.<formato>'

dove {nickname} è il nickname dell'utente.

Il valore di default è 'img/utenti/profilo/default.png'

L'utente può sostituire l'immagine profilo, nella propria pagina personale, con una nuova immagine.

Cinguettio

Ogni utente può pubblicare dei "cinguettii" ciascuno caratterizzato da:

- Un identificativo → {id_cinguettio}* (1,1)
- Dalla data e ora di creazione → {data_ora_cinguettio}* (1,1)

- Un tipo di cinguettio → **{tipo_cinguettio}* (1,1)**

***{id_cinguettio}** è un valore intero che deve seguire una numerazione progressiva per ogni utente e identifica univocamente il cinguettio di un utente.

***{data_ora_cinguettio}** corrisponde alla data e all'ora corrente nel momento in cui viene creato il cinguettio.

***{tipo_cinguettio}** ha alternativamente uno tra i seguenti valori:

- 'T' → per un cinguettio di testo
- 'F' → per un cinguettio di foto
- 'L' → per un cinguettio di luogo
-

I cinguettii possono essere alternativamente di tre tipi

- dei messaggi di testo
- delle foto
- dei luoghi

I cinguettii di tipo testo

Un cinguettio può essere un messaggio di testo. Ogni messaggio di testo è caratterizzato da:

- Un messaggio testuale → **{messaggio}* (1,1)**
- Se il messaggio è inappropriato → **{inappropriato}* (1,1)**
- Se il testo è oscurato → **{oscurato}* (1,1)**

***{messaggio}** ha una lunghezza tra 1 e 50 caratteri

***{inappropriato}** è un valore booleano che assume valore 'TRUE' se il messaggio è stato segnalato come "inappropriato" altrimenti assume valore 'FALSE'. Ogni utente può infatti segnalare un cinguettio di tipo testo di un altro utente che segue come "inappropriato".

Un utente non può segnalare un proprio cinguettio di tipo testo come "inappropriato" né un cinguettio che ha già segnalato come "inappropriato". Quando un messaggio di testo viene mostrato sulla bacheca di un utente, deve essere possibile visualizzare l'indicazione che un messaggio è "inappropriato" e l'elenco degli utenti che lo hanno definito come "inappropriato" ovvero il loro **{nickname}**.

***{oscurato}** è un valore booleano che assume valore 'TRUE' nel caso in cui un messaggio è oscurato altrimenti assume valore 'FALSE'. Quando un utente riceve più di tre segnalazioni che uno stesso messaggio è **{inappropriato}** il messaggio deve essere **{oscurato}**. Il cinguettio deve rimanere nella bacheca dell'utente, ma non è più possibile:

- leggerne il contenuto
- segnalare il messaggio come **{inappropriato}**
- vedere l'elenco degli utenti che l'hanno definito **{inappropriato}**.

I cinguettii di tipo luogo

Un cinguettio può essere un luogo. Per i luoghi si vuole memorizzare la posizione geografica pertanto ogni luogo è caratterizzato da:

- La latitudine → **{latitudine}* (1,1)**

- La longitudine → **{longitudine}*** (1,1)

{latitudine} è un valore a virgola mobile compreso tra -180.000000 e 180.000000. L'utente quando visualizza la **{latitudine}** la visualizza nel formato

- °gradi minuti" secondi" direzione N/S

{longitudine} è un valore a virgola mobile compreso tra -180.000000 e 180.000000. L'utente quando visualizza la **{longitudine}** la visualizza nel formato

- °gradi minuti" secondi" direzione E/W

I cinguettii di tipo foto

Un cinguettio può essere una foto. Per quanto riguarda le foto, si introduce una semplificazione. Non si richiede, infatti, di memorizzare le foto internamente alla base di dati, ma di memorizzarle all'interno del file system. Quando viene pubblicato un cinguettio di tipo foto bisogna creare una cartella nel file system se non c'è già in cui memorizzare le foto caricate dagli utenti. Dunque, per ogni foto è necessario memorizzare:

- Il nome del file → **{nome_foto}*** (1,1)
- Il percorso sul file system → **{path_foto}*** (1,1)
- Una descrizione opzionale della foto → **{descrizione_foto}*** (0,1)
- La data in cui è stata caricata la foto → **{data_caricamento}*** (1,1)

{nome_foto} è costituito dalla stringa

- '<nome_univoco>.<formato>'

{path_foto} è costituito dalla stringa

- 'img/utenti/upload/{nickname}/<nome_univoco>.<formato>'

dove **{nickname}** è il nickname dell'utente

{descrizione_foto} ha una lunghezza massima di 20 caratteri

{data_caricamento} corrisponde alla data corrente nel momento in cui viene caricata la foto.

Utente esperto

L'utente che ha almeno altri 3 utenti che lo seguono è definito "esperto". Se un utente esperto dovesse perdere dei seguaci ed avere un numero di seguaci inferiore a 3 smette di essere un utente esperto. Si vuole sapere a partire da quale data un utente viene considerato esperto.

- La data in cui un utente diventa esperto → **{data_esperto}*** (1,1)

{data_esperto} corrisponde alla data corrente nel momento in cui un utente diventa un utente esperto. Se un utente smette di essere esperto il suo valore diventa NULL. Un utente esperto può esprimere un giudizio di apprezzamento per i cinguettii di tipo foto pubblicati da tutti gli utenti che segue e per i propri cinguettii.

Giudizio foto

Ogni giudizio di apprezzamento deve essere corredato da:

- Un commento → **{commento}*** (1,1)
- Dalla data e ora → **{data_ora_commento}*** (1,1)

***{commento}** ha una lunghezza tra 1 e 50 caratteri

***{data_ora_commento}** corrisponde alla data e all'ora corrente nel momento in cui viene commentata una foto.

Si osservi che un utente esperto può esprimere più volte apprezzamento per una stessa foto ma mai più di 3 volte, in istanti diversi. Quando una foto viene mostrata in una bacheca, deve essere disponibile un link che mostri tutti gli apprezzamenti fatti a quella foto. L'indicazione dell'apprezzamento di una foto può essere fatto dalla bacheca, quando la foto viene mostrata.

Luogo preferito

Ogni utente può definire i propri luoghi preferiti. Ogni luogo preferito corrisponde ad un cinguettio di tipo "luogo" inserito dall'utente stesso o da un altro utente seguito. Uno stesso luogo può essere preferito da più utenti.

Un utente può visualizzare i propri luoghi preferiti nella propria pagina personale mentre può visualizzare i luoghi preferiti dagli altri utenti nella loro pagina personale.

Un utente non può definire più di 1 volta uno stesso cinguettio di tipo luogo come luogo preferito

Utenti seguaci e utenti seguiti

Ogni utente può "seguire" altri utenti del sistema ma non sé stesso.

Quando un utente A "segue" un utente B, l'utente A è in grado di vedere tutti i cinguettii pubblicati dall'utente B e accedere alle sue informazioni personali.

Un utente A per poter seguire un utente B deve richiedere a B di diventare suo seguace e B può decidere di accettare o rifiutare.

Nel caso in cui B accetta la richiesta, A diventa un suo seguace.

Il fatto che B rifiuta di essere seguito da A non viene memorizzato.

In questo modo l'utente A può nuovamente richiedere di seguire l'utente B anche se l'utente B ha rifiutato di essere seguito già una volta da A.

Fintanto che una richiesta è appesa, ovvero non è stata ancora accettata o rifiutata, l'utente A non può inviare una nuova richiesta all'utente B.

Un utente non può inviare una richiesta di seguire un utente che già segue.

Si vuole tenere traccia quindi per gli utenti che seguono altri utenti:

- Della data dell'invio di richiesta → **{data_ora_richiesta}*** (1,1)
- Data di risposta alla richiesta → **{data_ora_risposta}*** (0,1)
- Se la richiesta è stata accettata → **{richiesta_accettata}*** (1,1)

***{data_ora_richiesta}** corrisponde alla data e all'ora corrente nel momento in cui viene inviata una richiesta.

***{data_ora_risposta}** corrisponde alla data e all'ora corrente nel momento in cui viene accettata una richiesta. Se una richiesta è stata inviata ma non è stata ancora accettata assume valore 'NULL'. Inoltre deve essere sempre minore di **{data_ora_risposta}**. Il suo valore di default è 'NULL'

***{richiesta_accettata}** è un valore booleano che assume valore 'TRUE' se la richiesta è stata accettata

altrimenti assume valore 'FALSE' se la richiesta non è ancora stata accettata. Il suo valore di default è 'FALSE'.

In una apposita pagina del sito, denominata "chi mi segue", un utente vede l'elenco degli utenti seguaci con l'informazione della data a partire dalla quale è stata concessa l'autorizzazione. L'utente in questa pagina può:

- Seguire un utente seguace
- Smettere di seguire un utente seguace
- Avviare una ricerca per trovare nuovi utenti

Nella pagina "chi seguo" un utente vede l'elenco degli utenti seguiti con la possibilità di:

- Smettere di seguire gli utenti.
- Avviare una ricerca per trovare nuovi utenti.

La ricerca utenti

L'utente può ricercare altri utenti tramite due tipi di ricerca

- semplice
- avanzata

la ricerca "semplice" può essere effettuata con 1 dei seguenti parametri:

- {email_utente}
- {nickname}.

La "ricerca avanzata" si basa su diversi parametri quali:

- {nome}
- {cognome}
- {data_nascita}
- {nazione_nascita}
- {regione_nascita}
- {sesso}
- {tipo_hobby}
- {citta_residenza}

Per la ricerca "avanzata" i parametri sono opzionali quindi nel caso non vengano specificati occorre restituire tutti gli utenti.

La bacheca

Una delle pagine del sito, denominata "bacheca", deve permettere ad un utente A di vedere l'elenco dei propri cinguettii e di quelli di tutte le persone che l'utente segue.

Questi cinguettii sono ordinati in base alla data e ora di creazione con i cinguettii più recenti in alto.

Si richiede di decidere autonomamente quale sia la rappresentazione più adeguata per mostrare all'interno della bacheca diversi tipi di cinguettii

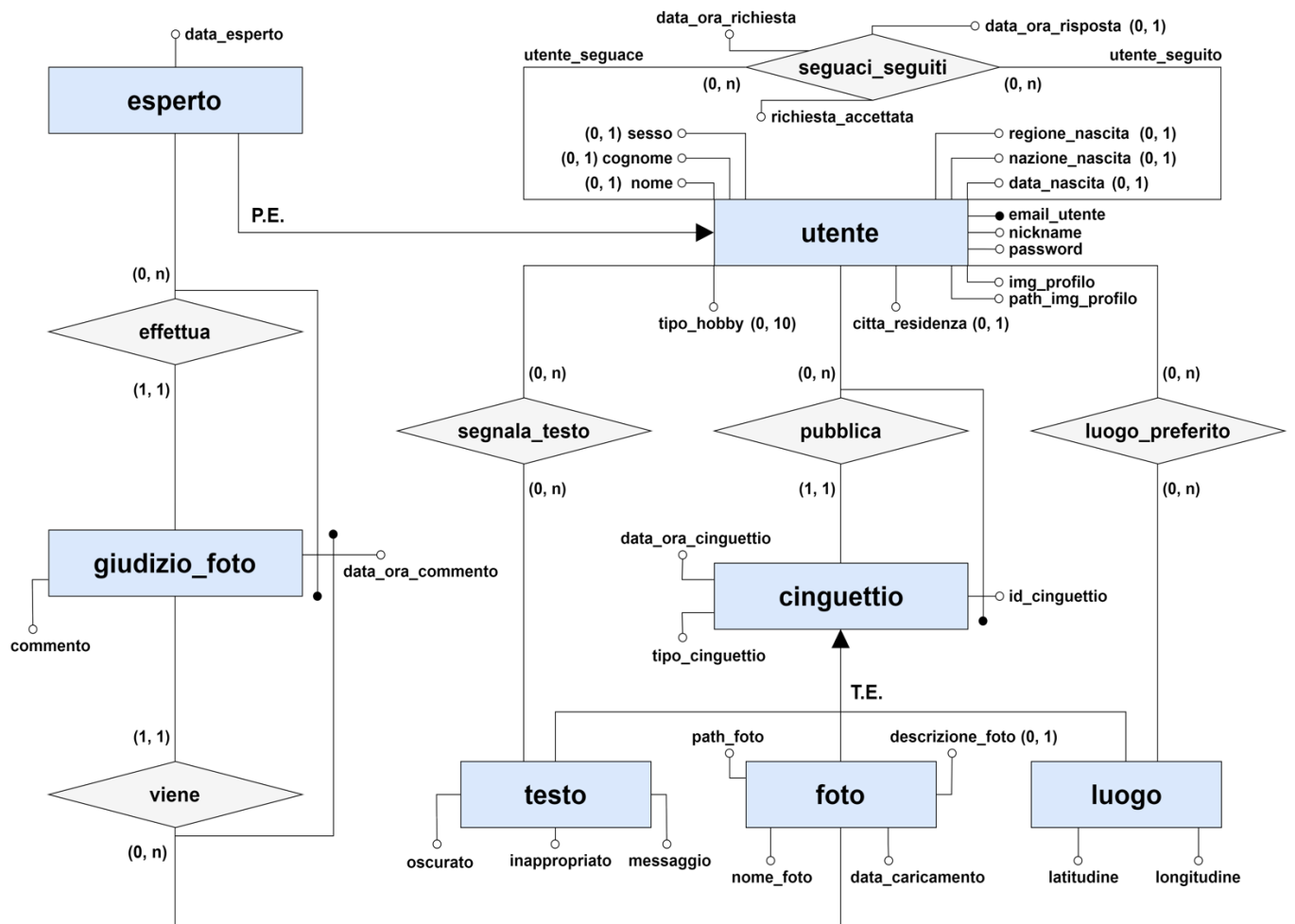
I dati personali degli altri utenti

I dati personali di un utente sono visualizzabili dagli altri utenti seguaci in una apposita pagina "dati utente".

Questa pagina mostra inoltre:

- quanti utenti l'utente sta seguendo
- quanti utenti lo seguono
- i luoghi preferiti

1.1 Schema ER



GERARCHIE DI GENERALIZZAZIONE SCHEMA ER

Entità padre	Entità figlia	Tipologia
utente	esperto	Parziale Esclusiva
cinguettio	testo luogo foto	Totale Esclusiva

1.2 Vincoli di dominio

Vincoli d'integrità

V1: L'attributo **email_utente** dell'entità **utente** ha una lunghezza massima di 70 caratteri ed ha il seguente formato '<utente>@<provider>.<dominio>' e può contenere lettere numeri e i seguenti caratteri [- _ .] è case insensitive.

V2: L'attributo **password** dell'entità **utente** ha una lunghezza 1 e 50 caratteri e contiene solo caratteri alfanumerici e i seguenti caratteri [_ - ! * .]

V3: L'attributo **nickname** dell'entità **utente** è univoco ha una lunghezza tra 1 e 20 caratteri contiene solo caratteri alfanumerici e seguenti caratteri [' " - _ * ! .] e spazi

V4: L'attributo **nome** dell'entità **utente** ha una lunghezza massima di 20 caratteri e può contenere solo caratteri alfabetici il carattere di spazio e il carattere di apostrofo.

V5: L'attributo **cognome** dell'entità **utente** ha una lunghezza massima di 20 caratteri e può contenere solo caratteri alfabetici il carattere di spazio e il carattere di apostrofo.

V6: L'attributo **data_nascita** dell'entità **utente** deve essere minore o uguale alla data corrente nel momento in cui viene inserita.

V7: L'attributo **nazione_nascita** dell'entità **utente** ha una lunghezza massima di 50 caratteri e contiene solo caratteri alfabetici il carattere di spazio e il carattere di apostrofo.

V8: L'attributo **regione_nascita** dell'entità **utente** ha una lunghezza massima di 50 caratteri e contiene solo caratteri alfabetici il carattere di spazio e il carattere di apostrofo.

V9: L'attributo **sex** dell'entità **utente** può assumere uno tra i seguenti valori {'f', 'm'} dove 'm' indica che si tratta di un utente di sesso maschile mentre 'f' indica un utente di sesso femminile.

V10: L'attributo **tipo_hobby** dell'entità **utente** può assumere i seguenti valori {'sport', 'musica', 'arte', 'tecnologia', 'film e serie tv', 'fotografia', 'moda e design', 'videogames', 'viaggiare'} l'utente può inserire un hobby aggiuntivo tramite una form testuale che non superi i 20 caratteri e che contiene solo caratteri alfabetici il carattere di spazio e il carattere di apostrofo.

V11: L'attributo **citta_residenza** dell'entità **utente** ha una lunghezza massima di 50 caratteri e contiene solo caratteri alfabetici il carattere di spazio e il carattere di apostrofo.

V12: L'attributo **id_cinguettio** dell'entità **cinguettio** è un valore intero che deve seguire una numerazione progressiva per ogni utente.

V13: L'attributo **data_ora_cinguettio** dell'entità **cinguettio** corrisponde alla data e all'ora corrente nel momento in cui viene creato il cinguettio.

V14: L'attributo **tipo_cinguettio** dell'entità **cinguettio** può assumere uno tra i seguenti valori {'T', 'F', 'L'} dove 'T' indica che si tratta di un cinguettio di tipo testo mentre 'F' che si tratta di un cinguettio di tipo foto 'L' che si tratta di un cinguettio di tipo luogo.

V15: L'attributo **messaggio** dell'entità **testo** ha una lunghezza massima di 50 caratteri e minima di 1 carattere.

V16: L'attributo **inappropriato** dell'entità **testo** è un valore booleano che assume valore 'TRUE' quando il cinguettio di tipo testo di un utente seguito viene segnalato come "inappropriato" da un altro utente altrimenti assume valore 'FALSE'.

V17: L'attributo **oscurato** dell'entità **testo** è un valore booleano che assume valore 'TRUE' nel caso in cui un cinguettio di tipo testo venga segnalato come inappropriato più di 3 volte altrimenti assume valore 'FALSE'.

V18: Ogni utente può segnalare un cinguettio di tipo testo di un altro utente che segue come "inappropriato". Un utente non può segnalare un proprio cinguettio di tipo testo come "inappropriato" né un cinguettio che ha già segnalato come "inappropriato".

V19: L'attributo **latitudine** dell'entità **luogo** è un valore a virgola mobile compreso tra -180.000000 e 180.000000

V20: L'attributo **latitudine** dell'entità **luogo** è un valore a virgola mobile compreso tra -180.000000 e 180.000000

V21: L'attributo **descrizione_foto** dell'entità **foto** ha una lunghezza massima di 20 caratteri.

V22: L'attributo **data_caricamento** dell'entità **foto** corrisponde alla data corrente nel momento in cui viene caricata la foto.

V23: L'utente che ha almeno altri 3 utenti che lo seguono è definito "esperto".

V24: Se un utente esperto ha un numero di seguaci inferiore a 3, smette di essere un utente esperto.

V25: L'attributo **data_esperto** dell'entità **esperto** corrisponde alla data corrente nel momento in cui un utente diventa un utente esperto.

V26: Un utente esperto può esprimere un giudizio di apprezzamento per i cinguettii di tipo foto pubblicati solo da utenti che segue o per i propri cinguettii di tipo foto. L'utente esperto non può esprimere un giudizio di apprezzamento più di 3 volte per uno stesso cinguettio di tipo foto.

V27: L'attributo **commento** dell'entità **giudizio_foto** ha una lunghezza massima di 50 caratteri e minima di 1 carattere.

V28: L'attributo **data_ora_commento** dell'entità **giudizio_foto** corrisponde alla data e all'ora corrente nel momento in cui viene commentata una foto.

V29: Un utente può definire come luogo preferito un proprio cinguettio di tipo luogo o un cinguettio di tipo luogo pubblicato da un altro utente che segue ma non può definire uno stesso cinguettio di tipo luogo come luogo preferito più di 1 volta.

V30: Un utente non può seguire sé stesso né può inviare una richiesta di seguire un utente che già segue

V31: L'attributo **data_ora_richiesta** dell'associazione **seguaci_seguiti** corrisponde alla data e all'ora corrente nel momento in cui viene inviata una richiesta di seguire un utente.

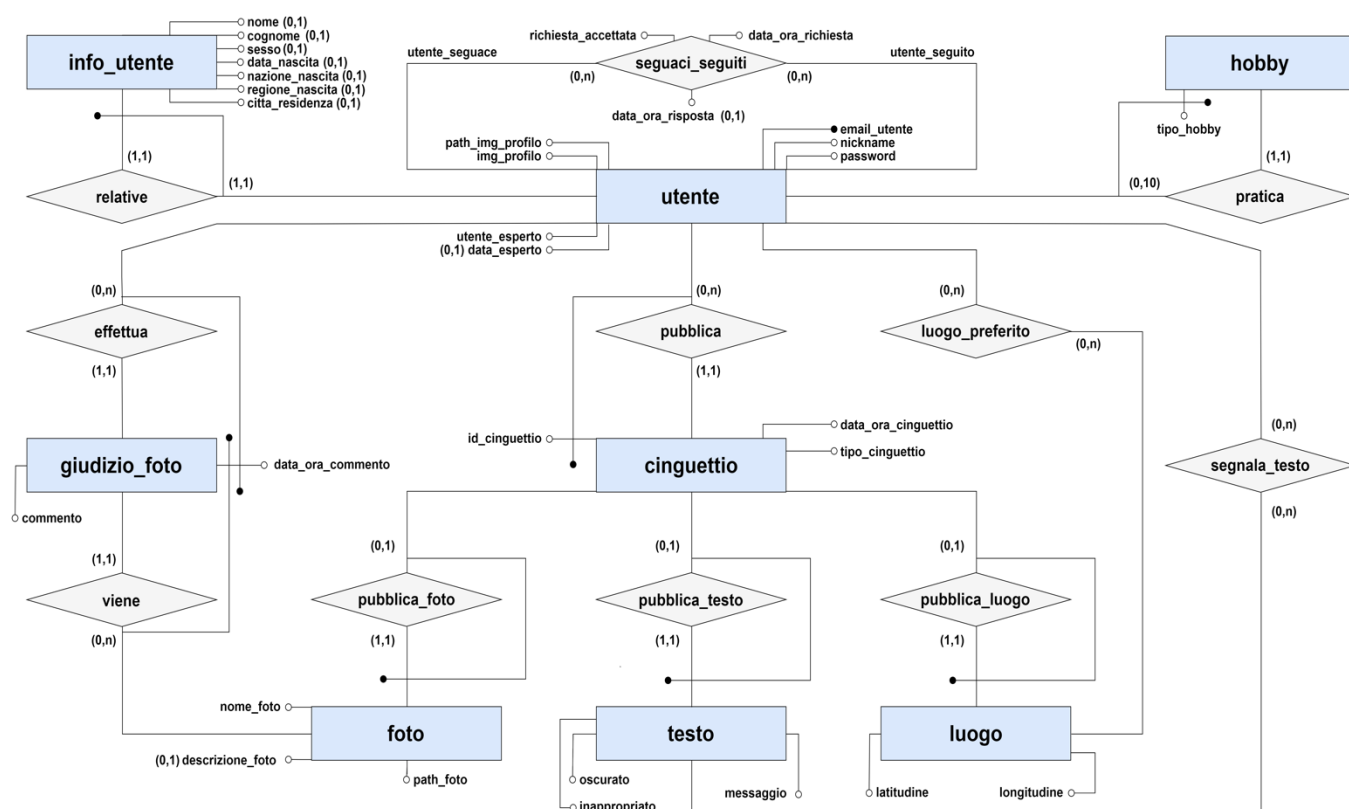
V32: L'attributo **data_ora_risposta** dell'associazione **seguaci_seguiti** corrisponde alla data e all'ora corrente nel momento in cui viene accettata una richiesta e deve essere maggiore dell'attributo **data_ora_richiesta** dell'associazione **seguaci_seguiti**. Se una richiesta è stata inviata ma non è stata ancora accettata assume valore 'NULL'.

V33: L'attributo **richiesta_accettata** dell'associazione **seguaci_seguiti** è un valore booleano che assume valore 'TRUE' se la richiesta è stata accettata altrimenti assume valore 'FALSE' se la richiesta non è ancora stata accettata.

V34: Un utente per poter vedere i cinguettii pubblicati da un altro utente accedere alle sue informazioni personali deve essere un suo seguace.

V35: Un utente non può chiedere ad un altro utente di seguirlo se ha già inviato in precedenza una richiesta a cui non è stata data ancora una risposta.

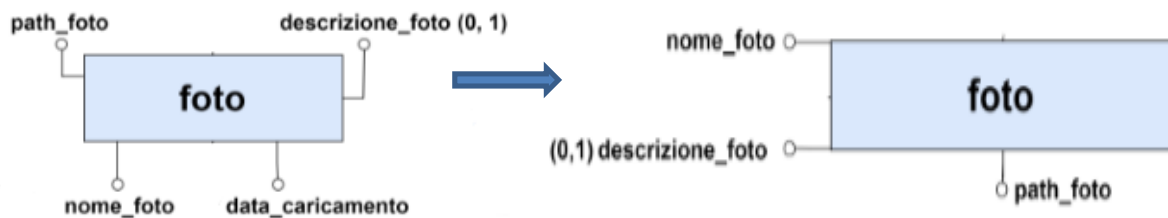
1.3 Schema ER ristrutturato



OPERAZIONI DI RISTRUTTURAZIONE SCHEMA ER

Passo (a) – analisi delle ridondanze

- Eliminazione dell'attributo **data_caricamento** dell'entità **foto**. La data di caricamento di una foto può essere ricavata dall'attributo **data_ora_cinguettio** dell'entità **cinguettio**. L'attributo è ridondante e considerate le operazioni interne alla base di dati si è optato per la sua eliminazione.



Passo (b) – partizionamento/accorpamento entità

- Considerate le operazioni interne al database si è deciso di partizionare l'entità **utente** in due entità:
 - info_utente**
 - utente**

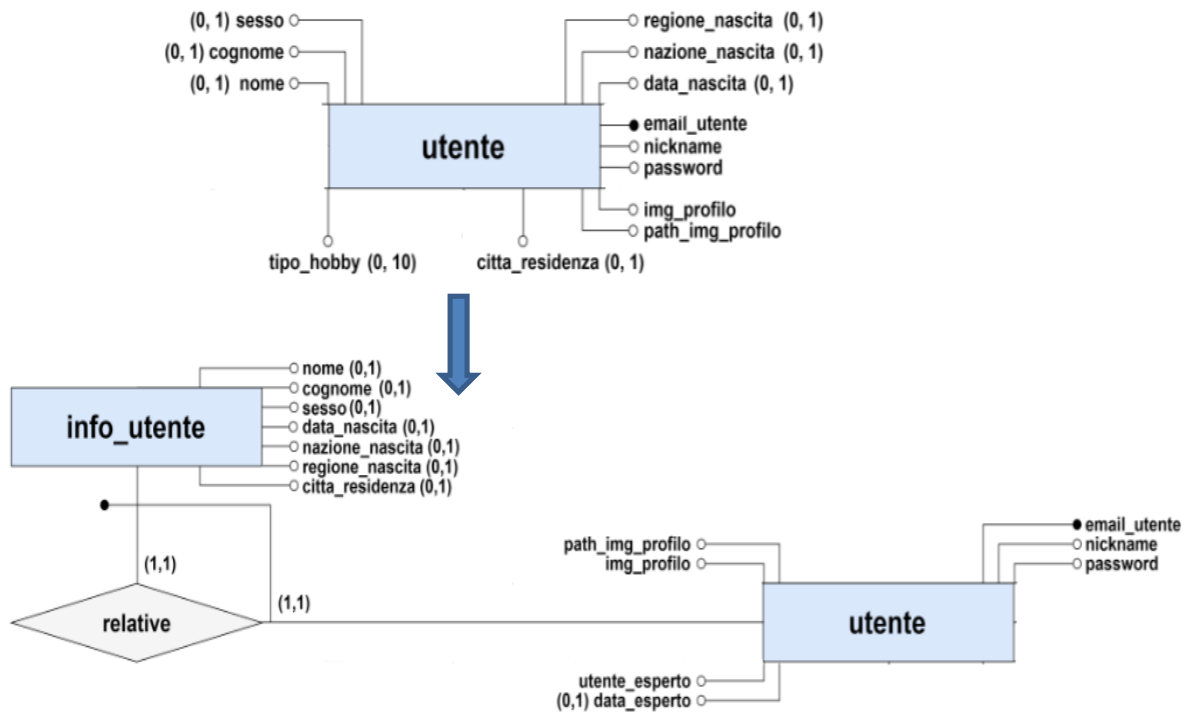
All'interno di **info_utente** sono contenute tutte le informazioni personali degli utenti:

- nome**
- cognome**
- sex**
- data_nascita**
- nazione_nascita**
- regione_nascita**
- città_residenza**

All'interno di **utente** sono invece contenute le informazioni di profilo dell'utente

ovvero:

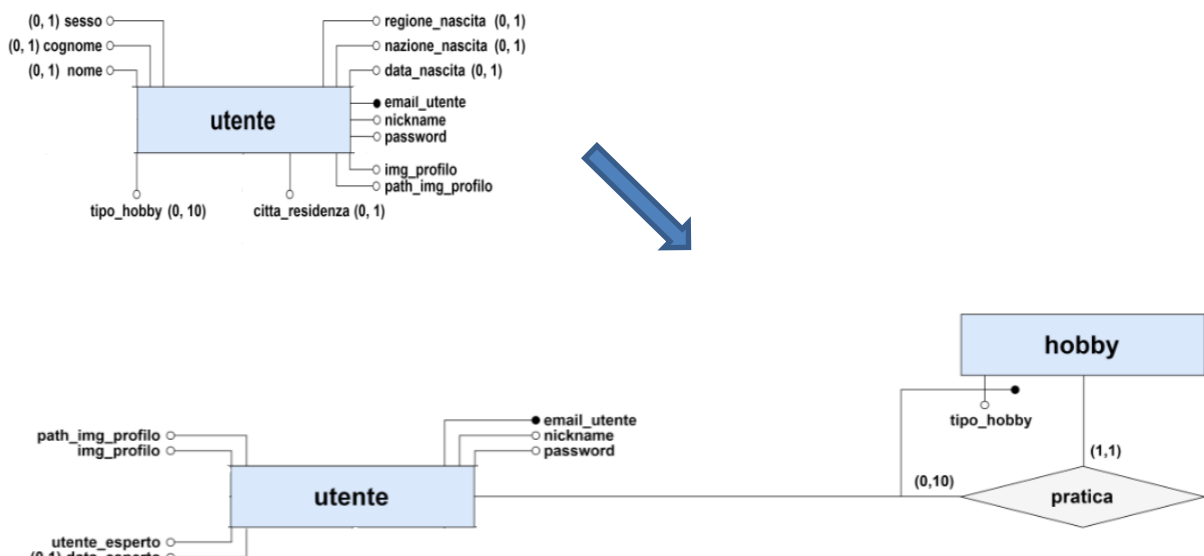
- email_utente**
- nickname**
- password**
- utente_esperto**
- data_esperto**
- img_profilo**
- path_img_profilo**



Questo partizionamento ci garantisce una serie di vantaggi in termini di efficienza. Le frequenti operazioni che riguardano l'entità **utente** richiedono l'accesso ad un limitato numero di informazioni, in particolare quelle di profilo, come **email_utente** e **nickname** ad esempio. L'accesso alle informazioni personali degli utenti come **nome**, **cognome**, **data_nascita** ecc. avviene invece più raramente. L'utilizzo di join ci costringerebbe quindi a lavorare con una mole di dati notevole e non indispensabile che andrebbe a inficiare sulle prestazioni. Sulla base di queste considerazioni si è pertanto deciso optare per un partizionamento.

Passo (c) – eliminazione degli attributi composti e multi valore

- Non sono presenti attributi composti
- L'attributo multi valore **tipo_hobby** dell'entità **utente** viene modellato con la definizione di una nuova entità **hobby**.



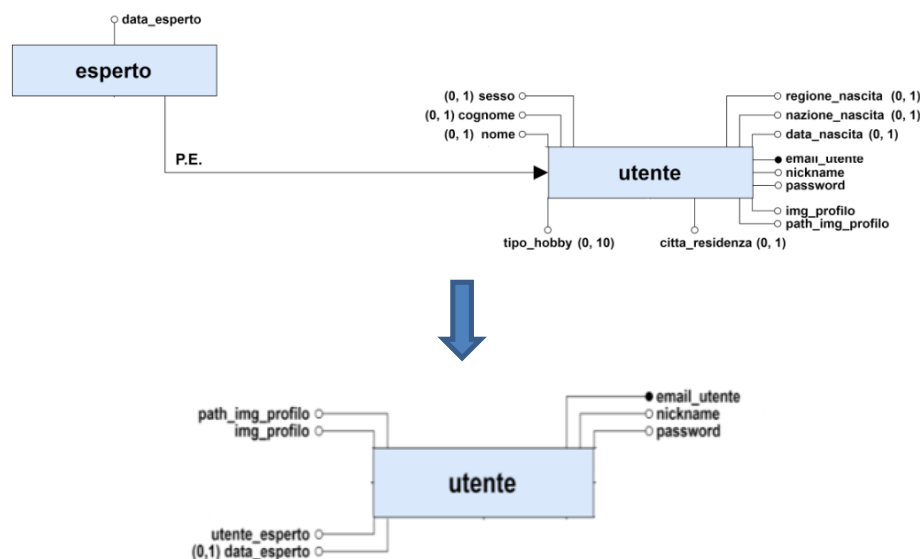
Passo (d) – eliminazione gerarchie di generalizzazione

- Le gerarchie di generalizzazione presenti sono:

Entità padre	Entità figlia	Tipologia
utente	esperto	Parziale Esclusiva
cinguettio	testo foto luogo	Totale Esclusiva

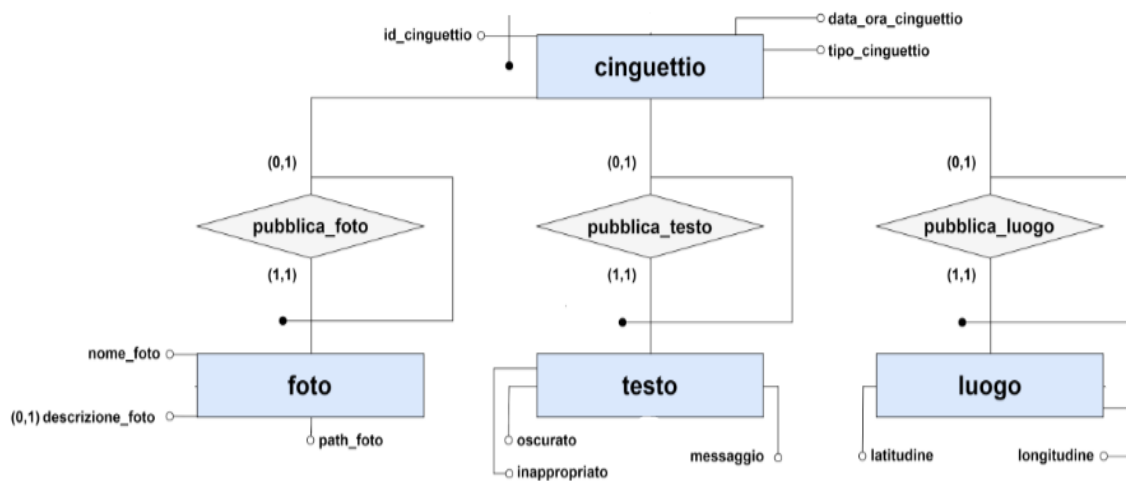
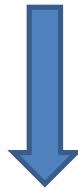
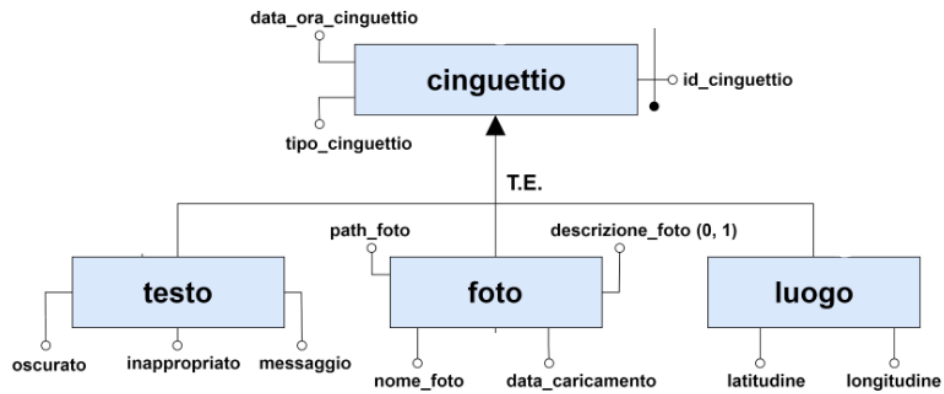
Per quanto riguarda l'entità figlia **esperto** si è deciso per la sua eliminazione e per il mantenimento dell'entità padre **utente**.

L'entità padre **utente** eredita l'attributo **data_esperto** e viene inserita all'interno di utente l'attributo **utente_esperto** per di identificare gli utenti esperti.



utente eredita anche l'associazione che legava esperto con l'entità **giudizio_foto** (vedere lo schema ER ristrutturato). Dato il numero esiguo di attributi presenti in **esperto** e considerate le operazioni di aggiornamento dello stato di un utente che da utente "normale" diventa **esperto** e viceversa, si è deciso di optare per questo tipo di ristrutturazione.

Per quanto riguarda la entità figlie **testo**, **foto**, **luogo** dell'entità padre **cinguettio**, la soluzione adottata è stata quella di mantenere sia l'entità padre che le entità figlie, sostituendo la gerarchia con delle associazioni. Il motivo alla base di questa scelta è stata quella di mantenere un controllo centralizzato degli **id_cinguettio** dei vari cinguettii. Eliminare l'entità padre avrebbe comportato dover inserire dei controlli per assicurare che ogni **id_cinguettio** fosse univoco per ogni tipo di cinguettio pubblicato da un medesimo utente. Eliminare invece le entità figlie e mantenere l'entità padre avrebbe reso l'entità **cinguettio** ridondante dal punto di vista dei dati. A fronte di queste considerazioni, si è pertanto deciso, di optare per il mantenimento sia dell'entità padre, che delle entità figlie.



Ogni volta che uno specifico utente crea un cinguettio viene generato un **id_cinguettio** il quale sarà alternativamente l'identificatore di un cinguettio di tipo **testo**, **luogo** o **foto**. In questo modo, siamo sicuri che un cinguettio pubblicato ad esempio da un utente A abbia un **id_cinguettio** univoco; Non sarà infatti possibile trovare 2 cinguettii di tipo diverso pubblicati dal medesimo utente con lo stesso identificatore.

VINCOLI D'INTEGRITA' RISTRUTTURAZIONE

Vincoli d'integrità aggiunti

V36: L'entità **cinguettio** deve essere alternativamente associata con l'entità **foto** o con l'entità **luogo** o con l'entità **testo**

V37: Quando l'entità **cinguettio** è associata con l'entità **luogo** il valore dell'attributo **tipo_cinguettio** deve essere impostato a 'L'. Quando l'entità **cinguettio** è associata con l'entità **testo** il valore dell'attributo **tipo_cinguettio** deve essere impostato a 'T'. Quando invece l'entità **cinguettio** è associata con l'entità **luogo** il valore dell'attributo **tipo_cinguettio** deve essere impostato a 'L'.

V38: Il valore dell'attributo **utente_esperto** dell'entità **utente** assume valore 'TRUE' quando un utente è seguito da almeno altri 3 utenti altrimenti vale 'FALSE'

V39: Una volta che un utente si registra al sistema l'attributo **email_utente** dell'entità **utente** dev'essere inserito all'interno dell'entità **info_utente**

Vincoli d'integrità modificati

V25: L'attributo **data_esperto** dell'entità **utente** corrisponde alla data corrente nel momento in cui un utente diventa un utente esperto altrimenti assume valore NULL .

Vincoli d'integrità eliminati

V22: L'attributo **data_caricamento** dell'entità **foto** corrisponde alla data corrente nel momento in cui viene caricata la foto

2. Progettazione logica e comandi SQL

SCHEMA REALAZIONALE

- **utente** (email_utente, nickname, password, utente_esperto, data_esperto₀,
img_profilo, path_img_profilo)
- **info_utente** (email_utente^{utente}, nome₀, cognome₀, sesso₀, data_nascita₀,
nazione_nascita₀, regione_nascita₀, citta_residenza₀)
- **hobby** (email_utente^{utente}, tipo_hobby)
- **cinguettio** (email_utente^{utente}, id_cinguettio, data_ora_cinguettio,
tipo_cinguettio)
- **testo** (email_utente^{cinguettio}, id_cinguettio^{cinguettio}, messaggio, inappropriato,
oscurato)
- **luogo** (email_utente^{cinguettio}, id_cinguettio^{cinguettio}, latitudine, longitudine)
- **foto** (email_utente^{cinguettio}, id_cinguettio^{cinguettio}, nome_foto, path_foto,
descrizione_foto₀)
- **giudizio_foto** (email_utente^{utente}, utente_giudicato^{foto}, id_cinguettio^{foto},
data_ora_commento, commento)
- **segnala_testo** (email_utente^{utente}, utente_segnalato^{testo}, id_cinguettio^{testo})
- **luogo_preferito** (email_utente^{utente}, utente_luogo^{luogo}, id_cinguettio^{luogo})
- **seguaci_seguiti** (utente_seguace^{utente}, utente_seguito^{utente},
data_ora_richiesta, richiesta_accettata, data_ora_risposta₀)

Le tabelle all'interno dello schema relazionale, sono formate da attributi semplici e da attributi non multivalore (esempio, tipo hobby).

Possiamo definire che lo schema è in 1NF, poiché ogni campo, contiene un solo valore. gli attributi multivalore diventano attributi semplici, spezzando la tupla che contiene il valore, in tante tuple quanti sono i valori descritti nell' attributo .

						email_utente	tipo_hobby	
<input type="checkbox"/>		Modifica		Copia		Elimina	alex@live.it	arte
<input type="checkbox"/>		Modifica		Copia		Elimina	alex@live.it	film e serie tv
<input type="checkbox"/>		Modifica		Copia		Elimina	alex@live.it	sport
<input type="checkbox"/>		Modifica		Copia		Elimina	alex@live.it	videogames
<input type="checkbox"/>		Modifica		Copia		Elimina	carmen@live.it	tecnologia
<input type="checkbox"/>		Modifica		Copia		Elimina	fra@gmail.com	fotografia

Possiamo definire che lo schema è 2NF poiché, è in 1NF e gli attributi delle tabelle sono dipendenti dall'intera chiave primaria.

- **utente** (email_utente, nickname, password, utente_esperto, data_esperto₀,
img_profilo, path_img_profilo)
- **info_utente** (email_utente^{utente}, nome₀, cognome₀, sesso₀, data_nascita₀,
nazione_nascita₀, regione_nascita₀, citta_residenza₀)

Possiamo definire che lo schema è 3NF poiché, è in 2NF e 1NF e gli attributi che non sono chiavi sono mutualmente indipendenti.

VINCOLI D'INTEGRITÀ SCHEMA RELAZIONALE

Vincoli d'integrità

V1: L'attributo **email_utente** della relazione **utente** ha una lunghezza massima di 70 caratteri ed ha il seguente formato '<utente>@<provider>.<dominio>' e può contenere lettere numeri e i seguenti caratteri [- _ .] è case insensitive. **[php]**

V2: L'attributo **password** dell'entità **utente** ha una lunghezza 1 e 50 caratteri e contiene solo caratteri alfanumerici e i seguenti caratteri [_ - ! * .] **[php]**

V3: L'attributo **nickname** della relazione è univoco ha una lunghezza tra 1 e 20 caratteri contiene solo caratteri alfanumerici e seguenti caratteri [' " - _ * ! .] e spazi **[php]**

V4: L'attributo **nome** della relazione **info_utente** ha una lunghezza massima di 20 caratteri e può contenere solo lettere il carattere di spazio e il carattere di apostrofo. **[php]**

V5: L'attributo **cognome** della relazione **info_utente** ha una lunghezza massima di 20 caratteri e può contenere solo lettere il carattere di spazio e il carattere di apostrofo. **[php]**

V6: L'attributo **data_nascita** della relazione **info_utente** deve essere minore o uguale alla data corrente nel momento in cui viene inserita. **[php]**

V7: L'attributo **nazione_nascita** della relazione **info_utente** ha una lunghezza massima di 50 caratteri e contiene solo lettere il carattere di spazio e il carattere di apostrofo. **[php]**

V8: L'attributo **regione_nascita** della relazione **info_utente** ha una lunghezza massima di 50 caratteri e contiene solo lettere il carattere di spazio e il carattere di apostrofo. **[php]**

V9: L'attributo **sex** della relazione **info_utente** può assumere uno tra i seguenti valori {'f', 'm'} dove 'm' indica che si tratta di un utente di sesso maschile mentre 'f' indica un utente di sesso femminile. **[php]**

V10: L'attributo **tipo_hobby** della relazione **hobby** può assumere i seguenti valori {'sport', 'musica', 'arte', 'tecnologia', 'film e serie tv', 'fotografia', 'moda e design', 'videogames', 'viaggiare'} l'utente può inserire un hobby aggiuntivo tramite una form testuale che non superi i 20 caratteri e che contiene solo lettere il carattere di spazio e il carattere di apostrofo. **[php]**

V11: L'attributo **citta_residenza** della relazione **info_utente** ha una lunghezza massima di 50 caratteri e contiene solo lettere il carattere di spazio e il carattere di apostrofo. **[php]**

V12: L'attributo **id_cinguettio** della relazione **cinguettio** è un valore intero che deve seguire una numerazione progressiva per ogni utente. **[trigger]**

V13: L'attributo **data_ora_cinguettio** della relazione **cinguettio** corrisponde alla data e all'ora corrente nel momento in cui viene creato il cinguettio. **[trigger]**

V14: L'attributo **tipo_cinguettio** della relazione **cinguettio** può assumere uno tra i seguenti valori {'T', 'F', 'L'} dove 'T' indica che si tratta di un cinguettio di tipo testo mentre 'F' che si tratta di un cinguettio di tipo foto 'L' che si tratta di un cinguettio di tipo luogo. **[trigger]**

V15: L'attributo **messaggio** della relazione **testo** ha una lunghezza massima di 50 caratteri e minima di 1 carattere. **[php]**

V16: L'attributo **inappropriato** della relazione **testo** è un valore booleano che assume valore 'TRUE' quando il cinguettio di tipo testo di un utente seguito viene segnalato come "inappropriato" da un altro utente altrimenti assume valore 'FALSE'. **[trigger]**

V17: L'attributo **oscurato** della relazione **testo** è un valore booleano che assume valore 'TRUE' nel caso in cui un cinguettio di tipo testo venga segnalato come inappropriato più di 3 volte altrimenti assume valore 'FALSE'. **[trigger]**

V18: Ogni utente può segnalare un cinguettio di tipo testo di un altro utente che segue come "inappropriato". Un utente non può segnalare un proprio cinguettio di tipo testo come "inappropriato" né un cinguettio che ha già segnalato come "inappropriato". **[php]**

V19: L'attributo **latitudine** della relazione **luogo** è un valore a virgola mobile compreso tra -180.000000 e 180.000000. **[php]**

V20: L'attributo **latitudine** della relazione **luogo** è un valore a virgola mobile compreso tra -180.000000 e 180.000000. **[php]**

V21: L'attributo **descrizione_foto** della relazione **foto** ha una lunghezza massima di 20 caratteri. **[php]**

V23: L'utente che ha almeno altri 3 utenti che lo seguono è definito "esperto". **[trigger]**

V24: Se un utente esperto ha un numero di seguaci inferiore a 3, smette di essere un utente esperto. **[trigger]**

V25: L'attributo **data_esperto** della relazione **utente** corrisponde alla data corrente nel momento in cui un utente diventa un utente esperto altrimenti assume valore NULL. **[trigger]**

V26: Un utente esperto può esprimere un giudizio di apprezzamento per i cinguettii di tipo foto pubblicati solo da utenti che segue o per i propri cinguettii di tipo foto. L'utente esperto non può esprimere un giudizio di apprezzamento più di 3 volte per uno stesso cinguettio di tipo foto. **[php]**

V27: L'attributo **commento** della relazione **giudizio_foto** ha una lunghezza massima di 50 caratteri e minima di 1 carattere. **[php]**

V28: L'attributo **data_ora_commento** della relazione **giudizio_foto** corrisponde alla data e all'ora corrente nel momento in cui viene commentata una foto. **[php]**

V29: Un utente può definire come luogo preferito un proprio cinguettio di tipo luogo o un cinguettio di tipo luogo pubblicato da un altro utente che segue ma non può definire uno stesso cinguettio di tipo luogo come luogo preferito più di 1 volta. **[php]**

V30: Un utente non può seguire sé stesso né può inviare una richiesta di seguire un utente che già segue **[php]**

V31: L'attributo **data_ora_richiesta** della relazione **seguaci_seguiti** corrisponde alla data e all'ora corrente nel momento in cui viene inviata una richiesta di seguire un utente. **[php]**

V32: L'attributo **data_ora_risposta** della relazione **seguaci_seguiti** corrisponde alla data e all'ora corrente nel momento in cui viene accettata una richiesta e deve essere maggiore dell'attributo **data_ora_richiesta** dell'associazione **seguaci_seguiti**. Se una richiesta è stata inviata ma non è stata ancora accettata assume valore 'NULL'. **[php]**

V33: L'attributo **richiesta_accettata** della relazione **seguaci_seguiti** è un valore booleano che assume valore 'TRUE' se la richiesta è stata accettata altrimenti assume valore 'FALSE' se la richiesta non è ancora stata accettata. **[php]**

V34: Un utente per poter vedere i cinguettii pubblicati da un altro utente accedere alle sue informazioni personali deve essere un suo seguace. **[php]**

V35: Un utente non può chiedere ad un altro utente di seguirlo se ha già inviato in precedenza una richiesta a cui non è stata data ancora una risposta. **[php]**

V36: La relazione **cinguettio** deve essere alternativamente associata con la relazione **foto**, **luogo** o **test**. **[trigger]**

V37: Quando la relazione **cinguettio** è associata con la relazione **luogo** il valore dell'attributo **tipo_cinguettio** deve essere impostato a 'L'. Quando la relazione **cinguettio** è associata con la relazione **testo** il valore dell'attributo **tipo_cinguettio** deve essere impostato a 'T'. Quando invece la relazione **cinguettio** è associata con la relazione **luogo** il valore dell'attributo **tipo_cinguettio** deve essere impostato a 'L'. **[trigger]**

V38: Il valore dell'attributo **utente_esperto** della relazione **utente** assume valore 'TRUE' quando un utente è seguito da almeno altri 3 utenti altrimenti vale 'FALSE'. **[trigger]**

V39: Una volta che un utente si registra al sistema l'attributo **email_utente** della relazione **utente** dev'essere inserito all'interno della relazione **info_utente**. **[trigger]**