

Documento di progetto

Gestione di un E-COMMERCE



Lo Bosco Alessandro

0739922

Miceli Gaspare

0732495

13 Febbraio 2023

1 Analisi dei requisiti

Introduzione:

Creazione di database per la gestione di un sito e-commerce.

Si possono registrare due tipi di utenti:

- **Venditori:** Proprietari di negozi fisici e non, che vogliono vendere online.
- **Acquirenti:** I compratori dei prodotti.

Ogni articolo è composto da una sola categoria, il prezzo, il negozio online che vende l'articolo, il magazzino fisico in cui si trova e le recensioni.

Per il prezzo può essere applicato uno sconto.

Del negozio oltre al nome si ha il recapito telefonico.

Le recensioni hanno un numero di stelle con un possibile commento.

L'acquirente ha il proprio indirizzo di consegna e può possedere più carte di credito.

L'acquirente possiede un solo indirizzo di consegna che potrà essere cambiato da ordine ad ordine.

1.1 Dettagli

Gli articoli sono caratterizzati da nome, quantità, prezzo, sconto, categoria e recensioni che sono composte da stella e commento.

Il negozio è composto da nome e recapito telefonico.

Per ogni magazzino ci sono immagazzinati più articoli diversi.

L'acquirente ha nome, cognome, genere, email, password, data di nascita e può possedere più carte di credito.

Le carte hanno il numero, il nome del titolare, il cvv, e la data di scadenza.

Un acquirente non può acquistare senza avere una carta registrata.

Un acquirente non può recensire un prodotto senza averlo acquistato.

Un acquirente può solo mettere una recensione a prodotto.

Un acquirente può possedere un solo indirizzo di consegna che casomai verrà cambiato.

2 Progettazione concettuale

Le operazioni che caratterizzano questa fase sono:

- Individuazione delle entità e degli attributi.
- Individuazione delle chiavi primarie.
- Individuazione delle associazioni.
- Individuazione degli eventuali attributi delle associazioni.
- Stesura del diagramma entità-relazione (E-R).
- Stesura del diagramma logico.

2.1 Individuazione delle entità

Dall'analisi dei requisiti abbiamo individuato le seguenti entità:

- **Negozio**
- **Prezzo**
- **Categoria**
- **Articolo**
- **Recensione**
- **Magazzino**
- **Indirizzo**
- **Acquirente**
- **Carta**
- **Città**
- **Regione**
- **Nazione**

2.2 Individuazione degli attributi delle entità

Per ogni entità abbiamo individuato i seguenti attributi

(primary key sottolineate e foreign key* con asterisco)

- **Prezzo:** id_prezzo, Cod_articolo*, Prezzo, Sconto.
- **Negozio:** Cod_negozio, Nome, Rec_telefonico.
- **Categoria:** id_categoria, Nome.
- **Articolo:** Cod_articolo, Cod_negozio*, Cod_magazzino*, id_categoria*, Nome, Descrizione, Quantità.
- **Recensioni:** Cod_recensioni, Cod_articolo*, id_acquirente*, Stelle, Commento.
- **Magazzino:** Cod_magazzino, id_indirizzo*.
- **Acquisto:** id_acquirente*, Cod_articolo*, Data, Acquistato, Quantità.
- **Acquirente:** id_acquirente, id_indirizzo_cons*, Nome, Cognome, E-mail, Genere, Password, Data_nascita.
- **Carta:** Num_carta, id_acquirente*, Titolare, Cvv, Scadenza_mese, Scadenza_anno.
- **Indirizzo:** id_indirizzo, id_città*, Nome, Num_civico.
- **Città:** id_città, id_regione*, Nome.
- **Regione:** id_regione, id_nazione*, nome.
- **Nazionalità:** id_nazione, nome.

2.3 Individuazione delle associazioni

- **Vendere:** Negozio - Articolo (N:N)
- **Costare:** Prezzo – Articolo (1:1)
- **Avere:** Categoria – Articolo (1:N)
- **Contenere:** Magazzino – Articolo (1:N)
- **Comprare:** Acquirente – Articolo (N:N)
- **Scrivere:** Acquirente – Recensione (N:1)
- **Possedere:** Articolo – Recensione (1:N)
- **Collocare:** Magazzino – Indirizzo (1:1)
- **Collocare:** Acquirente – Indirizzo (1:1)
- **Possedere:** Acquirente – Carta (1:N)
- **Comprendere:** Indirizzo – Città (N:1)
- **Comprendere:** Città – Regione (N:1)
- **Comprendere:** Regione – Nazione (N:1)

Comprare (Acquisto): Id_acquirente*, Cod_articolo*, Data, Acquistato, Quantità.

Schema Entità-Relazione

```

    erDiagram
        NEGOZIO }|--} ARTICOLO : VENDE
        PREZZO }|--} ARTICOLO : COSTARE
        CATEGORIA }|--} ARTICOLO : AVERE
        ARTICOLO }|--} MAGAZZINO : CONTIENE
        ARTICOLO }|--} INDIRIZZO : COLLOCARE
        ARTICOLO }|--} CITTÀ : COLLOCARE
        ARTICOLO }|--} REGIONE : COMPRENDERE
        ARTICOLO }|--} NAZIONE : COMPRENDERE
        ARTICOLO }|--} ACQUIRENTE : COMPRARE
        ARTICOLO }|--} CARTA : POSSEDERE
        ARTICOLO }|--} RECENSIONE : POSSEDERE
        MAGAZZINO }|--} INDIRIZZO : COLLOCARE
        INDIRIZZO }|--} CITTÀ : COMPRENDERE
        ACQUIRENTE }|--} CARTA : POSSEDERE
        ACQUIRENTE }|--} RECENSIONE : SCRIVERE
    
```

Il database sviluppato si baserà sul modello relazionale dei dati, perciò si parlerà di “tabelle”.

- Definizione del volume di ogni tabella
- Previsione della frequenza delle operazioni di ogni tipo
- Analisi delle ridondanze
- Rimozione delle generalizzazioni
- Stesura dello schema logico
- Vincoli di integrità non esprimibili

3.1 Definizione del volume di ogni tabella (Stima dopo 3 anni)

(E=Entità, R=Relazione)

Concetti	Tipo	Volume
Prezzo	E	7.200.000
Negozio	E	7.200
Categoria	E	1.000
Articolo	E	7.200.000
Recensioni	E	36.000.000
Magazzino	E	7.200
Acquirente	E	18.000
Carta	E	4.500
Indirizzo	E	25.200
Città	E	2.520
Regione	E	252
Nazione	E	25
Vendere (Negozio, Articolo)	R	2.268
Comprare (Acquirente, Articolo)	R	216.000

Calcolo dei volumi:

- **Entità:**

“Si registrano 500 acquirenti ogni mese e 20 nuovi acquisti”

- $\text{Acquirenti} = 500 \times 12 \times 3 = 18.000$

“Per ogni acquirente viene registrata in media una carta per 4 mesi”

- $\text{Carte} = \text{Acquirenti} / 4 = 4.500$

“Ci sono 200 nuovi negozi ogni mese e in ogni negozio vengono importati almeno 1000 articoli al mese”

- $\text{Negozi} = 200 \times 12 \times 3 = 7.200$
- $\text{Articoli} = 1.000 \times 7.200 = 7.200.000$
- $\text{Prezzi} = \text{Articoli}$
- $\text{Categorie} = 1.000$

“Per ogni articolo vengono scritte (in media) 5 recensioni al mese”

- $\text{Recensioni} = 7.200.000 \times 5 = 36.000.000$

“In ogni magazzino ci sono 1000 articoli diversi”

- $\text{Magazzini} = \text{Articoli} / 1000 = 7.200$

“Ogni mese vengono raccolti 700 nuovi indirizzi da diverse città”

- $\text{Indirizzi} = 700 \times 12 \times 3 = 25.200$
- $\text{Città} = \text{Indirizzi} / 10 = 2.520$
- $\text{Regioni} = \text{Città} / 10 = 252$
- $\text{Nazioni} = 25$

- **Relazioni:**

“Ogni negozio mette a disposizione almeno 63 articoli al mese”

- $\text{Vendere (Articoli messi a disposizione dal negozio)} = 63 \times 12 \times 3 = 2.268$

“Vengono comprati circa 6.000 articoli al mese”

- $\text{Comprare} = 6.000 \times 12 \times 3 = 216.000$

3.2 Previsione della frequenza delle operazioni di ogni tipo

Quelle con l'asterisco sono degli inserimenti o delle cancellazioni che coinvolgono una sola tabella e hanno un costo trascurabile.

Operazioni:

1. Inserimento Nazioni, Regioni, Città* (Sono preimpostati)
2. Inserimento Negozi, Articoli, Prezzi, Categorie, Magazzini* (Sono preimpostati)
3. Registrazione Acquirente
4. Inserimento carta di acquirente
5. Inserimento Articolo nel carrello*
6. Acquisto Articolo (update)
7. Inserimento recensione*
8. Cancellazione Acquirente*
9. Visualizzazione di articoli disponibili
10. Visualizzazione recensioni di articoli con prezzo compreso tra (€5 e €30)
11. Visualizzazione media stelle di ogni articolo(vista)
12. Accesso agli articoli più graditi da 4 stelle in su (media stelle)
13. Visualizzazione degli articoli per categoria (vista)
14. Visualizzazione costo totale del carrello di un acquirente
15. Controllo di possesso carta per acquirente
16. Per ogni negozio visualizzazione degli articoli con prezzo effettivo
17. Visualizzazione ordini effettuati dall'acquirente

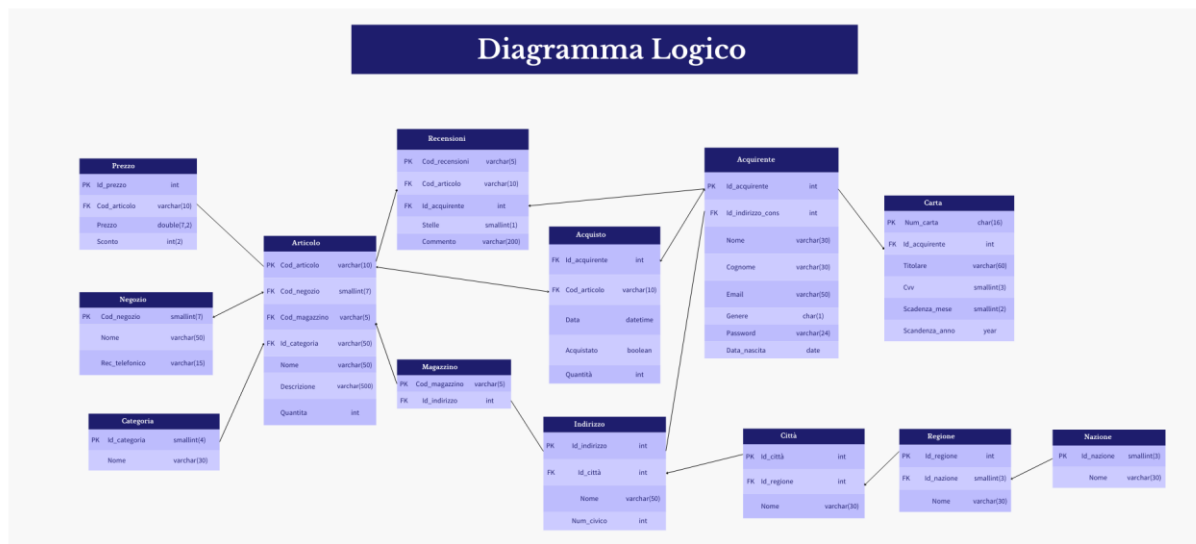
3.3 Analisi delle ridondanze

A seguito della nostra analisi non abbiamo trovato informazioni ridondanti.

3.4 Rimozione delle generalizzazioni

- Per gestire gli admin abbiamo previsto un utente speciale chiamato Nome: admin
Cognome: admin nella tabella acquirente.

3.5 Stesura del diagramma logico



3.6 Vincoli di integrità non esprimibili

Abbiamo individuato questi vincoli non esprimibili (che verranno risolti in fase di progettazione della parte php dell'e-commerce).

- Un acquirente non può scrivere una recensione per un articolo non comprato.
- Un acquirente può scrivere al più una recensione per lo stesso articolo.
- Un acquirente non può acquistare l'articolo senza aver inserito i dati della propria carta.

Abbiamo individuato vincoli non esprimibili gestibili da trigger:

- Modifica della quantità di un articolo in vendita nel momento dell'acquisto.

4 Implementazione delle tabelle, query e trigger

Il DBMS utilizzato per creare e gestire il database è MySQL di phpmyadmin.

4.1 Creazione delle tabelle

```
/*TABELLE*/
```

```
CREATE TABLE Negozio (  
    Cod_negozio smallint(7) NOT NULL AUTO_INCREMENT,  
    Nome varchar(50) NOT NULL,  
    Rec_telefonico varchar(15) NOT NULL,  
    PRIMARY KEY(Cod_negozio)  
);
```

```
CREATE TABLE Categoria (  
    Id_categoria smallint(4),  
    Nome varchar(30) NOT NULL,  
    PRIMARY KEY(Id_categoria)  
);
```

```
/*TABELLE GESTIONE DELL'INDIRIZZO*/
```

```
CREATE TABLE Nazione (  
    Id_nazione smallint(3) NOT NULL,  
    Nome varchar(30) NOT NULL,  
  
    PRIMARY KEY(Id_nazione)  
);
```

```
CREATE TABLE Regione (  
    Id_regione int NOT NULL,  
    Id_nazione smallint(3) not NULL,
```

```

    Nome varchar(30) NOT NULL,

    PRIMARY KEY(Id_regione),
    FOREIGN KEY(Id_nazione) REFERENCES Nazione(Id_nazione)
);

CREATE TABLE Citta (
    Id_citta int NOT NULL,
    Id_regione int NOT NULL,
    Nome varchar(30) NOT NULL,

    PRIMARY KEY(Id_citta),
    FOREIGN KEY(Id_regione) REFERENCES Regione(Id_regione)
);

CREATE TABLE Indirizzo (
    Id_indirizzo int NOT NULL AUTO_INCREMENT,
    Id_citta int NOT NULL,
    Nome varchar(50) NOT NULL,
    Num_civico int NOT NULL,

    PRIMARY KEY(Id_indirizzo),
    FOREIGN KEY(Id_citta) REFERENCES Citta(Id_citta)
);

/*////////////////////////////////////*/

```

```
CREATE TABLE Magazzino (  
    Cod_magazzino varchar(5) NOT NULL,  
    Id_indirizzo int NOT NULL,  
  
    PRIMARY KEY(Cod_magazzino),  
    FOREIGN KEY(Id_indirizzo) REFERENCES Indirizzo(Id_indirizzo)  
);
```

```
CREATE TABLE Articolo (  
    Cod_articolo varchar(10) NOT NULL,  
    Cod_negozio smallint(7) NOT NULL,  
    Cod_magazzino varchar(5) NOT NULL,  
    Id_categoria smallint(4) NOT NULL,  
    Nome varchar(50) NOT NULL,  
    Descrizione varchar(500) NOT NULL,  
    Quantita int NOT NULL DEFAULT 0  
    CHECK (Quantita>=0),  
  
    PRIMARY KEY(Cod_articolo),  
    FOREIGN KEY(Cod_negozio) REFERENCES Negozio(Cod_negozio),  
    FOREIGN KEY(Cod_magazzino) REFERENCES Magazzino(Cod_magazzino),  
    FOREIGN KEY(Id_categoria) REFERENCES Categoria(Id_categoria)  
);
```

```

CREATE TABLE Prezzo (
    Id_prezzo int AUTO_INCREMENT primary key,
    Cod_articolo varchar(10) NOT NULL,
    Prezzo double(7,2) NOT NULL
    CHECK (Prezzo>0),
    Sconto int(2) DEFAULT 0 /*Imposto uno sconto dello 0% se non viene
inserito nulla*/
    CHECK (Sconto>=0 AND Sconto<100),

    FOREIGN KEY(Cod_articolo) REFERENCES Articolo(Cod_articolo)
);
/*----- */

```

```

CREATE TABLE Acquirente (
    Id_acquirente int NOT NULL,
    id_indirizzo_cons int NOT NULL,
    Nome varchar(30) NOT NULL,
    Cognome varchar(30) NOT NULL,
    Email varchar(50) NOT NULL,
    Genere char(1) NOT NULL
    CHECK (Genere IN ('M', 'F', 'N')), /*Maschio, Femmina, Non specificato*/
    Password varchar(24) NOT NULL,
    Data_nascita date NOT NULL,

    PRIMARY KEY(Id_acquirente),
    FOREIGN KEY(Id_indirizzo_cons) REFERENCES Indirizzo(Id_indirizzo)
);

```

```

CREATE TABLE Recensioni (
    Cod_recensioni varchar(5) NOT NULL,
    Cod_articolo varchar(10) NOT NULL,
    Id_acquirente int NOT NULL,
    Stelle smallint(1) NOT NULL /*Se inserisci una recensione devi mette a
forza un num di stelle*/
    CHECK (Stelle IN (1,2,3,4,5)),
    Commento varchar(200) DEFAULT '',

    PRIMARY KEY(Cod_recensioni),
    FOREIGN KEY(Cod_articolo) REFERENCES Articolo(Cod_articolo),
    FOREIGN KEY(Id_acquirente) REFERENCES Acquirente(Id_acquirente)
);

```

```

CREATE TABLE Acquisto (
    Id_acquirente int NOT NULL,
    Cod_articolo varchar(10) NOT NULL,
    Data_acquisto datetime DEFAULT NOW(),
    Acquistato Boolean DEFAULT 0, /*Se è true è un ordine, se è false è ancora
nel carrello*/
    Quantita int NOT NULL
    CHECK (Quantita>=1),

    PRIMARY KEY (Id_acquirente, Cod_articolo),
    FOREIGN KEY(Id_acquirente) REFERENCES Acquirente(Id_acquirente),
    FOREIGN KEY(Cod_articolo) REFERENCES Articolo(Cod_articolo)
);

```

```

CREATE TABLE Carta (
    Num_carta char(16) NOT NULL,
    Id_acquirente int NOT NULL,
    Titolare varchar(60) NOT NULL,
    Cvv smallint(3) NOT NULL,
    Scadenza_mese smallint(2) NOT NULL
    CHECK (Scadenza_mese >0 && Scadenza_mese <13),
    Scadenza_anno year NOT NULL,

    PRIMARY KEY(Num_carta),
    FOREIGN KEY(Id_acquirente) REFERENCES Acquirente(Id_acquirente)
);

```

4.2 Inserimento trigger

```

/*Trigger per gestire gli acquisti*/
CREATE TRIGGER `TriggerCarrello` AFTER UPDATE ON `acquisto` FOR EACH ROW
UPDATE Articolo art
INNER JOIN Acquisto acq
ON art.Cod_articolo=acq.Cod_articolo
SET art.Quantita = art.Quantita - acq.Quantita
WHERE acq.Acquistato=1;

```

4.3 Query

4.3.1 Viste

```
/*Vista per il prezzo scontato*/
```

```
CREATE VIEW CalcolaPrezzoScontato (PrezzoScontato, CodArticolo) AS  
SELECT ROUND((prezzo.Prezzo-((prezzo.Prezzo*prezzo.Sconto)/100)), 2),  
articolo.Cod_articolo  
FROM prezzo, articolo  
WHERE prezzo.Cod_articolo = articolo.Cod_articolo;
```

```
/*VIEWS per vedere il PrezzoScontato*quantita */
```

```
CREATE VIEW PrezzoScontatoxQuantita (Id_acquirente, Cod_articolo,  
PrezzoQuantita, acquistato) AS  
SELECT acquisto.Id_acquirente , acquisto.Cod_articolo  
,(calcolaprezzoscontato.PrezzoScontato *acquisto.Quantita) AS Costo,  
acquisto.Acquistato  
FROM acquisto, calcolaprezzoscontato  
WHERE calcolaprezzoscontato.CodArticolo = acquisto.Cod_articolo;
```

```
/*media stelle per ogni articolo*/
```

```
CREATE VIEW MediaStelle (Cod_articolo, media) AS  
SELECT articolo.Cod_articolo, ROUND(AVG(recensioni.Stelle),2 )AS Media_Stelle  
FROM articolo, recensioni  
WHERE articolo.Cod_articolo = recensioni.Cod_articolo  
GROUP BY articolo.Cod_articolo;
```


4.3.2 Interrogazioni

Da operazione 1 a operazione 7 sono racchiuse in un file esterno (Operazioni di inserimento dei dati)

Operazione 8 (cancellazione acquirente)

```
/*Rimozione di uno specifico acquirente*/  
  
DELETE FROM carta WHERE Id_acquirente= '20098';  
  
DELETE FROM recensioni WHERE Id_acquirente = '20098';  
  
DELETE FROM acquisto WHERE Id_acquirente = '20098';  
  
DELETE FROM acquirente WHERE Id_acquirente = '20098';
```

Operazione 9 (visualizzazione articoli disponibili)

```
/*Per ogni negozio vedere tutti gli articoli con ognuno il proprio prezzo (e  
sconto) */  
  
SELECT negozio.Cod_negoziio, negozio.Nome, articolo.Cod_articolo,  
articolo.Nome, calcolaprezzoscontato.PrezzoScontato  
  
FROM negozio, articolo, calcolaprezzoscontato  
  
WHERE negozio.Cod_negoziio = articolo.Cod_negoziio AND articolo.Cod_articolo =  
calcolaprezzoscontato.CodArticolo
```

Operazione 10 (visualizzazione recensioni di articoli con prezzo compreso tra..)

```
/*Tutte le recensioni di un articolo che ha un prezzo compreso tra 5 euro e 30  
euro*/  
  
SELECT articolo.Nome, recensioni.Cod_recensioni, recensioni.Stelle,  
recensioni.Commento, calcolaprezzoscontato.PrezzoScontato  
  
FROM recensioni, articolo, prezzo, calcolaprezzoscontato  
  
WHERE recensioni.Cod_articolo = articolo.Cod_articolo AND  
  
prezzo.Cod_articolo = articolo.Cod_articolo AND
```

```
calcolaprezzoscontato.CodArticolo = articolo.Cod_articolo AND  
calcolaprezzoscontato.PrezzoScontato >=5 AND  
calcolaprezzoscontato.PrezzoScontato<=30
```

Operazione 11 & 12 (visualizzazione media stelle e articoli più graditi)

(Utilizzando la vista “mediastelle”)

```
/*Articoli che hanno come media 4 stelle in su*/  
SELECT mediastelle.Cod_articolo, mediastelle.media  
FROM mediastelle, articolo  
WHERE articolo.Cod_articolo = mediastelle.Cod_articolo  
AND mediastelle.media >= 4
```

Operazione 13 (visualizzazione articoli per categoria)

```
/*Vedere tutti gli articoli per categoria*/  
SELECT categoria.Nome, articolo.Nome  
FROM categoria, articolo  
WHERE categoria.Id_categoria = articolo.Id_categoria
```

Operazione 14 (visualizzazione costo totale del carrello di ogni acquirente)

```
/*Vedere tutti gli elementi del carrello per ogni acquirente (vista)*/  
SELECT acquirente.Nome, acquirente.Cognome, articolo.Nome,  
prezzoscontatoxquantita.PrezzoQuantita , acquisto.Quantita  
FROM articolo, acquisto, prezzoscontatoxquantita, acquirente  
WHERE articolo.Cod_articolo = acquisto.Cod_articolo AND acquisto.Id_acquirente  
= acquirente.Id_acquirente AND  
prezzoscontatoxquantita.Cod_articolo = acquisto.Cod_articolo AND  
acquisto.Acquistato = 0
```

Operazione 15 (controllo possesso carta)

```
/*Query per controllare se l'acquirente ha almeno una carta*/  
SELECT acquirente.Id_acquirente, acquirente.Nome, acquirente.Cognome, CASE  
WHEN acquirente.Id_acquirente IN (SELECT carta.Id_acquirente FROM carta)  
THEN 'SI'
```

```
        ELSE 'NO'
    END AS 'Ha la carta?'
FROM acquirente
```

Operazione 16 (per ogni negozio visualizzare articoli con prezzo effettivo)

/*Per ogni negozio vedere tutti gli articoli con ognuno il proprio prezzo (e sconto) */

```
SELECT negozio.Cod_negozio, negozio.Nome, articolo.Cod_articolo,
articolo.Nome, calcolaprezzoscontato.PrezzoScontato
FROM negozio, articolo, calcolaprezzoscontato
WHERE negozio.Cod_negozio = articolo.Cod_negozio
      AND articolo.Cod_articolo = calcolaprezzoscontato.CodArticolo
```

Operazione 17 (Visualizzazioni ordini effettuati dall'acquirente)

/*Vedere tutti gli ordini per ogni acquirente*/

```
SELECT acquirente.Nome, acquirente.Cognome, articolo.Nome,
prezzoscontatoxquantita.PrezzoQuantita , acquisto.Quantita
FROM articolo, acquisto, prezzoscontatoxquantita, acquirente
WHERE articolo.Cod_articolo = acquisto.Cod_articolo AND acquisto.Id_acquirente
= acquirente.Id_acquirente AND
prezzoscontatoxquantita.Cod_articolo = acquisto.Cod_articolo AND
acquisto.Acquistato = 1
GROUP BY acquirente.Nome, acquirente.Cognome
```