

Gestione di una mensa

Giovanni Fazi

0000768043

giovanni.fazi@studio.unibo.it

Michele Di Stefano

0000772273

michele.distefano@studio.unibo.it

Indice

Argomento	Pagina
1. Analisi dei requisiti	1
a. Requisiti espressi in linguaggio naturale	1
b. Glossario dei termini	2
c. Strutturazione dei requisiti	3
d. Specifica operazioni	4
2. Progettazione concettuale	4
2a. Identificazione delle entità e relazioni	5
2b. Un primo scheletro dello schema	5
2c. Sviluppo delle componenti dello scheletro	5
2d. Unione delle componenti nello schema finale ridotto	7
2e. Dizionario dei dati	7
3. Progettazione logica	10
3a. Tavole dei volumi e delle operazioni	10
3b. Ristrutturazione dello schema concettuale	11
3c. Normalizzazione	14
3d. Traduzione verso il modello relazionale	15
4. Codifica SQL	16
4a. Definizione dello schema	16
4b. Codifica delle operazioni	18
5. Bibliografia	21

1. Analisi dei requisiti

1a. Requisiti espressi in linguaggio naturale

Si vuole realizzare una base di dati per una mensa, di cui si desidera rappresentare i dati relativi agli alimenti, ai fornitori, ai clienti, alle carte fedeltà dei clienti, al personale, alle prenotazioni online ed ai menu'.

Per gli alimenti, identificati da un codice alimento, si vuole rappresentare il nome, la quantità, la data di scadenza ed il relativo fornitore.

Per i fornitori, identificati dalla ragione sociale, si vogliono rappresentare la quantità di merce consegnata, il codice della merce, il costo della merce, la data di consegna.

Per i clienti, identificati da un codice cliente, si vogliono rappresentare il nome, il cognome e l'identificativo della relativa carta fedeltà.

Per le carte fedeltà, si vuole rappresentare il codice cliente, i rispettivi punti accumulati e la data di scadenza dei punti.

Per le prenotazioni online si vuole rappresentare il codice identificativo del cliente, tavolo, data, ora, numero di persone, metodo di pagamento.

Per il personale si vuole rappresentare il codice fiscale, nome, cognome, stipendio, ruolo, durata contratto.

Per i menù è prevista la presenza di 3 tipologie diverse: vegano, senza glutine, normale.

Per ogni menù, identificato da un codice menù, si vogliono rappresentare le diverse portate: primo, secondo, contorno, dolce, bevanda; i relativi prezzi e l'apporto calorico.

In seguito ad una seconda intervista è stato specificato che:

- gli alimenti devono essere identificati anche dall'apporto calorico
- più fornitori possono consegnare gli stessi alimenti
- i clienti memorizzati nella base di dati sono solamente quelli registrati al sito web della mensa
- le singole pietanze dei 3 menù fissi possono cambiare

1b. Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Alimento	Prodotto consegnato dai fornitori	Vivanda	Fornitore, menù
Cliente	Chi acquista un prodotto o un servizio dietro corrispettivo	Acquirente	Menù, carta fedeltà, prenotazioni online, Personale
Carta Fedeltà	Carta punti associata ad ogni cliente registrato online	-	Cliente
Fornitore	Persona o società che fornisce un carico di alimenti.	Grossista	Alimenti, Personale
Menù	Elenco cibi e bevande ordinabili	Pasto completo	Cliente, prenotazioni online, Alimenti
Prenotazione online	Procedura online per prenotare posti a sedere in mensa	Riservazione	Cliente, carta fedeltà, menù
Personale	Chief e servienti cibo, addetti alla pulizia	Dipendenti	Cliente, Fornitori

1c. Specifica operazioni

1. Inserire un nuovo fornitore (1 volta ogni tre anni)
2. Inserire un nuovo dipendente (1 volta ogni quattro anni)
3. Inserire un nuovo alimento (in media 5 volte all'anno)
4. Inserire un nuovo utente registrato (in media 20 volte al mese)
5. Assegnare una prenotazione ad un cliente (in media 100 volte al giorno)
6. Calcolare l'apporto calorico totale dei menù giornalieri (1 volta al giorno)
7. Calcolare il totale delle entrate mensili (in media 1 volta al mese)
8. Far visualizzare ai dipendenti quali alimenti saranno consegnati un determinato giorno (in media 16 volte al giorno)
9. Visualizzare uno dei tre menù (in media 90 volte al giorno)
10. Visualizzare la quantità di un alimento presente in magazzino (in media 5 volte al mese)
11. Visualizzare gli alimenti in scadenza entro 3 giorni dal giorno corrente (in media 2 volte a settimana).
12. Effettuare una statistica su quanti punti hanno i clienti mediamente (1 volta al mese)
13. Visualizzare i membri del personale con contratto in scadenza in meno di un mese (in media 1 volta al mese)

1d. Strutturazione dei requisiti

❖ Frasi di carattere generale

Si vuole realizzare una base di dati per una mensa, di cui si desidera rappresentare i dati relativi agli alimenti, ai fornitori, ai clienti, alle loro carte fedeltà, al personale, alle prenotazioni online ed ai menu'.

❖ Frasi relative agli alimenti

Per gli alimenti, identificati da un codice alimento, si vuole rappresentare il nome, la quantità, la data di scadenza, il relativo fornitore e l'apporto calorico.

❖ Frasi relative ai fornitori

Per i fornitori, identificati dalla ragione sociale, si vogliono rappresentare la quantità di merce consegnata, il codice della merce, il costo della merce, la data di consegna periodica.

❖ Frasi relative ai clienti

Per i clienti, identificati da un codice cliente, si vogliono rappresentare il nome, il cognome e l'identificativo della relativa carta fedeltà.

❖ Frasi relative alle carte fedeltà

Per le carte fedeltà, si vuole rappresentare il codice cliente, i rispettivi punti accumulati e la data di scadenza dei punti.

❖ Frasi relative al personale

Per il personale si vuole rappresentare il codice fiscale, nome, cognome, stipendio, ruolo, durata contratto ed il rispettivo.

❖ Frasi relative alle prenotazioni online

Per le prenotazioni online si vuole rappresentare il codice identificativo del cliente, tavolo,

data, ora, numero di persone, metodo di pagamento.

❖ **Frase relative ai menù**

Per ogni menù, identificato da un codice menù, si vogliono rappresentare le diverse portate: primo, secondo, contorno, dolce, acqua, frutta; i relativi prezzi e l'apporto calorico.

2. Progettazione concettuale

Abbiamo seguito una strategia mista come sotto descritto.

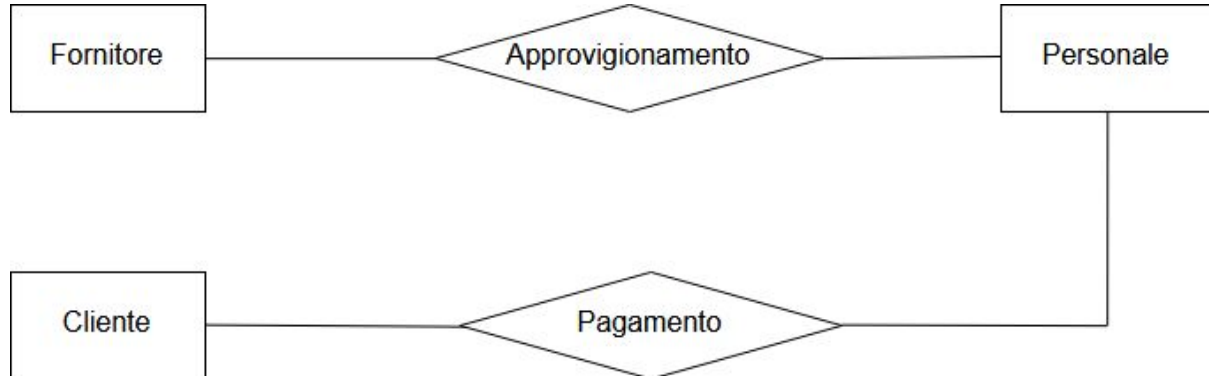
2a. Identificazione delle entità e relazioni (bottom-up)

Sono state identificate (seguendo la strategia bottom-up) inizialmente le seguenti entità: Cliente, Fornitore, Personale, Prenotazione online, Pasto completo, Alimenti, Carta fedeltà.

Le entità sopra elencate si possono suddividere in tre gruppi principali: Clienti che acquistano un pasto completo, Personale che include dagli chief agli addetti al magazzino alimentare, Fornitori che consegnano la merce.

2b. Un primo scheletro dello schema (top-down)

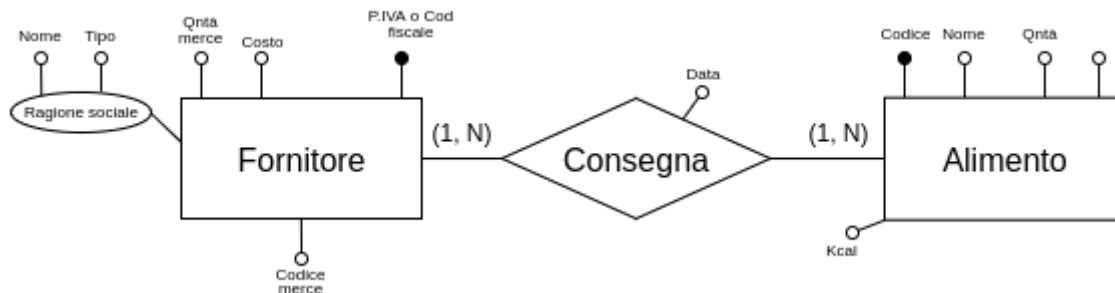
Ad un primo livello di astrazione, tenendo conto delle considerazioni fatte sopra è stato concepito il seguente primo scheletro di schema concettuale:



Per pagamento viene inteso il momento in cui il cliente arriva alla cassa e paga per il pasto che gli è stato preparato (a seguito di prenotazione online o di ordinazione sul momento).

2c. Sviluppo delle componenti dello scheletro (inside-out)

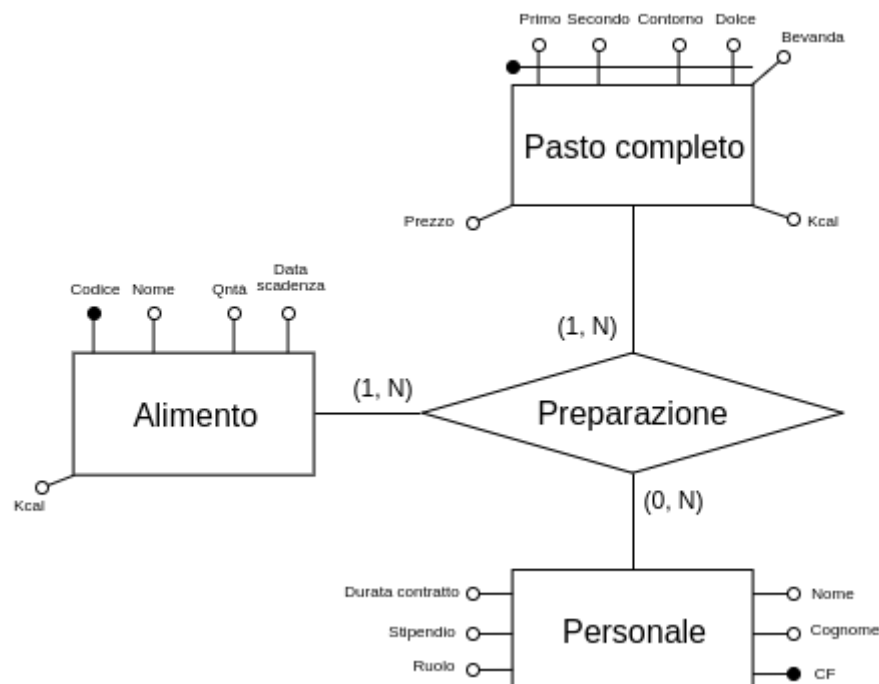
Fornitore:



L'approvvigionamento avviene tramite consegna di alimenti, è stato deciso che ci possono essere più fornitori per lo stesso alimento in quanto la mensa cambia le forniture periodicamente. L'associazione contiene l'attributo data, sarà usato per l'operazione 7.

E' stata utilizzata la relazione consegna del tipo molti-a-molti per catturare il fatto che più di un fornitore diverso possa provvedere alla consegna dei diversi alimenti, e che ogni alimento possa essere consegnato da uno o più fornitori.

Personale:



La scelta di una relazione ternaria è sembrata la più appropriata per modellare la associazione fra pasti, personale ed alimenti, in quanto le entità se considerate a coppie distinte, devono coinvolgere anche la terza entità nell' associazione:

(Alimento + Personale) permettono di preparare un Pasto Completo.

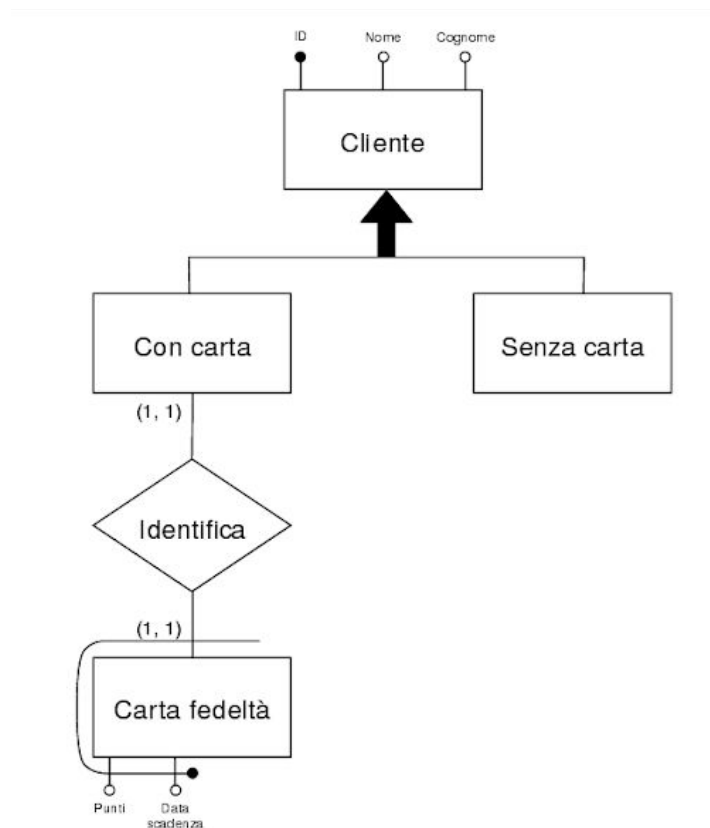
(Alimento + Pasto Completo) partecipano alla Preparazione tramite un cuoco del Personale.

(Personale + Pasto Completo) partecipano alla Preparazione utilizzando degli alimenti.

Le cardinalità sono state scelte considerando che:

- Non tutti gli addetti del Personale partecipano alla Preparazione (alcuni lavorano alla cassa, altri al magazzino)
- Uno o più alimenti possono concorrere alla preparazione di un pasto
- Dalla preparazione possono essere prodotti uno o più pasti completi

Cliente:

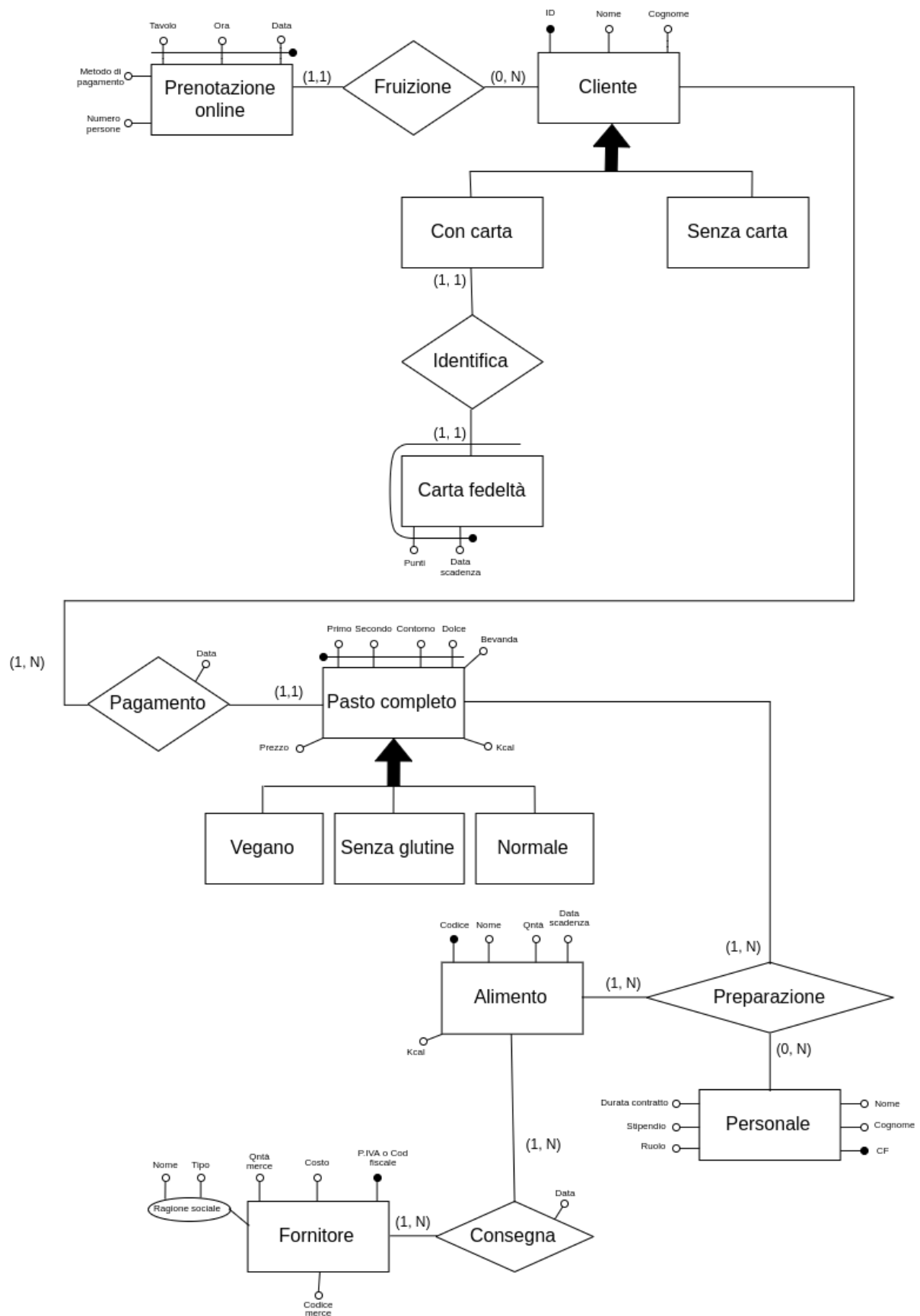


Il cliente è stato rappresentato utilizzando una generalizzazione, le cui entità figlie sono rispettivamente cliente con carta fedeltà e cliente senza carta.

Un cliente può richiedere una sola carta al massimo, perciò è stata usata la relazione uno ad uno.

Inoltre la carta contiene il codice identificativo del cliente, che viene generato nel momento in cui si è iscritto al sito della mensa, perciò è possibile utilizzarlo come identificatore esterno.

2d. Unione delle componenti nello schema finale



A partire dallo schema bozza iniziale, durante la realizzazione degli schemi ER ed utilizzando la metodologia inside-out, è stato deciso di introdurre una relazione ternaria tra le entità: pasto completo, alimento e personale.

Questa scelta ha portato a modificare la struttura dello schema ER iniziale nel quale erano previste soltanto le relazioni approvvigionamento e pagamento.

Nello schema finale compare inoltre la entità “Prenotazione Online” che si collega tramite “Fruizione” con il cliente . Ogni prenotazione viene effettuata a nome di un cliente registrato, abbiamo quindi usato la cardinalità 1 ad 1 per rappresentare la partecipazione della entità Prenotazione Online all’ associazione Fruizione. Viceversa non tutti i clienti registrati hanno eseguito una prenotazione (si pensi ai clienti appena registrati) ed inoltre un cliente può aver fatto più prenotazioni perciò abbiamo usato la cardinalità 0,N per rappresentare la partecipazione di Cliente in Fruizione.

3. Progettazione logica

3a. Tavole dei volumi e delle operazioni

Tavola dei volumi:

Concetto	Tipo	Volume
Prenotazione Online	E	720
Cliente	E	240
Cliente con carta	E	96
Cliente senza carta	E	144
Carta fedelta’	E	96
Pasto completo	E	3
Pasto completo Vegano	E	1
Pasto completo Senza Glutine	E	1
Pasto completo Normale	E	1
Personale	E	10
Alimento	E	30
Fornitore	E	10
Fruizione	R	700
Identifica	R	96
Pagamento	R	700
Preparazione	R	1000
Consegna	R	40

I volumi sono stati ottenuti in base alle seguenti stime:

- Prenotazione Online: per ogni cliente in media si hanno circa 3 prenotazioni online
- Cliente: clienti registrati in media all'anno (20 al mese)
- Pasto completo: numero di pasti offerti (somma delle 3 tipologie)
- Fruizione: circa il numero di prenotazioni, questo perché si suppone che non tutti i clienti registrati hanno effettuato obbligatoriamente una prenotazione
- Pagamento: dal momento che i clienti considerati sono quelli registrati online (non consideriamo quindi i clienti non registrati) stimiamo solo i pagamenti effettuati in seguito alle prenotazioni

- Preparazione: il numero di pasti completi preparati in media all'anno (contando anche quelli preparati per i clienti che non sono registrati), ovvero 400 prenotazioni annuali + 600 pasti per clienti che non prenotano
- Consegna: ogni fornitore consegna in media 4 alimenti, abbiamo quindi $10 \cdot 4$ occorrenze di consegna.

Tavola delle operazioni:

Operazione	Tipo	Frequenza
1 (Inserire un nuovo fornitore)	I	1 volta ogni 3 anni
2 (Inserire un nuovo dipendente)	I	1 volta ogni 4 anni
3 (Inserire un nuovo alimento)	I	5 volte all'anno
4 (Inserire un nuovo utente registrato)	I	20 volte al mese
5 (Assegnare una prenotazione ad un cliente)	I	100 volte al giorno
6 (Calcolare l'apporto calorico totale dei menù giornalieri)	I	1 volta al giorno
7 (Calcolare entrate mensili)	B	1 volta al mese
8 (Visualizzare gli alimenti che saranno consegnati un determinato giorno)	I	16 volte al giorno
9 (Visualizzare uno dei tre menù)	I	90 volte al giorno
10 (Visualizzare la qtà di un alimento in magazzino)	I	5 volte al mese
11 (Visualizzare gli alimenti in scadenza entro 3 giorni)	I	2 volte a settimana
12 (statistica su quanti punti hanno i clienti in media)	B	1 volta al mese
13 (Visualizzare i membri del personale con contratto in scadenza in meno di un mese)	B	1 volta al mese

3b. Ristrutturazione dello schema concettuale

Tavole degli accessi in presenza di ridondanza:

Operazione 1 Inserire un nuovo fornitore			
Concetto	Costrutto	Accessi	Tipo
Fornitore	Entità	1	W

Operazione 2 Inserire un nuovo dipendente			
Concetto	Costrutto	Accessi	Tipo
Personale	Entità	1	W

Operazione 3 Inserire un nuovo alimento			
Concetto	Costrutto	Accessi	Tipo
Alimento	Entità	1	W

Operazione 4 Inserire un nuovo utente registrato			
Concetto	Costrutto	Accessi	Tipo
Cliente	Entità	1	W

Operazione 5 Assegnare una prenotazione ad un cliente			
Concetto	Costrutto	Accessi	Tipo
Cliente	Entità	1	R
Fruizione	Relazione	1	W
Prenotazione online	Entità	1	W

Operazione 6 Calcolare l'apporto calorico totale dei menù giornalieri			
Concetto	Costrutto	Accessi	Tipo
Pasto completo	Entità	3	R
Preparazione	Relazione	3	R
Alimento	Entità	15	R
Pasto completo	Entità	3	W

Operazione 7 Calcolare entrate mensili			
Concetto	Costrutto	Accessi	Tipo
Prenotazione online	Entità	3000	R
Fruizione	Relazione	3000	R
Cliente	Entità	1000	R
Pagamento	Relazione	3000	R
Pasto completo	Entità	3000	R

Operazione 8 visualizzare gli alimenti che saranno consegnati un determinato giorno			
Concetto	Costrutto	Accessi	Tipo
Fornitore	Entità	7	R
Consegna	Relazione	7	R
Alimento	Entità'	28	R

Operazione 9 Visualizzare uno dei tre menù			
Concetto	Costrutto	Accessi	Tipo
Pasto completo	Entità	1	R

Operazione 10 Visualizzare la qtà di un alimento in magazzino			
Concetto	Costrutto	Accessi	Tipo
Alimento	Entità	1	R

Operazione 11 Visualizzare gli alimenti in scadenza entro 3 giorni			
Concetto	Costrutto	Accessi	Tipo
Alimento	Entità	30	R

Operazione 12 Statistica su quanti punti hanno i clienti in media			
Concetto	Costrutto	Accessi	Tipo
Cliente	Entità	320	R
Identifica	Associazione	320	R
Carta fedeltà	Entità	320	R

Operazione 13 Visualizzare i membri del personale con contratto in scadenza in meno di un mese			
Concetto	Costrutto	Accessi	Tipo
Personale	Entità	10	R

Eliminazione delle ridondanze:

E' stata riscontrata la presenza di un attributo derivabile da attributi di altre entità.

L'attributo Kcal dell'entità PastoCompleto si può derivare, tramite la relazione Preparazione, dall'attributo Kcal dell'entità Alimento, sommando le Kcal degli alimenti di cui un pasto è composto.

Prima di procedere con l'eventuale eliminazione del dato ridondante è necessario valutare gli indici di prestazione in merito alle operazioni 6 e 9:

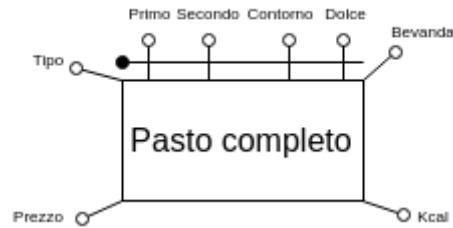
- **In presenza di dato ridondante** assumendo che il numero di Kcal di un PastoCompleto richieda 4 byte per essere memorizzato, si ha che il dato ridondante necessita di $3 \times 4 = 12$ byte di memoria aggiuntiva. Osservando la tavola degli accessi, l'operazione 9 richiederebbe 90 accessi giornalieri in lettura all'entità PastoCompleto per far visualizzare le kcal di un pasto. Il costo dell'operazione 6 sarebbe invece di 27 accessi (il costo degli accessi in scrittura è considerato doppio), ma si pagherebbe una sola volta al giorno. Per un totale quindi di 117 accessi al giorno in presenza di dato ridondante.
- **In assenza di dato ridondante** il calcolo delle calorie di un pasto verrebbe eseguito ogni volta che si effettua un accesso in lettura all'entità PastoCompleto e, considerando una media di 10 accessi (numero alimenti diviso numero pasti) in lettura all'associazione Preparazione per calcolare il numero di alimenti del singolo pasto e di 10 accessi in lettura all'entità Alimento (per leggere il valore dell'attributo Kcal del singolo alimento), ad ogni esecuzione si hanno $90 \times 20 = 1800$ accessi in lettura a fronte di un risparmio di 12 Bytes.

Pertanto in questo caso conviene mantenere il dato ridondante in quanto gli accessi in lettura necessari per calcolare il dato derivato sono molti di più rispetto a quelli da effettuare in sua assenza.

Eliminazione delle generalizzazioni:

La generalizzazione dell'entità Pasto Completo si può eliminare tramite il metodo di accorpamento figli nel genitore, in quanto le operazioni non fanno distinzione tra gli attributi di figli e genitori.

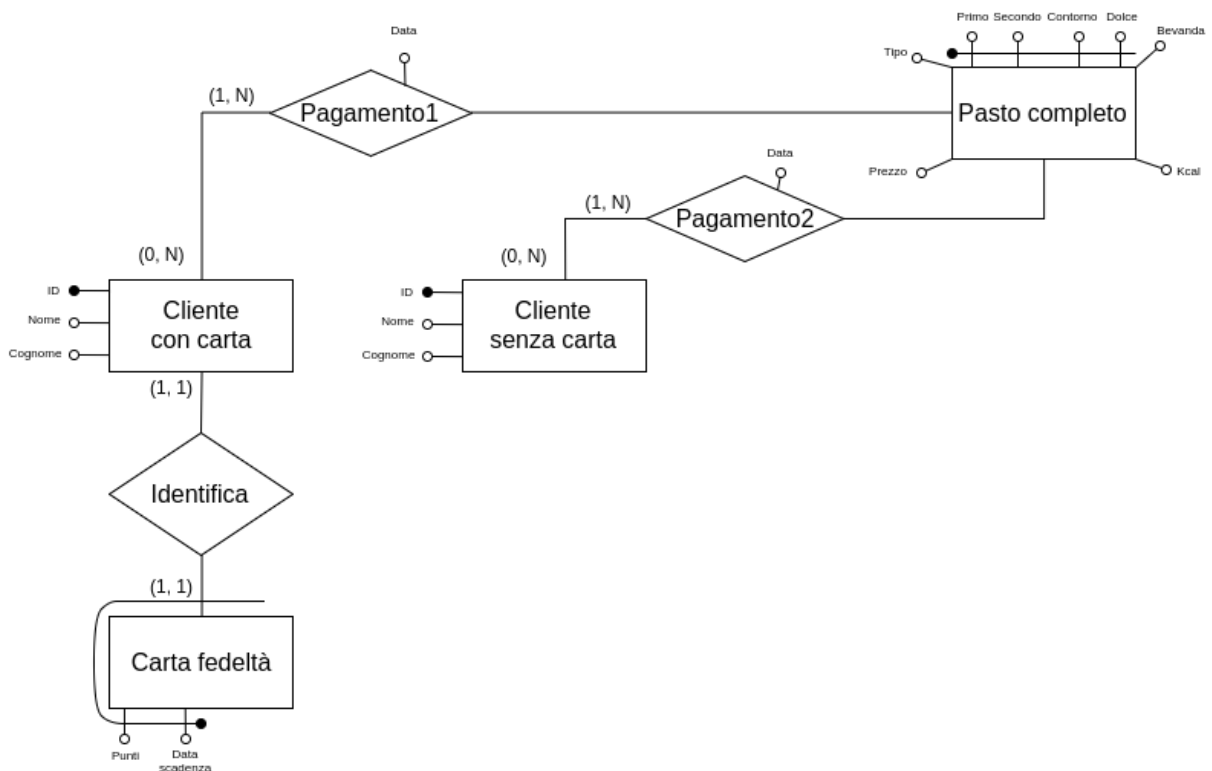
La generalizzazione è stata eliminata aggiungendo l'attributo "tipo" all'entità pasto completo.



La generalizzazione dell'entità cliente può essere eliminata utilizzando il metodo di accorpamento del genitore nelle figlie, in quanto si tratta di una generalizzazione totale (quindi un'occorrenza del padre è obbligatoriamente uno dei due figli).

Di conseguenza sono stati aggiunti gli attributi "id, nome, cognome" alle entità figlie "Con Carta, Senza Carta" ed è stata eliminata l'entità Cliente.

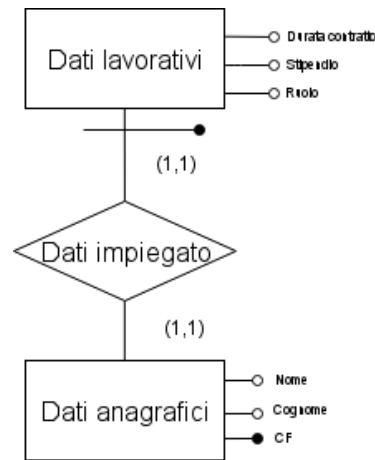
Dopodichè ad ognuna delle due entità è stata aggiunta la relazione Pagamento ("Pagamento1, Pagamento2") con l'entità Pasto Completo.



Accorpamenti e partizionamenti:

E' stato osservato che le operazioni che coinvolgono frequentemente l'entità Personale richiedono, per un'occorrenza di Personale, o solo informazioni di carattere anagrafico o solo informazioni relative alla sua situazione contrattuale.

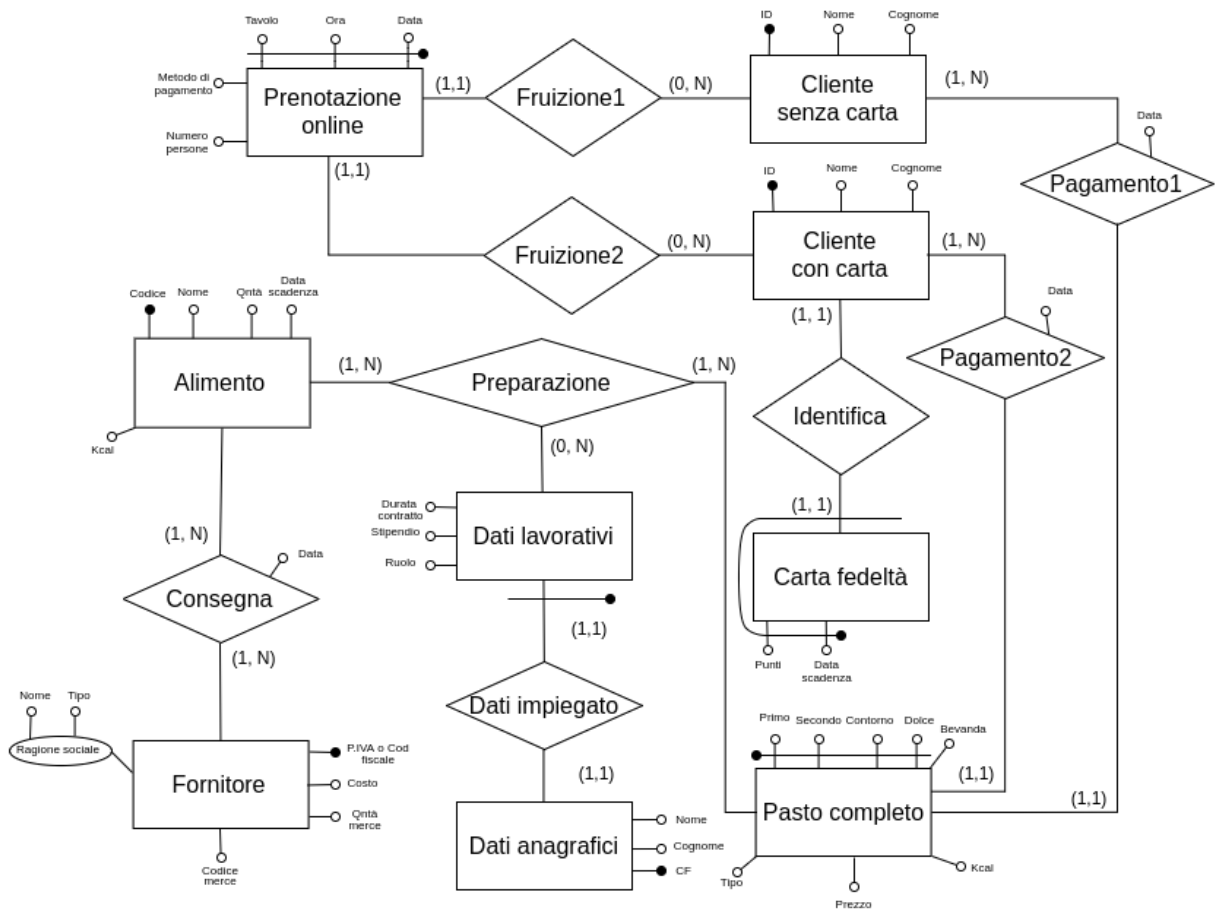
E' stata quindi applicata una decomposizione verticale dell'entità Personale, separando gli attributi ed associandoli a due entità distinte: Dati Anagrafici e Dati Lavorativi. Queste ultime due entità sono legate dalla relazione uno-ad-uno denominata Dati Impiegato.



Eliminazione degli attributi multivalore:

Non sono stati riscontrati attributi multivalore

Successivamente alle modifiche effettuate, lo schema ER ristrutturato si presenta come segue:



Elenco degli identificatori principali:

Nome entità	Identificatore
Prenotazione online	Data, Ora, Tavolo (numero, numero, numero)
Cliente con carta	ID (numero)
Cliente senza carta	ID (numero)
Carta fedeltà	Partita IVA o codice fiscale (stringa)
Pasto completo	Primo (stringa), secondo (stringa), contorno (stringa), dolce (stringa)
Dati lavorativi	Dati anagrafici (stringa)
Dati anagrafici	Codice fiscale (stringa)
Fornitore	Partita IVA o codice fiscale (stringa)
Alimento	Codice (numero)

3c. Traduzione verso il modello relazionale

Entità – Relazione	Traduzione
Prenotazione online	PrenotazioneOnline(<u>CodicePrenotazione</u> , Tavolo, Ora, Data, MetodoPagamento, NumPersone)
Cliente con carta	ClienteConCarta(<u>ID</u> , Nome, Cognome)
Cliente senza carta	ClienteSenzaCarta(<u>ID</u> , Nome, Cognome)
Carta fedeltà	CartaFedelta(<u>ClienteConCarta</u> , Punti, DataScadenza)
Pasto completo	PastoCompleto(<u>CodicePasto</u> , Primo, Secondo, Contorno, Dolce, Bevanda, Kcal, Prezzo)
Dati lavorativi	DatiLavorativi(<u>DatiAnagrafici</u> , DurataContratto, Stipendio, Ruolo)
Dati anagrafici	DatiAnagrafici(<u>CodFiscale</u> , Nome, Cognome)
Fornitore	Fornitore(<u>CodPar</u> , Costo, QntàMerce, RagioneSociale, CodiceMerce)
Alimento	Alimento(<u>Codice</u> , Kcal, Nome, Qntà, Data scadenza)
Consegna	Consegna(<u>Alimento</u> , <u>Fornitore</u> , Data)
Preparazione	Preparazione(<u>Alimento</u> , <u>PastoCompleto</u> , <u>DatiLavorativi</u>)
Fruizione1	Fruizione1(<u>PrenotazioneOnline</u> , <u>ClienteSenzaCarta</u>)
Fruizione2	Fruizione2(<u>PrenotazioneOnline</u> , <u>ClienteConCarta</u>)
Pagamento1	Pagamento1(<u>ClienteSenzaCarta</u> , <u>PastoCompleto</u> , Data)
Pagamento2	Pagamento2(<u>ClienteConCarta</u> , <u>PastoCompleto</u> , Data)

Le relazioni Identifica e Dati Impiegato non appaiono nella tabella perchè sono già state tradotte come conseguenza dell'identificazione esterna di Carta Fedeltà e Dati Lavorativi rispettivamente. Per evitare di avere chiavi composte da molti attributi sono state inoltre aggiunte due chiavi identificative così da ottenere indici di dimensioni ridotte: CodicePrenotazione nella relazione PrenotazioneOnline e CodicePasto nella relazione PastoCompleto.

Traduzione	Vincoli di riferimento
PrenotazioneOnline(<u>CodicePrenotazione</u> , Tavolo, Ora, Data, MetodoPagamento, NumPersone)	-
ClienteConCarta(<u>ID</u> , Nome, Cognome)	-
ClienteSenzaCarta(<u>ID</u> , Nome, Cognome)	-
CartaFedelta(<u>ClienteConCarta</u> , Punti, DataScadenza)	ClienteConCarta → ClienteConCarta.ID
PastoCompleto(<u>CodicePasto</u> , Primo, Secondo, Contorno, Dolce, Bevanda, Kcal, Prezzo)	-
DatiLavorativi(<u>DatiAnagrafici</u> , DurataContratto, Stipendio, Ruolo)	DatiAnagrafici → DatiAnagrafici.CodFiscale
DatiAnagrafici(<u>CodFiscale</u> , Nome, Cognome)	-
Fornitore(<u>CodPar</u> , Costo, QntàMerce, NomeRagioneSociale, TipoRagioneSociale, CodiceMerce)	-
Alimento(<u>Codice</u> , Kcal, Nome, Qntà, DataScadenza)	-
Consegna(<u>Alimento</u> , <u>Fornitore</u> , Data)	Fornitore → Fornitore.CodPar Alimento → Alimento.Codice
Preparazione(<u>Alimento</u> , <u>PastoCompleto</u> , <u>DatiLavorativi</u>)	Alimento → Alimento.Codice PastoCompleto → PastoCompleto.CodicePasto DatiLavorativi → DatiAnagrafici.CodFiscale
Fruizione1(<u>PrenotazioneOnline</u> , <u>ClienteSenzaCarta</u>)	ClienteSenzaCarta → ClienteSenzaCarta.ID PrenotazioneOnline → PrenotazioneOnline.CodicePrenotazione
Fruizione2(<u>PrenotazioneOnline</u> , <u>ClienteConCarta</u>)	ClienteConCarta → ClienteConCarta.ID PrenotazioneOnline → PrenotazioneOnline.CodicePrenotazione
Pagamento1(<u>ClienteSenzaCarta</u> , <u>PastoCompleto</u> , Data)	ClienteSenzaCarta → ClienteSenzaCarta.ID PastoCompleto → PastoCompleto.CodicePasto
Pagamento2(<u>ClienteConCarta</u> , <u>PastoCompleto</u> , Data)	ClienteConCarta → ClienteConCarta.ID PastoCompleto → PastoCompleto.CodicePasto

3d. Normalizzazione

Relazioni:

Analizzando lo schema relazionale si nota che Alimento contiene una dipendenza funzionale non banale: Nome -> Kcal, ovvero ogni nome determina un valore di kcalorie.

La relazione è stata normalizzata in forma normale di Boyce e Codd scomponendola come segue:

DatiNutrizionaliAlimento(Nome, Kcal)

DatiMagazzinoAlimento(Codice, Qntà, Nome, DataScadenza)

Le due nuove relazioni non contengono dipendenze funzionali. Inoltre la relazione è stata decomposta senza perdita in quanto rispetta la seguente condizione: l'insieme degli attributi comuni alle due relazioni è chiave (Nome) per almeno una delle relazioni decomposte (DatiNutrizionaliAlimento).

Viene inoltre garantita la conservazione delle dipendenze perchè ognuna delle dipendenze funzionali (Nome -> Kcal) dello schema originario coinvolge attributi che appaiono tutti insieme in uno degli schemi decomposti (DatiNutrizionaliAlimento)

La sezione di schema logico modificato appare come segue:

Traduzione	Vincoli di riferimento
DatiNutrizionaliAlimento(<u>Nome</u> , Kcal)	-
DatiMagazzinoAlimento(<u>Codice</u> , Qntà, Nome, DataScadenza)	Nome → DatiNutrizionaliAlimento.Nome
Consegna(<u>DatiMagazzinoAlimento</u> , <u>Fornitore</u> , Data)	Fornitore → Fornitore.CodPar DatiMagazzinoAlimento → DatiMagazzinoAlimento.Codice
Preparazione(<u>DatiMagazzinoAlimento</u> , <u>PastoCompleto</u> , <u>DatiLavorativi</u>)	DatiMagazzinoAlimento → DatiMagazzinoAlimento.Codice PastoCompleto → PastoCompleto.CodicePasto, DatiLavorativi → DatiAnagrafici.CodFiscale

4. Codifica SQL

4a. Definizione dello schema

```
CREATE TABLE DatiNutrizionaliAlimento(  
    Nome char(20) NOT NULL PRIMARY KEY,  
    Kcal INT NOT NULL  
);
```

```
CREATE TABLE DatiMagazzinoAlimento(  
    Codice INT NOT NULL PRIMARY KEY,  
    Quantita INT NOT NULL,  
    Nome CHAR(20) REFERENCES DatiNutrizionaliAlimento(Nome),  
    DataScadenza DATE NOT NULL  
);
```

```
CREATE TABLE Consegna(  
    DatiMagazzinoAlimento INT NOT NULL  
        REFERENCES DatiMagazzinoAlimento(Codice),  
    Fornitore CHAR(16) NOT NULL  
        REFERENCES Fornitore(CodPar),  
    Data DATE NOT NULL,
```



```

        PRIMARY KEY (DatiMagazzinoAlimento, Fornitore)
    );
CREATE TABLE Preparazione(
    DatiMagazzinoAlimento INT NOT NULL
        REFERENCES DatiMagazzinoAlimento(Codice),
    PastoCompleto INT NOT NULL
        REFERENCES PastoCompleto(CodicePasto),
    DatiLavorativi CHAR(16)
        REFERENCES DatiAnagrafici(CodFiscale),
    PRIMARY KEY(DatiMagazzinoAlimento,PastoCompleto, DatiLavorativi)
);

```

```

CREATE TABLE PrenotazioneOnline(
    CodicePrenotazione INT NOT NULL PRIMARY KEY,
    Tavolo INT NOT NULL,
    Ora TIME NOT NULL,
    Data DATE NOT NULL,
    MetodoPagamento CHAR(20) NOT NULL,
    NumPersone INT NOT NULL
);

```

```

CREATE TABLE ClienteConCarta(
    ID INT NOT NULL PRIMARY KEY,
    Nome CHAR(20) NOT NULL,
    Cognome CHAR(20) NOT NULL
);

```

```

CREATE TABLE ClienteSenzaCarta(
    ID INT NOT NULL PRIMARY KEY,
    Nome CHAR(20) NOT NULL,
    Cognome CHAR(20) NOT NULL
);

```

```

CREATE TABLE CartaFedelta(
    ClienteConCarta INT REFERENCES ClienteConCarta(ID),
    Punti SMALLINT DEFAULT 0,
    DataScadenza DATE NOT NULL
);

```

```

CREATE TABLE PastoCompleto(
    CodicePasto INT NOT NULL PRIMARY KEY,
    Primo CHAR(20) NOT NULL,
    Secondo CHAR(20) NOT NULL,
    Contorno CHAR(20) NOT NULL,
    Dolce CHAR(20) NOT NULL,
    Bevanda CHAR(20) NOT NULL,

```

```

        Kcal INT NOT NULL,
        Prezzo DECIMAL(5,2) NOT NULL);
CREATE TABLE DatiLavorativi(
    DatiAnagrafici CHAR(16) NOT NULL
        REFERENCES DatiAnagrafici(CodFiscale),
    ScadenzaContratto DATE NOT NULL,
    Stipendio INT NOT NULL,
    Ruolo CHAR(20) NOT NULL,
    PRIMARY KEY (DatiAnagrafici)
);

```

```

CREATE TABLE DatiAnagrafici(
    CodFiscale CHAR(16) NOT NULL PRIMARY KEY,
    Nome CHAR(20) NOT NULL,
    Cognome CHAR(20) NOT NULL
);

```

```

CREATE TABLE Fornitore(
    CodPar CHAR(16) NOT NULL PRIMARY KEY,
    Costo DECIMAL(5,2) NOT NULL,
    QuantitaMerce INT NOT NULL,
    NomeRagioneSociale CHAR(20) NOT NULL,
    TipoRagioneSociale CHAR(10) NOT NULL,
    CodiceMerce INT NOT NULL
);

```

```

CREATE TABLE Fruizione1(
    PrenotazioneOnline INT NOT NULL
        REFERENCES PrenotazioneOnline(CodicePrenotazione),
    ClienteSenzaCarta INT NOT NULL
        REFERENCES ClienteSenzaCarta(ID),
    PRIMARY KEY (PrenotazioneOnline, ClienteSenzaCarta)
);

```

```

CREATE TABLE Fruizione2(
    PrenotazioneOnline INT NOT NULL
        REFERENCES PrenotazioneOnline(CodicePrenotazione),
    ClienteConCarta INT NOT NULL
        REFERENCES ClienteConCarta(ID),
    PRIMARY KEY (PrenotazioneOnline, ClienteConCarta)
);

```

```

CREATE TABLE Pagamento1(
    ClienteSenzaCarta INT NOT NULL REFERENCES
        ClienteSenzaCarta(ID),
    PastoCompleto INT NOT NULL REFERENCES
        PastoCompleto(CodicePasto),
    Data DATETIME NOT NULL,
    PRIMARY KEY (ClienteSenzaCarta, Data)
);

```

```

CREATE TABLE Pagamento2(
    ClienteConCarta INT NOT NULL REFERENCES ClienteConCarta(ID),
    PastoCompleto INT NOT NULL REFERENCES
        PastoCompleto(CodicePasto),
    Data DATETIME NOT NULL,
    PRIMARY KEY (ClienteConCarta, Data)
);

```

4c. Codifica delle operazioni

1. Inserire un nuovo fornitore (1 volta ogni tre anni)

```

INSERT INTO Fornitore(CodPar, Costo, QuantitaMerce,
    NomeRagioneSociale, TipoRagioneSociale, CodiceMerce)
VALUES (...)

```

2. Inserire un nuovo dipendente (1 volta ogni quattro anni)

```

INSERT INTO DatiAnagrafici(CodFiscale, Nome, Cognome)
VALUES (...)

```

```

INSERT INTO DatiLavorativi(DatiAnagrafici, ScadenzaContratto,
    Stipendio, Ruolo)
VALUES (...)

```

3. Inserire un nuovo alimento (in media 5 volte all'anno)

```

INSERT INTO DatiMagazzinoAlimento(Codice, Quantita, Nome,
    DataScadenza)
VALUES (...)

```

```

INSERT INTO DatiNutrizionaliAlimento(Nome, Kcal)
VALUES (...)

```

4. Inserire un nuovo utente registrato (in media 20 volte al mese)

Se ha la carta fedelta'

```
INSERT INTO ClienteConCarta(ID, Nome, Cognome)
VALUES (...)
```

Se non ha la carta fedelta'

```
INSERT INTO ClienteSenzaCarta(ID, Nome, Cognome)
VALUES (...)
```

5. Assegnare una prenotazione ad un cliente (in media 100 volte al giorno)

```
INSERT INTO PrenotazioneOnline(CodicePrenotazione,
    Tavolo, Ora, Data, MetodoPagamento, NumPersone)
VALUES([value-1], ...)
```

Se il cliente non ha la carta fedelta'

```
INSERT INTO Fruizione1(PrenotazioneOnline,
    ClienteSenzaCarta) VALUES (value-1], ...)
```

Se il cliente ha la carta fedelta'

```
INSERT INTO Fruizione2(PrenotazioneOnline,
    ClienteConCarta) VALUES ([value-1], ...)
```

6. Calcolare l'apporto calorico totale dei menù giornalieri (1 volta al giorno)

```
SELECT [value-1]
FROM PastoCompleto
WHERE CodicePasto = [value-2]
```

```
UPDATE PastoCompleto
SET Kcal = Kcal -
(SELECT Kcal
FROM DatiNutrizionaliAlimento
WHERE [value-3] = DatiNutrizionaliAlimento.Nome)
WHERE PastoCompleto.CodicePasto = [value-2]
```

```
UPDATE PastoCompleto
SET [value-1] = [value-4]
WHERE PastoCompleto.CodicePasto = [value-2]
```

```

UPDATE PastoCompleto
SET Kcal = Kcal +
(SELECT Kcal
 FROM DatiNutrizionaliAlimento
 WHERE [value-4] = DatiNutrizionaliAlimento.Nome)
WHERE PastoCompleto.CodicePasto = [value-2]

```

7. Calcolare il totale delle entrate mensili (in media 1 volta al mese)

```

SELECT SUM(Prezzo) AS EntrateMese
FROM
(SELECT *
FROM Pagamento1 as P1
WHERE MONTH(P1.Data)=MONTH(CURDATE()))
UNION
SELECT *
FROM Pagamento2 as P2
WHERE MONTH(P2.Data)=MONTH(CURDATE()))
) AS EntrateCorrenti
JOIN PastoCompleto ON CodicePasto = PastoCompleto

```

8. Far visualizzare ai dipendenti quali alimenti saranno consegnati un determinato giorno (in media 16 volte al giorno)

```

SELECT *
FROM Consegna
JOIN DatiMagazzinoAlimento
ON Consegna.DatiMagazzinoAlimento =
DatiMagazzinoAlimento.Codice
WHERE Data = [value-1]

```

9. Visualizzare uno dei tre menù (in media 90 volte al giorno)

```

SELECT *
FROM PastoCompleto
WHERE CodicePasto = [value-1]

```

10. Visualizzare la quantità di un alimento presente in magazzino (in media 5 volte al mese)

```

SELECT Quantita
FROM DatiMagazzinoAlimento
WHERE Nome = [value-1]

```

11. Visualizzare gli alimenti in scadenza entro 3 giorni dal giorno corrente (in media 2 volte a settimana).

```
SELECT Nome
FROM DatiMagazzinoAlimento
WHERE DataScadenza <= DATE_ADD(CURDATE(), INTERVAL 3 DAY) AND
      DataScadenza >= CURDATE()
```

12. Effettuare una statistica su quanti punti hanno i clienti mediamente (1 volta al mese)

```
SELECT AVG(Punti) FROM CartaFedelta
```

13. Visualizzare i membri del personale con contratto in scadenza in meno di un mese (in media 1 volta al mese)

```
SELECT DL.ScadenzaContratto, DL.Ruolo, DA.Nome, DA.Cognome
FROM DatiLavorativi AS DL, DatiAnagrafici AS DA
WHERE DL.ScadenzaContratto <= DATE_ADD(CURDATE(), INTERVAL 30 DAY)
      AND DA.CodFiscale = DL.DatiAnagrafici;
```

5. Bibliografia

Atzeni P, Ceri S, Fraternali P, Paraboschi S, Torlone R. *Basi Di Dati: modelli e linguaggi di interrogazione*, Milano, McGraw-Hill, 2013, quarta edizione.