

# Relazione di Basi di dati

Anna Cisotto Bertocco

Sara Righetto

5 febbraio 2019

## Sommario

‘La Sofia’ è un’osteria-enoteca situata nel cuore del centro storico della città di Padova, in via Gabelli, a due passi dalla chiesa di Santa Sofia.

Specializzata in cucina veneta, può ospitare fino a 35 coperti ed è gestita dai due proprietari, lo chef Massimo Bertocco e il sommelier Gianni Banzato. A completamento dello staff, un aiuto cuoco e due cameriere.

I servizi offerti dal ristorante variano dalle classiche colazioni, pranzi e cene, ad ottimi e convenienti rinfreschi per lauree, rendendo la gestione del suddetto, anche se piccolo, esigente di tempo e attenzione: è infatti necessario tenere quotidianamente sotto controllo la disponibilità delle provviste in magazzino e in cantina, gestire la contabilità, in particolare le bollette e le buste paga, oltre a dirigere i dipendenti, accogliere gli ospiti con le prenotazioni che ne derivano.

Da qualche anno ‘La Sofia’ è inoltre in possesso di svariati account social, come Facebook e Twitter, che richiedono un continuo e costante aggiornamento, così da garantire un contatto quanto più stretto possibile con i clienti.

In tutto questo sorge la necessità di creare una base di dati efficiente, che alleggerisca il peso della gestione e porti a una migliore organizzazione, con conseguente potenziamento e perfezionamento del servizio offerto.

# 1 Analisi dei requisiti

## 1.1 Descrizione

Si vuole realizzare una base di dati che gestisca in maniera efficiente le informazioni relative all'organizzazione del ristorante "La Sofia" a Padova.

In particolare si vogliono conoscere i dati relativi a: provviste nel magazzino e nella cantina, contabilità (entrate ed uscite), dipendenti, turni di lavoro e prenotazioni effettuate per il locale.

Di ciascun **prodotto** si vuole sapere:

- Codice identificativo progressivo numerico
- Nome
- Quantità presente nel magazzino
- Prezzo
- Fornitore

Un prodotto può essere o un cibo o una bevanda.

Un cibo può essere fresco, a lunga conservazione o surgelato. Per ogni cibo si vuole sapere anche la data di scadenza.

Una bevanda, di cui si vuole sapere la marca, può essere non alcolica o alcolica. Gli alcolici si dividono in vini e superalcolici. Per entrambi si vuole sapere sia l'annata che il tipo.

Per i prodotti da frigo, o freschi, si vuole inoltre conoscere la data di scadenza.

Per quanto riguarda i rifornimenti, per ogni **fornitore** si vuole sapere:

- Partita IVA che lo identifica univocamente
- Nome azienda
- Recapito telefonico

Riguardo alla contabilità, di ogni **uscita ed entrata** si vuole sapere:

- Codice identificativo progressivo numerico
- Costo
- Data

La contabilità comprende sia le entrate che le uscite.

Di ogni uscita deve essere specificata anche la motivazione o causale che ha generato il costo.

Ogni uscita può corrispondere o all'acquisto di un prodotto o al salario di un dipendente.

Ogni entrata corrisponde ad un eventuale incasso derivante da un'ordine.

Di ogni **Ordine** si vuole sapere:

- Numero tavolo
- Numero di persone

- Cameriere che serve l'ordine
- Conto totale
- Giorno
- Ora

Ogni ordine è identificato dal numero del tavolo, il giorno e l'ora.  
Ogni ordine è preso e servito da un cameriere.

Per le **prenotazioni** effettuate per il locale si vuole sapere:

- Numero del tavolo
- Giorno della prenotazione
- Ora della prenotazione
- Nome del prenotante
- Numero di persone al tavolo

Ogni prenotazione è identificata univocamente da giorno, ora e numero del tavolo.  
Il numero di persone per tavolo non può superare il numero di coperti disponibili per quel tavolo.

Di ogni **Tavolo** si vuole sapere:

- Numero del tavolo che lo identifica univocamente
- Quantità di coperti disponibili

Riguardo al **personale** invece si vuole sapere:

- Codice fiscale, che identifica ogni dipendente univocamente
- Nome
- Cognome
- Giorno libero
- Stipendio

Il personale si divide in cuochi, sommelier o camerieri.

Il giorno di riposo non può corrispondere ad un giorno nei turni di lavoro.  
Solo i camerieri possono servire ai tavoli.

Riguardo ai **turni di lavoro** all'interno del locale, si vuole sapere:

- Codice del dipendente
- Giorno

Ogni turno di lavoro viene identificato univocamente dal giorno.

Ogni turno di lavoro comprende l'intera giornata, senza distinzione tra pranzo e cena.

## 1.2 Glossario dei termini

<b>Termine</b>	<b>Descrizione</b>	<b>Sinonimi</b>	<b>Collegamenti</b>
Prodotto	Cibo o bevanda presente nel magazzino	Provviste	Fornitore, Contabilità
Fornitore	Azienda fornitrice di prodotti	Azienda	Prodotto
Contabilità	Uscite ed entrate del ristorante	Spesa	Prodotto, Ordine, Personale
Ordine	Ordine effettuato dai clienti ad un determinato tavolo		Contabilità, Personale, Tavolo
Prenotazione	Prenotazione per un tavolo del ristorante		Tavolo
Tavolo	Tavolo		Prenotazione, Ordine
Personale	Dipendenti a servizio del locale	Dipendente	Contabilità, Ordine, Turni
Turni	Turni lavorativi dei dipendenti		Personale

### 1.3 Operazioni

**Operazione 1:** inserimento di un nuovo prodotto

**Operazione 2:** cancellazione di un nuovo prodotto

**Operazione 3:** modificare la quantità di un prodotto

**Operazione 4:** inserimento di una nuova prenotazione

**Operazione 5:** cancellazione di una prenotazione

**Operazione 6:** inserimento di un nuovo dipendente

**Operazione 7:** licenziamento di un dipendente

**Operazione 8:** modifica di un turno di lavoro

**Operazione 9:** stampa il nome, il codice e la quantità di tutti i cibi con data di scadenza a febbraio 2019 in ordine crescente di data

**Operazione 10:** stampa il codice, il nome, la marca e l'annata di tutti i vini con annata risalente a massimo il 2008 e prezzo inferiore a 20 euro ordinati per prezzo crescente

**Operazione 11:** stampa il numero dei tavoli liberi nel giorno 20/02/2019 con almeno 3 coperti

**Operazione 12:** stampa nome e cognome di tutti i dipendenti di turno il lunedì

**Operazione 13:** calcola il guadagno mensile dato dalla differenza tra la somma delle entrate e la somma delle uscite

**Operazione 14:** trova il cameriere che effettua più ordini in media al giorno per decidere se promuoverlo o no

## 1.4 Strutturazione dei requisiti

### **Frase di carattere generale**

Si vuole realizzare una base di dati che gestisca in maniera efficiente le informazioni relative all'organizzazione del ristorante "La Sofia" a Padova. In particolare si vogliono conoscere i dati relativi a provviste nel magazzino e nella cantina, contabilità (entrate ed uscite), dipendenti, turni di lavoro e prenotazioni effettuate per il locale.

### **Frase relative a prodotto**

Ogni prodotto ha: un codice identificativo progressivo numerico, un nome, una quantità, un prezzo ed un fornitore.

Un prodotto può essere o un cibo o una bevanda. Un cibo può essere fresco, a lunga conservazione o surgelato. Una bevanda, di cui si vuole sapere la marca, può essere non alcolica o alcolica. Gli alcolici si dividono in vini e superalcolici. Per entrambi si vuole sapere sia l'annata che il tipo. Per i vini i tipi possono essere rosso, bianco fermo, rosè, frizzante, passito. Per i superalcolici i tipi sono grappa, amaro, whisky, rum, aperol. Per i prodotti da frigo, o freschi, si vuole inoltre conoscere la data di scadenza.

### **Frase relative a fornitore**

Ogni fornitore ha: un codice dell'azienda corrispondente alla Partita IVA che lo identifica univocamente, un nome ed un recapito telefonico.

Ogni fornitore vende uno o più prodotti al ristorante.

### **Frase relative a contabilità**

Ogni entrata ed uscita ha: un codice identificativo progressivo numerico, un costo ed una data.

La contabilità comprende sia le entrate che le uscite. Di ogni uscita deve essere specificata anche la motivazione o causale che ha generato il costo. Ogni uscita può corrispondere o all'acquisto di un prodotto o al salario di un dipendente. Ogni entrata corrisponde ad un eventuale incasso derivante da un'ordine.

### **Frase relative a ordine**

Ogni ordine ha: un numero corrispondente al tavolo, il numero di persone per tavolo il cameriere che serve al tavolo, un totale, il giorno e l'ora.

Ogni ordine è identificato dal numero del tavolo, il giorno e l'ora. Ogni ordine è preso e servito da un cameriere.

### **Frase relative a prenotazione**

Ogni prenotazione ha: un nome del cliente prenotante, il giorno della prenotazione, l'ora della prenotazione, un numero di persone al tavolo. Ogni prenotazione è identificata univocamente da giorno, ora e numero del tavolo. Il numero di persone per tavolo non può superare il numero di coperti disponibili per quel tavolo.

#### **Frase relative a tavolo**

Ogni tavolo ha: un numero del tavolo che lo identifica univocamente, una quantità di coperti.

Ad un tavolo può corrispondere una prenotazione. Per ogni tavolo può corrispondere un ordine.

#### **Frase relative a personale**

Ogni dipendente ha: un codice fiscale, che identifica ogni dipendente univocamente, un nome ed un cognome, il giorno di riposo, uno stipendio.

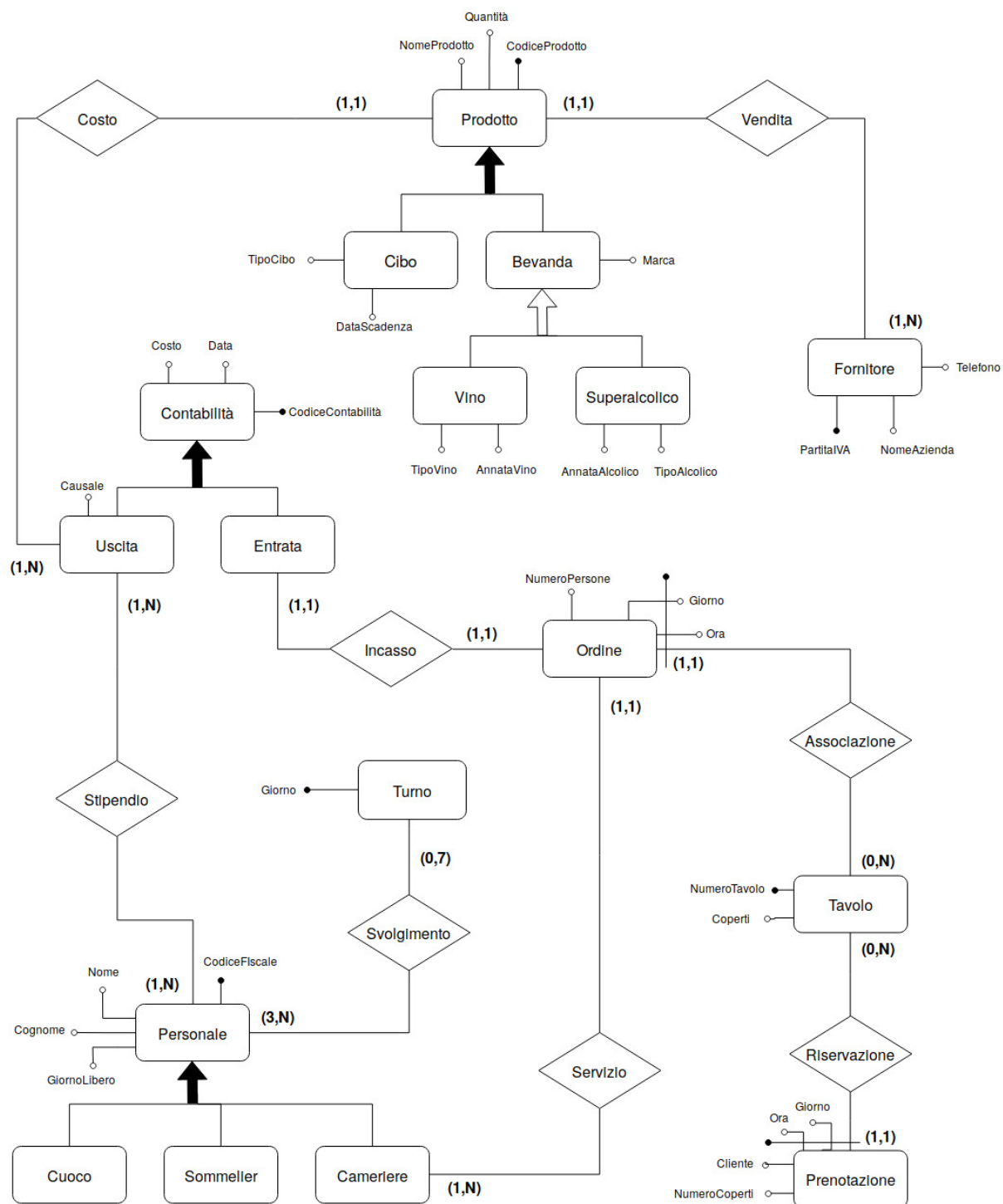
Il personale si divide in cuochi, sommelier o camerieri. Ogni dipendente non può coprire un turno di lavoro nel suo giorno libero. Solo i camerieri possono servire ai tavoli.

#### **Frase relative a turni di lavoro**

Ogni turno di lavoro ha: un codice del dipendente ed un giorno.

Ogni turno di lavoro viene identificato univocamente dal giorno. Ogni turno di lavoro comprende l'intera giornata, senza distinzione tra pranzo e cena.

## 2 Schema concettuale



### 2.1 Dizionario dei dati

#### Regole di vincolo

- Un dipendente non può svolgere più di 6 turni a settimana.
- Un cliente non può prenotare un tavolo già prenotato.

- Il numero del tavolo deve essere compreso tra 1 e 35.
- La quantità del prodotto non può essere negativa.

#### Lista delle entità

Entità	Descrizione	Attributi	Identificatore
Prodotto	Provvista del magazzino	CodiceProd, NomeProdotto, Quantità	CodiceProd
Cibo	Prodotto commestibile	Eredita da Prodotto, TipoCibo, DataScadenza	CodiceProd
Bevanda	Prodotto da bere	Eredita da prodotto, Marca	CodiceProd
Vino	Bevanda alcolica a base di uva	Eredita da bevanda, AnnataVino, TipoVino	CodiceProd
Superalcolico	Bevanda con grado alcolico elevato	Eredita da Bevanda, TipoAlcolico, AnnataAlcolico	CodiceProd
Fornitore	Azienda fornitrice di prodotti	Partitalva, NomeAzienda, Telefono	Partitalva
Contabilità	Gestione del denaro del locale	CodiceCont, Costo, Data	CodiceCont
Entrata	Entrata di denaro del ristorante	Eredita da contabilità	CodiceCont
Uscita	Uscita di denaro dal ristorante	Eredita da contabilità, Causale	CodiceCont
Ordine	Ordine effettuato per un determinato tavolo	NumeroTavolo, NumeroPersone, Ora, NumeroGiorno, Ora	NumeroTavolo, NumeroGiorno, Ora
Prenotazione	Prenotazione per un tavolo del ristorante	NumeroTavolo, NumeroCoperti, Ora, Cliente	NumeroTavolo, NumeroGiorno, Ora
Tavolo	Tavolo presente nel locale	NumeroTavolo, Coperti	NumeroTavolo
Personale	Insieme dei dipendenti del locale	CodiceFiscale, Nome, Cognome, GiornoLibero	CodiceFiscale
Cuoco	Dipendente addetto alla cucina	Eredita da personale	CodiceFiscale
Sommelier	Dipendente addetto alla mescita degli alcolici	Eredita da personale	CodiceFiscale
Cameriere	Dipendente che serve ai tavoli	Eredita da personale	CodiceFiscale

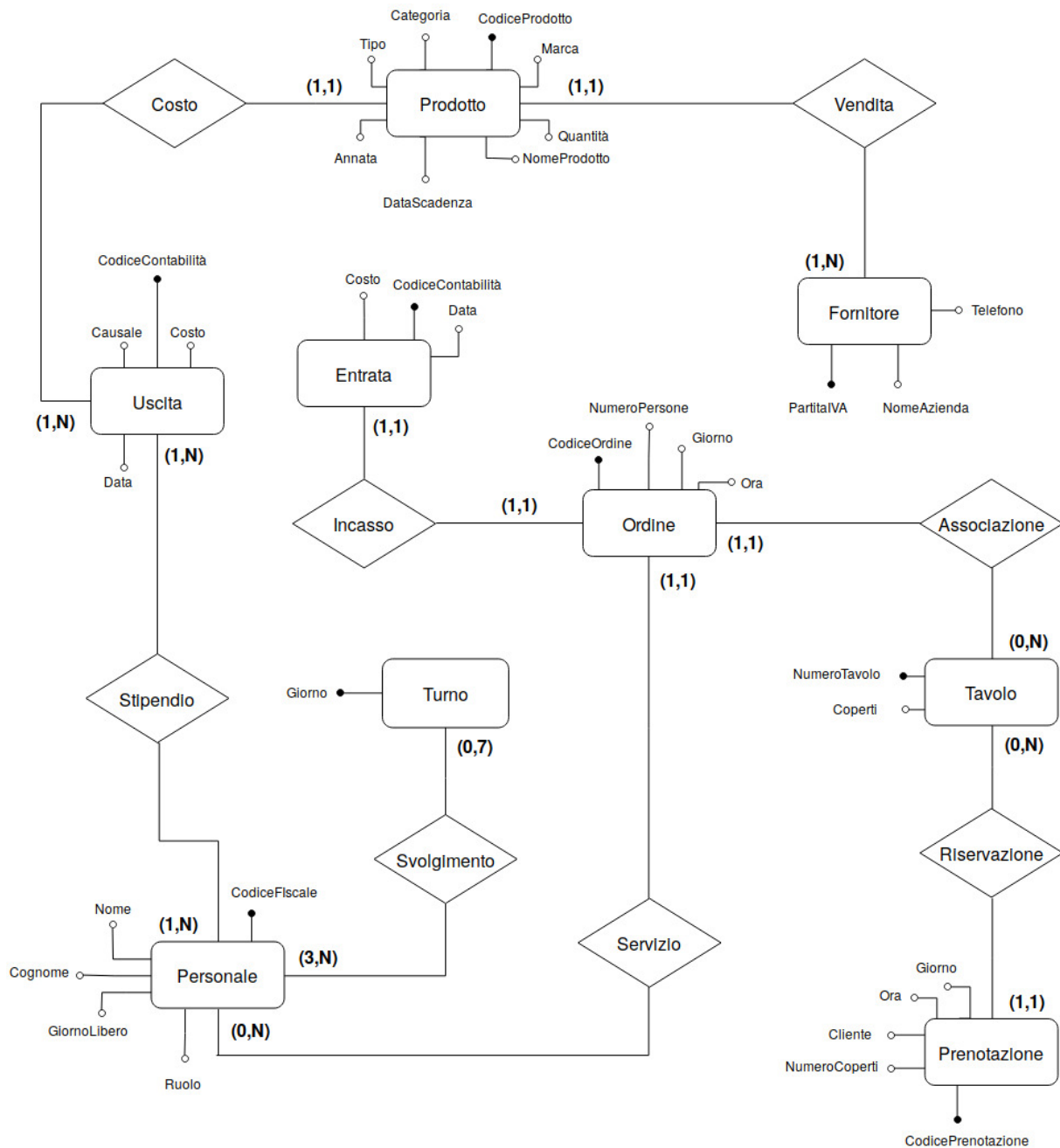


Turno	Turno di lavoro	Giorno	Giorno
-------	-----------------	--------	--------

### Lista delle relazioni

Relazione	Descrizione	Cardinalità	Attributi
Vendita	Associa il prodotto al suo fornitore	Prodotto(1,1) Fornitore(1,N)	
Costo	Associa il prezzo del prodotto all'uscita di denaro	Prodotto(1,1) Uscita(1,N)	
Incasso	Associa l'ordine all'entrata di denaro	Ordine(1,1) Entrata(1,1)	
Associazione	Associa l'ordine al tavolo	Ordine(1,1) Tavolo(0,N)	
Riservazione	Associa il tavolo alla prenotazione	Tavolo(0,N) Prenotazione(1,1)	
Servizio	Associa l'ordine al cameriere	Ordine(1,1) Cameriere(1,N)	
Svolgimento	Associa il dipendente al turno di lavoro	Personale(3,N) Turno(0,7)	
Stipendio	Associa lo stipendio del dipendente all'uscita del ristorante	Dipendente(1,N) Uscita(1,N)	

### 3 Schema concettuale ristrutturato



#### Eventuali note:

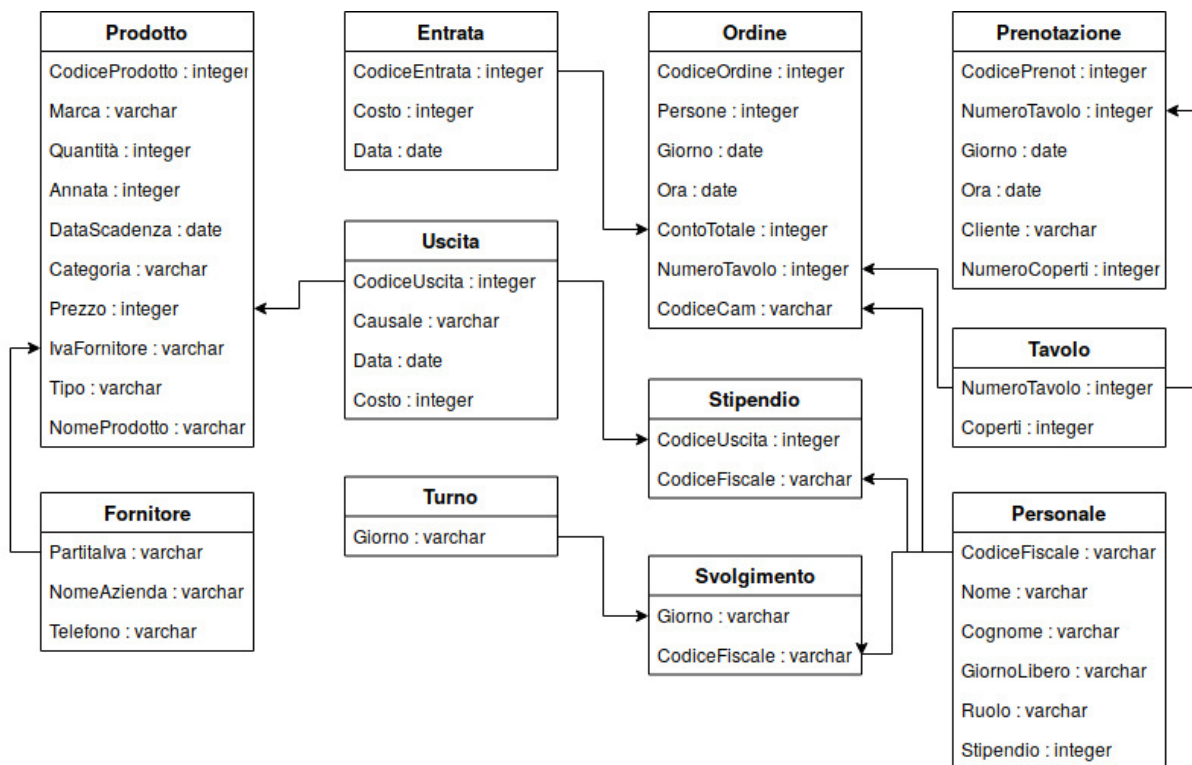
- Dato che i sistemi tradizionali per la gestione per le basi di dati non consentono di rappresentare direttamente una *generalizzazione* né tantomeno una gerarchia, risulta necessario trasformare questo costrutto in entità e relazioni. Per la nostra base di dati si è deciso di utilizzare due dei tre tipi di eliminazione: Accorpamento delle figlie della generalizzazione nel genitore e accorpamento del genitore della generalizzazione nelle figlie. Il primo metodo è stato utilizzato per l'eliminazione della gerarchia che coinvolge Vino, Superalcolico e Bevanda, per quella che coinvolge Cibo, Bevanda e Prodotto aggiungendo un attributo Categoria che distingue tra cibo e bevanda; il primo

metodo viene usato anche per la generalizzazione che comprende Personale, Cuoco, Sommelier e Cameriere, dove in Personale è stato aggiunto un attributo Ruolo.

Il secondo metodo è stato usato per accorpare Contabilità con le figlie Uscita ed Entrata.

- I sistemi di gestione di basi di dati richiedono generalmente di specificare una chiave primaria; nei casi in cui esistono entità per le quali sono specificati più identificatori bisogna decidere quali di questi usare come chiave primaria. Dato che un identificatore composto da uno o pochi attributi è da preferire a identificatori costituiti da molti attributi e per gli stessi motivi, un identificatore interno è da preferire a uno esterno gli *identificatori esterni* delle entità Ordine e Prenotazione vengono sostituiti con CodiceOrdine e CodicePrenotazione.

## 4 Schema logico



### 4.1 Schema logico-relazionale

**Fornitore**(PartitaIVA, NomeAzienda, Telefono),

**Prodotto**(CodiceProdotto, Categoria, Marca, Tipo, Annata, DataScadenza, NomeProdotto, Quantità, IVAFornitore, Prezzo),

**Uscita**(CodiceUscita, Costo, Data, Causale),

**Stipendio**( CodiceUscita, CodiceFiscale ),

**Entrata**(CodiceEntrata, Costo, Data),

**Personale**(CodiceFiscale, Nome, Cognome, Ruolo, GiornoLibero, Stipendio),  
**Turno**(Giorno),  
**Svolgimento**(Giorno, CodiceFiscale),  
**Ordine**(CodiceOrdine, NumeroPersone, Giorno, Ora, ContoTotale, NumeroTavolo, CodiceCameriere),  
**Tavolo**(NumeroTavolo, Coperti),  
**Prenotazione**(CodicePren, NumeroCoperti, Cliente, Giorno, Ora, NumeroTavolo)

## 5 Lista dei vincoli di integrità referenziale

- Fra l'attributo PartitaIVA della relazione Fornitore e l'attributo IVAFornitore della relazione Prodotto.
- Fra l'attributo CodiceUscita della relazione Uscita e l'attributo Prezzo della relazione Prodotto.
- Fra l'attributo CodiceUscita della relazione Uscita e la relazione Stipendio.
- Fra l'attributo CodiceFiscale della relazione Stipendio e la relazione Personale.
- Fra l'attributo CodiceEntrata della relazione Entrata e l'attributo ContoTotale della relazione Ordine.
- Fra l'attributo Giorno della relazione Turno e la relazione Svolgimento.
- Fra l'attributo CodiceFiscale della relazione Svolgimento e la relazione Personale.
- Fra l'attributo CodiceFiscale della relazione Personale e l'attributo CodiceCameriere della relazione Ordine.
- Fra l'attributo NumeroTavolo della relazione Tavolo e la relazione Ordine.
- Fra l'attributo NumeroTavolo della relazione Tavolo e la relazione Prenotazione.

## 6 Implementazione

### 6.1 Query

#### Operazione 9

---

```
1 -- stampa il nome, il codice e la quantita di tutti i cibi con data di scadenza a febbraio 2019 in ordine  
  crescente di data --  
2  
3 CREATE VIEW CibiInScadenza as  
4 SELECT DISTINCT C.CodiceProdotto, C.NomeProdotto, C.Quantita  
5 FROM prodotto AS C  
6 WHERE C.Categoria = 'Cibo' AND DataScadenza >= '2019-02-01' AND DataScadenza <= '  
  2019-02-28'  
7 ORDER BY DataScadenza;
```

---

#### Operazione 10

---

```
1 -- stampa il codice, il nome, la marca e l'annata di tutti i vini con annata risalente a massimo il 2008 e  
  prezzo inferiore a 20 euro  
2 -- ordinati per prezzo crescente --  
3  
4 CREATE VIEW viniEconomici as  
5 SELECT DISTINCT V.CodiceProdotto, V.NomeProdotto, V.Marca, V.Annata  
6 FROM prodotto AS V  
7 WHERE V.Categoria='Bevanda' AND V.Tipo='Vino' AND V.Annata <= 2008 AND V.Prezzo < 20  
8 ORDER BY Prezzo;
```

---

#### Operazione 11

---

```
1 -- stampa il numero dei tavoli liberi nel giorno 20/02/2019 con almeno 3 coperti  
2  
3 CREATE VIEW tavoliLberi as  
4 SELECT T.NumeroTavolo  
5 FROM tavolo AS T  
6 WHERE T.Coperti >= 3 AND T.NumeroTavolo NOT IN  
7 (SELECT P.NumeroTavolo FROM prenotazione AS P WHERE P.Giorno='2019-02-20');
```

---

#### Operazione 12

---

```
1 -- stampa nome e cognome di tutti i dipendenti di turno il lunedì --  
2  
3 CREATE VIEW dipendentiDiTurno as  
4 SELECT DISTINCT P.Nome, P.Cognome  
5 FROM personale AS P, svolgimentoTurno AS T  
6 WHERE T.CodiceFiscale=P.CodiceFiscale AND T.Giorno = 'Lunedì';
```

---

#### 6.1.1 Procedure

### Operazione 1

---

```
1 -- operazione 1: inserimento nuovo prodotto --
2 DELIMITER |
3 CREATE PROCEDURE nuovoProdotto (IN codice INTEGER, IN nome VARCHAR(20), IN marca
  VARCHAR(20), IN quantita INTEGER, IN scadenza DATE, IN annata DATE, IN categoria
  VARCHAR(20), IN tipo VARCHAR(20), IN prezzo FLOAT, IN fornitore VARCHAR(11))
4 BEGIN
5   START TRANSACTION;
6   INSERT INTO prodotto (CodiceProdotto, NomeProdotto, Marca, Quantita, Annata, DataScadenza,
    Categoria, Tipo, Prezzo, IvaFornitore) VALUES
7     (codice, nome, marca, quantita, annata, scadenza, categoria, tipo, prezzo, fornitore);
8   COMMIT;
9 END |
10 DELIMITER ;
```

---

### Operazione 2

---

```
1 -- operazione 2: cancellazione di un prodotto --
2 DELIMITER |
3 CREATE PROCEDURE eliminaProdotto (IN codice INTEGER)
4 BEGIN
5   DELETE FROM prodotto WHERE CodiceProdotto=codice;
6 END |
7 DELIMITER ;
```

---

### Operazione 3

---

```
1 -- modificare la quantita di un prodotto --
2 DELIMITER |
3 CREATE PROCEDURE modificaQuantita (IN codice INTEGER, IN quantita INTEGER)
4 BEGIN
5   UPDATE prodotto AS P
6   SET P.Quantita = P.Quantita + quantita
7   WHERE P.CodiceProdotto=codice;
8 END |
9 DELIMITER ;
```

---

### Operazione 4

---

```
1 -- operazione 4: inserimento di una nuova prenotazione --
2 DELIMITER |
3 CREATE PROCEDURE nuovaPrenotazione (IN codice INTEGER, IN coperti INTEGER, IN
  nomeCliente VARCHAR(20), IN giorno DATE, IN ora DATE, IN tavolo INTEGER)
4 BEGIN
5   START TRANSACTION;
6   INSERT INTO prenotazione(CodicePrenotazione, NumeroCoperti, Cliente, Giorno, Ora,
    NumeroTavolo) VALUES
7     (codice, coperti, nomeCliente, giorno, ora, tavolo);
8   COMMIT;
9 END |
10 DELIMITER ;
```

---

### Operazione 5

---

```
1  -- operazione 5: cancellazione di una prenotazione --
2 DELIMITER |
3 CREATE PROCEDURE eliminaPrenotazione (IN codice INTEGER)
4 BEGIN
5     DELETE FROM prenotazione WHERE CodicePrenotazione=codice;
6 END |
7 DELIMITER ;
```

---

### Operazione 6

---

```
1  -- operazione 6: inserisci nuovo dipendente --
2 DELIMITER |
3 CREATE PROCEDURE nuovoDipendente (IN cf VARCHAR(20), IN nome VARCHAR(20), IN cognome
    VARCHAR(20), IN giorno_libero VARCHAR(10), IN ruolo VARCHAR(20), IN stipendio FLOAT)
4 BEGIN
5     INSERT INTO personale (CodiceFiscale, Nome, Cognome, GiornoLibero, Ruolo, Stipendio)
        VALUES
6         (cf, nome, cognome, giorno_libero, ruolo, stipendio);
7 END |
8 DELIMITER ;
```

---

### Operazione 7

---

```
1  -- operazione 7: licenzia dipendente --
2 DELIMITER |
3 CREATE PROCEDURE licenziaDipendente (IN dipendente VARCHAR(20))
4 BEGIN
5     DELETE FROM personale WHERE CodiceFiscale=dipendente;
6 END |
7 DELIMITER ;
```

---

### Operazione 8

---

```
1  -- operazione 8: modifica turno di lavoro --
2 DELIMITER |
3 CREATE PROCEDURE modificaTurno (IN giorno VARCHAR(10), IN dipendente VARCHAR(20))
4 BEGIN
5     UPDATE svolgimento_turno SET CodiceFiscale=dipendente
6     WHERE Giorno=giorno;
7 END |
8 DELIMITER ;
```

---

## 6.2 Funzioni

### Operazione 13

---

```
1  -- calcola il guadagno mensile dato dalla differenza tra la somma delle entrate e la somma delle
    uscite --
2
3 DELIMITER |
```

```

4 CREATE FUNCTION guadagnoTot (mese DATE) RETURNS FLOAT
5 BEGIN
6     DECLARE Guadagno FLOAT;
7     DECLARE entrate FLOAT;
8     DECLARE uscite FLOAT;
9
10    SELECT SUM(U.Costo) INTO uscite
11    FROM uscita AS U
12    WHERE U.DataUscita >= '2019' + MONTH(mese) + '01' AND U.DataUscita <= '2019' + MONTH(
        mese) + '31';
13
14    SELECT SUM(E.Costo) INTO entrate
15    FROM entrata AS E
16    WHERE E.DataEntrata >= '2019' + MONTH(mese) + '01' AND E.DataEntrata <= '2019' + MONTH
        (mese) + '31';
17
18    SET Guadagno = entrate—uscite;
19
20    RETURN Guadagno;
21
22 END |

```

---

#### Operazione 14

```

1 -- trova il cameriere che effettua piu ordini in media al giorno per decidere se promuoverlo o no --
2
3 CREATE FUNCTION promozione (mese DATE, cameriere VARCHAR(20)) RETURNS BOOL
4 BEGIN
5     DECLARE ordini INTEGER;
6     DECLARE promozione BOOL;
7
8     SELECT COUNT(*) INTO ordini
9     FROM ordine AS O
10    WHERE O.CodiceCameriere=cameriere AND O.Giorno >= '2019' + MONTH(mese) + '01' AND O.
        giorno <= '2019' + MONTH(mese) + '31';
11
12    IF ordini >= 20 THEN SET promozione=true;
13    ELSE SET promozione=FALSE;
14    END IF;
15
16    RETURN promozione;
17
18 END |

```

---

## 6.3 Trigger

#### Regola di vincolo 1

```

1 -- un dipendente non puo lavorare nel suo giorno libero --
2 CREATE TRIGGER noGiorniLiberi
3 BEFORE INSERT ON svolgimentoTurno
4 FOR EACH ROW
5 BEGIN

```



```

6 DECLARE giornoLibero VARCHAR(10);
7
8 SELECT P.GiornoLibero INTO giornoLibero
9 FROM personale AS P;
10
11 IF giornoLibero NOT IN (SELECT T.Giorno FROM personale AS P, svolgimentoTurno AS T
12 WHERE P.CodiceFiscale=T.CodiceFiscale)
13 THEN CALL modificaTurno(new.giorno, new.dipendente);
14 END IF;
15 END |
16 DELIMITER ;

```

---

### Regola di vincolo 2

---

```

1 -- un dipendente non puo svolgere piu di 6 turni a settimana --
2 DELIMITER |
3 CREATE TRIGGER MaxTurni
4 BEFORE INSERT ON svolgimentoTurno
5 FOR EACH ROW
6 BEGIN
7 DECLARE turni INTEGER;
8
9 SELECT COUNT(CodiceFiscale) INTO turni
10 FROM svolgimentoTurno
11 WHERE CodiceFiscale=new.CodiceFiscale;
12
13 IF turni <= 6 THEN CALL inserisciTurno;
14 END IF;
15
16 END |
17 DELIMITER ;

```

---

### Regola di vincolo 3

---

```

1 -- Un cliente non puo prenotare un tavolo gia prenotato. --
2 DELIMITER |
3 CREATE TRIGGER tavoliPrenotati
4 BEFORE INSERT ON prenotazione
5 FOR EACH ROW
6 BEGIN
7 IF new.tavolo NOT IN (SELECT P.tavolo FROM prenotazione AS P)
8 THEN CALL inserisciPrenotazione;
9 END IF;
10 END |
11 DELIMITER ;

```

---