

Progetto di Basi di Dati

Catena di pizzerie

1. Raccolta delle specifiche della realtà d'interesse

Descrizione

Si vuole progettare la base di dati di una catena di pizzerie, che raccoglie le informazioni relative ai camerieri, ai tavoli ed ai pasti consumati nelle singole pizzerie della catena.

Ciascuna pizzeria, di cui interessa conoscere il nome, l'indirizzo, e la città, offre un insieme di beni alimentari. Nello specifico, un ristorante può offrire un insieme di pietanze, contraddistinte da nome, prezzo e contenuto calorico, e un insieme di bevande, ciascuna contraddistinta da nome, prezzo e volume. Inoltre, ciascuna pizzeria carica una cifra fissa (diversa da pizzeria a pizzeria) per ciascun coperto.

I camerieri sono caratterizzati da nome, cognome e da codice fiscale. Un cameriere, in giorni differenti, può lavorare in una sola delle pizzerie della catena, per cui è di interesse memorizzare la data e il numero di ore di lavoro trascorse da un cameriere in una singola pizzeria.

Ciascuna pizzeria è composta da un certo numero di tavoli. I tavoli sono identificati ciascuno da un numero progressivo (dipendente dalla pizzeria in cui sono sistemati), e sono contraddistinti dal numero di posti offerti. Durante una singola giornata di lavoro, un tavolo è assegnato ad un solo cameriere, mentre a ciascun cameriere sono assegnati solitamente più tavoli.

Un pasto è costituito dai cibi e dalle bevande che vengono consumati da un gruppo di clienti seduti ad un certo tavolo. Per ciascun pasto si vogliono conservare tutti i dettagli, quali il numero di commensali, la data in cui è stato consumato, le pietanze e le bevande consumate, specificando per ciascuna la relativa quantità, il tavolo a cui è stato consumato e l'importo totale del conto.

I clienti sono contraddistinti dal nome, dal cognome, dal numero di telefono e dal codice fiscale, e per ciascuno di essi si vuole conservare memoria di tutti i pasti che ha consumato.

Specifiche della realtà d'interesse

La realtà che andiamo a rappresentare riguarda la gestione di una catena di pizzerie situate in diverse città italiane, come Napoli, Salerno, Roma, Milano, etc.

Ogni pizzeria è identificata da un nome che la caratterizza e offre un menù composto da diversi beni alimentari, che possono essere delle pietanze o delle bevande, con rispettivi valori calorici o volume.

Il cliente che intende consumare un pasto, si accomoda ad un tavolo e verrà servito da un unico cameriere per tutta la durata del pasto.

Un cameriere si occuperà di più tavoli durante la stessa serata e le sue ore di lavoro, in una specifica pizzeria della catena, verranno memorizzate con rispettiva data.

Un tavolo è situato in una certa pizzeria e lì possono venire consumati dei pasti, tali pasti sono composti da un elenco di beni alimentari che i clienti scelgono di consumare, specificandone la quantità per ognuno di essi.

Glossario dei termini

Pizzeria: Locale pubblico in cui si preparano, si cuociono, generalmente in forno a legna, e si servono ai clienti pizze e simili, e anche altri piatti di facile preparazione.

Bene alimentare: Alimento commestibile offerto da una pizzeria che può essere di tipo solido (pietanza) oppure liquido (bevanda).

Cameriere: Individuo che svolge una mansione presso una determinata attività.

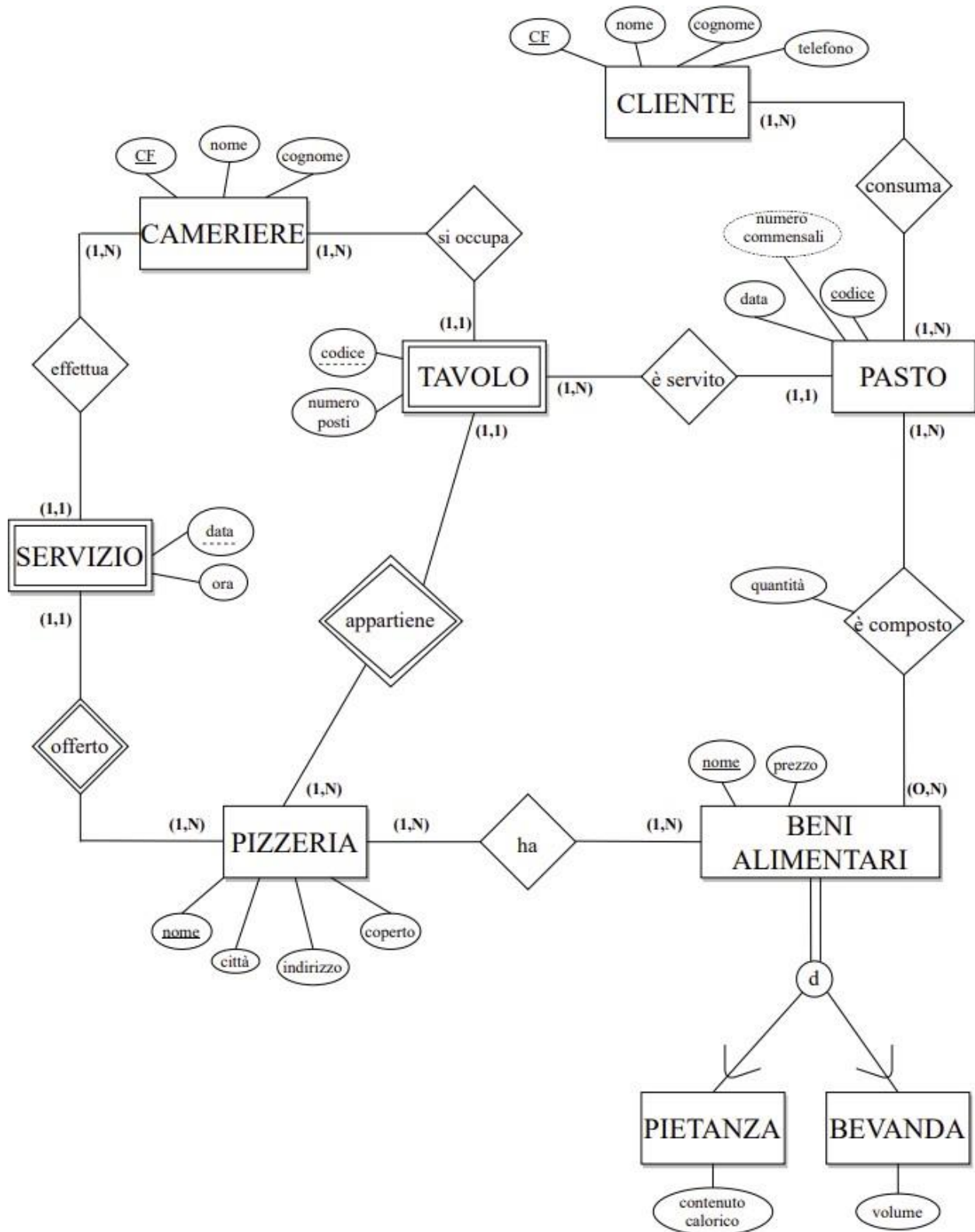
Pasto: Resoconto dei beni alimentari consumati in una determinata data da uno o più clienti all'interno di un'attività.

Cliente: Individuo che usufruisce dei servizi che offre una determinata attività

2. Progettazione concettuale della base di dati

Schema EER

Procedendo con la progettazione concettuale della base di dati, si ottiene il seguente schema EER:



Dizionario delle entità

Entità	Descrizione	Attributi	Identificatore
Pizzeria	Luogo in cui vengono consumati dei pasti.	<ul style="list-style-type: none"> nome città indirizzo coperto 	nome
Servizio	Un cameriere effettua un servizio in una pizzeria.	<ul style="list-style-type: none"> data ora 	data
Cameriere	Persona che lavora in una o più pizzerie.	<ul style="list-style-type: none"> CF nome cognome 	CF
Tavolo	Posto all'interno di una pizzeria dove è possibile consumare dei pasti.	<ul style="list-style-type: none"> codice numero posti 	codice
Pasto	Contiene il resoconto del pasto consumato da uno o più clienti.	<ul style="list-style-type: none"> codice data numero commensali 	codice
Clienti	Persona che usufruisce dei servizi offerti dalle pizzerie della catena.	<ul style="list-style-type: none"> CF nome cognome telefono 	CF
Beni alimentari	Alimento appartenente ad una pizzeria.	<ul style="list-style-type: none"> nome prezzo 	nome
Pietanza	Bene alimentare di tipo solido.	<ul style="list-style-type: none"> contenuto calorico 	-
Bevanda	Bene alimentare di tipo liquido.	<ul style="list-style-type: none"> volume 	-

Legenda:

- sotto-entità
- entità debole
- chiave parziale
- attributo ridondante

Dizionario delle relazioni

Relazione	Descrizione	Entità coinvolte	Attributi
Ha	Una pizzeria ha un insieme di beni alimentari	Pizzeria (1,N) Beni alimentari (1,N)	-
È composto	Un pasto è composto da beni alimentari.	Beni alimentari (0,N) Pasto (1,N)	quantità
Consuma	Un cliente consuma dei pasti della pizzeria.	Pasto (1,N) Cliente (1,N)	-
È servito	Un pasto è servito su un tavolo della pizzeria.	Pasto (1,1) Tavolo (1,N)	-
Si occupa	Un cameriere si occupa dei tavoli della pizzeria.	Tavolo (1,1) Cameriere (1,N)	-
Effettua	Un cameriere effettua un o più servizi.	Cameriere (1,N) Servizio (1,1)	-
Appartiene	Ad una pizzeria appartengono dei tavoli.	Tavolo (1,1) Pizzeria (1,N)	-
Offerto	Un servizio è offerto ad una pizzeria.	Pizzeria (1,N) Servizio (1,1)	-

Vincoli non esprimibili

L'attributo "numero commensali" dell'entità "Pasto" deve avere valore minore o uguale all'attributo "numero posti" dell'entità "Tavolo" a cui è associato.

L'attributo "coperto" dell'entità "Pizzeria" può avere valore massimo pari a 2.

L'attributo "ore" dell'entità "Servizio" deve avere valore minimo pari a 4 e valore massimo pari a 8.

3. Definizione delle procedure per la gestione della base di dati

Tavola dei volumi

Definiamo di seguito la tavola dei volumi della base di dati.

Concetto	Tipo	Carico applicativo
Pizzeria	E	20
Beni alimentari	E	60
Pietanza	E	40
Bevanda	E	20
Servizio	E	300
Cameriere	E	60
Tavolo	E	160
Pasto	E	1600
Cliente	E	800
Ha	R	600
Offerto	R	300
Effettua	R	300
Si occupa	R	160
Appartiene	R	160
Servito	R	1600
Consuma	R	2400

È composto	R	2000
------------	---	------

Tavola delle operazioni

Definiamo di seguito la tavola delle operazioni per la gestione dei dati memorizzati nella base di dati.

	Operazione	Tipo	Frequenza
1	Inserire una nuova pietanza	I	4/mm
2	Inserire un nuovo cliente	I	20/gg
3	Inserire una nuova pizzeria alla catena	I	1/aa
4	Inserire un nuovo tavolo	I	1/mm
5	Registrare un nuovo pasto	I	50/gg
6	Modificare il coperto di una data pizzeria della catena	I	2/aa
7	Rimuovere una data bevanda	I	2/aa
8	Visualizzare tutti i pasti in una specifica data	I	1/gg
9	Registrare un nuovo cameriere	I	2/aa
10	Visualizzare la pietanza che ha il prezzo maggiore	B	1/mm
11	Visualizzare la bevanda con prezzo minore	B	1/mm
12	Selezionare il conto più alto e il nome e il cognome dei clienti che l'hanno pagato in una determinata data	I	2/mm
13	Selezionare tutti i camerieri che hanno svolto più di 500 ore di lavoro in tutte le pizzerie della catena	B	1/aa

14	Aggiornare il prezzo di una data pietanza	I	2/aa
15	Selezionare il numero di commensali di un determinato pasto	I	5/gg
16	Selezionare i tavoli ai quali è stato consumato un pasto con più di 4 commensali	B	1/gg
17	Selezionare nome e cognome dei camerieri che hanno lavorato per tutte le pizzerie della catena	B	1/aa

4. Progettazione logica

Analisi delle ridondanze

Il dato ridondante è l'attributo "numero commensali" dell'entità Pasto. Infatti, sarebbe possibile ottenere il numero di commensali di un pasto attraverso il conto delle partecipazioni dell'entità Cliente nella relazione "Cliente consuma Pasto". Supponendo che l'attributo abbia un peso di 4 byte, essendo un intero, e considerato che il volume dell'entità Cliente è uguale a 800, il dato andrebbe ad occupare uno spazio totale di circa 3200 byte. Per decidere se mantenere o meno il dato ridondante è necessario calcolare, per le operazioni che lo coinvolgono, la differenza nel numero di accessi con e senza quest'ultimo.

Tavola degli accessi

Operazione 2 (Inserire un nuovo cliente)

Calcolo con ridondanza				Calcolo senza ridondanza			
Tabella	Tipo	Accessi	Tipo accessi	Tabella	Tipo	Accessi	Tipo accessi
Cliente	E	1	S	Cliente	E	1	S
Consuma	R	3	S	Consuma	R	3	S
Pasto	E	3	L				
Pasto	E	3	S				
Totale: $[3+(1+3+3) \times 2] \times 20 = 340$ a/gg				Totale: $[(3+1) \times 2] \times 20 = 160$ a/gg			

Operazione 15 (Selezionare il numero di commensali di un determinato pasto)

Calcolo con ridondanza				Calcolo senza ridondanza			
Tabella	Tipo	Accessi	Tipo accessi	Tabella	Tipo	Accessi	Tipo accessi
Pasto	E	1	L	Pasto	E	1	L
				Consuma	R	1,5	L
Totale: $1 \times 5 = 5$ a/gg				Totale: $(1+1,5) \times 5 = 12,5$ a/gg			

Operazione 16 (Selezionare i tavoli ai quali è stato consumato un pasto con più di 4 commensali)

Calcolo con ridondanza				Calcolo senza ridondanza			
Tabella	Tipo	Accessi	Tipo accessi	Tabella	Tipo	Accessi	Tipo accessi
Tavolo	E	160	L	Tavolo	E	160	L
È servito	R	1600	L	È servito	R	1600	L
Pasto	E	1600	L	Pasto	E	1600	L
				Consuma	R	2400	L
Totale: $(160+1600+1600) \times 1 = 3360$ a/gg				Totale: $(160+1600+1600+2400) \times 1 = 5760$ a/gg			

Totale accessi con ridondanza

$$340 + 5 + 3360 =$$

$$3705 \text{ a/gg} + 3200 \text{ byte}$$

Totale accessi senza ridondanza

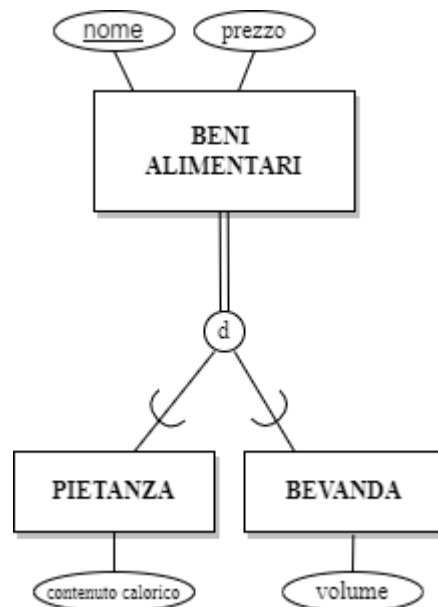
$$160 + 12,5 + 5760 =$$

$$5932,5 \text{ a/gg}$$

Dato il minor numero di accessi, è più efficiente scegliere di **mantenere** il dato ridondante “numero commensali”.

Eliminazione delle gerarchie

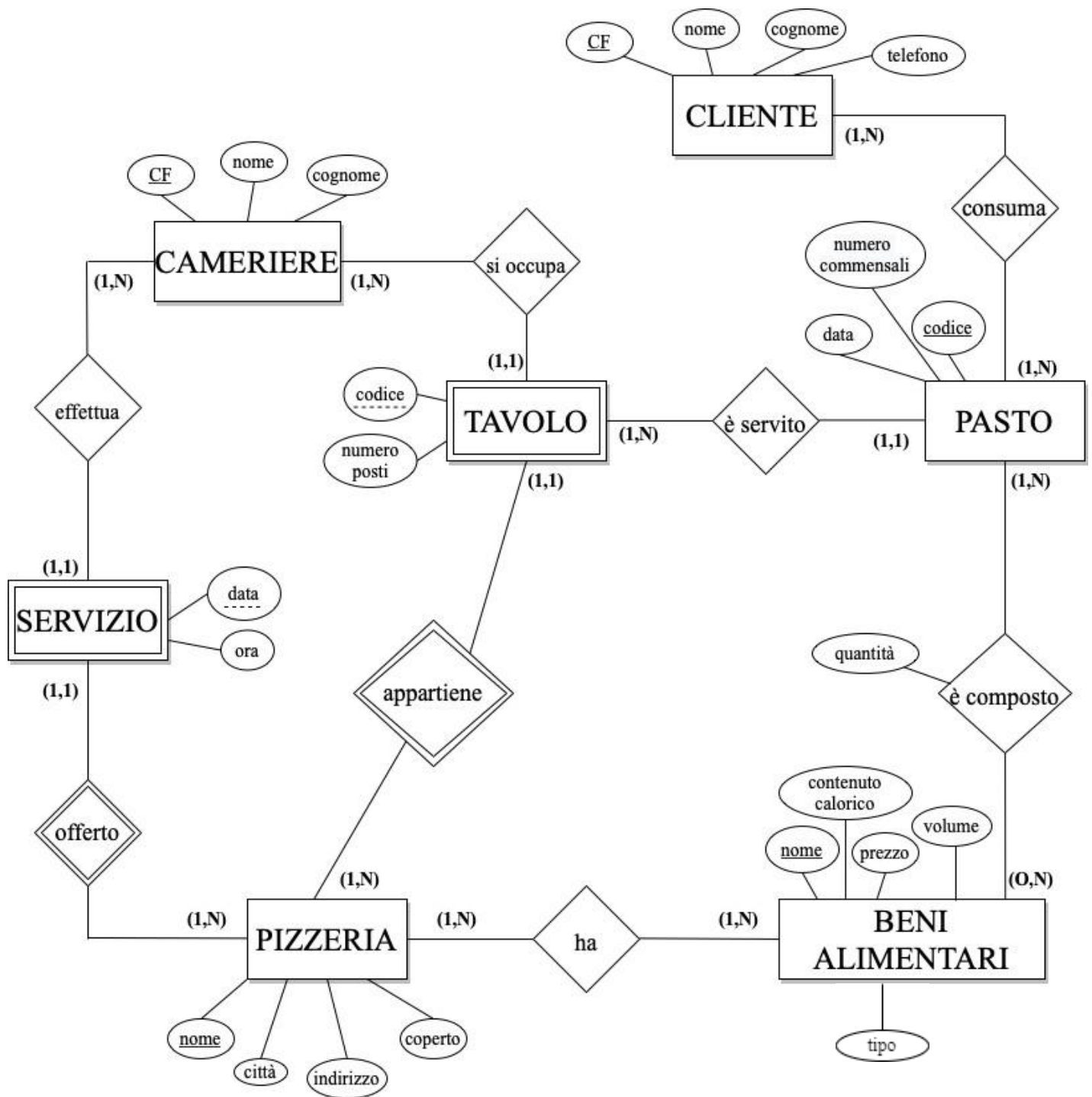
Nello schema inizialmente elaborato, è presente la seguente specializzazione dell'entità "Beni alimentari":



La scelta effettuata è l'accorpamento delle due entità figlie nel padre, con l'aggiunta di un attributo "tipo" per distinguere le due sottoclassi. Ciò è possibile perché la generalizzazione è totale e di conseguenza l'entità padre "Beni Alimentari" è obbligatoriamente o di tipo "pietanza" o di tipo "bevanda". In questo modo introduciamo valori NULL o per l'attributo "contenuto calorico" o per l'attributo "volume". Ristrutturiamo quindi come segue:



Schema EER ristrutturato



Schema relazionale

Si procede al mapping della base di dati

Pizzeria(nome, città, indirizzo, coperto)

BeniAlimentari(nome, prezzo, contenuto calorico*, volume*, tipo)

Servizio(data, pizzeria.nome↑, ore, cameriere.CF↑)

Cameriere(CF, nome, cognome)

Tavolo(codice, pizzeria.nome↑, numero posti, cameriere.CF↑)

Pasto(codice, data, numero commensali, tavolo.nome↑)

Cliente(CF, nome, cognome, telefono)

Ha(pizzeria.nome↑, beniAlimentari.nome↑)

ÈComposto(pasto.codice↑, beniAlimentari.nome↑, quantità)

Consuma(pasto.codice↑, cliente.CF↑)

Normalizzazione

Il database si presenta già in **prima forma normale** in quanto dopo la ristrutturazione tutti gli attributi sono atomici.

E' in **seconda forma normale** perchè, oltre ad essere già in prima forma normale, quando è presente una chiave primaria composta da più attributi tutte le dipendenze funzionali che la riguardano sono piene e non parziali.

E' in **terza forma normale** perchè, oltre ad essere già in seconda forma normale, in tutte le tabelle non sono presenti dipendenze transitive fra attributi non chiave e la chiave primaria.

5. Realizzazione della base di dati con MySQL

Di seguito il listato delle istruzioni MySQL per la creazione della base di dati:

```
1 • create database CatenaPizzeria;
2 • use CatenaPizzeria;
3
4 • create table Pizzeria
5 • (
6     nome varchar(50) NOT NULL,
7     citta varchar(20) NOT NULL,
8     indirizzo varchar(30) NOT NULL,
9     coperto int NOT NULL,
10    PRIMARY KEY (nome)
11 );
12
13 • create table BeniAlimentari
14 • (
15     nome varchar(50) NOT NULL,
16     prezzo float NOT NULL,
17     contenutoCalorico int,
18     volume float,
19     tipo varchar(10) NOT NULL,
20    PRIMARY KEY (nome)
21 );
22
23 • create table Cameriere
24 • (
25     CF char(16) NOT NULL,
26     nome varchar(30) NOT NULL,
27     cognome varchar(30) NOT NULL,
28    PRIMARY KEY (CF)
29 );
30
31 • create table Cliente
32 • (
33     CF char(16) NOT NULL,
34     nome varchar(30) NOT NULL,
35     cognome varchar(30) NOT NULL,
36     telefono char(10) NOT NULL,
37    PRIMARY KEY (CF)
38 );
39
40 • create table Servizio
41 • (
42     data DATE NOT NULL,
43     nome varchar(50) NOT NULL,
44     ore int NOT NULL,
45     CF char(16) NOT NULL,
46    PRIMARY KEY (data,nome),
47    FOREIGN KEY (nome) REFERENCES Pizzeria(nome)
48        ON DELETE CASCADE
49        ON UPDATE CASCADE,
50    FOREIGN KEY (CF) REFERENCES Cameriere(CF)
51        ON DELETE CASCADE
52        ON UPDATE CASCADE
53 );
54
```

```

55 • create table Tavolo
56 (
57     codice int NOT NULL,
58     nome varchar(50) NOT NULL,
59     numeroPosti int NOT NULL,
60     CF char(16) NOT NULL,
61     PRIMARY KEY (codice,nome),
62     FOREIGN KEY (nome) REFERENCES Pizzeria(nome)
63         ON DELETE CASCADE
64         ON UPDATE CASCADE,
65     FOREIGN KEY (CF) REFERENCES Cameriere(CF)
66         ON DELETE CASCADE
67         ON UPDATE CASCADE
68 );
69
70 • create table Pasto
71 (
72     codice int NOT NULL,
73     data DATE NOT NULL,
74     numeroCommensali int NOT NULL,
75     nome varchar(50) NOT NULL,
76     PRIMARY KEY (codice),
77     FOREIGN KEY (nome) REFERENCES Tavolo(nome)
78         ON DELETE CASCADE
79         ON UPDATE CASCADE
80 );
81

```

```

82 • create table Ha
83 (
84     pizzeriaNome varchar(50) NOT NULL,
85     alimentoNome varchar(50) NOT NULL,
86     PRIMARY KEY (pizzeriaNome,alimentoNome),
87     FOREIGN KEY (pizzeriaNome) REFERENCES Pizzeria(nome)
88         ON DELETE CASCADE
89         ON UPDATE CASCADE,
90     FOREIGN KEY (alimentoNome) REFERENCES BeniAlimentari(nome)
91         ON DELETE CASCADE
92         ON UPDATE CASCADE
93 );
94
95 • create table EComposto
96 (
97     codice int NOT NULL,
98     nome varchar(50) NOT NULL,
99     quantita int NOT NULL,
100     PRIMARY KEY (codice,nome),
101     FOREIGN KEY (codice) REFERENCES Pasto(codice)
102         ON DELETE CASCADE
103         ON UPDATE CASCADE,
104     FOREIGN KEY (nome) REFERENCES BeniAlimentari(nome)
105         ON DELETE CASCADE
106 );
107
108 • create table Consuma
109 (
110     codice int NOT NULL,
111     CF char(16) NOT NULL,
112     PRIMARY KEY (codice,CF),
113     FOREIGN KEY (codice) REFERENCES Pasto(codice)
114         ON DELETE CASCADE
115         ON UPDATE CASCADE,
116     FOREIGN KEY (CF) REFERENCES Cliente(CF)
117         ON DELETE CASCADE
118         ON UPDATE CASCADE
119 );
120
121

```

5. Implementazione query SQL

-- Operazione 1:

```
INSERT INTO BeniAlimentari(nome, prezzo, contenutoCalorico, tipo)
```

```
VALUES
```

```
(  
    ?,?,?,pietanza  
);
```

-- Operazione 2:

```
INSERT INTO Cliente(CF, nome, cognome, telefono)
```

```
VALUES
```

```
(  
    ?,?,?,?  
);
```

-- Operazione 3:

```
INSERT INTO Pizzeria(nome, citta, indirizzo, coperto)
```

```
VALUES
```

```
(  
    ?,?,?,?  
);
```

-- Operazione 4:

```
INSERT INTO Tavolo(codice, nome, numeroPosti, CF)
```

```
VALUES
```

```
(  
    ?,?,?,?  
);
```

-- Operazione 5:

```
INSERT INTO Pasto(codice, data, numeroCommensali, nome)
```

```
VALUES (
```

```
?, ?, ?, ?
```

```
);
```

-- Operazione 6:

```
UPDATE Pizzeria SET coperto = ? WHERE nome = ?;
```

-- Operazione 7:

```
DELETE
```

```
FROM BeniAlimentari
```

```
WHERE tipo="bevanda" AND nome = ?;
```

-- Operazione 8:

```
SELECT *
```

```
FROM Pasto AS p
```

```
WHERE p.data = ?;
```

-- Operazione 9:

```
INSERT INTO Cameriere(CF, nome, cognome)
```

```
VALUES
```

```
(
```

```
?, ?, ?
```

```
);
```

-- Operazione 10:

```
SELECT nome, prezzo
```

```
FROM BeniAlimentari
```

```
WHERE tipo='pietanza' AND prezzo >= ALL(SELECT b2.prezzo
```

```
FROM BeniAlimentari AS b2
```

```
WHERE b2.nome=nome);
```


-- Operazione 11

SELECT nome, prezzo

FROM BeniAlimentari

WHERE tipo='bevanda' AND prezzo <= ALL(SELECT b2.prezzo

FROM BeniAlimentari AS b2

WHERE b2.nome=nome);

-- Operazione 12:

CREATE VIEW contalImporto AS

SELECT c.CF, p.codice, SUM(e.quantita * b.prezzo) AS importo

FROM Cliente AS c, Consuma AS con, Pasto AS p, EComposto AS e, BeniAlimentari AS b

WHERE c.CF=con.CF AND con.codice=p.codice AND p.codice=e.codice AND e.nome=b.nome

GROUP BY c.CF, p.codice;

SELECT DISTINCT c.nome, c.cognome, i.importo

FROM Pasto AS p, Consuma AS con, Cliente AS c, contalImporto AS i

WHERE p.codice=con.codice AND con.CF=c.CF AND c.CF=i.CF

AND p.data= ? AND i.importo >= (SELECT MAX(importo) AS massimo

FROM contalImporto);

-- Operazione 13:

SELECT c.nome, c.cognome

FROM Cameriere AS c, Servizio AS s

WHERE s.CF=c.CF

GROUP BY c.CF

HAVING SUM(s.ore) > 500;

-- Operazione 14:

UPDATE BeniAlimentari SET prezzo = ? WHERE tipo="Pietanza" AND nome = ?;

-- Operazione 15:

```
SELECT p.numeroCommensali  
FROM Pasto AS p  
WHERE p.codice = ?;
```

-- Operazione 16:

```
SELECT t.codice, t.nome  
FROM Pasto AS p , Tavolo AS t  
WHERE p.numeroCommensali > 4 AND t.nome=p.nome;
```

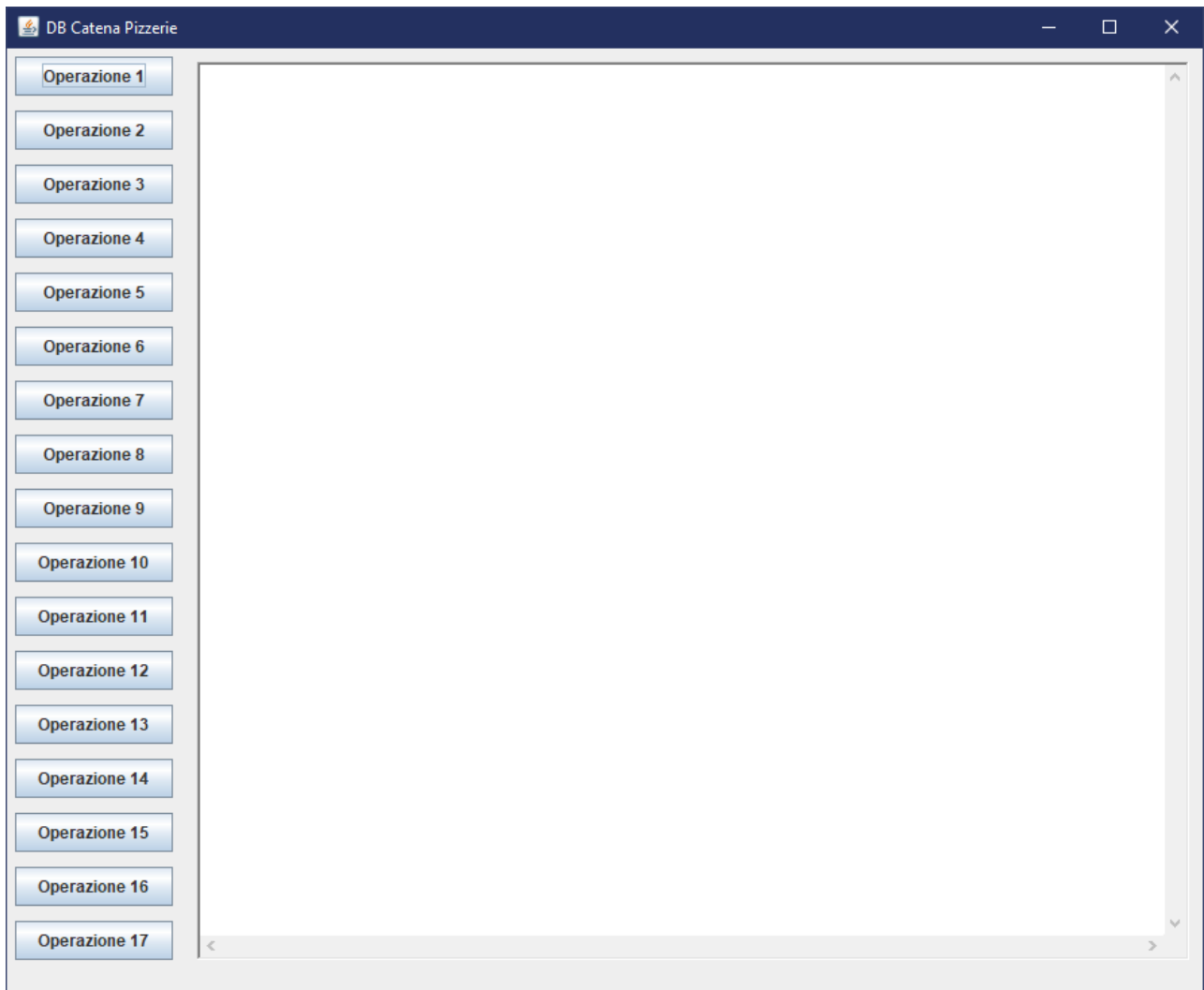
-- Operazione 17:

-- Selezionare nome e cognome dei camerieri che hanno lavorato per tutte le pizzerie della catena

```
SELECT c.nome, c.cognome  
FROM Cameriere AS c  
WHERE NOT EXISTS(SELECT *  
                  FROM Pizzeria AS p  
                  WHERE NOT EXISTS(SELECT *  
                                    FROM Servizio AS s, Cameriere AS c1  
                                    WHERE s.CF=c1.cf AND s.nome=p.nome AND c1.CF=c.CF));
```

7. Test dell'applicazione Java

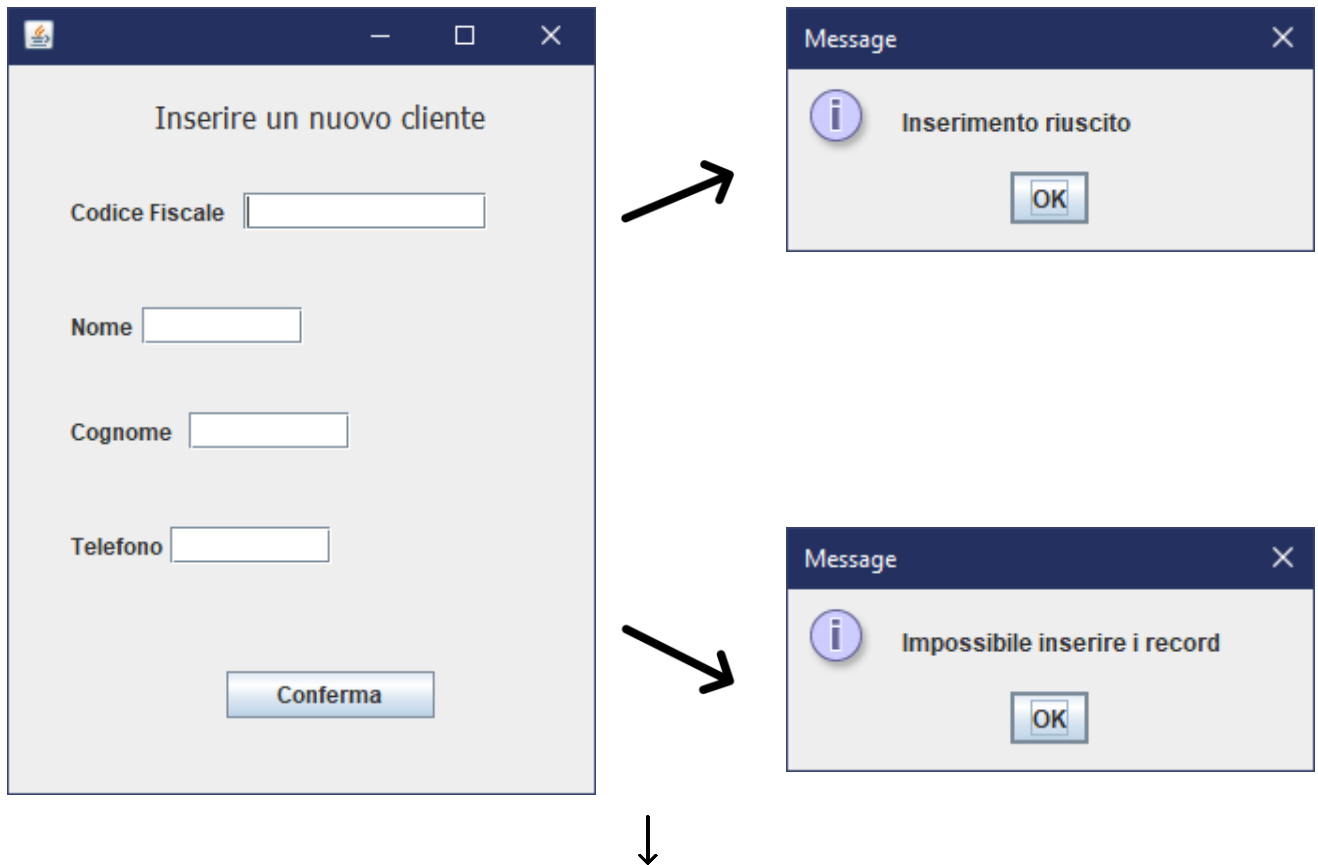
All'avvio dell'applicazione grafica ci viene presentato un frame dove sulla sinistra possiamo andare ad eseguire le varie operazioni e visualizzare il risultato o il resoconto in un'area di testo apposita.



Posizionando il mouse sui vari pulsanti è possibile visualizzare l'operazione ad essi associata.



Le operazioni di inserimento andranno ad aprire un ulteriore frame che permetterà di inserire tutti i valori desiderati. L'inserimento andato a buon fine verrà segnalato e un riepilogo dei dati appena registrati verrà stampato. In caso di fallimento dell'inserimento avremo un'ulteriore segnalazione.



The screenshot shows the "DB Catena Pizzeria" application window. On the left, there is a vertical list of buttons labeled "Operazione 1" through "Operazione 17". The "Operazione 1" button is highlighted. The main area of the window displays the text "Inserimento cliente effettuato" (Client insertion performed) followed by the details of the inserted client: "Codice Fiscale: DRSMRC45E76H703B", "Nome: Marco", "Cognome: De Rossi", and "Telefono: 3924565987".

Di seguito altri screenshot di risultati delle query

?

Inserire nome bevanda da rimuovere

Fanta

OKCancel

DB Catena Pizzerie

Operazione 1

Operazione 2

Operazione 3

Operazione 4

Operazione 5

Operazione 6

Operazione 7

Operazione 8

Operazione 9

Operazione 10

Operazione 11

Operazione 12

Operazione 13

Operazione 14

Operazione 15

Operazione 16

Operazione 17

Cancellamento bene alimentare: Fanta effettuato

Inserire la pizzeria

Sorbillo

Inserire il nuovo coperto

1

Conferma

DB Catena Pizzerie

Operazione 1

Operazione 2

Operazione 3

Operazione 4

Operazione 5

Operazione 6

Operazione 7

Operazione 8

Operazione 9

Operazione 10

Operazione 11

Operazione 12

Operazione 13

Operazione 14

Operazione 15

Operazione 16

Operazione 17

Aggiornamento coperto effettuato
Pizzeria: Sorbillo
Nuovo coperto: 1

Operazione 16

Selezionare i tavoli ai quali è stato consumato un pasto con più di 4 commensali

DB Catena Pizzerie

Operazione 1	Codice tavolo: 2 Pizzeria: Sorbillo
Operazione 2	Codice tavolo: 4 Pizzeria: Sorbillo
Operazione 3	Codice tavolo: 4 Pizzeria: Da Michele
Operazione 4	Codice tavolo: 7 Pizzeria: Da Michele
Operazione 5	
Operazione 6	
Operazione 7	
Operazione 8	
Operazione 9	
Operazione 10	
Operazione 11	
Operazione 12	
Operazione 13	
Operazione 14	
Operazione 15	
Operazione 16	
Operazione 17	