

Progetto di Basi di Dati

Piattaforma di annunci

Francesco Di Lauro

Progetto di Basi di Dati

Piattaforma di annunci

1. Raccolta delle specifiche della realtà di interesse: Descrizione

Si vuole realizzare un piattaforma che consenta a utenti di vendere e comprare beni quali auto, moto, accessori ed altro mediante l'utilizzo di inserzioni.

Degli utenti si vuole memorizzare nome, cognome, codice fiscale, data di nascita, indirizzo di residenza, numero di telefono, mail, e password per effettuare l'accesso. Gli utenti della piattaforma possono essere divisi in inserzionisti e acquirenti.

Gli inserzionisti pubblicano gli annunci. Degli annunci si è interessati al titolo, alla descrizione, alla data di inserzione, alla locazione dell'annuncio e da un valore che indichi se l'annuncio è stato approvato o meno dagli amministratori, affinché questo possa essere visualizzato.

Gli annunci sono identificati mediante un id univoco all'interno della piattaforma. Un annuncio riguarda un singolo prodotto, del quale si vuole memorizzare il nome, il prezzo e la condizione (stato di usura del prodotto) e se è stato venduto. Ogni prodotto inserito, deve appartenere a una o più categorie presenti sulla piattaforma. Ogni categoria ha un nome, ed è individuata da un codice.

L'utente acquirente può effettuare l'acquisto di un prodotto, inoltre ha la possibilità di contattare l'utente inserzionista, tramite l'utilizzo di messaggi, che contengono un id, il testo del messaggio, e la data di invio.

Gli utenti della piattaforma hanno la possibilità di memorizzare uno o più metodi di pagamento per effettuare un acquisto, e per ricevere gli importi. Il metodo di pagamento contiene il numero della carta, la data di scadenza e il CVC relativo. Dell'utente acquirente si vuole inoltre memorizzare il numero di acquisti effettuati sulla piattaforma.

Gli amministratori della piattaforma si occupano dell'inserimento di nuove categorie e della verifica degli annunci, in particolare si vuole tener traccia della corrispondenza tra l'annuncio, e l'amministratore che l'ha approvato. Degli amministratori si vogliono memorizzare il codice fiscale, il nome, il cognome, la mail, la password per accedere alla piattaforma, e il numero di telefono.

Specifiche della realtà d'interesse

La realtà che si vuole andare a rappresentare, riguarda la gestione di una piattaforma adibita alla pubblicazione di annunci.

La piattaforma si rivolge a tutti coloro che hanno intenzione di acquistare e/o vendere beni materiali suddivisi per categorie, in modo tale da facilitarne la ricerca. È possibile stabilire un contatto tra acquirente e venditore tramite una chat apposita, per richiedere informazioni sul prodotto e accordarsi su eventuali costi, spedizione e

metodi di pagamento (carte di credito). Riguardo il metodo di pagamento è possibile memorizzarlo sulla piattaforma stessa. Gli annunci pubblicati dai rivenditori devono rispettare regole imposte dagli amministratori della piattaforma, e solo dopo essere stati revisionati, possono essere pubblicati. Per ogni annuncio pubblicato riguardante un prodotto è possibile inserire un titolo di massimo 20 caratteri, indicarne la categoria, e una descrizione di massimo 200 caratteri. L'annuncio potrà essere inoltre modificato e cancellato dall'inserzionista. L'utente potrà indicare una locazione per l'annuncio, aiutando in questo modo l'acquirente, che potrà cercare in base alla posizione e ritirarlo lui stesso.

Il sistema deve avere le seguenti caratteristiche suddivise per profili di utenza:

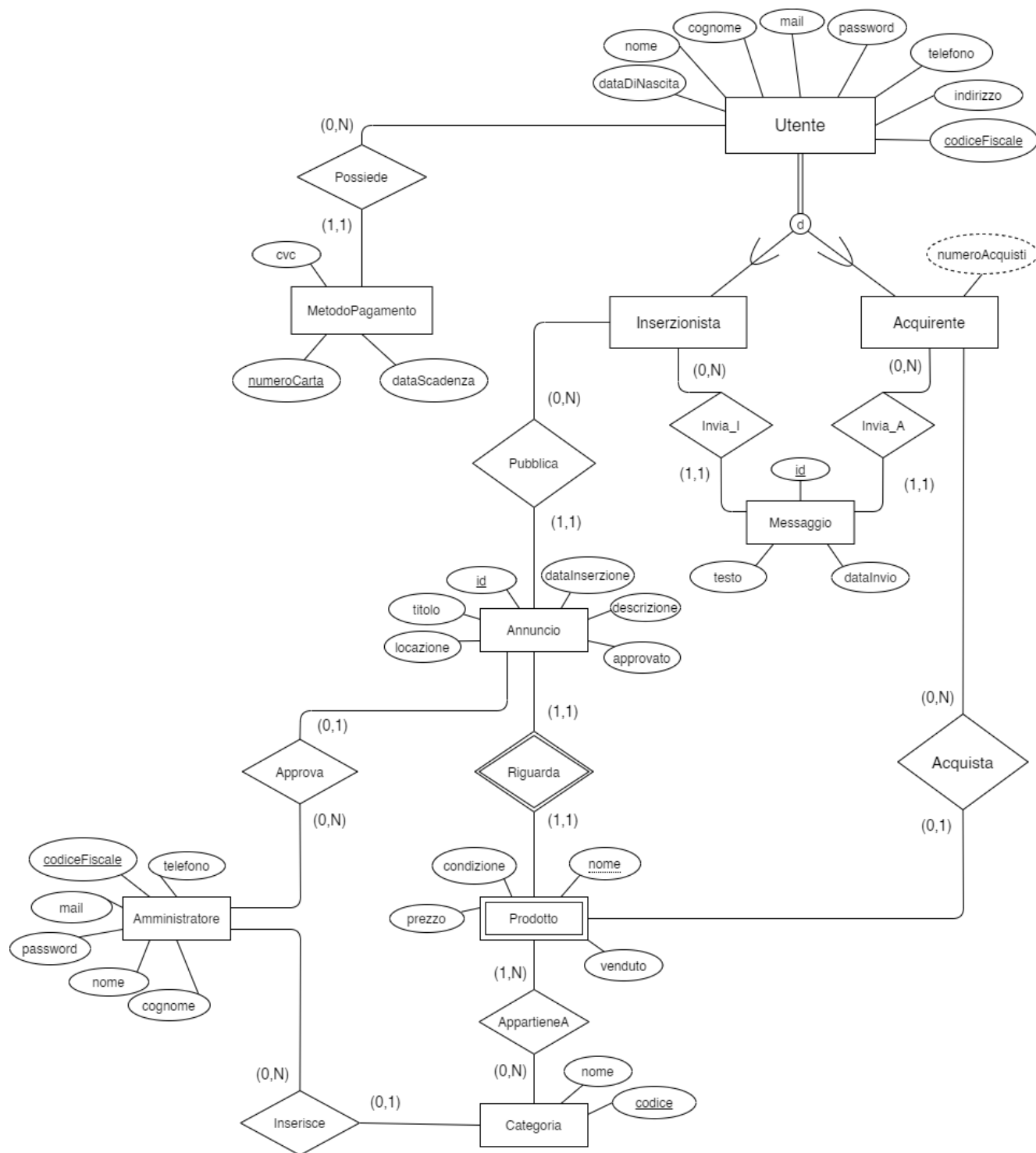
- Amministratore: gestisce gli annunci ed inserisce nuove categorie. Le categorie supportate “nativamente” dal sistema sono auto e moto. Altre categorie devono poter essere create dall'amministratore. L'utente amministratore approva le inserzioni dell'utente inserzionista.
- Utente inserzionista: si autentica al sistema per inserire annunci di prodotti.
- Utente acquirente: si autentica al sistema per visualizzare annunci ed effettuare acquisti.

Glossario dei termini:

Termine	Descrizione
Utente	Utente che si registra alla piattaforma con le proprie credenziali, e informazioni di base. L'utente al momento della registrazione dovrà inserire il tipo di account creato: se Inserzionista o semplice Acquirente.
Inserzionista	Un Utente che pubblica gli annunci relativi ai prodotti che desidera vendere
Acquirente	Un Utente che cerca tra gli annunci all'interno della piattaforma, per trovare prodotti di suo gradimento
Messaggio	Messaggio che un Utente intenzionato all'acquisto, manda all'inserzionista e viceversa, per quanto riguarda la risposta.
Amministratore	Un Moderatore della piattaforma, incaricato di verificare l'ideoneità degli annunci presentati dagli Inserzionisti
Annuncio	L'annuncio creato dall'Inserzionista, per presentare il prodotto
Prodotto	Il Prodotto messo in vendita dall' Inserzionista

2. Progettazione concettuale della base di dati

Schema EER



Dizionario delle entità

Nome Entità	Descrizione	Attributi	Identificatore
Utente	Utente della piattaforma	-nome -cognome -dataDiNascita -mail -password -telefono -indirizzo -codiceFocale	codiceFiscale
Inserzionista	Pubblica gli annunci	-numeroInserzioni (derivato)	
Acquirente	Effettua acquisti	-numeroAcquisti (derivato)	
Messaggio	Messaggio inviato dall'Acquirente all'Inserzionista	-id -testo -dataInvio	id
MetodoPagamento	La carta di credito che possiede l'acquirente	-cvc -numeroCarta -dataScadenza	numeroCarta
Annuncio	L'annuncio pubblicato dall'inserzionista	-id -titolo -locazione (informazione di vendita) -dataInserzione -descrizione -approvato (intero)	id
Prodotto	Il prodotto a cui si riferisce l'annuncio	-nome -condizione -prezzo -venduto	nome,id

Categoria	La categoria alla quale appartiene il prodotto	-nome -codice	codice
Amministratore	Il moderatore degli annunci sulla piattaforma	-nome -cognome -mail -password -codiceFiscale -telefono	codiceFiscale

Dizionario delle relazioni

Nome Relazione	Descrizione	Entità coinvolte
Pubblica	Un Inserzionista pubblica un annuncio	Inserzionista(0,n) Annuncio(1,1)
Riguarda	Associazione tra l'annuncio, e il prodotto che lo riguarda	Annuncio(1,1) Prodotto(1,1)
AppartieneA	Associazione tra il prodotto e la/le categoria/e a cui appartiene	Prodotto(1,n) Categoria(0,n)
Acquista	L'acquisto di un prodotto da parte di un acquirente	Acquirente(0,n) Prodotto(1,1)
Possiede	Associazione tra l'utente acquirente e i suoi metodi di pagamento (carte di credito)	Acquirente(0,n) MetodoPagamento(1,1)
Invia_A	Associazione tra un messaggio e l'Acquirente che l'ha spedito	Acquirente(0,n) Messaggio(1,1)
Invia_I	Associazione tra un messaggio, e l'Inserzionista che l'ha spedito	Inserzionista (0,n) Messaggio(1,1)
Approva	Associazione tra un Annuncio, e l'amministratore che l'ha approvato	Annuncio(0,1) (cardinalità minima 0, perché l'annuncio può

		essere stato inserito senza essere approvato) Amministratore(o,n)
Inserisce	Associazione tra una categoria, e l'amministratore che l'ha inserita	Categoria(o,1) (cardinalità minima 0 perché ci sono categorie "native" nel sistema) Amministratore(o,n)

Vincoli non esprimibili nello schema

- L'attributo "locazione" dell'entità annuncio, è della forma <CITTÀ(PROVINCIA)> e non può superare i 20 caratteri.
- L'attributo "descrizione" di Annuncio, ha un numero massimo di caratteri che è di 500.
- L'attributo "testo" di Messaggio, ha un numero massimo di caratteri che è di 500.

3. Carico applicativo e operazioni per la gestione della base di dati.

Tavola dei volumi

Concetto	Tipo	Volume
Utente	E	50
Inserzionista	E	10
Acquirente	E	40
Messaggio	E	1000
MetodoPagamento	E	70
Annuncio	E	100
Prodotto	E	100
Categoria	E	10
Amministratore	E	3
Invia_I	R	500
Invia_A	R	500
Possiede	R	70
Pubblica	R	100
Riguarda	R	100
AppartieneA	R	150
Acquista	R	30

Approva	R	80
Inserisce	R	8

Tavola delle operazioni

Operazione	Tipo	Frequenza
1. Registrare un nuovo utente sulla piattaforma	I	1/gg
2. Visualizzare tutti gli annunci inseriti da un dato Utente, ordinati per data	I	5/gg
3. Visualizzare l'acquirente che ha effettuato più acquisti sulla piattaforma	I	1/mm
4. Visualizzare tutti gli annunci approvati da un dato Amministratore	I	1/mm
5. Inserire un nuovo amministratore	I	1/mm
6. Visualizzare tutte le categorie, e gli amministratori che le hanno inserite	B	2/mm
7. Visualizzare tutti i metodi di pagamento di un dato acquirente	I	2/mm
8. Visualizzare tutti i messaggi inviati da un dato acquirente, ordinati per data	B	1/mm
9. Visualizzare il nome, il cognome e il numero di telefono di tutti gli inserzionisti che hanno pubblicato annunci approvati, la cui locazione è la provincia di Salerno (SA).	I	6/mm
10. Far approvare ad un amministratore un annuncio	I	5/gg
11. Modificare il titolo dell'annuncio di un dato Inserzionista	I	5/mm
12. Acquistare un prodotto che abbia condizione "nuovo"	I	2/gg

4. Progettazione logica

Analisi delle ridondanze

Il dato ridondante risulta essere l'attributo numeroAcquisti dell'entità acquirente. Questo attributo, può anche essere derivato andando a contare il numero di partecipazioni nella relazione "Acquista" verso l'entità Prodotto. Supponendo che la dimensione di numeroAcquisti sia di 4 byte, e sapendo che il volume dell'entità Acquirente è di 40, la memorizzazione di questo attributo nella base di dati occuperebbe circa 160 byte. Le operazioni coinvolte nel calcolo per l'analisi della ridondanza sono l'operazione 3: Visualizzare l'acquirente che ha effettuato più

acquisti sulla piattaforma, e l'operazione 12: Acquistare un prodotto che abbia condizione "nuovo". Entrambe le operazioni sono interattive.

Tavola degli accessi

Operazione 3: Visualizzare l'acquirente che ha effettuato più acquisti sulla piattaforma; frequenza: 1/mm.

Operazione 3 con ridondanza:

Concetto	Costrutto	Accessi	Tipo accesso
Acquirente	E	40	L

Totale accessi operazione 3 con ridondanza: $40 \cdot 1/\text{mm} = 40$ accessi al mese.

Operazione 3 senza ridondanza:

Concetto	Costrutto	Accessi	Tipo accesso
Acquirente	E	40	L
Acquista	R	30	L

Totale accessi operazione 3 senza ridondanza: $(40+30) \cdot 1/\text{mm} = 70$ accessi al mese.

Operazione 12: Acquistare un prodotto che abbia condizione "nuovo"; frequenza: 2/gg.

Operazione 12 con ridondanza:

Concetto	Costrutto	Accessi	Tipo accesso
Acquista	R	1	S
Acquirente	E	1	L
Acquirente	E	1	S

Totale accessi operazione 12 con ridondanza: $(2+1+2) \cdot 2/\text{gg} = 310$ accessi al mese.

Operazione 12 senza ridondanza:

Concetto	Costrutto	Accessi	Tipo accesso
Acquista	R	1	S

Totale accessi operazione 12 senza ridondanza: $(2 \cdot 1) \cdot 2/\text{gg} = 124$ accessi al mese.

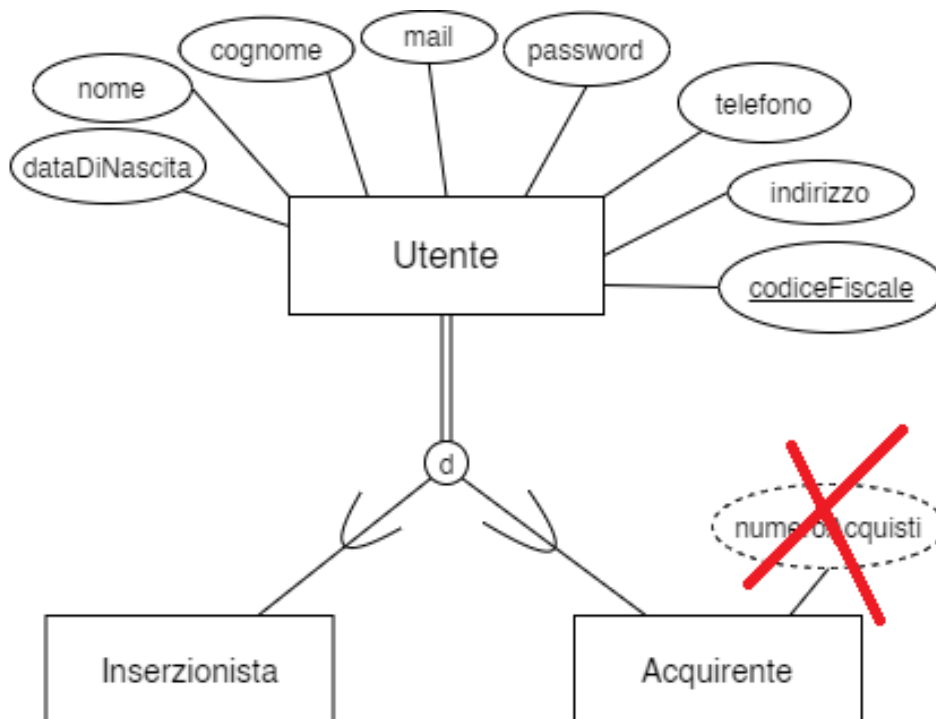
Totale accessi con ridondanza: $40 + 310 = 350$ accessi/mm

Totale accessi senza ridondanza: $70 + 124 = 194$ accessi/mm

Dato il minor numero di accessi, dovuti al fatto che le operazioni coinvolte nell'utilizzo efficiente del dato ridondante vengono effettuate 1 volta al mese, mentre l'aggiornamento del dato stesso, comporta una notevole quantità di accessi in un tempo inferiore, è più efficiente **non mantenere** il dato ridondante "numeroAcquisti".

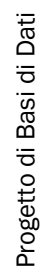
Eliminazione delle gerarchie

È necessario ora effettuare l'eliminazione della specializzazione dell'entità Utente:



La scelta presa è quella di effettuare l'accorpamento delle figlie della generalizzazione, nell'entità padre. In questo modo gli attributi di Inserzionista e Acquirente (numeroAcquisti, eliminato precedentemente nella fase di analisi delle ridondanze) e le relative associazioni vengono aggiunte all'entità genitore Utente, a cui verrà aggiunto un ulteriore attributo ruolo, per distinguere tra le due figure. In questo modo si evitano anche gli attributi NULL.

Lo schema EER ristrutturato risulta essere il seguente:



Traduzione dello schema concettuale in schema relazionale.

Di seguito lo schema relazionale della base di dati:

Utente(codiceFiscale, nome, cognome, mail, password, dataDiNascita, telefono, indirizzo, ruolo)

MetodoPagamento(numeroCarta, dataScadenza, cvc, Utente.codiceFiscale↑)

Messaggio(idMessaggio, testo, dataInvio, Utente.codiceFiscale1↑, Utente.codiceFiscale2↑)

Annuncio(idAnnuncio, dataInserzione, titolo, locazione, descrizione, approvato, Utente.codiceFiscale↑)

Prodotto(nome, Annuncio.idAnnuncio↑, condizione, prezzo, venduto)

Acquisto(Prodotto.nome↑, Annuncio.idAnnuncio↑, Utente.codiceFiscale↑)

Amministratore(codiceFiscale, nome, cognome, mail, password, telefono)

Approvato(Annuncio.idAnnuncio↑, Amministratore.codiceFiscale↑)

Categoria(codice, nome, Amministratore.codiceFiscale↑*)

AppartieneA(Categoria.codice↑, Prodotto.nome↑, Annuncio.idAnnuncio↑)

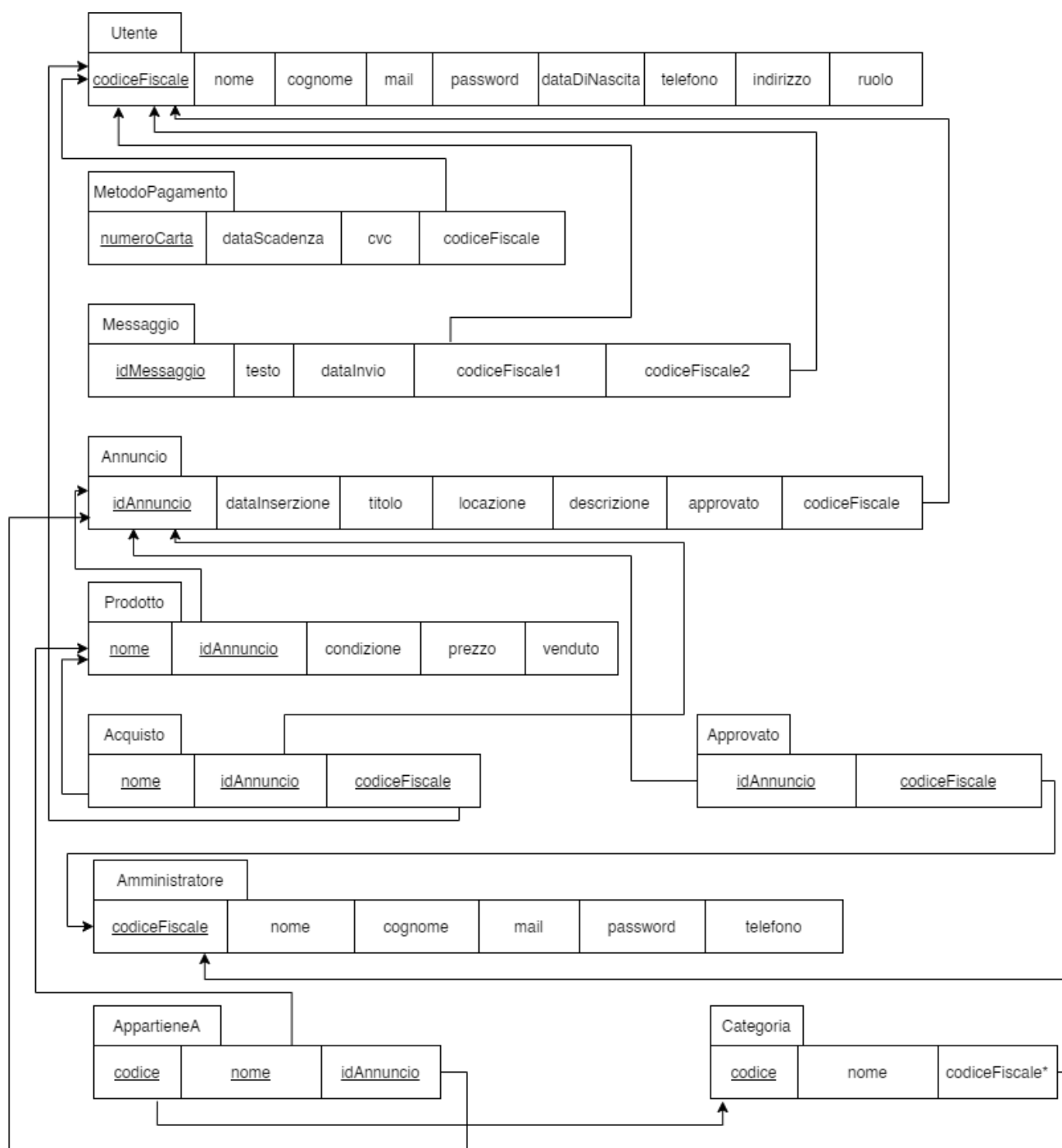
Normalizzazione

Il database risulta essere già normalizzato:

La base di dati è in prima forma normale poiché non vi sono attributi multivalore e/o composti.

La base di dati è in seconda forma normale poiché è in prima forma normale, e nessun attributo dipende parzialmente dalla chiave.

La base di dati risulta essere anche in terza forma normale poiché già in seconda forma normale, e inoltre non sono presenti dipendenze transitive fra attributi non chiave e chiave primaria.



5. Realizzazione della base di dati con MySQL

```
1 • drop schema if exists eSales;
2 • create schema eSales;
3 • use eSales;
4
5 • create table Utente(
6     codiceFiscale char(16) NOT NULL,
7     nome varchar(30) NOT NULL,
8     cognome varchar(30) NOT NULL,
9     mail varchar(40) NOT NULL UNIQUE,
10    pwd varchar(20) NOT NULL,
11    dataDiNascita date NOT NULL,
12    telefono varchar(15) NOT NULL,
13    indirizzo varchar(50) NOT NULL,
14    ruolo int NOT NULL,
15    PRIMARY KEY (codiceFiscale)
16 );
17
18 • create table MetodoPagamento(
19     numeroCarta varchar(16) NOT NULL,
20     dataScadenza date NOT NULL,
21     cvc smallint NOT NULL,
22     codiceFiscale char(16) NOT NULL,
23     PRIMARY KEY (numeroCarta),
24     FOREIGN KEY(codiceFiscale) REFERENCES Utente(codiceFiscale) on update cascade on delete cascade
25 );
26
27 • create table Messaggio(
28     idMessaggio int NOT NULL auto_increment,
29     testo varchar(500) NOT NULL,
30     dataInvio timestamp NOT NULL,
31     codiceFiscaleMittente char(16) NOT NULL,
32     codiceFiscaleDestinatario char(16) NOT NULL,
33     PRIMARY KEY (idMessaggio),
34     FOREIGN KEY (codiceFiscaleMittente) REFERENCES Utente(codiceFiscale) on update cascade on delete cascade,
35     FOREIGN KEY (codiceFiscaleDestinatario) REFERENCES Utente(codiceFiscale) on update cascade on delete cascade
36 );
```

```

38 • ⊖ create table Annuncio(
39     idAnnuncio int NOT NULL auto_increment,
40     dataInserzione date NOT NULL,
41     titolo varchar(100) NOT NULL,
42     locazione varchar(50) NOT NULL,
43     descrizione varchar(500) NOT NULL,
44     approvato boolean NOT NULL,
45     codiceFiscale char(16) NOT NULL,
46     PRIMARY KEY (idAnnuncio),
47     FOREIGN KEY(codiceFiscale) REFERENCES Utente(codiceFiscale) on update cascade on delete cascade
48 );
49
50 • ⊖ create table Prodotto(
51     nome varchar(20) NOT NULL,
52     idAnnuncio int NOT NULL,
53     condizione varchar(20) NOT NULL,
54     prezzo float NOT NULL,
55     venduto boolean DEFAULT false,
56     PRIMARY KEY(nome,idAnnuncio),
57     FOREIGN KEY(idAnnuncio) REFERENCES Annuncio(idAnnuncio) on update cascade on delete cascade
58 );
59
60 • ⊖ create table Acquisto(
61     nome varchar(20) NOT NULL,
62     idAnnuncio int NOT NULL,
63     codiceFiscale char(16) NOT NULL,
64     PRIMARY KEY(nome,idAnnuncio,codiceFiscale),
65     FOREIGN KEY(nome,idAnnuncio) REFERENCES Prodotto(nome,idAnnuncio) on update cascade on delete cascade,
66     FOREIGN KEY (codiceFiscale) REFERENCES Utente(codiceFiscale) on update cascade on delete cascade
67 );
68
69 • ⊖ create table Amministratore(
70     codiceFiscale char(16) NOT NULL PRIMARY KEY,
71     nome varchar(30) NOT NULL,
72     cognome varchar(30) NOT NULL,
73     mail varchar(40) NOT NULL,
74     pwd varchar(20) NOT NULL,
75     telefono varchar(15) NOT NULL
76 );

```

```

78 • ○ create table Approvato(
79     idAnnuncio int NOT NULL,
80     codiceFiscale char(16) NOT NULL,
81     PRIMARY KEY(idAnnuncio,codiceFiscale),
82     FOREIGN KEY(idAnnuncio) REFERENCES Annuncio(idAnnuncio)
83     on update cascade
84     on delete cascade,
85     FOREIGN KEY(codiceFiscale) REFERENCES Amministratore(codiceFiscale)
86     on update cascade
87     on delete cascade
88 );
89
90 • ○ create table Categoria(
91     codice int NOT NULL auto_increment,
92     nome varchar(20) NOT NULL,
93     codiceFiscale char(16),
94     PRIMARY KEY (codice),
95     FOREIGN KEY(codiceFiscale) REFERENCES Amministratore(codiceFiscale)
96     on delete set null
97     on update cascade
98 );
99
100 • ○ create table AppartieneA(
101     codice int NOT NULL,
102     nome varchar(20) NOT NULL,
103     idAnnuncio int NOT NULL,
104     PRIMARY KEY(codice,nome,idAnnuncio),
105     FOREIGN KEY(codice) REFERENCES Categoria(codice)
106     on update cascade
107     on delete cascade,
108     FOREIGN KEY(nome,idAnnuncio) REFERENCES Prodotto(nome,idAnnuncio)
109     on update cascade
110     on delete cascade
111 );
112

```


6. Implementazione query SQL

OPERAZIONE 1:

Registrare un nuovo utente sulla piattaforma

```
INSERT INTO Utente(codiceFiscale, nome, cognome, mail, pwd,  
dataDiNascita,telefono,indirizzo,ruolo) values (?, ?, ?, ?, MD5(?), ?, ?, ?, ?);
```

All'utente verrà data la possibilità di registrarsi come Acquirente (in tal caso la scelta di ruolo dovrà essere 0 (zero)), oppure inserzionista (scelta 1)

Operazione 2:

Visualizzare tutti gli annunci inseriti da un Utente, ordinati per data

```
SELECT * FROM Annuncio WHERE codiceFiscale= ?  
ORDER BY dataInserzione;
```

Operazione 3:

Visualizzare l'acquirente che ha effettuato più acquisti sulla piattaforma

```
SELECT *  
FROM Utente JOIN (SELECT Count(*) AS numeroAcquisti,codiceFiscale  
FROM Acquisto  
GROUP BY codiceFiscale  
Having MAX(numeroAcquisti) ) AS  
MaxAcquistiUtente ON  
Utente.codiceFiscale=MaxAcquistiUtente.codiceFiscale;
```

Operazione 4:

Visualizzare tutti gli annunci approvati da un dato Amministratore

```
SELECT * FROM Approvato WHERE codiceFiscale=?;
```

Operazione 5:

Inserire un nuovo Amministratore

```
INSERT INTO Amministratore (cognome, nome, codiceFiscale, telefono,  
mail,pwd) values (?, ?, ?, ?, MD5(?), ?);
```

Operazione 6:

Visualizzare tutte le categorie, e gli amministratori che le hanno inserite

```
SELECT C.nome, A.nome, A.cognome  
FROM Amministratore AS A, Categoria AS C  
WHERE A.codiceFiscale=C.codiceFiscale;
```

Operazione 7:

Visualizzare tutti i metodi di pagamento di un dato Acquirente

```
SELECT *  
FROM MetodoPagamento  
WHERE codiceFiscale=?;
```

Operazione 8:

Visualizzare tutti i messaggi inviati da un dato acquirente, ordinati per data

```
SELECT *  
FROM Messaggio  
WHERE codiceFiscaleMittente=? ORDER BY dataInvio DESC;
```

Operazione 9:

Visualizzare il nome, il cognome e il numero di telefono di tutti gli inserzionisti che hanno pubblicato annunci approvati, la cui locazione è la provincia di Salerno (SA).

```
SELECT nome,cognome, telefono  
FROM Utente  
WHERE codiceFiscale IN(SELECT codiceFiscale  
FROM Annuncio  
WHERE approvato=true AND  
locazione LIKE '%salerno%');
```

Operazione 10:

Far approvare ad un Amministratore un Annuncio

Vengono prima visualizzati gli annunci che non sono ancora verificati:

```
SELECT * FROM Annuncio WHERE approvato=false;
```

Poi si chiede all'Amministratore il suo codice fiscale, e l'id dell'annuncio da verificare.

```
UPDATE Annuncio SET approvato=true WHERE idAnnuncio=2;
```

```
INSERT INTO Approvato values(?,?);
```

Operazione 11:

Modificare il titolo dell'annuncio di un dato Inserzionista

Si chiede il codice Fiscale dell'inserzionista, si restituiscono in output tutti gli annunci dell'inserzionista, e poi si attua la modifica dopo aver inserito l'id dell'annuncio da modificare

```
SELECT * FROM Annuncio WHERE codiceFiscale=?;
```

```
UPDATE Annuncio SET titolo=? WHERE idAnnuncio=?;
```

Operazione 12:

Acquistare un prodotto che abbia condizione “nuovo”

Vengono visualizzati tutti i prodotti che abbiano condizione nuovo, non ancora venduti:

```
SELECT * FROM prodotto WHERE condizione LIKE '%nuovo%' AND  
venduto=false;
```

Viene inserito l'acquisto:

```
INSERT INTO Acquisto values(?,?,?);
```

E si provvede a impostare il prodotto come venduto:

```
UPDATE prodotto
```

```
SET venduto=true WHERE nome= ? AND idAnnuncio = ?;
```

7. Test dell'applicazione Java

L'interazione con l'utente avviene mediante un'interfaccia di tipo testuale, dove si espongono le operazioni che l'applicazione è in grado di eseguire. L'utente inserisce il codice della scelta da eseguire, e l'operazione corrispondente viene eseguita:

```
ESALES:
*****
0) ESCI
1) Inserisci nuovo utente
2) Visualizza gli annunci pubblicati da un utente
3) Visualizza utente con più acquisti
4) Visualizza annunci approvati da un amministratore
5) Inserisci un nuovo amministratore
6) Visualizza categorie e gli amministratori che le hanno inserite
7) Visualizza metodi di pagamento di un utente
8) Visualizza messaggi inviati da un utente
9) Visualizza inserzionisti che hanno pubblicato annunci approvati, in località Salerno
10) Fai approvare ad un amministratore un annuncio
11) Modificare il titolo di un annuncio di un dato inserzionista
12) Acquista un prodotto che abbia condizione 'nuovo'
*****
Inserisci scelta:
```

Una volta effettuata la scelta, vengono richiesti gli input in base all'operazione da effettuare:

```
ESales [Java Application] C:\Users\Utente\workspace\workspace\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v20201027-0507\jre\bin\javaw.exe (25 ott 2021, 19:53:46)
9) visualizza inserzionisti che hanno pubblicato annunci approvati, in località Salerno
10) Fai approvare ad un amministratore un annuncio
11) Modificare il titolo di un annuncio di un dato inserzionista
12) Acquista un prodotto che abbia condizione 'nuovo'
*****
Inserisci scelta: 1

----- Inserimento utente -----
Inserisci nome: Luigi
Inserisci cognome: Verdi
Inserisci codice fiscale: LRDies3mpio0130
Inserisci mail: lverdi@gmail.com
Inserisci password: lverdi
Inserisci data di nascita: 2000-01-03
Inserisci numero di telefono: 3288833321
Inserisci indirizzo: Via Roma 84
Inserisci il tipo di account (0 per Acquirente,1 per inserzionista): 0
DBConnectionPool Connessione OK

QUERY: com.mysql.cj.jdbc.ClientPreparedStatement: INSERT INTO Utente VALUES ('LRDies3mpio0130','Luigi','V
Inserimento Utente effettuato

ESALES:
*****
```

Viene restituita sull'interfaccia la query eseguita, e se l'inserimento è andato a buon fine o meno. Poi dopo un delay di 4 secondi, l'interfaccia fa ritornare l'utente al menù delle scelte.

```
Problems | Javadoc | Declaration | Console | Servers | Git Staging
ESales [Java Application] C:\Users\Utente\workspace\ESales\src\main\java\com\esales\justopenjdk\hotspot\jre\full\win32_x86_64_15.0.1.v20201027-0507\jre\bin\javaw.exe (26 ott 2021, 18:42:08)

4) Visualizza annunci approvati da un amministratore
5) Inserisci un nuovo amministratore
6) Visualizza categorie e gli amministratori che le hanno inserite
7) Visualizza metodi di pagamento di un utente
8) Visualizza messaggi inviati da un utente
9) Visualizza inserzionisti che hanno pubblicato annunci approvati, in località Salerno
10) Fai approvare ad un amministratore un annuncio
11) Modificare il titolo di un annuncio di un dato inserzionista
12) Acquista un prodotto che abbia condizione 'nuovo'
*****
Inserisci scelta: 3

----- Visualizza utente con più acquisti -----
QUERY: com.mysql.cj.jdbc.ClientPreparedStatement: SELECT *
FROM Utente JOIN (SELECT Count(*) AS numeroAcquisti, codiceFiscale
FROM Acquisto
GROUP BY codiceFiscale
Having MAX(numeroAcquisti) ) MaxAcquistiUtente ON Utente.codiceFiscale=MaxAcquistiUtente.codiceFiscale;
Printing 1 row from tables utente, null, acquisto
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| codiceFiscale | nome | cognome | mail | pwd | dataDiNascita | telefono | indirizzo | ruolo | numeroAcquisti | codi
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| R5SMRA80A01H703F | mario | rossi | mrossi@gmail.com | mrossi | 1980-01-01 | 081938132 | Via roma 81, Salerno | 0 | 3 | R5SMRA
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

ESALES:
*****
<

ESales [Java Application] C:\Users\Utente\workspace\ESales\src\main\java\com\esales\justopenjdk\hotspot\jre\full\win32_x86_64_15.0.1.v20201027-0507\jre\bin\javaw.exe (25 ott 2021, 19:53:46)
9) Visualizza inserzionisti che hanno pubblicato annunci approvati, in località Salerno
10) Fai approvare ad un amministratore un annuncio
11) Modificare il titolo di un annuncio di un dato inserzionista
12) Acquista un prodotto che abbia condizione 'nuovo'
*****
Inserisci scelta: 2

----- Visualizza annunci di un utente -----
Inserisci codice Fiscale Utente: RNCCLT40P48A831E
QUERY: com.mysql.cj.jdbc.ClientPreparedStatement: SELECT * FROM Annuncio WHERE codiceFiscale = 'RNCCLT40P48A831E' ORDER
Printing 5 rows from table annuncio
+-----+-----+-----+-----+-----+
| idAnnuncio | dataInserzione | titolo | locazione |
+-----+-----+-----+-----+-----+
| 10 | 2021-09-11 | 2010 Opel Astra | Milano (MI) | OPEL ASTRA 1.4 GPL - 5 P -ANNO IMMATRICOLAZIONE
| 7 | 2021-09-12 | Vendo Fiat punto 1.3 mjt | Napoli (Na) | Vendo Fiat Punto 1300 MJT SPORT con assetto a g
| 9 | 2021-09-12 | Fiat Panda nuova | Trapani (TP) | Fiat Panda Doppia chiave Cerchi in lega Comandi
| 8 | 2021-09-14 | Fiat Panda nuova | Torino (TI) | Fiat Panda lounge 1.2 benzina/gpl Impianto gpl
| 11 | 2021-09-14 | Ford Fiesta nuova | Milano (MI) | Vendesi:Ford Fiesta.Anno dicembre 2016.Full opt
+-----+-----+-----+-----+-----+

ESALES:
*****
<

9) Inserisci un nuovo amministratore
6) Visualizza categorie e gli amministratori che le hanno inserite
7) Visualizza metodi di pagamento di un utente
8) Visualizza messaggi inviati da un utente
9) Visualizza inserzionisti che hanno pubblicato annunci approvati, in località Salerno
10) Fai approvare ad un amministratore un annuncio
11) Modificare il titolo di un annuncio di un dato inserzionista
12) Acquista un prodotto che abbia condizione 'nuovo'
*****
Inserisci scelta: 9

----- Visualizza inserzionisti che hanno pubblicato annunci approvati, in località Salerno -----
DBConnectionPool Connessione OK

QUERY: SELECT nome,cognome, telefono
FROM Utente
WHERE codiceFiscale IN(SELECT codiceFiscale
FROM Annuncio
WHERE approvato=true AND locazione like '%salerno%')
Printing 2 rows from table utente
+-----+-----+-----+
| nome | cognome | telefono |
+-----+-----+-----+
| mario | verdi | 081938122 |
| Carlotta | Roncarolo | 0332736230 |
+-----+-----+-----+
```

```

Problems | Javadoc | Declaration | Console | Servers | Git Staging
Eclipse [Java Application] C:\Users\Utente\workspace\progetto_basi_dati\src\main\java\com\esales\app\Main.java (25 ott 2021, 19:53:46)
// Inserisci un nuovo amministratore
10) Fai approvare ad un amministratore un annuncio
11) Modificare il titolo di un annuncio di un dato inserzionista
12) Acquista un prodotto che abbia condizione 'nuovo'
*****
Inserisci scelta: 10

----- Fai approvare ad un Amministratore un annuncio -----
QUERY: SELECT * FROM Annuncio WHERE approvato=false
Printing 3 rows from table annuncio
+-----+-----+-----+-----+-----+
| idAnnuncio | dataInserzione | titolo | locazione | descrizione |
+-----+-----+-----+-----+-----+
| 2 | 2021-09-20 | Kawasaki Er 5 | Salerno (SA) | Vendo bellissima kawasaki er 5 500cc, con 36mila km ORIGINALI, immatricol |
| 9 | 2021-09-12 | Fiat Panda nuova | Trapani (TP) | Fiat Panda Doppia chiave Cerchi in lega Comandi a volante Vetri oscurati |
| 11 | 2021-09-14 | Ford Fiesta nuova | Milano (MI) | Vende:Ford Fiesta.Anno dicembre 2016.Full optional.Modello Titanium. |
+-----+-----+-----+-----+-----+

Inserisci l'id dell'annuncio da approvare: 11
Inserisci codice Fiscale amministratore: BRNDLD37H50B4960
QUERY: com.mysql.cj.jdbc.ClientPreparedStatement: UPDATE Annuncio SET approvato=true WHERE idAnnuncio=11
Inserimento effettuato

ESALES:
*****
0) ESCI
1) Inserisci nuovo utente
2) Visualizza gli annunci pubblicati da un utente

```

```

// Inserisci un nuovo amministratore
6) Visualizza categorie e gli amministratori che le hanno inserite
7) Visualizza metodi di pagamento di un utente
8) Visualizza messaggi inviati da un utente
9) Visualizza inserzionisti che hanno pubblicato annunci approvati, in località Salerno
10) Fai approvare ad un amministratore un annuncio
11) Modificare il titolo di un annuncio di un dato inserzionista
12) Acquista un prodotto che abbia condizione 'nuovo'
*****
Inserisci scelta: 12

----- Acquista un prodotto che abbia condizione 'nuovo' -----
DBConnectionPool Connessione OK

QUERY: com.mysql.cj.jdbc.ClientPreparedStatement: SELECT * FROM Prodotto WHERE condizione LIKE '%nuovo%' AND venduto=false;
Printing 4 rows from table prodotto
+-----+-----+-----+-----+-----+
| nome | idAnnuncio | condizione | prezzo | venduto |
+-----+-----+-----+-----+-----+
| Fiat panda | 8 | nuovo | 12000,000 | (BIT) |
| Fiat panda | 9 | nuovo | 12000,000 | (BIT) |
| Ford Fiesta | 11 | nuovo | 15000,000 | (BIT) |
| Iphone 10 pro | 6 | nuovo | 700,000 | (BIT) |
+-----+-----+-----+-----+-----+

Inserisci l'id dell'annuncio del prodotto da acquistare: 8
Inserisci nome del prodotto da acquistare: Fiat panda
Inserisci codice fiscale dell'utente: RSSMRA80A01H703F
QUERY: com.mysql.cj.jdbc.ClientPreparedStatement: INSERT INTO Acquisto VALUES ('Fiat panda',8,'RSSMRA80A01H703F')
QUERY: com.mysql.cj.jdbc.ClientPreparedStatement: UPDATE Prodotto SET venduto=true WHERE nome= 'Fiat panda' AND idAnnuncio = 8
Prodotto Acquistato!

ESALES:
*****

```