



## PROGETTO BASI DI DATI

(2017-2018)

### Abstract

Il ristorante le Tegnùe è un ristorante in spiaggia a Sottomarina situato a pochi metri dalla foce del Brenta. Un luogo tranquillo, lontano dai posti affollati e rumorosi del centro città, immerso nel verde delle palme e fiancheggiato da una splendida diga.

Una delle caratteristiche principali del ristorante le Tegnùe è sicuramente la genuinità dei prodotti: ogni giorno vengono raccolte le verdure direttamente dagli orti di proprietà della famiglia dei titolari situati nelle vicinanze.

La cucina del ristorante è incentrata principalmente sui piatti di pesce con un'ampia scelta di specialità, ma sono presenti anche molti altri piatti, dalla carne ai menù per i più piccoli per garantire un'esperienza unica in riva al mare.

Il ristorante viene acquistato dagli attuali titolari nel 2010 e da allora è in costante crescita e da pochi anni collabora con l'adiacente campeggio Oasi.

Ciò sta contribuendo ad un grande aumento di popolarità del ristorante non solo tra la gente del posto ma soprattutto tra i turisti e la gente da fuori.

Viene quindi presa la decisione da parte dei titolari di effettuare un ampliamento strutturale, un aumento dei dipendenti e di introdurre dei sistemi informatici per il supporto alla gestione.

Viene quindi redatto un database per rendere più semplice ed efficiente la gestione del locale, specialmente per quanto riguarda l'organizzazione del personale, le prenotazioni dei tavoli, il sistema di ordinazione ai tavoli, i rapporti con i fornitori ed un metodo di archiviazione e controllo dei pagamenti.

### Analisi dei requisiti

#### Introduzione

Si vuole realizzare una base di dati per l'organizzazione del ristorante le Tegnùe.

Nel ristorante lavorano diversi dipendenti ognuno dei quali è identificato da un codice operatore e dispone di un salario, si vogliono inoltre memorizzare: codice dipendente, nome, cognome, sesso, indirizzo, data di nascita, data di assunzione, numero di telefono ed il codice IBAN su cui accreditare lo stipendio.

Ogni dipendente può lavorare o nella cucina o nella sala, ed ha inoltre uno specifico ruolo:

Per quanto riguarda la cucina:

- Lavapiatti: si occupa di lavare tutto il necessario per il servizio e si occupa inoltre della pulizia della sala della cucina e dei servizi.
- Cuoco: ha il compito di preparare i piatti per i clienti, e di preparare tutto il necessario per la linea della cucina.
- Chef: dirige il personale della cucina durante i servizi e si occupa di ordinare tutti i prodotti necessari per la cucina.

Per la sala:

- Cameriere: si occupa del servizio ai tavoli e di prendere le ordinazioni ai tavoli, sistema tutto il necessario per il servizio.

Ballarin Federico - 1123718

Bullo Valentina - 1123714

- Banconiere: si occupa di preparare tutte le bibite e di gestire la cassa.
- Caposala: ha un ruolo di direzione del personale di sala ed il compito di ordinare tutti i vari prodotti necessari per il ristorante.

Ad ogni dipendente viene assegnato un turno, ogni turno con una data di inizio, fine e un'ora di inizio, fine. Il ristorante ha molti tavoli, alcuni nella sala interna ed altri nella sala esterna, ogni tavolo è identificato dal rispettivo numero ed ha inoltre il numero massimo di posti e una piccola descrizione della posizione del tavolo all'interno della sala.

I tavoli possono essere riservati per i clienti che hanno effettuato una prenotazione, per ogni prenotazione vogliamo memorizzare: la data e l'ora di arrivo dei clienti, il numero di persone, il numero del tavolo da riservare, il cliente che ha effettuato la prenotazione ed eventuali note.

Per ogni cliente viene invece registrato il nome, il cognome, il numero di telefono in modo da poter essere contattato ed è riconosciuto univocamente da un codice cliente.

Le Ordinazioni vengono prese al tavolo dai camerieri, ogni ordinazione ha un codice che la identifica e viene registrata la data, l'ora, il cameriere che effettua l'ordinazione e il numero del tavolo che effettua l'ordinazione.

Nel dettaglio ogni ordinazione è composta dal codice dell'ordinazione di appartenenza, l'elemento del menù che viene richiesto andando inoltre a memorizzarne la quantità, il prezzo totale e se necessaria una piccola nota. Ogni dettaglio è identificato univocamente da un codice.

Viene salvato nel database anche il menù che comprende tutto ciò che può essere ordinato dai tavoli, ogni elemento del menù ha un rispettivo codice e nome, una descrizione, un prezzo unitario e un codice gruppo in quanto i vari articoli sono suddivisi in gruppi per agevolare la ricerca dalla cassa e dai palmari. Ogni gruppo ha un suo codice e un nome.

Si vogliono inoltre memorizzare i vari pagamenti che vengono effettuati, per ogni pagamento vogliamo salvare la data e l'ora, il tipo di pagamento, il totale pagato e se applicato, lo sconto. Ogni pagamento viene identificato da un codice pagamento e viene anche salvato il codice del banconiere che ha incassato il pagamento per motivi di sicurezza, in quanto ogni banconiere è pienamente responsabile della cassa ed in caso di anomalie negli incassi avere il nome del dipendente che ha gestito il pagamento agevola molto la ricerca del responsabile.

Alcuni clienti provengono dal campeggio ed hanno la possibilità di effettuare un pagamento elettronico attraverso un braccialetto con il 10% di sconto.

Il caposala e lo chef hanno il compito di ordinare i vari prodotti necessari per il funzionamento del ristorante, viene quindi creato e salvato nella base di dati un catalogo per permettere di ordinare i prodotti più adeguati e convenienti che è composto dalla lista di tutti i prodotti ordinabili sia per la sala che per la cucina, salvando per ogni prodotto il rispettivo codice, il nome, una descrizione, il prezzo e il codice del fornitore che lo fornisce.

Di ogni fornitore si vuole invece ricordare la ragione sociale della ditta, la partita iva, l'indirizzo e il numero di telefono.

## Glossario dei termini

Termine	Descrizione	Collegamento
Dipendente	Lavoratore del ristorante. Può lavorare in sala o in cucina. In sala: cameriere, banconiere, caposala. In cucina: lavapiatti, cuoco, chef.	Turno, prodotto, ordinazione.
Turno	Turno di lavoro di ogni dipendente.	Dipendente.
Tavolo	Tavolo dove i clienti si siedono per mangiare nel ristorante.	Prenotazione, ordinazione.
Cliente	Persona che si siede al tavolo del ristorante ed ordina o che decide di prenotare un tavolo.	Prenotazione.
Prenotazione	Permette ad un cliente di riservare un tavolo per un'ora e una data futura precisa.	Cliente, tavolo.
Elemento Menù	Piatto e bibita che un cliente può ordinare al tavolo.	Ordinazione, gruppo.

Gruppo	Insieme o categoria di elementi del menu con caratteristiche comuni.	Menù.
Ordinazione	Lista di elementi del menù registrati dai caposala che una specifica tavola richiede.	Dipendente(Cameriere), tavolo, elemento-menù, pagamento.
Dettaglio ordinazione	Lista dei piatti richiesti in un particolare ordine.	Ordinazione, elemento-menù.
Pagamento	Pagamento che permette di estinguere i vari ordini che sono stati richiesti.	Ordinazione,Dipendente(Banconiere).
Prodotto	Prodotto che viene ordinato dai responsabili per garantire i vari servizi che il ristorante offre. Ci sono prodotti per la sala e prodotti per la cucina.	Fornitore, Dipendente (Caposala, Chef).
Fornitore	Ditta che vende e consegna i prodotti ordinati al ristorante.	Prodotto.

## Operazioni previste

OPERAZIONE	TIPO	FREQUENZA
1.Assunzione di un nuovo dipendente	B	10 volte all'anno
2.Assegnazione dei turni ai dipendenti	I	1 volta al giorno
3.Registrazione ordine effettuato ad un tavolo	B	400 volte al giorno
4.Inserimento di un dettaglio di uno specifico ordine	I	2500 volte al giorno
5.Incasso da parte di un banconiere di un pagamento	I	400 volte al giorno
6.Aggiunta di un elemento nel menù	I	3 volte al mese
7.Modifica al prezzo di un elemento del menù	I	10 volte all'anno
8.Inserimento di una nuova prenotazione	I	50 volte al giorno
9.Ordine di un prodotto da parte del caposala/chef	I	20 volte a settimana
10.Variazione Prezzo di un prodotto	I	5 volte a settimana
11.Report degli incassi giornalieri	B	1 volta al giorno

## Strutturazione di requisiti

Frasi relative a "DIPENDENTE"
<p>Nel ristorante lavorano diversi dipendenti ognuno dei quali è identificato da un codice operatore e dispone di un salario, si vogliono inoltre memorizzare: codice dipendente, nome, cognome, sesso, indirizzo, data di nascita, data di assunzione, numero di telefono ed il codice IBAN su cui accreditare lo stipendio.</p> <p>Ogni dipendente può lavorare o nella cucina o nella sala, ed ha inoltre uno specifico ruolo:</p> <p>Per quanto riguarda la cucina:</p> <ul style="list-style-type: none"> <li>• Lavapiatti: si occupa di lavare tutto il necessario per il servizio e si occupa inoltre della pulizia della sala della cucina e dei servizi.</li> <li>• Cuoco: ha il compito di preparare i piatti per i clienti, e di preparare tutto il necessario per la linea della cucina.</li> <li>• Chef: dirige il personale della cucina durante i servizi e si occupa di ordinare tutti i prodotti necessari per la cucina.</li> </ul> <p>Per la sala:</p> <ul style="list-style-type: none"> <li>• Cameriere: si occupa del servizio ai tavoli e di prendere le ordinazioni ai tavoli, sistema tutto il necessario per il servizio.</li> <li>• Banconiere: si occupa di preparare tutte le bibite e di gestire la cassa.</li> <li>• Caposala: ha un ruolo di direzione del personale di sala ed il compito di ordinare tutti i vari prodotti necessari per il ristorante.</li> </ul>

Frasi relative a "TURNO"
Ad ogni dipendente viene assegnato un turno, ogni turno con una data di inizio, fine e un'ora di inizio, fine.

#### Frase relative a "TAVOLO"

Il ristorante ha molti tavoli, alcuni nella sala interna ed altri nella sala esterna, ogni tavolo è identificato dal rispettivo numero ed ha inoltre il numero massimo di posti e una piccola descrizione della posizione del tavolo all'interno della sala.

#### Frase relative a "CLIENTE"

Per ogni cliente viene invece registrato il nome, il cognome, il numero di telefono in modo da poter essere contattato ed è riconosciuto univocamente da un codice cliente. Alcuni clienti provengono dal campeggio.

#### Frase relative a "PRENOTAZIONE"

I tavoli possono essere riservati per i clienti che hanno effettuato una prenotazione, per ogni prenotazione vogliamo memorizzare: la data e l'ora di arrivo dei clienti, il numero di persone, il numero del tavolo da riservare, il cliente che ha effettuato la prenotazione ed eventuali note.

#### Frase relative a "ELEMENTO MENU"

Viene salvato nel database anche il menù che comprende tutto ciò che può essere ordinato dai tavoli, ogni elemento del menù ha un rispettivo codice e nome, una descrizione, un prezzo unitario e un codice gruppo.

#### Frase relative a "GRUPPO"

I vari articoli sono suddivisi in gruppi per agevolare la ricerca dalla cassa e dai palmari. Ogni gruppo ha un suo codice e un nome.

#### Frase relative a "ORDINAZIONE"

Le Ordinanze vengono prese al tavolo dai camerieri, ogni ordinazione ha un codice che la identifica e viene registrata la data, l'ora, il cameriere che effettua l'ordinazione e il numero del tavolo che effettua l'ordinazione.

#### Frase relative a "DETTAGLIO ORDINAZIONE"

Nel dettaglio ogni ordinazione è composta dal codice dell'ordinazione di appartenenza, l'elemento del menù che viene richiesto andando inoltre a memorizzarne la quantità, il prezzo totale e se necessaria una piccola nota. Ogni dettaglio è identificato univocamente da un codice.

#### Frase relative a "PAGAMENTO"

Si vogliono inoltre memorizzare i vari pagamenti che vengono effettuati, per ogni pagamento vogliamo salvare la data e l'ora, il tipo di pagamento, il totale pagato e se applicato, lo sconto. Ogni pagamento viene identificato da un codice pagamento e viene anche salvato il codice del banconiere che ha incassato il pagamento per motivi di sicurezza, in quanto ogni banconiere è pienamente responsabile della cassa ed in caso di anomalie negli incassi avere il nome del dipendente che ha gestito il pagamento agevola molto la ricerca del responsabile. Alcuni clienti provengono dal campeggio ed hanno la possibilità di effettuare un pagamento elettronico attraverso un braccialetto con il 10% di sconto.

Ballarin Federico - 1123718

Bullo Valentina - 1123714

#### Frasi relative a "PRODOTTO"

Il caposala e lo chef hanno il compito di ordinare i vari prodotti necessari per il funzionamento del ristorante, viene quindi creato e salvato nella base di dati un catalogo per permettere di ordinare i prodotti più adeguati e convenienti che è composto dalla lista di tutti i prodotti ordinabili sia per la sala che per la cucina, salvando per ogni prodotto il rispettivo codice, il nome, una descrizione, il prezzo e il codice del fornitore che lo fornisce.

#### Frasi relative a "FORNITORE"

Di ogni fornitore si vuole invece ricordare la ragione sociale della ditta, la partita iva, l'indirizzo e il numero di telefono.

## Progettazione concettuale

### Analisi delle entità

-Le entità senza attributi derivanti da generalizzazioni non sono rappresentate.

#### Dipendente

CodiceDipendente	VARCHAR	Identifica univocamente i dipendenti del ristorante	Chiave
Nome	VARCHAR	Nome del dipendente	
Cognome	VARCHAR	Cognome del dipendente	
Sesso	VARCHAR	Sesso del dipendente	
Indirizzo	ATTRIBUTO COMPOSTO	Formato da: Via, Città, Provincia, Regione. Di tipo VARCHAR.	
DataNascita	DATE	Data di nascita del dipendente	
DataAssunzione	DATE	Data di inizio del lavoro nel ristorante del dipendente	
Stipendio	DECIMAL	Retribuzione corrisposta al dipendente	
IBAN	VARCHAR	Codice IBAN del dipendente dove accreditare lo stipendio	
NumeroTelefono	INTEGER	Numero di telefono per contattare il dipendente	

#### Turno

Dipendente	VARCHAR	Identifica il dipendente	Chiave
DataInizio	DATE	Data di inizio del turno	
DataFine	DATE	Data di fine del turno	
OraInizio	TIME	Ora di inizio del turno	
OraFine	TIME	Ora di fine del turno	

#### Tavolo

NumeroTavolo	TINYINT	Identifica univocamente i tavoli del ristorante	Chiave
NumeroPosti	TINYINT	Numero di posti massimo del tavolo	
Descrizione	VARCHAR	Descrizione della posizione del tavolo all'interno della sala	

#### Cliente

CodiceCliente	VARCHAR	Identifica univocamente il Cliente	Chiave
Nome	VARCHAR	Nome del cliente	
Cognome	VARCHAR	Cognome del cliente	
NumeroTelefono	INTEGER	Numero di telefono del cliente	

#### Prenotazione

CodicePrenotazione	VARCHAR	Identifica univocamente la prenotazione	Chiave
Data	DATE	Data per il quale viene riservato il tavolo	
Ora	TIME	Ora per la quale viene riservato il tavolo	
NumeroPersone	SMALLINT	Numero di persone per il quale riservare il tavolo	
Note	VARCHAR	Note della prenotazione	

Ballarin Federico - 1123718

Bullo Valentina - 1123714

ElementoMenù			
CodiceElemento	VARCHAR	Identifica univocamente l'elemento del menù	Chiave
Nome	VARCHAR	Nome dell'elemento del menù	
Descrizione	VARCHAR	Breve descrizione dell'elemento del menù	
Prezzo	DECIMAL	Prezzo unitario dell'elemento del munù	

Gruppo			
CodiceGruppo	VARCHAR	Identifica univocamente ogni gruppo	Chiave
Nome	VARCHAR	Nome del gruppo	

Ordinazione			
CodiceOrdinazione	VARCHAR	Identifica univocamente ogni ordinazione	Chiave
Data	DATE	Data in cui viene effettuata l'ordinazione	
Ora	TIME	Ora in cui viene effettuata l'ordinazione	

DettaglioOrdinazione			
CodiceDettaglio	VARCHAR	Identifica univocamente ogni dettaglio di ogni ordinazione	Chiave
Elemento	VARCHAR	Identifica l'elemento che viene ordinato	
Quantità	INTEGER	Indica la quantità dell'elemento che viene richiesto	
Nota	VARCHAR	Nota riguardante l'elemento ordinato	
PrezzoTotale	DECIMAL	Prezzo totale della riga dell'ordine	

Pagamento			
CodicePagamento	VARCHAR	Identifica univocamente ogni pagamento	Chiave
TipoPagamento	VARCHAR	Metodo con il quale viene effettuato il pagamento	
ImportoTotale	DECIMAL	Importo pagato	
Sconto	DECIMAL	Sconto applicato	

Prodotto			
CodiceProdotto	VARCHAR	Identifica univocamente il prodotto	Chiave
Nome	VARCHAR	Nome del prodotto	
Prezzo	DECIMAL	Prezzo unitario del prodotto	
Descrizione	VARCHAR	Descrizione del prodotto	

Fornitore			
CodiceFornitore	VARCHAR	Identifica univocamente il fornitore	Chiave
RagioneSociale	VARCHAR	Nome dell'azienda fornitrice	
PartitaIVA	VARCHAR	Partita IVA dell'azienda fornitrice	
Indirizzo	ATTRIBUTO COMPOSTO	Formato da: Via, Città, Provincia, Regione. Di tipo VARCHAR.	
NumeroTelefono	INTEGER	Numero di telefono per contattare il fornitore	

## Generalizzazioni

- Tavolo è generalizzazione totale ed esclusiva delle entità: TavoloInterno e TavoloEsterno.
- Dipendente è generalizzazione totale ed esclusiva delle entità: DipendenteSala e DipendenteCucina.
- DipendenteSala è generalizzazione totale ed esclusiva delle entità: Lavapiatti, Cuoco e Chef.
- DipendenteCucina è generalizzazione totale ed esclusiva delle entità: Cameriere, Banconiere e Caposala.
- Cliente è generalizzazione parziale dell'entità: ClienteCamping.
- Prodotto è generalizzazione completa e sovrapposta delle entità: ProdottoSala e ProdottoCucina.

## Analisi delle relazioni e delle cardinalità

- Dipendente - Turno: **Assegnato**

Ballarin Federico - 1123718

Bullo Valentina - 1123714

- Ad un dipendente sono assegnati uno o più turni **(1,N)**
- Un turno è assegnato ad un solo dipendente **(1,1)**
- Tavolo - Prenotazione: Occupazione
  - Un tavolo può venire occupato da delle prenotazioni **(0,N)**
  - Una prenotazione va ad occupare una sola tavola **(1,1)\***

\*Nel caso sia necessario unire più tavoli per sistemare una prenotazione viene comunque scelto un unico numero per identificare la nuova tavola, in particolare il numero della tavola più alto che compone il tavolo.

- Prenotazione - Cliente: Richiesta
  - Una prenotazione è richiesta da un solo cliente **(1,1)**
  - Un cliente può richiedere una o più prenotazioni **(1,N)**
- Tavolo - Ordinazione: Appartenenza
  - Ad un tavolo possono appartenere più ordinazioni **(0,N)\***
  - Un'ordinazione appartiene ad un solo tavolo **(1,1)**

\* È possibile che a qualche tavolo appartengano zero ordinazioni, ad esempio in seguito ad un ampliamento della sala i nuovi tavoli non hanno nessuna ordinazione.

- Ordinazione - Cameriere: Effettuato
  - Un'ordinazione è effettuata da un solo cameriere **(1,1)**
  - Un cameriere può effettuare più ordinazioni **(1,N)**
- Ordinazione - DettaglioOrdinazione: Composizione
  - Un'ordinazione è composta da uno o più dettagli **(1,N)**
  - Un dettaglio va a comporre una sola ordinazione **(1,1)**
- Ordinazione - Pagamento: Corrispondenza
  - Ad un'ordinazione corrisponde un solo pagamento **(1,1)**
  - Un pagamento corrisponde ad una sola ordinazione **(1,1)**
- DettaglioOrdinazione - ElementoMenù: Costituito
  - Un dettaglio ordine è formato da un solo elemento del menù **(1,1)**
  - Un elemento del menù può andare a formare più dettagli ordine **(0,N)\***

\*È possibile che qualche elemento del menù non vada a formare nessun dettaglio ordine, ad esempio nel caso l'elemento sia appena stato inserito nel menù.

- ElementoMenù - Gruppo: Appartenenza
  - Un elemento del menù appartiene ad un solo gruppo **(1,1)**
  - Un gruppo è formato da uno o più elementi del menù **(1,N)**
- Banconiere - Pagamento: Incasso
  - Un banconiere può incassare uno o più pagamenti **(1,N)**
  - Un pagamento viene incassato da un solo banconiere **(1,1)**
- Caposala - ProdottoSala: Ordine\*\*
  - Un caposala può ordinare dei prodotti per la sala **(0,N)**
  - Un prodotto per la sala può essere ordinato da più capisala **(0,N)**
- Chef - ProdottoCucina: Ordine\*\*
  - Uno chef può ordinare dei prodotti per la cucina **(0,N)**
  - Un prodotto per la cucina può essere ordinato da più chef **(0,N)**

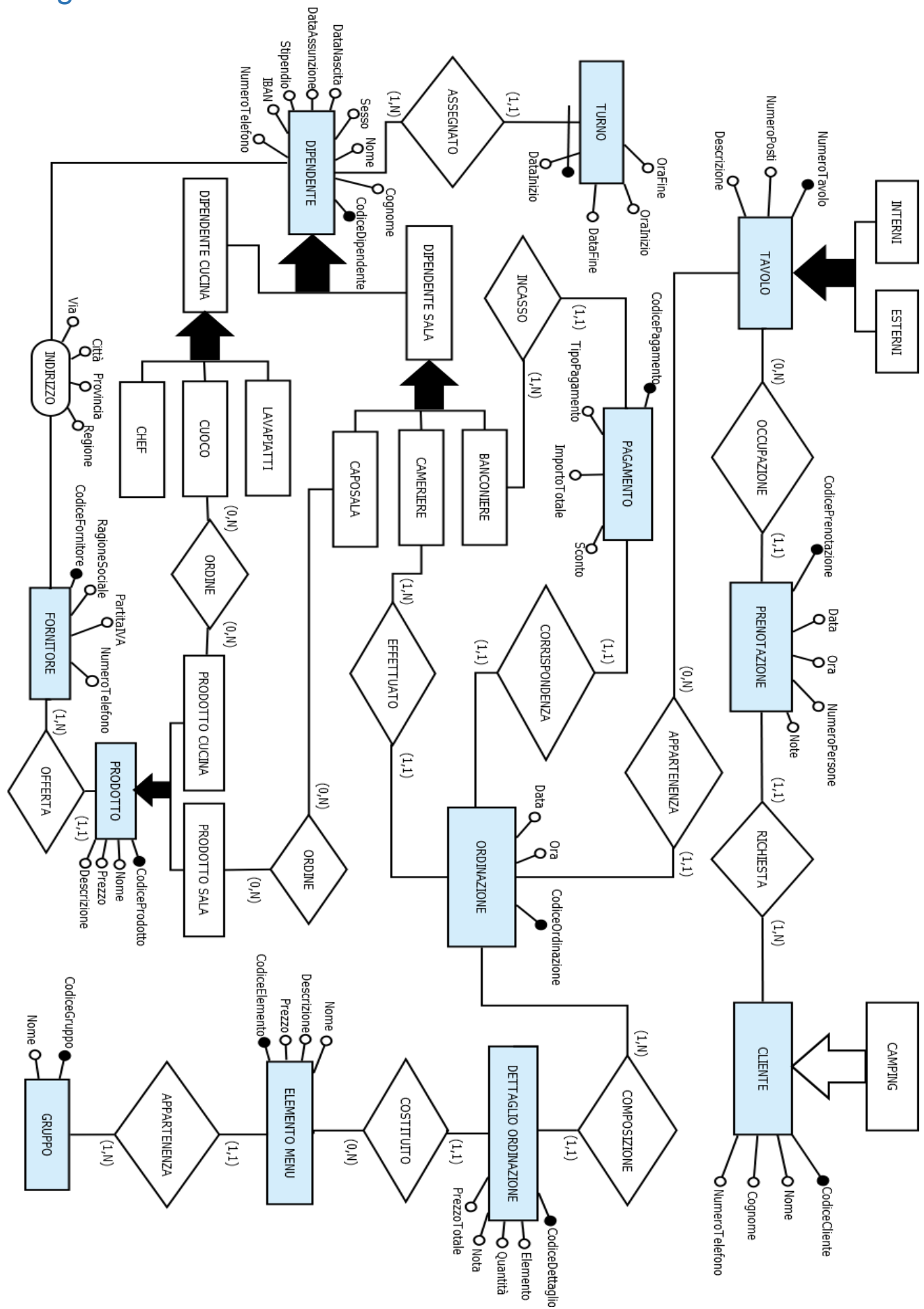
\*\*Nella relazione sono presenti gli attributi Quantità e Data in quanto ogni prodotto può essere ordinato molte volte in date diverse e con quantità variabili.

- Prodotto - Fornitore: Offerta
  - Un prodotto viene offerto da un solo fornitore **(1,1)\***
  - Un fornitore offre uno o più prodotti **(1,N)**

\*Fornitori diversi possono offrire lo stesso tipo di prodotto, ad esempio se lo stesso prodotto è offerto da due fornitori diversi nella tabella prodotto ci saranno due istanze per quel prodotto, una per il prodotto offerto da un fornitore e una per lo stesso prodotto offerto dall'altro, quindi ogni prodotto nella tabella prodotto è offerto da un solo fornitore.



## Diagramma E/R





# Progettazione logica

## Analisi delle ridondanze

Una ridondanza è presente nell'entità **Pagamento** nella quale il valore dell'attributo **ImportoTotale** è derivabile andando a visitare le tabelle: **Ordinazione**, **DettaglioOrdinazione**, e **ElementoMenù**, andando a verificare a che ordine corrisponde il pagamento e calcolando il totale in base a quali e quanti prodotti sono stati ordinati. Viene però deciso di mantenere l'attributo anche se ridondante in quanto considerato di fondamentale importanza ed utilità.

Una altra ridondanza è presente nell'entità **DettaglioOrdinazione**, nella quale il valore dell'attributo **PrezzoTotale** si può calcolare visitando la tabella **ElementoMenù** e moltiplicando il prezzo unitario dell'elemento per la quantità richiesta.

L'operazione coinvolta è la numero 4: Inserimento di un dettaglio di uno specifico ordine (2500 volte al giorno).

Si procede quindi con la valutazione del costo di quest'ultima nel caso di ridondanza e di assenza di essa.

**Tavola dei volumi**

Concetto	Tipo	Volume
DettaglioOrdine	E	50000
Costituito	R	50000
ElementoMenù	E	60

**CON RIDONDANZA**

Concetto	Costrutto	Accessi	Tipo
ElementoMenù	E	1	L
Costituito	R	1	S
DettaglioOrdine	E	1	L
DettaglioOrdine	E	1	S

**SENZA RIDONDANZA**

Concetto	Costrutto	Accessi	Tipo
DettaglioOrdine	E	1	S
Costituito	R	1	S

Con ridondanza: In presenza di ridondanza prima di inserire un nuovo dettaglio ordine bisogna calcolare il prezzo totale andando a moltiplicare il prezzo unitario dell'elemento per la quantità richiesta.

- Totale scritture:  $2 \cdot 2500 = 5000 \cdot 2 = 10000$  (le scritture sono contate due volte)
- Totale letture:  $2 \cdot 2500 = 5000$
- Totale accessi: 15000

Senza ridondanza: In assenza di ridondanza non c'è bisogno di nessun accesso in lettura in quanto basta solo inserire il dettaglio ordine formato dall'elemento del menù.

- Totale scritture:  $2 \cdot 2500 = 5000 \cdot 2 = 10000$  (le scritture sono contate due volte)
- Totale letture:  $0 \cdot 2500 = 0$
- Totale accessi: 10000

In conclusione, dopo aver esaminato il totale degli accessi per ogni caso, viene scelto di rimuovere l'attributo **PrezzoTotale** dall'entità **DettaglioOrdinazione** e di derivarlo dalla relazione **Costituito** attraverso una funzione, risparmiando 5000 accessi giornalieri.

## Eliminazione generalizzazioni

Sono presenti diverse generalizzazioni nello schema concettuale, procediamo ad analizzarle singolarmente al fine di effettuare la traduzione verso lo schema logico.

### Tavolo

L'entità Tavolo ha due entità figlie: TavoloInterno e TavoloEsterno, esse possiedono gli stessi attributi dell'entità padre, e anzi sono la stessa entità con la differenza che i tavoli sono divisi per la loro posizione, i tavoli situati nella sala interna e quelli nella sala esterna.

Si decide quindi di accorpate le entità figlie dentro l'entità padre andando ad aggiungere un nuovo attributo Sala che può assumere i valori "Interna" se il tavolo appartiene alla sala interna ed "esterna" se appartiene a quella esterna.

Ballarin Federico - 1123718

Bullo Valentina - 1123714

## Dipendente

Questa generalizzazione presenta due entità figlie principali: DipendenteSala e DipendenteCucina, che a loro volta hanno altre entità figlie, una per ogni mansione lavorativa.

Gli attributi della entità padre sono comuni a quelli di tutte le altre entità figlie e allo stesso modo le entità figlie non presentano attributi particolari che le differenziano tra di loro o dall'entità padre.

Sono presenti sia operazioni che non fanno distinzione fra le entità figlie e quella padre, sia operazioni che riguardano solamente delle entità figlie distinguendole dall'entità padre.

Considerando che la generalizzazione che si sta studiando è totale ed esclusiva viene presa la decisione di accorpare tutte le entità figlie in una nuova entità Mansione per identificare il ruolo del dipendente all'interno del ristorante.

Causa principale di questa scelta è il fatto che nel caso si fosse deciso di accorpare l'entità padre nelle entità figlie avremmo dovuto creare tante entità quante le entità figlie, ripetendo per ognuna di esse gli stessi dieci attributi con conseguente spreco di memoria.

## Cliente

ClienteCamping è un sottoinsieme dell'entità Cliente e possiede gli stessi attributi di quest'ultima.

Viene però deciso di mantenere l'informazione della provenienza dei clienti per monitorare la quantità dei clienti provenienti dal campeggio, si procede quindi con l'accorpamento dell'entità figlia ClienteCamping all'interno dell'entità padre Cliente con l'aggiunta di un attributo booleano Camping che permette di identificare i clienti provenienti dal campeggio (1) e non (0).

## Prodotto

L'entità Prodotto ha due entità figlie, ProdottoSala e ProdottoCucina, esse possiedono gli stessi attributi dell'entità padre, e anzi sono la stessa entità solo che vengono divisi i prodotti per il loro utilizzo in sala o in cucina.

Si decide quindi di accorpare le entità figlie dentro l'entità padre per evitare spreco di memoria andando ad aggiungere un nuovo attributo Utilizzo che identifica i prodotti utili per la sala con il valore "Sala", quelli per la cucina con il valore "Cucina" e quelli utili sia per la sala che per la cucina con il valore "Sala/Cucina".

## Ulteriori modifiche

### Introduzione entità Indirizzo

Per rappresentare le informazioni fornite dall'attributo composto indirizzo viene introdotta una nuova entità Indirizzo con gli attributi: Via, Città, Provincia, Regione. Le entità collegate ad Indirizzo sono l'entità Cliente e Fornitore. L'entità indirizzo è quindi così definita:

Indirizzo			
CodiceIndirizzo	VARCHAR	Identifica univocamente l'indirizzo	Chiave
Via	VARCHAR	Via dell'indirizzo	
Città	VARCHAR	Città dell'indirizzo	
Provincia	VARCHAR	Provincia dell'indirizzo	
Regione	VARCHAR	Regione dell'indirizzo	

### Ordine: Dipendente --> Prodotto

La relazione Ordine non è rappresentabile nel modello relazionale, né tantomeno implementabile in una base di dati reale. Decidiamo quindi di sostituirla con una nuova entità Ordine così definita:

Ordine			
Dipendente	VARCHAR	Identifica univocamente l'indirizzo	Chiave
Data	DATE	Via dell'indirizzo	
Prodotto	VARCHAR	Città dell'indirizzo	
Quantità	SMALLINT	Provincia dell'indirizzo	

Ballarin Federico - 1123718

Bullo Valentina - 1123714

## Traduzione nel modello relazionale

Tenendo in considerazione le modifiche effettuate sopra, andiamo ad ottenere la seguente progettazione logica.

**Dipendente**(CodiceDipendente, Nome, Cognome, Sesso, Indirizzo, Mansione, DataNascita, DataAssunzione, Stipendio, IBAN, NumeroTelefono)

*Vincolo di integrità referenziale tra Mansione in Dipendente e CodiceMansione in Mansione.*

*Vincolo di integrità referenziale tra Indirizzo in Dipendente e CodiceIndirizzo in Indirizzo.*

**Mansione**(CodiceMansione, Descrizione)

**Turno**(Dipendente, DataInizio, DataFine, OraInizio, OraFine)

*Vincolo di integrità referenziale tra Dipendente in Turno e CodiceDipendente in Dipendente.*

**Tavolo**(NumeroTavolo, NumeroPosti, Descrizione, Sala)

**Cliente**(CodiceCliente, Nome, Cognome, NumeroTelefono, Camping)

**Prenotazione**(CodicePrenotazione, Cliente, Tavolo, Data, Ora, NumeroPersone, Note)

*Vincolo di integrità referenziale tra Cliente in Prenotazione e CodiceCliente in Cliente.*

*Vincolo di integrità referenziale tra Tavolo in Prenotazione e NumeroTavolo in Tavolo.*

**ElementoMenù**(CodiceElemento, Nome, Descrizione, Prezzo, Gruppo)

*Vincolo di integrità referenziale tra Gruppo in ElementoMenù e CodiceGruppo in Gruppo.*

**Gruppo**(CodiceGruppo, Nome, Descrizione)

**Ordinazione**(CodiceOrdinazione, Operatore, Tavolo, Data, Ora)

*Vincolo di integrità referenziale tra Operatore in Ordinazione e CodiceDipendente in Dipendente.*

*Vincolo di integrità referenziale tra Tavolo in Ordinazione e NumeroTavolo in Tavolo.*

**DettaglioOrdine**(CodiceDettaglio, Ordinazione, Elemento, Quantità, Nota)

*Vincolo di integrità referenziale tra Ordine in DettaglioOrdine e CodiceOrdinazione in Ordinazione.*

*Vincolo di integrità referenziale tra Elemento in DettaglioOrdine e CodiceElemento in Elemento.*

**Pagamento**(CodicePagamento, Ordinazione, TipoPagamento, ImportoTotale, Banconiere, Sconto)

*Vincolo di integrità referenziale tra Ordinazione in Pagamento e CodiceOrdinazione in Ordinazione.*

*Vincolo di integrità referenziale tra Banconiere in Pagamento e CodiceDipendente in Dipendente.*

**Prodotto**(CodiceProdotto, Fornitore, Nome, Prezzo, Descrizione, Utilizzo)

*Vincolo di integrità referenziale tra Fornitore in Prodotto e CodiceFornitore in Fornitore.*

**Ordine**(Dipendente, Data, Prodotto, Quantità)

*Vincolo di integrità referenziale tra Dipendente in Ordine e CodiceDipendente in Dipendente.*

*Vincolo di integrità referenziale tra Prodotto in Ordine e CodiceProdotto in Prodotto.*

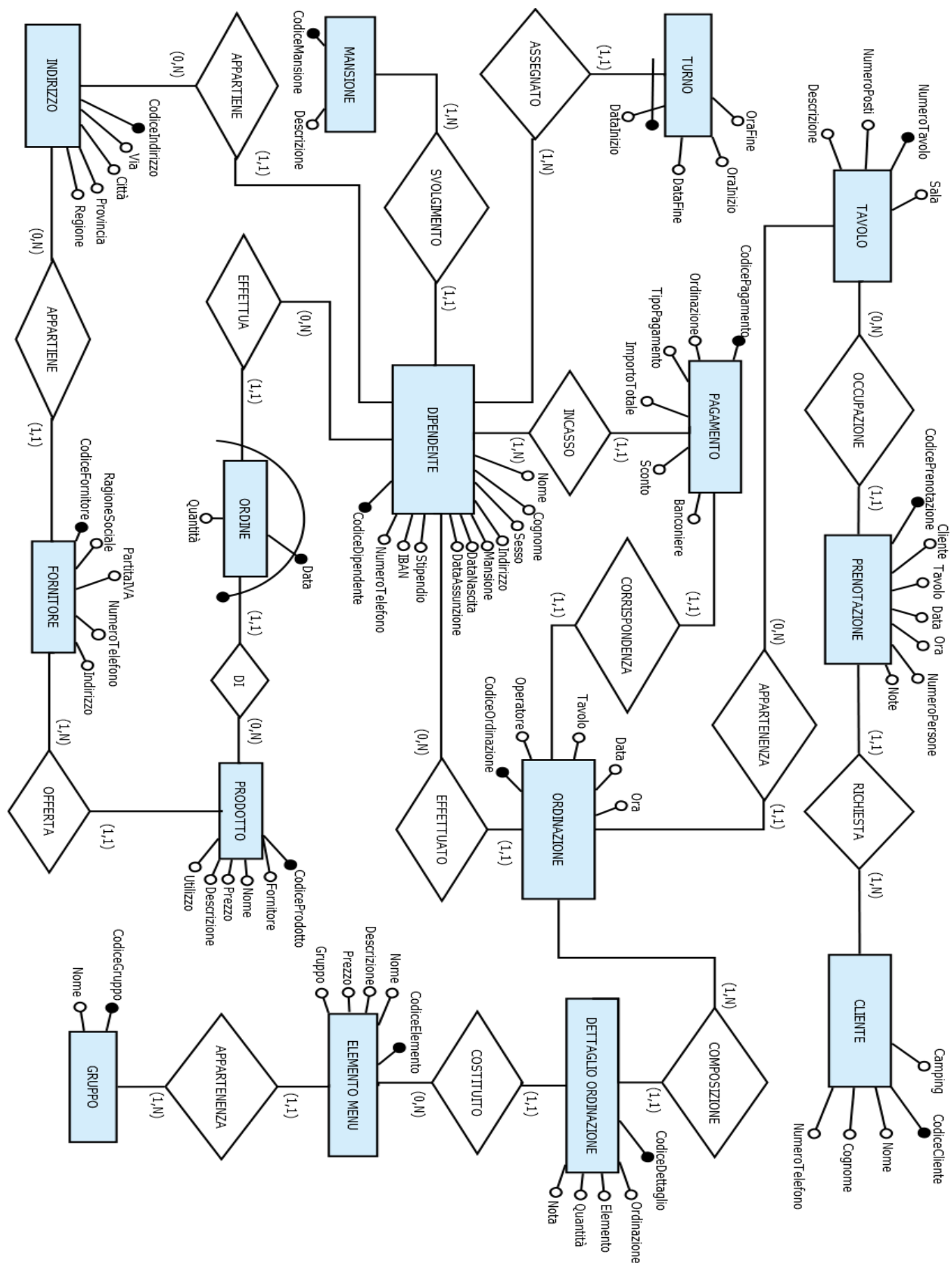
**Fornitore**(CodiceFornitore, RagioneSociale, PartitaIVA, Indirizzo, NumeroTelefono)

*Vincolo di integrità referenziale tra Indirizzo in Fornitore e CodiceIndirizzo in Indirizzo.*

**Indirizzo**(CodiceIndirizzo, Via, Città, Provincia, Regione)

Bullo Valentina - 1123714

## Schema logico ristrutturato



## Query, procedure, trigger e funzioni

### Funzioni

1. La funzione **TotaleOrdinazione** effettua il calcolo del prezzo totale di una specifica ordinazione al tavolo. Essa prende in input il CodiceOrdinazione dell'ordinazione.

DELIMITER !!

```
CREATE FUNCTION TotaleOrdinazione(CO VARCHAR(20)) RETURNS DECIMAL(7,2)
```

```
BEGIN
```

```
DECLARE Totale DECIMAL(7,2);
```

```
SELECT SUM(Quantità*Prezzo) INTO Totale
```

```
FROM DettaglioOrdinazione JOIN ElementoMenù ON Elemento=CodiceElemento
```

```
WHERE Ordinazione=CO;
```

```
IF Totale IS NULL THEN
```

```
SET Totale=0;
```

```
END IF;
```

```
RETURN Totale;
```

```
END!!
```

```
DELIMITER ;
```

2. La funzione **TotaleOrdine** effettua il calcolo del costo totale di uno specifico ordine di un prodotto. Essa prende in input la chiave della tabella ordine.

DELIMITER !!

```
CREATE FUNCTION TotaleOrdine(dip VARCHAR(20), dt DATE, pr VARCHAR(20)) RETURNS  
DECIMAL(7,2)
```

```
BEGIN
```

```
DECLARE Totale DECIMAL(7,2);
```

```
SELECT SUM(Quantità*Prezzo) INTO Totale
```

```
FROM Ordine JOIN Prodotto ON Prodotto=CodiceProdotto
```

```
WHERE Dipendente=dip AND Data=dt AND Prodotto=pr;
```

```
IF Totale IS NULL THEN
```

```
SET Totale=0;
```

```
END IF;
```

```
RETURN Totale;
```

```
END!!
```

```
DELIMITER ;
```

3. La funzione **NuOrdinazioniElemento** calcola il numero di volte che uno specifico elemento è stato ordinato ai tavoli in un determinato periodo di tempo. Essa prende in input il nome del prodotto e la data di inizio e fine dell'arco temporale.

DELIMITER !!

```
CREATE FUNCTION NuOrdinazioniElemento(NomeE VARCHAR(20), DataI DATE, DataF DATE)  
RETURNS INTEGER
```

```
BEGIN
```

```
DECLARE Totale INTEGER;
```

```
SELECT SUM(Quantità) INTO Totale
```

```
FROM Ordinazione JOIN DettaglioOrdinazione ON CodiceOrdinazione=Ordinazione JOIN ElementoMenù  
ON Elemento=CodiceElemento
```

```
WHERE Nome=NomeE AND DataI <= Data AND DataF >= Data;
```

```
IF Totale IS NULL THEN
```

```
SET Totale=0;
```

```
END IF;
```

```
RETURN Totale;
```

```
END!!
```

```
DELIMITER ;
```

Ballarin Federico - 1123718

Bullo Valentina - 1123714

4. La funzione **TotaleDettaglio** calcola il prezzo totale di uno specifico dettaglio di una ordinazione. Essa prende in input il codice del dettaglio che si vuole analizzare.

DELIMITER !!

```
CREATE FUNCTION TotaleDettaglio(CD VARCHAR(20)) RETURNS DECIMAL(7,2)
```

```
BEGIN
```

```
DECLARE Totale DECIMAL(7,2);
```

```
SELECT Quantità*Prezzo INTO Totale
```

```
FROM DettaglioOrdinazione JOIN ElementoMenù ON Elemento=CodiceElemento
```

```
WHERE CodiceDettaglio=CD;
```

```
IF Totale IS NULL THEN
```

```
SET Totale=0;
```

```
END IF;
```

```
RETURN Totale;
```

```
END!!
```

```
DELIMITER ;
```

## Query e procedure

1. Query che mostra i cinque elementi più venduti del menu mostrando per ogni elemento anche il numero di volte che è stato ordinato ed il totale dei guadagni incassati dalla vendita.

```
SELECT CodiceElemento, nome, SUM(Quantità) AS Ordinazioni, SUM(Quantità*Prezzo) AS
```

```
TotaleGuadagnato
```

```
FROM ElementoMenù JOIN DettaglioOrdinazione ON CodiceElemento=Elemento
```

```
GROUP BY CodiceElemento, nome
```

```
ORDER BY Ordinazioni DESC, TotaleGuadagnato DESC
```

```
LIMIT 5;
```

### OUTPUT

CodiceElemento	nome	Ordinazioni	TotaleGuadagnato
E02	bollito misto	4	80.00
E07	tiramisù	4	20.00
E06	prosciutto e melone	3	39.00
E04	filetto di branzino	2	26.00
E01	CocaCola	2	6.00

2. Query che mostra per ogni cameriere il numero di ordinazioni che hanno effettuato ed il numero medio di elementi ordinati.

```
DROP VIEW IF EXISTS NuEleOrd;
```

```
CREATE VIEW NuEleOrd AS
```

```
SELECT Ordinazione, SUM(Quantità) AS ElePerOrd
```

```
FROM DettaglioOrdinazione
```

```
GROUP BY Ordinazione;
```

```
SELECT CodiceDipendente, Nome, COUNT(*) AS NumeroOrdinazioni, AVG(ElePerOrd) AS
```

```
MediaElementi
```

```
FROM Dipendente JOIN Ordinazione ON CodiceDipendente=Operatore JOIN NuEleOrd ON
```

```
CodiceOrdinazione=Ordinazione
```

```
GROUP BY CodiceDipendente, nome;
```

### OUTPUT

CodiceDipendente	Nome	NumeroOrdinazioni	MediaElementi
D05	Claudia	2	4.5000
D10	Mark	1	7.0000

Ballarin Federico - 1123718  
Bullo Valentina - 1123714

**3. Query che mostra tutti i clienti provenienti dal campeggio che hanno effettuato almeno due prenotazioni**

```
SELECT CodiceCliente, Nome, COUNT(*) AS NuPrenotazioni
FROM Cliente JOIN Prenotazione ON CodiceCliente=Cliente
WHERE Camping='1'
GROUP BY CodiceCliente
HAVING NuPrenotazioni>1;
```

**OUTPUT**

CodiceCliente	Nome	NuPrenotazioni
C02	Paolo	3

**4. Query che mostra tutti i gruppi riguardanti piatti che si mangiano (bibite escluse) ordinati per numero di elementi appartenenti a ciascun gruppo**

```
SELECT CodiceGruppo, Gruppo.Nome, Count(*) AS ElementiGruppo
FROM Gruppo JOIN ElementoMenù ON CodiceGruppo=Gruppo
WHERE Gruppo.Nome<>'bibite'
GROUP BY CodiceGruppo, Gruppo.Nome;
ORDER BY ElementiGruppo DESC;
```

**OUTPUT**

CodiceGruppo	Nome	ElementiGruppo
G02	antipasti	2
G03	primi	2
G04	secondi	2
G05	contorni	1
G06	piatti freddi	1
G07	dessert	1

**5. Query che calcola il totale dei giorni lavorativi dei dipendenti della sala mostrandone nome e cognome.**

```
SELECT Nome, Cognome, COUNT(*) AS GiornateLavorative
FROM Turno JOIN Dipendente ON Dipendente=CodiceDipendente
WHERE Mansione='M01' OR Mansione='M02' OR Mansione='M03'
GROUP BY Nome, Cognome
ORDER BY GiornateLavorative DESC;
```

**OUTPUT**

Nome	Cognome	GiornateLavorative
Federico	Ballarin	2
David	Varisco	1
Valentina	Bullo	1

**6. Query come la precedente ma per i dipendenti della cucina.**

```
SELECT Nome, Cognome, COUNT(*) AS GiornateLavorative
FROM Turno JOIN Dipendente ON Dipendente=CodiceDipendente
WHERE Mansione='M04' OR Mansione='M05' OR Mansione='M06'
GROUP BY Nome, Cognome
ORDER BY GiornateLavorative DESC;
```

**OUTPUT**

Nome	Cognome	GiornateLavorative
Jack	Tchamp	2
Gianni	Marella	1
Gianfranco	Agatea	1



Ballarin Federico - 1123718  
Bullo Valentina - 1123714

- 7. Procedura che elimina una particolare prenotazione, riceve il codice della prenotazione in input. Molto utile In quanto succede molto spesso che a causa pioggia i clienti decidano di annullare la prenotazione essendo il ristorante in spiaggia.**

DELIMITER !!

```
CREATE PROCEDURE EliminaPrenotazione(pid VARCHAR(20))
```

```
BEGIN
```

```
DELETE FROM Prenotazioni WHERE pid=CodicePrenotazione;
```

```
END !!
```

```
DELIMITER ;
```

- 8. Procedura che elimina uno specifico ordine, riceve in input il dipendente, la data e il prodotto.**

DELIMITER !!

```
CREATE PROCEDURE EliminaOrdini(did VARCHAR(20), dt DATE, pd VARCHAR(20) )
```

```
BEGIN
```

```
DELETE FROM Ordine WHERE did=Dipendente AND dt=Data AND pd=Prodotto;
```

```
END !!
```

```
DELIMITER ;
```

## Trigger

- 1. Trigger che controlla che i valori inseriti nella tabella Turno siano corretti. Cioè che la data di inizio non sia maggiore di quella di fine, che l'ora di inizio non sia maggiore di quella di fine e che la data di inizio non sia minore della data di assunzione.**

DELIMITER !!

```
CREATE TRIGGER CheckTurni BEFORE INSERT ON Turno
```

```
FOR EACH ROW
```

```
BEGIN
```

```
DECLARE msg VARCHAR(255);
```

```
DECLARE dataass DATE;
```

```
SELECT DataAssunzione INTO dataass FROM Dipendente WHERE CodiceDipendente=New.Dipendente;
```

```
IF New.OraInizio >= New.OraFine OR dataass>New.DataInizio OR New.DataInizio >= New.DataFine
```

```
THEN
```

```
SET msg='ERRORE! I valori inseriti sono errati';
```

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
```

```
END IF;
```

```
END!!
```

```
DELIMITER ;
```

- 2. Trigger che controlla che gli ordini dei prodotti vengano effettuati dai responsabili e che ordinino solo prodotti riguardanti la loro area di competenza.**

DELIMITER !!

```
CREATE TRIGGER CheckOrdini BEFORE INSERT ON Ordine
```

```
FOR EACH ROW
```

```
BEGIN
```

```
DECLARE msg VARCHAR(255);
```

```
DECLARE uso VARCHAR(20);
```

```
DECLARE mans VARCHAR(20);
```

```
SELECT Utilizzo INTO uso FROM Prodotto WHERE New.Prodotto=CodiceProdotto;
```

```
SELECT Descrizione INTO mans FROM Dipendente JOIN Mansione ON Mansione=CodiceMansione
```

```
WHERE New.Dipendente=CodiceDipendente;
```

```
IF mans<>'caposala' AND mans<>'chef' THEN
```

```
SET msg='ERRORE! Dipendente non autorizzato';
```

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
```

Ballarin Federico - 1123718

Bullo Valentina - 1123714

```
ELSEIF (mans='chef' AND uso='sala') OR (mans='caposala' AND uso='cucina') THEN  
SET msg='ERRORE! Prodotto non ordinabile';  
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;  
END IF;  
END!!  
DELIMITER ;
```

**3. Trigger che controlla che l'importo totale di ogni pagamento corrisponda al totale degli elementi ordinati meno l'eventuale sconto.**

```
DROP TRIGGER IF EXISTS CheckPagamenti;  
DELIMITER !!  
CREATE TRIGGER CheckPagamenti BEFORE INSERT ON Pagamento  
FOR EACH ROW  
BEGIN  
DECLARE msg VARCHAR(255);  
DECLARE tot DECIMAL(7,2);  
SELECT SUM(Quantità*Prezzo) INTO tot FROM DettaglioOrdinazione JOIN ElementoMenù ON  
Elemento=CodiceElemento WHERE dettaglioordinazione.Ordinazione = New.Ordinazione;  
IF New.ImportoTotale <> (tot - New.Sconto) THEN  
SET msg='ERRORE! importo totale non corrisponde';  
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;  
END IF;  
END!!  
DELIMITER ;
```