|  |
| --- |
| **Università degli Studi di Salerno Corso di Ingegneria del Software** |

**GAP  
System Design Document (Dati Persistenti)**

****

Data: 04/12/2021

**Partecipanti:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Giammarino Emanuele | 0512108088 |
| Adinolfi Giacinto | 0512107764 |

**5. Dati persistenti**

GAP ha necessità di salvare diversi dati persistenti come prodotti, utenti, fatture, ecc. È risultato opportuno, quindi, di adottare un Database di tipo relazionale come supporto al salvataggio di tali dati. La scelta di un DBMS rapido e relativamente leggero per svolgere tale compito è ricaduta su MySQL che offre numerosi vantaggi quali:

• Gestione della concorrenza;

• Integrità dei dati;

• Backup e recupero dei dati;

• Efficienza ed indipendenza.

Le interazioni con il database (connessione e query) sono realizzate utilizzando le API JDBC.

**5.1 Descrizione delle entità del Database**

**Utente***: indica gli utenti registrati al database, ma si riferisce anche ai vari gestori del database.*

CF char(16)

Nome varchar(15)

Cognome varchar(15)

Email varchar(50)

Passw varchar(50)

Indirizzo\_Fatturazione varchar(40)

Telefono varchar(50)

primary key(Email)

**Utente\_Ruolo***: indica le associazioni tra gli utenti del database e i ruoli a loro assegnati.*

Email varchar(50)

Ruolo varchar(15)

primary key(Email, Ruolo)

foreign key (Email) references Utente(Email)

foreign key (Ruolo) references Ruolo(Nome)

**Ruolo***: indica l’insieme dei ruoli che si possono ricoprire all’interno del sistema.*

Nome varchar(15) not null

primary key(nome)

**Prodotto***: indica i prodotti che si possono acquistare e che sono presenti all’interno del catalogo del sito web.*

Codice int

Nome varchar(25)

Altezza int

Profondità int

Larghezza int

Tipologia varchar(25)

Quantita int

Prezzo float

Sconto int

primary key(Codice)

**Recensione***: indica le recensioni scritte dagli utenti*

Testo varchar(250)

Data\_Recensione date

Codice int

Email char(50)

primary key (Email,Codice)

foreign key (Email) references utente(Email)

foreign key (Codice) references Prodotto (Codice)

**Ordine***: indica gli ordini memorizzati all’interno del database, effettuati dal cliente*

Num\_Ordine int,

Email char(50)

Prezzo\_Totale int

Num\_Prodotti int

data\_Ordine varchar(15)

Numero\_Carta varchar(16)

Data\_Scadenza\_Carta date

CVV\_Carta varchar(3)

primary key(Num\_ordine)

foreign key(Email) references Utente(Email)

**Composizione\_Ordine***: indica, per ogni prodotto coinvolto all’interno di un ordine, la sua quantità e il materiale scelto per il prodotto.*

Num\_Ordine int

Codice int

Id\_Materiale varchar(20)

Quantita int

primary key(Num\_ordine, Codice, Id\_Materiale)

foreign key(Codice) references Prodotto(Codice)

foreign key(Num\_Ordine) references Ordine(Num\_ordine)

**Materiale***: indica i materiali con cui possono essere combinati i prodotti.*

Id int

Tipologia\_Materiale varchar(20)

Colore varchar(20)

primary key(Id)

**Composizione\_Prodotto***: indica i materiali che compongono un prodotto*

Id int

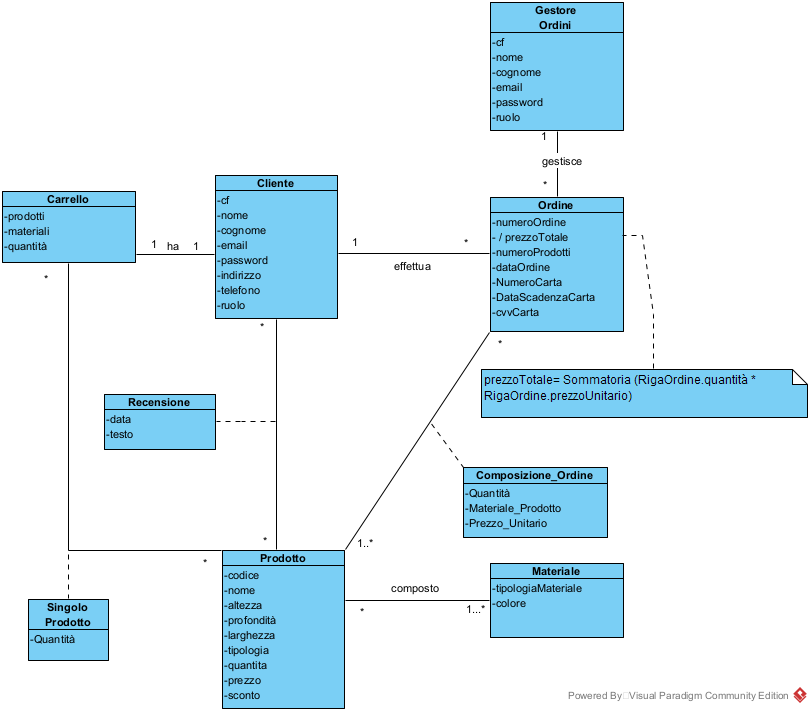
Codice int

primary key(Id, Codice)

foreign key(Codice) references Prodotto(Codice)

foreign key(Id) references Materiale(Id)

**5.2 Class Diagram**

****

**5.4 Schema logico**

