

Base di dati

Sistema per la compravendita di vino

Di Arturo Pesaro e Riccardo Galvani

- Introduzione: Specifiche
- Schema E/R
- Trasformazione da modello E/R a modello Relazionale
- Codice SQL per creazione tabelle
- Codice SQL per interrogazioni e inserimenti/cancellazioni
- Interfaccia

INTRODUZIONE: SPECIFICHE

Una piattaforma web che commercializza prodotti enogastronomici di consumo in cui ogni produttore di vini possa registrarsi e vendere i suoi prodotti on line. Ha con le seguenti caratteristiche:

1. Possibilità di visualizzare il catalogo prodotti, navigabile per categoria merceologica, marchio, ricerca libera, ecc. Possibilità di vedere il singolo prodotto con tutti i dettagli e le caratteristiche
2. Gestione di prodotti in push (per cui spingere la vendita) con inserimento in una “vetrina in home page” o in un area promo
4. Possibilità di visualizzare prodotti graditi al consumatore per analogia con lo storico prodotti visualizzati dall’utente
5. Gestione di una wish list (lista dei desideri)
6. Possibilità di inserire prodotti nel carrello e di effettuare l’acquisto di più prodotti in diverse quantità. Predisporre la gestione dei prezzi, del totale carrello, la gestione della disponibilità di magazzino, impedendo di poter acquistare quantità non disponibili
8. Possibilità di inserire l’indirizzo di consegna, di simulare il pagamento, e di confermare l’ordine
9. Possibilità di visualizzare lo stato dell’ordine e lo storico degli ordini effettuati
10. Gestione di coupon o buoni sconto in fase di acquisto, possibilità di visualizzare il tracking dell’ordine in consegna.
11. Possibilità degli amministratori di visualizzare gli utenti, verificare il numero degli ordini per utente, bloccare eventuali utenti.
12. Possibilità da parte del produttore di gestire il catalogo (inserimento, modifica). Possibilità di gestire la disponibilità di magazzino
13. Possibilità di visualizzare gli ordini effettuati e di marcarli come consegnati
14. Gestione del tracking dell’ordine, con simulazione di tutti i cambi di stato (consegnato al corriere, in viaggio, consegnato al destinatario, ecc.)
15. Gestione dei coupon o buoni sconto
16. Frontend multilingue
18. Durante a registrazione possibilità di selezionare il tipo di utente: utente compratore oppure utente venditore/produttore.

Il progetto scelto consiste nella realizzazione di un sistema che simuli quello per la compravendita di vino utilizzando una piattaforma ecommerce.

Il sistema è pensato principalmente per essere utilizzato sia da un utente compratore, sia da uno venditore, i quali loggandosi con le proprie credenziali possono usufruire dei servizi che il sistema mette a disposizione. I servizi che entrambi i tipi di utente hanno in comune sono:

- la modifica dei dati personali, con cui l'utilizzatore può cambiare i suoi dati aggiornando così il nome, cognome, password, via e città di residenza ed infine email. Il suo identificativo e tipo di utente non possono essere cambiati.
- la possibilità di contattare gli amministratori nel caso di problemi che riguardano la parte tecnica del sistema, quali crash, visualizzazione errata dei prodotti etc. Per ulteriori problemi, si specifica che gli amministratori del sistema fungono la funzione di *moderatori* (eliminare utenti) in quanto non hanno accesso diretto alla merce ma possono creare coupon utilizzabili per scontare i prodotti.

Prendendo l'esempio un utente *compratore*, i servizi messi a disposizione sono i seguenti:

- wishlist, ossia aggiungere/rimuovere prodotti in una lista dei desideri così da avere sempre a disposizione la merce di cui si è interessati.
- lista degli ordini, in cui può visualizzare in ogni momento lo stato, la data dell'ordine avvenuto e di consegna e l'identificativo dell'ordine stesso così da essere sempre aggiornato sull'evolversi delle sue transazioni.
- carrello, funzionalità che garantisce in tempo reale di aggiungere o togliere prodotti ad una lista che in un momento successivo potrà essere confermata così da passare alla parte del pagamento e spedizione.
- selezionando un prodotto, nella pagina che andrà a visualizzare, l'utente potrà trovare un riquadro (sulla destra) in cui vi compariranno i suoi prodotti più congeniali.

Un cliente *venditore* invece può usufruire di questi servizi:

- lista prodotti, in cui trova un elenco di tutta la merce da lui venduta visualizzando in particolare la quantità e il prezzo dei prodotti. Da questa pagina ha a disposizione altri comandi che gli permettono di modificare i dati di un determinato prodotto, eliminare un prodotto esaurito o non più in vendita, inserire dettagli nella lingua scelta della merce ed infine aggiungere nuovi prodotti alla propria lista.

Un cliente *amministratore* invece può usufruire di questi servizi:

- lista degli ordini, in cui può visualizzare in ogni momento lo stato, la data dell'ordine avvenuto e di consegna e l'identificativo dell'ordine stesso.
- modifica dello stato di un ordine (consegnato, in viaggio, consegnato al corriere...)
- creazione coupon, può immettere sconti personalizzati sulla merce acquistata dai singoli utenti compratori.
- eliminare utenti.

Le fasi principali dell'acquisto di merce sono illustrate più precisamente di seguito.

1. Dopo essersi loggato, il *l'acquirente* seleziona il prodotto che intende comprare. Nel caso la merce a cui è interessato non compaia nella pagina principale, può utilizzare il motore di ricerca che andrà a selezionare i vini in base al tipo scelto dall'utente; si può anche fare una ricerca più mirata scrivendo il nome, la regione o qualunque parola chiave che possa essere ricondotta al vino che si desidera visualizzare. Se l'utente ha già precedentemente aggiunto l'articolo alla sua lista dei desideri, lo può visualizzare cliccando l'apposito bottone.
2. Trovato il prodotto desiderato, se lo si vuole acquistare basta selezionare la quantità e aggiungere il tutto al carrello personale. In caso contrario, se si vuole comunque tenere in considerazione il prodotto per il futuro oppure se si vuole fare un paragone con altri prodotti dello stesso tipo, si può aggiungere questi alla wishlist.
3. Si procede con la compilazione dell'ordine, in cui si inseriscono via, città e CAP della destinazione della merce. Una volta compilato e confermato, si procede al pagamento vero e proprio.
4. Per il pagamento, si seleziona il tipo di carta di credito utilizzata, compilando successivamente i campi anagrafici richiesti(nome, cognome), inserendo il tipo di spedizione e confermando il tutto. Delle transazioni monetarie se ne occuperanno le banche dei relativi conti correnti. Una volta confermato il pagamento, l'utente venditore interessato spedisce nel più breve tempo possibile, seguendo le richieste del compratore, la merce e l'utente consumatore potrà informarsi sullo stato degli ordini dalla sua lista degli ordini che verrà aggiornata dagli amministratori in tempo reale.

SCHEMA E/R

ENTITA' E ASSOCIAZIONI

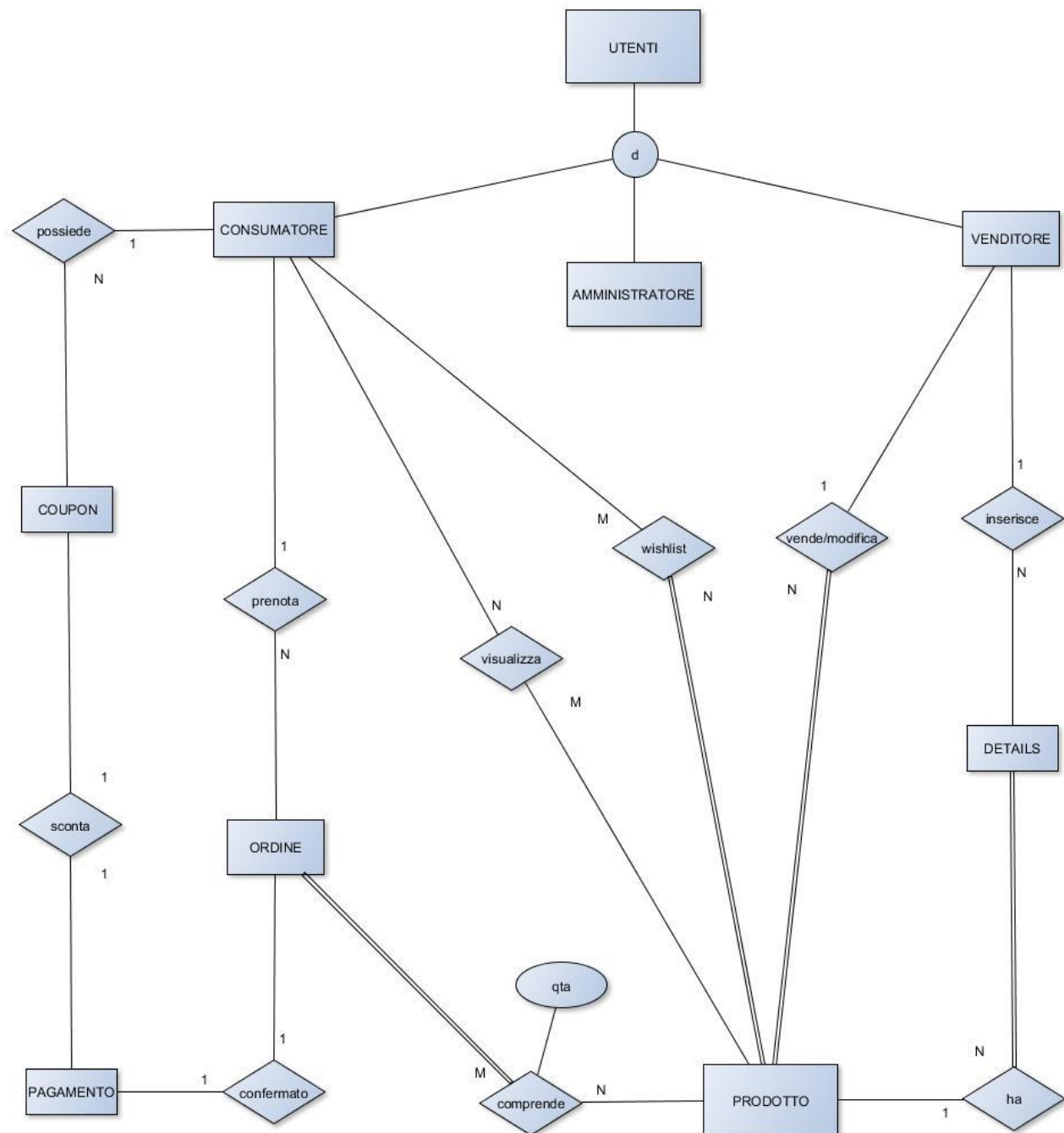


Figura 1.1: modello E/R

ENTITA'

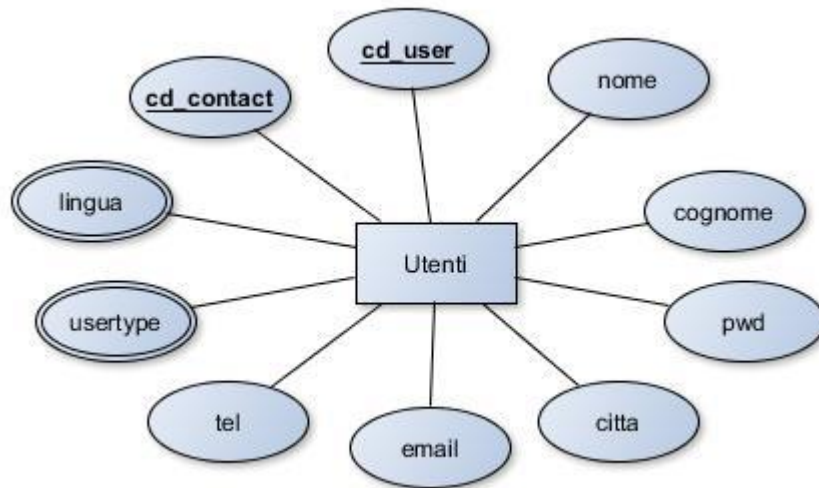


Figura 1.2: modello entità Utenti

UTENTI

Entità che rappresenta i dati personali di una persona.

Gli attributi che la caratterizzano sono:

- cd_contact (chiave primaria): ad ogni utente il sistema associa un numero che lo identifichi univocamente.
- cd_user (chiave primaria): l'username che l'utente sceglie di utilizzare per essere riconosciuto all'interno del sistema.
- nome
- cognome
- password: parola segreta utilizzata per il login
- città
- email
- tel: il telefono utilizzato (fisso o mobile)
- usertype: la specializzazione (compratore, venditore) in cui l'utente ha deciso di specializzarsi.
- lingua: ogni utente può selezionare la lingua in cui vorrà leggere le descrizioni dei prodotti

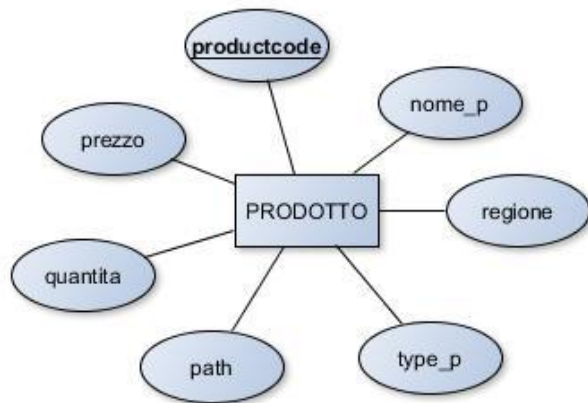


Figura 1.3: modello entità Prodotto

PRODOTTO

Entità che caratterizza la merce venduta.

Gli attributi che la caratterizzano sono:

- productcode(chiave primaria): il codice che identifica univocamente un determinato prodotto.
- nome_p: il nome della merce
- regione: la regione di origine(si può specificare anche il vigneto d'appartenenza)
- type_p: il tipo del prodotto(bianco, rosso, rosè o cibo).
- path: il percorso dell'immagine della merce
- quantità
- prezzo

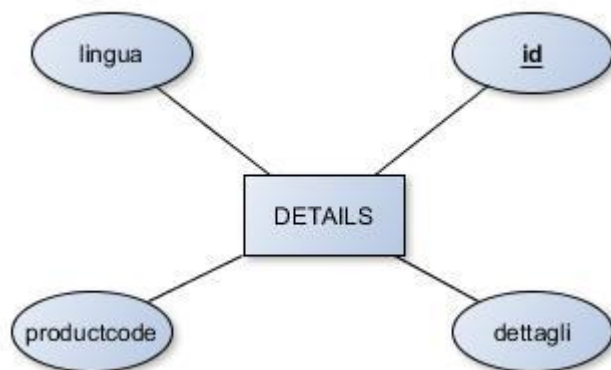


Figura 1.4: modello entità Details

DETAILS

Entità che descrive dettagliatamente e in varie lingue un prodotto.

Gli attributi che la caratterizzano sono:

- id (chiave primaria): identificativo di un determinato dettaglio
- productcode: l'identificativo del prodotto
- lingua: la lingua in cui sono scritte le informazioni sulla merce
- dettagli: la descrizione del prodotto

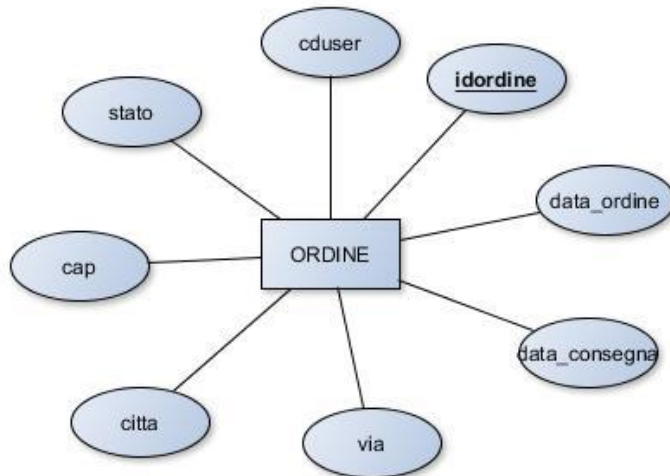


Figura 1.6: modello entità Ordine

ORDINE

Entità che rappresenta un ordine che un utente può eseguire.

Gli attributi che la caratterizzano sono:

- idordine(chiave primaria): identificativo dell'ordine
- data_ordine: la data di emissione dell'ordine
- data_consegna: data in cui la merce è stata consegnata
- stato: lo stato in cui si trova la merce (spedito/consegnato)
- cap: il codice di avviamento postale
- citta
- via
- cduser: l'identificativo del compratore

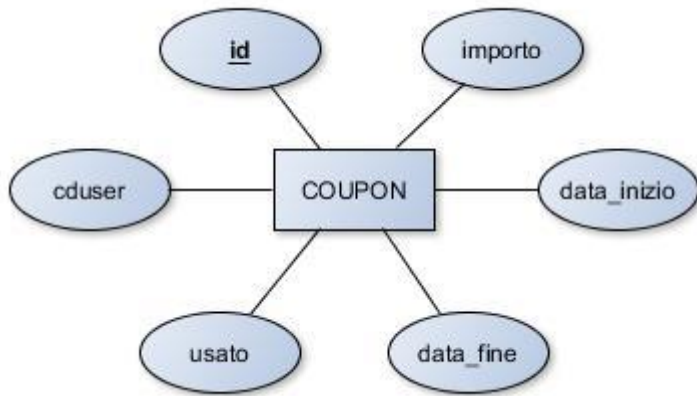


Figura 1.7: modello entità Coupon

COUPON

Entità che rappresenta gli sconti che un venditore può fornire ad un consumatore sulla merce da lui venduta.

Gli attributi che la caratterizzano sono:

- id(chiave primaria): identificativo del coupon
- importo: l'ammontare dello sconto
- data_inizio: da quando si può usufruire dello sconto
- data_fine: fino a quanto dura lo sconto erogato
- usato: se lo sconto è stato usato dal consumatore sarà cancellato
- cduser: identificativo del consumatore che può usufruire dello sconto

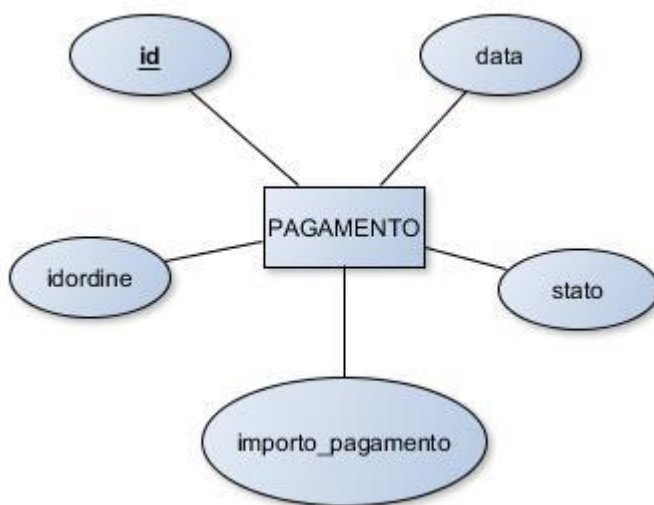


Figura 1.8: modello entità Pagamento

PAGAMENTO

Entità che rappresenta il pagamento effettuato dal compratore.

Gli attributi che la caratterizzano sono:

- id(chiave primaria): identificativo pagamento
- data: la data di immissione del pagamento
- stato: se il pagamento è stato effettuato correttamente o è ancora in fase di transizione
- importo_pagamento
- idordine: l'identificativo dell'ordine a cui è riferito il pagamento

ASSOCIAZIONI

comprende

Associazione che intercorre fra entità PRODOTTO e entità ORDINE. Specifica che ogni *prodotto* può comporre un'*ordine*. In questa associazione è presente un attributo : qta (quantità) ,che mi indica la quantità di ogni prodotto che andrà a comporre l'*ordine*. ORDINE ha un vincolo di partecipazione totale poiché non possono esistere ordini senza prodotti. Cardinalità: N-M. Ogni PRODOTTO può essere presente in M *ORDINI* e ogni ORDINE può essere composto da N PRODOTTI.

visualizza

Associazione che intercorre fra entità CONSUMATORE ed entità PRODOTTO. Specifica che un utente può visualizzare uno o più prodotti e questa visualizzazione deve essere salvata in un record così da poter conoscere le preferenze dell'utente. Cardinalità:N-M . Ogni CONSUMATORE può visualizzare M prodotti , viceversa ogni PRODOTTO può essere visualizzato da N *CONSUMATORI*.

wishlist

Associazione che intercorre fra entità CONSUMATORE ed entità PRODOTTO. Specifica che una *wishlist* è composta da dei prodotti ed è posseduta da un *utente*. Wishlist ha un vincolo di partecipazione totale poiché non può esistere una wishlist senza dei prodotti che la compongono. Cardinalità: N-M . Ogni *WISHLIST* è composta da N PRODOTTI , mentre ogni PRODOTTO può comporre più *WISHLIST*.

vende/modifica

Associazione tra VENDITORE e PRODOTTO. Specifica che un utente venditore può gestire la propria merce vendendola e modificandone i contenuti. Il vincolo di partecipazione totale specifica che un PRODOTTO non esiste se un VENDITORE non lo possiede. La cardinalità è del tipo 1-N poichè un VENDITORE *vende* più prodotti, ma all'interno del sistema non ci possono essere più venditori che propongono la stessa merce.

inserisce

Legame tra le entità VENDITORE e DETAILS; il venditore inserisce informazioni sul PRODOTTO in diverse lingue. Cardinalità 1-N dato che un VENDITORE può *inserire* molteplici descrizioni, e un DETAILS non può essere inserito da più utenti.

ha

DETAILS è legato all'entità PRODOTTO e riporta le informazioni caratteristiche e dell'uso delle merci. La cardinalità è del tipo 1-N, infatti un prodotto possiede più DETAILS (nelle varie lingue inserite dal VENDITORE) ma i dettagli sono propri e unici di un preciso PRODOTTO.

sconta

Lega un COUPON ad un PAGAMENTO, stabilisce quindi di quanto la merce andrà scontata. La cardinalità è 1-1 in quanto un COUPON può essere usato in un solo *pagamento*, e un PAGAMENTO può essere scontato di un solo COUPON.

prenota

Associazione che lega entità CONSUMATORE all'entità ORDINE. La cardinalità è 1-N poichè un CONSUMATORE può *prenotare* più ordini, ma un ORDINE è univocamente legato ad un compratore.

confermato

Legame tra PAGAMENTO e ORDINE, associa un determinato ORDINE al suo relativo PAGAMENTO. Cardinalità 1-1, un PAGAMENTO fa riferimento ad un solo ORDINE, e viceversa su un ORDINE si può effettuare un solo PAGAMENTO.

possiede

Associazione tra CONSUMATORE e COUPON. Cardinalità 1-N, un CONSUMATORE può possedere più COUPON, ma non viceversa.

TRASFORMAZIONE DA MODELLO E/R A MODELLO RELAZIONALE

Il modello relazionale è un modello logico di rappresentazione dei dati, per passare dal modello E-R al modello relazionale occorre seguire i seguenti passi:

1. traduzione di tipi di entità
2. traduzione di tipi di associazioni binarie 1:1
3. traduzione di tipi di associazioni binarie 1:N
4. traduzione di tipi di associazioni binarie M:N

Traduzione dei tipi di entità

prodotto							
<u>productcode</u>	nome_p	regione	type_p	quantita	prezzo	path	cduser

utenti										
<u>cd_contact</u>	<u>cduser</u>	pwd	nome	cognome	email	citta	via	tel	usertype	lingua

ordine							
<u>idordine</u>	cduser	via	citta	cap	stato	data_consegna	data_ordine

coupon					
<u>id</u>	cduser	importo	data_inizio	data_fine	usato

details			
<u>id</u>	lingua	dettagli	productcode

pagamento				
<u>id</u>	data	idordine	importo_pagamento	stato

Figura 2.1: modello relazionale, traduzione tipi di entità

Traduzione di tipi di associazioni binarie 1:1

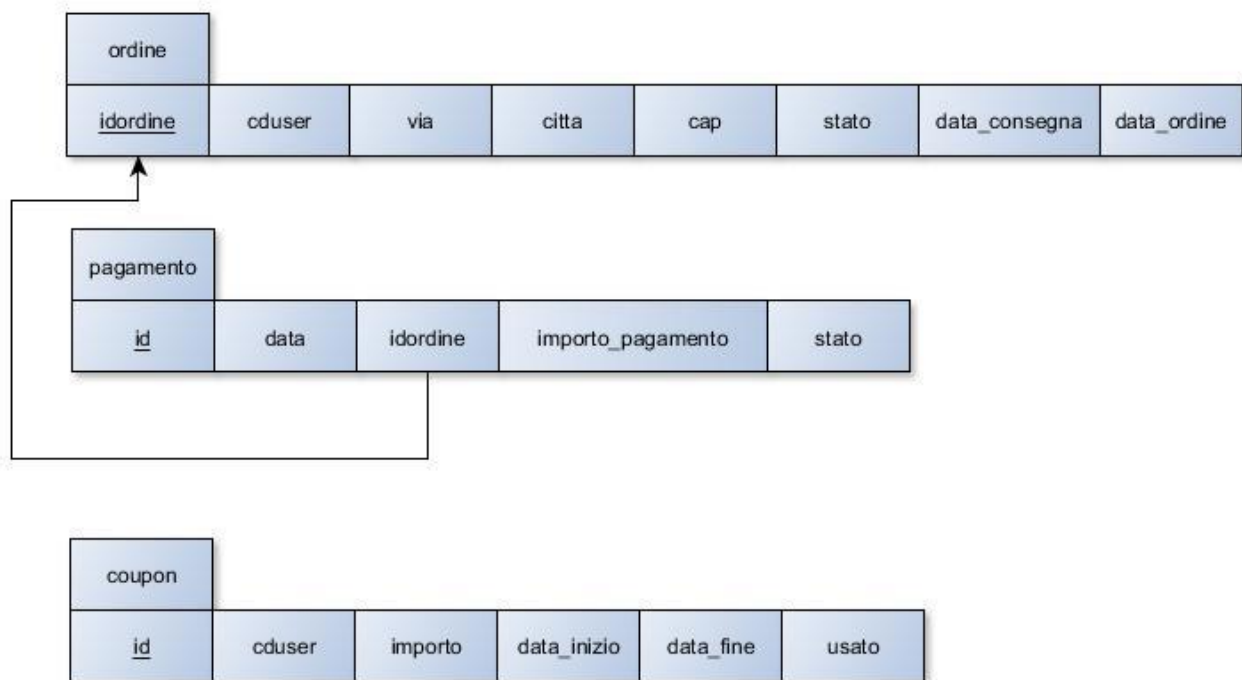


Figura 2.2: modello relazionale traduzione relazioni 1:1

Traduzione di tipi di associazioni binarie 1:N

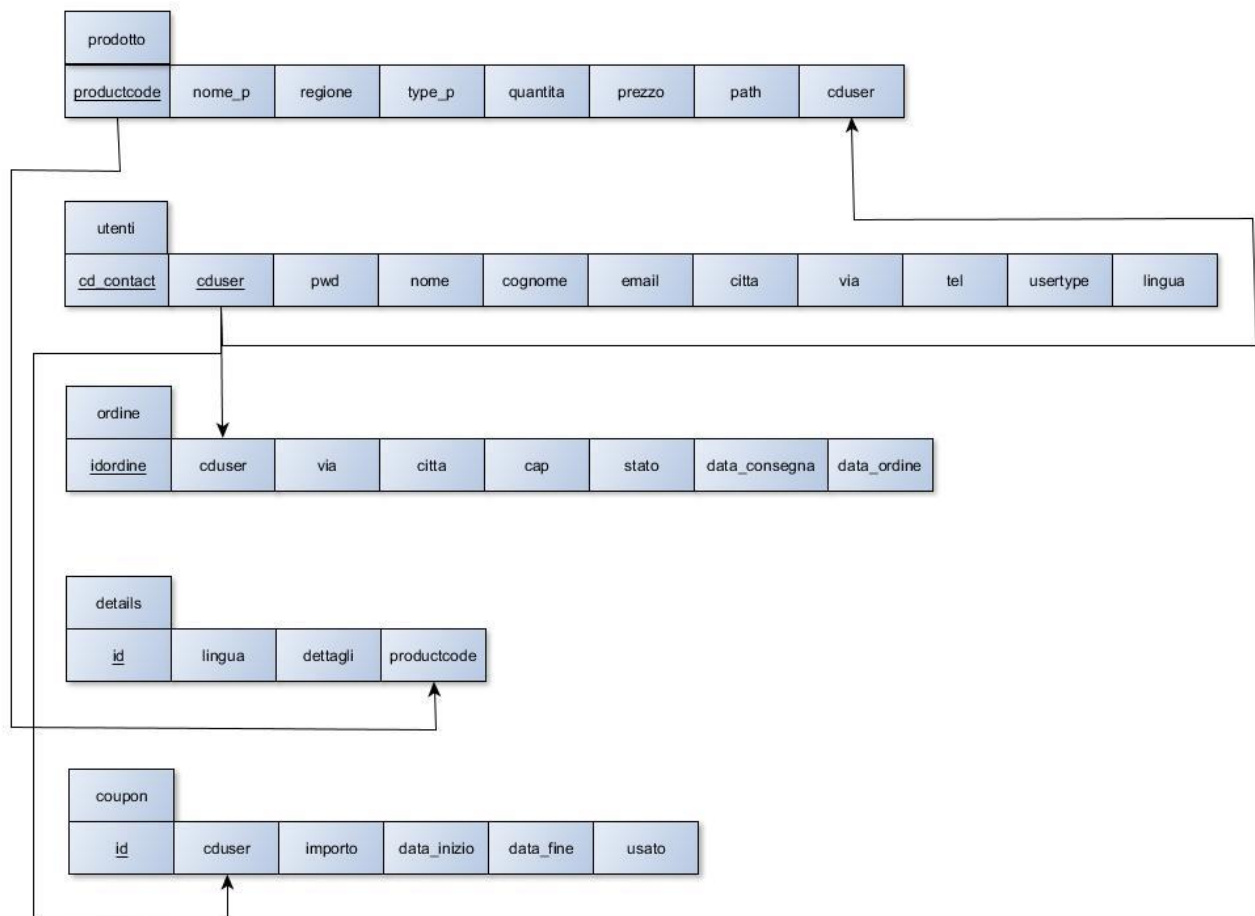


Figura 2.3: modello relazionale traduzione relazioni 1:N

Traduzione di tipi di associazioni binarie N:M



Figura 2.2: modello relazionale traduzione relazioni N:M

Considerazioni su forme normali:

Lo schema relazionale risulta essere in 1NF (prima forma normale) in quanto è stato tradotto dal Diagramma ER utilizzando le apposite regole. Le chiavi primarie delle relazioni sono dei contatori AUTO_INCREMENT.

Schema relazionale completo:

prodotto							
<u>productcode</u>	nome_p	regione	type_p	quantita	prezzo	path	cduser

utenti										
<u>cd_contact</u>	<u>cduser</u>	pwd	nome	cognome	email	citta	via	tel	usertype	lingua

wishlist		
<u>cduser</u>	<u>productcode</u>	

ordine							
<u>idordine</u>	cduser	via	citta	cap	stato	data_consegna	data_ordine

coupon					
<u>id</u>	cduser	importo	data_inizio	data_fine	usato

details			
<u>id</u>	lingua	dettagli	productcode

pagamento				
<u>id</u>	data	idordine	importo_pagamento	stato

visualizza		
<u>cduser</u>	<u>productcode</u>	

comprende		
<u>productcode</u>	<u>idordine</u>	qta

CODICE SQL DI CREAZIONE DELLE TABELLE

In seguito viene riportato il codice sql esportato da phpMyAdmin. Comprende i costrutti sql che permettono di creare le tabelle prodotto, utenti, wishlist, ordine, coupon, details, pagamento, visualizza e comprende. Per l'implementazione del modello E-R si può utilizzare MySql Workbench.

```
--  
-- Database: `cantina`  
--  
-----  
  
--  
-- Struttura della tabella `coupon`  
--  
  
CREATE TABLE IF NOT EXISTS `coupon` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `IMPORTO` double NOT NULL,  
  `DATA_INIZIO` date NOT NULL,  
  `DATA_FINE` date NOT NULL,  
  `USATO` tinyint(1) DEFAULT '0',  
  `CDUSER` varchar(20) NOT NULL,  
  PRIMARY KEY (`ID`),  
  KEY `iutente_fk_idx` (`CDUSER`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=8 ;
```

```
--  
-- Struttura della tabella `details`  
--  
  
CREATE TABLE IF NOT EXISTS `details` (  
  `productCode` int(11) NOT NULL,  
  `lingua` varchar(45) NOT NULL,  
  `dettagli` varchar(5000) NOT NULL,  
  `id` int(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  FULLTEXT KEY `lingua` (`lingua`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--  
-- Struttura della tabella `ordine`  
--  
  
CREATE TABLE IF NOT EXISTS `ordine` (  
  `idordine` int(11) NOT NULL AUTO_INCREMENT,  
  `data_ordine` date NOT NULL,  
  `via` varchar(45) NOT NULL,  
  `citta` varchar(55) NOT NULL,  
  `cap` varchar(10) NOT NULL,  
  `data_consegna` date NOT NULL,
```

```
`stato` varchar(60) NOT NULL DEFAULT 'in ritiro',
`cduser` varchar(50) NOT NULL,
PRIMARY KEY (`idordine`),
KEY `ordiniutente_fk_idx` (`cduser`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=66 ;
```

```
--
-- Struttura della tabella `pagamento`
--
```

```
CREATE TABLE IF NOT EXISTS `pagamento` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`data` date NOT NULL,
`stato` varchar(45) NOT NULL,
`importo_pagamento` double NOT NULL,
`idordine` int(11) NOT NULL,
PRIMARY KEY (`id`),
KEY `pagordine_fk_idx` (`idordine`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=32 ;
```

```
-----
```

```
--
-- Struttura della tabella `prodotti`
--
```

```
CREATE TABLE IF NOT EXISTS `prodotti` (
`PRODUCTCODE` int(11) NOT NULL DEFAULT '0',
`NOME_P` varchar(100) DEFAULT NULL,
`REGIONE` varchar(100) DEFAULT NULL,
`TYPE_P` varchar(100) DEFAULT NULL,
`QUANTITA` varchar(200) DEFAULT NULL,
`PREZZO` varchar(200) DEFAULT NULL,
`PATH` varchar(100) DEFAULT NULL,
`CDUSER` varchar(50) NOT NULL,
PRIMARY KEY (`PRODUCTCODE`),
KEY `acontatti_fk_cd_user` (`NOME_P`),
FULLTEXT KEY `CDUSER` (`CDUSER`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-----
```

```
--
-- Struttura della tabella `utenti`
--
```

```
CREATE TABLE IF NOT EXISTS `utenti` (
`CD_CONTACT` int(11) NOT NULL DEFAULT '0',
`CD_USER` varchar(100) DEFAULT NULL,
`PWD` varchar(100) NOT NULL,
`NOME` varchar(100) DEFAULT NULL,
`COGNOME` varchar(100) DEFAULT NULL,
`EMAIL` varchar(200) DEFAULT NULL,
```

```

`VIA` varchar(200) DEFAULT NULL,
`CITTA` varchar(100) DEFAULT NULL,
`TEL` varchar(50) DEFAULT NULL,
`USERTYPE` varchar(1) DEFAULT NULL,
`LINGUA` varchar(30) NOT NULL,
PRIMARY KEY (`CD_CONTACT`),
KEY `acontatti_fk_cd_user` (`CD_USER`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

--
-- Struttura della tabella `visualizza`
--

```

```

CREATE TABLE IF NOT EXISTS `visualizza` (
`cduser` varchar(20) NOT NULL,
`productcode` int(11) NOT NULL,
PRIMARY KEY (`productcode`,`cduser`),
KEY `utente_fk_idx` (`cduser`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

--
-- Struttura della tabella `wishlist`
--

```

```

CREATE TABLE IF NOT EXISTS `wishlist` (
`cduser` varchar(50) NOT NULL,
`productcode` int(11) NOT NULL,
PRIMARY KEY (`productcode`,`cduser`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

--
-- Struttura della tabella `comprende`
--

```

```

CREATE TABLE IF NOT EXISTS `comprende` (
`productcode` int(11) NOT NULL,
`idordine` int(11) NOT NULL,
`qta` int(11) NOT NULL,
PRIMARY KEY (`productcode`,`idordine`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

CODICE SQL PER INTERROGAZIONI E INSERIMENTI/CANCELLAZIONI

Query per la visualizzazione della vetrina principale della home page. Prendo lista prodotti random dall'database.

```
sql=" SELECT * FROM prodotto order by rand()";
```

Prendo lista dei prodotti "nuovi" ossia appena inseriti dai *venditori*. Il risultato lo metto in ordine decrescente così da selezionare gli ultimi vini aggiunti.

```
sql=" SELECT * FROM prodotto ORDER BY productcode DESC";
```

Ricerca dei prodotti partendo dal nome, dettaglio, regione del vino inserito. Il risultato lo raggruppo per nome prodotto.

```
sql="SELECT * FROM prodotto AS P JOIN details AS D ON P.productcode = D.productcode WHERE  
(P.nome_p LIKE '%" + search + "%' or D.dettagli LIKE '%" + search + "%' or P.regione LIKE '%" + search + "%')  
GROUP BY nome_p";
```

Metodo per ricavare il numero degli ordini effettuati da un determinato utente

```
sql = "SELECT COUNT(*) as nordini FROM ordine WHERE  
cduser='" + Conversion.getDatabaseString(cduser) + "'";
```

Metodo per ricavare tutti gli ordini

```
sql = "SELECT * FROM ordine ORDER BY data_ordine DESC";
```

Metodo per cambiare lo stato di un ordine

```
sql = "UPDATE ordine SET stato='" + Conversion.getDatabaseString(stato) WHERE idordine='" + idordine;
```

Ricevere la wishlist dell'utente loggato.

```
sql=" SELECT * FROM wishlist " WHERE cduser='" + Conversion.getDatabaseString(cduser) + "'";
```

Metodo per ottenere una lista dei coupon attivi dell'utente

```
sql = " SELECT * FROM coupon WHERE usato=0 AND cduser='" + Conversion.getDatabaseString(cduser)+"' ";
```

Metodo che restituisce i prodotti visualizzati dall'utente

```
sql = "SELECT * FROM visualizza WHERE cduser='"+Conversion.getDatabaseString(cduser)+"' ORDER BY  
RAND() LIMIT 1";
```

INTERFACCIA

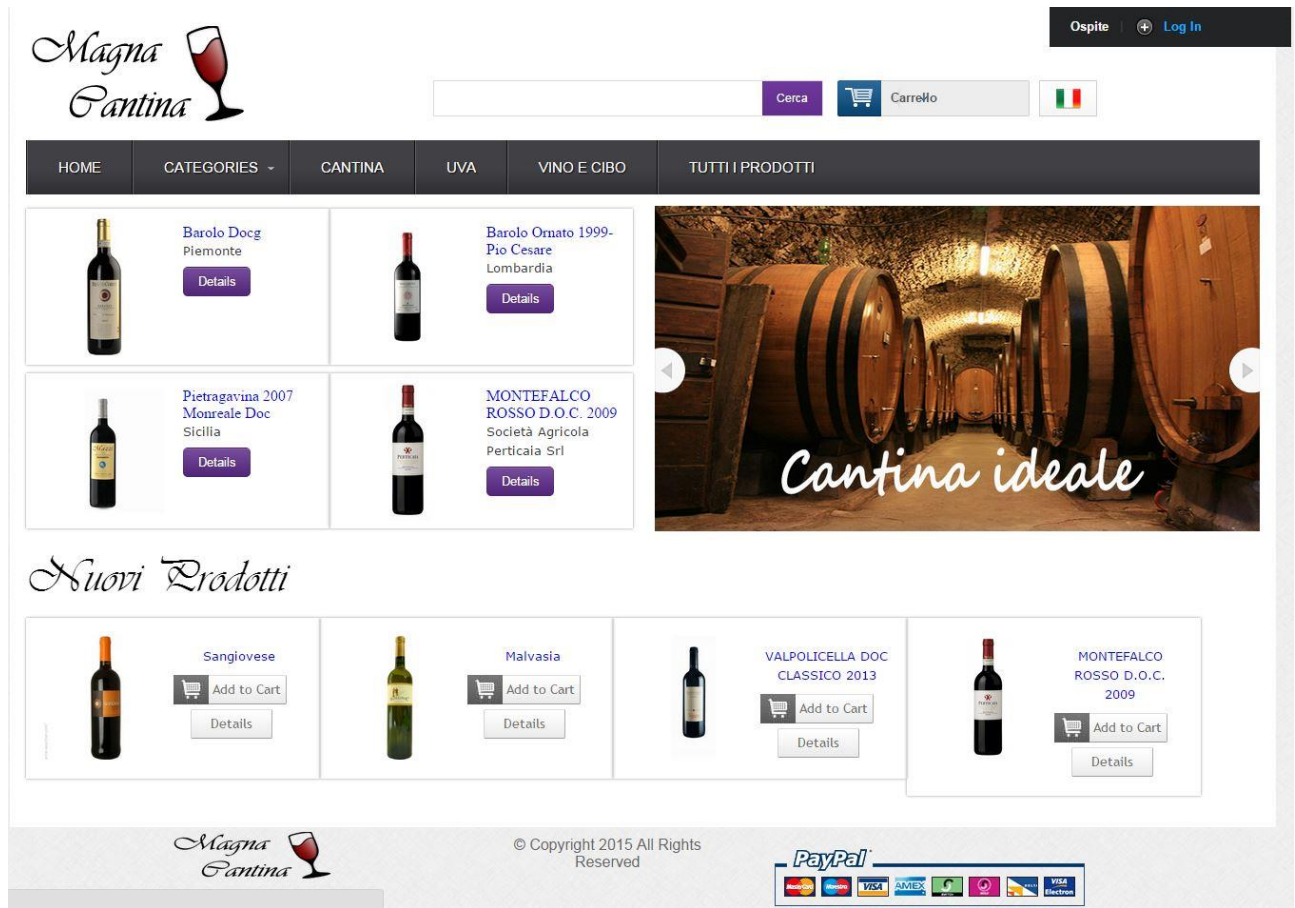


Figura 3.1: Homepage dell'interfaccia web al sistema

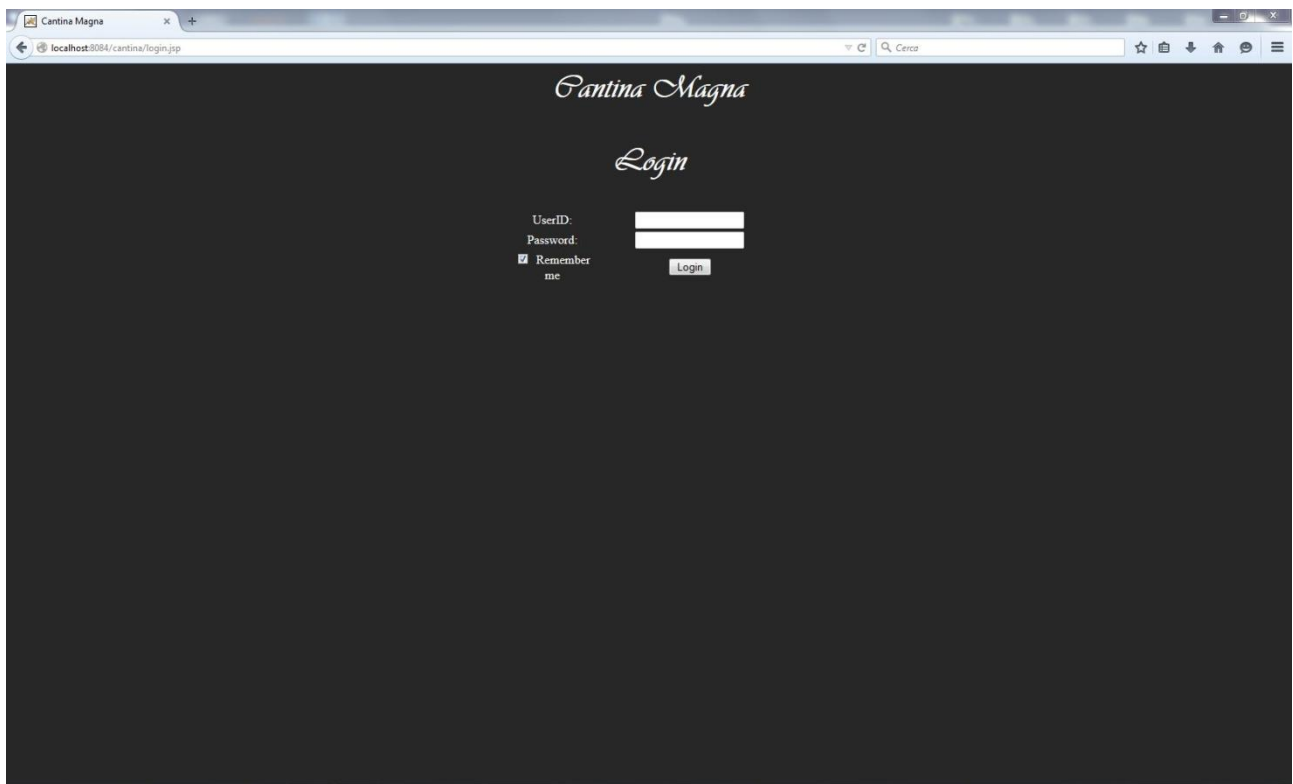


Figura 3.2: Pagina di login utenti

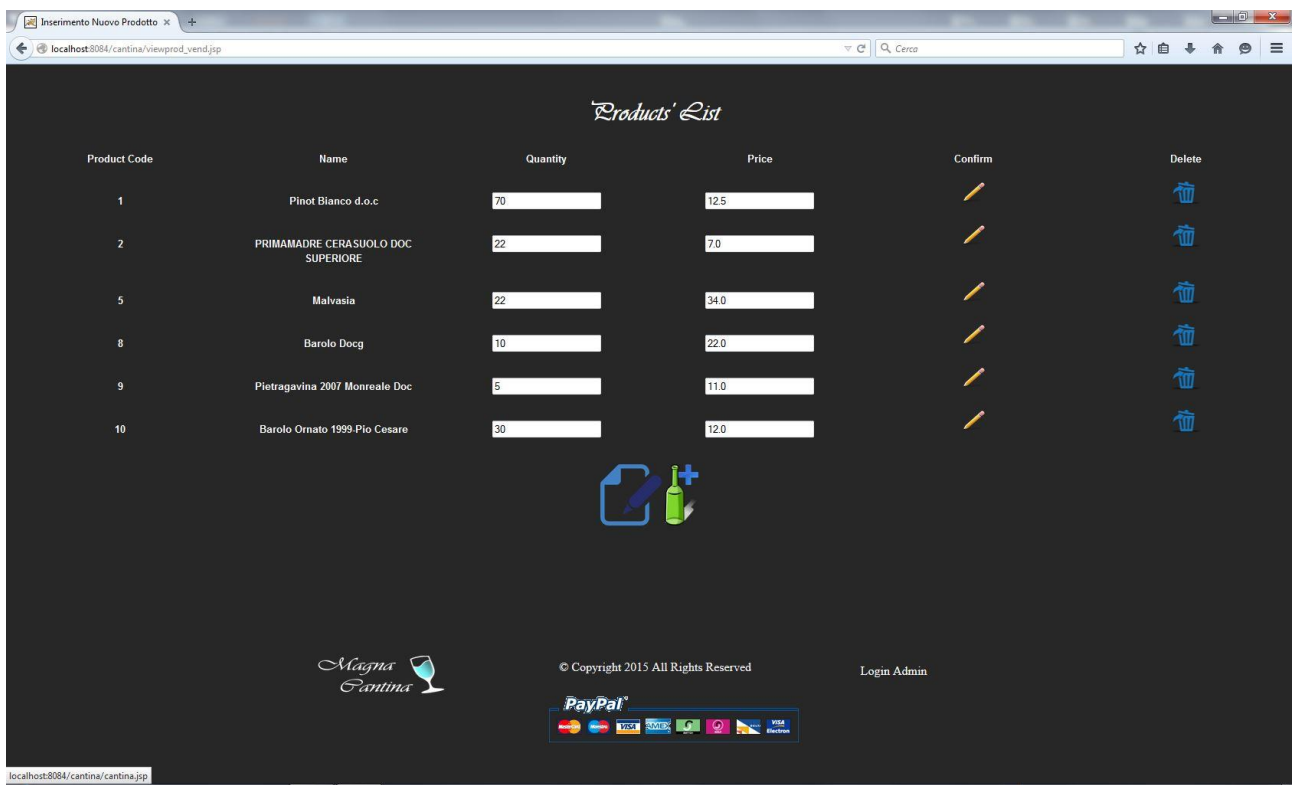


Figura 3.3: Products' list dell'utente venditore

Modifica i tuoi dati

User ID (non modificabile)

Nome

Cognome

Password

Indirizzo

Città

Telefono

E-Mail

Tipo Utente

Figura 3.4: Pagina di modifica dei dati personali

Magna Cantina

Emperor | [Modify](#) | [Order List](#) | [Coupon](#) | [Users' List](#) | [Logout](#)

HOME | CATEGORIES | BODEGA | UVAS | VINO Y COMIDA | TODOS LOS PRODUCTOS

Order ID

Change status ☒ In ritiro ☐ Consegnato ☐ Consegnato al corriere ☐ In viaggio

ID Ordine	Data Ordine	Data Consegna	Status	User ID
82	12/05/2015	15/05/2015	in ritiro	t
81	12/05/2015	15/05/2015	in ritiro	t
80	12/05/2015	15/05/2015	in ritiro	t
79	12/05/2015	15/05/2015	in ritiro	t
78	12/05/2015	15/05/2015	in ritiro	t
77	12/05/2015	15/05/2015	in ritiro	t
76	12/05/2015	15/05/2015	in ritiro	t
75	12/05/2015	15/05/2015	In viaggio	t
74	12/05/2015	15/05/2015	in ritiro	t
73	12/05/2015	15/05/2015	in ritiro	t
72	12/05/2015	15/05/2015	in ritiro	t
69	11/05/2015	14/05/2015	Consegnato	t
68	03/05/2015	06/05/2015	In viaggio	t
67	01/05/2015	04/05/2015	in ritiro	t
66	01/05/2015	04/05/2015	in ritiro	t
65	01/05/2015	04/05/2015	in ritiro	albe

Figura 3.5: Pagina di modifica degli ordini da parte di un amministratore.

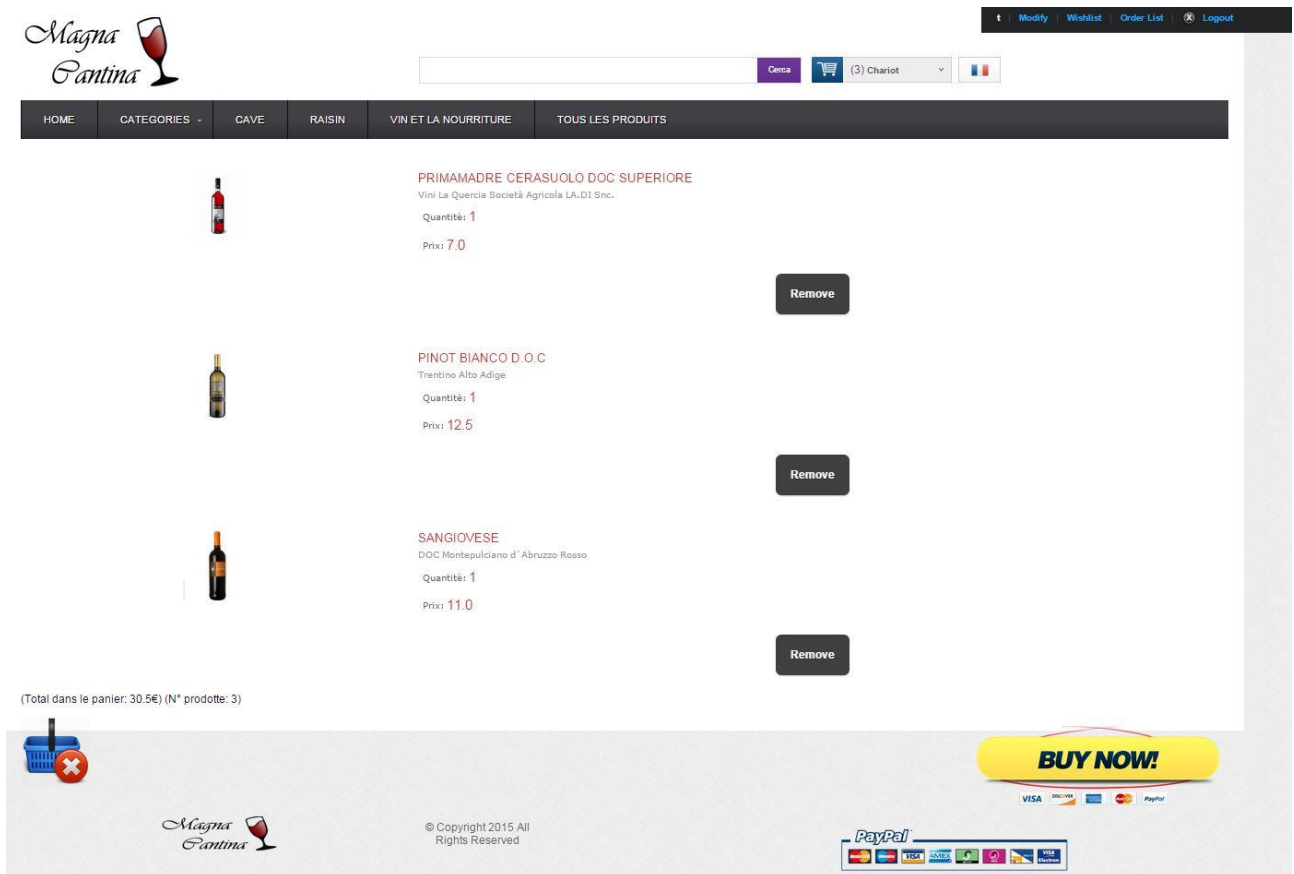


Figura 3.6: Vista carrello da parte di un compratore.