

PROGETTO BASI DI DATI

E- COMMERCE DI STRUMENTI MUSICALI

1. Raccolta delle specifiche della realtà di interesse

Si vuole progettare una base di dati per la gestione di **un e-commerce di strumenti musicali**.

La visione dei prodotti è garantita tramite l'accesso ad un **catalogo**, suddiviso secondo la tipologia di strumento, accessori, interfacce audio/attrezzatura e software.

Per procedere all'acquisto di un prodotto in vendita è necessaria la creazione di un **account** identificato da username, dati anagrafici dell'utente, e-mail e password.

Ogni **utente** può effettuare uno o più acquisti scegliendo tra i diversi tipi di articoli.

L'utente selezionerà l'articolo aggiungendolo in un **carrello**, successivamente procederà all'acquisto del prodotto, inserendo l'indirizzo di residenza e l'intestatario dell'ordine, specificato in fase di ordine. All'interno del carrello sarà indicata la quantità di prodotti ordinati e, in caso, i singoli articoli di cui desideriamo più unità.

Ogni **articolo** deve appartenere ad una categoria che può essere visualizzata da tutti gli utenti, della quale si conosce il nome, il codice univoco del prodotto, tipologia e una breve descrizione. Ad ogni prodotto sarà indicata la sua disponibilità e quantità.

Dopodiché verrà indirizzato verso la transazione, la verifica del pagamento e la conferma dell'acquisto. Assegnazione del punto di destinazione. L'acquisto verrà poi affidato ad un corriere e verrà spedito entro un intervallo di 24/48 ore lavorative.

2. Specifiche della realtà di interesse

Si tratta della gestione di una piattaforma online sulla quale è possibile acquistare o visionare i prodotti di interesse. Negli ultimi anni le piattaforme di e-commerce sono diventate particolarmente utili in quanto la reperibilità di un negozio fisico è particolarmente limitata.

L'utente avrà la possibilità di scegliere i prodotti a cui è interessato tramite la ricerca per categorie.

PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

Per categorie si intendono i diversi tipi di strumenti musicali e accessori.

Per strumenti musicali intendiamo le diverse tipologie di essi:

- STRUMENTI A CORDE: basso (4 corde, 5 corde, acustico), chitarra (acustica, semi-acustica, classica, elettrica), ukulele.
- TASTIERE e PIANOFORTI: synth, tastiere, pianoforti (digitali, elettrici, piano, pianoforte classico), organi (elettrici, classici).
- PERCUSSIONI: set batteria (acustica, elettronica), set percussioni.
- STRUMENTI A FIATO: flauti, ottoni, tromboni.
- ATTREZZATURA: custodie, amplificatori, microfono, cavi.
- ARCHI: violini, viole, violoncelli, contrabbassi.
- SOFTWARE: daw, software educativi, librerie musicali.

3. Glossario dei termini

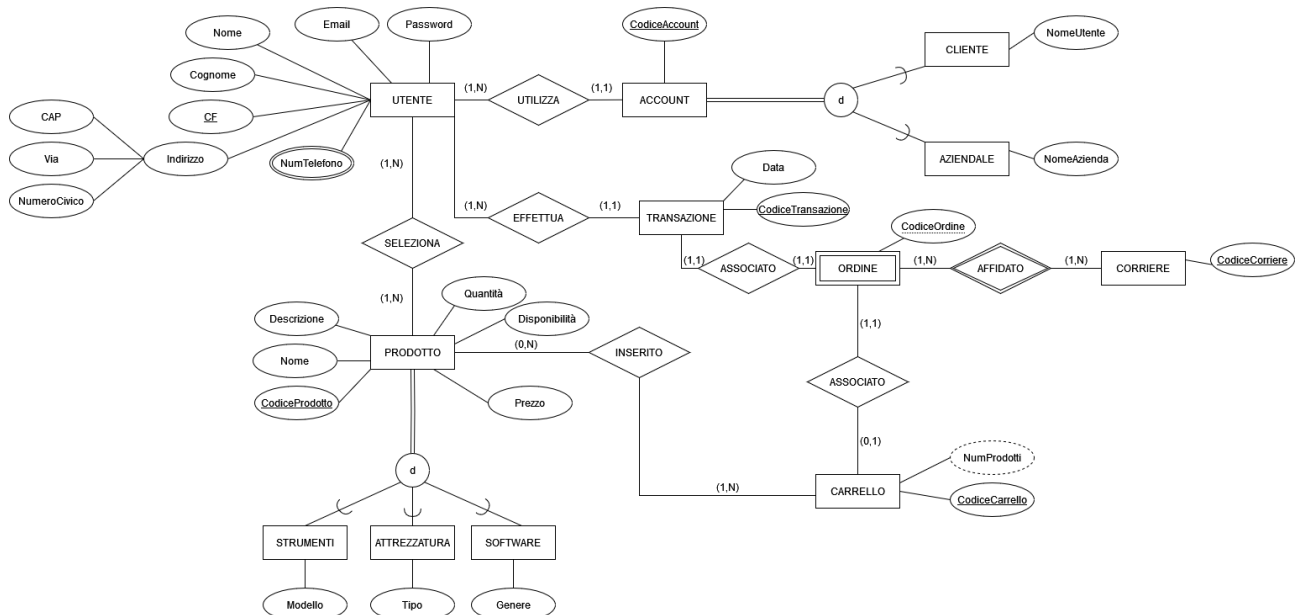
Termini	Significato
Username	Nome dell'utente definito alla creazione del profilo utente
Dati anagrafici	Nome, cognome, residenza, data di nascita, codice fiscale dell'utente
E-mail	Indirizzo di posta elettronica che l'utente inserisce al momento della creazione del proprio profilo utente
Password	Parola chiave (univoca?) scelta dall'utente al momento della creazione del profilo
Produttore	Chi ha prodotto un determinato strumento, software o accessorio.
Carrello	Lista degli elementi che l'utente desidera acquistare
Articolo	Strumento che deve essere acquistato/visionato
Account Cliente	Identità dell'utente, all'interno dell'e-commerce
Pagamento	Transazione eseguita dall'utente

PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

Account Aziendale	Identità dell'utente all'interno dell'e-commerce
Ordine	Articolo che viene acquistato dall'utente

4. Progettazione concettuale della base di dati Schema EER

Procedendo con la progettazione concettuale della base di dati, si ottiene il seguente schema EER:



5. Dizionario delle entità

Legenda: **sotto-entità**, **attributo multivalore**, **attributo composto**, **attributo ridondante**, **entità debole**, **chiave candidata**.

Dizionario delle entità

Entità	Descrizione	Attributi	Identificatore
Utente	Colui che utilizza l'account	Nome Cognome E-mail Password NumTelefono Indirizzo	CF
Account	Profilo personale creato in fase di registrazione	/	CodiceAccount
Prodotto	Oggetto osservato/acquistato dall'utente	Descrizione Nome Quantità Disponibilità Prezzo	CodiceProdotto
Ciente	Tipo di utente che acquista il prodotto	NomeUtente	/
Aziendale	Tipo di utente che vende il prodotto	NomeAzienda	/

PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

Strumenti	Specializzazione del prodotto acquistato	Modello	/
Attrezzatura	Specializzazione del prodotto acquistato	Tipo	/
Software	Specializzazione del prodotto acquistato	Genere	/
Carrello	Contenitore dei prodotti che l'utente osserva	NumProdotti	CodiceCarrello
Transazione	Mezzo per l'acquisto del contenuto dei prodotti	Data	CodiceTransazione
Ordine	Insieme dei prodotti acquistati dall'utente	CodiceOrdine	CodiceOrdine
Corriere	Colui che effettua il trasporto e la consegna dell'ordine	/	CodiceCorriere

6. Dizionario delle relazioni.

Relazione	Descrizione	Entità Coinvolte	Attributi
Utilizza	Un utente utilizza un account	Utente (1, N) Account (1, 1)	/
Seleziona	Un utente seleziona il prodotto	Utente (1, N) Prodotto (1, N)	/
Inserito	Il prodotto viene inserito in un carrello	Prodotto (0, N) Carrello (1, N)	/
Associato	L'ordine viene associato ad un carrello	Carrello (0, 1) Ordine (1, 1)	/
Associato	L'ordine viene associato ad una transazione	Ordine (1, 1) Transazione (1, 1)	/
Effettua	L'utente procede all'acquisto effettuando una transazione	Utente (1, N) Transazione (1, 1)	/
Affidato	Dopo che l'utente effettua una transazione, l'ordine viene affidato ad un corriere	Ordine (1, N) Corriere (1, N)	/

7. Definizione delle procedure per la gestione della base di dati

Tavola dei volumi

Definiamo di seguito la tavola dei volumi della base di dati.

Concetto	Tipo	Carico Applicativo
Utente	Entità	20
Account	Entità	20
Prodotto	Entità	150
Cliente	Entità	10
Aziendale	Entità	10
Strumenti	Entità	50
Attrezzatura	Entità	50
Software	Entità	50
Carrello	Entità	500
Transazione	Entità	40
Ordine	Entità	300
Corriere	Entità	1000
Utilizza	Relazione	600
Seleziona	Relazione	4500
Inserito	Relazione	75000

PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

Associato (Carrello)	Relazione	300
Associato (Transazione)	Relazione	40
Effettua	Relazione	30
Affidato	Relazione	5000

N.B: Una transazione è associata a un utente. Ipotizzando che un utente acquisti mediamente due prodotti, la transazione dev'essere calcolata in base a questo.

Ipotizzando che a ogni corriere vengano affidati 5 ordini, ogni corriere spedisce 5000 ordini.

Tavola delle operazioni

Definiamo di seguito la tavola delle operazioni per la gestione dei dati memorizzati nella base di dati.

Operazione	Tipo	Frequenza
1. Creare un nuovo account	I	1/mese
2. Selezionare un prodotto	I	300/mese
3. Acquistare un prodotto	I	2/mese
4. Creare un nuovo ordine	B	5/giorno
5. Affidare l'ordine a un corriere	B	5/giorno
6. Trovare il numero di strumenti acquistati	I	1/mese
7. Contare numero prodotti	B	1/giorno

PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

8. Aggiornare disponibilità prodotti	I	30/mese
9. Inserire descrizione prodotto	I	4/mese
10. Aggiornare password account	I	2/anno
11. Contare il numero di ordini affidato al corriere	B	5/giorno

8. Progettazione logica

Analisi delle ridondanze

Il dato ridondante è l'attributo "NumProdotti" dell'entità Carrello. Infatti, sarebbe possibile ottenere il numero di prodotti attraverso il conteggio della quantità dei prodotti dell'entità "Prodotto". Supponendo che l'attributo abbia un peso di 2 byte, essendo un normale intero, e considerato che il volume dell'entità Carrello è uguale a 500, il dato andrebbe ad occupare uno spazio totale di circa 1000 byte. Per decidere se mantenere o meno il dato ridondante è necessario calcolare, per le operazioni che lo coinvolgono, la differenza nel numero di accessi con e senza quest'ultimo.

Tavola degli accessi

Operazione 3: Acquistare un prodotto.

Calcolo con ridondanza				Calcolo senza ridondanza			
Tabella	Tipo	Accessi	Tipo Accessi	Tabella	Tipo	Accessi	Tipo Accessi
Prodotto	E	1	L	Ordine	E	1	S
Prodotto	E	1	S	Inserito	R	1	S
Inserito	R	1	L	Transazione	E	1	S

PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

Carrello	E	1	L	Prodotto	E	1	L
Carrello	E	1	S	Prodotto	E	1	S
Associato	R	1	L				
Ordine	E	1	L				
Ordine	E	1	S				
Associato	R	1	L				
Transazione	E	1	S				
Totale		$6L + (4S * 2) = 14 * 2 / \text{mese} = \mathbf{28} / \text{mese}$				$1L + (4S * 2) = 9 * 2 / \text{mese} = \mathbf{18} / \text{mese}$	

Operazione 6: Trovare il numero di strumenti acquistati.

Calcolo con ridondanza				Calcolo senza ridondanza			
Tabella	Tipo	Accessi	Tipo Accessi	Tabella	Tipo	Accessi	Tipo Accessi
Prodotto	E	1	L	Prodotto	E	1	L
Inserito	R	1	L	Inserito	R	1	L
Carrello	E	1	L	Carrello	E	1	L
Associato	R	1	L				
Ordine	E	1	L				
Totale		$5L = \mathbf{5} / \text{mese}$		$3L = \mathbf{3} / \text{mese}$			

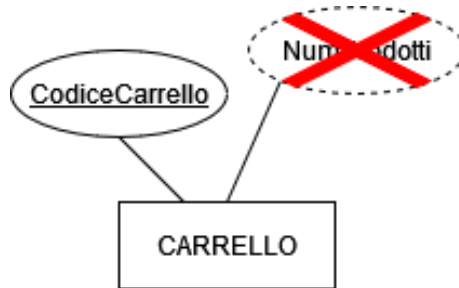
Operazione 8: Aggiornare disponibilità prodotti.

Calcolo con ridondanza				Calcolo senza ridondanza			
Tabella	Tipo	Accessi	Tipo Accessi	Tabella	Tipo	Accessi	Tipo Accessi
Prodotto	E	1	L	Prodotto	E	1	L
Prodotto	E	1	S	Prodotto	E	1	S
Inserito	R	1	L				
Carrello	E	1	L				
Carrello	E	1	S				
Totale		$3L + (2S * 2) = 7 * 30 / \text{mese} = \mathbf{210} / \text{mese}$		$1L + (1S * 2) = 3 * 30 / \text{mese} = \mathbf{90} / \text{mese}$			

PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

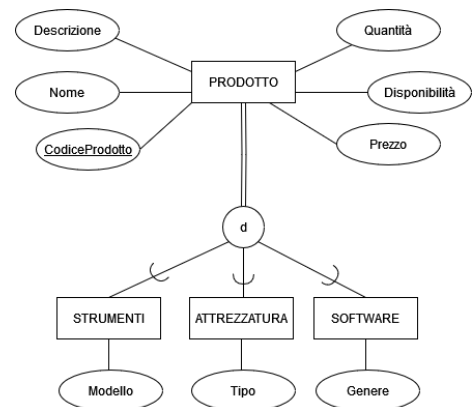
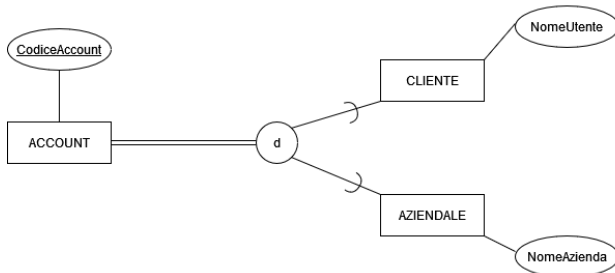
Totale accessi con ridondanza	Totale accessi senza ridondanza
$28+5+210=$ 243 accessi + 1000 byte	$18+3+90=$ 111 accessi

Dato il minor numero di accessi e lo spreco di spazio pari a 1000 byte, è più efficiente scegliere di non mantenere il dato ridondante "NumProdotti".



Eliminazione delle gerarchie

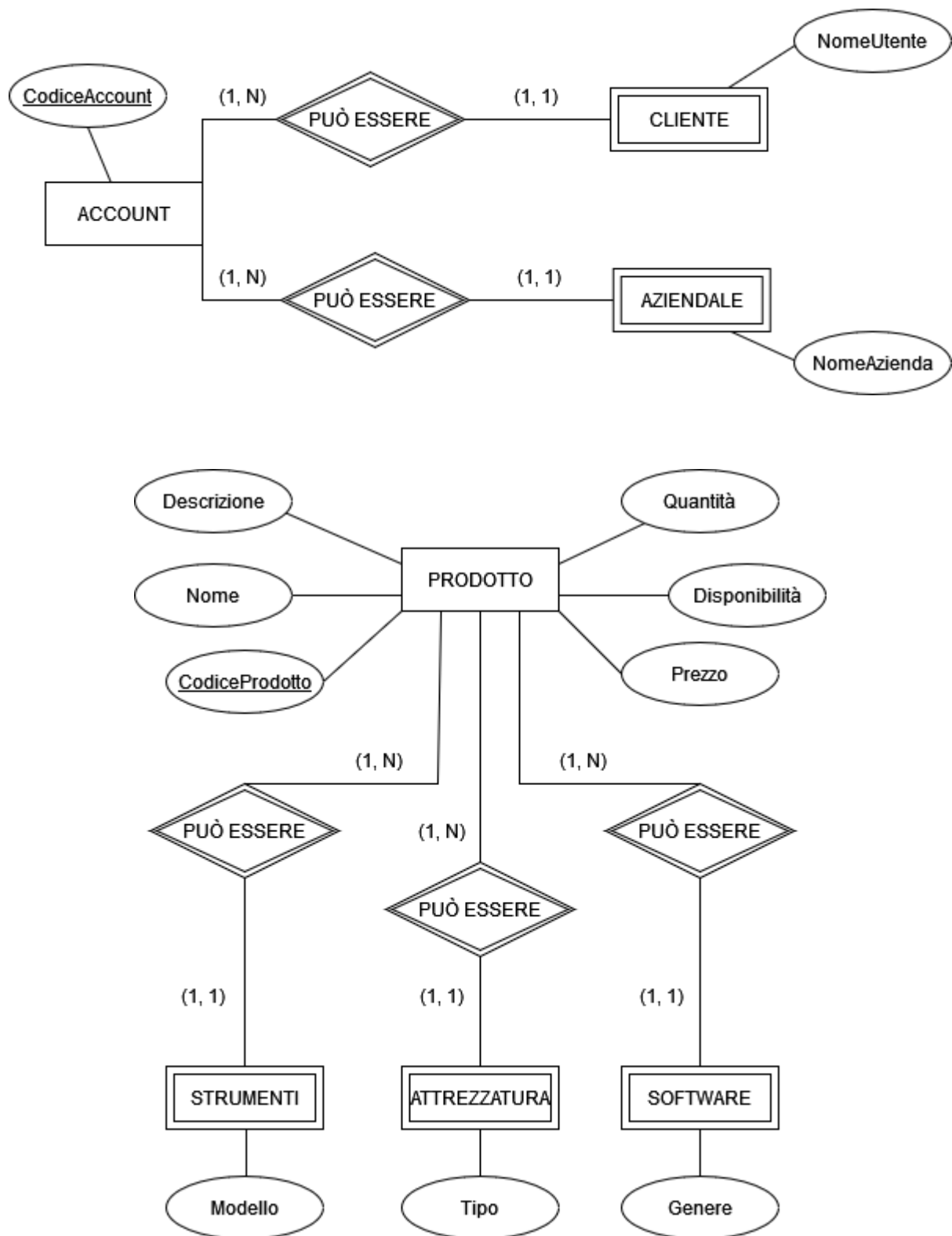
Nello schema inizialmente elaborato, sono presenti le seguenti specializzazioni delle entità "Account" e "Prodotto":



In questa fase di progettazione logica, è necessario individuare un metodo efficace di ristrutturazione che permetta l'eliminazione di queste gerarchie. La scelta effettuata è la creazione di nuove relazioni, che rendono le entità figlie nuove entità deboli. Ciò è necessario affinché si preservi la separazione dei concetti e si evitino attributi NULL.

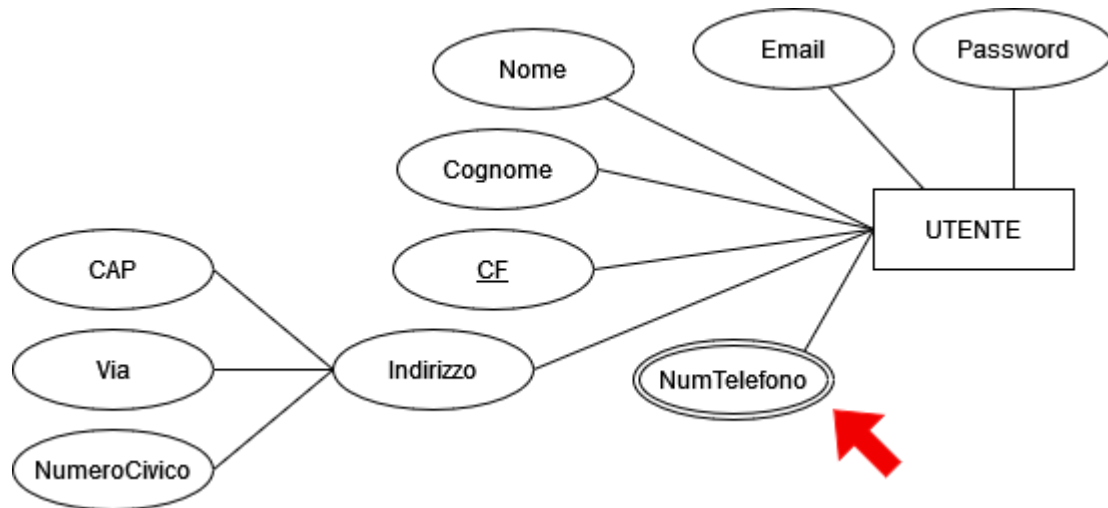
PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

Ristrutturiamo quindi come segue:



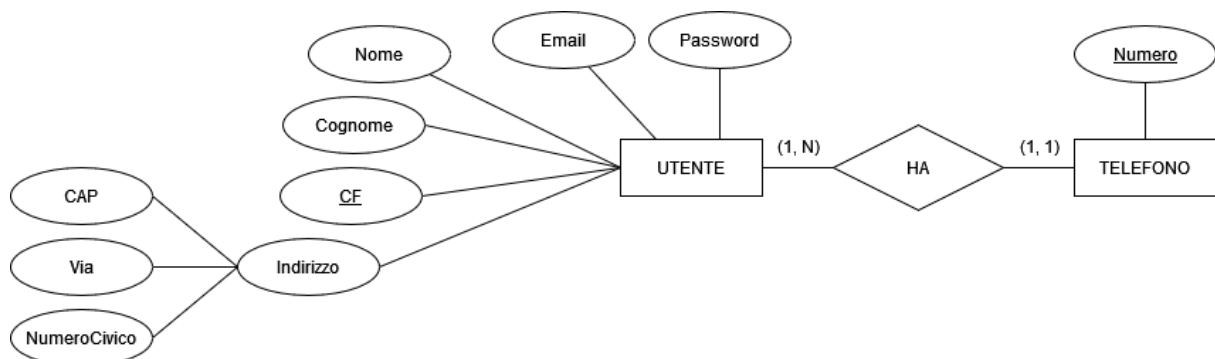
Eliminazione dell'attributo multivalore

Nello schema inizialmente elaborato, compare un attributo multivalore:



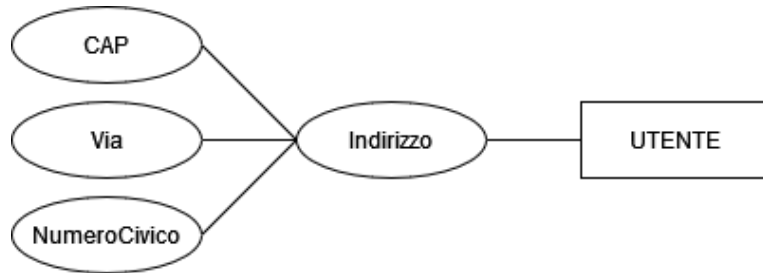
Tale forma di attributo va risolto in maniera differente in fase di progettazione logica.

Si sceglie quindi di definire una nuova entità "Telefono", in relazione con l'entità "Utente":

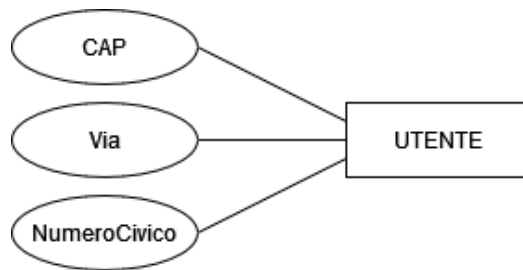


Eliminazione dell'attributo composto

Nello schema inizialmente elaborato, compare un attributo composto:



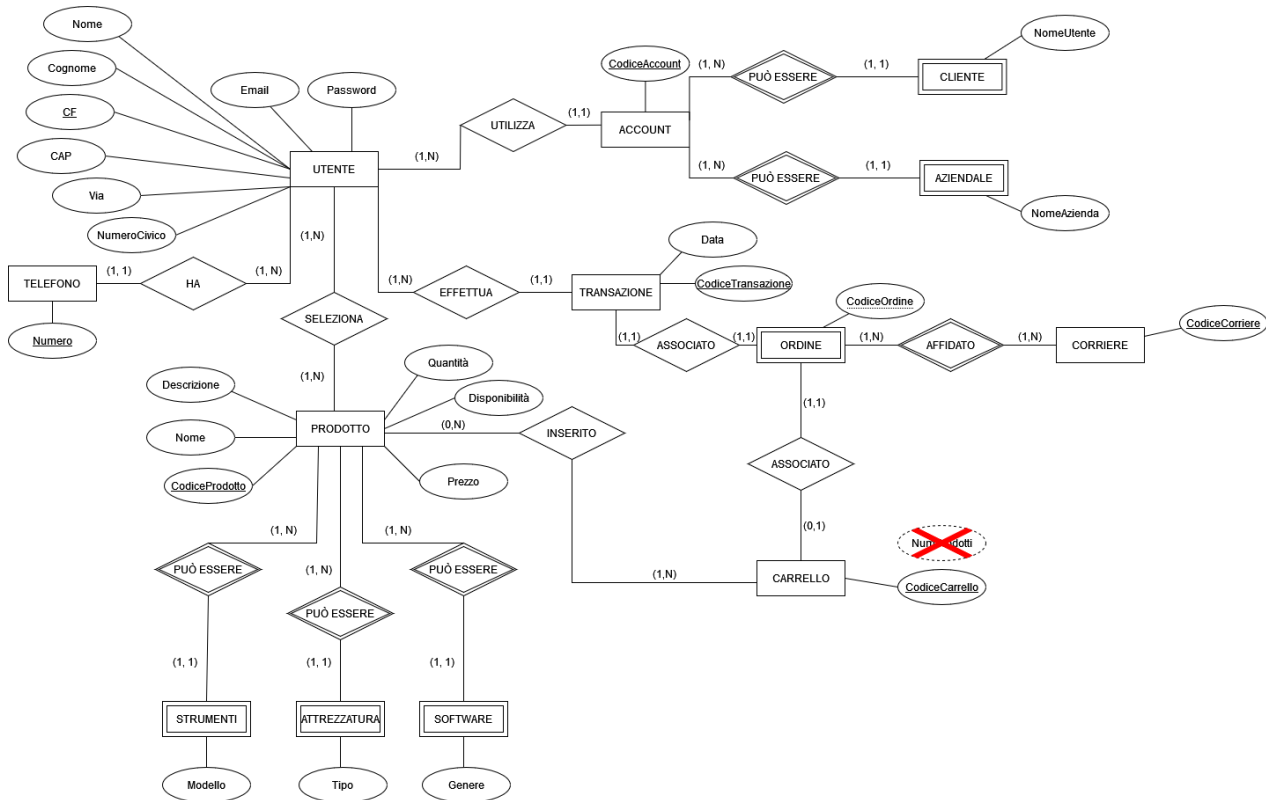
Tale forma di attributo va risolto associando direttamente gli attributi componenti all'entità a cui è associato l'attributo:



PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

Schema EER ristrutturato

Al termine della fase di ristrutturazione, lo schema EER completo che ne deriva è il seguente:



9. Schema relazionale

Si procede al mapping della base di dati:

Utente (CF, Nome, Cognome, CAP, Via, NumeroCivico, Email, Password)

Account (CodiceAccount, Utente.CF ↑)

Cliente (NomeUtente, Account.CodiceAccount ↑)

Aziendale (NomeAzienda, Account.CodiceAccount ↑)

Telefono (Numero, Utente.CF ↑)

Seleziona (Utente.CF ↑, Prodotto.CodiceProdotto ↑)

PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

Prodotto(CodiceProdotto, Nome, Descrizione, Quantità, Disponibilità, Prezzo)

Strumenti (Modello, Prodotto.CodiceProdotto ↑)

Attrezzatura (Tipo, Prodotto.CodiceProdotto ↑)

Software (Genere, Prodotto.CodiceProdotto ↑)

Inserito (Prodotto.CodiceProdotto ↑, Carrello.CodiceCarrello ↑)

Carrello (CodiceCarrello)

Corriere (CodiceCorriere)

Ordine (Carrello.CodiceCarrello ↑, CodiceOrdine,
Corriere.CodiceCorriere ↑)

Transazione(CodiceTransazione, Data, Ordine.CodiceOrdine ↑,
Ordine.Corriere.CodiceCorriere ↑, Utente.CF ↑)

Normalizzazione

Il database si presenta già normalizzato.

È infatti in prima forma normale in quanto tutti gli attributi sono atomici dopo la ristrutturazione (è stato infatti eliminato l'attributo multivalore "NumTelefono" nell'entità Utente e l'attributo composto "Indirizzo"). È in seconda forma normale perché, oltre ad essere già in 1NF, quando è presente una chiave primaria composta da più attributi tutte le dipendenze funzionali che la riguardano sono piene e non parziali. È in terza forma normale perché, oltre ad essere già in 2NF, in tutte le tabelle non sono presenti dipendenze transitive fra attributi non chiave e la chiave primaria.

10. Realizzazione della base di dati con MySQL

Di seguito il listato delle istruzioni MySQL per la creazione della base di dati:

```
1 • DROP database IF exists strumenti;
2 • CREATE database strumenti;
3 • use strumenti;
4
5 • CREATE TABLE Utente(
6     cf VARCHAR(16) NOT NULL,
7     nome VARCHAR(15) NOT NULL,
8     cognome VARCHAR(15) NOT NULL,
9     cap INT NOT NULL,
10    via VARCHAR(20) NOT NULL,
11    numeroCivico INT NOT NULL,
12    email VARCHAR(20) NOT NULL,
13    pw VARCHAR(20) NOT NULL,
14    PRIMARY KEY(cf)
15 );
16
17 • CREATE TABLE Account(
18     codAccount VARCHAR(10) NOT NULL,
19     cfCodAccount VARCHAR(16) NOT NULL,
20     PRIMARY KEY (codAccount),
21     FOREIGN KEY (cfCodAccount) References Utente(cf)
22 );
23
24 • CREATE TABLE Cliente (
25     nomeUtente VARCHAR(20) NOT NULL,
26     codUtenteCliente VARCHAR(10) NOT NULL,
27     PRIMARY KEY (codUtenteCliente),
28     FOREIGN KEY (codUtenteCliente) REFERENCES Account(codAccount)
29 );
```

PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

```
30
31 • ⊖ CREATE TABLE Aziendale(
32     nomeAzienda VARCHAR(10) NOT NULL,
33     codAccountAzienda VARCHAR(10) NOT NULL,
34     PRIMARY KEY (codAccountAzienda),
35     FOREIGN KEY (codAccountAzienda) references Account (codAccount)
36 );
37
38 • ⊖ CREATE TABLE Telefono(
39     numero INT NOT NULL,
40     codCF VARCHAR(16) NOT NULL,
41     PRIMARY KEY (numero),
42     FOREIGN KEY (codCF) references Utente(cf)
43 );
44
45 • ⊖ CREATE TABLE Prodotto (
46     CodiceProdotto INT NOT NULL,
47     Nome VARCHAR(50) NOT NULL,
48     Descrizione VARCHAR(255) NOT NULL,
49     Quantità INT NOT NULL,
50     Disponibilità VARCHAR(40) NOT NULL,
51     Prezzo DECIMAL(10,2) NOT NULL,
52     PRIMARY KEY (CodiceProdotto)
53 );
54
```

PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

```
55 • ○ CREATE TABLE Selezione (  
56     UtenteCF VARCHAR(16) NOT NULL,  
57     CodiceProdotto INT NOT NULL,  
58     FOREIGN KEY (UtenteCF) REFERENCES Utente(CF),  
59     FOREIGN KEY (CodiceProdotto) REFERENCES Prodotto(CodiceProdotto),  
60     PRIMARY KEY (UtenteCF, CodiceProdotto)  
61 );  
62  
63  
64 • ○ CREATE TABLE Strumenti (  
65     Modello VARCHAR(50) NOT NULL,  
66     CodiceProdotto INT NOT NULL,  
67     FOREIGN KEY (CodiceProdotto) REFERENCES Prodotto(CodiceProdotto),  
68     PRIMARY KEY (CodiceProdotto)  
69 );  
70  
71 • ○ CREATE TABLE Attrezzatura (  
72     Tipo VARCHAR(50) NOT NULL,  
73     CodiceProdotto INT NOT NULL,  
74     FOREIGN KEY (CodiceProdotto) REFERENCES Prodotto(CodiceProdotto),  
75     PRIMARY KEY (CodiceProdotto)  
76 );  
77
```

PROGETTO BASI DI DATI
E – COMMERCE DI STRUMENTI MUSICALI

```
78 • ○ CREATE TABLE Software (  
79     Genere VARCHAR(50) NOT NULL,  
80     CodiceProdotto INT NOT NULL,  
81     FOREIGN KEY (CodiceProdotto) REFERENCES Prodotto(CodiceProdotto),  
82     PRIMARY KEY (CodiceProdotto)  
83 );  
84  
85 • ○ CREATE TABLE Carrello (  
86     CodiceCarrello INT NOT NULL,  
87     PRIMARY KEY (CodiceCarrello)  
88 );  
89  
90 • ○ CREATE TABLE Inserito (  
91     CodiceProdotto INT NOT NULL,  
92     CodiceCarrello INT NOT NULL,  
93     FOREIGN KEY (CodiceCarrello) REFERENCES Carrello(CodiceCarrello),  
94     FOREIGN KEY (CodiceProdotto) REFERENCES Prodotto(CodiceProdotto),  
95     PRIMARY KEY (CodiceProdotto, CodiceCarrello)  
96 );  
97  
98 • ○ CREATE TABLE Corriere (  
99     CodiceCorriere INT NOT NULL,  
100     PRIMARY KEY (CodiceCorriere)  
101 );  
102  
103 • ○ CREATE TABLE Ordine (  
104     Carrello_CodiceCarrello INT NOT NULL,  
105     CodiceOrdine INT NOT NULL,  
106     Corriere_CodiceCorriere INT NOT NULL,  
107     FOREIGN KEY (Corriere_CodiceCorriere) REFERENCES Corriere(CodiceCorriere),  
108     FOREIGN KEY (Carrello_CodiceCarrello) REFERENCES Carrello(CodiceCarrello),  
109     PRIMARY KEY (CodiceOrdine, Corriere_CodiceCorriere)  
110 );  
111  
112 • ○ CREATE TABLE Transazione (  
113     CodiceTransazione INT NOT NULL,  
114     Data DATETIME NOT NULL,  
115     CodiceOrdine INT NOT NULL,  
116     CodCorriere INT NOT NULL,  
117     Utente_CF VARCHAR(16),  
118  
119     PRIMARY KEY (CodiceTransazione),  
120     FOREIGN KEY (CodiceOrdine, CodCorriere) REFERENCES Ordine(CodiceOrdine, Corriere_CodiceCorriere),  
121     FOREIGN KEY (Utente_CF) REFERENCES Utente(CF)  
122 );  
123
```

11. Implementazione query SQL

Operazione 1:

```
INSERT INTO Account (codAccount, cfCodAccount) VALUES (?, ?);
```

Operazione 2:

```
INSERT INTO Seleziona (UtenteCF, CodiceProdotto) VALUES (?, ?);
```

Operazione 3:

```
INSERT INTO Inserito (CodiceProdotto, codiceCarrello) VALUES (?, ?);
```

```
INSERT INTO Ordine (Carrello_CodiceCarrello, CodiceOrdine,  
Corriere_CodiceCorriere) VALUES (?, ?, ?);
```

Operazione 4:

```
INSERT INTO Carrello (CodiceCarrello) VALUES (?);
```

```
INSERT INTO Ordine (Carrello_CodiceCarrello, CodiceOrdine,  
Corriere_CodiceCorriere) VALUES (?, ?, ?);
```

Operazione 5:

```
UPDATE Ordine SET Corriere_CodiceCorriere = ?
```

```
WHERE CodiceOrdine = ?;
```

Operazione 6:

```
SELECT COUNT(s.CodiceProdotto) AS NumeroStrumentiAcquistati
```

```
FROM Seleziona s
```

```
JOIN Inserito i ON s.CodiceProdotto = i.CodiceProdotto
```

```
JOIN Carrello c ON i.CodiceCarrello = c.CodiceCarrello
```

```
JOIN Ordine o ON c.CodiceCarrello = o.Carrello_CodiceCarrello;
```

Operazione 7:

```
SELECT COUNT(*) AS NumProdotti  
FROM Prodotto;
```

Operazione 8:

```
UPDATE Prodotto SET Disponibilità = ?  
WHERE CodiceProdotto = ?;
```

Operazione 9:

```
UPDATE Prodotto SET Descrizione = ?  
WHERE CodiceProdotto = ?;
```

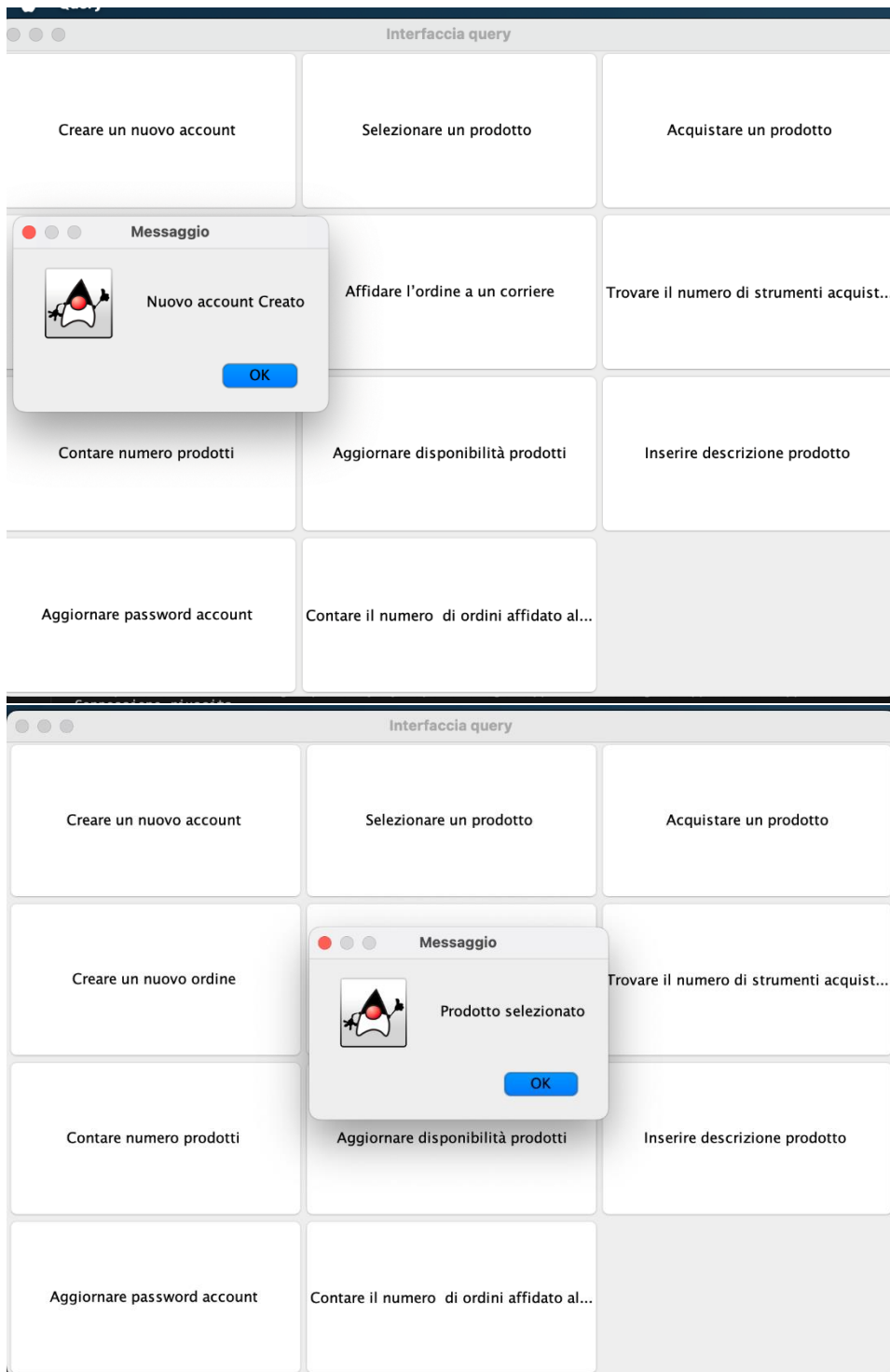
Operazione 10:

```
UPDATE Utente SET pw = ?  
WHERE cf =  
      (SELECT cfCodAccount FROM Account WHERE codAccount = ?);
```

Operazione 11:

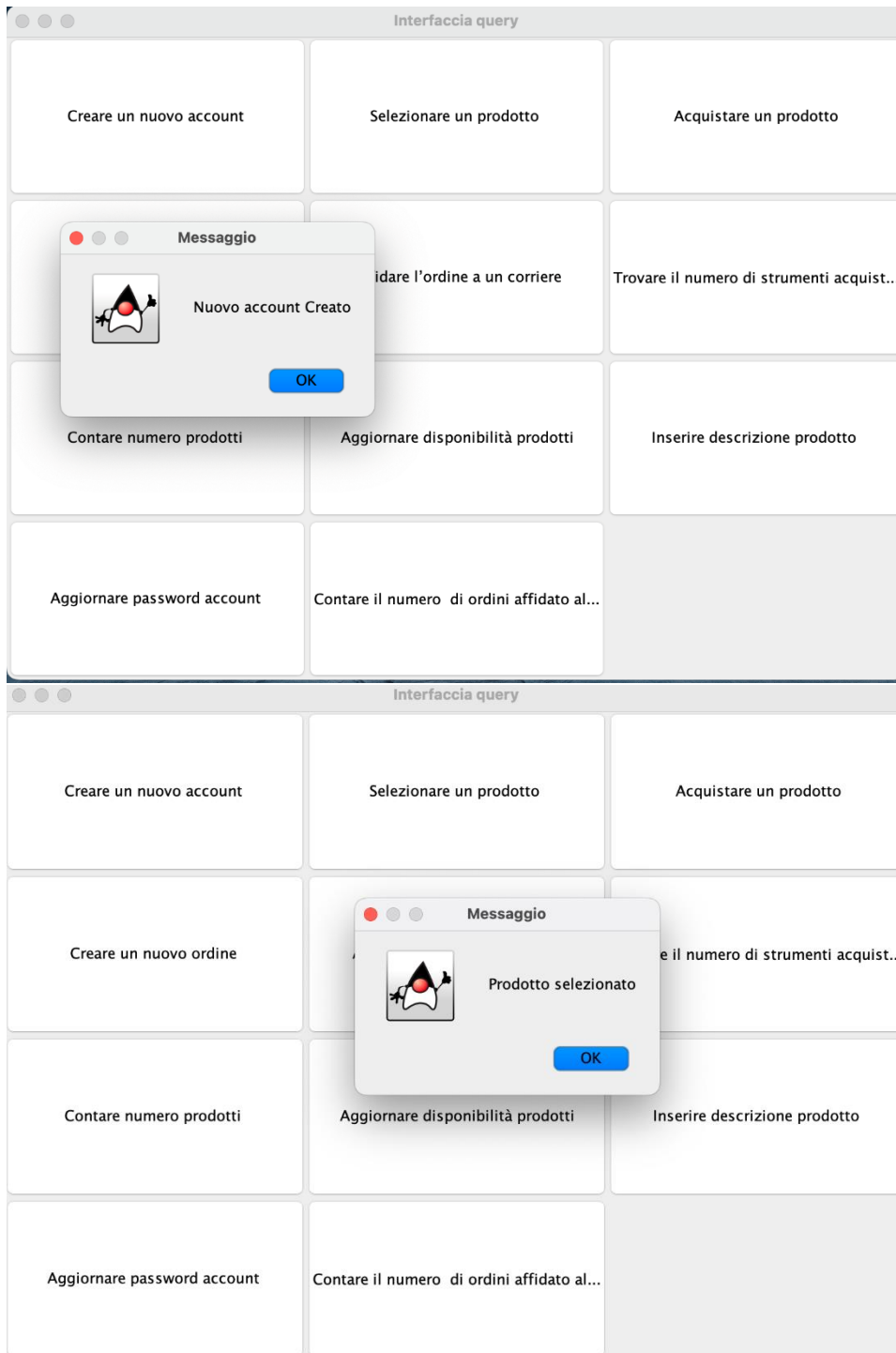
```
SELECT COUNT(*) AS NumeroOrdini  
FROM Ordine  
WHERE Corriere_CodiceCorriere = ?;
```

12. Test dell'applicazione Java



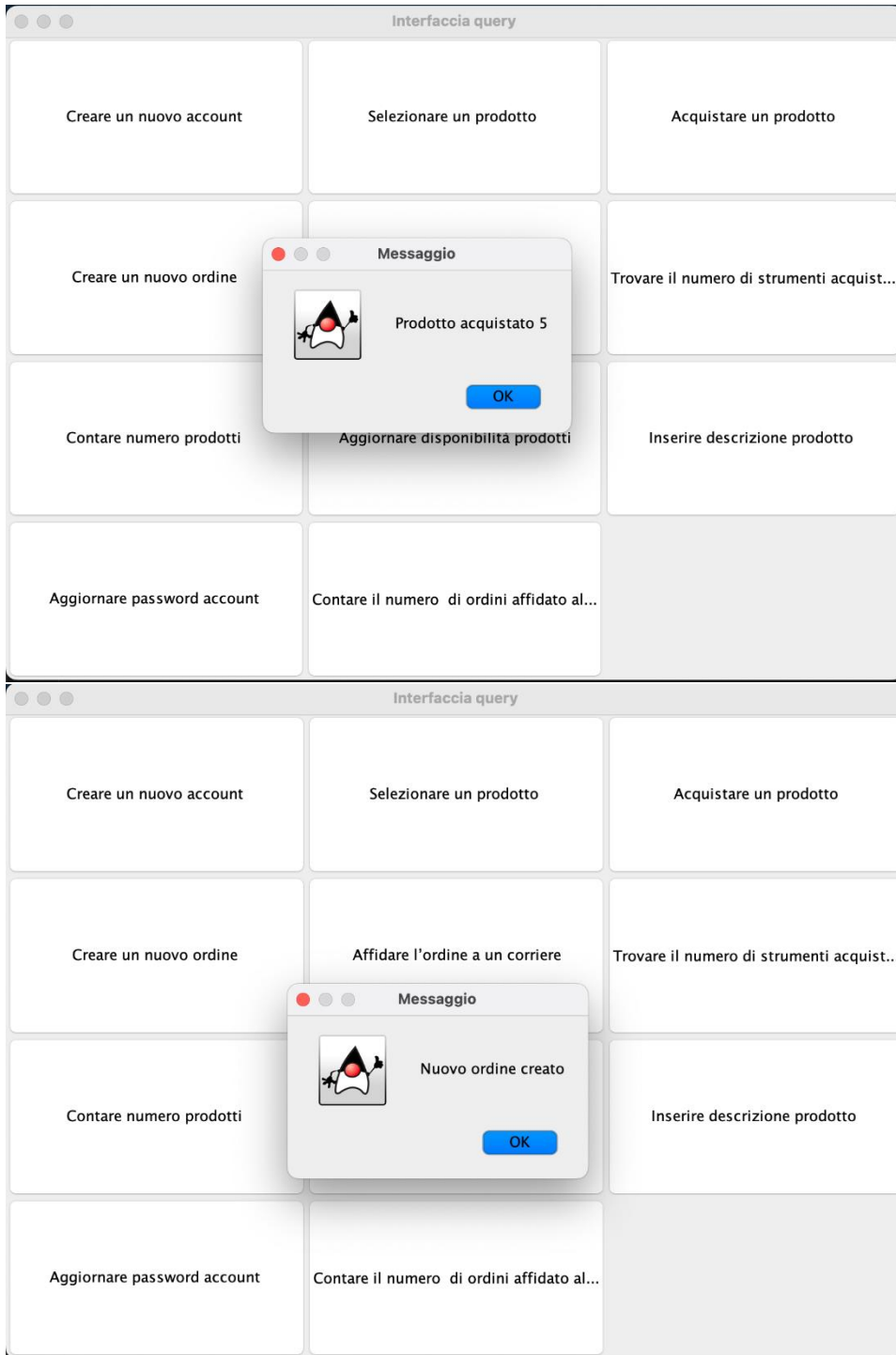
PROGETTO BASI DI DATI

E – COMMERCE DI STRUMENTI MUSICALI



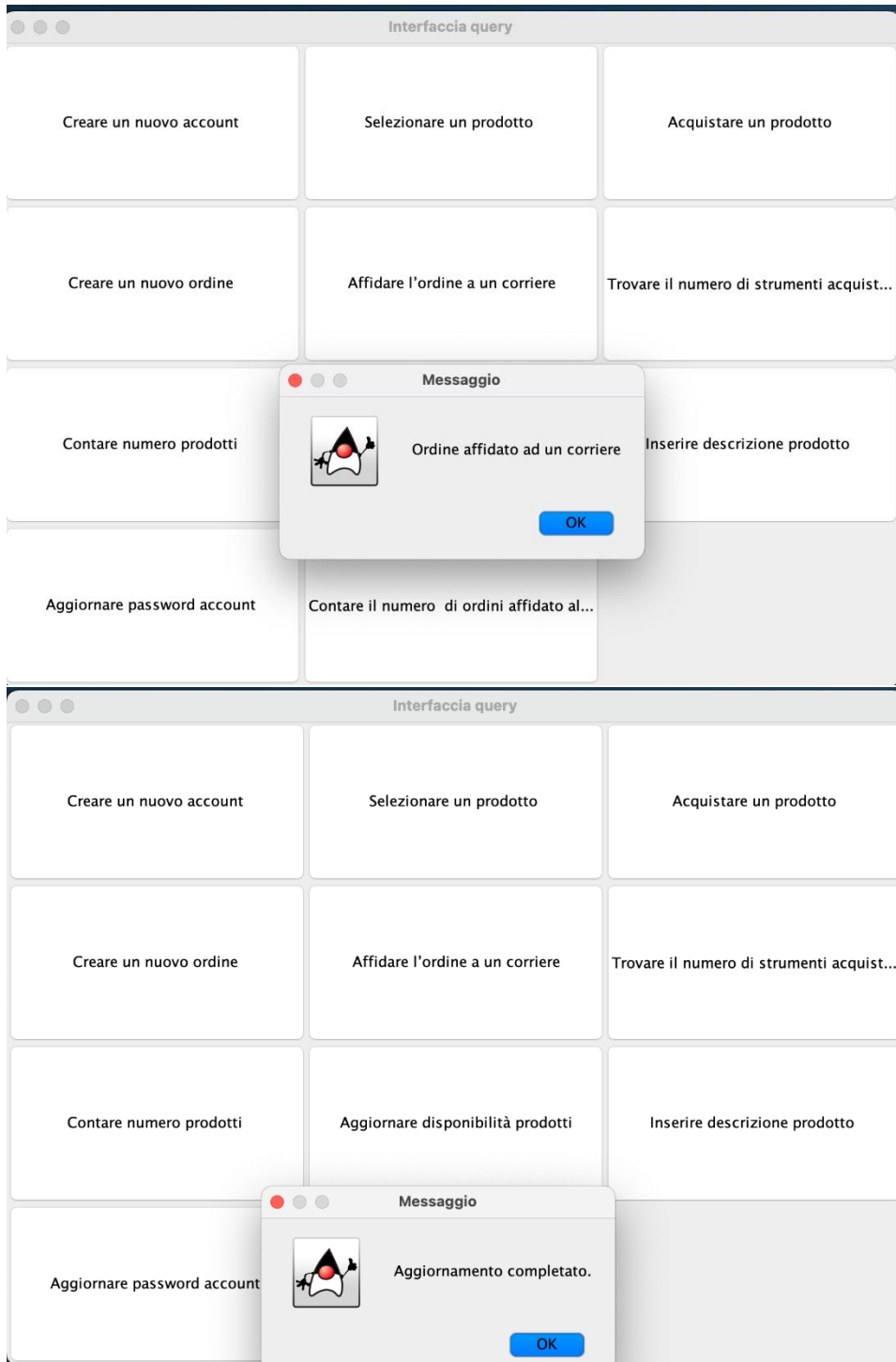
PROGETTO BASI DI DATI

E – COMMERCE DI STRUMENTI MUSICALI



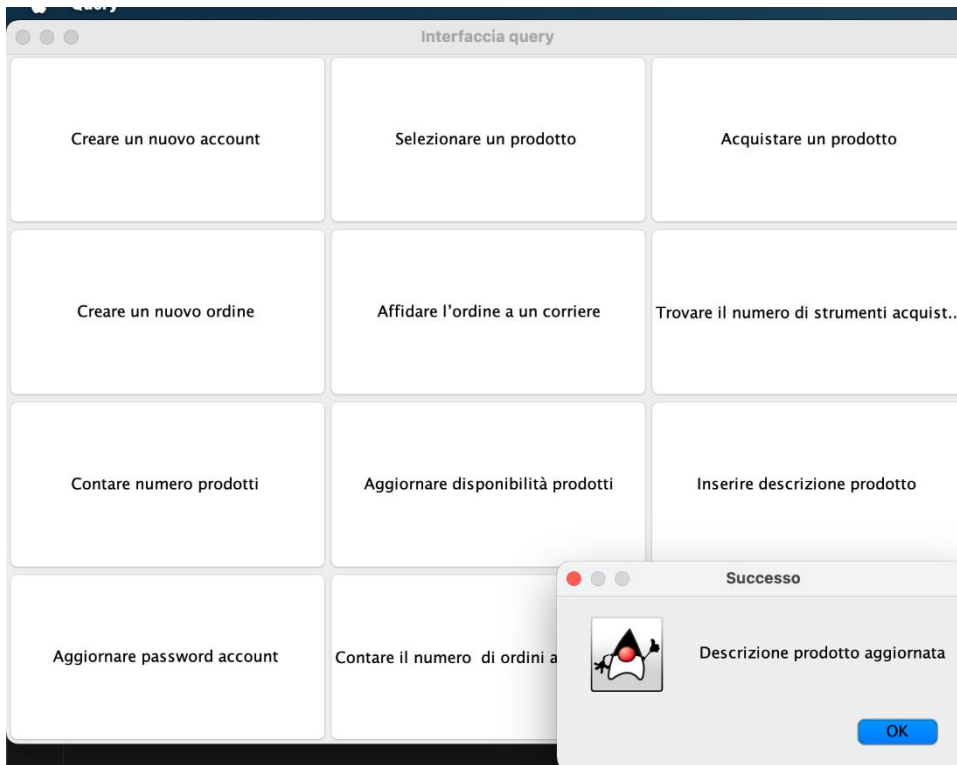
PROGETTO BASI DI DATI

E – COMMERCE DI STRUMENTI MUSICALI



PROGETTO BASI DI DATI

E – COMMERCE DI STRUMENTI MUSICALI



PROGETTO BASI DI DATI

E – COMMERCE DI STRUMENTI MUSICALI

