

Università degli Studi di Salerno

Corso di Ingegneria del Software

Pizzeria “La Saporita”

Gestione dati persistenti

Versione 2.0



Data 23/01/19

Partecipanti:

Nome	Matricola
Michela Giovanna Scarpone	0512104490
Raffaella Romano	0512103774

Scritto da:	Scarpone/Romano
--------------------	-----------------

Revision History

Data	Versione	Descrizione	Autore
02/01/19	1.0	Prima revisione del documento	Scarpone/Romano
23/01/19	2.0	Ultima revisione del documento	Scarpone/Romano

Indice

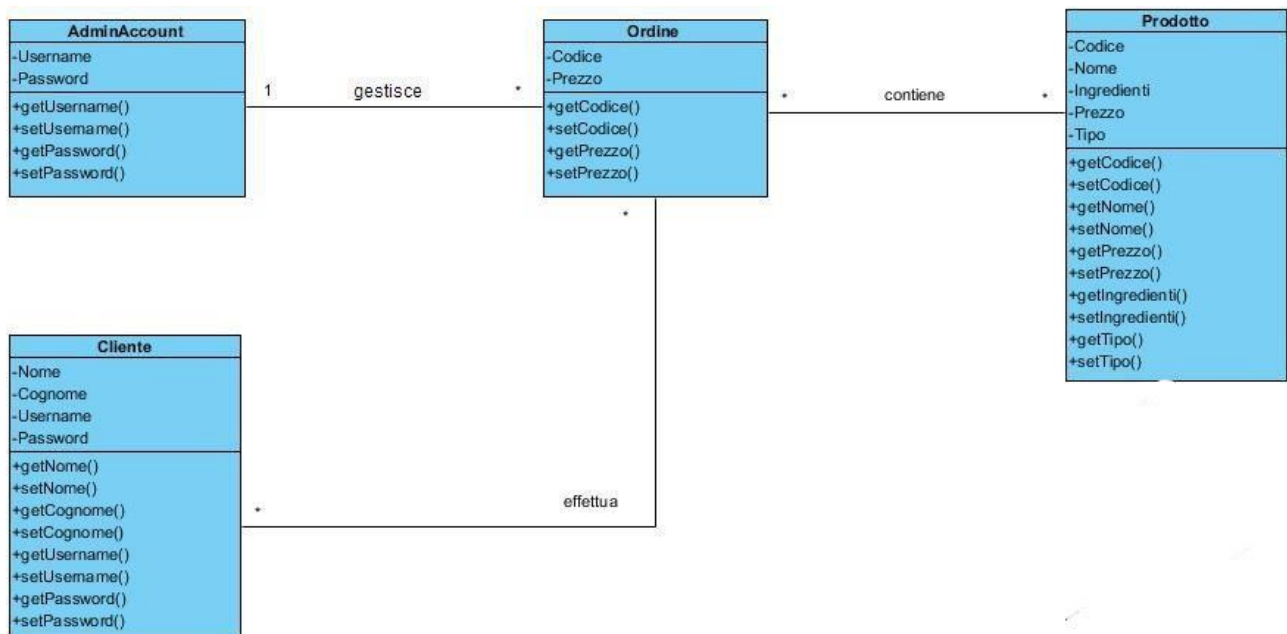
- **Gestione dei dati persistenti**
- **Diagramma Entity-Relationship**
- **Struttura delle tabelle**

Gestione dei dati persistenti

Per memorizzare i dati persistenti abbiamo scelto un database relazionale in modo che possa essere gestito agevolmente l'accesso concorrente ai dati e l'utilizzo del DBMS garantisca la consistenza dei dati stessi.

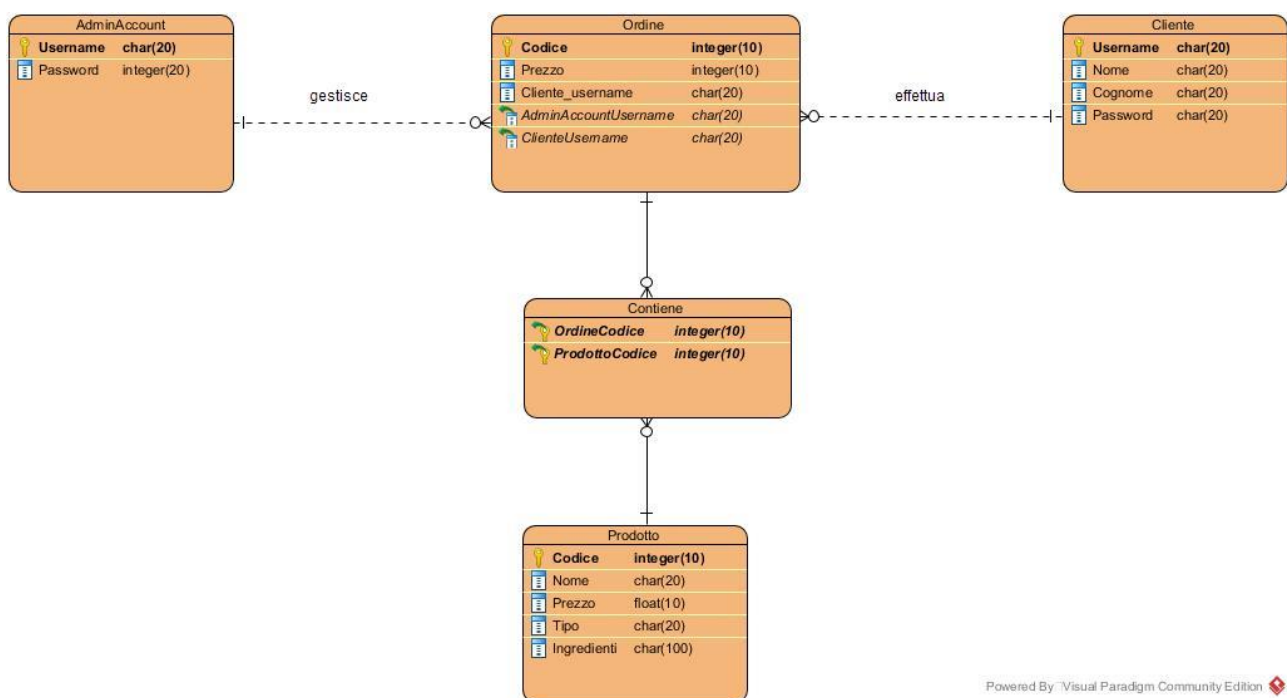
La scelta di un DBMS è stato il giusto compromesso per essere il più possibile coerenti con i design goals stabiliti, infatti, questa soluzione porta diversi vantaggi quali:

- **Privatezza dei dati:** Un DBMS permette un accesso protetto ai dati. Utenti diversi possono avere accesso a diverse porzioni della base di dati e possono essere abilitati a diverse operazioni su di esse.
- **Imposizioni di vincoli di integrità sui dati:** Un DBMS permette di specificare diversi tipi di vincoli per mantenere l'integrità dei dati e controlla che tali vincoli siano soddisfatti quando la base di dati cambia
- **Atomicità delle operazioni:** Un DBMS permette di effettuare sequenze di operazioni in modo atomico. Ciò significa che l'intera sequenza di operazioni viene eseguita con successo oppure nessuna di queste operazioni ha alcun effetto sui dati della base. L'atomicità delle transazioni permette di mantenere uno stato della base di dati consistente con la realtà modellata.
- **Affidabilità dei dati:** Un DBMS offre dei metodi per salvare copie dei dati e per ripristinare lo stato della base di dati in caso di guasti software e hardware.



- Le informazioni dell'attore Amministratore e del Responsabile , vengono memorizzate e quindi rappresentate dalla tabella AdminAccount dove il tipo va ad identificare le due tipologie.
- Le informazioni dell'utente, sono memorizzate e rappresentate dalla tabella Cliente dove l'username va ad identificare ogni singolo utente registrato nel sistema.
- Gli utenti sono connessi attraverso la relazione effettua ordine, che salva le informazioni relative agli ordini del sistema, così da specificare che ogni utente può effettuare uno o più ordini di diversi prodotti.
- Le informazioni riguardanti gli ordini dei prodotti sono collegate tramite la relazione contiene che specifica il prodotto o i prodotti che sono stati ordinati.
- Le informazioni riguardanti i prodotti sono contenute nella tabella prodotto che è collegata agli ordini in quanto si possono effettuare ordini in riferimento ai prodotti.
- Le informazioni riguardanti le offerte sono contenute nella tabella Offerte che è collegata alla tabella Prodotto tramite la relazione riguarda.

Diagramma Entity-Relationship



Struttura delle Tabelle

Di seguito sono riportate tutte le tabelle che vanno a formare il nostro database per la gestione di tutte le informazioni del sistema. In ogni tabella è indicato: il nome, il compito e tutti gli attributi ad essa associati, con relativi vincoli e tipo.

adminAccount		
Campo	Vincolo	Tipo
Username	Lunghezza massima: 20 cifre, chiave primaria	char
Password	Lunghezza massima: 20 cifre, not null	char

Cliente		
Campo	Vincolo	Tipo
Username	Lunghezza massima: 20 cifre, chiave primaria	Char
Nome	Lunghezza massima: 20 cifre, not null	char
Cognome	Lunghezza massima: 20 cifre, not null	char
Password	Lunghezza massima: 20 cifre, not null	char

Ordine		
Campo	Vincolo	Tipo
codice	chiave primaria, not null	Integer
Prezzo	Lunghezza default	Float
Cliente_username	Lunghezza massima: 20 cifre, foreign key	Char

Contiene		
Campo	Vincolo	Tipo
Prodotto_Codice	chiave esterna	Integer
Ordine_Codice	chiave esterna	Integer

Prodotto

Campo	Vincolo	Tipo
Codice	chiave primaria	Integer
Nome	Lunghezza massima: 20 cifre, not null	char
Prezzo	Lunghezza massima: default, not null	Float
Tipo	Lunghezza massima: 20 cifre, not null	char
Ingredienti	Lunghezza massima: 100 cifre	char