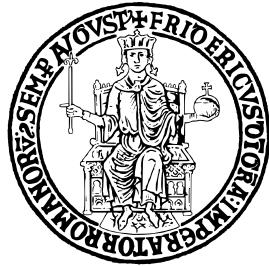


UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II



SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

CORSO DI LAUREA TRIENNALE IN INFORMATICA

UNINAFOODLAB DATABASE

Anno Accademico 2024–2025

# Indice

<b>1 Descrizione del progetto</b>	<b>3</b>
1.1 Descrizione sintetica del problema . . . . .	3
<b>2 Progettazione Concettuale</b>	<b>4</b>
2.1 Introduzione . . . . .	4
2.2 Diagrammi UML e ER . . . . .	4
2.3 Ristrutturazione del modello concettuale . . . . .	5
2.4 Identificazione delle chiavi primarie . . . . .	5
2.5 Identificazione delle chiavi esterne . . . . .	5
2.6 Rimozione degli attributi multipli . . . . .	5
2.7 Rimozione delle gerarchie di specializzazione . . . . .	6
2.8 Analisi delle ridondanze . . . . .	6
2.9 Class Diagram ristrutturato . . . . .	6
2.10 Dizionario delle classi . . . . .	6
2.11 Dizionario delle associazioni . . . . .	7
2.12 Dizionario dei vincoli . . . . .	7
<b>3 Progettazione Logica</b>	<b>8</b>
3.1 Schema Logico . . . . .	8
<b>4 Progettazione Fisica</b>	<b>9</b>
4.1 Definizione tabelle . . . . .	9
4.1.1 Definizione della tabella TOPIC . . . . .	9
4.1.2 Definizione della tabella CHEF . . . . .	10
4.1.3 Definizione della tabella UTENTE . . . . .	10
4.1.4 Definizione della tabella CORSO . . . . .	11
4.1.5 Definizione della tabella SESSIONEINPRESENZA . . . . .	11
4.1.6 Definizione della tabella SESSIONEONLINE . . . . .	12
4.1.7 Definizione della tabella ADESIONE . . . . .	12
4.1.8 Definizione della tabella PREPARA . . . . .	13
4.1.9 Definizione della tabella RICETTA . . . . .	13
4.1.10 Definizione della tabella INGREDIENTE . . . . .	13
4.1.11 Definizione della tabella COMPOSIZIONE . . . . .	14
4.1.12 Definizione della tabella NOTIFICA . . . . .	14
4.1.13 Definizione della tabella RICEVE . . . . .	15
4.1.14 Definizione della tabella INSEGNA . . . . .	15
4.1.15 Definizione della tabella ISCRIZIONE . . . . .	16
4.2 Triggers, Procedures, Functions e Vincoli . . . . .	16
4.2.1 Funzione per calcolare il numero di adesioni . . . . .	16
4.2.2 Trigger per controllare che l'anno del corso sia valido . . . . .	17

4.2.3	Trigger per controllare che l'aula della sessione in presenza non sia occupata . . . . .	18
4.2.4	Trigger per controllare che l'utente che aderisce ad una sessione in presenza sia iscritto al corso . . . . .	19
4.2.5	Trigger per controllare che lo chef insegni il topic del corso che tiene . . . . .	20
4.2.6	Trigger per controllare che le sessioni siano programmate in date successive alla data di inizio del corso . . . . .	21
4.2.7	Trigger per controllare che la data di inizio del corso sia valida . . . . .	21
4.2.8	Trigger per controllare che la data delle sessioni sia valida . . . . .	22
4.2.9	Trigger per controllare che il corso non abbia raggiunto il limite di 50 iscrizioni .	22
4.2.10	Trigger per controllare che il numero sessione sia coerente con il numero della sessione precedente . . . . .	23
4.2.11	Trigger per controllare che non ci sia una sovrapposizione di date tra le sessioni in presenza per lo stesso chef . . . . .	25
4.2.12	Trigger per controllare che gli username degli chef e degli utenti siano univoci .	26
<b>5</b>	<b>Popolamento del database</b>	<b>28</b>
5.1	Inserimento dei dati . . . . .	28
5.1.1	Query di inserimento degli chef . . . . .	28
5.1.2	Query di inserimento degli utenti . . . . .	29
5.1.3	Query di inserimento dei corsi . . . . .	30
5.1.4	Query di inserimento delle sessioni in presenza . . . . .	31
5.1.5	Query di inserimento delle sessioni online . . . . .	32
5.1.6	Query di inserimento delle adesioni . . . . .	32
5.1.7	Query di inserimento delle ricette . . . . .	32
5.1.8	Query di inserimento degli ingredienti . . . . .	37
5.1.9	Query di inserimento della composizione delle ricette . . . . .	42
5.1.10	Query di inserimento dei topic . . . . .	43
5.1.11	Query di inserimento della tabella insegnna . . . . .	44
5.1.12	Query di inserimento della tabella prepara . . . . .	44
5.1.13	Query di inserimento delle iscrizioni . . . . .	45

# Capitolo 1

## Descrizione del progetto

### 1.1 Descrizione sintetica del problema

Si vuole realizzare una base di dati relazionale per la gestione di corsi di cucina tematici. Gli chef possono registrare corsi su specifici argomenti, specificando una data di inizio e una frequenza delle sessioni. Ogni corso è articolato in più sessioni, che possono essere di due tipi: online, oppure in presenza, in cui gli utenti svolgono attività pratiche. Gli utenti possono iscriversi a più corsi e, nel caso delle sessioni pratiche, devono fornire una adesione esplicita per confermare la loro partecipazione. Ogni sessione pratica prevede la preparazione di una o più ricette, ciascuna delle quali richiede una specifica lista di ingredienti. Le adesioni vengono utilizzate per pianificare correttamente la quantità di ingredienti necessari, evitando così sprechi alimentari. Il sistema prevede la possibilità di notificare gli utenti iscritti ad un corso in caso di variazioni di programma.

# Capitolo 2

## Progettazione Concettuale

### 2.1 Introduzione

Una volta definito e analizzato il problema possiamo procedere con la progettazione concettuale della base di dati. In questa fase, l'obiettivo è quello di creare un modello concettuale che rappresenti le entità, le relazioni e le caratteristiche del dominio. Tale schema viene rappresentato mediante un diagramma delle classi UML e un diagramma Entità-Relazioni (ER).

### 2.2 Diagrammi UML e ER

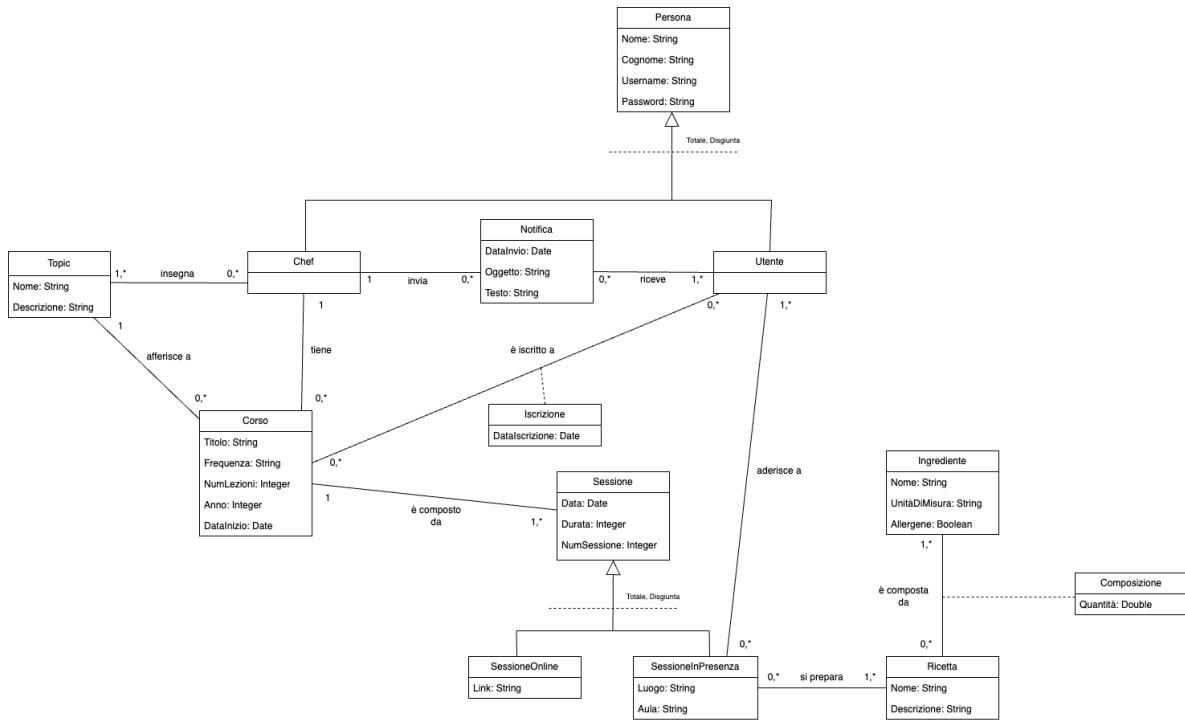


Figura 2.1: Diagramma delle classi UML



Figura 2.2: Diagramma Entità-Relazioni

## 2.3 Ristrutturazione del modello concettuale

Un'operazione fondamentale per il passaggio dal modello concettuale a quello logico è la ristrutturazione. Questo processo implica la trasformazione delle entità e delle relazioni del modello concettuale, al fine di renderne più agevole l'implementazione in un database relazionale. Durante questa fase, si identificano le chiavi primarie e le chiavi esterne, si rimuovono gli attributi multipli e le gerarchie di specializzazione, si normalizzano le tabelle per ridurre la ridondanza dei dati.

## 2.4 Identificazione delle chiavi primarie

Le chiavi primarie sono gli attributi che identificano in modo univoco ogni record in una tabella. Nella nostra ristrutturazione abbiamo fatto uso di chiavi surrogate, ossia chiavi artificiali create per garantire l'unicità dei record nel caso in cui non esistano chiavi naturali adatte. Queste chiavi non hanno un significato intrinseco, ma sono utilizzate per facilitare le operazioni sulle tabelle.

## 2.5 Identificazione delle chiavi esterne

Le chiavi esterne sono gli attributi che collegano le tabelle tra loro, rappresentando le relazioni tra le entità. Il loro scopo è quello di assicurare che i dati siano coerenti e che le relazioni siano mantenute.

## 2.6 Rimozione degli attributi multipli

Non sono presenti attributi multipli nel modello concettuale.

## 2.7 Rimozione delle gerarchie di specializzazione

Sono presenti due gerarchie di specializzazione:

- 'Persona' si specializza in "Chef" e "Utente"
  - 'Sessione' si specializza in "Sessione Online" e "Sessione in Presenza"

## 2.8 Analisi delle ridondanze

Non sono presenti ridondanze significative nel modello concettuale.

## 2.9 Class Diagram ristrutturato

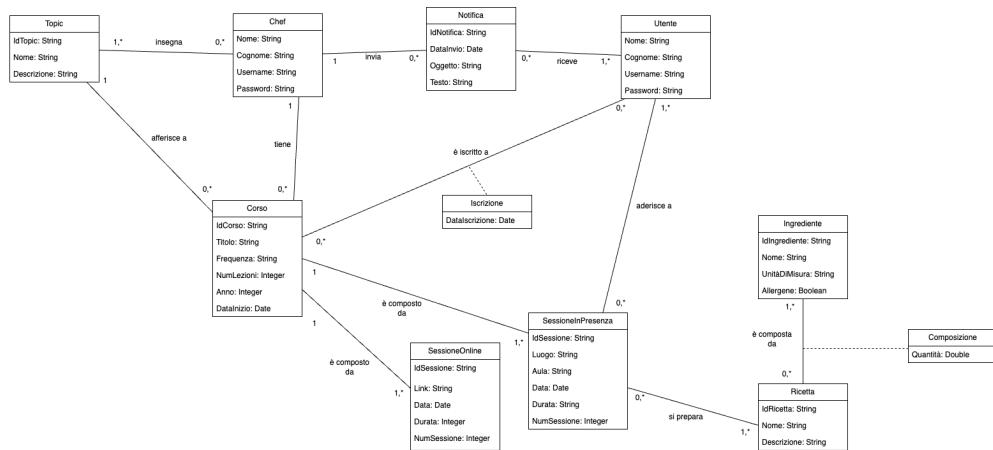


Figura 2.3: Diagramma UML ristrutturato

## 2.10 Dizionario delle classi

**2.11 Dizionario delle associazioni**

**2.12 Dizionario dei vincoli**

# Capitolo 3

## Progettazione Logica

### 3.1 Schema Logico

**TOPIC**(IdTopic, Nome, Descrizione)

**INSEGNA**(IdTopic, UsernameChef)

**CORSO**(IdCorso, Titolo, Frequenza, NumLezioni, DataInizio, IdTopic, UsernameChef)

**CHEF**(Username, Nome, Cognome, Password)

**UTENTE**(Username, Nome, Cognome, Password)

**NOTIFICA**(IdNotifica, DataInvio, Oggetto, Testo, UsernameChef)

**RICEVE**(IdNotifica, UsernameUtente)

**ISCRIZIONE**(UsernameUtente, IdCorso, DataIscrizione)

**SESSIONE ONLINE**(IdSessione, Link, Data, Durata, NumSessione, IdCorso)

**SESSIONE IN PRESENZA**(IdSessione, Luogo, Aula, Data, Durata, NumSessione, IdCorso)

**ADESIONE**(UsernameUtente, IdSessionePresenza)

**RICETTA**(IdRicetta, Nome, Descrizione)

**PREPARA**(IdSessionePresenza, IdRicetta)

**INGREDIENTE**(IdIngrediente, Nome, UnitàDiMisura, Allergene)

**COMPOSIZIONE**(IdRicetta, IdIngrediente, Quantità)

# Capitolo 4

## Progettazione Fisica

La base di dati è stata implementata utilizzando PostgreSQL, un sistema di gestione di basi di dati relazionali open source. La progettazione fisica della base di dati è stata effettuata seguendo le linee guida del modello relazionale, definendo le tabelle, le chiavi primarie e le chiavi esterne per garantire l'integrità referenziale tra le entità. Qui di seguito sono riportate le definizioni delle tabelle utilizzate nel database ed eventuali triggers, procedures, functions o vincoli implementati.

### 4.1 Definizione tabelle

Seguono le definizioni delle tabelle estratte dallo script di creazione del database.

#### 4.1.1 Definizione della tabella TOPIC

```
1 CREATE TABLE Topic (
2     idTopic SERIAL PRIMARY KEY,
3     Nome VARCHAR(1000) NOT NULL UNIQUE,
4     Descrizione VARCHAR(1000),
5
6     CONSTRAINT CK_Topic_Nome CHECK (Nome IN ('Cucina Italiana', 'Pizza
7         Napoletana', 'Cucina Cinese',
8         'Cucina Messicana', 'Cucina Giapponese', 'Cucina Indiana', 'Cucina
9             Francese',
10            'Cucina Spagnola', 'Cucina Americana', 'Cucina Mediterranea',
11            'Pasticceria', 'Panificazione')) -- controllo che il nome del topic
12            sia un valore valido
13 );
```

Listing 4.1: Creazione tabella Topic

La tabella **Topic** rappresenta le categorie tematiche dei corsi di cucina. Ogni topic ha un identificatore univoco generato automaticamente e un nome che deve essere scelto tra i valori predefiniti specificati nel constraint CHECK.

#### 4.1.2 Definizione della tabella CHEF

```
1 CREATE TABLE Chef (
2     Nome VARCHAR(50) NOT NULL ,
3     Cognome VARCHAR(50) NOT NULL ,
4     Username VARCHAR(100) PRIMARY KEY ,
5     Password VARCHAR(100) NOT NULL
6 );
```

Listing 4.2: Creazione tabella Chef

La tabella **Chef** rappresenta gli chef che possono tenere corsi di cucina. Ogni chef ha un identificatore univoco (Username) e le informazioni personali come nome, cognome e password.

#### 4.1.3 Definizione della tabella UTENTE

```
1 CREATE TABLE Utente (
2     Nome VARCHAR(50) NOT NULL ,
3     Cognome VARCHAR(50) NOT NULL ,
4     Username VARCHAR(100) PRIMARY KEY ,
5     Password VARCHAR(100) NOT NULL
6 )
```

Listing 4.3: Creazione tabella Utente

La tabella **Utente** rappresenta gli utenti che possono iscriversi ai corsi di cucina. Ogni utente ha un identificatore univoco (Username) e le informazioni personali come nome, cognome e password.

#### 4.1.4 Definizione della tabella CORSO

```

1 CREATE TABLE Corso (
2     idCorso SERIAL PRIMARY KEY,
3     Titolo VARCHAR(50) NOT NULL,
4     Frequenza VARCHAR(50) DEFAULT 'Settimanale',
5     NumLezioni Integer NOT NULL,
6     Anno Integer NOT NULL,
7     DataInizio Date NOT NULL,
8     IdTopic INTEGER NOT NULL,
9     UsernameChef VARCHAR(100) NOT NULL,
10
11    CONSTRAINT FK_Corso_Topic FOREIGN KEY (IdTopic) REFERENCES
12        Topic(idTopic), -- foreign key verso Topic
13    CONSTRAINT FK_Corso_Chef FOREIGN KEY (UsernameChef) REFERENCES
14        Chef.Username ON DELETE CASCADE, -- foreign key verso Chef, su
15        eliminazione di un chef elimina tutti i suoi corsi
16    CONSTRAINT CK_Frequenza CHECK (Frequenza IN ('Settimanale', 'Ogni
17        giorno', 'Ogni due giorni', 'Ogni tre giorni')), -- controllo che la
18        frequenza del corso sia un valore valido
19    CONSTRAINT CK_NumLezioni CHECK (NumLezioni BETWEEN 1 AND 100), --
20        controllo che il numero di lezioni sia compreso tra 1 e 100
21    CONSTRAINT CK_Anno CHECK (Anno = EXTRACT(YEAR FROM DataInizio)) --
22        controllo che l'anno sia uguale all'anno di inizio
23 )

```

Listing 4.4: Creazione tabella Corso

La tabella **Corso** rappresenta i corsi di cucina. Ogni corso ha un identificatore univoco (IdCorso), un titolo, una frequenza, il numero di lezioni, una data di inizio e un riferimento al topic e allo chef che lo insegna.

#### 4.1.5 Definizione della tabella SESSIONEINPRESENZA

```

1 CREATE TABLE SessioneInPresenza (
2     IdSessione SERIAL PRIMARY KEY,
3     Luogo VARCHAR(50) NOT NULL,
4     Aula VARCHAR(50) NOT NULL,
5     Data TIMESTAMP NOT NULL,
6     Durata Integer NOT NULL,
7     NumSessione Integer NOT NULL,
8     IdCorso INTEGER NOT NULL,
9
10    CONSTRAINT FK_SessioneInPresenza_Corso FOREIGN KEY (IdCorso) REFERENCES
11        Corso(idCorso) ON DELETE CASCADE, -- foreign key verso Corso, su
12        eliminazione di un corso elimina tutte le sue sessioni
13    CONSTRAINT CK_Luogo CHECK (Luogo IN ('MonteSantangelo', 'ViaClaudio',
14        'PiazzaleTecchio', 'PortaDiMassa', 'ViaMarina')), -- controllo che il
15        luogo sia un valore valido
16    CONSTRAINT CK_Durata CHECK (Durata >= 60 AND Durata <= 180) -- controllo
17        che la durata sia compresa tra 60 e 180 minuti
18 )

```

Listing 4.5: Creazione tabella Sessione in Presenza

La tabella **SessioneInPresenza** rappresenta le sessioni in presenza dei corsi di cucina. Ogni sessione ha un identificatore univoco (IdSessione), un luogo, un'aula, una data, una durata, un numero di sessione e un riferimento al corso a cui appartiene.

#### 4.1.6 Definizione della tabella SESSIONEONLINE

```

1 CREATE TABLE SessioneOnline (
2     IdSessione SERIAL PRIMARY KEY ,
3     Link VARCHAR(1000) NOT NULL ,
4     Data TIMESTAMP NOT NULL ,
5     Durata Integer NOT NULL ,
6     NumSessione Integer NOT NULL ,
7     IdCorso INTEGER NOT NULL ,
8
9     CONSTRAINT FK_SessioneOnline_Corso FOREIGN KEY (IdCorso) REFERENCES
10        Corso(idCorso) ON DELETE CASCADE , -- foreign key verso Corso, su
11        eliminazione di un corso elimina tutte le sue sessioni online
12     CONSTRAINT CK_Link CHECK (Link LIKE 'https://%'), -- controllo che il
13        link abbia un formato valido
14     CONSTRAINT CK_Durata CHECK (Durata >= 60 AND Durata <= 180) -- controllo
15        che la durata sia compresa tra 60 e 180 minuti
16 )

```

Listing 4.6: Creazione tabella Sessione Online

La tabella **SessioneOnline** rappresenta le sessioni online dei corsi di cucina. Ogni sessione ha un identificatore univoco (IdSessione), un link, una data, una durata, un numero di sessione e un riferimento al corso a cui appartiene.

#### 4.1.7 Definizione della tabella ADESIONE

```

1 CREATE TABLE Adesione (
2     UsernameUtente VARCHAR(100) NOT NULL ,
3     IdSessionePresenza INTEGER NOT NULL ,
4
5     CONSTRAINT PK_Adesione PRIMARY KEY (UsernameUtente , IdSessionePresenza) ,
6         -- chiave primaria composta
7     CONSTRAINT FK_Adesione_Utente FOREIGN KEY (UsernameUtente) REFERENCES
8        Utente(username) ON DELETE CASCADE , -- foreign key verso Utente, su
9        eliminazione di un utente elimina tutte le sue adesioni
10    CONSTRAINT FK_Adesione_SessionePresenza FOREIGN KEY (IdSessionePresenza)
11        REFERENCES SessioneInPresenza(IdSessione) ON DELETE CASCADE --
12        foreign key verso SessioneInPresenza , su eliminazione di una sessione
13        elimina tutte le adesioni ad essa collegate
14 )

```

Listing 4.7: Creazione tabella Adesione

La tabella **Adesione** rappresenta le adesioni degli utenti alle sessioni in presenza dei corsi di cucina. Ogni adesione ha un riferimento all'utente e alla sessione in presenza a cui si riferisce. Questa tabella è necessaria per calcolare il numero di ingredienti necessari per ogni sessione in presenza.

#### 4.1.8 Definizione della tabella PREPARA

```
1 CREATE TABLE Prepara (
2     IdSessionePresenza INTEGER NOT NULL,
3     IdRicetta INTEGER NOT NULL,
4
5     CONSTRAINT PK_Prepara PRIMARY KEY (IdSessionePresenza, IdRicetta), --
6         chiave primaria composta
7     CONSTRAINT FK_Prepara_SessionePresenza FOREIGN KEY (IdSessionePresenza)
8         REFERENCES SessioneInPresenza(IdSessione) ON DELETE CASCADE, --
9             foreign key verso SessioneInPresenza, su eliminazione di una sessione
10            elimina tutte le preparazioni ad essa collegate
11     CONSTRAINT FK_Prepara_Ricetta FOREIGN KEY (IdRicetta) REFERENCES
12         Ricetta(IdRicetta) --foreign key verso Ricetta
13 )
```

Listing 4.8: Creazione tabella Prepara

La tabella **Prepara** rappresenta la relazione tra le sessioni in presenza e le ricette. Ogni record in questa tabella associa una sessione in presenza a una ricetta specifica.

#### 4.1.9 Definizione della tabella RICETTA

```
1 CREATE TABLE Ricetta (
2     IdRicetta SERIAL PRIMARY KEY,
3     Nome VARCHAR(100) NOT NULL UNIQUE,
4     Descrizione VARCHAR(1000) NOT NULL
5 )
```

Listing 4.9: Creazione tabella Ricetta

La tabella **Ricetta** rappresenta le ricette associate alle sessioni in presenza dei corsi di cucina. Ogni ricetta ha un identificatore univoco (IdRicetta), un nome e una descrizione.

#### 4.1.10 Definizione della tabella INGREDIENTE

```
1 CREATE TABLE Ingrediente (
2     IdIngrediente SERIAL PRIMARY KEY,
3     Nome VARCHAR(100) NOT NULL UNIQUE,
4     Allergene BOOLEAN NOT NULL,
5     UnitàMisura VARCHAR(50) NOT NULL
6 )
```

Listing 4.10: Creazione tabella Ingrediente

La tabella **Ingrediente** rappresenta gli ingredienti utilizzati nelle ricette. Ogni ingrediente ha un identificatore univoco (IdIngrediente), un nome, un'unità di misura e un flag che indica se è un allergene.

#### 4.1.11 Definizione della tabella COMPOSIZIONE

```
1 CREATE TABLE Composizione (
2     IdIngrediente INTEGER NOT NULL,
3     IdRicetta INTEGER NOT NULL,
4     Quantità FLOAT NOT NULL,
5
6     CONSTRAINT PK_Composizione PRIMARY KEY (IdIngrediente, IdRicetta), --
7         chiave primaria composta
8     CONSTRAINT FK_Composizione_Ingrediente FOREIGN KEY (IdIngrediente)
9         REFERENCES Ingrediente(IdIngrediente), -- foreign key verso
10        Ingrediente
11     CONSTRAINT FK_Composizione_Ricetta FOREIGN KEY (IdRicetta) REFERENCES
12        Ricetta(IdRicetta), -- foreign key verso Ricetta
13     CONSTRAINT CK_Quantità_Positiva CHECK (Quantità > 0) -- controllo
14        quantità ingrediente positiva
15 )
```

Listing 4.11: Creazione tabella Composizione

La tabella **Composizione** rappresenta la relazione tra le ricette e gli ingredienti. Ogni record in questa tabella associa una ricetta a varo ingredienti che la compongono, specificando la quantità necessaria per ciascuno di essi.

#### 4.1.12 Definizione della tabella NOTIFICA

```
1 CREATE TABLE Notifica (
2     IdNotifica SERIAL PRIMARY KEY,
3     UsernameChef VARCHAR(100) NOT NULL,
4     Oggetto VARCHAR(1000) NOT NULL,
5     Testo VARCHAR(1000),
6     DataInvio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
7
8     CONSTRAINT FK_notifica_chef FOREIGN KEY (UsernameChef) REFERENCES
9         Chef(Username) ON DELETE CASCADE -- foreign key verso Chef, su
10        eliminazione di un chef elimina tutte le notifiche da lui inviate
11 )
```

Listing 4.12: Creazione tabella Notifica

La tabella **Notifica** rappresenta le notifiche inviate dagli chef agli utenti iscritti ai corsi. Ogni notifica ha un identificatore univoco (IdNotifica), una data di invio, un oggetto, un testo e un riferimento allo chef che l'ha inviata.

#### 4.1.13 Definizione della tabella RICEVE

```
1 CREATE TABLE Riceve (
2     IdNotifica INTEGER NOT NULL ,
3     UsernameUtente VARCHAR(100) NOT NULL ,
4
5     CONSTRAINT PK_Riceve PRIMARY KEY (IdNotifica, UsernameUtente), -- chiave
6         primaria composta
7     CONSTRAINT FK_Riceve_Notifica FOREIGN KEY (IdNotifica) REFERENCES
8         Notifica(IdNotifica) ON DELETE CASCADE, -- foreign key verso
9             Notifica, su eliminazione di una notifica elimina tutte le ricezioni
10            ad essa collegate
11     CONSTRAINT FK_Riceve_Utente FOREIGN KEY (UsernameUtente) REFERENCES
12         Utente.Username) ON DELETE CASCADE -- foreign key verso Utente, su
13             eliminazione di un utente elimina tutte le sue ricezioni
14 )
```

Listing 4.13: Creazione tabella Riceve

La tabella **Riceve** rappresenta la relazione tra le notifiche e gli utenti che le ricevono. Ogni record in questa tabella associa una notifica a un utente specifico, consentendo di tenere traccia delle notifiche inviate a ciascun utente.

#### 4.1.14 Definizione della tabella INSEGNA

```
1 CREATE TABLE Insegna (
2     IdTopic INTEGER NOT NULL ,
3     UsernameChef VARCHAR(1000) NOT NULL ,
4
5     CONSTRAINT PK_Insegna PRIMARY KEY (IdTopic, UsernameChef), -- chiave
6         primaria composta
7     CONSTRAINT FK_Insegna_Topic FOREIGN KEY (IdTopic) REFERENCES
8         Topic(idTopic), -- foreign key verso Topic
9     CONSTRAINT FK_Insegna_Chef FOREIGN KEY (UsernameChef) REFERENCES
10        Chef.Username) ON DELETE CASCADE -- foreign key verso Chef, su
11            eliminazione di un chef elimina tutte le relative righe in Insegna
12 )
```

Listing 4.14: Creazione tabella Insegna

La tabella **Insegna** rappresenta la relazione tra i topic e gli chef che li insegnano. Ogni record in questa tabella associa un topic a uno chef specifico, consentendo di tenere traccia dei topic insegnati da ciascun chef.

#### 4.1.15 Definizione della tabella ISCRIZIONE

```
1 CREATE TABLE Iscrizione (
2     IdCorso INTEGER NOT NULL,
3     UsernameUtente VARCHAR(100) NOT NULL,
4     DataIscrizione TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
5
6     CONSTRAINT PK_Iscrizione PRIMARY KEY (IdCorso, UsernameUtente), -- chiave
7         primaria composta
8     CONSTRAINT FK_Iscrizione_Corso FOREIGN KEY (IdCorso) REFERENCES
9         Corso(idCorso) ON DELETE CASCADE, -- foreign key verso Corso, su
10        eliminazione di un corso elimina tutte le sue iscrizioni
11     CONSTRAINT FK_Iscrizione_Utente FOREIGN KEY (UsernameUtente) REFERENCES
12         Utente(username) ON DELETE CASCADE -- foreign key verso Utente, su
13        eliminazione di un utente elimina tutte le sue iscrizioni
14 )
15 )
```

Listing 4.15: Creazione tabella Iscrizione

La tabella **Iscrizione** rappresenta le iscrizioni degli utenti ai corsi di cucina. Ogni iscrizione ha un riferimento all'utente e al corso a cui si riferisce, insieme alla data di iscrizione. Questa tabella è necessaria per tenere traccia degli utenti iscritti a ciascun corso e per gestire le notifiche relative ai corsi.

## 4.2 Triggers, Procedures, Functions e Vincoli

Durante la progettazione fisica della base di dati, sono stati implementati alcuni triggers, procedures, functions e vincoli per garantire l'integrità dei dati e automatizzare alcune operazioni. Di seguito sono riportati alcuni esempi:

### 4.2.1 Funzione per calcolare il numero di adesioni

```
1 CREATE OR REPLACE FUNCTION CalcoloNumeroAdesioni(idSessione INTEGER) --
2     funzione per calcolare il numero di adesioni a una sessione in presenza
3 RETURNS INTEGER
4 LANGUAGE plpgsql
5 AS $$
6 DECLARE
7     numeroAdesioni INTEGER;
8 BEGIN
9     SELECT COUNT(*) INTO numeroAdesioni
10    FROM Adesione AS a
11   WHERE a.idSessione = idSessione;
12   IF numeroAdesioni = 0 THEN
13       RAISE EXCEPTION 'Nessuna adesione trovata per la sessione con ID %',
14                         idSessione;
15   END IF;
16   RETURN numeroAdesioni;
17 END;
18 $$;
```

Listing 4.16: Funzione per calcolare il numero di adesioni

Questa funzione calcola il numero di adesioni per una sessione in presenza. Necessaria per pianificare correttamente la quantità di ingredienti necessari per ogni sessione in presenza.

#### 4.2.2 Trigger per controllare che l'anno del corso sia valido

```
1 CREATE OR REPLACE FUNCTION ControlloAnnoCorso()
2 RETURNS TRIGGER
3 LANGUAGE plpgsql
4 AS $$
5 BEGIN
6     -- Controlla se l'anno del corso è compreso tra l'anno corrente e il
7     -- prossimo
8     IF NEW.Anno < EXTRACT(YEAR FROM CURRENT_DATE) OR
9         NEW.Anno > EXTRACT(YEAR FROM CURRENT_DATE) +1 THEN
10        RAISE EXCEPTION 'L''anno del corso deve essere compreso tra l''anno
11        corrente e il prossimo.';
12    END IF;
13    RETURN NEW;
14
15
16 CREATE TRIGGER TriggerAnnoCorso
17 BEFORE INSERT OR UPDATE ON Corso
18 FOR EACH ROW
19 EXECUTE FUNCTION ControlloAnnoCorso();
```

Listing 4.17: Trigger per controllare l'anno del corso

Questo trigger viene attivato prima dell'inserimento o aggiornamento di un nuovo corso e verifica che l'anno del corso sia valido. Se l'anno non è valido, il trigger genera un errore e impedisce l'inserimento del corso. Questo è utile per garantire che i corsi siano pianificati correttamente e che le date siano coerenti con l'anno accademico corrente o successivo.

#### 4.2.3 Trigger per controllare che l'aula della sessione in presenza non sia occupata

```

1 CREATE OR REPLACE FUNCTION ControlloAulaOccupata()
2 RETURNS TRIGGER
3 LANGUAGE plpgsql
4 AS $$
5 BEGIN
6     -- Controlla se esiste già una sessione in presenza nello stesso luogo e
7     -- aula che si sovrappone con la nuova sessione
8     IF EXISTS (
9         SELECT 1
10        FROM SessioneInPresenza AS sip
11       WHERE sip.Luogo = NEW.Luogo
12         AND sip.Aula = NEW.Aula
13        -- La nuova sessione inizia durante una sessione esistente
14       AND (NEW.Data >= sip.Data AND NEW.Data < sip.Data + (sip.Durata
15          || ' minutes')::INTERVAL)
16        -- La nuova sessione finisce durante una sessione esistente
17       OR (NEW.Data + (NEW.Durata || ' minutes')::INTERVAL >
18           sip.Data AND NEW.Data + (NEW.Durata || ' minutes')::INTERVAL <=
19             sip.Data + (sip.Durata || ' minutes')::INTERVAL)
20        -- La nuova sessione contiene completamente una sessione
21        esistente
22       OR (NEW.Data <= sip.Data AND NEW.Data + (NEW.Durata || ' minutes')::INTERVAL >=
23             sip.Data + (sip.Durata || ' minutes')::INTERVAL)
24    ) THEN
25        RAISE EXCEPTION 'L''aula % nel luogo % è occupata in data %',
26                      NEW.Aula, NEW.Luogo, NEW.Data;
27    END IF;
28
29    RETURN NEW;
30 END;
$$;

CREATE TRIGGER Tr_Insert_Aula_Sessione_Presenza
BEFORE INSERT OR UPDATE ON SessioneInPresenza
FOR EACH ROW
EXECUTE FUNCTION ControlloAulaOccupata();

```

Listing 4.18: Trigger per controllare l'aula della sessione in presenza

Questo trigger viene attivato prima dell'inserimento o aggiornamento di una sessione in presenza e verifica che l'aula specificata non sia già occupata da un'altra sessione in presenza nello stesso orario. Se l'aula è occupata, il trigger genera un errore e impedisce l'inserimento della sessione. Questo è utile per evitare conflitti di programmazione e garantire che le sessioni in presenza possano essere svolte senza problemi logistici.

#### 4.2.4 Trigger per controllare che l'utente che aderisce ad una sessione in presenza sia iscritto al corso

```

1 CREATE OR REPLACE FUNCTION ControlloAdesioniCorso()
2 RETURNS TRIGGER
3 LANGUAGE plpgsql
4 AS $$
5 DECLARE
6     corso_id INTEGER;
7 BEGIN
8     -- Recupera l'IdCorso dalla sessione in presenza
9     SELECT sip.IdCorso
10    INTO corso_id
11   FROM SessioneInPresenza AS sip
12  WHERE sip.IdSessione = NEW.IdSessionePresenza;
13
14     -- Controlla se l'utente non è iscritto al corso e nel caso solleva
15     -- un'eccezione
16     IF NOT EXISTS (
17         SELECT 1
18         FROM Iscrizione AS i
19        WHERE i.IdCorso = corso_id
20        AND i.UsernameUtente = NEW.UsernameUtente
21     ) THEN
22         RAISE EXCEPTION 'L''utente % non è iscritto al corso %.',(
23             NEW.UsernameUtente, (SELECT Titolo FROM Corso WHERE idCorso =
24             corso_id));
25     END IF;
26
27     RETURN NEW;
28 END;
29 $$;
30
31 CREATE TRIGGER TR_Insert_Adesione
32 BEFORE INSERT ON Adesione
33 FOR EACH ROW
34 EXECUTE FUNCTION ControlloAdesioniCorso();

```

Listing 4.19: Trigger per controllare l'iscrizione dell'utente

Questo trigger viene attivato prima dell'inserimento di una nuova adesione e verifica che l'utente che sta aderendo a una sessione in presenza sia iscritto al corso corrispondente. Se l'utente non è iscritto al corso, il trigger genera un errore e impedisce l'inserimento dell'adesione. Questo è utile per garantire che solo gli utenti iscritti ai corsi possano partecipare alle sessioni in presenza, mantenendo la coerenza dei dati e le regole del sistema.

#### 4.2.5 Trigger per controllare che lo chef insegni il topic del corso che tiene

```
1 CREATE OR REPLACE FUNCTION ControlloInsegna()
2 RETURNS TRIGGER
3 LANGUAGE plpgsql
4 AS $$
5 BEGIN
6     -- Controlla se lo Chef non insegnava il Topic del Corso e nel caso solleva
7     -- un'eccezione
8     IF NOT EXISTS (
9         SELECT 1
10        FROM Insegna AS i
11       WHERE NEW.IdTopic = i.IdTopic
12         AND i.UsernameChef = NEW.UsernameChef
13     ) THEN RAISE EXCEPTION 'Lo Chef % non insegnava il Topic %, è quindi
14     impossibilitato a creare il corso.', 
15         NEW.UsernameChef, (SELECT t.Nome FROM Topic AS t WHERE t.IdTopic =
16             NEW.IdTopic);
17     END IF;
18     RETURN NEW;
19 END;
20 $$;
21
22 CREATE TRIGGER Tr_Insert_Corso_Topic
23 AFTER INSERT ON Corso
24 FOR EACH ROW
25 EXECUTE FUNCTION ControlloInsegna();
```

Listing 4.20: Trigger per controllare il topic del corso

Questo trigger viene attivato prima dell'inserimento di un nuovo corso e verifica che lo chef che sta insegnando il corso sia effettivamente associato al topic del corso stesso. Se lo chef non insegnava il topic specificato, il trigger genera un errore e impedisce l'inserimento del corso. Questo è utile per garantire che i corsi siano tenuti da chef competenti nei rispettivi argomenti, mantenendo la qualità dell'insegnamento.

#### 4.2.6 Trigger per controllare che le sessioni siano programmate in date successive alla data di inizio del corso

```
1 CREATE OR REPLACE FUNCTION ControlloSessioneCorsoData()
2 RETURNS TRIGGER
3 LANGUAGE plpgsql
4 AS $$
5 BEGIN
6     -- Controlla che ogni sessione sia programmata in una data successiva
7     -- all'inizio del corso
8     IF NEW.Data < (SELECT DataInizio FROM Corso WHERE IdCorso = NEW.IdCorso)
9     THEN
10        RAISE EXCEPTION 'La data della sessione deve essere successiva alla
11        data di inizio del corso';
12    END IF;
13    RETURN NEW;
14 END;
15 $$;
16
17 CREATE TRIGGER TR_Insert_Data_Sessione_In_Presenza_con_Corso
18 BEFORE INSERT OR UPDATE ON SessioneInPresenza
19 FOR EACH ROW
20 EXECUTE FUNCTION ControlloSessioneCorsoData();
21
22 CREATE TRIGGER TR_Insert_Data_Sessione_Online_con_Corso
23 BEFORE INSERT OR UPDATE ON SessioneOnline
24 FOR EACH ROW
25 EXECUTE FUNCTION ControlloSessioneCorsoData();
```

Listing 4.21: Trigger per controllare le date delle sessioni

Questo trigger viene attivato prima dell'inserimento o aggiornamento di una sessione e verifica che la data della sessione sia successiva alla data di inizio del corso a cui appartiene. Se la data della sessione non è valida, il trigger genera un errore e impedisce l'inserimento della sessione. Questo è utile per garantire che le sessioni siano programmate correttamente in relazione al corso, evitando conflitti di programmazione e garantendo che le sessioni si svolgano dopo l'inizio del corso stesso.

#### 4.2.7 Trigger per controllare che la data di inizio del corso sia valida

```
1 CREATE OR REPLACE FUNCTION ControllaDataInizio()
2 RETURNS TRIGGER
3 LANGUAGE plpgsql
4 AS $$
5 BEGIN
6     -- Controlla che la data di inizio del corso sia maggiore o uguale alla
7     -- data corrente
8     IF NEW.DataInizio < CURRENT_DATE THEN
9         RAISE EXCEPTION 'La data di inizio del corso non può essere nel
10         passato: %', NEW.DataInizio;
11     END IF;
12     RETURN NEW;
13 END;
14 $$;
15
16 CREATE TRIGGER TR_Insert_DataInizio_Corso
```

```

15 BEFORE INSERT OR UPDATE ON Corso
16 FOR EACH ROW
17 EXECUTE FUNCTION ControllaDataInizio();

```

Listing 4.22: Trigger per controllare la data di inizio del corso

Questo trigger viene attivato prima dell'inserimento o aggiornamento di un nuovo corso e verifica che la data di inizio del corso sia valida. Se la data non è valida, il trigger genera un errore e impedisce l'inserimento del corso. Questo è utile per garantire che i corsi siano pianificati correttamente e che le date siano coerenti.

#### 4.2.8 Trigger per controllare che la data delle sessioni sia valida

```

1 CREATE OR REPLACE FUNCTION ControlloDataSessione()
2 RETURNS TRIGGER
3 LANGUAGE plpgsql
4 AS $$ 
5 BEGIN
6     -- Controlla che la data della sessione sia maggiore o uguale alla data
7     -- corrente
8     IF NEW.Data < CURRENT_TIMESTAMP THEN
9         RAISE EXCEPTION 'La sessione non può essere programmata per una data
10            passata: %', NEW.Data;
11    END IF;
12    RETURN NEW;
13 END;
14 $$;
15
16
17 CREATE TRIGGER TR_Insert_DataSessione_Online
18 BEFORE INSERT OR UPDATE ON SessioneOnline
19 FOR EACH ROW
20 EXECUTE FUNCTION ControlloDataSessione();
21
22 CREATE TRIGGER TR_Insert_DataSessione_Presenza
23 BEFORE INSERT OR UPDATE ON SessioneInPresenza
24 FOR EACH ROW
25 EXECUTE FUNCTION ControlloDataSessione();

```

Listing 4.23: Trigger per controllare la data delle sessioni

Questo trigger viene attivato prima dell'inserimento o aggiornamento di una sessione e verifica che la data della sessione sia valida. Se la data non è valida, il trigger genera un errore e impedisce l'inserimento della sessione. Questo è utile per garantire che le sessioni siano programmate correttamente e che le date siano coerenti.

#### 4.2.9 Trigger per controllare che il corso non abbia raggiunto il limite di 50 iscrizioni

```

1 CREATE OR REPLACE FUNCTION ControlloLimiteIscrizioni()
2 RETURNS TRIGGER
3 LANGUAGE plpgsql
4 AS $$ 
5 DECLARE
6     MaxNumeroIscrizioni INTEGER:= 50;
7     NumeroIscrizioni INTEGER;
8 BEGIN
9     SELECT COUNT(*) INTO NumeroIscrizioni
10    FROM Iscrizione
11   WHERE IdCorso = NEW.IdCorso;
12   IF NumeroIscrizioni >= MaxNumeroIscrizioni THEN
13       RAISE EXCEPTION 'Il numero massimo di iscrizioni per il corso % è
14           stato raggiunto (% iscritti). Non è possibile aggiungere
15           ulteriori iscrizioni.',
16           (SELECT Titolo FROM Corso WHERE idCorso = NEW.IdCorso),
17           NumeroIscrizioni;
18   END IF;
19
20   RETURN NEW;
21 END;
22 $$;
23
24 CREATE TRIGGER TR_Insert_Iscrizione
25 BEFORE INSERT ON Iscrizione
26 FOR EACH ROW
27 EXECUTE FUNCTION ControlloLimiteIscrizioni();

```

Listing 4.24: Trigger per controllare il limite di iscrizioni al corso

Questo trigger viene attivato prima dell'inserimento di una nuova iscrizione e verifica che il corso a cui l'utente sta cercando di iscriversi non abbia già raggiunto il limite di 50 iscrizioni. Se il limite è stato raggiunto, il trigger genera un errore e impedisce l'inserimento dell'iscrizione. Questo è utile per garantire che i corsi non superino il numero massimo di partecipanti, mantenendo la qualità dell'insegnamento e la gestione delle risorse.

#### 4.2.10 Trigger per controllare che il numero sessione sia coerente con il numero della sessione precedente

```

1 CREATE OR REPLACE FUNCTION ControlloNumSessione()
2 RETURNS TRIGGER
3 LANGUAGE plpgsql
4 AS $$ 
5 DECLARE
6     MaxNumSessioni INT;
7     SessioniEsistenti INT;
8 BEGIN
9     -- Conta tutte le sessioni esistenti per questo corso
10    SELECT COUNT(*) INTO SessioniEsistenti
11    FROM (
12        SELECT IdCorso FROM SessioneOnline WHERE IdCorso = NEW.IdCorso
13        UNION
14        SELECT IdCorso FROM SessioneInPresenza WHERE IdCorso = NEW.IdCorso
15    );
16

```

```

17    -- Se non ci sono sessioni esistenti, deve essere la prima
18    IF SessioniEsistenti = 0 THEN
19        IF NEW.NumSessione <> 1 THEN
20            RAISE EXCEPTION 'Il numero di sessione % non può essere diverso
21                da 1 per la prima sessione del corso %',
22                NEW.NumSessione, NEW.IdCorso;
23        END IF;
24        RETURN NEW;
25    END IF;

26    -- Trova il numero massimo di sessione esistente da entrambe le tabelle
27    SELECT MAX(NumSessione) INTO MaxNumSessioni
28    FROM (
29        SELECT NumSessione FROM SessioneOnline WHERE IdCorso = NEW.IdCorso
30        UNION
31        SELECT NumSessione FROM SessioneInPresenza WHERE IdCorso = NEW.IdCorso
32    );
33

34    -- Controlla che il nuovo numero sia il successivo
35    IF NEW.NumSessione <= MaxNumSessioni THEN
36        RAISE EXCEPTION 'Il numero di sessione % non può essere minore o
37            uguale al numero dell''ultima sessione % per il corso %',
38            NEW.NumSessione, MaxNumSessioni, (SELECT Titolo FROM Corso WHERE
39                idCorso = NEW.IdCorso);
40    ELSIF NEW.NumSessione > MaxNumSessioni + 1 THEN
41        RAISE EXCEPTION 'Il numero di sessione % non può essere maggiore di %
42            per il corso %',
43            NEW.NumSessione, MaxNumSessioni + 1, (SELECT Titolo FROM Corso
44                WHERE idCorso = NEW.IdCorso);
45    END IF;

46    RETURN NEW;
47 END;
48 $$;

49 -- Trigger per SessioneOnline
50 CREATE TRIGGER TR_Insert_NumSessione_Online
51 BEFORE INSERT ON SessioneOnline
52 FOR EACH ROW
53 EXECUTE FUNCTION ControlloNumSessione();

54 -- Trigger per SessioneInPresenza
55 CREATE TRIGGER TR_Insert_NumSessione_Presenza
56 BEFORE INSERT ON SessioneInPresenza
57 FOR EACH ROW
58 EXECUTE FUNCTION ControlloNumSessione();

```

Listing 4.25: Trigger per controllare il numero della sessione

Questo trigger viene attivato prima dell'inserimento di una sessione e verifica che il numero della sessione sia coerente con il numero della sessione precedente. Se il numero della sessione non è valido, il trigger genera un errore e impedisce l'inserimento della sessione. Questo è utile per garantire che le sessioni siano programmate in modo sequenziale e che non ci siano salti nei numeri delle sessioni, mantenendo la coerenza del sistema.

#### 4.2.11 Trigger per controllare che non ci sia una sovrapposizione di date tra le sessioni in presenza per lo stesso chef

```
1 CREATE OR REPLACE FUNCTION ControlloSovrapposizione()
2 RETURNS TRIGGER
3 LANGUAGE plpgsql
4 AS $$
5 DECLARE
6     ChefUsername VARCHAR(100);
7 BEGIN
8     -- Ottieni lo username del chef per il corso
9     SELECT c.UsernameChef INTO ChefUsername
10    FROM Corso AS c
11   WHERE c.idCorso = NEW.IdCorso;
12
13     -- Controlla se esistono sovrapposizioni con le sessioni del chef
14    IF EXISTS (
15        SELECT 1
16        FROM (
17            -- seleziona data di inizio e fine delle sessioni online dello
18            -- chef
19            SELECT so.Data, so.Data + (so.Durata || ' minutes')::INTERVAL AS
20                DataFine
21            FROM SessioneOnline AS so
22            JOIN Corso AS c ON so.IdCorso = c.idCorso
23            WHERE c.UsernameChef = ChefUsername
24
25            UNION ALL
26
27            -- seleziona data di inizio e fine delle sessioni in presenza
28            -- dello chef
29            SELECT sip.Data, sip.Data + (sip.Durata || ' minutes')::INTERVAL
30                AS DataFine
31            FROM SessioneInPresenza AS sip
32            JOIN Corso AS c ON sip.IdCorso = c.idCorso
33            WHERE c.UsernameChef = ChefUsername
34        ) AS SessioniEsistenti
35        WHERE NEW.Data < DataFine
36        AND NEW.Data + (NEW.Durata || ' minutes')::INTERVAL > Data
37    ) THEN
38        RAISE EXCEPTION 'Lo chef % ha già una sessione programmata che si
39        sovrappone con l''orario % - %.', 
40        ChefUsername, NEW.Data, NEW.Data + (NEW.Durata || '
41        minutes')::INTERVAL;
42    END IF;
43
44    RETURN NEW;
45 END;
46 $$;
47
48 -- Trigger per SessioneOnline
49 CREATE TRIGGER TR_Controllo_Sovrapposizione_Online
50 BEFORE INSERT ON SessioneOnline
51 FOR EACH ROW
52 EXECUTE FUNCTION ControlloSovrapposizione();
```

```

48 -- Trigger per SessioneInPresenza
49 CREATE TRIGGER TR_Controllo_Sovrapposizione_Presenza
50 BEFORE INSERT ON SessioneInPresenza
51 FOR EACH ROW
52 EXECUTE FUNCTION ControlloSovrapposizione();

```

Listing 4.26: Trigger per controllare la sovrapposizione di date tra le sessioni in presenza

Questo trigger viene attivato prima dell'inserimento di una sessione in presenza e verifica che non ci sia una sovrapposizione di date tra le sessioni in presenza per lo stesso chef. Se c'è una sovrapposizione, il trigger genera un errore e impedisce l'inserimento della sessione. Questo è utile per evitare conflitti di programmazione e garantire che gli chef possano gestire le loro sessioni in presenza senza problemi logistici.

#### 4.2.12 Trigger per controllare che gli username degli chef e degli utenti siano univoci

```

1 -- Trigger per controllare username unico tra Chef e Utente
2
3 -- Function per verificare username unico per Chef
4 CREATE OR REPLACE FUNCTION check_chef_username_unique()
5 RETURNS TRIGGER AS $$%
6 BEGIN
7     IF EXISTS (SELECT 1 FROM Utente WHERE Username = NEW.Username) THEN
8         RAISE EXCEPTION 'Username % già esistente!', NEW.Username;
9     END IF;
10
11    RETURN NEW;
12 END;
13 $$ LANGUAGE plpgsql;
14
15 -- Function per verificare username unico per Utente
16 CREATE OR REPLACE FUNCTION check_utente_username_unique()
17 RETURNS TRIGGER AS $$%
18 BEGIN
19     IF EXISTS (SELECT 1 FROM Chef WHERE Username = NEW.Username) THEN
20         RAISE EXCEPTION 'Username % già esistente!', NEW.Username;
21     END IF;
22
23    RETURN NEW;
24 END;
25 $$ LANGUAGE plpgsql;
26
27 -- Trigger per Chef
28 CREATE TRIGGER trigger_chef_username_unique
29     BEFORE INSERT OR UPDATE ON Chef
30     FOR EACH ROW
31     EXECUTE FUNCTION check_chef_username_unique();
32
33 -- Trigger per Utente
34 CREATE TRIGGER trigger_utente_username_unique
35     BEFORE INSERT OR UPDATE ON Utente
36     FOR EACH ROW
37     EXECUTE FUNCTION check_utente_username_unique();

```

Listing 4.27: Trigger per controllare l'unicità degli username

Questo trigger viene attivato prima dell'inserimento o aggiornamento di un nuovo chef o utente e verifica che lo username sia univoco. Se lo username è già presente nel database, il trigger genera un errore e impedisce l'inserimento o l'aggiornamento. Questo è utile per garantire che gli username degli chef e degli utenti siano univoci, evitando conflitti e garantendo l'integrità dei dati.

# Capitolo 5

## Popolamento del database

### 5.1 Inserimento dei dati

Il database è stato popolato con dati di esempio per testare le funzionalità e verificare il corretto funzionamento delle query. Sono stati inseriti corsi, sessioni, adesioni, ricette e ingredienti, in modo da simulare un ambiente di apprendimento realistico. Di seguito sono riportati alcuni esempi di query utilizzate per il popolamento del database.

#### 5.1.1 Query di inserimento degli chef

```
1  INSERT INTO Chef(Nome, Cognome, Username, Password)
2  VALUES
3      ('Giuseppe', 'Esposito', 'ChefPino', 'PasswordGiuseppe'),
4      ('Maria', 'Rossi', 'ChefMaria', 'PasswordMaria'),
5      ('Antonio', 'Bianchi', 'ChefAntonio', 'PasswordAntonio'),
6      ('Francesca', 'Verdi', 'ChefFrancesca', 'PasswordFrancesca'),
7      ('Luca', 'Gallo', 'ChefLuca', 'PasswordLuca'),
8      ('Christian', 'Ranavolo', 'Il_Nerdone_11', 'PasswordChristian'),
9      ('Elena', 'Martini', 'ChefElena', 'PasswordElena'),
10     ('Marco', 'Conti', 'ChefMarco', 'PasswordMarco'),
11     ('Sara', 'Ferrari', 'ChefSara', 'PasswordSara'),
12     ('Alessandro', 'Romano', 'ChefAle', 'PasswordAlessandro'),
13     ('Kenshiro', 'Hokuto', 'KenIlGuerriero', 'PasswordKenshiro'),
14     ('Giorgio', 'Bruno', 'ChefGiorgio', 'PasswordGiorgio'),
15     ('Valentina', 'Greco', 'ChefValentina', 'PasswordValentina'),
16     ('Simone', 'Rinaldi', 'ChefSimone', 'PasswordSimone'),
17     ('Chiara', 'Fontana', 'ChefChiara', 'PasswordChiara'),
18     ('Matteo', 'Leone', 'ChefMatteo', 'PasswordMatteo'),
19     ('Giulia', 'Caruso', 'ChefGiulia', 'PasswordGiulia'),
20     ('Giuseppe', 'Piccirillo', 'ChefPeppe', 'PasswordGiuseppe'),
21     ('Roberta', 'Marino', 'ChefRoberta', 'PasswordRoberta'),
22     ('Francesco', 'Gatti', 'ChefFrancesco', 'PasswordFrancesco'),
23     ('Elisa', 'Rossi', 'ChefElisa', 'PasswordElisa'),
24     ('Nicola', 'Barbieri', 'ChefNicola', 'PasswordNicola'),
25     ('Martina', 'Sanna', 'ChefMartina', 'PasswordMartina'),
26     ('Alessia', 'Piras', 'ChefAlessia', 'PasswordAlessia'),
27     ('Alessandro', 'Bologna', 'FranchinoErCriminale', 'PasswordAlessandro'),
28     ('Laura', 'Moretti', 'ChefLaura', 'PasswordLaura'),
```

```

29   ('Stefano', 'Ruggiero', 'ChefStefano', 'PasswordStefano'),
30   ('Mirko', 'Alessandrini', 'CiccioGames', 'PasswordMirko'),
31   ('Gabriele', 'Marchetti', 'ChefGabriele', 'PasswordGabriele'),
32   ('Ilaria', 'Ricci', 'ChefIlaria', 'PasswordIlaria'),
33   ('Tommaso', 'Fontanella', 'ChefTommaso', 'PasswordTommaso');

```

Listing 5.1: Query di inserimento degli chef

Questa query inserisce gli chef nel database, specificando il loro username, nome, cognome e password. Gli chef sono i responsabili dei corsi di cucina.

### 5.1.2 Query di inserimento degli utenti

```

1  INSERT INTO Utente(Nome, Cognome, Username, Password)
2  VALUES ('Giuliano', 'Laccetti', 'giuliano.laccetti', 'NapoliCampione2026'),
3    ('Luigi', 'Mario', 'luigi.mario', 'passwordLuigi'),
4    ('Mario', 'Mario', 'mario.mario', 'passwordMario'),
5    ('Christian', 'Ranavolo', 'IL_NERDONE_11', 'Zombie11'),
6    ('Bruno', 'Capuozzo', 'bohqualcosaboh', 'passwordBruno'),
7    ('Niccolò', 'Cavaldo', 'pete_990', 'passwordNiccolò'),
8    ('Carmine', 'Amato', 'xFireFox27', 'passwordCarmine'),
9    ('Kenneth', 'Caselli', 'Zeb89', 'passwordZeb'),
10   ('Antonio', 'Capuozzo', 'antonio.capuozzo', 'passwordAntonio'),
11   ('Giordano', 'Bruno', 'giordano.bruno', 'passwordGiordano'),
12   ('Francesco', 'De Luca', 'francesco.deluca', 'passwordFrancesco'),
13   ('Alessandro', 'Rossi', 'alessandro.rossi', 'passwordAlessandro'),
14   ('Federico', 'Borghi', 'federico.borghi', 'passwordFederico'),
15   ('Matteo', 'Ruggiero', 'matteo.ruggiero', 'passwordMatteo'),
16   ('Stefano', 'Martini', 'stefano.martini', 'passwordStefano'),
17   ('Giorgio', 'Bianchi', 'giorgio.bianchi', 'passwordGiorgio'),
18   ('Luca', 'Verdi', 'luca.verdi', 'passwordLuca'),
19   ('Marco', 'Storti', 'marco.storti', 'passwordMarco'),
20   ('Andrea', 'Simpatico', 'andrea.simpatico', 'passwordAndrea'),
21   ('Elena', 'Scalzi', 'elena.scalzi', 'passwordElena'),
22   ('Chiara', 'Gallo', 'chiara.gallo', 'passwordChiara'),
23   ('Sara', 'Botticelli', 'sara.botticelli', 'passwordSara'),
24   ('Giulia', 'Manzoni', 'giulia.manzoni', 'passwordGiulia'),
25   ('Martina', 'Rinaldi', 'martina.rinaldi', 'passwordMartina'),
26   ('Alice', 'Gaudino', 'alice.gaudino', 'passwordAlice'),
27   ('Francesca', 'Ferrari', 'francesca.ferrari', 'passwordFrancesca'),
28   ('Renato', 'Cicala', 'renato.cicala', 'passwordRenato'),
29   ('Davide', 'Buonauro', 'davide.buonauro', 'passwordDavide'),
30   ('Davide', 'Esposito', 'davide.esposito', 'passwordDavide'),
31   ('Morgan', 'De Sanctis', 'morgan.desantis', 'passwordMorgan'),
32   ('Ezequiel', 'Lavezzi', 'pocho', 'passwordEzequiel'),
33   ('Edinson', 'Cavani', 'matador', 'passwordEdinson'),
34   ('Marek', 'Hamsik', 'marekiaro', 'passwordMarek'),
35   ('Dries', 'Mertens', 'ciro', 'passwordDries'),
36   ('Kalidou', 'Koulibaly', 'kalidou.koulibaly', 'passwordKalidou'),
37   ('Kvicha', 'Kvaratskhelia', 'kvaradona', 'passwordKvicha'),
38   ('Donkey', 'Kong', 'donkey.kong', 'passwordDonkey'),
39   ('Diddy', 'Kong', 'diddy.kong', 'passwordDiddy'),
40   ('Cranky', 'Kong', 'cranky.kong', 'passwordCranky'),
41   ('Dixie', 'Kong', 'dixie.kong', 'passwordDixie')

```

```

42 ('Funky', 'Kong', 'funky.kong', 'passwordFunky'),
43 ('Akira', 'Toriyama', 'akira.toriyama', 'passwordAkira'),
44 ('Goku', 'Son', 'goku.son', 'passwordGoku'),
45 ('Bulma', 'Briefs', 'bulma.briefs', 'passwordBulma'),
46 ('Trunks', 'Briefs', 'trunks.briefs', 'passwordTrunks'),
47 ('Gohan', 'Son', 'gohan.son', 'passwordGohan'),
48 ('Goten', 'Son', 'goten.son', 'passwordGoten'),
49 ('Buu', 'Majin', 'majin.buu', 'passwordBuu'),
50 ('Dio', 'Brando', 'dio.brando', 'passwordDio'),
51 ('Joseph', 'Joestar', 'joseph.joestar', 'passwordJoseph'),
52 ('Jotaro', 'Kujo', 'jotaro.kujo', 'passwordJotaro'),
53 ('Giorno', 'Giovanna', 'giorno.giovanna', 'passwordGiorno'),
54 ('Josuke', 'Higashikata', 'josuke.higashikata', 'passwordJosuke'),
55 ('Jolyne', 'Cujoh', 'jolyne.cujoh', 'passwordJolyne'),
56 ('Enrico', 'Pucci', 'enrico.pucci', 'passwordEnrico'),
57 ('William', 'Zeppeli', 'william.zeppeli', 'passwordWilliam'),
58 ('Caesar', 'Zeppeli', 'caesar.zeppeli', 'passwordCaesar'),
59 ('Rohan', 'Kishibe', 'rohan.kishibe', 'passwordRohan'),
60 ('Suzy', 'Q', 'suzy.q', 'passwordSuzy'),
61 ('Yasuho', 'Hirose', 'yasuho.hirose', 'passwordYasuho'),
62 ('Kira', 'Yoshikage', 'kira.yoshikage', 'passwordKira'),
63 ('Okuyasu', 'Nijimura', 'okuyasu.nijimura', 'passwordOkuyasu'),
64 ('KNOW', 'PMW', 'know.pmw', 'passwordKNOW'),
65 ('Kirua', 'THB', 'kirua.thb', 'PasswordKirua'),
66 ('Brown', 'THB', 'brown.thb', 'PasswordBrown'),
67 ('Gon', 'Freecss', 'gon.freecss', 'PasswordGon'),
68 ('Killua', 'Zoldyck', 'killua.zoldyck', 'PasswordKillua'),
69 ('Salvatore', 'Esposito', 'salvatore.esposito', 'passwordSalvatore'),
70 ('Marco', 'D'Amore', 'marco.damore', 'passwordMarco'),
71 ('Gennaro', 'Savastano', 'gennaro.savastano', 'passwordGennaro'),
72 ('Ciro', 'Di Marzio', 'ciro.dimazio', 'passwordCiro')

```

Listing 5.2: Query di inserimento degli utenti

Questa query inserisce gli utenti nel database, specificando il loro username, nome, cognome e password. Gli utenti sono coloro che si iscrivono ai corsi di cucina.

### 5.1.3 Query di inserimento dei corsi

```

1 INSERT INTO Corso(Titolo, Frequenza, NumLezioni, Anno, DataInizio, IdTopic,
2 UsernameChef)
VALUES
3   ('Corso di cucina italiana', 'Ogni giorno', 100, 2025, '2025-09-08',
4     (SELECT idTopic FROM Topic WHERE Nome = 'Cucina Italiana'),
5     'KenIlGuerriero'),
6   ('Corso di pizza napoletana', 'Ogni due giorni', 90, 2025, '2025-10-12',
7     (SELECT idTopic FROM Topic WHERE Nome = 'Pizza Napoletana'),
8     'ChefMaria'),
9   ('Corso di cucina cinese', 'Settimanale', 20, 2025, '2025-08-02', (SELECT
10      idTopic FROM Topic WHERE Nome = 'Cucina Cinese'), 'ChefAntonio'),
11   ('Corso di cucina messicana', 'Ogni tre giorni', 20, 2025, '2025-11-24',
12     (SELECT idTopic FROM Topic WHERE Nome = 'Cucina Messicana'),
13     'ChefFrancesca'),

```

```

7   ('Corso di cucina giapponese', 'Settimanale', 30, 2025, '2025-09-10',
8     (SELECT idTopic FROM Topic WHERE Nome = 'Cucina Giapponese'),
9     'ChefLuca'),
10    ('Corso di cucina indiana', 'Settimanale', 25, 2025, '2025-09-15',
11      (SELECT idTopic FROM Topic WHERE Nome = 'Cucina Indiana'),
12      'Il_Nerdone_11'),
13    ('Corso di cucina francese', 'Ogni due giorni', 40, 2025, '2025-12-01',
14      (SELECT idTopic FROM Topic WHERE Nome = 'Cucina Francese'),
15      'ChefElena'),
16    ('Corso di cucina spagnola', 'Ogni tre giorni', 70, 2025, '2025-08-20',
17      (SELECT idTopic FROM Topic WHERE Nome = 'Cucina Spagnola'),
18      'ChefMarco'),
19    ('Corso di cucina americana', 'Ogni tre giorni', 15, 2025, '2025-10-05',
20      (SELECT idTopic FROM Topic WHERE Nome = 'Cucina Americana'),
21      'ChefSara'),
22    ('Corso di cucina mediterranea', 'Ogni giorno', 100, 2025, '2025-09-25',
23      (SELECT idTopic FROM Topic WHERE Nome = 'Cucina Mediterranea'),
24      'FranchinoErCriminale'),
25    ('Corso di pasticceria', 'Ogni giorno', 100, 2025, '2025-09-30', (SELECT
26      idTopic FROM Topic WHERE Nome = 'Pasticceria'), 'ChefGiorgio'),
27    ('Corso di panificazione', 'Ogni due giorni', 90, 2025, '2025-11-15',
28      (SELECT idTopic FROM Topic WHERE Nome = 'Panificazione'),
29      'ChefValentina');

```

Listing 5.3: Query di inserimento dei corsi

Questa query inserisce i corsi nel database, specificando il titolo, la frequenza, il numero di lezioni, la data di inizio, il topic e lo chef che insegna il corso. I corsi sono le categorie tematiche dei corsi di cucina.

#### 5.1.4 Query di inserimento delle sessioni in presenza

```

1 INSERT INTO SessioneInPresenza (Luogo, Aula, Data, Durata, NumSessione,
2 IdCorso) VALUES
3 ('MonteSantangelo', 'Aula1', '2025-10-01 10:00:00', 120, 1, (SELECT idCorso
4   FROM Corso WHERE Titolo = 'Corso di cucina italiana')),
5 ('ViaClaudio', 'Aula2', '2025-10-02 11:00:00', 90, 1, (SELECT idCorso FROM
6   Corso WHERE Titolo = 'Corso di cucina messicana')),
7 ('PiazzaleTecchio', 'Aula3', '2025-10-03 12:00:00', 150, 1, (SELECT idCorso
8   FROM Corso WHERE Titolo = 'Corso di cucina giapponese')),
9 ('PortaDiMassa', 'Aula4', '2025-10-04 13:00:00', 180, 1, (SELECT idCorso FROM
10  Corso WHERE Titolo = 'Corso di cucina francese')),
11 ('ViaMarina', 'Aula5', '2025-10-05 14:00:00', 75, 1, (SELECT idCorso FROM
12  Corso WHERE Titolo = 'Corso di cucina spagnola')),
13 ('MonteSantangelo', 'Aula1', '2025-10-02 10:00:00', 120, 2, (SELECT idCorso
14   FROM Corso WHERE Titolo = 'Corso di cucina italiana')));

```

Listing 5.4: Query di inserimento delle sessioni in presenza

Questa query inserisce le sessioni in presenza nel database, specificando il luogo, l'aula, la data, la durata, il numero di sessione e il corso a cui appartiene. Le sessioni in presenza sono le sessioni pratiche dei corsi di cucina.

### 5.1.5 Query di inserimento delle sessioni online

```
1 INSERT INTO SessioneOnline (Link , Data , Durata , NumSessione , IdCorso) VALUES
2 ('https://example.com/session1' , '2025-10-03 10:00:00' , 120 , 3 , (SELECT
3     idCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana')),
4 ('https://example.com/session2' , '2025-10-04 11:00:00' , 90 , 4 , (SELECT
5     idCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana')),
6 ('https://example.com/session3' , '2025-11-03 12:00:00' , 150 , 1 , (SELECT
7     idCorso FROM Corso WHERE Titolo = 'Corso di cucina americana')),
8 ('https://example.com/session4' , '2025-11-04 13:00:00' , 180 , 1 , (SELECT
9     idCorso FROM Corso WHERE Titolo = 'Corso di cucina cinese')),
10 ('https://example.com/session5' , '2025-09-05 14:00:00' , 75 , 1 , (SELECT
11     idCorso FROM Corso WHERE Titolo = 'Corso di cucina indiana'));
```

Listing 5.5: Query di inserimento delle sessioni online

Questa query inserisce le sessioni online nel database, specificando il link, la data, la durata, il numero di sessione e il corso a cui appartiene. Le sessioni online sono le sessioni teoriche dei corsi di cucina.

### 5.1.6 Query di inserimento delle adesioni

```
1 INSERT INTO Adesione (UsernameUtente , IdSessionePresenza) VALUES
2 ('mario.bros' , (SELECT IdSessione FROM SessioneInPresenza WHERE Luogo =
3     'MonteSantangelo' AND Aula = 'Aula1' AND Data = '2025-10-01 10:00:00')),
4 ('francesco.deluca' , (SELECT IdSessione FROM SessioneInPresenza WHERE Luogo =
5     'ViaClaudio' AND Aula = 'Aula2' AND Data = '2025-10-02 11:00:00')),
6 ('kalidou.koulibaly' , (SELECT IdSessione FROM SessioneInPresenza WHERE Luogo =
7     'PiazzaleTecchio' AND Aula = 'Aula3' AND Data = '2025-10-03 12:00:00')),
8 ('rohan.kishibe' , (SELECT IdSessione FROM SessioneInPresenza WHERE Luogo =
9     'PortaDiMassa' AND Aula = 'Aula4' AND Data = '2025-10-04 13:00:00')),
10 ('donkey.kong' , (SELECT IdSessione FROM SessioneInPresenza WHERE Luogo =
11     'ViaMarina' AND Aula = 'Aula5' AND Data = '2025-10-05 14:00:00')),
12 ('goku.son' , (SELECT IdSessione FROM SessioneInPresenza WHERE Luogo =
13     'MonteSantangelo' AND Aula = 'Aula1' AND Data = '2025-10-04 10:00:00'));
```

Listing 5.6: Query di inserimento delle adesioni

Questa query inserisce le adesioni degli utenti alle sessioni in presenza dei corsi di cucina, specificando l'utente e la sessione in presenza a cui si riferiscono.

### 5.1.7 Query di inserimento delle ricette

```
1 INSERT INTO Ricetta(Nome , Descrizione)
2 VALUES
3     --Cucina Italiana
4     ('Spaghetti Aglio Olio e Peperoncino' , 'Spaghetti con aglio , olio
5         d'oliva e peperoncino.' ),
6     ('Pasta e Patate' , 'Pasta con patate e un soffritto di altre
7         verdure.' ),
8     ('Carbonara' , 'Pasta con crema di uovo pastorizzato , pecorino romano e
9         pepe condita con del guanciale croccante.' ),
```

7 ('Pesto alla Genovese', 'Un pesto di basilico fresco, pinoli e  
8      parmigiano. Utilizzato come condimento per pasta o bruschette.'),  
9 ('Polpette al Sugo', 'Polpette di carne e mollica di pane bagnata  
10     cotte in un sugo di pomodoro.'),  
11 ('Parmigiana di Melanzane', 'Preparazione in teglia a base di  
12     melanzane fritte, salsa di pomodoro e mozzarella e parmigiano.'),  
13 ('Arancini di Riso', 'Sfere di riso risottato allo zafferano con  
14     diverse possibilità di ripieno, impanate e fritte.'),  
15 ('Caponata', 'Stufato di melanzane, pomodori, sedano, cipolle e  
16     olive.'),  
17 ('Cacciucco', 'Zuppa a base di pesce fresco, crostacei e molluschi.'),  
18 ('Zuppa di Cozze', 'Cozze, crostacei, polpo e molluschi cucinate nella  
19     propria acqua e pomodorini, servita su crostini di pane e condita  
20     con olio piccante. Tradizionalmente mangiata il giovedì Santo.'),  
21 ('Pasta alla Norma', 'Pasta con pomodoro, melanzane fritte e ricotta  
22     infornata.'),  
23 ('Pasta alla Puttanesca', 'Pasta con un sugo di pomodoro, capperi e  
24     olive nere.'),  
25 ('Pasta al Forno', 'Ricetta di recupero, si tratta di pasta avanzata,  
26     condita a piacere e ripassata in forno.'),  
27 ('Ragù alla Bolognese', 'Sugo di pomodoro con carne macinata e un  
28     soffritto di varie verdure.'),  
29 ('Ragù Napoletano', 'Sugo di pomodoro con carne di manzo dalla lunga  
30     cottura.'),  
31 ('Sartù di Riso', 'Riso condito con ragù di carne, piselli e  
32     mozzarella, cotto in forno.'),  
33 ('Risotto ai Funghi Porcini', 'Un risotto cremoso con funghi porcini  
34     freschi.'),  
35 ('Risotto alla Milanese', 'Risotto con zafferano e brodo di carne.'),  
36 ('Risotto al Nero di Seppia', 'Risotto con seppie e il loro  
   inchiostro.'),  
37 ('Spaghetti alle Vongole', 'Spaghetti con vongole veraci, aglio e  
38     prezzemolo.'),  
39 ('Spaghetti con le Sarde', 'Spaghetti con sarde fresche, finocchietto  
40     selvatico, pinoli, olive nere e pangrattato'),  
41 ('Sarde a Beccafico', 'Sarde farcite con pangrattato, pinoli e uvetta,  
42     arrotolate e cotte in forno.'),  
43 ('Cotoletta alla Milanese', 'Fetta di lombo di vitello impanata e  
44     fritta in burro.'),  
45 ('Fritto Misto', 'Frittura di pesce e verdure.'),  
46 ('Insalata di Polpo', 'Insalata di polpo con patate, prezzemolo e olio  
47     d'oliva.'),  
48 ('Orata al Sale', 'Orata cotta in crosta di sale.'),  
49 ('Lasagna alla Bolognese', 'Strati di pasta alternati a strati di ragù  
50     di carne e besciamella.'),  
51 --Pizza Napoletana  
52 ('Pizza Margherita', 'Pizza con base pomodoro condita con mozzarella e  
53     basilico.'),  
54 ('Pizza Romana', 'Pizza con base pomodoro condita con acciughe e  
55     capperi e olive.'),  
56 ('Pizza Ortolana', 'Pizza con base Fior di Latte condita con diversi  
57     ortaggi di stagione.'),  
58 ('Pizza Diavola', 'Pizza con base pomodoro condita con salame piccante  
59     e peperoncino.'),  
60 ('Pizza Capricciosa', 'Pizza con base pomodoro condita con funghi,  
61     carciofini sott'olio, prosciutto cotto e olive.'),

37 ('Pizza Fritta', 'Base di impasto per la pizza condita per metà con  
38 pomodoro, provola, ricotta, salumi e pepe, richiusa su se stessa e  
fritta in olio bollente.'),  
39 ('Montanara', 'Pizza fritta condita con pomodoro, basilico e  
parmigiano.'),  
40 ('Pizza Filetto', 'Pizza con base Fior di Latte condita con pomodorini  
freschi tagliati a filetto e basilico.'),  
41 ('Pizza Salsiccia e Friarielli', 'Pizza con base Fior di Latte condita  
con salsiccia e friarielli.'),  
42 ('Pizza Zeb89', 'La pizza più fiera del mondo.'),  
--Cucina Spagnola  
43 ('Tapas', 'Piccole porzioni di cibo servite come antipasto.'),  
44 ('Tortilla di Patate', 'Frittata di uova e patate cotta in padella.'),  
45 ('Tortilla di Cipolle', 'Frittata di uova e cipolle cotta in  
padella.'),  
46 ('Gazpacho', 'Zuppa fredda a base di pomodori, cetrioli, peperoni e  
cipolla.'),  
47 ('Paella', 'Riso cotto con zafferano e vari ingredienti come pollo,  
coniglio, frutti di mare o verdure.'),  
48 ('Churros', 'Dolce di patata fritto a forma di bastoncino, spesso  
servito con cioccolata calda.'),  
49 ('Polpo a la Gallega', 'Polpo lessato e condito con paprika, olio  
d'oliva e sale grosso.'),  
50 ('Crema Catalana', 'Dolce al cucchiaio a base di crema pasticcera con  
una crosta di zucchero caramellato.'),  
51 ('Empanada Fritta', 'Pasta con ripieni di vario genere e fritta.'),  
52 ('Empanada al Forno', 'Pasta con ripieni di vario genere e cotta in  
forno.'),  
--Cucina Messicana  
53 ('Chili di Carne', 'Stufato di carne macinata con fagioli, verdure e  
varie spezie e odori.'),  
54 ('Nachos con Formaggio', 'Tortilla chips di mais con formaggio fuso.'),  
55 ('Guacamole', 'Salsa a base di avocado, lime, cipolla e pomodoro.'),  
56 ('Tacos', 'Pasta di mais a forma di conchiglia, ripiena di chili,  
verdure e formaggio.'),  
57 ('Burrito', 'Sfoglia di pasta simile alle piadine ripiena di chili e  
formaggio, avvolta su se stessa e piastrata.'),  
58 ('Quesadilla', 'Tortilla di mais ripiena di formaggio e altri  
ingredienti, cotta in padella.'),  
--Cucina Giapponese  
59 ('Nigiri', 'Riso sushi servito con pesce crudo posato sopra.'),  
60 ('Sashimi', 'Fette di pesce crudo.'),  
61 ('Maki', 'Riso sushi avvolto in alga nori con ripieno di pesce e  
verdure.'),  
62 ('Uramaki', 'Ripieno di pesce e verdure avvolto in alga nori, a sua  
volta avvolta in riso sushi'),  
63 ('Tempura', 'Verdure o pesce fritti in una pastella leggera.'),  
64 ('Gyoza', 'Ravioli giapponesi ripieni di carne e verdure, cotti al  
vapore o alla piastra.'),  
65 ('Onigiri', 'Polpette di riso ripiene di pesce o verdure, avvolte in  
alga nori.'),  
66 ('Ramen di Carne', 'Udon in brodo di carne con carne di maiale, uova e  
verdure'),  
67 ('Ramen di Pesce', 'Udon in brodo di pesce con pesce, uova e  
verdure.'),

```

70 ('Soba', 'Noodle giapponesi a base di grano saraceno da intingere in
71     salsa tsuyu, serviti freddi o caldi.'),  

72 ('Tonkatsu', 'Cotoletta di maiale impanata e fritta, servita con salsa
73     tonkatsu.'),  

74 ('Takoyaki', 'Polpette di pastella ripiene di polpo, cotte su una
75     piastra speciale.'),  

76 ('Taiyaki', 'Dolce a forma di pesce ripieno di marmellata di fagioli
77     dolci o altre marmellate.'),  

78 ('Mochi', 'Dolce di pasta di riso glutinoso, ripieno di fagioli
79     dolci.'),  

80 --Cucina Cinese  

81 ('Baozi', 'Conosciuti come ravioli cinesi, sono panini al vapore
82     ripieni di carne o verdure.'),  

83 ('Dim Sum', 'Piccole porzioni di cibo servite in cestini di bambù, a
84     base di carne, pesce o verdure.'),  

85 ('Riso Fritto', 'Riso saltato in padella con uova e verdure.'),  

86 ('Involtini Primavera', 'Involtini di sfoglia di riso ripieni di
87     verdure.'),  

88 ('Anatra alla Pechinese', 'Anatra arrosto con pelle croccante'),  

89 ('Pollo Kung Pao', 'Pollo saltato in padella con arachidi, peperoni e
90     salsa di soia.'),  

91 --Cucina Americana  

92 ('Beef Jerky', 'Carne di manzo speziata ed essiccata.'),  

93 ('Hamburger', 'carne macinata di manzo pressata in forma di disco e
94     cotta alla griglia.'),  

95 ('NewYork Style Pizza', 'Pizza molto grande dalla base sottile,
96     servita in grandi fette.'),  

97 ('Chicago Style Pizza', 'Pizza con base spessa, ripiena di formaggio e
98     salsa di pomodoro.'),  

99 ('Alette di Pollo', 'Alette di pollo marinate e condite con salsa
100    barbecue, cotte al forno.'),  

101 ('Pollo Fritto', 'Pollo marinato, impanato e fritto.'),  

102 ('Pulled Pork', 'Carne di maiale speziata e cotta lentamente in
103    affumicatrice.'),  

104 ('Fettuccine Alfredo', 'Pasta condita con burro, panna e parmigiano.'),  

105 ('Brownies', 'Dolce a base di cioccolato e burro di forma quadrata.'),  

106 ('Cheesecake', 'Dolce di formaggio cremoso con base di biscotto.'),  

107 --Cucina Indiana  

108 ('Pollo Tandoori', 'Pollo marinato in yogurt e spezie, cotto in forno
109    tandoor.'),  

110 ('Curry', 'Carne o Verdure cotti in un mix di spezie e latte di
111    cocco.'),  

112 ('Pane Naan', 'Pane indiano cotto in forno tandoor.'),  

113 ('Samosa', 'Involtini di sfoglia sottile di forma triangolare, ripieni
114    di carne o verdure.'),  

115 ('Chicken Tikka', 'Pollo marinato e cotto in forno, alla griglia o al
116    curry.'),  

117 --Cucina Francese  

118 ('Soupe a l''oignon', 'Zuppa a base di cipolle caramellate e brodo,
119     servita con crostini di pane.'),  

120 ('Escargot', 'Lumache cotte in burro all''aglio e prezzemolo.'),  

121 ('Coq au Vin', 'Pollo cotto in vino rosso con funghi, cipolle e
122     pancetta.'),  

123 ('Ratatouille', 'Stufato di verdure estive.'),  

124 ('Boeuf Bourguignon', 'Stufato di manzo cotto in vino rosso con funghi
125     e cipolle.'),
```

```

105 ('Crepes', 'Pasta sottile e morbida, farcita con ingredienti dolci o
106     salati.'),  

107 ('Croissant', 'Dolce di pasta sfoglia a forma di mezzaluna.'),  

108 ('Macarons', 'Dolcetti di pasta di mandorle.'),  

109 ('Omelette', 'Frittata tipica francese dal cuore cremoso.'),  

110 ('Quiche Lorraine', 'Torta salata con uova, panna e pancetta.'),  

111 --Panificazione  

112 ('Pane Bianco', 'Pane bianco con crosta sottile.'),  

113 ('Pane Integrale', 'Pane integrale con crosta spessa.'),  

114 ('Focaccia Ligure', 'Focaccia alta e soffice condita con olio d''oliva
115     e sale.'),  

116 ('Focaccia Barese', 'Focaccia condita con pomodorini, olive e
117     origano.'),  

118 ('Pane di Segale', 'Pane scuro e fitto, fatto con farina di segale.'),  

119 ('Pane Carasau', 'Pane tipico della Sardegna, sottile e croccante.'),  

120 ('Pane Toscano', 'Pane caratterizzato dall''assenza di sale
121     nell''impasto.'),  

122 ('Pane di Altamura', 'Pane tipico della Puglia, fatto con semola di
123     grano duro.'),  

124 ('Baguette', 'Pane lungo e sottile, tipico della cucina francese.'),  

125 ('Ciabatta', 'Pane croccante all''esterno e morbido all''interno,
126     spesso utilizzato per fare panini piastrati.'),  

127 ('Piadina Romagnola', 'Sfoglia di pasta sottile, cotta su una
128     piastra.'),  

129 ('Pizza in Teglia', 'Pizza cotta in teglia e farcita a piacere.'),  

130 ('Casatiello', 'Panificato rustico tipico della Pasqua napoletana,
131     farcito con salumi e formaggi.'),  

132 --Pasticceria  

133 ('Torta di Mele', 'Dolce a base di mele e cannella.'),  

134 ('Torta Caprese', 'Dolce a base di cacao e mandorle, spolverata con
135     zucchero a velo.'),  

136 ('Torta della Nonna', 'Torta di pasta frolla ripiena di crema
137     pasticcera e pinoli.'),  

138 ('Torta di Ciliegie', 'Dolce a base di ciliegie fresche, farina,
139     zucchero e uova.'),  

140 ('Torta Ricotta e Pere', 'Dolce a base di ricotta e pere con una base
     di pasta di nocciola.'),  

141 ('Biscotto all''Amarena', 'Biscotto con crosta di zucchero e ripieno
     di amarene.'),  

142 ('Cantucci', 'Biscotti secchi alle mandorle.'),  

143 ('Torta Sacher', 'Torta a base di cacao e marmellata di albicocche,
     ricoperta di cioccolato fondente temperato.'),  

144 ('Strudel di Mele', 'Dolce a base di mele, uvetta e cannella avvolto
     in una pasta sottile.'),  

145 ('Baklava', 'Dolce a base di pasta fillo, noci e miele.'),  

146 ('Babà', 'Dolce lievitato imbevuto in una bagna di rum.'),  

147 ('Zeppole di San Giuseppe', 'Dolce di pasta choux farcita con crema
     pasticcera e amarene.'),  

148 ('Sfogliatella Riccia', 'Dolce di pasta sfoglia ripieno di ricotta,
     semola e canditi.'),  

149 ('Sfogliatella Frolla', 'Dolce di pasta frolla ripieno di ricotta,
     semola e canditi.'),  

150 ('Pastiera Napoletana', 'Dolce pasquale a base di grano e ricotta.'),  

151 ('Tiramisu', 'Dolce composto da strati di savoiardi inzuppati nel
     caffè e strati di mascarpone con spolverata finale di cacao.'),
```

```

141     ('Creme brûlée', 'Dolce al cucchiaio a base di crema pasticcera con
142         una crosta di zucchero caramellato.),
143     ('Gelato', 'Dolce freddo a base di latte, proponibile in diversi
144         gusti.),
145     ('Sorbetto', 'Dolce freddo a base di frutta e ghiaccio.),
146     ('Crostata di Frutta', 'Dolce a base di pasta frolla ripiena di crema
147         pasticcera e coperta di frutta fresca e gelatina.),
148     ('Torrone', 'Dolce a base di mandorle e miele, preparato nel periodo
149         della Commemorazione dei Defunti.),
150     ('Panettone', 'Dolce natalizio a base di farina, uova, burro e
151         canditi.),
152     ('Pandoro', 'Dolce natalizio a forma di stella, alto e soffice,
153         spolverato con zucchero a velo.');
```

Listing 5.7: Query di inserimento delle ricette

Questa query inserisce le ricette nel database, specificando il nome e la descrizione della ricetta. Le ricette sono associate alle sessioni in presenza dei corsi di cucina.

### 5.1.8 Query di inserimento degli ingredienti

```

1 INSERT INTO Ingrediente(Nome, Allergene, UnitàMisura)
2 VALUES
3     -- Pasta e cereali
4     ('Spaghetti', true, 'g'),
5     ('Penne', true, 'g'),
6     ('Rigatoni', true, 'g'),
7     ('Linguine', true, 'g'),
8     ('Fusilli', true, 'g'),
9     ('Paccheri', true, 'g'),
10    ('Sfoglia per lasagne', true, 'g'),
11    ('Riso Carnaroli', false, 'g'),
12    ('Riso basmati', false, 'g'),
13    ('Udon', true, 'g'),
14    ('Noodles', true, 'g'),
15    ('Soba', true, 'g'),
16    ('Farina 00', true, 'g'),
17    ('Farina integrale', true, 'g'),
18    ('Farina di segale', true, 'g'),
19    ('Semola di grano duro', true, 'g'),
20    ('Grano cotto', true, 'g'),
21
22    -- Verdure e ortaggi
23    ('Aglio', false, 'spicchi'),
24    ('Cipolla bianca', false, 'g'),
25    ('Cipolla rossa', false, 'g'),
26    ('Cipollotto', false, 'g'),
27    ('Scalogno', false, 'g'),
28    ('Sedano', false, 'g'),
29    ('Carote', false, 'g'),
30    ('Cavolo', false, 'g'),
31    ('Patate', false, 'g'),
32    ('Pomodori San Marzano Pelati', false, 'g'),
33    ('Pomodorini ciliegino', false, 'g'),
```

```

34 ('Passata di pomodoro', false, 'ml'),
35 ('Concentrato di pomodoro', false, 'g'),
36 ('Melanzane', false, 'g'),
37 ('Zucchine', false, 'g'),
38 ('Peperoni', false, 'g'),
39 ('Peperoncino', false, 'pz'),
40 ('Basilico', false, 'foglie'),
41 ('Prezzemolo', false, 'g'),
42 ('Rosmarino', false, 'g'),
43 ('Origano', false, 'g'),
44 ('Timo', false, 'g'),
45 ('Finocchietto selvatico', false, 'g'),
46 ('Friarielli', false, 'g'),
47 ('Spinaci', false, 'g'),
48 ('Rucola', false, 'g'),
49 ('Lattuga', false, 'g'),
50 ('Cetrioli', false, 'g'),
51 ('Avocado', false, 'g'),
52 ('Piselli', false, 'g'),
53 ('Fagioli cannellini', false, 'g'),
54 ('Fagioli borlotti', false, 'g'),
55 ('Fagioli neri', false, 'g'),
56 ('Ceci', false, 'g'),
57 ('Lenticchie', false, 'g'),
58 ('Olive nere', false, 'g'),
59 ('Olive verdi', false, 'g'),
60 ('Capperi sotto sale', false, 'g'),
61
62 -- Funghi
63 ('Funghi porcini', false, 'g'),
64 ('Champignon', false, 'g'),
65 ('Funghi shiitake', false, 'g'),
66 ('Fungo Rosso', false, 'pz'),
67 ('Fungo Chiodino', false, 'g'),
68 ('Tartufo nero', false, 'g'),
69
70 -- Pesce
71 ('Salmone', true, 'g'),
72 ('Tonno', true, 'g'),
73 ('Orata', true, 'g'),
74 ('Branzino', true, 'g'),
75 ('Sogliola', true, 'g'),
76 ('Merluzzo', true, 'g'),
77 ('Sarde', true, 'g'),
78 ('Acciughe', true, 'g'),
79 ('Acciughe sott'olio', true, 'g'),
80 ('Sgombro', true, 'g'),
81 ('Rana pescatrice', true, 'g'),
82 ('Dentice', true, 'g'),
83 ('San Pietro', true, 'g'),
84
85 -- Molluschi e crostacei
86 ('Vongole veraci', true, 'g'),
87 ('Cozze', true, 'g'),
88 ('Telline', true, 'g'),
89 ('Seppie', true, 'g'),

```

```

90 ('Calamari', true, 'g'),
91 ('Polpo', true, 'g'),
92 ('Moscardini', true, 'g'),
93 ('Gamberi', true, 'g'),
94 ('Gamberetti', true, 'g'),
95 ('Scampi', true, 'g'),
96 ('Granchi', true, 'g'),
97 ('Astice', true, 'g'),
98 ('Aragosta', true, 'g'),
99 ('Inchiostro di seppia', true, 'ml'),

100
101 -- Carni
102 ('Manzo macinato', false, 'g'),
103 ('Vitello', false, 'g'),
104 ('Lombo di vitello', false, 'g'),
105 ('Braciole di maiale', false, 'g'),
106 ('Lonza di maiale', false, 'g'),
107 ('Pancetta', false, 'g'),
108 ('Guanciale', false, 'g'),
109 ('Speck', false, 'g'),
110 ('Prosciutto crudo', false, 'g'),
111 ('Prosciutto cotto', false, 'g'),
112 ('Salame piccante', false, 'g'),
113 ('Salsiccia', false, 'g'),
114 ('Pollo intero', false, 'g'),
115 ('Petto di pollo', false, 'g'),
116 ('Cosce di pollo', false, 'g'),
117 ('Alette di pollo', false, 'g'),
118 ('Anatra', false, 'g'),
119 ('Coniglio', false, 'g'),
120 ('Agnello', false, 'g'),
121
122 -- Latticini e formaggi
123 ('Mozzarella di bufala', true, 'g'),
124 ('Fior di latte', true, 'g'),
125 ('Provola', true, 'g'),
126 ('Parmigiano Reggiano', true, 'g'),
127 ('Pecorino Romano', true, 'g'),
128 ('Ricotta fresca', true, 'g'),
129 ('Ricotta infornata', true, 'g'),
130 ('Mascarpone', true, 'g'),
131 ('Formaggio Spalmabile', true, 'g'),
132 ('Formaggio cheddar', true, 'g'),
133 ('Latte intero', true, 'ml'),
134 ('Panna fresca', true, 'ml'),
135 ('Panna da cucina', true, 'ml'),
136 ('Burro', true, 'g'),
137 ('Yogurt greco', true, 'ml'),
138
139 -- Uova
140 ('Uova di gallina', true, 'pz'),
141 ('Tuorli d'uovo', true, 'pz'),
142 ('Albumi', true, 'pz'),
143
144 -- Oli e condimenti
145 ('Olio extravergine d'oliva', false, 'ml'),

```

```

146 ('Olio di semi', false, 'ml'),
147 ('Aceto balsamico', false, 'ml'),
148 ('Aceto di vino bianco', false, 'ml'),
149 ('Sale fino', false, 'g'),
150 ('Sale grosso', false, 'g'),
151 ('Pepe nero', false, 'g'),
152 ('Pepe bianco', false, 'g'),
153 ('Zafferano', false, 'g'),
154 ('Paprika dolce', false, 'g'),
155 ('Paprika piccante', false, 'g'),
156 ('Curry in polvere', false, 'g'),
157 ('Curcuma', false, 'g'),
158 ('Cannella', false, 'g'),
159 ('Noce moscata', false, 'g'),
160
161     -- Frutta secca e semi
162 ('Pinoli', true, 'g'),
163 ('Mandorle', true, 'g'),
164 ('Noci', true, 'g'),
165 ('Nocciole', true, 'g'),
166 ('Arachidi', true, 'g'),
167 ('Uvetta', false, 'g'),
168
169     -- Frutta fresca
170 ('Limoni', false, 'pz'),
171 ('Lime', false, 'pz'),
172 ('Arance', false, 'pz'),
173 ('Mele', false, 'g'),
174 ('Pere', false, 'g'),
175 ('Ciliegie', false, 'g'),
176 ('Amarene', false, 'g'),
177 ('Fragole', false, 'g'),
178 ('Mirtilli', false, 'g'),
179 ('Lamponi', false, 'g'),
180 ('Ananas', false, 'g'),
181
182     -- Pane e panificati
183 ('Pane bianco', true, 'g'),
184 ('Mollica di pane', true, 'g'),
185 ('Pangrattato', true, 'g'),
186 ('Crostini', true, 'pz'),
187 ('Pane per hamburger', true, 'pz'),
188 ('Tortilla di mais', true, 'pz'),
189 ('Tortilla di grano', true, 'pz'),
190 ('Piadina', true, 'pz'),
191
192     -- Brodi e salse
193 ('Brodo di carne', false, 'ml'),
194 ('Brodo di pollo', false, 'ml'),
195 ('Brodo vegetale', false, 'ml'),
196 ('Brodo di pesce', true, 'ml'),
197 ('Besciamella', true, 'ml'),
198 ('Ragù bolognese', false, 'ml'),
199 ('Salsa barbecue', false, 'ml'),
200 ('Salsa worcestershire', true, 'ml'),
201 ('Salsa piccante', false, 'ml'),

```

```

202 ('Maionese', true, 'ml'),
203 ('Ketchup', false, 'ml'),
204 ('Senape', false, 'ml'),
205
206     -- Ingredienti asiatici
207 ('Alga nori', false, 'fogli'),
208 ('Alga wakame', false, 'g'),
209 ('Miso', true, 'g'),
210 ('Mirin', false, 'ml'),
211 ('Sake', false, 'ml'),
212 ('Zenzero fresco', false, 'g'),
213 ('Wasabi', false, 'g'),
214 ('Latte di cocco', false, 'ml'),
215 ('Salsa di soia', true, 'ml'),
216 ('Salsa teriyaki', true, 'ml'),
217 ('Salsa tsuyu', true, 'ml'),
218 ('Salsa tonkatsu', true, 'ml'),
219 ('Tempura mix', true, 'g'),
220
221     -- Dolci e pasticceria
222 ('Zucchero', false, 'g'),
223 ('Zucchero di canna', false, 'g'),
224 ('Zucchero a velo', false, 'g'),
225 ('Miele', false, 'g'),
226 ('Sciroppto d'acero', false, 'ml'),
227 ('Cacao amaro', false, 'g'),
228 ('Cioccolato fondente', true, 'g'),
229 ('Cioccolato al latte', true, 'g'),
230 ('Cioccolato bianco', true, 'g'),
231 ('Nutella', true, 'g'),
232 ('Marmellata', false, 'g'),
233 ('Confettura di albicocche', false, 'g'),
234 ('Savoiardi', true, 'g'),
235 ('Biscotti', true, 'g'),
236 ('Caffè espresso', false, 'ml'),
237 ('Liquore al caffè', false, 'ml'),
238 ('Rum', false, 'ml'),
239 ('Marsala', false, 'ml'),
240 ('Amaretto', false, 'ml'),
241 ('Crema pasticcera', true, 'ml'),
242 ('Gelatina', false, 'g'),
243 ('Lievito per dolci', false, 'g'),
244 ('Lievito di birra', false, 'g'),
245 ('Bicarbonato', false, 'g'),
246 ('Vaniglia', false, 'g'),
247 ('Estratto di vaniglia', false, 'ml'),
248 ('Canditi', false, 'g'),
249
250     -- Vari
251 ('Ghiaccio', false, 'g'),
252 ('Acqua', false, 'ml'),
253 ('Vino bianco', false, 'ml'),
254 ('Vino rosso', false, 'ml'),
255 ('Birra', false, 'ml'),
256 ('Carciofini sott'olio', false, 'g'),
257 ('Pomodori secchi', false, 'g'),

```

```
258 ('Tortilla chips', true, 'g');
```

Listing 5.8: Query di inserimento degli ingredienti

Questa query inserisce gli ingredienti nel database, specificando il nome, l'unità di misura e se è un allergene.

### 5.1.9 Query di inserimento della composizione delle ricette

```
1 -- Cucina Italiana
2
3 -- Spaghetti Aglio Olio e Peperoncino
4 INSERT INTO Composizione (IdIngrediente, IdRicetta, Quantita)
5 VALUES
6   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Spaghetti'),
7    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Spaghetti Aglio Olio e
8     Peperoncino'), 100),
9   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Aglio'),
10    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Spaghetti Aglio Olio e
11     Peperoncino'), 3),
12   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Olio extravergine
13     d''oliva'),
14    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Spaghetti Aglio Olio e
15     Peperoncino'), 40),
16   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Peperoncino'),
17    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Spaghetti Aglio Olio e
18     Peperoncino'), 1),
19   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Prezzemolo'),
20    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Spaghetti Aglio Olio e
21     Peperoncino'), 10);
22
23 -- Pasta e Patate
24 INSERT INTO Composizione (IdIngrediente, IdRicetta, Quantita)
25 VALUES
26   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Rigatoni'),
27    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Pasta e Patate'), 80),
28   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Patate'),
29    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Pasta e Patate'), 200),
30   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Sedano'),
31    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Pasta e Patate'), 50),
32   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Carote'),
33    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Pasta e Patate'), 50),
34   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Cipolla bianca'),
35    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Pasta e Patate'), 60),
36   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Concentrato di
37     pomodoro'),
38    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Pasta e Patate'), 20);
39
40 -- Carbonara
41 INSERT INTO Composizione (IdIngrediente, IdRicetta, Quantita)
42 VALUES
43   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Spaghetti'),
44    (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Carbonara'), 100),
45   ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Guanciale'),
```

```

39   (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Carbonara'), 50),
40 ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Uova di gallina'),
41  (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Carbonara'), 2),
42 ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Pecorino Romano'),
43  (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Carbonara'), 40),
44 ((SELECT IdIngrediente FROM Ingrediente WHERE Nome = 'Pepe nero'),
45  (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Carbonara'), 2);

```

Listing 5.9: Query di inserimento della composizione delle ricette

Questa query inserisce la composizione delle ricette nel database, specificando la ricetta, l'ingrediente e la quantità necessaria per ciascuno di essi. Per brevità abbiamo omesso parte del codice, dato che raggiunge 2500 righe.

### 5.1.10 Query di inserimento dei topic

```

1 INSERT INTO Topic(Nome, Descrizione)
2 VALUES ('Cucina Italiana', 'La cucina italiana è famosa in tutto il mondo ed è
3 una delle più apprezzate. Caratterizzata da ingredienti di alta qualità
4 e territoriali, offre piatti salati come pasta, risotti e lasagne ma
5 anche piatti dolci come l'iconico tiramisù.'),
6 ('Pizza Napoletana', 'La pizza napoletana è un simbolo della cucina
7 italiana, ingredienti freschi come pomodoro San Marzano e
8 mozzarella di bufala rendono queste preparazioni gustose e
9 uniche. '),
10 ('Cucina Cinese', 'La cucina cinese è ricca di sapori e varietà, con
11 piatti come il riso fritto, il dim sum e i famosissimi baozi. '),
12 ('Cucina Messicana', 'La cucina messicana è nota per i suoi sapori
13 piccanti, i suoi piatti a base di carne e l'uso di ingredienti
14 come mais, fagioli e peperoncini. '),
15 ('Cucina Giapponese', 'La cucina giapponese è famosa per la sua
16 attenzione alla presentazione ed i suoi sapori complessi
17 influenzati da diverse cucine, dalla europea alla più vicina
18 cinese. '),
19 ('Cucina Indiana', 'La cucina indiana è caratterizzata da spezie
20 aromatiche e piatti come curry, naan e biryani. '),
21 ('Cucina Francese', 'La cucina francese è rinomata per la sua eleganza
22 e varietà, con piatti iconici come coq au vin e ratatouille. '),
23 ('Cucina Spagnola', 'La cucina spagnola è conosciuta per le tapas, la
24 paella oltre ad una grande tradizione di salumi. '),
25 ('Cucina Americana', 'La cucina americana offre una vasta gamma di
26 piatti di provenienza e derivazione diverse dovuta alla
27 multietnicità degli USA, dagli hamburger alla New York style pizza
28 e alla cucina Tex-Mex. '),
29 ('Cucina Mediterranea', 'La cucina mediterranea si concentra su
30 ingredienti freschi come pesce e verdure, ed è nota per i suoi
31 benefici per la salute. '),
32 ('Pasticceria', 'La pasticceria è la branca della cucina che si occupa
33 della preparazione di dolci. '),
34 ('Panificazione', 'La panificazione è la branca della cucina che si
35 occupa della preparazione dei panificati. ');

```

Listing 5.10: Query di inserimento dei topic

Questa query inserisce i topic nel database, specificando il nome e la descrizione del topic. I topic rappresentano le categorie tematiche dei corsi di cucina.

### 5.1.11 Query di inserimento della tabella insegna

```

1  INSERT INTO Insegna(IdTopic, UsernameChef)
2  VALUES
3      ((SELECT idTopic FROM Topic WHERE Nome = 'Cucina Italiana'),
4       'KenIlGuerriero'),
5      ((SELECT idTopic FROM Topic WHERE Nome = 'Pizza Napoletana'),
6       'ChefMaria'),
7      ((SELECT idTopic FROM Topic WHERE Nome = 'Cucina Cinese'), 'ChefAntonio'),
8      ((SELECT idTopic FROM Topic WHERE Nome = 'Cucina Messicana'),
9       'ChefFrancesca'),
10     ((SELECT idTopic FROM Topic WHERE Nome = 'Cucina Giapponese'),
11      'ChefLuca'),
12     ((SELECT idTopic FROM Topic WHERE Nome = 'Cucina Indiana'),
13      'Il_Nerdone_11'),
14     ((SELECT idTopic FROM Topic WHERE Nome = 'Cucina Francese'), 'ChefElena'),
15     ((SELECT idTopic FROM Topic WHERE Nome = 'Cucina Spagnola'), 'ChefMarco'),
16     ((SELECT idTopic FROM Topic WHERE Nome = 'Cucina Americana'), 'ChefSara'),
17     ((SELECT idTopic FROM Topic WHERE Nome = 'Cucina Mediterranea'),
18      'FranchinoErCriminale'),
19     ((SELECT idTopic FROM Topic WHERE Nome = 'Pasticceria'), 'ChefGiorgio'),
20     ((SELECT idTopic FROM Topic WHERE Nome = 'Panificazione'),
21      'ChefValentina');

```

Listing 5.11: Query di inserimento della tabella insegna

Questa query inserisce i record nella tabella **Insegna**, che rappresenta la relazione tra i topic e gli chef che li insegnano.

### 5.1.12 Query di inserimento della tabella prepara

```

1  INSERT INTO Prepara (IdSessionePresenza, IdRicetta) VALUES
2  ((SELECT IdSessione FROM SessioneInPresenza WHERE Data = '2025-10-01
3   10:00:00' AND IdCorso = (SELECT idCorso FROM Corso WHERE Titolo = 'Corso
4   di cucina italiana')),
5   (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Carbonara')),
6  ((SELECT IdSessione FROM SessioneInPresenza WHERE Data = '2025-10-02
7   10:00:00' AND IdCorso = (SELECT idCorso FROM Corso WHERE Titolo = 'Corso
8   di cucina italiana')),
9   (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Ragù alla Bolognese')),
10  ((SELECT IdSessione FROM SessioneInPresenza WHERE Data = '2025-10-03
11   12:00:00' AND IdCorso = (SELECT idCorso FROM Corso WHERE Titolo = 'Corso
12   di cucina giapponese')),
13   (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Maki')),
14  ((SELECT IdSessione FROM SessioneInPresenza WHERE Data = '2025-10-04
15   13:00:00' AND IdCorso = (SELECT idCorso FROM Corso WHERE Titolo = 'Corso
16   di cucina francese')),
17   (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Ratatouille')),
18  ((SELECT IdSessione FROM SessioneInPresenza WHERE Data = '2025-10-05
19   14:00:00' AND IdCorso = (SELECT idCorso FROM Corso WHERE Titolo = 'Corso
20   di cucina spagnola')));

```

```

11   (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Paella')),
12 ((SELECT IdSessione FROM SessioneInPresenza WHERE Data = '2025-10-02
13   11:00:00' AND IdCorso = (SELECT idCorso FROM Corso WHERE Titolo = 'Corso
di cucina messicana')),
13   (SELECT IdRicetta FROM Ricetta WHERE Nome = 'Tacos'));

```

Listing 5.12: Query di inserimento della tabella prepara

Questa query inserisce i record nella tabella **Prepara**, che rappresenta la relazione tra le sessioni in presenza e le ricette che vengono preparate durante le sessioni.

### 5.1.13 Query di inserimento delle iscrizioni

```

1 INSERT INTO Iscrizione(IdCorso, UsernameUtente)
2 VALUES
3 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di pizza napoletana' AND
4 Anno = 2025), 'Zeb89'),
4 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana' AND
5 Anno = 2025), 'giuliano.laccetti'),
5 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina cinese' AND Anno
= 2025), 'luigi.mario'),
6 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina messicana' AND
7 Anno = 2025), 'mario.mario'),
7 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina giapponese' AND
8 Anno = 2025), 'IL_NERDONE_11'),
8 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina indiana' AND Anno
= 2025), 'bohqualcosaboh'),
9 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina francese' AND
10 Anno = 2025), 'pete_990'),
10 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina spagnola' AND
11 Anno = 2025), 'xFireFox27'),
11 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina americana' AND
12 Anno = 2025), 'Zeb89'),
12 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina mediterranea' AND
13 Anno = 2025), 'antonio.capuozzo'),
13 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di pasticceria' AND Anno =
2025), 'giordano.bruno'),
14 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di panificazione' AND Anno
= 2025), 'francesco.deluca'),
15 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana' AND
16 Anno = 2025), 'alessandro.rossi'),
16 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina cinese' AND Anno
= 2025), 'federico.borghi'),
17 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina messicana' AND
18 Anno = 2025), 'matteo.ruggiero'),
18 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina giapponese' AND
19 Anno = 2025), 'stefano.martini'),
19 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina indiana' AND Anno
= 2025), 'giorgio.bianchi'),
20 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina francese' AND
21 Anno = 2025), 'luca.verdi'),
21 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina spagnola' AND
22 Anno = 2025), 'marco.storti'),
22 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina americana' AND
23 Anno = 2025), 'andrea.simpatico'),
23 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina mediterranea' AND
Anno = 2025), 'elena.scalzi'),

```

```

24 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di pasticceria' AND Anno =
25   2025), 'chiara.gallo'),
26 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di panificazione' AND Anno =
27   2025), 'sara.botticelli'),
28 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana' AND
29   Anno = 2025), 'giulia.manzoni'),
30 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina cinese' AND Anno =
31   2025), 'martina.rinaldi'),
32 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina messicana' AND
33   Anno = 2025), 'alice.gaudino'),
34 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina giapponese' AND
35   Anno = 2025), 'francesca.ferrari'),
36 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina indiana' AND Anno =
37   2025), 'renato.cicala'),
38 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina francese' AND
39   Anno = 2025), 'davide.buonauro'),
40 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina spagnola' AND
41   Anno = 2025), 'davide.esposito'),
42 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina americana' AND
43   Anno = 2025), 'morgan.desantis'),
44 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina mediterranea' AND
45   Anno = 2025), 'pocho'),
46 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di pasticceria' AND Anno =
47   2025), 'matador'),
48 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di panificazione' AND Anno =
49   2025), 'marekiaro'),
50 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana' AND
51   Anno = 2025), 'ciro'),
52 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina cinese' AND Anno =
53   2025), 'kalidou.koulibaly'),
54 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina messicana' AND
55   Anno = 2025), 'kvaradona'),
56 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina giapponese' AND
57   Anno = 2025), 'donkey.kong'),
58 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina indiana' AND Anno =
59   2025), 'diddy.kong'),
60 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina francese' AND
61   Anno = 2025), 'cranky.kong'),
62 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina spagnola' AND
63   Anno = 2025), 'dixie.kong'),
64 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina americana' AND
65   Anno = 2025), 'funky.kong'),
66 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina mediterranea' AND
67   Anno = 2025), 'akira.toriyama'),
68 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana' AND
69   Anno = 2025), 'goku.son'),
70 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di panificazione' AND Anno =
71   2025), 'bulma.briefs'),
72 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana' AND
73   Anno = 2025), 'trunks.briefs'),
74 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina cinese' AND Anno =
75   2025), 'gohan.son'),
76 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina messicana' AND
77   Anno = 2025), 'goten.son'),
78 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina giapponese' AND
79   Anno = 2025), 'majin.buu'),

```

```

52 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina indiana' AND Anno
53   = 2025), 'dio.brando'),
54 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina francese' AND
55   Anno = 2025), 'joseph.joestar'),
56 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina spagnola' AND
57   Anno = 2025), 'jotaro.kujo'),
58 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina americana' AND
59   Anno = 2025), 'giorno.giovanna'),
60 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina mediterranea' AND
61   Anno = 2025), 'josuke.higashikata'),
62 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di pasticceria' AND Anno =
63   2025), 'jolyne.cujoh'),
64 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di panificazione' AND Anno
65   = 2025), 'enrico.pucci'),
66 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana' AND
67   Anno = 2025), 'william.zeppeli'),
68 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina cinese' AND Anno
69   = 2025), 'caesar.zeppeli'),
70 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina francese' AND
71   Anno = 2025), 'rohan.kishibe'),
72 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina giapponese' AND
73   Anno = 2025), 'suzy.q'),
74 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina indiana' AND Anno
75   = 2025), 'yasuho.hirose'),
76 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina francese' AND
77   Anno = 2025), 'kira.yoshikage'),
78 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina spagnola' AND
79   Anno = 2025), 'giorgio.bianchi'),
80 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina americana' AND
81   Anno = 2025), 'luca.verdi'),
82 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina mediterranea' AND
83   Anno = 2025), 'marco.storti'),
84 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di pasticceria' AND Anno =
85   2025), 'andrea.simpatico'),
86 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di panificazione' AND Anno
87   = 2025), 'elena.scalzi'),
88 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana' AND
89   Anno = 2025), 'chiara.gallo'),
90 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina cinese' AND Anno
91   = 2025), 'sara.botticelli'),
92 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina messicana' AND
93   Anno = 2025), 'giulia.manzoni'),
94 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina giapponese' AND
95   Anno = 2025), 'martina.rinaldi'),
96 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina indiana' AND Anno
97   = 2025), 'alice.gaudino'),
98 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina francese' AND
99   Anno = 2025), 'francesca.ferrari'),
100 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina spagnola' AND
101   Anno = 2025), 'renato.cicala'),
102 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina americana' AND
103   Anno = 2025), 'davide.buonaурio'),
104 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina mediterranea' AND
105   Anno = 2025), 'davide.esposito'),
106 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di pasticceria' AND Anno =
107   2025), 'morgan.desantis'),

```

```

80 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di panificazione' AND Anno
81     = 2025), 'pocho'),
82 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana' AND
83     Anno = 2025), 'matador'),
84 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina cinese' AND Anno
85     = 2025), 'marekiaro'),
86 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina messicana' AND
87     Anno = 2025), 'ciro'),
88 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina giapponese' AND
89     Anno = 2025), 'kalidou.koulibaly'),
90 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina indiana' AND Anno
91     = 2025), 'kvaradona'),
92 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina spagnola' AND
93     Anno = 2025), 'donkey.kong'),
94 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina spagnola' AND
95     Anno = 2025), 'diddy.kong'),
96 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina americana' AND
97     Anno = 2025), 'cranky.kong'),
98 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina italiana' AND
99     Anno = 2025), 'mario.mario'),
100 ((SELECT IdCorso FROM Corso WHERE Titolo = 'Corso di cucina messicana' AND
101    Anno = 2025), 'francesco.deluca')

```

Listing 5.13: Query di inserimento delle iscrizioni

Questa query inserisce le iscrizioni degli utenti ai corsi di cucina, specificando l'utente, il corso e la data di iscrizione.