

## **SSD Dati persistenti**

### **1. Pasticceria**

- a. id: int(10)
- b. nome: varchar(50)
- c. email: varchar(30)
- d. descrizione: varchar(500)
- e. indirizzo: varchar(100)
- f. telefono: varchar(30)
- g. urlLogo: varchar(60)
- h. urlWebSite: varchar(60)

La tabella Pasticceria memorizza i dati relativi alle pasticcerie registrate al sistema aggiunte dagli utenti registrati come proprietari. È identificata univocamente dall'id, un numero intero.

### **2. Prodotto**

- a. codice: int(10)
- b. nome: varchar(60)
- c. quantità: int(10)
- d. minScorta: int(10)
- e. prezzo: decimal(10,2)
- f. pasticceria: int(10)

La tabella prodotto memorizza i dati relativi ai prodotti di un inventario. Un loro insieme andrà a comporre le ricette di una pasticceria. È identificata univocamente dal codice, un numero intero. Ogni prodotto verrà associato alla relativa pasticceria per via della chiave esterna pasticceria.

### **3. Ricetta**

- a. codice: int(10)
- b. nome: varchar(30)
- c. ore: int(10)

- d. minuti: int(10)
- e. procedimento: varchar(200)
- f. prezzoAcquisto: decimal(10,2)
- g. prezzoVendita: decimal(10,2)
- h. insale: tinyint(1)
- i. pasticceria: int(10)

La tabella Ricetta memorizza tutti i dati delle ricette memorizzate nel sistema per permettere al pasticciere di consultare quest'ultima quando dovrà portare a termine una prenotazione effettuata da un cliente. È identificata univocamente dal codice, un numero intero.

Ogni ricetta sarà associata alla relativa pasticceria per via della chiave esterna pasticceria

#### **4. Uscita**

- a. id: int(10)
- b. descrizione: varchar(50)
- c. data: date
- d. importo: decimal(10,2)
- e. tipo: varchar(30)
- f. pasticceria: int(10)

La tabella Uscita memorizza tutte le spese sostenute da una pasticceria. È identificata univocamente dall'id, un numero intero.

Ogni uscita sarà associata alla relativa pasticceria per via della chiave esterna pasticceria

## 5. Entrata

- a. id: int(10)
- b. descrizione: varchar(50)
- c. data: date
- d. importo: decimal(10,2)
- e. pasticceria: int(10)

La tabella Entrata memorizza tutte i guadagni di una pasticceria. È identificata univocamente dall'id, un numero intero.

Ogni entrata sarà associata alla relativa pasticceria per via della chiave esterna pasticceria

## 6. Utente

- a. email: varchar(30)
- b. password: varchar(30)
- c. nome: varchar(30)
- d. cognome: varchar(30)
- e. telefono: varchar(30)
- f. ruolo: varchar(20)
- g. numPrenotazioni: int(10)
- h. pasticceria: int(10)

La tabella Utente memorizza tutti i dati degli Utenti che si registrano al sistema: clienti, proprietari di pasticcerie e dipendenti (magazziniere,

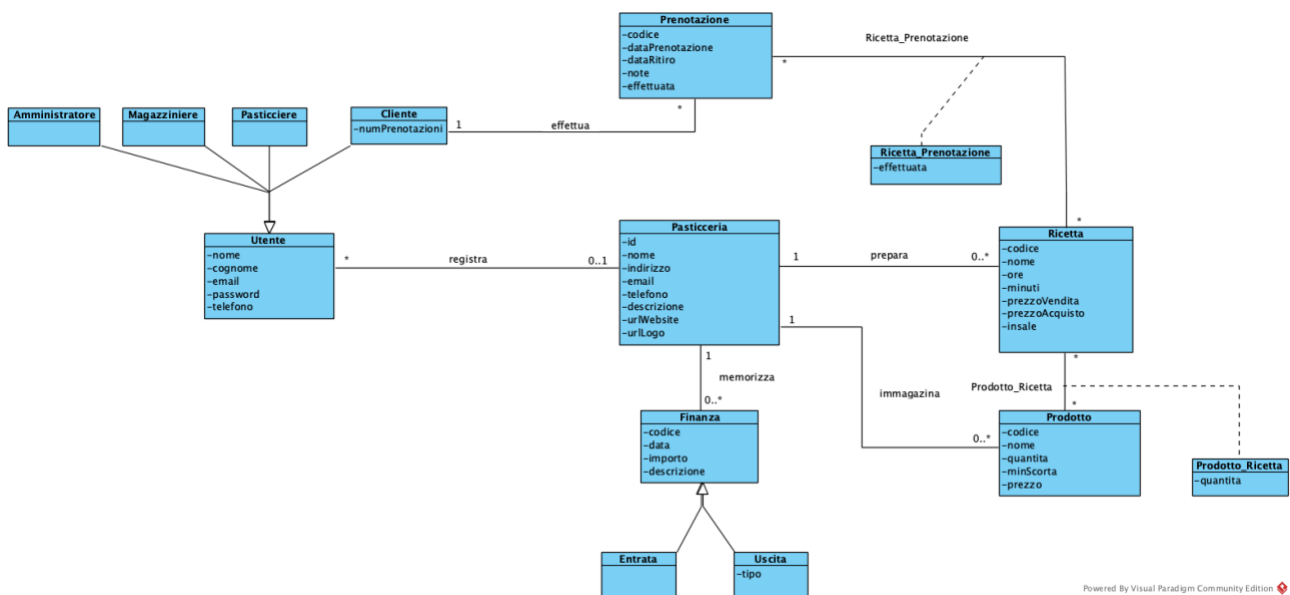
pasticciere). È identificata univocamente dal campo email. Ogni utente, a meno del cliente, viene associato ad una pasticceria per mezzo della chiave esterna pasticceria.

## 7. Prenotazione

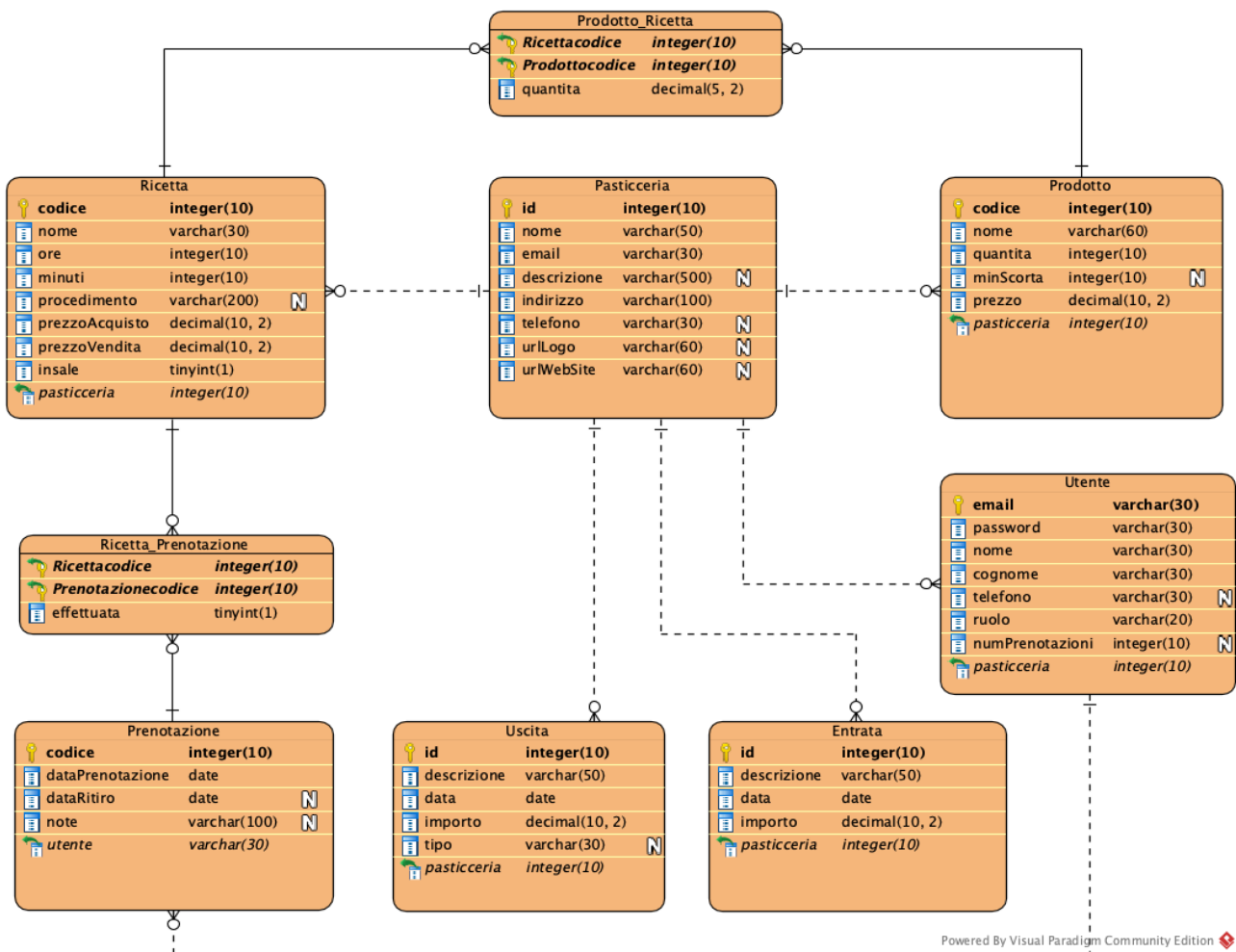
- a. codice: int(10)
- b. dataPrenotazione: date
- c. dataRitiro: date
- d. note: varchar(100)
- e. utente: varchar(30)

La tabella Prenotazione memorizza i dati relativi ad una prenotazione effettuata da un generico cliente che si è autenticato al sistema. È identificata univocamente dal codice, un numero intero. Ogni prenotazione viene associata ad uno specifico utente per mezzo della chiave esterna utente.

## Diagrammi dei dati persistenti



## Schema logico



## Motivazioni

Si è scelto di usare un DBMS di tipo relazionale poiché esso permette di accedere in modo semplice e efficiente ad una base di dati mantenendone la consistenza, la privacy e l'affidabilità.

I vantaggi dell'uso di un DBMS sono i seguenti:

**Accesso ai dati tramite linguaggio SQL:** tale linguaggio permette la creazione delle strutture che contengono i dati, l'inserimento, la cancellazione, l'aggiornamento dei dati e il recupero delle informazioni dalla base di dati.

**Accesso efficiente ai dati:** un DBMS ha molti modi per ottimizzare l'accesso all'informazione. La base di dati è solitamente memorizzata in memoria secondaria (disco rigido). Un DBMS permette di creare dei file ausiliari (indici) che permettono l'accesso veloce ai dati su disco. Inoltre, spesso un DBMS mantiene porzioni della base di dati in memoria centrale velocizzando in

questo modo l'accesso ai dati. Infine, ogni interrogazione prima di essere eseguita viene ottimizzata scegliendo un piano efficiente di esecuzione sulla base degli indici esistenti.

**Indipendenza dei dati:** un DBMS permette di accedere ai dati logici indipendentemente dalla loro rappresentazione fisica. Quest'ultima può cambiare senza che i metodi di accesso ai dati logici debbano essere modificati. Si parla di indipendenza fisica dei dati.

**Accesso concorrente ai dati:** un DBMS permetta a più utenti di accedere contemporaneamente alla base di dati. Più utenti possono accedere nello stesso istante a dati diversi. Inoltre, un DBMS fa in modo che l'accesso concorrente agli stessi dati non generi anomalie, cioè inconsistenza nello stato della base di dati rispetto alla realtà modellata.

**Imposizione di vincoli di integrità sui dati.**

**Privacy dei dati.**