

# Base di dati pezzi di ricambio

[Della Monica Gerardo \(0512103166\)](#)

[Finelli Simone \(0512102992\)](#)

[Robustelli Antonio \(0512103130\)](#)

[Sessa Giuseppe \(0512102956\)](#)

---

## **Indice:**

### **1. Analisi dei requisiti**

- 1.1 Descrizione sintetica della realtà**
- 1.2 Descrizione formale del problema**
- 1.3 Glossario dei termini principale del dominio**

### **2. Progettazione concettuale**

- 2.1 Schema E-R**

### **3. Modello relazionale**

- 3.1 Realizzazione della base di dati**

# 1. Analisi e raccolta dei requisiti

## 1.1 Descrizione sintetica della realtà

Si vuole realizzare una base di dati che gestisca la vendita di pezzi di ricambio per auto. Un cliente, navigando sul sito sceglie il ricambio che serve per il proprio interesse e si registra per poter effettuare l'acquisto. Dopo aver selezionato i pezzi che cercava va ad effettuare l'acquisto.

## 1.2 Descrizione formale del problema

La base di dati che si vuole realizzare deve gestire i pezzi di ricambio acquistati dai clienti. A tal proposito, possiamo notare che un pezzo di ricambio, ossia un prodotto, può essere acquistato da uno e un solo utente e, viceversa, che un utente può acquistare più prodotti. I prodotti sono caratterizzati da un codice, dal modello, dal marchio, dal prezzo, dalla quantità e da una breve descrizione. Inoltre, un prodotto, potrebbe essere messo in offerta ed avere, quindi, un prezzo scontato.

Un utente, che si collega al sito di e-commerce mediante questo software, può registrarsi al sito e diventare un potenziale cliente.

Un utente registrato, avrà uno ed un solo indirizzo ma, viceversa, un indirizzo potrebbe corrispondere ad uno o più utenti registrati. Un indirizzo, in questo caso, è costituito da un CAP, da una via e da una città. Un utente registrato, invece, è caratterizzato da un codice fiscale che lo identifica univocamente, dal nome, dal cognome, dalla data di nascita, da un numero di telefono, da un numero di cellulare, da un'e-mail e da una password. L'utente registrato, però, potrebbe anche essere il gestore di tale sito e svolgere funzioni diverse.

Il software che si vuole realizzare, infatti, dovrà consentire ai gestori di: autenticarsi, visualizzare lo storico degli acquisti di tutti i clienti e caricare dei nuovi prodotti da poter vendere. Il software dovrà consentire agli utenti registrati di: autenticarsi, effettuare acquisti, tenere traccia degli acquisti effettuati, tenere traccia dei prodotti scelti per un eventuale acquisto, eliminare uno o più prodotti dalla lista dei prodotti scelti per un eventuale acquisto e affiliarsi ad officine presso cui montare i prodotti acquistati.

Il software, inoltre, deve consentire agli utenti la ricerca dei pezzi di ricambio tramite il loro codice, la loro marca oppure il loro modello. Ogni prodotto visualizzato dal sito di e-commerce, tramite il supporto del software, avrà anche

delle foto ad esso associato. In particolare, un prodotto ha una o più foto che lo descrivono ma, viceversa, una foto ha uno ed un solo prodotto a cui si riferisce.

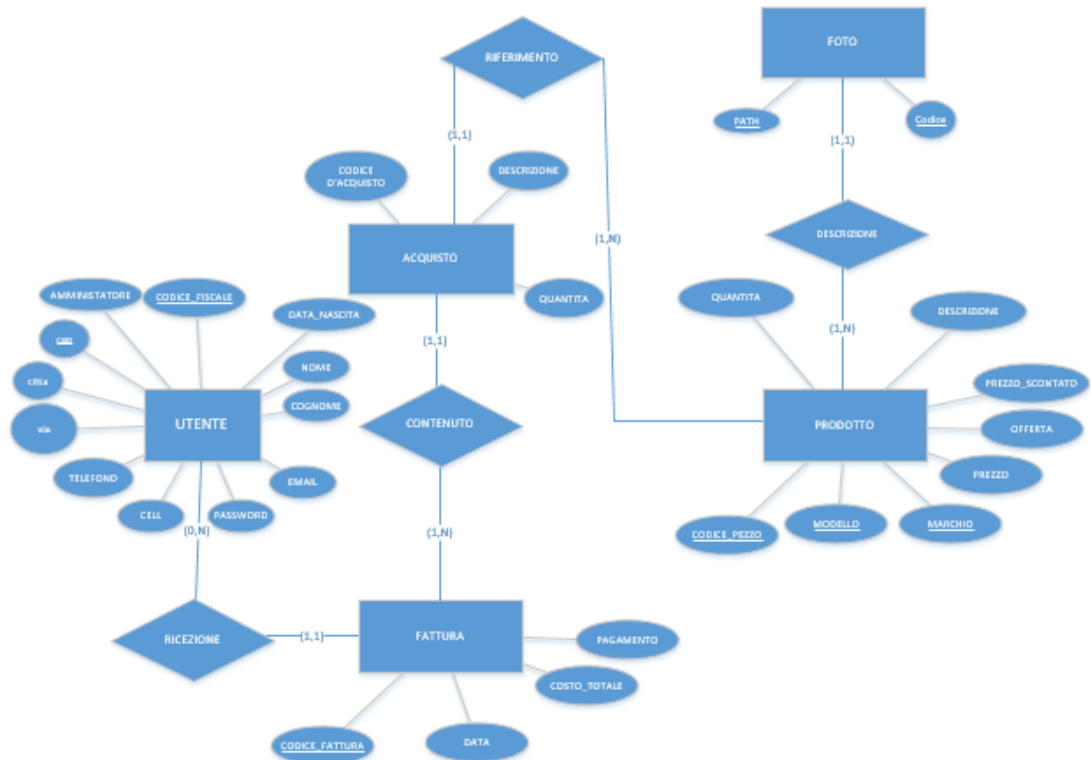
Il software, come già descritto in precedenza, deve tenere traccia degli acquisti effettuati dai vari clienti, i quali possono avere una o più fatture, viceversa, una fattura è in possesso di uno e un solo cliente, o utente registrato. Una fattura è caratterizzata da un codice fattura che la identifica univocamente, da una data di rilascio della stessa, da un costo totale e dal tipo di pagamento effettuato. Una fattura, di conseguenza, si riferisce ad uno o più prodotti acquistati in un certo momento. Un prodotto acquistato è caratterizzato da un codice di acquisto che lo identifica univocamente e da una descrizione.

### 1.3 Glossario dei termini principali del dominio

TERMINE	DESCRIZIONE	SINONIMI
<b>Utente</b>	<p>Persona che si registra al sito utilizzato per la vendita dei prodotti di ricambio online.</p> <p>Persona che assolve il ruolo di Gestore del sito di vendita di prodotti di ricambio online.</p>	Cliente Gestore
<b>Prodotto</b>	Pezzi di ricambio a disposizione per la vendita	-----
<b>Foto</b>	Immagine associata ai determinati prodotti in vendita sul sito.	-----
<b>Acquisto</b>	Rappresenta il prodotto che è stato acquistato dal cliente	-----
<b>Fattura</b>	Documento con le indicazioni della merce fornita, dell'ammontare dell'importo e delle relative condizioni di pagamento, che il venditore trasmette al compratore al momento della spedizione della merce ordinata.	-----

## 2.Progettazione concettuale

### 2.1 Schema E-R



## 3. Modello relazionale

### 3.1. Realizzazione base di dati in SQL

```
drop database if exists shop_autoricambi;  
create database shop_autoricambi;  
use shop_autoricambi;
```

Create table Utente

```
(  
  cf char(16) NOT NULL,  
  nome char(40) NOT NULL,  
  cognome char(40) NOT NULL,  
  data_nascita varchar(15) NOT NULL,  
  telefono char(10) NOT NULL,  
  cellulare char(10) NOT NULL,  
  email varchar(40) NOT NULL,  
  passw varchar(40) NOT NULL,  
  via varchar(40) NOT NULL,  
  cap char(5) NOT NULL,  
  citta varchar(40) NOT NULL,  
  amministratore boolean default 0,  
  unique(email),  
  primary key(cf)  
);
```

create table Fattura

```
(  
  codice_fattura int auto_increment,  
  dataF varchar(50),  
  costo_totale float,  
  pagamento varchar(30) NOT NULL default 'consegna',  
  cliente char(16) NOT NULL,  
  primary key(codice_fattura),  
  foreign key(cliente) references utente(cf) on delete no action on update no action  
);
```

Create table Prodotto

```
(  
  cod_pezzo varchar(20) NOT NULL,  
  marchio varchar(30) NOT NULL,  
  modello varchar(30) NOT NULL,  
  prezzo float(2) NOT NULL,  
  offerta boolean default 0,  
  prezzo_scontato float(2),  
  descrizione longtext NOT NULL,  
  quantita int NOT NULL default 0,
```

```
    primary key(cod_pezzo)
);
```

```
create table Acquisto
(
    codice_acquisto int auto_increment,
    descrizione longtext not null,
    quantita int not null,
    fattura int not null,
    prodotto varchar(20),
    primary key (codice_acquisto),
    foreign key (fattura) references Fattura(codice_fattura) on delete no action on update no
action,
    foreign key (prodotto) references Prodotto(cod_pezzo) on delete no action on update no
action
);
```

```
create table Foto
(
    num_foto int auto_increment NOT NULL,
    path longtext NOT NULL,
    prodotto varchar(20),
    primary key(num_foto),
    foreign key (prodotto) references Prodotto(cod_pezzo) on delete no action on update no
action
);
```