Corso Basi di Dati B aa 2018/2019 - LABORATORIO
ESERCIZIO di Progettazione e realizzazione di una base dati

Relazione di Stefano Peddoni – Matricola: 839138

INDICE

1. Progettazione Concettuale:

1.1 Requisiti Iniziali: PAG 3
1.2 Glossario dei termini PAG 4
1.3 Requisiti rivisti e strutturati in gruppi di frasi omogenee PAG 5
1.4 Schema E-R PAG 6

2. Progettazione Logica:

2.1 Regole Aziendali: PAG 7
2.2 Tavola dei volumi: PAG 8
2.3 Tavola delle operazioni: PAG 10
2.4 Analisi delle ridondanze: PAG 11
2.5 Scelta degli identificatori principali: PAG 19
2.6 Schema E-R Ristrutturato + regole aziendali PAG 20
2.7 Schema Relazionale: PAG 22

3. Implementazione:

3.1 DDL (Data Definition Language) PAG 24
3.2 DML (Data Manipulation Language) PAG 32
3.3 Modifica ed eliminazione delle tabelle PAG 33

REQUISITI INIZIALI

Si vuole realizzare una base di dati per la gestione di una piattaforma in cui gli utenti possano scoprire e condividere ricette.

Per poter caricare le loro ricette, gli utenti devono registrarsi fornendo email, password e un nome utente che sarà associato a tutti i loro contributi.

Le ricette hanno un titolo, una descrizione, una foto di copertina, alcune informazioni su difficoltà, tempo di preparazione, ecc. ed una lista di ingredienti.

Per ogni ingrediente si specificano il nome, la quantità ed eventuali annotazioni.

Inoltre, ogni ricetta appartiene ad una categoria che identifica il tipo di portata (es. antipasto, primo piatto, …) e, in qualche caso, anche ad una o più categorie che identificano la dieta associata (es. vegetariana, senza glutine, ...).

Oltre che dagli utenti, le ricette possono essere pubblicate dalla redazione del sito.

Le ricette della redazione possono avere come autore un cuoco famoso.

Esplorando la pagina di un cuoco sul sito, i visitatori possono leggerne una breve bio e accedere a tutte le ricette di cui questi è l'autore (di cui è subito visibile anche il numero).

Gli utenti possono seguire un cuoco per visualizzare le sue ricette in primo piano nella propria pagina personale.

Le persone possono inoltre salvare le ricette tra i propri preferiti e singoli ingredienti in una "lista della spesa".

I preferiti e le cose da comprare sono visualizzati nella pagina personale degli utenti, insieme ad un'indicazione del numero di ricette e del numero di commenti pubblicati.

I visitatori possono infatti anche commentare le ricette, eventualmente rispondendo ad un commento già pubblicato da qualcun altro.

Anche la redazione può pubblicare dei commenti, ma solo in risposta ai commenti lasciati dai visitatori. Per ogni commento vengono visualizzati: lo *username* e la foto profilo dell'autore (o la dicitura "Redazione"), la data e il testo.

Ad ogni ricetta sono associate delle ricette simili.

Due ricette si considerano simili se hanno almeno una categoria in comune e hanno come autore lo stesso cuoco o utente.

Quando una persona cancella la propria iscrizione al sito, anche i suoi dati personali sono cancellati, mentre le ricette che ha condiviso e i suoi commenti vengono mantenuti.

Sono invece cancellati insieme ai dati personali i preferiti, gli ingredienti contenuti nella lista della spesa, la lista dei cuochi eventualmente seguiti.

GLOSSARIO DEI TERMINI

TERMINE	DESCRIZIONE	TERMINI COLLEGATI	SINONIMI
UTENTE	Dati personali caratteristici dell'utente	-Ricette -Cuoco -Commento	Visitatori Persone
RICETTE	E' un insieme di istruzioni per compiere un procedimento di trasformazione in generale, che attraverso varie alterazioni di una o più sostanze base (ingredienti), sia fisiche (prodotte tramite azioni) che chimiche (prodotte tramite mescolamenti e mutamenti nelle sostanze), dà come risultato qualcosa di diverso dalle materie originarie, solitamente di valore o utilità maggiore a quella degli ingredienti stessi.	-Utente -Commento -Ingredienti -Cuoco	
INGREDIENTI	Insieme di ingredienti per creare una ricetta	-Ricette	Componenti
CUOCO	Pagina personale del cuoco	-Utente -Ricette	
COMMENTO	Commento sul sito ricette	-Utente -Ricette	

REQUISITI RIVISTI E STRUTTURATI IN GRUPPI DI FRASI OMOGENEE

***Frasi* di CARATTERE GENERALE**: Si vuole realizzare una base di dati per la gestione di una piattaforma in cui gli utenti possano scoprire e condividere ricette.

***Frasi* relative agli UTENTI**: Per poter caricare le loro ricette, gli utenti devono registrarsi fornendo email, password e un nome utente che sarà associato a tutti i loro contributi.

Gli utenti possono seguire un cuoco per visualizzare le sue ricette in primo piano nella propria pagina personale. Possono leggerne una breve bio e accedere a tutte le ricette di cui questi è l'autore (di cui è subito visibile anche il numero). Possono infatti anche commentare le ricette, eventualmente rispondendo ad un commento già pubblicato da qualcun altro.

Quando un utente cancella la propria iscrizione al sito, anche i suoi dati personali sono cancellati, mentre le ricette che ha condiviso e i suoi commenti vengono mantenuti.

Gli utenti possono inoltre salvare le ricette tra i propri preferiti e singoli ingredienti in una "lista della spesa".

***Frasi* relative alle RICETTE**: Le ricette hanno un titolo, una descrizione, una foto di copertina, alcune informazioni su difficoltà, tempo di preparazione, ecc. ed una lista di ingredienti.

Ogni ricetta appartiene ad una categoria che identifica il tipo di portata (es. antipasto, primo piatto, …) e, in qualche caso, anche ad una o più categorie che identificano la dieta associata (es. vegetariana, senza glutine, ...).

Possono essere pubblicate sia dagli utenti che dalla redazione del sito.

Possono avere come autore un cuoco famoso.

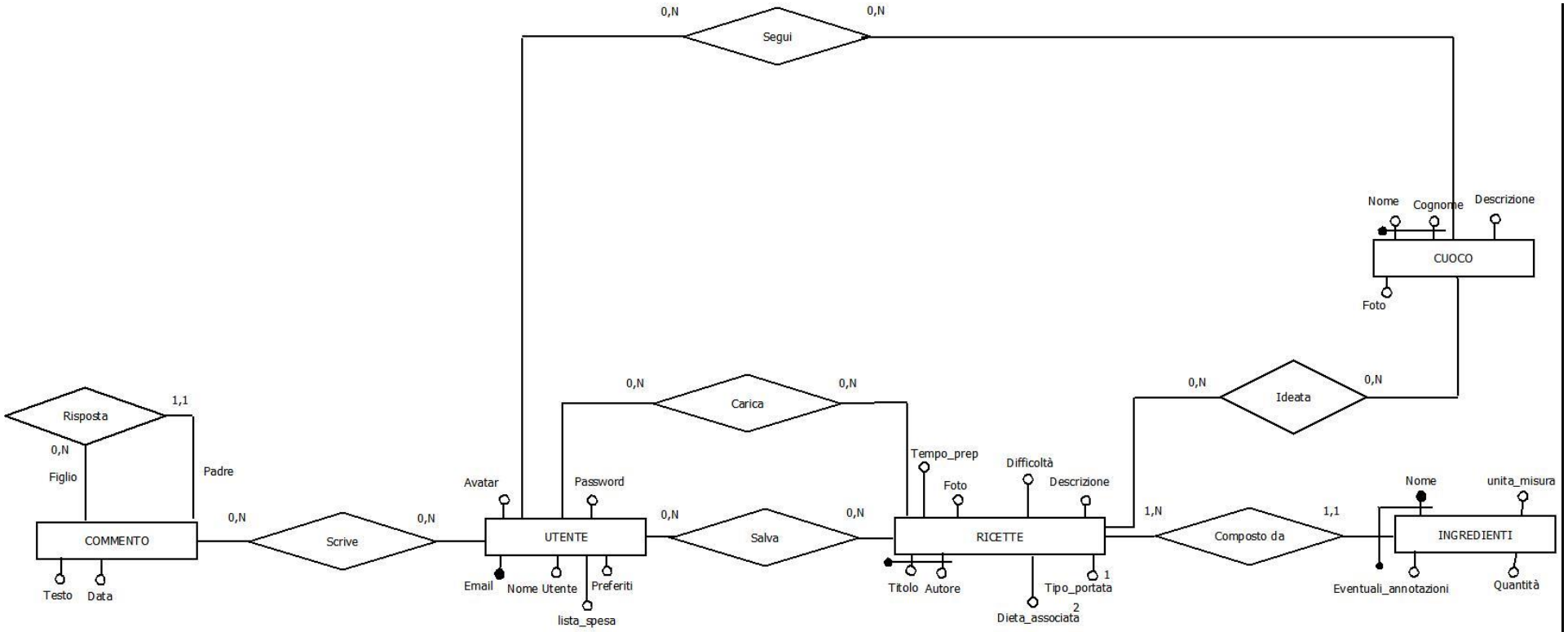
Ad ogni ricetta sono associate delle ricette simili.

Due ricette si considerano simili se hanno almeno una categoria in comune e hanno come autore lo stesso cuoco o utente.

***Frasi relative agli* INGREDIENTI**: Per ogni ingrediente si specificano il nome, la quantità ed eventuali annotazioni.

Frasi relative al CUOCO: Esplorando la pagina di un cuoco sul sito, gli utenti possono leggerne una breve bio e accedere a tutte le ricette di cui questi è l'autore (di cui è subito visibile anche il numero), è compresa anche una foto profilo.

Frasi relative ai COMMENTI: Per ogni commento vengono visualizzati: lo *username* e la foto profilo dell'autore (o la dicitura "Redazione"), la data e il testo. Anche la redazione può pubblicare dei commenti, ma solo in risposta ai commenti lasciati dagli utenti.



1: il tipo di portata può essere : dolce, primo, secondo, antipasto
2: la dieta associata può essere : senza glutine, veg, senza lattosio

REGOLE AZIENDALI

RV01: Gli utenti possono commentare le ricette rispondendo ad un commento già pubblicato da qualcun altro.

RV02: La redazione del sito può pubblicare commenti, ma solo in risposta ai commenti lasciati dagli utenti.

RV03: Due ricette possono considerarsi simili se hanno almeno una categoria in comune e hanno come autore lo stesso cuoco o utente.

RV04: Quando un utente si cancella dal sito vengono mantenute le ricette che ha condiviso e i suoi commenti.

RV05: Quando un utente si cancella dal sito, vengono cancellati: i dati personali, i preferiti, gli ingredienti contenuti nella lista della spesa e la lista dei cuochi eventualmente seguiti.

RD01: L'avatar associato sia al singolo utente che al singolo cuoco viene ricavato attraverso una stringa alfanumerica che rappresenta il percorso del file system dove risiede l'immagine fisicamente.

RD02: L'immagine di copertina associata al sito di ricette e alla pagina del cuoco viene ricavata attraverso una stringa alfanumerica che rappresenta il percorso del file system dove risiede l'immagine fisicamente.

TAVOLA DEI VOLUMI

Nella tavola dei volumi ci riferiremo alle unità di misura:

- MIGLIAIA: K
- MILIONI: MLN
- MILIARDI: MLD

L'analisi si basa su informazioni trovate in rete sulla piattaforma GIALLO ZAFFERANO, laddove non sono state trovate informazioni, procedo con delle assunzioni.

Concetto	Tipo	Volume
UTENTE	E	900 MLN
RICETTE	E	5.000
INGREDIENTI	E	45.000
CUOCO	E	40
COMMENTO	E	35.000
SCRIVE	R	2.25 x 10^12
SALVA	R	9x10^9
CARICA	R	500
COMPOSTO DA	R	45.000
SEGUI	R	1.35 x 10^10
RISPOSTA	R	7.000
IDEATA	R	2.000

Nel 2019 gli utenti iscritti al sito Giallo Zafferano supera i 14 milioni al mese e 7 milioni di fan sui social network, mentre nel 2011 erano di 2 milioni al mese.

A Gennaio 2019 le ricette condivise sul sito sono all'incirca 4.400 che contengono ora altre informazioni come calorie e valori nutrizionali fondamentali, e per ogni piatto forniscono indicazioni riguardanti intolleranze e regimi alimentari: senza glutine, senza lattosio, basso nichel, vegetariano.

Nello stesso anno, il tipo di portata (Primo, Secondo, Antipasto, Contorno, Dolci, Lievitati, Piatti Unici) ammonta a 7. Il tipo di dieta associata (Light, Senza Glutine, Senza Lattosio, Vegetariano, Basso Nichel) ammonta a 5.

Il numero di cuochi registrati sul sito Giallo Zafferano è all'incirca 40.

Procediamo con le assunzioni del caso :

Gli utenti nell'anno 2019 superano i 14 milioni di iscritti al mese; siccome nel 2011 gli iscritti erano appena di 2 milioni al mese, ipotizzo che il numero totale di utenti sulla piattaforma Giallo Zafferano alla fine del 2019 è all'incirca di 900 MLN.

Il numero di ricette a Gennaio 2019 era di 4.400, ipotizzo che a fine 2019 le ricette siano aumentate fino al raggiungimento di 5.000 ricette totali. Ad ogni ricetta ipotizzo che servono 9 ingredienti ciascuna, per un totale di 45.000.

Per l'entità COMMENTO, suppongo che per ogni ricetta ci siano almeno 7 commenti; quindi, il numero totale dei commenti sono all'incirca di 35.000.

Per l'associazione SCRIVE, suppongo che ogni utente pubblichi all'incirca 1 commento ogni 2 ricette, quindi 900 MLN x 2.500 = 2.25 x 10^12.

Per l'associazione SALVA, suppongo che gli utenti siano all'incirca 900 MLN e che ciascun utente salvi al più 100 ricette ciascuno, quindi il totale per l'associazione SALVA è di 900 MLN * 100 = 90.000.000.000 (9x10^9).

Per l'associazione CARICA, siccome le ricette totali sul sito Giallo Zafferano sono all'incirca 5.000 tra (ricette caricate dal cuoco, ricette caricate dalla redazione e ricette caricate dall'utente) suppongo che vengano caricate all'incirca 500 ricette da parte degli utenti.

Per l'associazione COMPOSTO DA, suppongo che ogni ricetta abbia all'incirca 9 ingredienti ciascuno, per un totale di 45.000 ingredienti, quindi 9*5.000 = 45.000.

Per l'associazione SEGUI, suppongo che ogni utente segua all'incirca 15 cuochi e che il totale sia di 1.35 x 10^10, quindi 900 MLN x 15 = 1.35 x 10^10.

Per l'associazione RISPOSTA, suppongo che ogni 5 commenti pubblicati dall'utente, siccome i commenti sono all'incirca 35.000, se avviene una risposta ogni 5 commenti, allora si avranno 7.000 risposte ai commenti pubblicati.

Per l'associazione IDEATA, siccome le ricette totali sul sito Giallo Zafferano sono all'incirca 5.000 tra (ricette caricate dal cuoco, ricette caricate dalla redazione e ricette caricate dall'utente) suppongo che vengano caricate all'incirca 2.000 ricette da parte del cuoco.

TAVOLA DELLE OPERAZIONI

Operazione	Descrizione	Tipo	Frequenza
1	L'utente X si registra al sistema	I	14 MLN/ mese
2	L'utente X inizia a seguire il Cuoco	I	1 MLN/giorno
3	L'utente X pubblica un commento	I	10 K/giorno
4	L'utente X risponde ad un commento	I	5 K/giorno
5	L'utente X aggiorna il numero di commenti pubblicati E	I	5 K/mese
6	L'utente X aggiorna il numero di ricette salvate E	I	6 K/mese
7	L'utente X aggiorna la propria lista della spesa E	I	7 K/mese
8	Il cuoco X aggiorna il numero di ricette pubblicate E	I	100 K/giorno
9	Determinare il numero dei commenti pubblicati da un utente X	B	40 K/anno
10	Determinare il numero di ricette salvate dall'utente X	B	100 K/giorno
11	Determinare le ricette possedenti un cuoco famoso	B	100 K/giorno

ANALISI DELLE RIDONDANZE

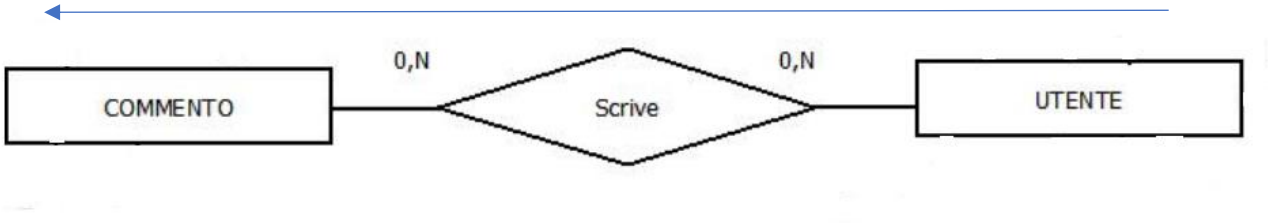
Con le seguenti analisi, ci occuperemo di individuare ed analizzare le possibili ridondanze all'interno del progetto, per quanto riguarda il rapporto costi/benefici, al fine di ottenere un vantaggio durante l'esecuzione delle operazioni.

Decidiamo di valutare i seguenti punti:

- Attributo n_commenti_pubblicati interno all'entità UTENTE, che rappresenta il totale di commenti pubblicati all'interno del sito Giallo Zafferano sulle ricette da lui seguite.
- Attributo n_ricette_salvate interno all'entità UTENTE, che rappresenta il numero totale di ricette salvate da parte di ogni utente sulla propria pagina personale.
- Attributo n_ricette_pubblicate interno all'entità CUOCO, che rappresenta il totale di ricette pubblicate sul sito Giallo Zafferano da parte di esso.

Attributo: n_commenti_pubblicati:

Spazio: L'attributo n_commenti_pubblicati sarà rappresentato da un numero intero su 4 byte.



CONCETTO	TIPO	VOLUME
Commento	E	35 K
Utente	E	900 MLN
Scrive	R	2.25 x 10^12

Avendo Utente pari a 900 MLN, ciò comporterebbe un aumento dello spazio occupato dalla relazione di : 4 byte * 900 MLN = 3.600 MB

Operazioni coinvolte : 5,9

Operazione 5: L'utente X aggiorna il numero di commenti pubblicati E

Frequenza: E' difficile fornire un valore realistico; stimeremo quindi la frequenza di 5.000 al mese.

Accessi:

- Assenza di ridondanza: Innanzitutto bisogna determinare X. Per fare questo bisogna accedere in scrittura su commento.

Infine accediamo in scrittura su Scrive.
Avendo supposto un volume di 5.000 accessi al mese, si ha un numero di accessi totali di 5.000 * 2 = 10.000 accessi in scrittura.

Le informazioni richieste sono prelevate da :

- Commento: 1 accesso
- Scrive: 1 accesso

CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Commento	E	S	1
Scrive	R	S	1

- Presenza di ridondanza: In questo caso si aggiungerebbe un'eventuale accesso in scrittura per incrementare il contatore n_commenti_pubblicati di UTENTE. Registriamo quindi un aumento degli accessi che diventano di 5.000 * 4 = 20.000 accessi totali al mese, di cui 15.000 accessi in scrittura e 5.000 accessi in lettura.

Le informazioni richieste sono prelevate da :

- Commento: 1 accesso in scrittura
- Utente: 1 accesso in lettura e 1 accesso in scrittura
- Scrive: 1 accesso in scrittura

CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Commento	E	S	1
Scrive	R	S	1
Utente	E	L	1
Utente	E	S	1

Operazione 9: Contare il numero dei commenti pubblicati di un utente X

Frequenza: Suppongo che ci siano 40.000 operazioni di questo tipo all'anno.

Accessi:

- Assenza di ridondanza: Prima di tutto, bisogna individuare l'utente X.
  Per fare ciò, accediamo in lettura su Utente individuando l'identificatore.
  Infine andiamo ad analizzare l'associazione Scrive alla ricerca dei commenti pubblicati.
  Avremo quindi 1*2.25*10^12 = 2.25*10^12
  Avendo da fare 40.000 operazioni all'anno, bisogna effettuare 40.000 * 2.25*10^12 accessi totali all'anno.

Le informazioni richieste sono prelevate da :

- Utente: 1 accesso

- Scrive : 2.25 x 10^12 accessi

CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Utente	E	L	1
Scrive	R	L	2.25*10^12

- Presenza di ridondanza: è necessario effettuare un solo accesso in lettura su UTENTE e in particolare su n_commenti_pubblicati. Cosi facendo risparmiamo 40.000*2.25*10^12 accessi, effettuando solo 40 K accessi all'anno.

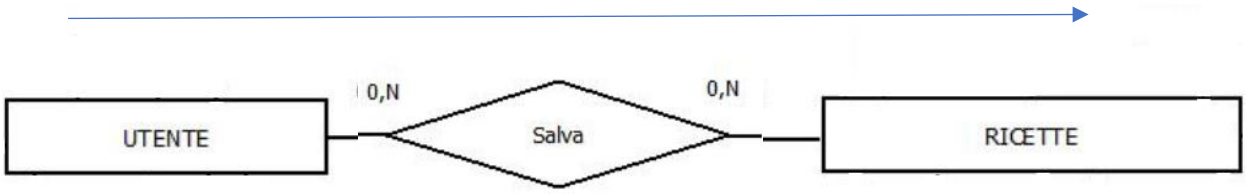
CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Utente	E	L	1

Analizzando i due scenari con e senza ridondanza, noto che si ha un risparmio negli accessi a fronte di un aumento in peso di 3.600 MB.

La differenza è notevole: poiché le prestazioni in caso di ridondanza sono migliori, conviene mantenere l'attributo n_commenti_pubblicati.

Attributo: *n_ricette_salvate:*

Spazio: L'attributo n_ricette_salvate sarà rappresentato da un numero intero su 4 byte.



CONCETTO	*TIPO*	*VOLUME*
Ricette	E	5.000
Utente	E	900 MLN
Salva	R	9x10^9

Avendo Utente pari a 900 MLN, ciò comporterebbe un aumento dello spazio occupato dalla relazione di : 4 byte * 900 MLN = 3.600 MB

Operazioni coinvolte : 6,10

Operazione 6: L'utente X aggiorna il numero di ricette salvate E

Frequenza: Suppongo che ci siano 6.000 operazioni di questo tipo ogni mese.

Accessi:

- Assenza di ridondanza: Innanzitutto bisogna determinare X. Per fare questo bisogna accedere in scrittura su ricette.
  Infine ci spostiamo su Salva per trovare il numero di ricette salvate E.
  Avendo supposto un volume di 6.000 accessi al mese, si ha un numero di accessi totali di 6.000 * 2 = 12.000 accessi in scrittura al mese.

Le informazioni richieste sono prelevate da :

- Ricette: 1 accesso in scrittura
- Salva: 1 accesso in scrittura

CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Ricette	E	S	1
Salva	R	S	1

- Presenza di ridondanza: In questo caso si aggiungerebbe un'eventuale accesso in scrittura per incrementare il contatore n_ricette_salvate di UTENTE. Registriamo quindi un aumento degli accessi che diventano di 6.000 * 4 = 24.000 accessi totali, di cui 18.000 accessi in scrittura e 6.000 accessi in lettura.

Le informazioni richieste sono prelevate da :

- Ricette: 1 accesso in scrittura
- Utente: 1 accesso in lettura e 1 accesso in scrittura
- Salva: 1 accesso in scrittura

CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Ricette	E	S	1
Utente	E	L	1
Salva	R	S	1
Utente	E	S	1

Operazione 10: Determinare il numero di ricette salvate dall'utente X

Frequenza: Suppongo che ci siano 100.000 operazioni al giorno.

Accessi:

- Assenza di ridondanza: Prima di tutto, bisogna individuare l'utente X.
  Per fare ciò, accediamo in lettura su Utente individuando l'identificatore.
  Infine andiamo ad analizzare l'associazione condivide per verificare il totale di ricette condivise da ogni singolo utente.
  Avremo quindi 1*9x10^9 = 9x10^9
  Avendo da fare 100.000 operazioni al giorno bisogna effettuare 100.000*9x10^9 operazioni al giorno.

Le informazioni richieste sono prelevate da :

- Utente: 1 accesso
- Salva: 9x10^9 accessi

CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Utente	E	L	1
Salva	R	L	9x10^9

- Con ridondanza: è necessario effettuare un solo accesso in lettura su UTENTE e in particolare su n_ricette_salvate. Cosi facendo risparmiamo 100.000*9x10^9 accessi, effettuando solo 100.000 operazioni al giorno.

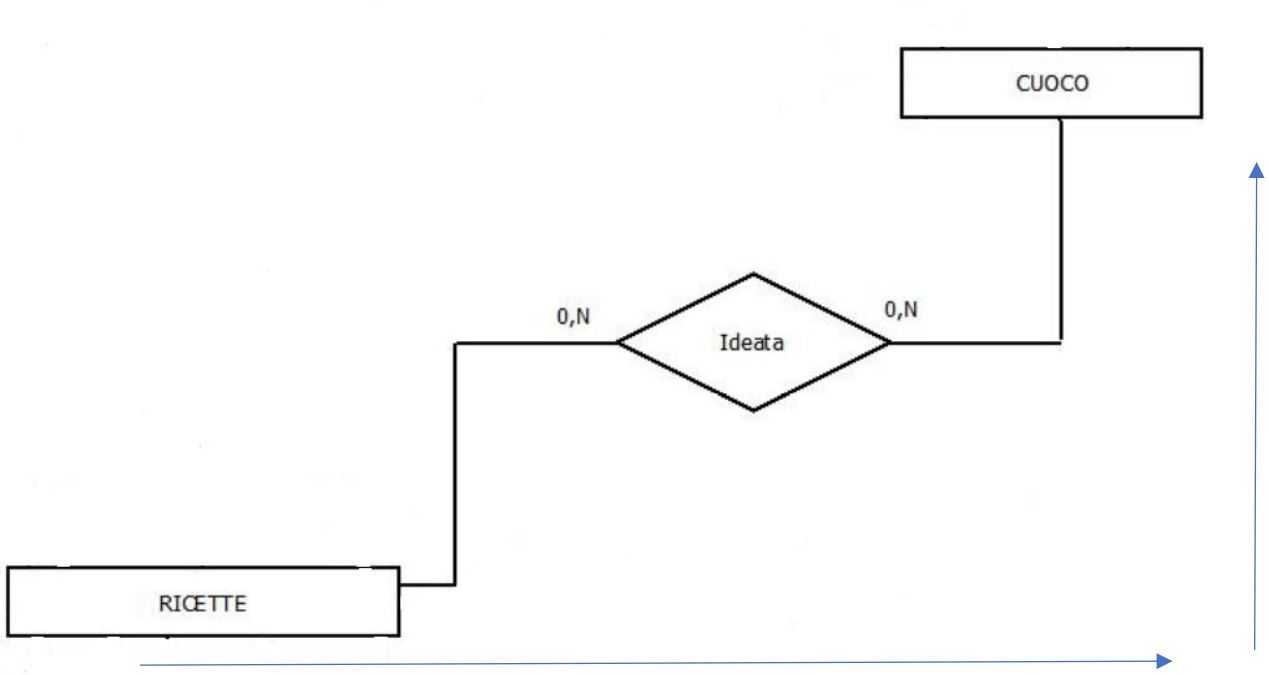
CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Utente	E	L	1

Analizzando i due scenari con e senza ridondanza, noto che si ha un risparmio negli accessi a fronte di un aumento in peso di 3.600 MB.

La differenza è notevole: poiché le prestazioni in caso di ridondanza sono migliori, conviene mantenere l'attributo n_ricette_salvate

Attributo: *n_ricette_pubblicate:*

Spazio: l'attributo n_ricette_pubblicate sarà rappresentato da un numero intero su 4 byte.



CONCETTO	*TIPO*	*VOLUME*
Ricette	E	5.000
Cuoco	E	40
Ideata	R	2.000

Avendo CUOCO con cardinalità pari a 40, comporterebbe un aumento dello spazio di 4 byte* 40 = 1.280 byte

Operazioni coinvolte : 8,11

Operazione 8: il cuoco X aggiorna il numero di ricette pubblicate E

Frequenza: Suppongo che ci siano 100.000 operazioni al giorno.

Accessi:

- Assenza di ridondanza: Innanzitutto bisogna determinare X. Per fare questo bisogna accedere in lettura su ricette per visualizzarne l'autore.
  Infine ci spostiamo su Ideata per trovare il numero di ricette pubblicate E.
  Avendo supposto un volume di 100.000 accessi al giorno, si ha un numero di accessi totali di 100.000 * 2 = 200.000 accessi in scrittura al giorno.

Le informazioni richieste sono prelevate da :

- Ricette: 1 accesso in scrittura
- Ideata: 1 accesso in scrittura

CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Ricette	E	S	1
Condivide	R	S	1

- Presenza di ridondanza: In questo caso si aggiungerebbe un'eventuale accesso in scrittura per incrementare il contatore n_ricette_pubblicate di CUOCO. Registriamo quindi un aumento degli accessi che diventano di 100.000 * 4 = 400.000 (300.000 accessi in scrittura e 100.000 accessi in lettura).

Le informazioni richieste sono prelevate da :

- Ricette: 1 accesso in scrittura
- Cuoco: 1 accesso in lettura e 1 accesso in scrittura
- Ideata: 1 accesso in scrittura

CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Ricette	E	S	1
Cuoco	E	L	1
Ideata	R	S	1
Cuoco	E	S	1

Operazione 10: Determinare le ricette possedenti un cuoco famoso

Frequenza: Suppongo che ci siano 100.000 operazioni al giorno.

Accessi:

- Assenza di ridondanza: Prima di tutto, bisogna individuare il cuoco X.
  Per fare ciò, accediamo in lettura su Cuoco individuando l'identificatore.
  Infine andiamo ad analizzare l'associazione ideata per verificare il totale di ricette pubblicate da ogni singolo cuoco.
  Avremo quindi 1*40= 40
  Avendo da fare 100.000 operazioni al giorno bisogna effettuare 100.000*40 = 4 MLN di operazioni al giorno.

Le informazioni richieste sono prelevate da :

- Cuoco: 1 accesso
- Ideata: 40 accessi

CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Cuoco	E	L	1
Ideata	R	L	40

- Con ridondanza: è necessario effettuare un solo accesso in lettura su CUOCO e in particolare su n_ricette_pubblicate. Cosi facendo risparmiamo 100.000*40 accessi, effettuando solo 100.000 operazioni al giorno.

CONCETTO	*COSTRUTTO*	*TIPO*	*ACCESSI*
Cuoco	E	L	1

Analizzando i due scenari con e senza ridondanza, noto che si ha un risparmio negli accessi a fronte di un aumento in peso di 1.280 byte.

La differenza è notevole: poiché le prestazioni in caso di ridondanza sono migliori, conviene mantenere l'attributo n_commenti_pubblicati.

Conclusione: dopo aver analizzato le eventuali ridondanze, scelgo di mantenere tutte le ridondanze individuate, ovvero: n_ricette_salvate e n_commenti_pubblicati da parte dell'utente e n_ricette_pubblicate da parte del cuoco.

SCELTA DEGLI IDENTIFICATORI PRINCIPALI

UTENTE:

Prendendo in considerazione lo schema E-R, la chiave dell'utente è email che sarà quindi rappresentata attraverso una stringa alfanumerica. Non è conveniente utilizzare un campo VARCHAR di queste dimensioni come indice primario della relazione, il che ci porta ad introdurre un identificativo autoincrementale di tipo intero (4 byte) per rappresentare la chiave di utente. Viene resa, come chiave identificativa dell'entità UTENTE, un id_utente.

RICETTE:

Prendendo in considerazione lo schema E-R, le chiavi delle ricette sono l'autore e il titolo che saranno quindi rappresentate attraverso una stringa alfanumerica. Non è conveniente utilizzare un campo VARCHAR di queste dimensioni come indice primario della relazione, il che ci porta ad introdurre un identificativo autoincrementale di tipo intero (4 byte) per rappresentare la chiave di ricette. Viene resa, come chiave identificativa dell'entità RICETTE, un id_ricette.

INGREDIENTI:

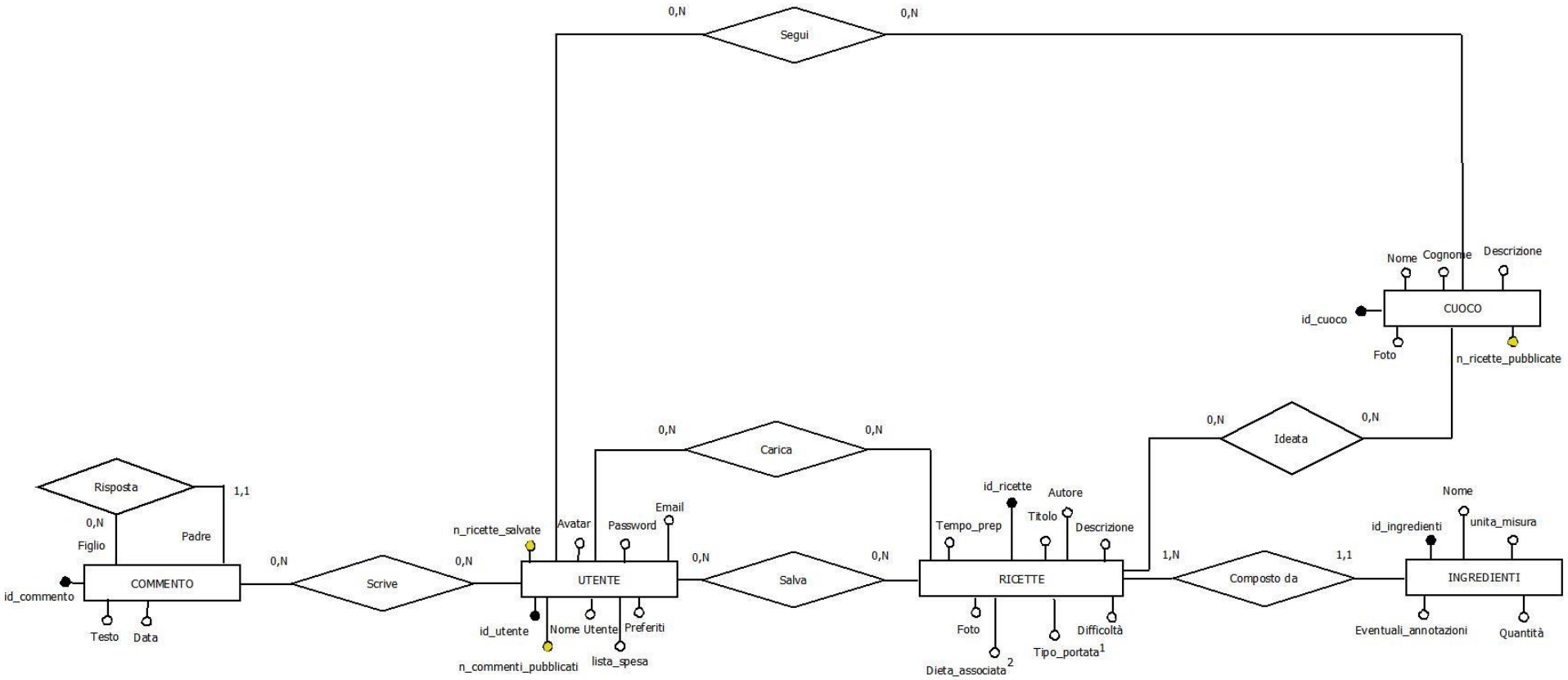
Prendendo in considerazione lo schema E-R, la chiave di ingredienti è il nome che sarà quindi rappresentata attraverso una stringa alfanumerica. Non è conveniente utilizzare un campo VARCHAR di queste dimensioni come indice primario della relazione, il che ci porta ad introdurre un identificativo autoincrementale di tipo intero (4 byte) per rappresentare la chiave di lista ingredienti. Viene resa, come chiave identificativa dell'entità INGREDIENTI, un id_ingredienti.

***CUOCO*:**

Prendendo in considerazione lo schema E-R, le chiavi della redazione sono il nome e il cognome del cuoco. Siccome potrebbero esistere delle omonimie, scelgo di introdurre un identificativo autoincrementale di tipo intero (4byte) per rappresentare la chiave del cuoco. Viene resa, come chiave identificativa dell'entità *CUOCO*, un id_cuoco.

***COMMENTO*:**

Prendendo in considerazione lo schema E-R, commento è un'entità debole, la cui esistenza è quindi legata strettamente all'utente. Introduciamo quindi, un identificativo autoincrementale di tipo intero (4 byte) per rappresentare la chiave di commento. Viene resa, come chiave identificativa dell'entità *COMMENTO*, un id_commento.



1: il tipo di portata può essere : dolce, primo, secondo, antipasto
2: la dieta associata può essere : senza glutine, veg, senza lattosio

REGOLE AZIENDALI

RV01: Gli utenti possono commentare le ricette rispondendo ad un commento già pubblicato da qualcun altro.

RV02: La redazione del sito può pubblicare commenti, ma solo in risposta ai commenti lasciati dagli utenti.

RV03: Due ricette possono considerarsi simili se hanno almeno una categoria in comune e hanno come autore lo stesso cuoco o utente.

RV04: Quando un utente si cancella dal sito vengono mantenute le ricette che ha condiviso e i suoi commenti.

RV05: Quando un utente si cancella dal sito, vengono cancellati: i dati personali, i preferiti, gli ingredienti contenuti nella lista della spesa e la lista dei cuochi eventualmente seguiti.

RD01: L'avatar associato sia al singolo utente che al singolo cuoco viene ricavato attraverso una stringa alfanumerica che rappresenta il percorso del file system dove risiede l'immagine fisicamente.

RD02: L'immagine di copertina associata al sito di ricette e alla pagina del cuoco viene ricavata attraverso una stringa alfanumerica che rappresenta il percorso del file system dove risiede l'immagine fisicamente.

RDO3: Il numero di commenti pubblicati da parte di ogni singolo utente si ricava mediante n_commenti_pubblicati di Utente.

RDO4: Il numero delle ricette salvate da parte di ogni singolo utente sulla propria pagina personale si ricava mediante n_ricette_salvate di Utente.

RDO5: Il numero delle ricette pubblicate da parte di ogni singolo cuoco si ricava mediante n_ricette_pubblicate di Cuoco.

SCHEMA RELAZIONALE

UTENTE (id_utente, nome_utente, email, password, avatar, attivo, n_ricette_salvate, n_commenti_pubblicati)

RICETTE (id_ricette, titolo, autore, descrizione, foto_copertina, difficoltà, tempo_di_preparazione, dieta_associata, tipo_portata)

INGREDIENTI (id_ingredienti, titolo, autore, nome_ingrediente, quantita, eventuali annotazioni, unita_di_misura)

CUOCO (id_cuoco, nome_cuoco, cognome_cuoco, foto, descrizione, n_ricette_pubblicate)

COMMENTO (id_commento, data. testo)

Prendiamo in considerazione le associazioni molti a molti come tabelle:

SCRIVE (id_utente, id_commento)

SALVA (id_utente, id_ricette)

CARICA (id_utente, id_ricette)

SEGUI (id_utente, id_cuoco)

RISPOSTA (id_padre, id_figlio, id_commento)

IDEATA (id_cuoco, id_ricette)

Scelta degli indici:

TABELLA	INDICE PRIMARIO
UTENTE	id_utente
RICETTE	id_ricette
INGREDIENTI	id_ingredienti
CUOCO	id_cuoco
COMMENTO	id_commento
SCRIVE	id_utente, id_commento
SALVA	id_utente, id_ricette
CARICA	id_utente, id_ricette
SEGUI	id_utente, id_cuoco
RISPOSTA	id_padre, id_figlio, id_commento
IDEATA	id_cuoco, id_ricette

DDL (Data Definition Language)

Creazione delle tabelle:

// creazione della tabella utente

```
-- Table: public."Utente"

-- DROP TABLE public."Utente";

CREATE TABLE public."Utente"
(
   id_utente integer NOT NULL,
   nome_utente character varying(70) COLLATE pg_catalog."default" NOT NULL,
   email character varying(70) COLLATE pg_catalog."default" NOT NULL,
   password character varying(256) COLLATE pg_catalog."default" NOT NULL,
   avatar character varying(256) COLLATE pg_catalog."default",
   attivo boolean NOT NULL DEFAULT false,
   n_ricette_salvate integer NOT NULL DEFAULT 0,
   n_commenti_pubblicati integer NOT NULL DEFAULT 0,

   CONSTRAINT "Utente_pkey" PRIMARY KEY (id_utente),
   CONSTRAINT email_key UNIQUE (email)
,
   CONSTRAINT user_is_attivo CHECK (
CASE
   WHEN attivo = true THEN n_ricette_salvate = '-1'::integer
   ELSE NULL::boolean
END)

)
WITH (
   OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public."Utente"
   OWNER to dbstefanopeddoni;
```

// creazione della tabella ricette

```
-- Table: public.ricette

-- DROP TABLE public.ricette;

CREATE TABLE public.ricette
(
```

```
    id_ricette integer NOT NULL,
    titolo character varying(256) COLLATE pg_catalog."default" NOT NULL,
    autore character varying(256) COLLATE pg_catalog."default" NOT NULL,
    descrizione text COLLATE pg_catalog."default" NOT NULL,
    foto_copertina character varying(256) COLLATE pg_catalog."default" NOT NULL,
    difficolta character varying(14) COLLATE pg_catalog."default" NOT NULL,
    tempo_di_preparazione integer NOT NULL,
    dieta_associata character varying(14) COLLATE pg_catalog."default" NOT NULL,
    tipo_portata character varying(14) COLLATE pg_catalog."default" NOT NULL,

    CONSTRAINT ricette_pkey PRIMARY KEY (id_ricette),
    CONSTRAINT autore_key UNIQUE (autore),
    CONSTRAINT titolo_key UNIQUE (titolo),

    CONSTRAINT difficolta CHECK (difficolta::text = 'facile' ::text OR difficolta::text = 'media'
    ::text OR difficolta::text = 'difficile'::text),

    CONSTRAINT tempo_di_preparazione CHECK (tempo_di_preparazione > 0),

    CONSTRAINT dieta_associata CHECK (dieta_associata::text = 'light' ::text OR
    dieta_associata::text = 'senza glutine' ::text OR dieta_associata::text = 'senza lattosio' ::text OR
    dieta_associata::text = 'vegetariano' ::text OR dieta_associata::text = 'basso nichel' ::text),

    CONSTRAINT tipo_portata CHECK (tipo_portata::text = 'primo' ::text OR tipo_portata::text
    = 'secondo' ::text OR tipo_portata::text = 'antipasto' ::text OR tipo_portata::text = 'contorno'
    ::text OR tipo_portata::text = 'dolci' ::text OR tipo_portata::text = 'lievitati' ::text OR
    tipo_portata::text = 'piatti unici' ::text)


)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.ricette
    OWNER to dbstefanopeddoni;
```

// creazione della tabella ingredienti

```
-- Table: public.ingredienti

-- DROP TABLE public.ingredienti;

CREATE TABLE public.ingredienti
(
```

```
    id_ingredienti integer NOT NULL,
    titolo character varying(256) COLLATE pg_catalog."default" NOT NULL,
    autore character varying(256) COLLATE pg_catalog."default" NOT NULL,
    nome character varying(256) COLLATE pg_catalog."default" NOT NULL,
    quantita integer NOT NULL,
    eventuali_annotazioni text COLLATE pg_catalog."default",
    unita_di_misura character varying(256) COLLATE pg_catalog."default",

    CONSTRAINT ingredienti_pkey PRIMARY KEY (id_ingredienti),
    CONSTRAINT autore FOREIGN KEY (autore)
        REFERENCES public.ricette (autore) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE CASCADE,
    CONSTRAINT titolo FOREIGN KEY (titolo)
        REFERENCES public.ricette (titolo) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE CASCADE

)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.ingredienti
    OWNER to dbstefanopeddoni;
```

// creazione della tabella cuoco

```
-- Table: public.cuoco

-- DROP TABLE public.cuoco;

CREATE TABLE public.cuoco
(
    id_cuoco integer NOT NULL DEFAULT nextval('cuoco_id_cuoco_seq'::regclass),
    nome_cuoco character varying(256) COLLATE pg_catalog."default" NOT NULL,
    cognome_cuoco character varying(256) COLLATE pg_catalog."default" NOT NULL,
    foto character varying(256) COLLATE pg_catalog."default" NOT NULL,
    descrizione text COLLATE pg_catalog."default" NOT NULL,
    n_ricette_pubblicate integer NOT NULL,

    CONSTRAINT cuoco_pkey PRIMARY KEY (id_cuoco),
    CONSTRAINT n_ricette_pubblicate CHECK (n_ricette_pubblicate >= 0)
)
```

```
WITH (
   OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.cuoco
   OWNER to dbstefanopeddoni;
```

// creazione della tabella commento

```
-- Table: public.commento

-- DROP TABLE public.commento;

CREATE TABLE public.commento
(
   id_commento integer NOT NULL DEFAULT nextval('commento_id_commento_seq'::regclass),
   data timestamp without time zone DEFAULT now( ),
   testo text COLLATE pg_catalog."default" NOT NULL,
   CONSTRAINT commento_pkey PRIMARY KEY (id_commento)
)


WITH (
   OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.commento
   OWNER to dbstefanopeddoni;
```

// creazione della tabella scrive

```
-- Table: public.scrive

-- DROP TABLE public.scrive;

CREATE TABLE public.scrive
(
   id_utente integer NOT NULL,
   id_commento integer NOT NULL,

   CONSTRAINT id_commento FOREIGN KEY (id_commento)
      REFERENCES public.commento (id_commento) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE NO ACTION,
```

```
    CONSTRAINT id_utente FOREIGN KEY (id_utente)
        REFERENCES public."Utente" (id_utente) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;


ALTER TABLE public.scrive
    OWNER to dbstefanopeddoni;
```

// creazione della tabella carica

```
-- Table: public.carica

-- DROP TABLE public.carica;

CREATE TABLE public.carica
(
    id_utente integer NOT NULL,
    id_ingredienti integer NOT NULL,

    CONSTRAINT id_ingredienti FOREIGN KEY (id_ingredienti)
        REFERENCES public.ingredienti (id_ingredienti) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE CASCADE,
    CONSTRAINT id_utente FOREIGN KEY (id_utente)
        REFERENCES public."Utente" (id_utente) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;


ALTER TABLE public.carica
    OWNER to dbstefanopeddoni;
```

// creazione della tabella ideata

```
-- Table: public.ideata
```

```
-- DROP TABLE public.ideata;

CREATE TABLE public.ideata
(
   id_cuoco integer NOT NULL,
   id_ingredienti integer NOT NULL,

   CONSTRAINT id_cuoco FOREIGN KEY (id_cuoco)
      REFERENCES public.cuoco (id_cuoco) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE NO ACTION,
   CONSTRAINT id_ingredienti FOREIGN KEY (id_ingredienti)
      REFERENCES public.ricette (id_ingredienti) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE CASCADE
)
WITH (
   OIDS = FALSE
)
TABLESPACE pg_default;


ALTER TABLE public.ideata
   OWNER to dbstefanopeddoni;
```

// creazione della tabella segui

```
-- Table: public.segui

-- DROP TABLE public.segui;

CREATE TABLE public.segui
(
   id_utente integer NOT NULL,
   id_cuoco integer NOT NULL,

   CONSTRAINT id_cuoco FOREIGN KEY (id_cuoco)
      REFERENCES public.cuoco (id_cuoco) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE NO ACTION,
   CONSTRAINT id_utente FOREIGN KEY (id_utente)
      REFERENCES public."Utente" (id_utente) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE NO ACTION
)
WITH (
   OIDS = FALSE
```

```
)
TABLESPACE pg_default;


ALTER TABLE public.segui
   OWNER to dbstefanopeddoni;
```

// creazione della tabella salva

```
-- Table: public.salva

-- DROP TABLE public.salva;

CREATE TABLE public.salva
(
   id_utente integer NOT NULL,
   id_ricette integer NOT NULL,

   CONSTRAINT id_ricette FOREIGN KEY (id_ricette)
      REFERENCES public.ricette (id_ricette) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE CASCADE,
   CONSTRAINT id_utente FOREIGN KEY (id_utente)
      REFERENCES public."Utente" (id_utente) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE CASCADE
)
WITH (
   OIDS = FALSE
)
TABLESPACE pg_default;


ALTER TABLE public.salva
   OWNER to dbstefanopeddoni;
```

// creazione della tabella risposta

```
-- Table: public.risposta

-- DROP TABLE public.risposta;
```

```
CREATE TABLE public.risposta
(
	id_padre integer NOT NULL,
	id_figlio integer NOT NULL,

  CONSTRAINT id_padre FOREIGN KEY (id_padre)
	REFERENCES public.commenti (id_commento) MATCH SIMPLE
	ON UPDATE NO ACTION
	ON DELETE NO ACTION,
   CONSTRAINT id_figlio FOREIGN KEY (id_figlio)
	REFERENCES public.commenti (id_commento) MATCH SIMPLE
	ON UPDATE NO ACTION
	ON DELETE NO ACTION
)
WITH (
   OIDS = FALSE
)
TABLESPACE pg_default;


ALTER TABLE public.risposta
   OWNER to dbstefanopeddoni;
```

DML (Data Manipulation Language)

Inserimento dati nelle tabelle:

//inserisco i dati all'interno della tabella utente

INSERT INTO "Utente" (id_utente, nome_utente, email, password, avatar, attivo, n_ricette_salvate, n_commenti_pubblicati) VALUES (1010, 'paolorox', 'Paolo.Rossi@gmail.com', 'paolino999', '/image/profile/Paolo/Rossi./paolorox/image.jpg', true, 4, 9)

INSERT INTO "Utente" (id_utente, nome_utente, email, password, avatar, attivo, n_ricette_salvate, n_commenti_pubblicati) VALUES (1255, 'lucaverdi', 'Luca.Verdi@gmail.com', 'luk934 ', '/image/profile/Luca/Verdi./lucaverdi/image.jpg', true, 7, 9)

INSERT INTO "Utente" (id_utente, nome_utente, email, password, avatar, attivo, n_ricette_salvate, n_commenti_pubblicati) VALUES (1477, 'annarossi', 'Anna.Rossi@gmail.com', 'annetta22 ', '/image/profile/Anna/Rossi./annarossi/image.jpg', true, 10, 9)

INSERT INTO "Utente" (id_utente, nome_utente, email, password, avatar, attivo, n_ricette_salvate, n_commenti_pubblicati) VALUES (5669, 'andrealori', 'Andrea.Lori@gmail.com', 'andrellox ', '/image/profile/Andrea/Lori./andrealori/image.jpg', true, 11, 9)

INSERT INTO "Utente" (id_utente, nome_utente, email, password, avatar, attivo, n_ricette_salvate, n_commenti_pubblicati) VALUES (7888, 'alexmeli', 'Ale.Meli@gmail.com', 'alemedd ', '/image/profile/Ale/Meli./alexmeli/image.jpg', true, 15, 9)

//inserisco i dati all'interno della tabella ricette

INSERT INTO ricette (id_ricette, titolo, autore, descrizione, foto_copertina, difficoltà, tempo_di_preparazione, dieta_associata, tipo_portata, costo) VALUES (4456, 'Tartufi rossi', 'Giallo Zafferano' 'Quest'anno prendete per la gola i vostri ospiti con dei graziosi dolcetti vestiti a festa, i tartufi rossi!', '/image/profile/Tartufi/Rossi./image.jpg', 'facile', 40, 'senza lattosio', 'dolce', 'medio');

INSERT INTO ricette (id_ricette, titolo, autore, descrizione, foto_copertina, difficoltà, tempo_di_preparazione, dieta_associata, tipo_portata, costo) VALUES (1226, 'Cannelloni', 'Silvio Sau', 'Succulente lasagne con un irresistibile sugo fresco', '/image/profile/Cannelloni./image.jpg', 'facile', 90, 'senza lattosio', 'primo', 'medio');

INSERT INTO ricette (id_ricette, titolo, autore, descrizione, foto_copertina, difficoltà, tempo_di_preparazione, dieta_associata, tipo_portata, costo) VALUES (5558, 'Tiramisu', 'Roberto Conti' 'Ottimo dolce con della freschissima crema al mascarpone!', '/image/profile/Tiramisu./image.jpg', 'facile', 40, 'vegetariano', 'dolce', 'basso');

// inserisco i dati all'interno della tabella cuoco

INSERT INTO cuoco (id_cuoco, nome_cuoco, cognome_cuoco, foto, descrizione, n_ricette_pubblicate) VALUES (1580, 'Roberto', 'Conti', '/image/profile/Roberto/Conti./image.jpg', 'Famoso chef', NULL);

INSERT INTO cuoco (id_cuoco, nome_cuoco, cognome_cuoco, foto, descrizione, n_ricette_pubblicate)

```
VALUES (2250, 'Ugo', 'Alciati', '/image/profile/Ugo/Alciati./image.jpg', 'Famoso chef', NULL);

INSERT INTO cuoco (id_cuoco, nome_cuoco, cognome_cuoco, foto, descrizione, n_ricette_pubblicate)
VALUES (3588, 'Fabio', 'Abbattista', '/image/profile/Fabio/Abbattista./image.jpg', 'Famoso chef',
NULL);
```

Eliminazione delle tabelle:

```
// eliminazione della tabella utente
TRUNCATE TABLE public."Utente";
DROP TABLE public."Utente";

// eliminazione della tabella ricette
TRUNCATE TABLE public.ricette;
DROP TABLE public.ricette;

// eliminazione della tabella cuoco
TRUNCATE TABLE public.cuoco;
DROP TABLE public.cuoco;
```

Modifica delle tabelle:

```
//inserimento di una nuova colonna nella tabella utente
-- Column: public."Utente".citta

-- ALTER TABLE public."Utente" DROP COLUMN citta;

ALTER TABLE public."Utente"
        ADD COLUMN citta character varying(256) COLLATE pg_catalog."default" NOT NULL;

//tabella utente con l'inserimento della nuova colonna citta
-- Table: public."Utente"

-- DROP TABLE public."Utente";

CREATE TABLE public."Utente"
(
   id_utente integer NOT NULL,
   nome_utente character varying(70) COLLATE pg_catalog."default" NOT NULL,
   email character varying(70) COLLATE pg_catalog."default" NOT NULL,
   password character varying(256) COLLATE pg_catalog."default" NOT NULL,
   avatar character varying(256) COLLATE pg_catalog."default",
   attivo boolean NOT NULL DEFAULT false,
   n_ricette_salvate integer NOT NULL DEFAULT 0,
   n_commenti_pubblicati integer NOT NULL DEFAULT 0,
   citta character varying(256) COLLATE pg_catalog."default" NOT NULL,

   CONSTRAINT "Utente_pkey" PRIMARY KEY (id_utente),
```

```
    CONSTRAINT nome_utente_key UNIQUE (email),
    CONSTRAINT user_is_attivo CHECK (
CASE
    WHEN attivo = true THEN n_ricette_salvate = '-1'::integer
    ELSE NULL::boolean
END)

)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
ALTER TABLE public."Utente"
    OWNER to dbstefanopeddoni;


//cancellazione della colonna appena inserita nella tabella utente
-- Column: public."Utente".citta

-- ALTER TABLE public."Utente" DROP COLUMN citta;

ALTER TABLE public."Utente"
        DROP COLUMN citta;


//inserimento di una nuova colonna nella tabella ricette
-- Column: public.ricette.costo

-- ALTER TABLE public.ricette DROP COLUMN costo;

ALTER TABLE public.ricette
        ADD COLUMN costo character varying(35) COLLATE pg_catalog."default" NOT NULL;

//tabella ricette con l'inserimento della nuova riga costo

-- Table: public.ricette

-- DROP TABLE public.ricette;

CREATE TABLE public.ricette
(
    id_ricette integer NOT NULL,
    titolo character varying(256) COLLATE pg_catalog."default" NOT NULL,
    autore character varying(256) COLLATE pg_catalog."default" NOT NULL,
    descrizione text COLLATE pg_catalog."default" NOT NULL,
    foto_copertina character varying(256) COLLATE pg_catalog."default" NOT NULL,
    difficolta character varying(14) COLLATE pg_catalog."default" NOT NULL,
    tempo_di_preparazione integer NOT NULL,
    dieta_associata character varying(14) COLLATE pg_catalog."default" NOT NULL,
    tipo_portata character varying(14) COLLATE pg_catalog."default" NOT NULL,
```

```
    costo character varying(35) COLLATE pg_catalog."default" NOT NULL,

    CONSTRAINT ricette_pkey PRIMARY KEY (id_ricette),
    CONSTRAINT ricette_autore_key UNIQUE (autore),
    CONSTRAINT ricette_titolo_key UNIQUE (titolo),

    CONSTRAINT difficolta CHECK (difficolta::text = 'facile' ::text OR difficolta::text = 'media'
    ::text OR difficolta::text = 'difficile'::text),

    CONSTRAINT tempo_di_preparazione CHECK (tempo_di_preparazione > 0),

    CONSTRAINT dieta_associata CHECK (dieta_associata::text = 'light' ::text OR
    dieta_associata::text = 'senza glutine' ::text OR dieta_associata::text = 'senza lattosio' ::text OR
    dieta_associata::text = 'vegetariano' ::text OR dieta_associata::text = 'basso nichel' ::text),

    CONSTRAINT tipo_portata CHECK (tipo_portata::text = 'primo' ::text OR tipo_portata::text
    = 'secondo' ::text OR tipo_portata::text = 'antipasto' ::text OR tipo_portata::text = 'contorno'
    ::text OR tipo_portata::text = 'dolci' ::text OR tipo_portata::text = 'lievitati' ::text OR
    tipo_portata::text = 'piatti unici' ::text),

    CONSTRAINT costo CHECK (costo::text = 'molto basso' ::text OR costo::text = 'basso' ::text
    OR costo::text = 'medio' ::text OR costo::text = 'alto' ::text OR costo::text = 'elevato ::text OR
    costo::text = 'molto elevato' ::text)


)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.ricette
    OWNER to dbstefanopeddoni;
```