

Progetto di Basi di Dati

~ Click&Eat ~

- Servizio online di ordinazione e consegna pasti -

1. Raccolta delle specifiche della realtà d'interesse

Descrizione

Si vuole progettare una base di dati per la gestione di un servizio online di ordinazione e consegna pasti.

L'utente si registra alla piattaforma tramite la creazione di un account, sul quale vengono memorizzati i suoi dati, quali nome, cognome, indirizzo, numero di telefono, e-mail (che lo identifica) e password. Si tiene traccia anche del numero di ordini effettuati.

L'utente effettua ordini presso i locali presenti sulla piattaforma. Ogni locale è caratterizzato da un nome, un indirizzo, un numero di telefono e dalla sua tipologia (pizzeria, ristorante, etc....).

Ciascun locale ha uno o più menù caratterizzati da un nome, dal numero di pagine e numero di prodotti di cui sono composti.

Ogni prodotto ha un nome e prezzo e può essere una bibita o una pietanza. Le bibite possono essere alcoliche o analcoliche; invece, per le pietanze si tiene conto degli ingredienti.

Ogni ordine è identificato da un codice ed è caratterizzato dall'orario in cui è stato effettuato, dal prezzo totale, da eventuali note da parte dell'utente e può comprendere più volte lo stesso prodotto.

I locali dispongono di uno o più fattorini che consegnano (entro l'orario previsto) gli ordini. Il fattorino è identificato dal codice fiscale e si tiene conto di nome, cognome e numero di telefono.

Ogni utente può scrivere delle recensioni per i locali caratterizzate da data, ora, numero di stelle e una descrizione.

Ogni locale può mettere a disposizione dei coupon per permettere agli utenti di usufruire di sconti sugli ordini. Il coupon è caratterizzato da un codice, spesa minima, valore dello sconto, data inizio e data scadenza.

Inoltre, i locali possono rilasciare una fidelity card che tiene conto del numero delle consumazioni, identificata da un codice e dal nome del locale. Ogni account può possedere una o più fidelity card.

Specifiche della realtà d'interesse

La realtà d'interesse che andiamo a rappresentare riguarda la gestione di un servizio online che permette a un cliente di ordinare e ricevere direttamente a casa uno o più pasti da parte dei locali associati alla piattaforma.

Negli ultimi anni, i servizi online di ordinazione e consegna pasti sono diventati sempre più comuni, data la loro comodità e facilità di utilizzo.

Inoltre, i locali che aderiscono a questo tipo di iniziativa continuano ad aumentare data la grande affluenza di utenti che utilizzano questi servizi.

Sono sempre di più, infatti, le persone che preferiscono la consegna a domicilio, specialmente in mancanza di mezzi per raggiungere fisicamente il locale.

La realtà d'interesse presa in considerazione vuole che l'utente si registri alla piattaforma tramite un account, dal quale è possibile effettuare ordini presso il locale interessato (tra cui pizzerie, ristoranti, fast food etc....) scegliendo tra i prodotti offerti da quel locale. La scelta è effettuata tramite uno o più menù digitali che vengono proposti. Questi ultimi sono composti da vari prodotti, che possono essere pietanze o bibite, il cui prezzo può variare a seconda del locale selezionato.

Ogni locale dispone di uno o più fattorini che consegnano gli ordini, la consegna sarà effettuata entro un orario stabilito nel momento in cui l'ordine è stato confermato.

In seguito a ogni ordine, un utente ha la possibilità di scrivere una recensione in merito alla sua esperienza, contenente una nota per il locale in questione con un punteggio che può variare da 1 a 5.

Inoltre, ogni locale può mettere a disposizione dei coupon che danno la possibilità di applicare uno sconto sulla spesa complessiva dell'ordine.

I coupon vanno utilizzati tramite un codice che l'utente deve inserire al momento dell'ordine.

I locali possono anche rilasciare una fidelity card che permette agli utenti di ricevere un pasto (1 pietanza + 1 bibita) in omaggio ogni 10 consumazioni. Ogni account può possedere una fidelity card per ogni locale che la offre.

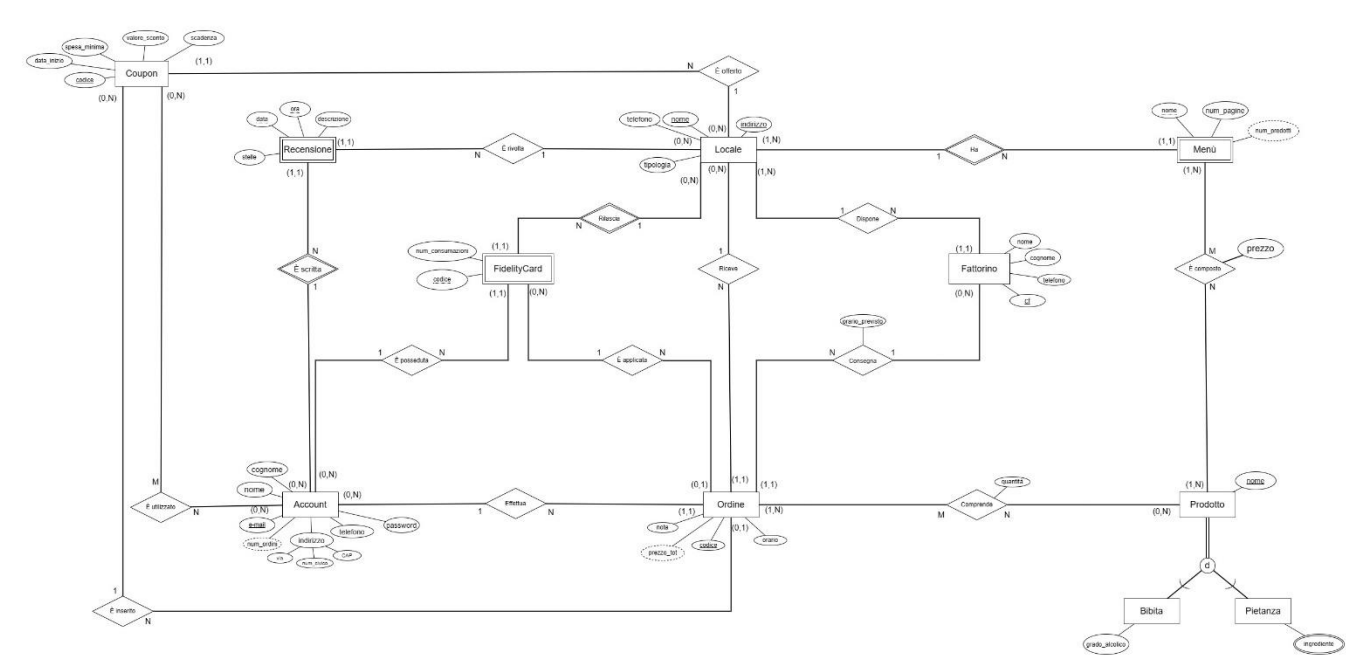
Glossario dei termini

| Termine | Significato |
|---------------|---|
| Account | Insieme dei dati identificativi di un utente che gli consentono l’accesso al servizio e il suo utilizzo. |
| Coupon | Buono per acquistare dei prodotti a prezzo ridotto. |
| Fidelity card | Carta elettronica finalizzata a tener conto del numero di consumazioni degli utenti per poi offrire un omaggio. |
| Prodotto | Pietanza o bibita selezionata dal menù offerto dal locale. |

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 debole
- Sotto-entità
- Chiave parziale
- Attributo multivalore
- Attributo composto
- Attributo derivato

| Entità | Descrizione | Attributi | Identificatore |
|-------------------|---|--|---------------------------|
| Account | Identità di un utente sulla piattaforma | <ul style="list-style-type: none"> ➤ e-mail ➤ password ➤ telefono ➤ nome ➤ cognome ➤ <u>indirizzo</u> ➤ <u>num_ordini</u> | e-mail |
| Locale | Negoziato presso il quale effettuare un ordine | <ul style="list-style-type: none"> ➤ nome ➤ indirizzo ➤ tipologia ➤ telefono | nome indirizzo |
| <u>Menù</u> | Lista dei prodotti offerti da un locale | <ul style="list-style-type: none"> ➤ nome ➤ num_pagine ➤ <u>num_prodotto</u> | <u>nome</u> |
| Prodotto | Pietanza o bibita selezionata dal menù offerto dal locale | <ul style="list-style-type: none"> ➤ nome | nome |
| <u>Bibita</u> | Bevanda offerta dal locale | <ul style="list-style-type: none"> ➤ grado_alcolico | / |
| <u>Pietanza</u> | Piatto offerto dal locale | <ul style="list-style-type: none"> ➤ <u>ingrediente</u> | / |
| Ordine | Acquisto effettuato dal cliente | <ul style="list-style-type: none"> ➤ codice ➤ orario ➤ <u>prezzo_tot</u> ➤ nota | codice |
| <u>Recensione</u> | Commento rilasciato dal cliente in merito alla qualità del servizio | <ul style="list-style-type: none"> ➤ data ➤ ora ➤ stelle ➤ descrizione | <u>data</u> <u>ora</u> |
| Coupon | Buono sconto offerto dal locale | <ul style="list-style-type: none"> ➤ codice ➤ data_inizio ➤ scadenza ➤ spesa_minima ➤ valore_sconto | codice |
| Fattorino | Dipendente che si occupa della consegna degli ordini | <ul style="list-style-type: none"> ➤ cf ➤ nome ➤ cognome ➤ telefono | cf |

| | | | |
|---------------------|---|--|---------------|
| <u>FidelityCard</u> | Carta che tiene conto delle consumazioni degli utenti | <ul style="list-style-type: none"> ➤ codice ➤ num_consumazioni | <u>codice</u> |
|---------------------|---|--|---------------|

Dizionario delle relazioni

| Relazione | Descrizione | Entità coinvolte | Attributi |
|--------------|--|---|-----------------|
| È utilizzato | Il coupon è utilizzato dall'account | <ul style="list-style-type: none"> ➤ Coupon (0, N) ➤ Account (0, N) | / |
| È scritta | La recensione è scritta dall'account | <ul style="list-style-type: none"> ➤ Recensione (1,1) ➤ Account (0, N) | / |
| È rivolta | La recensione è rivolta al locale | <ul style="list-style-type: none"> ➤ Recensione (1,1) ➤ Locale (0, N) | / |
| Effettua | L'account effettua l'ordine | <ul style="list-style-type: none"> ➤ Account (0, N) ➤ Ordine (1, 1) | / |
| È composto | Il menù è composto dai prodotti | <ul style="list-style-type: none"> ➤ Menù (1, N) ➤ Prodotto (1, N) | prezzo |
| È offerto | Il coupon è offerto dal locale | <ul style="list-style-type: none"> ➤ Coupon (1, 1) ➤ Locale (0, N) | / |
| Riceve | Il locale riceve l'ordine | <ul style="list-style-type: none"> ➤ Locale (0, N) ➤ Ordine (1, 1) | / |
| Ha | Il locale ha uno o più menù | <ul style="list-style-type: none"> ➤ Locale (1, N) ➤ Menù (1,1) | / |
| Comprende | L'ordine comprende dei prodotti | <ul style="list-style-type: none"> ➤ Ordine (1, N) ➤ Prodotto (0, N) | quantità |
| Dispone | Il locale dispone di uno o più fattorini | <ul style="list-style-type: none"> ➤ Locale (1, N) ➤ Fattorino (1, 1) | / |
| Consegna | Il fattorino consegna gli ordini | <ul style="list-style-type: none"> ➤ Fattorino (0, N) ➤ Ordine (1, 1) | orario_previsto |
| Rilascia | Il locale rilascia la fidelity card | <ul style="list-style-type: none"> ➤ Locale (0, 1) ➤ FidelityCard (1, 1) | / |
| È posseduta | La fidelity card è posseduta da un account | <ul style="list-style-type: none"> ➤ FidelityCard (1, 1) ➤ Account (0, N) | / |
| È applicata | La fidelity card è applicata all'ordine | <ul style="list-style-type: none"> ➤ FidelityCard (0, N) ➤ Ordine (0, 1) | / |
| È inserito | Il coupon è inserito nell'ordine | <ul style="list-style-type: none"> ➤ Coupon (0, N) ➤ Ordine (0, 1) | / |

Vincoli non esprimibili nello schema

Oltre ciò che è deducibile dallo schema EER, si tenga conto dei seguenti vincoli:

- Il valore dell'attributo *stelle* dell'entità Recensione deve essere compreso tra 1 e 5.
- L'attributo *descrizione* dell'entità Recensione non deve superare i 1000 caratteri.
- La fidelity card può essere utilizzata per ottenere l'omaggio ogni volta che il valore dell'attributo *num_consumazioni* = 10.
- L'attributo *nota* dell'entità Ordine non deve superare i 500 caratteri.
- L'attributo *password* dell'entità Account deve essere composto da minimo 8 caratteri tra cui:
 - un carattere maiuscolo;
 - un carattere minuscolo;
 - un carattere numerico.

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

Tavola dei volumi

| Concetto | Tipo | Carico applicativo |
|--------------|------|--------------------|
| Account | E | 50 |
| Locale | E | 10 |
| Menù | E | 20 |
| Prodotto | E | 200 |
| Bibita | SE | 60 |
| Pietanza | SE | 140 |
| Ordine | E | 150 |
| Recensione | E | 100 |
| Coupon | E | 5 |
| Fattorino | E | 20 |
| FidelityCard | E | 150 |
| È utilizzato | R | 15 |
| È scritta | R | 100 |
| È rivolta | R | 100 |
| Effettua | R | 150 |
| È composto | R | 800 |
| È offerto | R | 5 |
| Riceve | R | 150 |
| Ha | R | 20 |

| | | |
|-------------|---|-----|
| Comprende | R | 750 |
| Dispone | R | 20 |
| Consegna | R | 150 |
| Rilascia | R | 150 |
| È posseduta | R | 150 |
| È applicata | R | 10 |
| È inserito | R | 30 |

Tavola delle operazioni

| Numero | Operazione | Tipo | Frequenza |
|--------|--|------|-----------|
| 1 | Creare un nuovo account | I | 5/mese |
| 2 | Registrare un nuovo locale | I | 2/anno |
| 3 | Registrare un nuovo fattorino | I | 3/anno |
| 4 | Scrivere una recensione | I | 400/mese |
| 5 | Effettuare un ordine | I | 600/mese |
| 6 | Inserire prodotto nel menù | I | 1/mese |
| 7 | Sottoscrivere una fidelity card | I | 15/mese |
| 8 | Visualizzare le fidelity card possedute da un account | I | 40/mese |
| 9 | Visualizzare il numero di ordini effettuati da un account | I | 5/mese |
| 10 | Visualizzare nome e numero del fattorino che consegna l'ordine | I | 500/mese |
| 11 | Visualizzare i prodotti compresi in un ordine | I | 600/mese |
| 12 | Offrire un nuovo coupon | I | 4/mese |

| | | | |
|----|--|---|----------|
| 13 | Visualizzare tutti i locali che hanno ricevuto recensioni con una media ≥ 3 stelle | B | 300/mese |
| 14 | Selezionare tutte le pizzerie che si trovano in una data località | I | 400/mese |
| 15 | Visualizzare i dati e l'orario previsto dell'ordine | I | 600/mese |
| 16 | Rimuovere un prodotto dal menù | I | 4/anno |
| 17 | Visualizzare i menù di un locale | I | 650/mese |
| 18 | Modificare il prezzo di un prodotto | I | 10/anno |
| 19 | Lista delle bibite non alcoliche di un dato locale | I | 100/mese |
| 20 | Visualizzare gli ingredienti di una pietanza | I | 200/mese |
| 21 | Visualizzare i prodotti di un menù | I | 650/mese |
| 22 | Visualizzare i locali che offrono sia coupon sia fidelity card | B | 100/mese |
| 23 | Visualizzare quante volte un locale ha ricevuto un ordine che comprende un dato prodotto in un certo mese. | I | 10/mese |

4. Progettazione logica

Analisi ridondanze

I dati ridondanti sono gli attributi: *num_ordini*, *num_prodotti*, *prezzo_tot*.

1. *num_ordini* è ridondante perché è derivabile dal conteggio delle partecipazioni dell'entità "Account" alla relazione "Effettua";
2. *num_prodotti* è ridondante perché è derivabile dal conteggio delle partecipazioni dell'entità "Menù" alla relazione "È composto";
3. *prezzo_tot* è ridondante perché è derivabile dalla somma dell'attributo prezzo (della relazione "È composto") di ogni prodotto compreso nell'ordine.

Tavola degli accessi (1)

- Attributo: *num_ordini*.
- Operazioni coinvolte: 5, 9.
- Memoria occupata: 200 byte (supponendo che un dato di tipo int occupi 4 byte, moltiplichiamo questo valore per il volume dell'entità "Account").

Operazione 5: effettuare un ordine

| Con ridondanza | | | | Senza ridondanza | | | |
|--|-----------|---------|------|--|-----------|---------|------|
| concetto | costrutto | accesso | tipo | concetto | costrutto | accesso | tipo |
| Ordine | E | 1 | S | Ordine | E | 1 | S |
| Effettua | R | 1 | S | Effettua | R | 1 | S |
| Account | E | 1 | L | È inserito | R | 0,2 | S |
| Account | E | 1 | S | Comprende | R | 5 | S |
| È inserito | R | 0,2 | S | Consegna | R | 1 | S |
| Comprende | R | 5 | S | È applicata | R | 0,07 | S |
| Consegna | R | 1 | S | Riceve | R | 1 | S |
| È applicata | R | 0,07 | S | | | | |
| Riceve | R | 1 | S | | | | |
| Tot: [(1+1+1+1+0.2+5+1+0.07+1) *2+1] *7200 = 169488 accessi/anno | | | | Tot: [(1+1+0.2+5+1+0.07+1) *2] *7200 = 133488 accessi/anno | | | |

Operazione 9: visualizzare il numero di ordini effettuati da un account

| Con ridondanza | | | | Senza ridondanza | | | |
|---|-----------|---------|------|--|-----------|---------|------|
| concetto | costrutto | accesso | tipo | concetto | costrutto | accesso | tipo |
| Account | E | 1 | L | Account | E | 1 | L |
| | | | | Effettua | R | 3 | L |
| Tot: $1 \cdot 60 = 60$ accessi/anno | | | | Tot: $(1+3) \cdot 60 = 240$ accessi/anno | | | |

| Totale accessi con ridondanza | Totale accessi senza ridondanza |
|--|--|
| $169488 + 60 =$ 169548 accessi/anno + 200 byte | $133488 + 240 =$ 133728 accessi/anno |

Dato il minor numero di accessi senza ridondanza conviene eliminare l'attributo ridondante.



Tavola degli accessi (2)

- Attributo: *num_prodotti*.
- Operazioni coinvolte: 6, 16.
- Memoria occupata: 80 byte (supponendo che un dato di tipo int occupi 4 byte, moltiplichiamo questo valore per il volume dell'entità "Menù").

Operazione 6: inserire prodotto nel menù

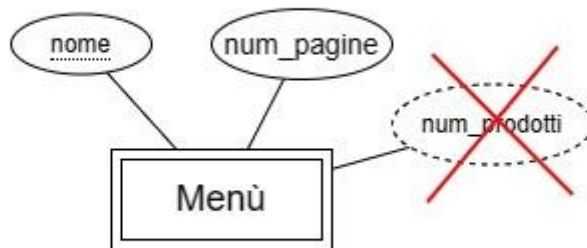
| Con ridondanza | | | | Senza ridondanza | | | |
|--|-----------|---------|------|--|-----------|---------|------|
| concetto | costrutto | accesso | tipo | concetto | costrutto | accesso | tipo |
| Menù | E | 1 | L | Menù | E | 1 | L |
| Menù | E | 1 | S | È composto | R | 1 | S |
| È composto | R | 1 | S | | | | |
| Tot: $[(1+1) \cdot 2+1] \cdot 12 =$ 60 accessi/anno | | | | Tot: $(1 \cdot 2+1) \cdot 12 =$ 36 accessi/anno | | | |

Operazione 16: rimuovere un prodotto dal menù

| Con ridondanza | | | | Senza ridondanza | | | |
|---|-----------|---------|------|---|-----------|---------|------|
| concetto | costrutto | accesso | tipo | concetto | costrutto | accesso | tipo |
| Menù | E | 1 | L | Menù | E | 1 | L |
| Menù | E | 1 | S | È composto | R | 1 | L |
| È composto | R | 1 | L | È composto | R | 1 | S |
| È composto | R | 1 | S | | | | |
| Tot: $[(1+1) * 2 + 1 + 1] * 4 =$ 24 accessi/anno | | | | Tot: $(1 * 2 + 1 + 1) * 4 =$ 16 accessi/anno | | | |

| Totale accessi con ridondanza | Totale accessi senza ridondanza |
|--|---------------------------------|
| 60 + 24 = 84 accessi/anno + 80 byte | 36 + 16 = 52 accessi/anno |

Dato il minor numero di accessi senza ridondanza conviene eliminare l'attributo ridondante.

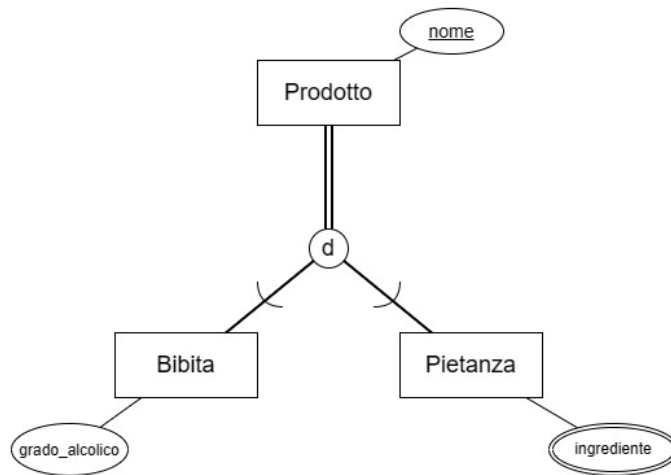


Analisi attributo "prezzo_tot"

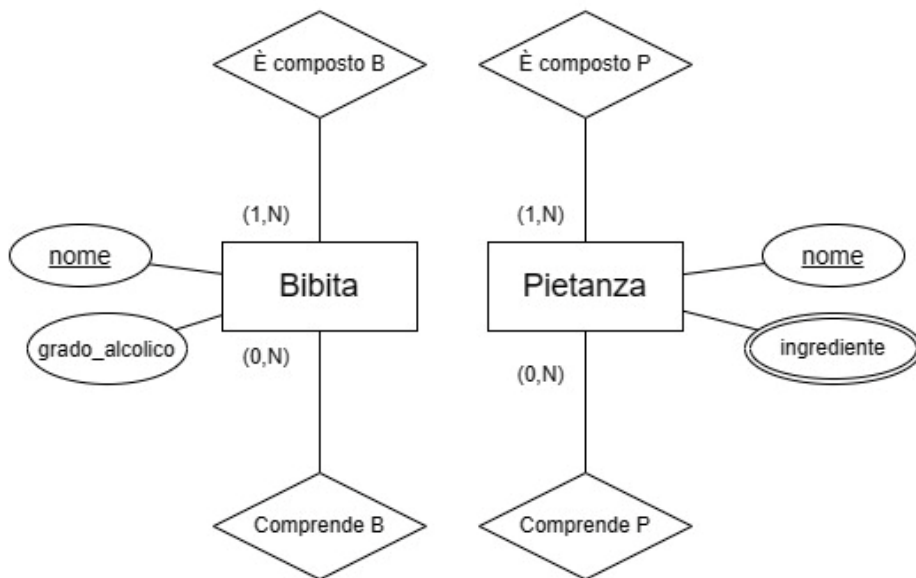
prezzo_tot è derivabile dalla somma dell'attributo prezzo (della relazione "È composto") di ogni prodotto compreso nell'ordine; tuttavia, tale valore deve essere memorizzato in una variabile al momento dell'ordine, poiché, se calcolato successivamente, il prezzo di qualche prodotto potrebbe essere stato modificato e quindi il risultato potrebbe risultare inconsistente rispetto al prezzo effettivo di quell'ordine.

Eliminazione delle gerarchie

Nello schema iniziale, è presente la specializzazione dell'entità "Prodotto":

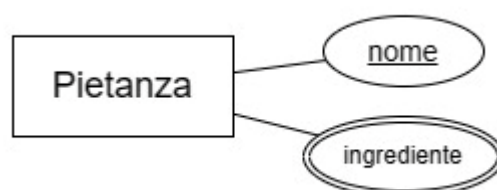


Il metodo risolutivo scelto è l'accorpamento dell'entità padre nelle figlie. Tale scelta è stata effettuata per evitare di memorizzare 200 attributi NULL, ridurre il numero degli accessi e mantenere i due concetti separati.

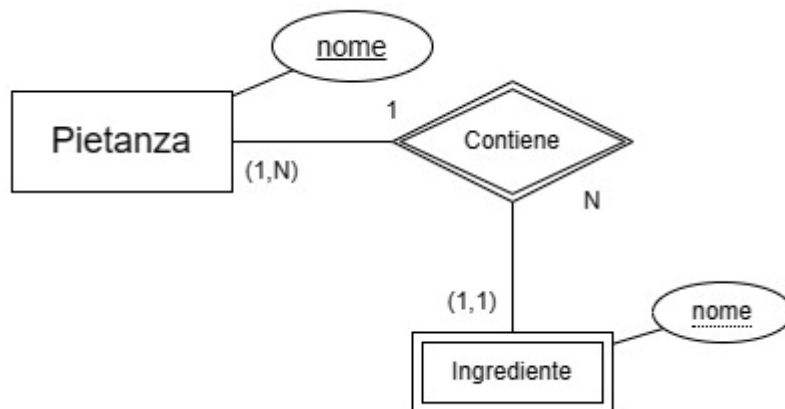


Eliminazione dell'attributo multivalore

Nello schema iniziale è presente l'attributo multivalore "ingrediente":



Per risolvere questa forma di attributo definiamo una nuova entità debole “Ingrediente” in relazione con “Pietanza”.

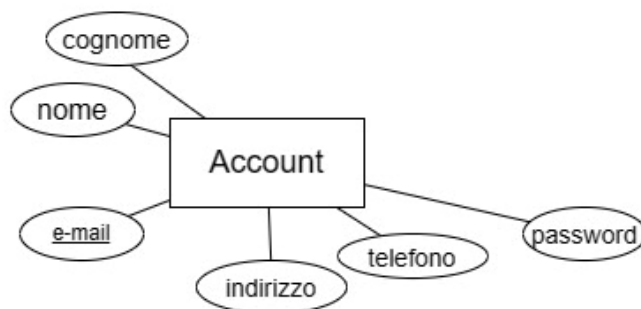


Eliminazione dell'attributo composto

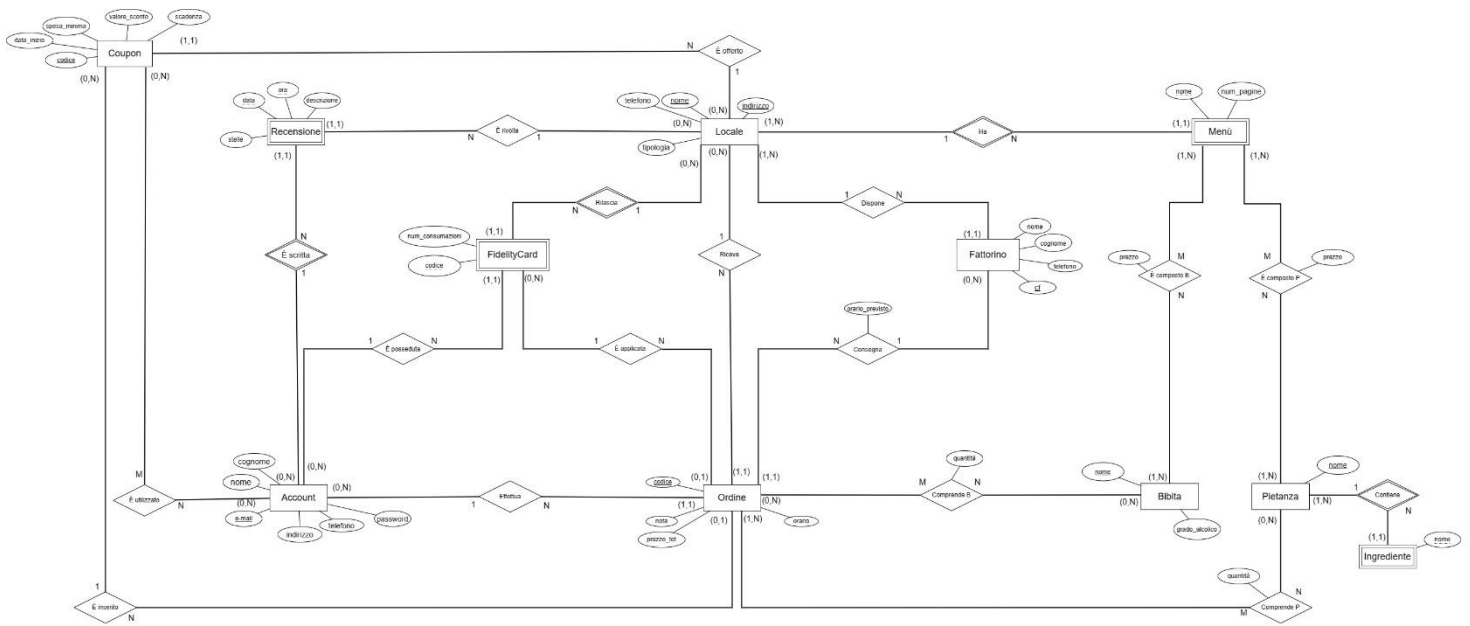
Nello schema iniziale è presente l'attributo composto “indirizzo”:



Per risolvere questa forma di attributo, raggruppiamo le sotto parti in “indirizzo”:



Schema EER ristrutturato



Schema relazionale

Mapping

Coupon (codice, data_inizio, spesa_minima, valore_sconto, scadenza, Locale.nome↑, Locale.indirizzo↑)

Locale (nome, indirizzo, telefono, tipologia)

Recensione (data, ora, Account.e-mail↑, descrizione, stelle, Locale.nome↑, Locale.indirizzo↑)

Menù (nome, Locale.nome↑, Locale.indirizzo↑, num_pagine)

FidelityCard (codice, Locale.nome↑, Locale.indirizzo↑, num_consumazioni, Account.e-mail↑)

Fattorino (cf, nome, cognome, telefono, Locale.nome↑, Locale.indirizzo↑)

Account (e-mail, nome, cognome, indirizzo, telefono, password)

Ordine (codice, nota, prezzo_tot, orario, Account.e-mail↑, Locale.nome↑, Locale.indirizzo↑, Fattorino.cf↑, orario_previsto)

Bibita (nome, grado_alcolico)

Pietanza (nome)

Ingrediente (nome, Pietanza.nome↑)

È utilizzato (Coupon.codice↑, Account.e-mail↑)

È inserito (Coupon.codice↑, Ordine.codice↑)

È applicata (FidelityCard.codice↑, FidelityCard.Locale.nome↑,
FidelityCard.Locale.indirizzo↑, Ordine.codice↑)

È composto B (Menù.nome↑, Menù.Locale.nome↑, Menù.Locale.indirizzo↑,
Bibita.nome↑, prezzo)

È composto P (Menù.nome↑, Menù.Locale.nome↑, Menù.Locale.indirizzo↑,
Pietanza.nome↑, prezzo)

Comprende B (Ordine.codice↑, Bibita.nome↑, quantità)

Comprende P (Ordine.codice↑, Pietanza.nome↑, quantità)

Normalizzazione

Il database si presenta già normalizzato.

Il database è in 1NF siccome non sono presenti attributi composti o multivalore. Infatti, l'attributo composto "indirizzo" dell'entità "Account" e l'attributo multivalore "ingrediente" dell'entità "Pietanza" sono stati rimossi in fase di ristrutturazione.

Il database è anche in 2NF poiché, oltre ad essere in 1NF, non sono presenti dipendenze parziali nelle relazioni dove la chiave primaria è composta da attributi multipli.

È anche in 3NF perché, oltre ad essere in 2NF, non esistono dipendenze transitive tra un attributo non chiave e la chiave primaria.

5. Realizzazione della base di dati con MySQL

```
1 • create schema ClickAndEat;
2 • use ClickAndEat;
3
4 • create table Locale (
5     nome varchar(30) NOT NULL,
6     indirizzo varchar(60) NOT NULL,
7     tipologia varchar(20) NOT NULL,
8     telefono varchar(15) NOT NULL,
9     PRIMARY KEY(nome,indirizzo)
10 );
11
12 • create table Coupon (
13     codice char(10) PRIMARY KEY NOT NULL,
14     data_inizio date NOT NULL,
15     spesa_minima double NOT NULL,
16     valore_sconto int NOT NULL,
17     scadenza date NOT NULL,
18     locale_nome varchar(30) NOT NULL,
19     locale_indirizzo varchar(60) NOT NULL,
20     FOREIGN KEY(locale_nome,locale_indirizzo) REFERENCES Locale(nome,indirizzo) ON UPDATE cascade ON DELETE cascade
21 );
22
23 • create table Menù (
24     nome varchar(15) NOT NULL,
25     locale_nome varchar(30) NOT NULL,
26     locale_indirizzo varchar(60) NOT NULL,
27     num_pagine int NOT NULL,
28     PRIMARY KEY (nome,locale_nome,locale_indirizzo),
29     FOREIGN KEY(locale_nome,locale_indirizzo) REFERENCES Locale(nome,indirizzo) ON UPDATE cascade ON DELETE cascade
30 );
31
32 • create table Fattorino (
33     cf char(16) PRIMARY KEY NOT NULL,
34     nome varchar(15) NOT NULL,
35     cognome varchar(15) NOT NULL,
36     telefono varchar(15) NOT NULL,
37     locale_nome varchar(30) NOT NULL,
38     locale_indirizzo varchar(60) NOT NULL,
39     FOREIGN KEY(locale_nome,locale_indirizzo) REFERENCES Locale(nome,indirizzo) ON UPDATE cascade ON DELETE cascade
40 );
41
42 • create table AccountUser (
43     email varchar(30) PRIMARY KEY NOT NULL,
44     passw varchar(20) NOT NULL,
45     nome varchar(30) NOT NULL,
46     cognome varchar(30) NOT NULL,
47     indirizzo varchar(60) NOT NULL,
48     telefono varchar(15) NOT NULL
49 );
50
51 • create table FidelityCard (
52     codice char(10) NOT NULL,
53     locale_nome varchar(30) NOT NULL,
54     locale_indirizzo varchar(60) NOT NULL,
55     num_consumazioni int NOT NULL DEFAULT 0,
56     account_email varchar(30) NOT NULL,
57     PRIMARY KEY (codice,locale_nome,locale_indirizzo),
58     FOREIGN KEY(locale_nome,locale_indirizzo) REFERENCES Locale(nome,indirizzo) ON UPDATE cascade ON DELETE cascade,
59     FOREIGN KEY(account_email) REFERENCES AccountUser(email) ON UPDATE cascade ON DELETE cascade
60 );
```



```

62 • create table Ordine (
63     codice char(10) PRIMARY KEY NOT NULL,
64     nota text(500),
65     prezzo_tot double NOT NULL,
66     orario timestamp DEFAULT current_timestamp NOT NULL,
67     account_email varchar(30) NOT NULL,
68     locale_nome varchar(30) NOT NULL,
69     locale_indirizzo varchar(60) NOT NULL,
70     fattorino_cf char(16) NOT NULL,
71     orario_previsto time NOT NULL,
72     FOREIGN KEY(account_email) REFERENCES AccountUser(email) ON UPDATE cascade ON DELETE cascade,
73     FOREIGN KEY(locale_nome,locale_indirizzo) REFERENCES Locale(nome,indirizzo) ON UPDATE cascade ON DELETE cascade,
74     FOREIGN KEY(fattorino_cf) REFERENCES Fattorino(cf) ON UPDATE cascade ON DELETE cascade
75 );
76
77 • create table Bibita (
78     nome varchar(20) PRIMARY KEY NOT NULL,
79     grado_alcolico int NOT NULL
80 );
81
82 • create table Pietanza (
83     nome varchar(30) PRIMARY KEY NOT NULL
84 );
85
86 • create table Ingrediente (
87     nome varchar(20) NOT NULL,
88     pietanza_nome varchar(30) NOT NULL,
89     PRIMARY KEY (nome,pietanza_nome),
90     FOREIGN KEY(pietanza_nome) REFERENCES Pietanza(nome) ON UPDATE cascade ON DELETE cascade
91 );
92
93 • create table Recensione (
94     data_ora timestamp DEFAULT current_timestamp NOT NULL,
95     account_email varchar(30) NOT NULL,
96     descrizione text(1000),
97     stelle enum('1','2','3','4','5') NOT NULL,
98     locale_nome varchar(30) NOT NULL,
99     locale_indirizzo varchar(60) NOT NULL,
100     PRIMARY KEY(data_ora,account_email),
101     FOREIGN KEY(account_email) REFERENCES AccountUser(email) ON UPDATE cascade ON DELETE cascade,
102     FOREIGN KEY(locale_nome,locale_indirizzo) REFERENCES Locale(nome,indirizzo) ON UPDATE cascade ON DELETE cascade
103 );
104
105 • create table È_utilizzato (
106     coupon_codice char(10) NOT NULL,
107     account_email varchar(30) NOT NULL,
108     PRIMARY KEY(coupon_codice,account_email),
109     FOREIGN KEY(coupon_codice) REFERENCES Coupon(codice) ON UPDATE cascade ON DELETE cascade,
110     FOREIGN KEY(account_email) REFERENCES AccountUser(email) ON UPDATE cascade ON DELETE cascade
111 );
112
113 • create table È_inserito (
114     coupon_codice char(10) NOT NULL,
115     ordine_codice char(10) NOT NULL,
116     PRIMARY KEY(coupon_codice,ordine_codice),
117     FOREIGN KEY(coupon_codice) REFERENCES Coupon(codice) ON UPDATE cascade ON DELETE cascade,
118     FOREIGN KEY(ordine_codice) REFERENCES Ordine(codice) ON UPDATE cascade ON DELETE cascade
119 );
120

```

```

121 • create table È_applicata (
122     fidelityCard_codice char(10) NOT NULL,
123     fidelityCard_locale_nome varchar(30) NOT NULL,
124     fidelityCard_locale_indirizzo varchar(60) NOT NULL,
125     ordine_codice char(10) NOT NULL,
126     PRIMARY KEY(fidelityCard_codice,fidelityCard_locale_nome,fidelityCard_locale_indirizzo,ordine_codice),
127     FOREIGN KEY(fidelityCard_codice,fidelityCard_locale_nome,fidelityCard_locale_indirizzo)
128         REFERENCES FidelityCard(codice,locale_nome,locale_indirizzo) ON UPDATE cascade ON DELETE cascade,
129     FOREIGN KEY(ordine_codice) REFERENCES Ordine(codice) ON UPDATE cascade ON DELETE cascade
130 );
131
132 • create table È_composto_B (
133     menù_nome varchar(15) NOT NULL,
134     menù_locale_nome varchar(30) NOT NULL,
135     menù_locale_indirizzo varchar(60) NOT NULL,
136     bibita_nome varchar(20) NOT NULL,
137     prezzo double NOT NULL,
138     PRIMARY KEY(menù_nome,menù_locale_nome,menù_locale_indirizzo,bibita_nome),
139     FOREIGN KEY(menù_nome,menù_locale_nome,menù_locale_indirizzo)
140         REFERENCES Menù(nome,locale_nome,locale_indirizzo) ON UPDATE cascade ON DELETE cascade,
141     FOREIGN KEY(bibita_nome) REFERENCES Bibita(nome) ON UPDATE cascade ON DELETE cascade
142 );
143
144 • create table È_composto_P (
145     menù_nome varchar(15) NOT NULL,
146     menù_locale_nome varchar(30) NOT NULL,
147     menù_locale_indirizzo varchar(60) NOT NULL,
148     pietanza_nome varchar(30) NOT NULL,
149     prezzo double NOT NULL,
150     PRIMARY KEY(menù_nome,menù_locale_nome,menù_locale_indirizzo,pietanza_nome),
151     FOREIGN KEY(menù_nome,menù_locale_nome,menù_locale_indirizzo)
152         REFERENCES Menù(nome,locale_nome,locale_indirizzo) ON UPDATE cascade ON DELETE cascade,
153     FOREIGN KEY(pietanza_nome) REFERENCES Pietanza(nome) ON UPDATE cascade ON DELETE cascade
154 );
155
156 • create table Comprende_B (
157     ordine_codice char(10) NOT NULL,
158     bibita_nome varchar(20) NOT NULL,
159     PRIMARY KEY(ordine_codice,bibita_nome),
160     FOREIGN KEY(ordine_codice) REFERENCES Ordine(codice) ON UPDATE cascade ON DELETE cascade,
161     FOREIGN KEY(bibita_nome) REFERENCES Bibita(nome) ON UPDATE cascade ON DELETE cascade
162 );
163
164 • create table Comprende_P (
165     ordine_codice char(10) NOT NULL,
166     pietanza_nome varchar(30) NOT NULL,
167     PRIMARY KEY(ordine_codice,pietanza_nome),
168     FOREIGN KEY(ordine_codice) REFERENCES Ordine(codice) ON UPDATE cascade ON DELETE cascade,
169     FOREIGN KEY(pietanza_nome) REFERENCES Pietanza(nome) ON UPDATE cascade ON DELETE cascade
170 );

```

```
171
172 • ALTER TABLE Comprende_B ADD quantità int DEFAULT 1;
173 • ALTER TABLE Comprende_P ADD quantità int DEFAULT 1;
```

6. Implementazione query SQL

Operazione 1

Creare un nuovo account

```
• INSERT INTO accountuser(email,passw,nome,cognome,indirizzo,telefono)
  VALUES(?, ?, ?, ?, ?, ?);
```

Operazione 2

Registrare un nuovo locale

```
• INSERT INTO locale(nome,indirizzo,tipologia,telefono)
  VALUES(?, ?, ?, ?);
```

Operazione 3

Registrare un nuovo fattorino

```
• INSERT INTO fattorino(cf,nome,cognome,telefono,locale_nome,locale_indirizzo)
  VALUES (?, ?, ?, ?, ?, ?);
```

Operazione 4

Scrivere una recensione

```
• INSERT INTO recensione(data_ora,account_email,descrizione,stelle,locale_nome,locale_indirizzo)
  VALUES (?, ?, ?, ?, ?, ?);
```

Operazione 5

Effettuare un ordine

```
• INSERT INTO ordine(codice,nota,prezzo_tot,orario,account_email,locale_nome,locale_indirizzo,fattorino_cf,orario_previsto)
  VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?);
```

Operazione 6

Inserire prodotto nel menù

Per inserire una pietanza:

```
• INSERT INTO è_composto_p
  VALUES(?, ?, ?, ?, ?);
```

Per inserire una bibita:

- **INSERT INTO** è_composto_b
VALUES(?, ?, ?, ?, ?);

Operazione 7

Sottoscrivere una fidelity card

- **INSERT INTO** fidelitycard(codice,locale_nome,locale_indirizzo,num_consumazioni,account_email)
VALUES (?, ?, ?, ?, ?);

Operazione 8

Visualizzare le fidelity card possedute da un account

- **SELECT** codice,locale_nome,num_consumazioni
FROM fidelitycard
WHERE account_email=?;

Operazione 9

Visualizzare il numero di ordini effettuati da un account

- **SELECT** count(*) AS num_ordini
FROM ordine
WHERE account_email=?;



Operazione 10

Visualizzare nome e numero del fattorino che consegna l'ordine

- **SELECT** nome,cognome,telefono
FROM fattorino,ordine
WHERE fattorino_cf=cf AND codice=?;

Operazione 11

Visualizzare i prodotti compresi in un ordine

-  (**SELECT** pietanza_nome AS prodotto,quantità
FROM ordine,comprende_p
WHERE ordine_codice=codice and codice=?)
UNION
 (**SELECT** bibita_nome,quantità
FROM ordine,comprende_b
WHERE ordine_codice=codice and codice=?);

Operazione 12

Offrire un nuovo coupon

- **INSERT INTO** coupon(codice,data_inizio,spesa_minima,valore_sconto,scadenza,locale_nome,locale_indirizzo)
VALUES(?, ?, ?, ?, ?, ?, ?);

Operazione 13

Visualizzare tutti i locali che hanno ricevuto recensioni con una media ≥ 3 stelle

- **SELECT** nome,telefono,avg(stelle) AS Valutazione_media
FROM recensione,locale
WHERE nome=locale_nome **AND** indirizzo=locale_indirizzo
GROUP BY nome,indirizzo
HAVING Valutazione_media ≥ 3 ;

Operazione 14

Selezionare tutte le pizzerie che si trovano in una data località

- **SELECT** nome,indirizzo,telefono
FROM locale
WHERE tipologia="Pizzeria" **AND** indirizzo **LIKE** "%?";

Operazione 15

Visualizzare i dati e l'orario previsto dell'ordine

- **SELECT** codice,locale_nome,prezzo_tot,orario **AS** data_e_orario,orario_previsto **AS** orario_consegna,nota
FROM ordine
WHERE codice=? ;

Operazione 16

Rimuovere un prodotto dal menù

Per rimuovere una pietanza:

- **DELETE**
FROM è_composto_p
WHERE menù_nome=? **AND** menù_locale_nome=? **AND** menù_locale_indirizzo=? **AND** pietanza_nome=;

Per rimuovere una bibita:

- **DELETE**
FROM è_composto_b
WHERE menù_nome=? **AND** menù_locale_nome=? **AND** menù_locale_indirizzo=? **AND** bibita_nome=;

Operazione 17

Visualizzare i menù di un locale

- ```
SELECT nome
FROM menù
WHERE locale_nome=? AND locale_indirizzo=?;
```

## Operazione 18

Modificare il prezzo di un prodotto

Per modificare il prezzo di una pietanza:

- ```
update è_composto_p set prezzo=? where pietanza_nome=? AND menù_locale_nome=? AND menù_locale_indirizzo=? AND menù_nome=?;
```

Per modificare il prezzo di una bibita:

- ```
update è_composto_b set prezzo=? where bibita_nome=? AND menù_locale_nome=? AND menù_locale_indirizzo=? AND menù_nome=?;
```

## Operazione 19

Lista delle bibite non alcoliche di un dato locale

```
. • SELECT nome
: FROM bibita,è_composto_b
: WHERE grado_alcolico=0 AND nome=bibita_nome AND menu_locale_nome=? AND menù_locale_indirizzo=?;
```

## Operazione 20

Visualizzare gli ingredienti di una pietanza

- ```
SELECT nome
FROM ingrediente
WHERE pietanza_nome=?;
```

Operazione 21

Visualizzare i prodotti di un menù

```
. • (SELECT bibita_nome AS nome,prezzo
: FROM è_composto_b
: WHERE menù_nome=? AND menù_locale_nome=? AND menù_locale_indirizzo=?)
: UNION
: (SELECT pietanza_nome,prezzo
: FROM è_composto_p
: WHERE menù_nome=? AND menù_locale_nome=? AND menù_locale_indirizzo=?);
```


Operazione 22

Visualizzare i locali che offrono sia coupon sia fidelity card

- ```
SELECT l.nome,l.indirizzo,l.telefono
FROM locale as l,coupon as c
WHERE l.nome=c.locale_nome
 AND l.indirizzo=c.locale_indirizzo
 AND (l.nome,l.indirizzo) IN (
 SELECT l1.nome,l1.indirizzo
 FROM locale as l1,fidelitycard as f
 WHERE l1.nome=f.locale_nome
 AND l1.indirizzo=f.locale_indirizzo
);
```

## Operazione 23

Visualizzare quante volte un locale ha ricevuto un ordine che comprende un dato prodotto in un certo mese.

- ```
SELECT count(o.codice) AS Ordinazioni_totali
FROM ordine AS o,locale
WHERE nome=? AND indirizzo=? AND nome=o.locale_nome AND indirizzo=o.locale_indirizzo AND o.orario LIKE "????-??%"
      AND (o.codice IN(
          SELECT o1.codice
          FROM ordine as o1,comprende_p
          WHERE o1.codice=ordine_codice AND pietanza_nome=?
      )
      OR o.codice IN
      (
          SELECT o2.codice
          FROM ordine AS o2,comprende_b
          WHERE o2.codice=ordine_codice AND bibita_nome=?
      )
);
```

7. Interfaccia per testare le query

The screenshot shows a window titled "REGISTRAZIONE" with a sidebar on the left containing buttons labeled Op1 through Op23. The main area contains a registration form with the following fields and labels:

- Inserisci nome:
- Inserisci cognome:
- Inserisci e-mail:
- Inserisci password:
- Inserisci indirizzo:
- Inserisci telefono:

A "Salva" button is located at the bottom right of the form area.

The screenshot shows a window titled "EFFETTUA UN ORDINE" with a sidebar on the left containing buttons labeled Op1 through Op23. The main area contains an order form with the following fields and labels:

- E-mail:
- Dati locale:
 - Nome:
 - Indirizzo:
- Nome menù pietanze:
- Nome menù bibite:
- Pietanze e quantità:
- Bibite e quantità:
- NOTA:

A "Continua" button is located at the bottom right of the form area.

Op1

Op2

Op3

Op4

Op5

Op6

Op7

Op8

Op9

Op10

Op11

Op12

Op13

Op14

Op15

Op16

Op17

Op18

Op19

Op20

Op21

Op22

Op23

LOCALI RECENSITI CON UNA MEDIA DI STELLE >=3

| nome | telefono | Valutazione_media |
|------------------|------------|-------------------|
| Sorbillo | 081446843 | 3.5 |
| Nonna Concettina | 3476945123 | 5.0 |

Op1

Op2

Op3

Op4

Op5

Op6

Op7

Op8

Op9

Op10

Op11

Op12

Op13

Op14

Op15

Op16

Op17

Op18

Op19

Op20

Op21

Op22

Op23

VISUALIZZA LE TUE FIDELITY CARD

Inserisci e-mail:

giuseppeverdi@gmail.com

Invio

| codice | locale_nome | num_consumazioni |
|------------|-------------|------------------|
| PIADINA150 | Da Pascoli | 9 |

Op1

Op2

Op3

Op4

Op5

Op6

Op7

Op8

Op9

Op10

Op11

Op12

Op13

Op14

Op15

Op16

Op17

Op18

Op19

Op20

Op21

Op22

Op23

ECCO COSA HAI ORDINATO:

Inserisci codice del tuo ordine:

ORDINE1111

Invio

| prodotto | quantità |
|------------------|----------|
| Pizza Diavola | 1 |
| Pizza Margherita | 1 |
| Birra Peroni | 1 |
| Coca Cola | 2 |

Op1

Op2

Op3

Op4

Op5

Op6

Op7

Op8

Op9

Op10

Op11

Op12

Op13

Op14

Op15

Op16

Op17

Op18

Op19

Op20

Op21

Op22

Op23

CERCA INTORNO A TE!

Inserisci la tua città:

Napoli

Cerca

| nome | indirizzo | telefono |
|----------|-----------------------------|-----------|
| Sorbillo | Via dei tribunali,32,80016, | 081446643 |

