

Progetto di Basi di Dati

L'obiettivo del progetto è la realizzazione di un database di un sito web per la gestione di blog culinari.

Il sistema permetterà di automatizzare il processo di raccolta delle ricette dei blog permettendo agli utenti di fruire delle informazioni specifiche ad esse correlate.

In particolare, sarà possibile, per gli utenti registrati:

- Seguire un determinato blog per accedere a tutte le ricette in esso contenute e possono esprimere le loro preferenze utilizzando dei likes, assegnare un punteggio da 1 a 5 stelle avendo inoltre la possibilità di lasciare delle recensioni.
- Accedere all'elenco delle ricette preferite (quelle su cui hanno messo dei likes) in modo da poter ritrovare facilmente le ricette che preferiscono.
- Ricercare le ricette in base agli ingredienti.
- Visualizzare l'elenco dei piatti del giorno dei blog che segue.

Docente: Sabrina Mantaci

Alessandro Alaimo **0695754**

Francesco La Rosa **0696471**

Dario Zappata **0699471**

Analisi dei requisiti

Il primo passo per procedere nella progettazione di una base di dati è l'analisi dei requisiti. Essa viene effettuata dopo una preliminare fase di raccolta, cioè l'individuazione delle caratteristiche che il sito web deve avere, con conseguente identificazione e risoluzione dei problemi ad esse legate. L'analisi è dunque l'organizzazione dei requisiti raccolti.

Gestione dei blog

Il sito deve essere gestito in maniera tale da poter permettere a diversi blog di postare le loro ricette. Per ciascun blog si deve, d'altro canto, poter specificare i generi di appartenenza e si deve essere in grado di ospitare le diverse inserzioni in essi contenuti.

Gestione delle ricette

Ogni ricetta presente sulla base di dati avrà una descrizione data direttamente dal blogger che l'ha pubblicata, tenendo quindi presente le indicazioni su quali ingredienti adottare per il piatto e come preparare tutti gli ingredienti descritti per essere cucinati, con le indicazioni per la loro cottura e tempistiche di cottura. Per ciascuna ricetta, il blogger potrà inoltre classificare la dieta e il genere di appartenenza, ed in fine gli strumenti adoperati per l'esecuzione del piatto.

Sistema per la gestione degli utenti

Affinché sia possibile soddisfare quanto richiesto è necessario implementare un sistema per la gestione degli utenti. Si delineano due categorie distinte di utenti: da una parte vi sono i blogger abilitati alla gestione dei blog ai fini dell'inserimento delle ricette, dall'altra gli utenti registrati al sito che andranno a consultare e quindi usufruire delle ricette disponibili.

In particolare agli utenti sarà permesso mettere likes e recensire le ricette dei soli blog da essi seguiti.

Glossario

Alla luce di quanto espresso sopra risulta utile un glossario alla comprensione e precisazione dei termini usati. Esso contiene, per ogni termine, una breve descrizione, i possibili sinonimi e i termini in esso contenuti con i quali esiste un legame logico.

Termine	Descrizione	Sinonimi	Collegamenti
Blog	Pagina di un sito web in cui sono contenute tutte le ricette.		Utente, Inserzione, Genere, Ricetta
Utente registrato	Utente che usufruisce delle ricette messe a disposizione dai blog.		Blog, Ricetta
Blogger	Utente che gestisce un blog e pubblica le ricette.		Blog, Ricette
Ricetta	Indicazione relativa agli ingredienti, alle dosi, alla preparazione di una pietanza	Piatto	Dieta, Strumento, Ingrediente, Blog, Utente Registrato, Genere,Blogger
Genere	Stile culinario con cui possono essere classificate le ricette		Blog, Ricetta
Inserzione	Annuncio pubblicitario presente all'interno del blog		Blog
Dieta	Particolare stile alimentare		Ricetta
Ingrediente	Ciò di cui si compone una ricetta		Ricetta
Strumento	Utensile utilizzato per l'esecuzione di una ricetta		Ricetta

Decomposizione

Da quanto è stato raccolto e analizzato durante la fase di analisi è utile trarre delle frasi omogenee.

Blog

Ogni blog, accolto dal sito, è identificato da un nome e una descrizione. Sono gestiti da blogger e su di essi vengono giornalmente pubblicate delle ricette. Ciascun blog possiede dei followers tra gli utenti registrati. E in ogni blog possono essere presenti delle inserzioni pubblicitarie. Per ciascuno di essi si può specificare, inoltre, uno o più generi culinari.

Ricetta

Relativamente alle ricette si indica il nome, il tempo di esecuzione, e la difficoltà, il blog che la contiene ed il blogger che l'ha pubblicata. Oltre agli ingredienti sono indicati gli strumenti e i passi necessari per la corretta esecuzione della ricetta. Per ciascuna ricetta viene indicata la portata o occasione, il genere culinario e le diete specifiche a cui appartiene.

Sistema per la gestione degli utenti

Ogni utente è identificato dalla propria email, rappresentiamo inoltre la sua password di accesso al portale, e i suoi dati anagrafici. Gli utenti possono essere di due tipi, utenti registrati (visualizzano le ricette messe a disposizione nei blog) e i blogger (gestiscono le ricette). Gli utenti registrati possono seguire vari blog ed esprimere, relativamente alle ricette in essi pubblicati, le proprie preferenze e opinioni servendosi di like e recensioni. I voti sono stabiliti da un sistema di classificazione che permette di assegnare un punteggio mediante l'assegnazione da 1 fino a 5 stelle.

Progettazione

A questo punto si procede con le fasi di progettazione che porteranno a definire, per affinamenti successivi, uno schema concettuale per il database.

Strategia

La strategia adottata al fine della realizzazione del progetto è l'inside-out, un caso particolare della bottom-up. In una prima fase sono stati identificati dei concetti principali, dalle considerazioni su di essi si sono definite delle entità, i loro attributi e le loro relazioni, fino a definire lo schema completo.

Nella bottom-up tradizionale le specifiche iniziali sono suddivise in componenti via via sempre più piccole, fino a quando non si ottengono delle componenti elementari. Esse vengono rappresentate da semplici schemi concettuali che vengono poi integrati tra di loro fino a giungere allo schema concettuale finale.

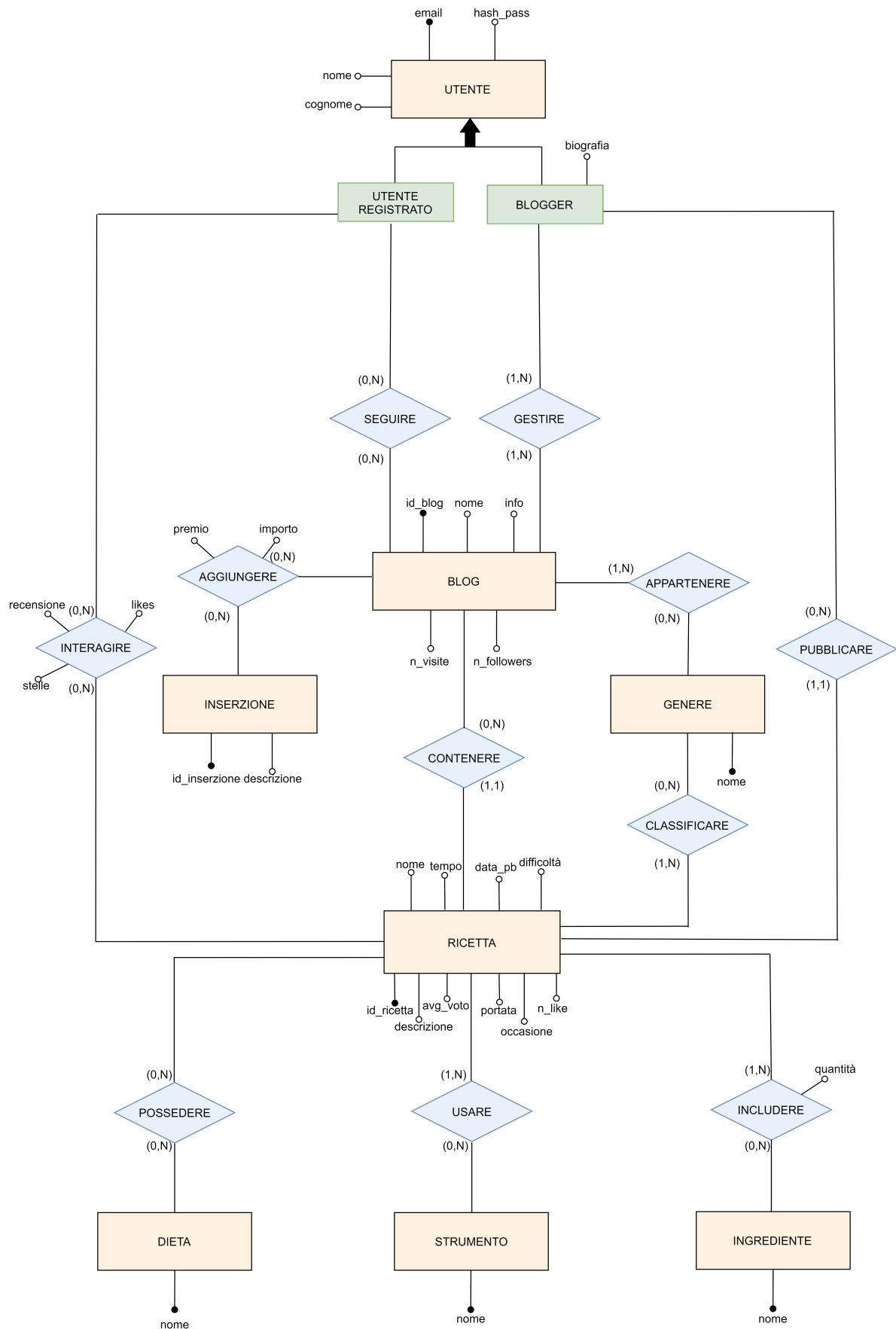
L'inside-out presenta il vantaggio di non richiedere fasi di integrazione per via della sua estensione a "macchia d'olio". Lo svantaggio è quello di dover esaminare tutte le specifiche per individuare i concetti non ancora rappresentati ed in relazione con quelli già presenti nello schema.

I concetti base a partire dai quali si può procedere alla rappresentazione di tutto lo schema sono:

- gestione utenti
- gestione del blog
- gestione delle ricette

Schema concettuale

Lo schema concettuale è un modello teorico per la rappresentazione concettuale e grafica dei dati.



Gestione degli utenti

Per prima cosa è stata definita l'entità **utente**, con due generalizzazioni, una per i blogger ed una per gli utenti registrati. Dai **blogger** partirà poi la relazione per la gestione dei blog e per la pubblicazione delle **ricette**. Dagli **utenti registrati** invece partiranno le relazioni per seguire i blog ed interagire con le ricette.

Gestione blog

Il sito tiene traccia dei **blog** culinari che abbiamo definito come un' entità.

Ogni blog è dunque collegato al **blogger** che lo gestisce, agli **utenti registrati** per determinare il numero dei follower, all'**inserzione** pubblicitaria presente al suo interno la quale renderà in base al numero di follower e di accessi al blog , ai **generi** culinari che determineranno le tipologie delle ricette al suo interno. Infine ogni blog è collegato alle **ricette** che contiene.

Gestione ricette

La ricetta viene ad essere l'entità fondamentale di tutto lo schema.

Alcuni degli elementi che descrivono la ricetta sono stati inseriti in entità a parte, come gli **ingredienti** e gli **strumenti** da utilizzare per consentire la molteplicità di utilizzo di questi elementi.

Inoltre, ogni ricetta sarà collegata:

- al **blog** che la contiene
- al blogger che la **pubblica**
- a una o più **diete** ad essa associata
- ad uno o più **generi** culinari
- agli **utenti registrati** , che alla visione della ricetta, possono decidere se lasciare dei likes, una votazione in stelle ed eventualmente una recensione.

Operazioni

Dopo aver raccolto e analizzato i dati, bisogna procedere con una fase analoga anche per le operazioni da effettuare sui dati, identificando anche informazioni riguardo la frequenza con cui avvengono. In questo specifico contesto si distinguono tre classi di operazioni.

Operazioni Utente

- **Accounting**

Sono le operazioni per la registrazione (o eventualmente la cancellazione) di un utente al sito, indicando i suoi dati tra cui email e password per effettuare l'accesso. Risultano essere necessarie per poter ricercare e visualizzare le ricette dei vari blog.

In media si registrano o si cancellano dal sito 200 nuovi utenti al mese.

- **Interazione con ricette**

Consiste nella visualizzazione con eventuale aggiunta di like, votazione o recensione di una ricetta in un determinato blog da parte degli utenti registrati.

Il sito riceve circa 5000 visualizzazioni di ricette al giorno da parte degli utenti.

Operazioni Blog

- **Creazione blog**

Consiste nell'operazione di creazione (o eventualmente la cancellazione) del blog da parte di un blogger indicandone i vari dati. In media si registrano o si cancellano dal sito 15 blog al mese.

- **Inserimento inserzione**

Potrebbero esserci degli inserimenti di inserzioni pubblicitarie nei vari blog del sito.

- **Premio da inserzione**

Le aziende che associano una inserzione a un blog possono premiarli con una certa cifra in denaro che dipende dal "successo" del blog stesso: il successo viene determinato in base al numero di followers e di visualizzazioni che ciascun blog racimola.

Operazioni Ricette

- **Pubblicazione ricette**

Questa operazione consiste nella pubblicazione da parte di un blogger di una ricetta descrivendone i vari dati, strumenti, ingredienti da utilizzare, generi e diete di appartenenza.

I blogger inseriscono ciascuno una ricetta al giorno.

- **Ricerca ricette**

Inserendo al più tre ingredienti l'utente può effettuare una ricerca più specifica delle ricette.

- **Visualizzazione elenco ricette del giorno**

Con cadenza giornaliera, ogni utente può visualizzare l'elenco dei piatti del giorno dei blog che segue.

- **Visualizzazione elenco ricette preferite**

Gli utenti possono richiedere di accedere all'elenco delle ricette preferite, cioè quelle a cui hanno messo dei like.

Altre operazioni

- **Classifica migliori blog/ricette:**

Ogni anno il sito premia i tre migliori blog ospitati dal sito, dove la classifica viene stilata in base al numero dei followers, e le tre migliori ricette per cui la classifica viene fatta in base alla media dei punteggi ottenuti, ma solo per quelle ricette che hanno ricevuto almeno 200 like.

- **Operazione automatiche database**

Ogni anno automaticamente vengono eliminati dal sito i blog che non hanno avuto attività durante l'anno.

Ristrutturazione

La fase di ristrutturazione è importante per semplificare la traduzione del modello concettuale nel modello logico, e per ottimizzare le prestazioni. In questa fase risulta utile considerare la mole di dati coinvolti e la frequenza con le quali le operazioni vi operano, seppur in maniera indicativa.

Tavola dei volumi

Supponiamo di avere

- 24.000 utenti di cui 20.400 utenti registrati e 3.600 bloggers;
- 1.200 blog di cui ognuno è gestito in media da 3 bloggers;
- Ogni blog contiene in media 2.000 ricette di cui ognuna ha in media 4 strumenti ed è composta da 9 ingredienti;
- Consideriamo 150 tipi di inserzioni diverse ed ogni blog contiene in media 15 inserzioni;
- 580 ingredienti;
- 200 strumenti;
- 50 diete;
- 500 generi;
- Consideriamo che ogni utente segua in media 30 blog;
- Consideriamo che ogni blog appartenga in media a 15 generi culinari diversi;
- Consideriamo che ogni ricetta sia classificata in media in 4 generi culinari diversi ed appartenga ad 8 diete differenti;
- Consideriamo inoltre che ogni utente interagisca con il 20% delle ricette pubblicate dai blog che segue.

Concetto	Tipo	Volume
Utente	E	24.000
Blog	E	1.200
Inserzione	E	150
Genere	E	500
Ricetta	E	2.400.000
Strumento	E	200
Dieta	E	50
Ingrediente	E	580
Gestire	A	3.600
Seguire	A	612.000
Aggiungere	A	18.000
Appartenere	A	18.000
Classificare	A	9.600.000
Contenere	A	2.400.000
Interagire	A	244.800.000
Possedere	A	19.200.000
Includere	A	21.600.000
Usare	A	9.600.000
Pubblicare	A	2.400.000

Tavola delle frequenze delle operazioni

La seguente tabella rappresenta delle stime su quelle che potrebbero essere le frequenze con cui le varie operazioni vengono effettuate giornalmente.

Operazione	Frequenza giornaliera
Registrazione utente	6,7
Cancellazione utente	6,7
Registrazione blog	0,5
Cancellazione blog	0,5
Pubblicazione ricette	1.200
Visualizzazioni ricette al giorno	5.000
Visualizzazione elenco ricette del giorno	20.400
Classifica migliori blog	0,003
Classifica migliori ricette	0,003
Ricerca ricette per ingredienti	3.500
Premio da inserzione	0,033

Tavola degli accessi

Alla luce dei volumi e delle frequenze rappresentate sopra, si può determinare il costo delle varie operazioni.

Per fare ciò si costruisce una tavola degli accessi basata su uno schema di navigazione.

Registrazione utente:

Concetto	Costrutto	Accesso	Tipo
Utente	Entità	1	Scrittura

- Accessi in scrittura : 1
- Costo : $(6,7 \times 2) = 13,4$

Cancellazione utente:

Concetto	Costrutto	Accesso	Tipo
Utente	Entità	1	Lettura
Utente	Entità	1	Scrittura

- Accessi in lettura : 1
- Accessi in scrittura : 1
- Costo : $(6,7 + 6,7 \times 2) = 20,1$

Registrazione blog:

Concetto	Costrutto	Accesso	Tipo
Blog	Entità	1	Scrittura
Gestire	Relazione	3	Scrittura
Blogger	Entità	3	Lettura
Appartenere	Relazione	15	Scrittura
Genere	Entità	15	Lettura
Aggiungere	Relazione	15	Scrittura
Inserzione	Entità	15	Lettura

- Accessi in lettura: 33
- Accessi in scrittura: 34
- Costo : $(33 + 2 \times 34) = 101 \times 0,5 = 50,5$

Cancellazione blog:

Concetto	Costrutto	Accesso	Tipo
Blog	Entità	1	Lettura
Blog	Entità	1	Scrittura

- Accessi in lettura : 1
- Accessi in scrittura : 1
- Costo : (1 + 1 x 2) = 3 x 0,5 = **1,5**

Pubblicazione ricetta:

Concetto	Costrutto	Accesso	Tipo
Blogger	Entità	1	Lettura
Pubblicare	Relazione	1	Scrittura
Ricetta	Entità	1	Scrittura
Blog	Entità	1	Lettura
Contenere	Relazione	1	Scrittura
Classificare	Relazione	4	Scrittura
Genere	Entità	4	Lettura
Includere	Relazione	9	Scrittura
Ingrediente	Entità	9	Lettura
Usare	Relazione	4	Scrittura
Strumento	Entità	4	Lettura
Possedere	Relazione	8	Scrittura
Dieta	Entità	8	Lettura

- Accessi in lettura : 27
- Accessi in scrittura : 28
- Costo : (27 + 28 x 2) = 83 * 1.200 = **99.600**

Visualizzazioni ricette al giorno:

Concetto	Costrutto	Accesso	Tipo
Utente Registrato	Entità	1	Lettura
Interagire	Relazione	1	Lettura
Ricetta	Entità	1	Lettura
Contenere	Relazione	1	Lettura
Blog	Entità	1	Lettura
Blog	Entità	1	Scrittura
Classificare	Relazione	4	Lettura
Genere	Entità	4	Lettura
Includere	Relazione	9	Lettura
Ingrediente	Entità	9	Lettura
Usare	Relazione	4	Lettura
Strumento	Entità	4	Lettura
Possedere	Relazione	8	Lettura
Dieta	Entità	8	Lettura

- Accessi in lettura : 55
- Accessi in scrittura : 1
- Costo : $(55 + 1 \times 2) \times 5.000 = \mathbf{285.000}$

Visualizzazione elenco ricette del giorno:

Concetto	Costrutto	Accesso	Tipo
Utente Registrato	Entità	1	Lettura
Seguire	Relazione	1	Lettura
Blog	Entità	1	Lettura
Blog	Entità	1	Scrittura
Contenere	Relazione	1	Lettura
Ricetta	Entità	1	Lettura
Classificare	Relazione	4	Lettura
Genere	Entità	4	Lettura
Includere	Relazione	9	Lettura
Ingrediente	Entità	9	Lettura
Usare	Relazione	4	Lettura
Strumento	Entità	4	Lettura
Possedere	Relazione	8	Lettura
Dieta	Entità	8	Lettura

- Accessi in lettura : 55
- Accessi in scrittura : 1
- Costo : $(55 + 1 \times 2) \times 20.400 = \mathbf{1.162.800}$

Classifica migliori blog:

Concetto	Costrutto	Accesso	Tipo
Blog	Entità	1.200	Lettura

- Accessi in lettura : 1.200
- Costo : $1.200 \times 0,003 = \mathbf{3,6}$

Classifica migliori ricette:

Concetto	Costrutto	Accesso	Tipo
Ricetta	Entità	2.400.000	Lettura

- Accessi in lettura : 2.400.000
- Costo : $2.400.000 \times 0,003 = \mathbf{7.200}$

Ricerca ricette per ingredienti:

Concetto	Costrutto	Accesso	Tipo
Ricetta	Entità	2.400.000	Lettura
Includere	Relazione	3	Lettura
Ingrediente	Entità	3	Lettura

- Accessi in lettura : 2.400.006
- Costo : $2.400.006 \times 3.500 = \mathbf{8.400.021.000}$

Premio da inserzione:

Concetto	Costrutto	Accesso	Tipo
Blog	Entità	1.200	Lettura

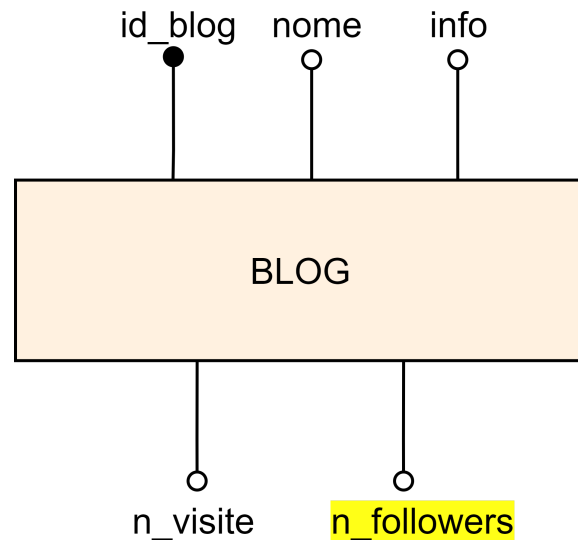
- Accessi in lettura : 1.200
- Costo : $1.200 \times 0,033 = \mathbf{39,6}$

Analisi delle ridondanze

Una ridondanza in uno schema entity-relationship è una informazione significativa ma derivabile da altre. In questa fase si decide se eliminare o mantenere le ridondanze.

Dalla visione dello schema concettuale si evince che ne sono presenti. Segue una analisi basata sul costo delle operazioni in assenza di esse.

Numero followers



L'attributo **n_followers** dei blog risulta essere ridondante, infatti esso viene calcolato facendo il count dei record dell'associazione **seguire**.

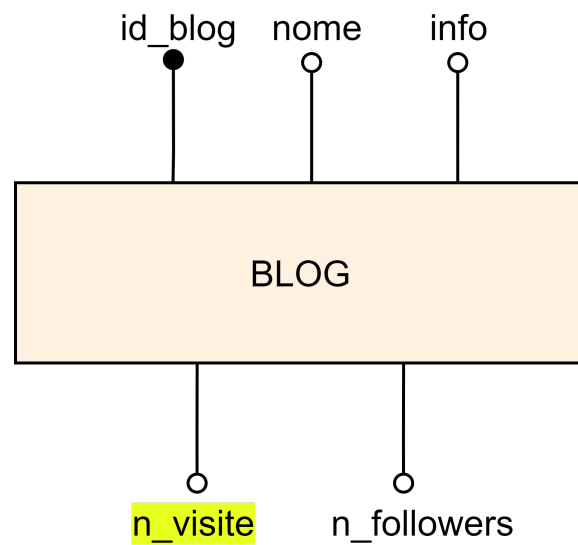
Considerando, ed esempio, l'operazione di **classifica migliori blog** si ha che l'assenza della ridondanza porterebbe ad un incremento dei costi:

Concetto	Costrutto	Accesso	Tipo
Blog	Entità	1.200	Lettura
Seguire	Relazione	680	Lettura

- Accessi in lettura : 1.880
- Costo : $1.880 \times 0,003 = \mathbf{5,64}$

Il costo passerebbe quindi da **3,6** ad essere **5,64**. Quindi risulta conveniente mantenere la ridondanza.

Numero visite



L'attributo **n_visite** dei blog risulta essere ridondante, infatti esso viene calcolato facendo il count dei record dell'associazione **interagire** delle ricette che appartengono al singolo blog.

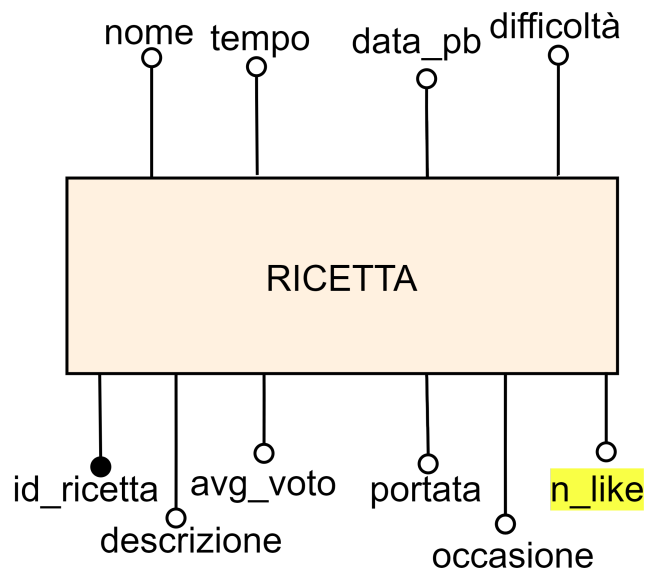
Considerando, ed esempio, l'operazione di **premio da inserzione** si ha che l'assenza della ridondanza porterebbe ad un incremento dei costi:

Concetto	Costrutto	Accesso	Tipo
Blog	Entità	1.200	Lettura
Contenere	Relazione	2.000	Lettura
Ricetta	Entità	2.000	Lettura
Interagire	Relazione	400	Lettura

- Accessi in lettura : 5.600
- Costo : $5.600 \times 0,033 = \mathbf{184,8}$

Il costo passerebbe quindi da **39,6** ad essere **184,8**. Quindi risulta conveniente mantenere la ridondanza.

Numero like



L'attributo **n_like** della ricetta risulta essere ridondante, infatti esso viene calcolato facendo il count dei record dell'associazione **interagire** in cui è presente il like.

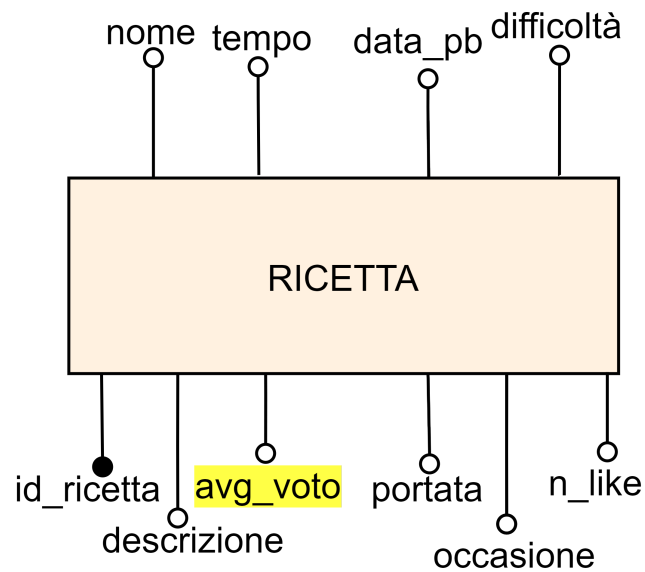
Considerando, ed esempio, l'operazione di **classifica migliori ricette** si ha che l'assenza della ridondanza porterebbe ad un incremento dei costi:

Concetto	Costrutto	Accesso	Tipo
Ricetta	Entità	2.400.000	lettura
Interagire	Relazione	400	lettura

- Accessi in lettura : 2.400.400
- Costo : $2.400.400 \times 0,003 = \mathbf{7.201,2}$

Il costo passerebbe quindi da **7.200** ad essere **7.201,2** . Quindi risulta conveniente mantenere la ridondanza.

Media voto



L'attributo **avg_voto** della ricetta risulta essere ridondante, infatti esso viene calcolato facendo la media dei voti dei record dell'associazione **interagire** in cui è presente la votazione.

Considerando, ed esempio, l'operazione di **classifica migliori ricette** si ha che l'assenza della ridondanza porterebbe ad un incremento dei costi:

Concetto	Costrutto	Accesso	Tipo
Ricetta	Entità	2.400.000	lettura
Interagire	Relazione	400	lettura

- Accessi in lettura : 2.400.400
- Costo : $2.400.400 \times 0,003 = \mathbf{7.201,2}$

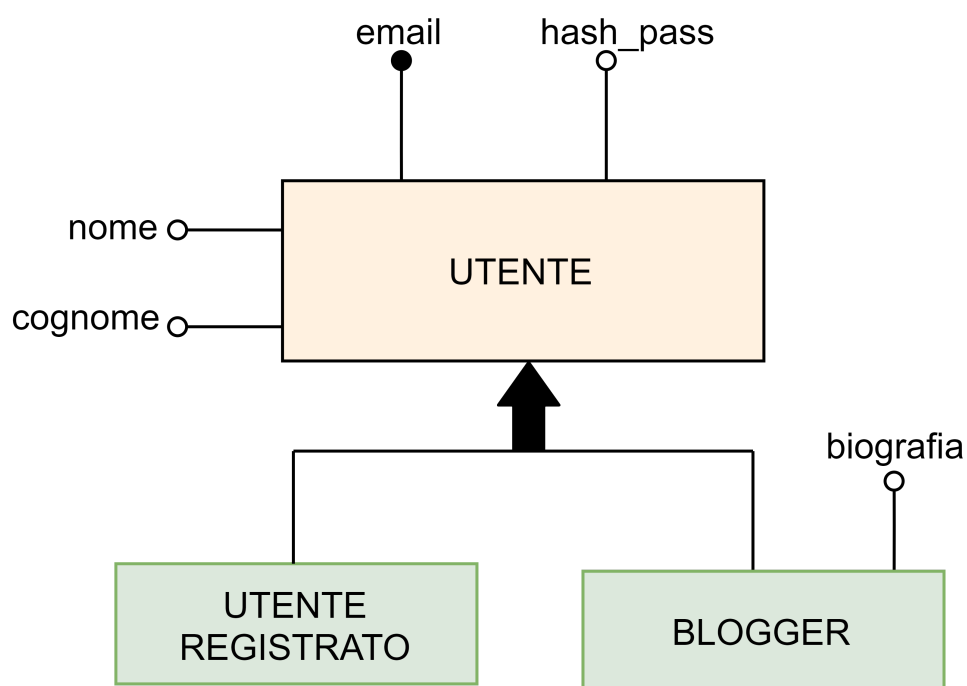
Il costo passerebbe quindi da **7.200** ad essere **7.201,2** . Quindi risulta conveniente mantenere la ridondanza.

Rimozione delle generalizzazioni

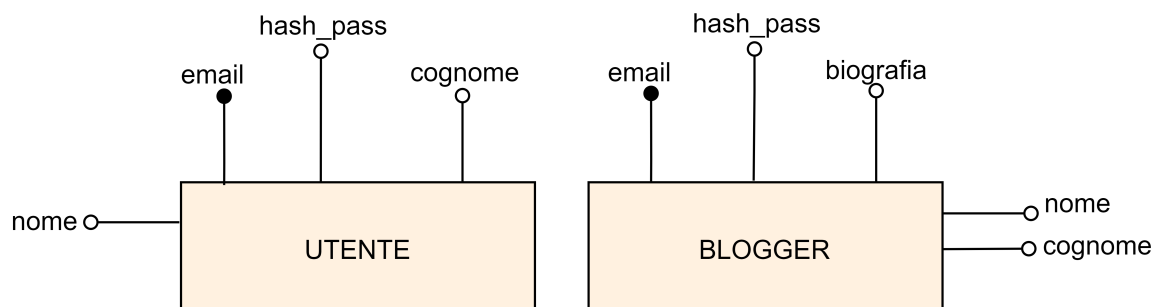
Poiché il modello relazionale non può rappresentare direttamente delle generalizzazioni si deve procedere alla loro eliminazione.

Utente

Prima: Dato che esistono due categorie di utenti risulta opportuno effettuare una generalizzazione totale di utente in utente registrato e blogger. A questo punto i due saranno differenziati solo per essere coinvolti in relazioni diverse e da un attributo aggiuntivo in blogger.



Dopo: Una possibile eliminazione di questa generalizzazione si ottiene mediante l'**accorpamento del genitore della generalizzazione nelle figlie**: l'entità genitore viene eliminata, e le entità figlie ereditano le proprietà (attributi, identificatore e relazioni) dell'entità genitore.



Schema concettuale ristrutturato

Dalla ristrutturazione si ottiene uno schema concettuale ristrutturato, pronto per essere tradotto in schema logico.

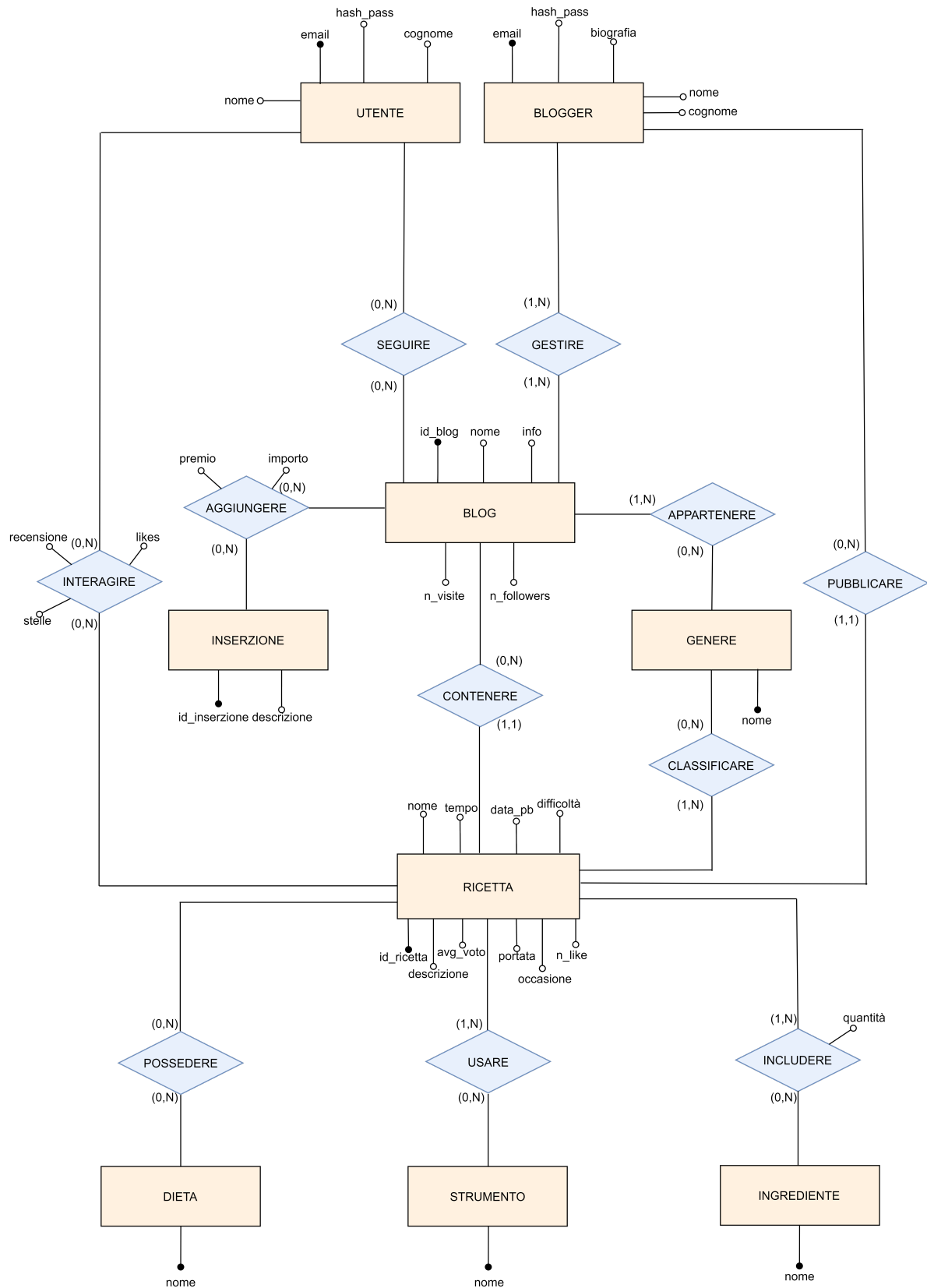


Tavola dei volumi dopo la ristrutturazione

Segue la tavola dei volumi alla luce delle ristrutturazioni effettuate.

Concetto	Tipo	Volume
Utente	E	20.400
Blogger	E	3.600
Blog	E	1.200
Inserzione	E	150
Genere	E	500
Ricetta	E	2.400.000
Strumento	E	200
Dieta	E	50
Ingrediente	E	580
Gestire	A	3.600
Seguire	A	612.000
Aggiungere	A	18.000
Appartenere	A	18.000
Classificare	A	9.600.000
Contenere	A	2.400.000
Interagire	A	244.800.000
Possedere	A	19.200.000
Includere	A	21.600.000
Usare	A	9.600.000
Pubblicare	A	2.400.000

Vincoli non esprimibili dallo schema concettuale

Oltre ai vincoli di integrità referenziale, naturalmente garantiti dalla presenza di relazioni, esistono altri tipi di vincoli che non possono essere rappresentati sulla base del solo schema concettuale.

Un utente può interagire solo con ricette di blog seguiti:

Dallo schema sopra riportato non è indicato nessun vincolo sull'utente registrato che è libero di recensire e visualizzare ciascuna ricetta di un blog registrato al sito, per questo motivo l'apparato dovrà mettere a disposizione tale opzione solo agli utenti che seguono tali blog.

Schema logico

Lo schema logico è una traduzione dello schema concettuale ristrutturato, dove vengono indicate le entità e le relazioni molti a molti con i loro attributi. E' la definizione delle tabelle relazionali, dalle quali si definiscono le tabelle SQL.

Le parole sottolineate indicano i vincoli di chiave primaria, le parole con l'asterisco i vincoli di chiave esterna.

- Utente(email, nome, cognome, hash_pass)
- Seguire(utente*, blog*)
- Interagire(utente*, ricetta*, recensione, likes, stelle)
- Blogger(email, nome, cognome, hash_pass, biografia)
- Gestire(blogger*, blog*)
- Blog(id_blog, nome, info, n_visite, n_followers)
- Aggiungere(blog*, inserzione*, importo, premio)
- Appartenere(blog*, genere*)
- Ricetta(id_ricetta, blogger*, blog*, nome, descrizione, data_pb, tempo, difficoltà, avg_voto, n_like, occasione, portata)
- Classificare(ricetta*, genere*)
- Includere(ricetta*, ingrediente*, quantità)
- Possedere(ricetta*, dieta*)
- Usare(ricetta*, strumento*)
- Genere(nome)
- Ingrediente(nome)
- Dieta(nome)
- Strumento(nome)
- Inserzione(id_inserzione, descrizione)

Codice SQL

Avendo definito nel dettaglio tutte le specifiche del database si può passare alla sua effettiva implementazione su un DBMS. Segue il codice SQL per poter fare ciò secondo le specifiche del database management system MySQL.

Creazione e definizione del database

Creazione del database `Sito_blog_cucina`

```
create database Sito_blog_cucina
```

Definizione della tabella `Utente`

Note: Il campo per la password è indicato come `hash_password` poiché il sistema verrà predisposto per non memorizzare le password in chiaro, di ogni password infatti si memorizzerà il relativo risultato di una funzione di hashing.

```
create table Utente (  
    email varchar(25) Primary key,  
    nome varchar(15) not null,  
    cognome varchar(15) not null,  
    hash_pass varchar(20) not null  
);
```

Definizione della tabella `Blogger`

```
create table Blogger (  
    email varchar(25) Primary key,  
    nome varchar(15) not null,  
    cognome varchar(15) not null,  
    hash_pass varchar(20) not null,  
    biografia varchar(20)  
);
```

Definizione della tabella `Blog`

```
create table Blog (  
    id_blog varchar(7) Primary key,  
    nome varchar(15) not null,  
    info varchar(1000) not null,  
    n_visite int default 0 check (n_visite>=0),  
    n_followers int default 0 check (n_followers>=0)  
);
```

Definizione della tabella Ricetta

```
create table Ricetta (  
    id_ricetta varchar(7) Primary key,  
    blogger varchar(25),  
    blog varchar(7),  
    nome varchar(20) not null,  
    descrizione varchar(7000) not null,  
    data_pb date not null,  
    tempo time not null,  
    difficolta varchar(15) not null,  
    avg_voto numeric(3,2) default 0 check (avg_voto>=0),  
    n_like int default 0 check (n_like>=0),  
    occasione varchar(25) not null,  
    portata varchar(20) not null,  
    foreign key (blogger) references Blogger(email) on delete set null,  
    foreign key (blog) references Blog(id_blog) on delete cascade  
);
```

Definizione della tabella Genere

```
create table Genere (  
    nome varchar(15) primary key  
);
```

Definizione della tabella Ingrediente

```
create table Ingrediente (  
    nome varchar(30) primary key  
);
```

Definizione della tabella Dieta

```
create table Dieta (  
    nome varchar(20) primary key  
);
```

Definizione della tabella Strumento

```
create table Strumento (  
    nome varchar(20) primary key  
);
```

Definizione della tabella Inserzione

```
create table Inserzione (  
    id_inserzione varchar(7) primary key,  
    descrizione varchar(50) not null  
);
```

Definizione della tabella `Seguire`

```
create table Seguire (  
    utente varchar(25),  
    blog varchar(7),  
    foreign key (utente) references Utente(email) on delete cascade,  
    foreign key (blog) references Blog(id_blog) on delete cascade,  
    primary key (utente, blog)  
);
```

Definizione della tabella `Interagire`

```
create table Interagire (  
    utente varchar(25),  
    ricetta varchar(7),  
    recensione varchar(400),  
    likes boolean default false,  
    stelle smallint check (stelle>0 and stelle<6),  
    foreign key (utente) references Utente(email) on delete cascade,  
    foreign key (ricetta) references Ricetta(id_ricetta) on delete cascade,  
    primary key (utente, ricetta)  
);
```

Definizione della tabella `Gestire`

```
create table Gestire (  
    blogger varchar(25),  
    blog varchar(7),  
    foreign key (blogger) references Blogger(email) on delete cascade,  
    foreign key (blog) references Blog(id_blog) on delete cascade,  
    primary key (blogger, blog)  
);
```

Definizione della tabella `Aggiungere`

```
create table Aggiungere (  
    blog varchar(7),  
    inserzione varchar(7),  
    importo numeric(6,2) not null,  
    premio numeric(6,2) default 0,  
    foreign key (blog) references Blog(id_blog) on delete cascade,  
    foreign key (inserzione) references Inserzione(id_inserzione)  
    on delete cascade,  
    primary key (blog, inserzione)  
);
```

Definizione della tabella `Appartenere`

```
create table Appartenere (  
    blog varchar(7),  
    genere varchar(15),  
    foreign key (blog) references Blog(id_blog) on delete cascade,  
    foreign key (genere) references Genere(nome) on delete cascade,  
    primary key (blog,genere)  
);
```

Definizione della tabella `Classificare`

```
create table Classificare (  
    ricetta varchar(7),  
    genere varchar(15),  
    foreign key (ricetta) references Ricetta(id_ricetta) on delete cascade,  
    foreign key (genere) references Genere(nome) on delete cascade,  
    primary key (ricetta,genere)  
);
```

Definizione della tabella `Includere`

```
create table Includere (  
    ricetta varchar(7),  
    ingrediente varchar(30),  
    quantita numeric(6,2) not null,  
    foreign key (ricetta) references Ricetta(id_ricetta) on delete cascade,  
    foreign key (ingrediente) references Ingrediente(nome) on delete cascade,  
    primary key (ricetta,ingrediente)  
);
```

Definizione della tabella `Possedere`

```
create table Possedere (  
    ricetta varchar(7),  
    dieta varchar(20),  
    foreign key (ricetta) references Ricetta(id_ricetta) on delete cascade,  
    foreign key (dieta) references Dieta(nome) on delete cascade,  
    primary key (ricetta,dieta)  
);
```

Definizione della tabella `Usare`

```
create table Usare (  
    ricetta varchar(7),  
    strumento varchar(20),  
    foreign key (ricetta) references Ricetta(id_ricetta) on delete cascade,  
    foreign key (strumento) references Strumento(nome) on delete cascade,  
    primary key (ricetta,strumento)  
);
```

Operazioni

Registrazione utente

Registrazione di un nuovo utente al sito indicando i suoi dati tra cui anche email e password per accedere.

```
insert into Utente
values (<EMAIL_UTENTE>, <NOME_UTENTE>, <COGNOME_UTENTE>, <PASSWORD_UTENTE>);
```

Cancellazione utente

Cancellazione di un utente registrato al sito.

```
delete from Utente
where email = <EMAIL_UTENTE>;
```

Registrazione blog

Operazione di creazione del blog da parte di un blogger indicandone i vari dati.

```
insert into Blog (id_blog, nome, info)
values (<ID_BLOG>, <NOME_BLOG>, <INFO_BLOG>);
```

Cancellazione blog

Cancellazione di un blog pubblicato nel sito.

```
delete from Blog
where id_blog = <ID_BLOG>;
```

Pubblicazione ricette

Pubblicazione da parte di un blogger di una ricetta descrivendone i vari dati.

```
insert into Ricetta (id_ricetta, blogger, blog, nome, descrizione, data_pb,
tempo, difficolta, occasione, portata)
values (<ID_RICETTA>, <NOME_BLOGGER>, <NOME_BLOG>, <NOME_RICETTA>,
<DESCRIZIONE_RICETTA>, <DATA_RICETTA>, <TEMPO_RICETTA>, <DIFFICOLTA_RICETTA>,
<OCCASIONE_RICETTA>, <PORTATA_RICETTA>);
```

Visualizzazione elenco ricette del giorno

Con cadenza giornaliera, ogni utente può visualizzare l'elenco delle ricette del giorno dei blog che segue.

```
select Ricetta.nome, descrizione, tempo, difficolta, occasione, portata
from Utente
join Seguire on Utente.email = Seguire.utente
join Ricetta on Ricetta.blog = Seguire.blog
where Utente = <EMAIL_UTENTE>
and data_pb = curdate();
```

Visualizzazione elenco ricette preferite

Ogni utente può visualizzare l'elenco delle ricette preferite cioè quelle in cui ha lasciato like.

```
select Ricetta.nome, descrizione, tempo, difficolta, occasione, portata
from Utente
join Interagire on Utente.email = Interagire.utente
join Ricetta on Interagire.ricetta = Ricetta.id_ricetta
where Utente = <EMAIL_UTENTE>
and likes = true
order by Ricetta.nome;
```

Visualizzazione ricette dei blog seguiti

Ogni utente può accedere a tutte le ricette dei blog che seguono.

```
select Ricetta.nome, descrizione, tempo, difficolta, occasione, portata
from Utente
join Seguire on Utente.email = Seguire.utente
join Ricetta on Ricetta.blog = Seguire.blog
where Utente = <EMAIL_UTENTE>;
```

Inserimento interazione con ricetta

Quando gli utenti visualizzano una ricetta possono esprimere le loro preferenze quali `<LIKES>`, `<RECENSIONE>` e `<STELLE>`.

```
insert into Interagire(utente, ricetta, recensione, likes, stelle)
values (<EMAIL_UTENTE>, <ID_RICETTA>, <RECENSIONE>, <LIKES>, <STELLE>);
```


Trigger per aggiornamento numero visite blog

Ogni volta che un utente interagisce con una ricetta viene incrementato l'attributo **n_visite** del blog che la contiene. Per automatizzare questo processo utilizziamo un trigger.

```
create trigger UpdateVisite
after insert on Interagire
for each row
update Blog
set n_visite = n_visite + 1
where new.ricetta = any (select Ricetta.id_ricetta
                        from Ricetta
                        where Ricetta.blog=Blog.id_blog);
```

Trigger per aggiornamento numero follower blog

Ogni volta che un utente segue un blog viene incrementato l'attributo **n_followers** di quel determinato blog. Per automatizzare questo processo utilizziamo dei trigger. Specifichiamo 2 casi, il primo che incrementa **n_followers** del blog seguito, ed il secondo che decrementa **n_followers** una volta cancellata l'associazione.

```
create trigger UpdateFollowers
after insert on Seguire
for each row
update Blog
set n_followers = n_followers + 1
where Blog.id_blog = new.blog;

create trigger UpdateFollowers2
after delete on Seguire
for each row
update Blog
set n_followers = n_followers - 1
where Blog.id_blog = old.blog;
```

Classifica migliori blog

Ogni anno si premiano i tre migliori blog ospitati dal sito, dove la classifica viene fatta in base al numero dei followers.

```
select nome, n_followers
from Blog
order by n_followers DESC, n_visite DESC
limit 3;
```

Trigger per aggiornamento numero like ricetta

Ogni volta che un utente lascia un like in una ricetta viene incrementato l'attributo **n_like** della ricetta. Per automatizzare questo processo utilizziamo dei trigger. Inseriamo 3 casi, il primo in cui il like viene aggiunto subito dopo la prima visualizzazione, il secondo quando viene modificato successivamente in true ed il terzo in cui viene modificato in false.

```
create trigger UpdateLike
after insert on Interagire
for each row
update Ricetta
set n_like = n_like + 1
where Ricetta.id_ricetta = new.ricetta and new.likes = true;

create trigger UpdateLike2
after update on Interagire
for each row
update Ricetta
set n_like = n_like + 1
where Ricetta.id_ricetta = new.ricetta and new.likes = true and old.likes=false;

create trigger UpdateLike3
after update on Interagire
for each row
update Ricetta
set n_like = n_like - 1
where Ricetta.id_ricetta = new.ricetta and new.likes = false and old.likes=true;
```

Trigger per aggiornamento media voti ricetta

Ogni volta che un utente lascia un voto in una ricetta viene aggiornato l'attributo **avg_voto** di quella ricetta. Per automatizzare questo processo utilizziamo un trigger.

```
create trigger UpdateMediaVoto
after insert on Interagire for each row
update Ricetta
set avg_voto = (( select sum(stelle)
                  from Interagire
                  where Interagire.ricetta = new.ricetta
                  and Interagire.stelle is not null ) /
                ( select count(*)
                  from Interagire
                  where
                    Interagire.ricetta = new.ricetta
                    and Interagire.stelle is not null))
where (Ricetta.id_ricetta = new.ricetta) and (new.stelle>0) and (new.stelle<6);
```

Classifica migliori ricette

Ogni anno il sito premia le tre migliori ricette, per cui la classifica viene fatta in base alla media dei punteggi ottenuti, ma solo per quelle ricette che hanno ricevuto almeno 200 like.

```
select nome, avg_voto
from Ricetta
where n_like >200
order by avg_voto DESC, n_like DESC
limit 3;
```

Ricerca ricette per ingredienti

Indicando fino a tre ingredienti (<INGREDIENTE1>, <INGREDIENTE2> ed <INGREDIENTE3>), il database deve essere in grado di restituire tutte le ricette che contengono quegli ingredienti, visualizzandole nell'ordine di preferenza complessiva, che viene calcolata come il numero di like ricevuti da tutti gli utenti.

```
select distinct Ricetta.nome, Ricetta.descrizione, tempo, difficolta, occasione,
portata
from Ricetta
join Includere on Includere.ricetta = Ricetta.id_ricetta
where (<INGREDIENTE1> and <INGREDIENTE2> and <INGREDIENTE3>) IN
      (select Ingrediente.nome
       from Ingrediente
       join Includere on
         Ingrediente.nome=Includere.ingrediente)
order by n_like;
```

Premio da inserzione

I blog possono essere premiati da parte delle aziende pubblicitarie che inseriscono in essi le inserzioni con un premio in denaro che dipende dal "successo" del blog (numero di follower, numero di accessi).

Si ottengono tutti i blog in cui è stata pubblicata una determinata <ID_INSERTIONE> .

```
select Blog.id_blog
from Blog
join Aggiungere on Blog.id_blog = Aggiungere.blog
where Aggiungere.inserzione = <ID_INSERTIONE>;
```

Una volta scelto il <BLOG_SELEZIONATO> da premiare ne aggiorna il premio mediante un <MULTIPLICATORE_VISITE> e un <MULTIPLICATORE_FOLLOWERS> a discrezione dell'azienda.

```
update Aggiungere
set premio = (<MULTIPLICATORE_VISITE> * <BLOG_SELEZIONATO>.n_visite) +
             (<MULTIPLICATORE_FOLLOWERS> * <BLOG_SELEZIONATO>.n_followers)
where blog = <BLOG_SELEZIONATO>
and inserzione = <ID_INSERTIONE>;
```

Cancellazione blog inattivo

Ogni anno l'amministratore elimina dal sito i blog che non hanno avuto attività durante l'anno.

Creiamo un evento che con cadenza annuale eliminerà i blog inattivi automaticamente.

```
create event CancellazioneBlog
on schedule every 1 year
starts '2020-01-01 00:00:00'
do
delete
from Blog
where ( DATEDIFF(curdate(), (select max(data_pb)
                             from Ricetta
                             where Blog.id_blog = Ricetta.blog))) >= 365;
```