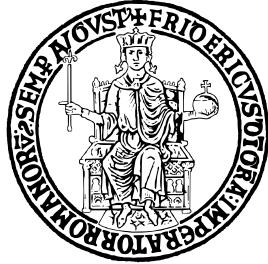


|||||| HEAD

===== 5d67e63b4626e799c8bdb5ab5e8d8e7f73749462

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

Contents

1 Descrizione del progetto	5
1.1 Descrizione sintetica del problema	5
2 Progettazione Concettuale	7
2.1 Introduzione	7
HEAD	
2.2 Diagrammi	7
3 Progettazione Logica	9
3.1 Schema Logico	9
4 Progettazione Fisica	11
4.1 Definizione tabelle	11
4.2 Definizione della tabella TOPIC	11
4.3 Definizione della tabella CHEF	12
4.4 Definizione della tabella UTENTE	12
4.5 Definizione della tabella CORSO	13
4.6 Definizione della tabella SESSIONEINPRESENZA	14
4.7 Definizione della tabella SESSIONEONLINE	14
=====	
2.2 Diagrammi UML e ER	7
2.3 Ristrutturazione del modello concettuale	8
2.4 Identificazione delle chiavi primarie	8
2.5 Identificazione delle chiavi esterne	8
2.6 Rimozione degli attributi multipli	8
2.7 Rimozione delle gerarchie di specializzazione	9
2.8 Analisi delle ridondanze	9
2.9 Class Diagram ristrutturato	9
2.10 Dizionario delle classi	9
2.11 Dizionario delle associazioni	9
2.12 Dizionario dei vincoli	9
3 Progettazione Logica	11
3.1 Schema Logico	11
4 Progettazione Fisica	13
4.1 Definizione tabelle	13
5d67e63b4626e799c8bdb5ab5e8d8e7f73749462	

Chapter 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.

Chapter 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

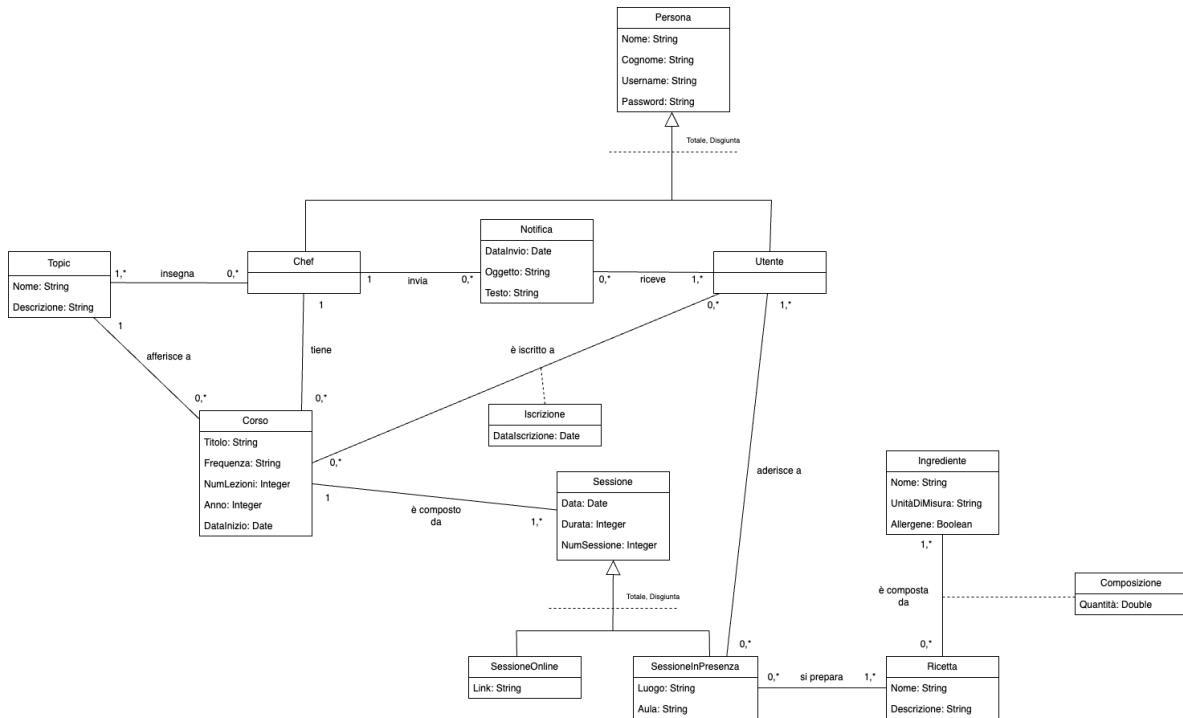


Figure 2.1: Diagramma delle classi UML

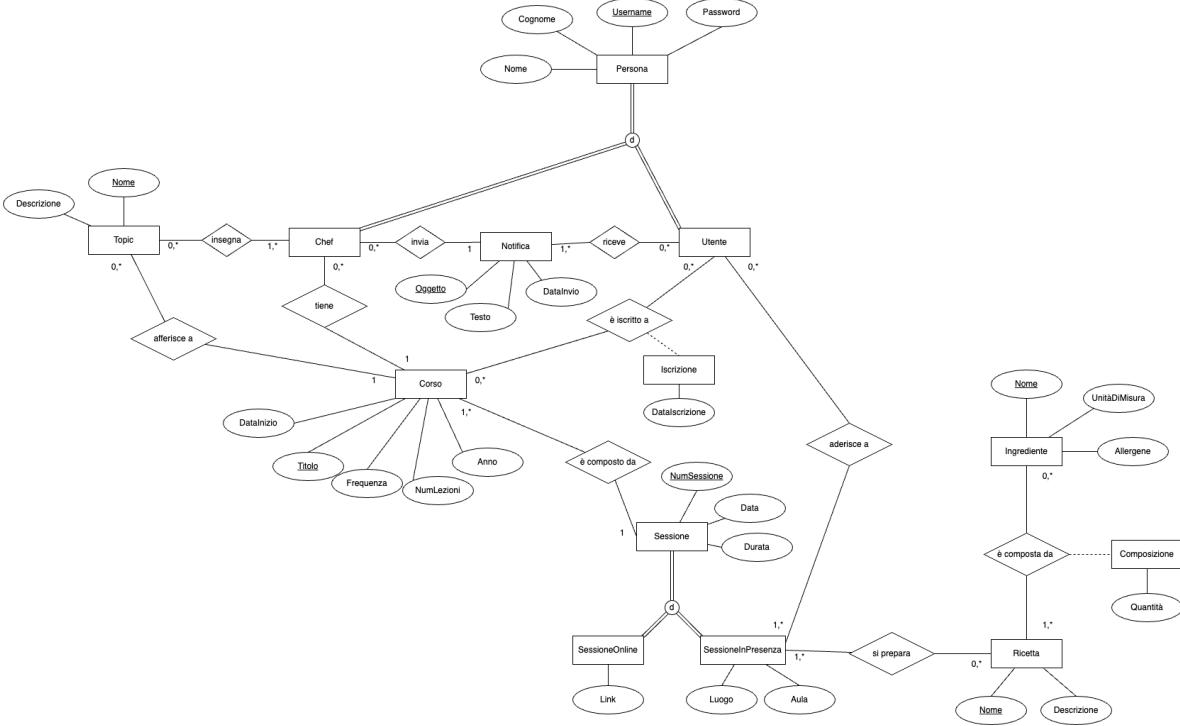


Figure 2.2: Diagramma Entità-Relazioni

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

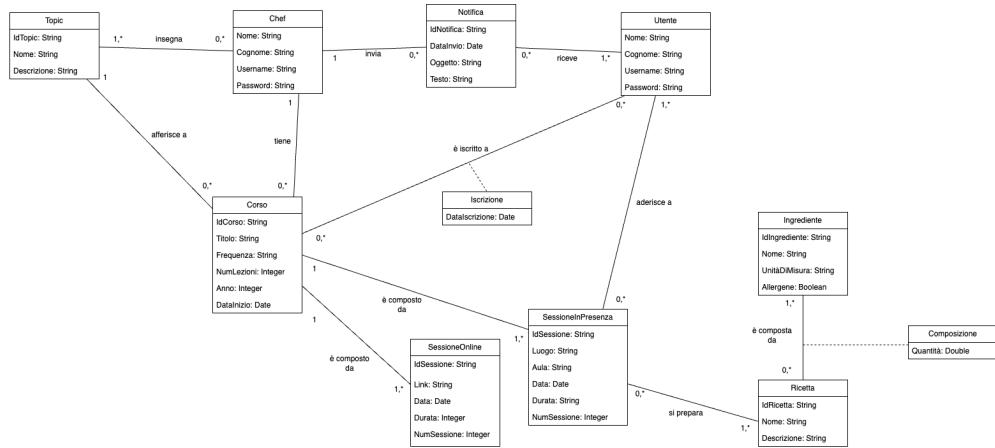


Figure 2.3: Diagramma UML ristrutturato

2.10 Dizionario delle classi

2.11 Dizionario delle associazioni

2.12 Dizionario dei vincoli

Chapter 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)

Chapter 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.

4.2 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.

4.3 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 insegnare i corsi di cucina. Ogni chef ha un identificatore univoco (Username) e le informazioni personali come nome, cognome e password.

4.4 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.

4.5 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.

4.6 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.

4.7 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.