



La Palme D'Or

Colella Maria Immacolata
Sorrentino Carlo
| Basi di dati 2021/2022 |

Progetto di Basi di Dati

- Ristorante La Palme D'Or-

1. Raccolta delle specifiche della realtà d'interesse

Si vuole progettare una base di dati per la gestione di un **ristorante**.

Ogni cliente viene identificato univocamente dal codice fiscale e si vuole memorizzare nome, cognome ed eventuali numeri di telefono.

Il cliente può effettuare una prenotazione di cui si vuole tener traccia del numero di persone al tavolo, se il posto è all'aperto o al coperto, data e il codice che la identifica.

Vi è il personale di sala che è identificato dal codice fiscale e del quale si vuole conoscere nome, cognome ed email aziendale.

Il personale di sala è composto unicamente da cassieri e camerieri. Ogni cliente è servito da un solo cameriere, mentre quest'ultimo può servire più clienti; si vuole tener traccia del numero di ordini serviti da ogni cameriere.

Per ogni ordine, richiesto da un cliente, si vuole memorizzare l'id dell'ordine (che lo identifica) ed eventuali modifiche apportate ad uno dei piatti ordinati.

Di ogni piatto, identificato da un id, si vuole conoscere il nome, il prezzo, la descrizione e gli allergeni.

Ogni chef prepara i piatti degli ordini che arrivano in cucina; di ognuno di essi si vuole memorizzare nome, cognome, il numero di eventuali Stelle Michelin, attestati di corsi di cucina e il codice fiscale che lo identifica.

Infine, il cassiere stampa lo scontrino del cliente, identificato univocamente da un numero di serie e di cui si vuole conoscere la data di emissione, l'importo e i dettagli delle ordinazioni effettuate.

Specifiche della realtà d'interesse

La realtà che andiamo a rappresentare riguarda la gestione di un ristorante che, a causa del notevole aumento e incremento di ordini e clienti nell'ultimo periodo, ha deciso di fornirsi di un database per la gestione dell'attività commerciale.

Per un'attività di ristorazione è importante tener traccia di tutto ciò che riguarda le entrate, le prenotazioni dei tavoli, il menù, gli impiegati... Questo database ha lo scopo di facilitare la visualizzazione e l'aggiornamento dei dati.

Nella realtà che andiamo a rappresentare, è fondamentale un personale di sala capace di interagire con la clientela in modo efficiente mostrando sempre disponibilità nel soddisfare le loro richieste.

In questo ristorante abbiamo sia addetti alla cassa che al servizio dei tavoli.

I cassieri si specializzano nel tener traccia degli scontrini e dei pagamenti dei clienti poiché la ricevuta del pagamento è fondamentale in quanto è l'unico mezzo per attestare le effettive entrate del ristorante.

I camerieri sono coloro che interagiscono maggiormente con i clienti e con gli chef; si occupano delle ordinazioni, del servizio ai tavoli e di lasciare un buon ricordo ai clienti in modo da invogliarli a ritornare al ristorante.

Il cliente è la componente fondamentale, egli può effettuare la prenotazione del tavolo in una determinata data avendo la possibilità di scegliere se mangiare all'aperto o all'interno. I clienti possono effettuare un numero indefinito di ordini scegliendo tra i piatti che compongono il menù creato appositamente dagli chef.

Questi hanno il compito di preparare i piatti con il massimo impegno così da soddisfare al meglio tutte le richieste inviate dal cameriere. Inoltre, gli chef hanno valide qualifiche quali stelle Michelin e attestati conseguiti a seguito di corsi di formazione per dimostrare la loro professionalità. Tutte le operazioni che si andranno ad effettuare saranno possibili solo su dati memorizzati e aggiornati alla fine della giornata lavorativa.

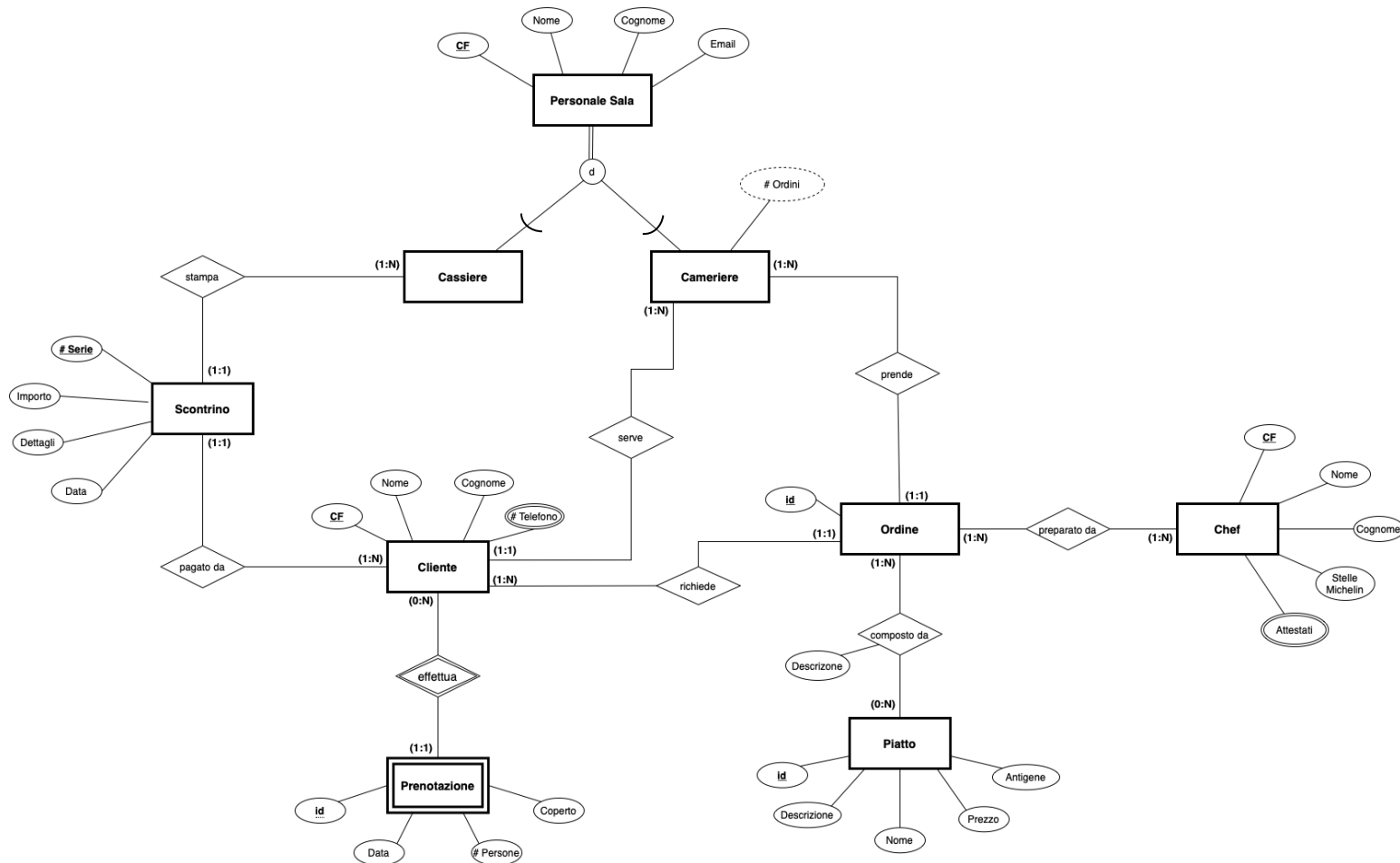
Glossario dei termini

Termine	Significato
Personale sala	Operatore che gestisce in maniera diretta le richieste della clientela.
Cassiere	Si occupa del pagamento dei clienti e dell'amministrazione degli scontrini.
Cameriere	Si occupa di portare gli ordini allo chef e di servire i piatti ai clienti.
Scontrino	Documento rilasciato dal registratore di cassa, che attesta il pagamento delle ordinazioni effettuate.
Cliente	Persona che acquisisce i servizi e i prodotti messi a disposizione dal ristorante.
Prenotazione	Documento attestante che il cliente ha occupato un tavolo per un giorno e un orario stabiliti.
Ordine	Comanda sulla quale si annotano piatti e bevande richieste dal cliente.
Piatto	Pietanze (cibo e bevande) che compongono il menù o l'eventuale ordine da parte di un cliente.
Chef	Personale incaricato della preparazione dei piatti richiesti dai clienti.

2. Progettazione concettuale della base di dati

Schema EER

Procedendo con la progettazione concettuale della base di dati, lo schema EER ottenuto è il seguente:



Dizionario delle entità

Legenda: sotto-entità, attributo multivalore, attributo ridondante, entità debole, chiave candidata

Entità	Descrizione	Attributi	Identificatore
Personale Sala	Dipendente che interagisce con la clientela	- CF - Nome - Cognome - Email	- CF
Cassiere	Si occupa del pagamento dei clienti	- CF - Nome - Cognome - Email	- CF
Cameriere	Si occupa di prendere gli ordini	- CF - Nome - Cognome - Email - # Ordini	- CF
Scontrino	Documento che attesta il pagamento	- # Serie - Data - Prezzo - Dettagli	- # Serie
Cliente	Persona che acquisisce i servizi	- CF_cliente - Nome - Cognome - # Telefono	- CF_cliente
Prenotazione	Un cliente occupa un tavolo	- Id_prenotazione - Data - # Persone - Coperto	- Id_prenotazione
Ordine	Comanda dei piatti ordinati	- Id_ordine	- Id_ordine
Piatto	Pietanza del menù	- Id_piatto - Nome - Descrizione - Prezzo - Allergeni	- Id_piatto
Chef	Persona che prepara i piatti	- CF_chef - Nome - Cognome - Stelle Michelin - Attestati	- CF_chef

Dizionario delle relazioni

Relazione	Descrizione	Entità coinvolte	Attributi
Stampa	Un cassiere stampa gli scontrini	Cassiere (1:N) Scontrino (1:1)	/
Pagato da	Un cliente paga uno scontrino	Scontrino (1:1) Cliente (1:N)	/
Effettua	Un cliente può effettuare un ordine	Cliente (0:N) Prenotazione (1:1)	/
Serve	Un cameriere serve un cliente	Cameriere (1:N) Cliente (1:1)	/
Prende	Un cameriere prende degli ordini	Cameriere (1:N) Ordine (1:1)	/
Composto da	Un ordine è composto da piatti	Ordine (1:N) Piatto (0:N)	- Descrizione
Preparato da	Un ordine è preparato dagli chef	Ordine (1:N) Chef (1:N)	/
Richiede	Un cliente richiede un ordine	Cliente (1:N) Ordine (1:1)	/

Vincoli non esprimibili nello schema

- L'attributo "coperto" dell'entità "Prenotazione" può valere 0 o 1 (0 -> esterno, 1 -> interno).

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

Tavola dei volumi

Concetto	Tipo	Carico applicativo
Personale sala	E	20
Cassiere	E	2
Cameriere	E	18
Scontrino	E	140
Cliente	E	90
Prenotazione	E	70
Ordine	E	270
Piatto	E	54
Chef	E	6
Stampa	R	140
Pagato da	R	140
Effettua	R	70
Serve	R	90
Prende	R	270
Composto da	R	1080
Preparato da	R	2160
Richiede	R	270

Ipotizziamo di avere 20 dipendenti addetti alla sala di cui 2 sono cassieri mentre la restante parte sono camerieri e di avere 6 chef.

Ipotizziamo di avere una media di 90 clienti al giorno.

Essendo che un cliente può usufruire dei servizi del ristorante più volte, il numero di scontrini sarà sicuramente superiore del numero di clienti, circa 140.

Dato che tra scontrino e la relazione stampa la partecipazione è 1:1, il carico applicativo della relazione sarà 140.

Lo stesso ragionamento vale per la relazione pagato da.

Il carico applicativo delle prenotazioni è sicuramente inferiore al numero di clienti poiché può verificarsi che alcuni clienti si presentino al ristorante senza aver prenotato. Infatti, anche la relazione effettua ha carico applicativo pari a 70.

Un cliente può effettuare più ordini nell'arco di una giornata, in media 3 ordini per cliente, quindi gli ordini saranno 270.

Dato che la partecipazione da ordine a cliente sulla relazione richiede è 1:1 il suo volume sarà pari a quello dell'entità ordine: 270.

La relazione serve tra cliente e cameriere ha carico 90 poiché la partecipazione da cliente è 1:1.

Lo stesso vale per la relazione prende, dato che il volume di ordine è 270, il carico della relazione sarà 270.

Abbiamo ipotizzato che nel menù ci saranno 54 piatti che lo compongono.

In media ogni ordine è composto da 4 piatti quindi la relazione composto da ha volume 270×4 cioè 1080.

Si suppone che la realizzazione di un piatto a volte possa richiedere la collaborazione di più chef quindi il carico applicativo di preparato da sarà di $1080 + 1080/2 = 1620$.

Tavola delle operazioni

	Operazioni	Tipo	Frequenza
1	Aggiungere un cliente	I	15/gg
2	Aggiungere una prenotazione	I	10/gg
3	Aggiungere un piatto	I	1/m
4	Un cameriere prende un ordine di un cliente	I	4/gg
5	Stampare, se esistono, il numero di stelle Michelin di uno chef	I	1/a
6	Stampare il numero di ordini totali preparati da ogni Chef	B	1/m
7	Stampare in ordine decrescente, per ogni cameriere, il numero di ordini presi	B	2/g
8	Trovare il guadagno totale di un mese scelto	B	4/gg
9	Stampare il nome e la descrizione dei piatti che hanno il prezzo inferiore a un importo dato dall'utente	B	2/gg
10	Stampare il codice fiscale, nome e cognome di tutti i cassieri che non hanno stampato nessuno scontrino	B	5/m

4. Progettazione logica

Analisi delle ridondanze

Il dato ridondante “#Ordini” appartiene all’entità “Cameriere”. Infatti, sarebbe possibile ottenere il numero di ordini presi attraverso le partecipazioni di Cameriere nella relazione “Cameriere prende Ordine”.

Supponiamo che l’attributo abbia peso 4 byte, essendo un intero, e consideriamo il volume dell’entità Cameriere pari a 18; il dato andrebbe ad occupare circa **72 byte**.

Per decidere se mantenere o meno il dato ridondante è necessario calcolare il numero di accessi con e senza ridondanza, per le operazioni che lo coinvolgono.

Le operazioni che coinvolgono “# Ordini” sono:

Op4) Quest’operazione va analizzata poiché ha un’alta frequenza e il dato ridondante andrebbe aggiornato;

Op7) Vale la stessa motivazione dell’op4.

Tavola degli accessi

Operazione 4

Calcolo con ridondanza

Calcolo senza ridondanza

	Concetto	Costrutto	Accesso	Tipo	Concetto	Costrutto	Accesso	Tipo
	Cliente	E	1	L				
	Serve	R	1	L	Cliente	E	1	L
	Cameriere	E	1	L	Richiede	R	3	S
	Cameriere	E	1	S				
	Richiede	R	3	S				
Totale	$(3L + 4S) \times 4 = (3 + 4 \times 2) \times 4 = 44$ a/gg				$(1L + 3S) \times 4 = (1 + 3 \times 2) \times 4 = 28$ a/gg			

Operazione 7

Calcolo con ridondanza

Calcolo senza ridondanza

	Concetto	Costrutto	Accesso	Tipo	Concetto	Costrutto	Accesso	Tipo
	Cameriere	E	18	L				
					Cameriere	E	18	L
					Prende	R	270	L
Totale	$(18L) \times 2 = 18 \times 2 = 36$ a/gg				$(288L) \times 2 = 288 \times 2 = 576$ a/gg			

Totale accessi con ridondanza

$$44 + 36 =$$

$$80 \text{ a/gg} + 72 \text{ byte}$$

Totale accessi senza ridondanza

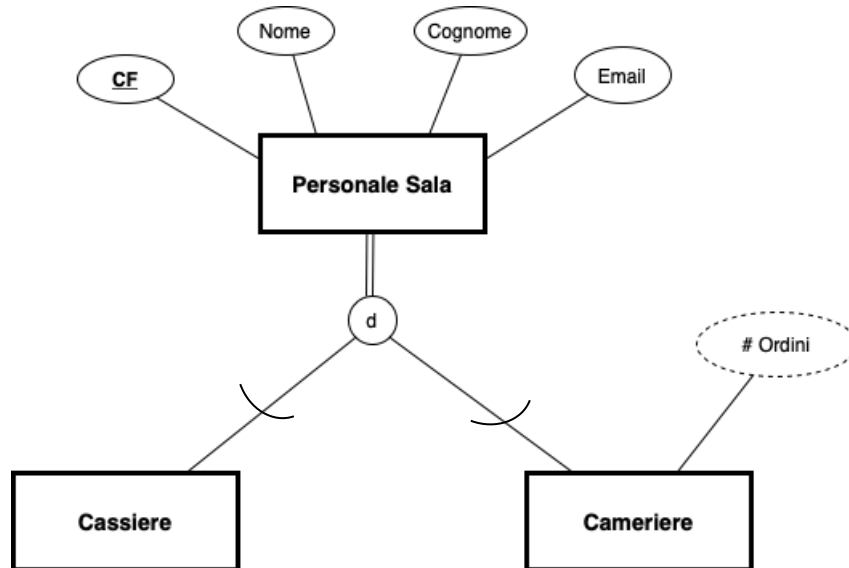
$$28 + 576 =$$

$$604 \text{ a/gg}$$

Dato che il numero di accessi è minore, conviene **mantenere** il dato ridondante “# Ordini”.

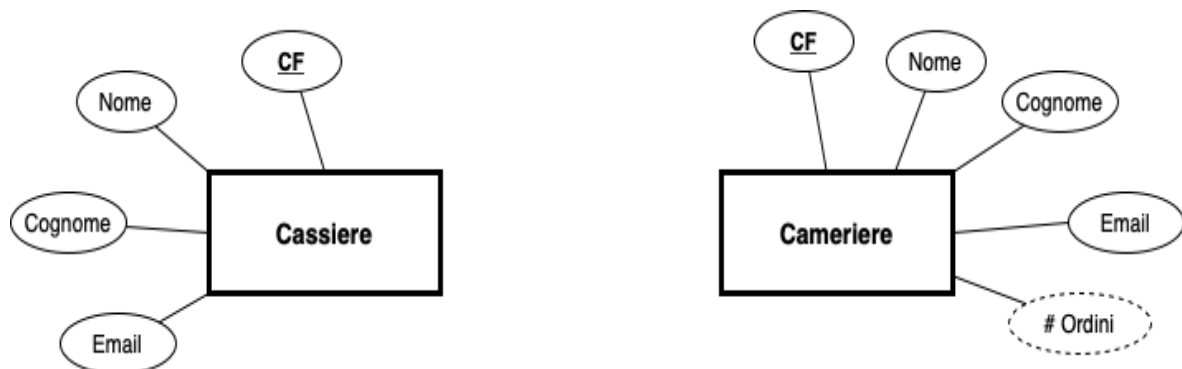
Eliminazione delle gerarchie

Nello schema EER, è presente la seguente specializzazione dell'entità "Personale Sala":



In questa fase è necessario ristrutturare in modo efficace per eliminare la gerarchia. La scelta migliore è quella di accorpare l'entità padre nelle entità figlie in quanto la superclasse non viene mai coinvolta nelle operazioni e la partecipazione è totale; se così non fosse avremmo delle occorrenze di "Prenotazione Sala" che non sarebbero state rappresentate.

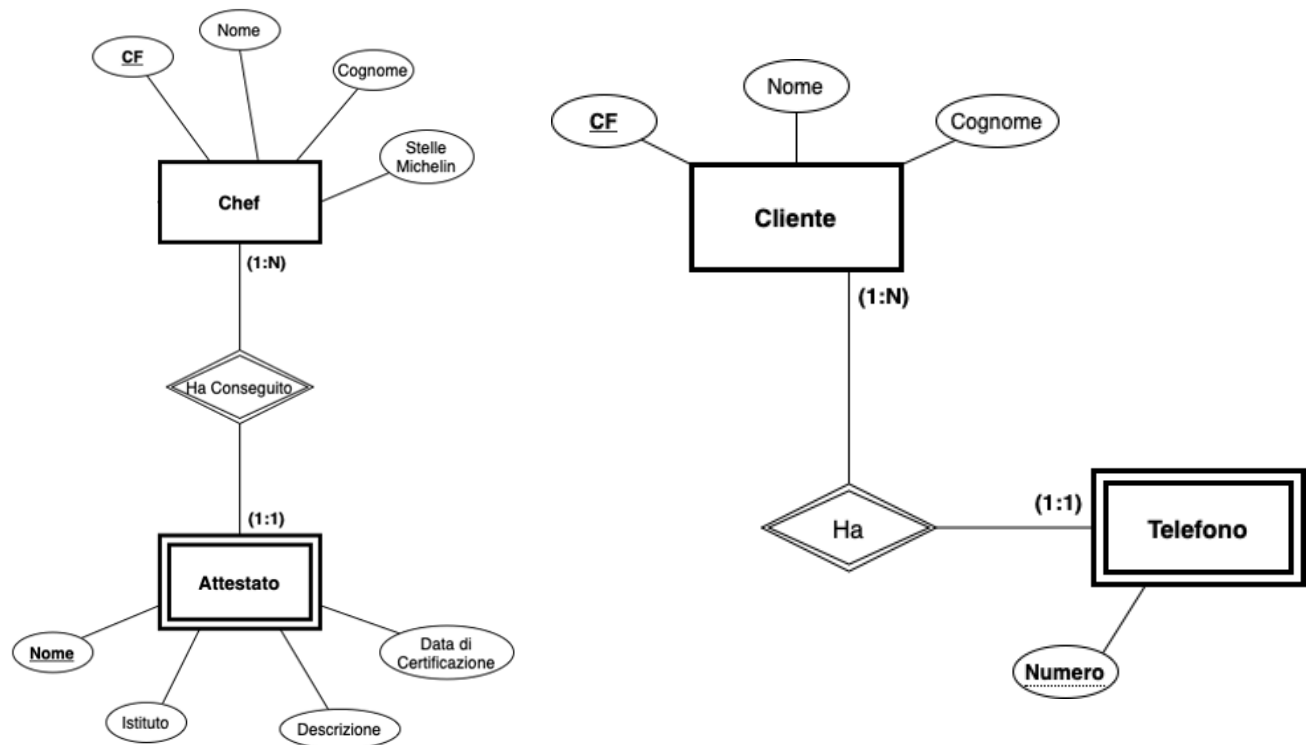
Ristrutturiamo:



Eliminazione dell'attributo multivalore

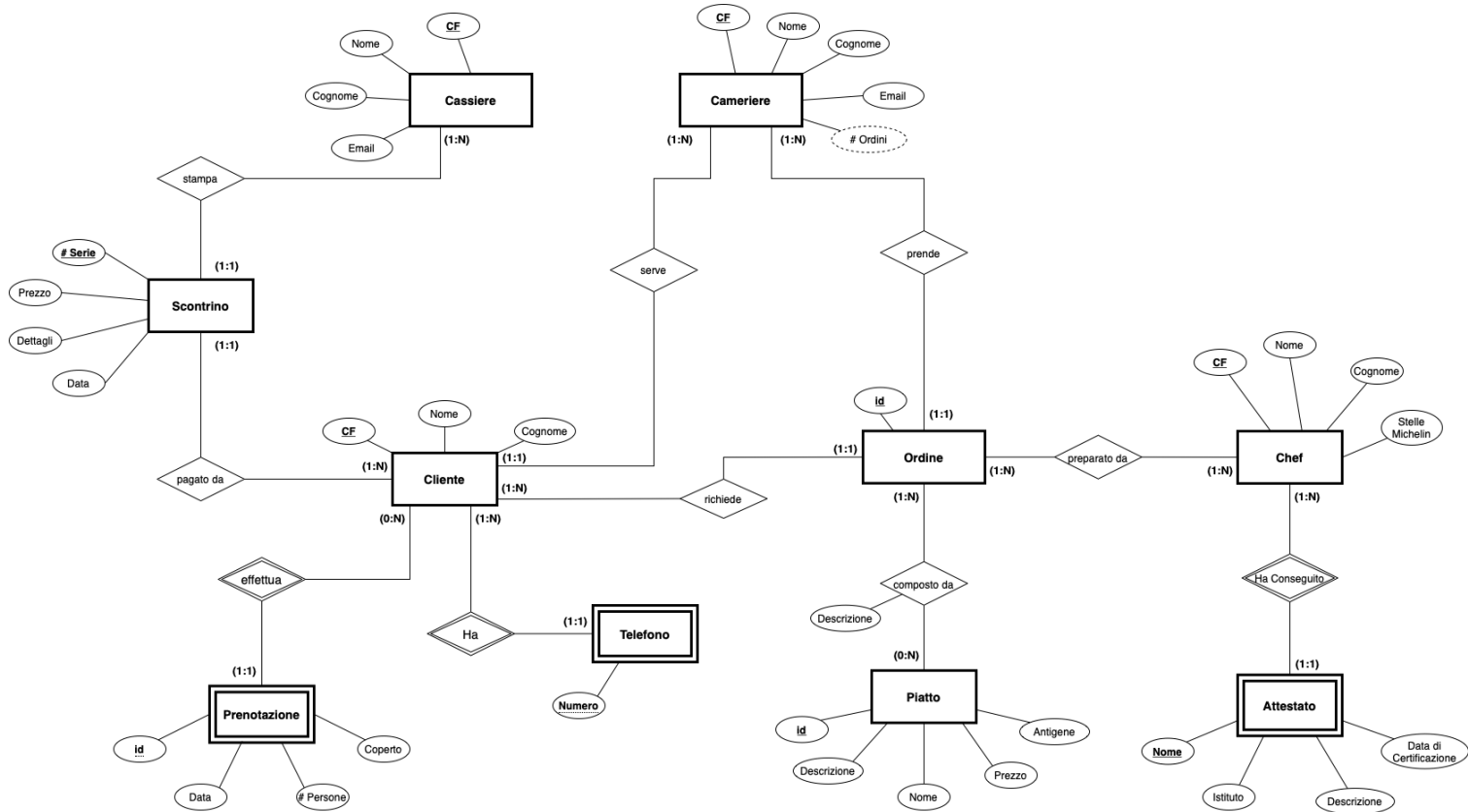
Nello schema EER compaiono due attributi multivalore: “# Telefono” dell’entità “Cliente” e “Attestati” dell’entità “Chef”. Questi tipi di attributi vanno ristrutturati creando due nuove entità:

- 1) Creo l’entità debole “Telefono” in relazione con Cliente;
- 2) Creo l’entità debole “Attestati” in relazione con Chef.



Schema EER ristrutturato

Al termine della fase di ristrutturazione, lo schema EER diventa:



Schema Relazionale

Si procede al mapping della base di dati:

```
>
> Cassiere (CF, nome, cognome, email)
> Cameriere (CF, nome, cognome, # ordine, email)
> Scontrino (# Serie, data, prezzo, dettagli, Cassiere.CF↑, Cliente.CF↑)
> Cliente (CF, nome, cognome, Cameriere.CF↑)
> Prenotazione (id, Cliente.CF↑, data, # persone, coperto)
> Telefono (Numero, Cliente.CF↑)
> Ordine (id, # tavolo, Cameriere.CF↑, Cliente.CF↑)
> Piatto (id, nome, descrizione, prezzo, allergeni)
> Composto da (Ordine.id↑, Piatto.id↑, descrizione)
> Chef (CF, nome, cognome, stelle michelin)
> Preparato da (Ordine.id↑, Chef.CF↑)
> Attestato (nome, ente, descrizione, anno conseguimento, Chef.CF↑)
>
```

Normalizzazione

Il database è in **prima forma normale** poiché non sono presenti attributi multivalore o composti (infatti gli attributi multivalore “attestati” dell’entità Chef e “telefono” dell’entità Cliente sono state già ristrutturare); tutti gli attributi sono atomici.

È in **seconda forma normale** perché è in 1NF e non sono presenti schemi di relazione in cui gli attributi dipendono parzialmente dalla chiave primaria (che è anche unica per ogni schema). Proprio per questo non è possibile avere un caso in cui la 2NF non venga rispettata.

È in **terza forma normale** poiché, oltre ad essere in 2NF, non sono presenti dipendenze transitive tra attributi non chiave e chiave primaria.

5. Realizzazione della base di dati con MySQL

Di seguito sono riportate le istruzioni utilizzate per creare il database con MySQL:

```
1 • create schema LaPalmedOr;
2 • use LaPalmedOr;
3
4 • CREATE TABLE Waiter
5 (
6     CF_waiter VARCHAR(16) PRIMARY KEY NOT NULL,
7     Name VARCHAR(32) NOT NULL,
8     Surname VARCHAR(32) NOT NULL,
9     Number_orders int DEFAULT(0) NOT NULL
10 );
11
12 • CREATE TABLE Cashier
13 (
14     CF_cashier VARCHAR(16) PRIMARY KEY NOT NULL,
15     Name VARCHAR(32) NOT NULL,
16     Surname VARCHAR(32) NOT NULL
17 );
18
19 • CREATE TABLE Customer
20 (
21     CF_customer VARCHAR(16) PRIMARY KEY NOT NULL,
22     Name VARCHAR(32),
23     Surname VARCHAR(32) NOT NULL,
24     CF_waiter VARCHAR(16) NOT NULL,
25     FOREIGN KEY (CF_waiter) REFERENCES Waiter(CF_waiter) ON UPDATE cascade ON DELETE cascade
26 );
27
28 • CREATE TABLE Receipt
29 (
30     SerialNumber_receipt INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
31     Date DATETIME NOT NULL,
32     Tot float NOT NULL,
33     Details TEXT(1024),
34     CF_cashier VARCHAR(16) NOT NULL,
35     CF_customer VARCHAR(16) NOT NULL,
36     FOREIGN KEY (CF_cashier) REFERENCES Cashier(CF_cashier) ON UPDATE cascade ON DELETE cascade,
37     FOREIGN KEY (CF_customer) REFERENCES Customer(CF_customer) ON UPDATE cascade ON DELETE cascade
38 );
39
40 • CREATE TABLE Reservation
41 (
42     Id_reservation INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
43     NumTable INT NOT NULL,
44     Date DATETIME NOT NULL,
45     Seated INT DEFAULT(0),
46     InDoor TINYINT DEFAULT(1),
47     CF_customer VARCHAR(16) NOT NULL,
48     FOREIGN KEY (CF_customer) REFERENCES Customer (CF_customer) ON UPDATE cascade ON DELETE cascade
49 );
50
```

```

51 • CREATE TABLE PhoneNumber
52 (
53     Number INT PRIMARY KEY NOT NULL,
54     CF_customer VARCHAR(16) NOT NULL,
55     FOREIGN KEY (CF_customer) REFERENCES Customer (CF_customer) ON UPDATE cascade ON DELETE cascade
56 );
57
58 • CREATE TABLE CustomerOrder
59 (
60     Id_customerOrder INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
61     CF_waiter VARCHAR(16) NOT NULL,
62     CF_customer VARCHAR(16) NOT NULL,
63     FOREIGN KEY (CF_waiter) REFERENCES Waiter(CF_waiter) ON UPDATE cascade ON DELETE cascade,
64     FOREIGN KEY (CF_customer) REFERENCES Customer(CF_customer) ON UPDATE cascade ON DELETE cascade
65 );
66
67
68 • CREATE TABLE Dish
69 (
70     Id_dish INT AUTO_INCREMENT PRIMARY KEY NOT NULL,
71     Name VARCHAR(64) NOT NULL,
72     Price FLOAT NOT NULL,
73     Description VARCHAR(256),
74     Allergens VARCHAR(64)
75 );
76
77 • CREATE TABLE Chef
78 (
79     CF_chef VARCHAR(16) PRIMARY KEY NOT NULL,
80     Name VARCHAR(32),
81     Surname VARCHAR(16) NOT NULL,
82     MichelinStar INT DEFAULT(0)
83 );
84
85 • CREATE TABLE ComposedBy
86 (
87     Id_customerOrder INT NOT NULL,
88     Id_dish INT NOT NULL,
89     Description VARCHAR(128),
90     FOREIGN KEY (Id_customerOrder) REFERENCES CustomerOrder(Id_customerOrder) ON UPDATE cascade ON DELETE cascade,
91     FOREIGN KEY (Id_dish) REFERENCES Dish(Id_dish) ON UPDATE cascade ON DELETE cascade
92 );
93
94 • CREATE TABLE MadeBy
95 (
96     Id_customerOrder INT NOT NULL,
97     CF_chef VARCHAR(16) NOT NULL,
98     FOREIGN KEY (Id_customerOrder) REFERENCES CustomerOrder (Id_customerOrder) ON UPDATE cascade ON DELETE cascade,
99     FOREIGN KEY (CF_chef) REFERENCES Chef(CF_chef) ON UPDATE cascade ON DELETE cascade

```

```

100 );
101
102 • CREATE TABLE Certification
103 (
104     Name VARCHAR(32) PRIMARY KEY NOT NULL,
105     Institution VARCHAR(32) NOT NULL,
106     Description VARCHAR(64) NOT NULL,
107     Date_of_certification DATE NOT NULL,
108     CF_chef VARCHAR(16) NOT NULL,
109     FOREIGN KEY (CF_chef) REFERENCES Chef (CF_chef)
110 );
111
112 • INSERT INTO Waiter(CF_waiter, Name, Surname, Number_orders)
113 VALUES
114     ("BNCPLA95M13F839H", "Adriano", "Verde", 6),
115     ("AAAAA01A018880", "Paolo", "Bianchi", 0),
116     ("RSSLGU87S07F205G", "Luigi", "Rossi", 0),
117     ("RNAGNN86H06A944W", "Giovanni", "Rana", 0),
118     ("FRRLCU80A01G702S", "Luca", "Ferrari", 0),
119     ("GLLRRT02P29L781J", "Roberto", "Gallo", 3),
120     ("FNTMHL89A07H501Q", "Michele", "Fontana", 0),
121     ("TTTFNC83A16D643M", "Francesco", "Totti", 0),
122     ("MRTMLN91R46L049I", "Melania", "Moretti", 2),
123     ("BRBNGL80A41E715L", "Angela", "Barbieri", 0),
124     ("CRRS092E58D969E", "Rosa", "Carota", 6),
125     ("CNCLCU87P05G388L", "Luca", "Cinacchi", 0),
126     ("NPLCRI97A08F839X", "Ciro", "Napoli", 0),
127     ("LNENNA91R24C351K", "Anna", "Leone", 0),
128     ("PRSSMN87L60G762W", "Simona", "Parisi", 0),
129     ("DRCGNN71H63G224I", "Giovanna", "D'Arco", 0),
130     ("DNGCLD80A41E791U", "Claudia", "De Angelis", 0),
131     ("MRTMRC80A01H703S", "Marco", "Martinelli", 0)
132 ;
133
134 • INSERT INTO Customer(CF_customer, Name, Surname, CF_waiter)
135 VALUES
136     ("CRUGLL01P60E791X", "Giallo", "Cuore", "BRBNGL80A41E715L"),
137     ("BSLMTT04D09B963F", "Mattia", "Basile", "GLLRRT02P29L781J"),
138     ("RSSMRA80A01G224W", "Mario", "Rossi", "LNENNA91R24C351K"),
139     ("RSSMRA80A41F205B", "Maria", "Rossi", "MRTMRC80A01H703S"),
140     ("PRRPLA80A41A783Z", "Paola", "Perrotta", "CNCLCU87P05G388L")
141 ;
142
143 • INSERT INTO Cashier(CF_cashier, Name, Surname)
144 VALUES
145     ("CLLMWM02B48E791O", "Maria Immacolata", "Colella"),
146     ("SRRRCRL01E31L245B", "Carlo", "Sorrentino")
147 ;
148

```



```

149 • INSERT INTO Dish(Name, Price, Description, Allergens)
150 VALUES
151     ("Gnocchi alla sorrentina", 18.00, "Gnocchi gratinati al forno, filanti e gustosi con pomodoro, grana e mozzarella!", "uova, latte, sesamo, glutine"),
152     ("Spaghetti alla nerano", 16.00, "Gli spaghetti alla Nerano sono un primo piatto realizzato con le zucchine fritte e provolone del Monaco.", "glutine, latte"),
153     ("Impepata di cozze", 14.00, "L'impepata di cozze è una ricetta della tradizione marinara napoletana.", "molluschi"),
154     ("Babà al rum", 20.00, "I babà al rum, o babbà, sono dolci della pasticceria napoletana tradizionale", "alcol, glutine, latte"),
155     ("Scialatielli ai frutti di mare", 18.00, "Un primo piatto ai frutti di mare semplice e adatto a tutte le stagioni.", "molluschi, glutine"),
156     ("Gamberi di Sicilia", 15.00, "Gamberi di Sicilia, tartare di mozzarella, verdure in agrodolce, colatura di cime di rapa.", "crostacei, latte, glutine"),
157     ("Linguine di Gragnano", 16.00, "Linguine di Gragnano, calamaretti, salsa al pane di segale", "crostacei, glutine"),
158     ("Spaghetto allo zafferano", 18.00, "Spaghetto allo zafferano, ricci di mare, quinoa croccante.", "glutine, pesce"),
159     ("Plin di anatra", 40.00, "Plin di anatra, zuppetta di fegato grasso, latte di bufala e lampone.", "latte, glutine"),
160     ("Sgombro", 28.00, "Sgombro, avocado, caviale e champagne.", "pesce, alcol"),
161     ("Piccione al Banyuls", 36.00, "Piccione, fegato grasso al grù di cacao, salsa al Banyuls.", "alcol, glutine, arachidi"),
162     ("Rombo chiodato", 31.00, "Rombo chiodato, zucca, spinaci e nocciole.", "glutine, frutta a guscio"),
163     ("Trancetto esotico", 21.00, "Dolce al cioccolato e frutti esotici.", "frutta a guscio, latte, glutine, uova"),
164     ("Cheesecake rafano e lampone", 17.00, "Lampone, rafano e lievito madre.", "latte, glutine, frutta a guscio")
165 ;
166
167 • INSERT INTO Chef(CF_chef, Name, Surname, MichelinStar)
168 VALUES
169     ("CNNNNW5D16F839V", "Antonino", "Cannavacciuolo", 4),
170     ("BRB8RN62A12F083G", "Bruno", "Barbieri", 7),
171     ("CRCCRL65R08D136M", "Carlo", "Cracco", 1),
172     ("LCTGRG63D07L682R", "Giorgio", "Locatelli", 1),
173
174     ("BRTNRC79A01G713I", "Enrico", "Bartolini", 9),
175     ("RMSGDN66S08L736A", "Gordon", "Ramsay", 16)
176 ;
177
178 • INSERT INTO CustomerOrder( CF_waiter, CF_customer)
179 VALUES
180     ("BNCPLA95M13F839H", "CRUGLL01P60E791X"),
181     ("GLLRRT02P29L781J", "BSLMTT04D09B963F"),
182     ("CRTRSO92E58D969E", "RSSMRA80A01G224W"),
183     ("BNCPLA95M13F839H", "RSSMRA80A01G224W"),
184     ("GLLRRT02P29L781J", "BSLMTT04D09B963F"),
185     ("AAAAAA01A01888O", "CRUGLL01P60E791X"),
186     ("PRSSMN87L60G762W", "RSSMRA80A01G224W"),
187     ("LNENNA91R24C351K", "RSSMRA80A01G224W"),
188     ("LNENNA91R24C351K", "BSLMTT04D09B963F"),
189     ("PRSSMN87L60G762W", "CRUGLL01P60E791X"),
190     ("GLLRRT02P29L781J", "RSSMRA80A41F205B"),
191     ("MRTMRC80A01H703S", "RSSMRA80A41F205B"),
192     ("MRTMRC80A01H703S", "BSLMTT04D09B963F"),
193     ("MRTMRC80A01H703S", "RSSMRA80A41F205B"),
194     ("MRTMRC80A01H703S", "RSSMRA80A41F205B"),
195     ("BNCPLA95M13F839H", "RSSMRA80A01G224W"),
196     ("NPLCRI97A08F839X", "BSLMTT04D09B963F"),
197     ("CRTRSO92E58D969E", "CRUGLL01P60E791X")

```

```

197 ;
198
199 • INSERT INTO MadeBy(Id_customerOrder, CF_chef)
200 VALUES
201     (1, "CINNINN75D16F839V"),
202     (2, "LCTGRG63D07L682R"),
203     (3, "BRBBRN62A12F083G"),
204     (4, "CRCCRL65R08D136M"),
205     (5, "LCTGRG63D07L682R"),
206     (6, "BRBBRN62A12F083G"),
207     (7, "CINNINN75D16F839V"),
208     (8, "LCTGRG63D07L682R"),
209     (9, "BRBBRN62A12F083G"),
210     (10, "CINNINN75D16F839V"),
211     (11, "LCTGRG63D07L682R"),
212     (12, "RMSGDN66S08L736A"),
213     (13, "LCTGRG63D07L682R"),
214     (14, "LCTGRG63D07L682R"),
215     (15, "BRBBRN62A12F083G"),
216     (16, "BRBBRN62A12F083G"),
217     (17, "RMSGDN66S08L736A"),
218     (18, "RMSGDN66S08L736A")
219 ;
220

```

```

221 • INSERT INTO Receipt(Date, Tot, Details, CF_cashier, CF_customer)
222 VALUES
223     ('2022-1-31 13:59:22', 51, "Gnocchi alla sorrentina, Spaghetti alla nerano, Cheesecake rafano e lampone", "SRRCL01E31L245B", "CRUGLL01P60E791X"),
224     ('2022-1-31 16:57:22', 70, "Piccione al Banyuls, Babà al rum, Impepata di cozze", "SRRCL01E31L245B", "BSLMTT04D098963F"),
225     ('2022-1-31 21:51:22', 58, "Spaghetto allo zafferano, Plin di anatra", "CLLMMW02B48E7910", "RSSMRA80A01G224W"),
226     ('2022-1-31 12:53:22', 51, "Gnocchi alla sorrentina, Spaghetti alla nerano, Cheesecake rafano e lampone", "CLLMMW02B48E7910", "RSSMRA80A41F205B"),
227     ('2022-2-1 15:52:22', 70, "Piccione al Banyuls, Babà al rum, Impepata di cozze", "CLLMMW02B48E7910", "PRRPLA80A41A783Z")
228 ;
229

```

6. Implementazione query SQL

Operazione1

```
INSERT INTO Customer (CF_customer, Name, Surname, CF_waiter)
VALUES
    ( ?, ?, ?, ?)
;
```

```
INSERT INTO PhoneNumber(Number, CF_customer)
VALUES
    (?, ?)
;
```

```
SELECT Number
FROM PhoneNumber
WHERE CF_customer=?;
```

Operazione2

```
SELECT CF_customer FROM Customer WHERE CF_customer = ?;
```

Questa select ricerca il cliente che deve effettuare la prenotazione;

```
INSERT INTO Reservation(Date, Seated, InDoor, CF_customer)
VALUES
    ( ?, ?, ?, ?)
;
```

Operazione3

```
INSERT INTO Dish (Name, Price, Description, Allergens)
VALUES
    ( ?, ?, ?, ?)
;
```

```
SELECT *
FROM Dish
WHERE Name=? AND Description=?;
```

Stampa la tupla del piatto appena aggiunto

Operazione4

```
SELECT CF_waiter  
FROM Customer  
WHERE CF_waiter=?
```

Seleziono il CF del cameriere in base al cliente;

```
INSERT INTO CustomerOrder(CF_waiter, CF_customer)  
VALUES  
    ( ?, ?)
```

;

Inserisco un nuovo ordine;

```
SELECT COUNT(*)  
FROM CustomerOrder
```

Sapendo quante tuple ho in CustomerOrder riesco a ricavare l'id dell'ordine essendo questo un intero autoincrement;

```
SELECT Id_customerOrder  
FROM CustomerOrder  
WHERE Id_customerOrder= ? AND CF_waiter= ? AND  
CF_customer= ?
```

Seleziono effettivamente l'id dell'ordine;

```
INSERT INTO ComposedBy(Id_customerOrder, Id_dish, Description)  
VALUES  
    (?, ?, ?)
```

;

Inserisco nello schema ComposedBy ogni piatto dell'ordine con chiave esterna pari all'ordine appena effettuato;

```
SELECT SUM(p.Price)  
FROM Dish AS p, ComposedBy AS c  
WHERE p.Id_dish = c.Id_dish AND Id_customerOrder;
```

Calcolo la somma dello scontrino

```
INSERT INTO Receipt(Date, Tot, CF_cashier, CF_customer)  
VALUES  
    (?, ?, ?, ?);
```

```
SELECT COUNT(*)  
FROM Receipt
```

Contando il numero di tuple presenti nello schema di relazione, deduciamo il numero di serie dello scontrino.

```
SELECT *  
FROM Receipt  
WHERE SerialNumber_receipt = ?;
```

Operazione5

```
SELECT Name, Surname, MichelinStar  
FROM Chef  
WHERE MichelinStar != 0 AND CF_chef=?;
```

Operazione6

```
SELECT c.CF_chef, c.Name, c.Surname, COUNT(*) AS nOrdini  
FROM Chef AS c, MadeBy AS m  
WHERE c.CF_chef = m.CF_chef  
GROUP BY CF_chef;
```

Operazione7

```
SELECT CF_waiter, Name, Surname, Number_orders  
FROM Waiter  
ORDER BY Number_orders DESC;
```

Operazione8

```
SELECT SUM(Tot) AS totM  
FROM Receipt  
WHERE Date like '?-?%';
```

Operazione9

```
SELECT Name, Price, Description, Allergens  
FROM Dish  
WHERE Price <= ?;
```

Operazione10

```
SELECT c.CF_cashier, c.Name, c.Surname  
FROM Cashier AS c  
WHERE NOT EXISTS (SELECT *  
                   FROM Receipt AS r  
                   WHERE c.CF_cashier=r.CF_cashier);
```