

2021-
2022

NEGOZIO DI STRUMENTI MUSICALI ONLINE
GIORGIO COLELLA 05946

1. Raccolta delle specifiche della realtà d'interesse

Si vuole progettare una base di dati per la gestione di un negozio online di strumenti musicali. Per prima cosa si vuole memorizzare il cliente identificato dal codice fiscale e fornito di nome, cognome, indirizzo, email, password, data di nascita, numero acquisti totali, nome della carta di credito e scadenza e numero/i di telefono.

Il cliente può scrivere recensioni relative ad un prodotto e di tali recensioni si vuole memorizzare il numero, la descrizione ed il voto.

Degli ordini effettuati si vuole memorizzare l'id che li identifica e la data. Dei pagamenti di tali ordini si vuole memorizzare un id che li identifica, il tipo, la data e l'importo.

Gli ordini hanno una spedizione di cui si vuole memorizzare il codice che li identifica, la data di partenza, la data di arrivo ed il costo.

I prodotti sono identificati da un codice ed hanno una marca, un colore, un prezzo ed una quantità disponibile. Inoltre essi si dividono non solo ma anche in batterie (di cui vogliamo sapere il numero di rullanti), chitarre elettriche (di cui vogliamo sapere il tipo di pickup ed il numero di corde) e tastiere (di cui vogliamo sapere il numero di stili selezionabili).

Specifiche della realtà d'interesse

La realtà che vado a rappresentare è un e-commerce classico di quella che è ormai l'era di Amazon ed Ebay.

Il tutto avviene in maniera sequenziale con la creazione di un ordine relativo ad un prodotto con tanto di spedizione e pagamento. Un cliente può inoltre effettuare delle recensioni per aiutare sia il venditore che altri clienti alla valutazione dei prodotti venduti.

I prodotti venduti sono diversi e non tutti sono elencati poiché solo di quelli presenti ci interessa conoscere delle particolari specifiche. Le batterie infatti vengono differenziate in base al numero di rullanti e le chitarre elettriche in base al tipo di pickup (es. single coil, humbucker, p90, ecc...) e al numero di corde e le tastiere vengono differenziate dalla quantità di stili che è possibile utilizzare.

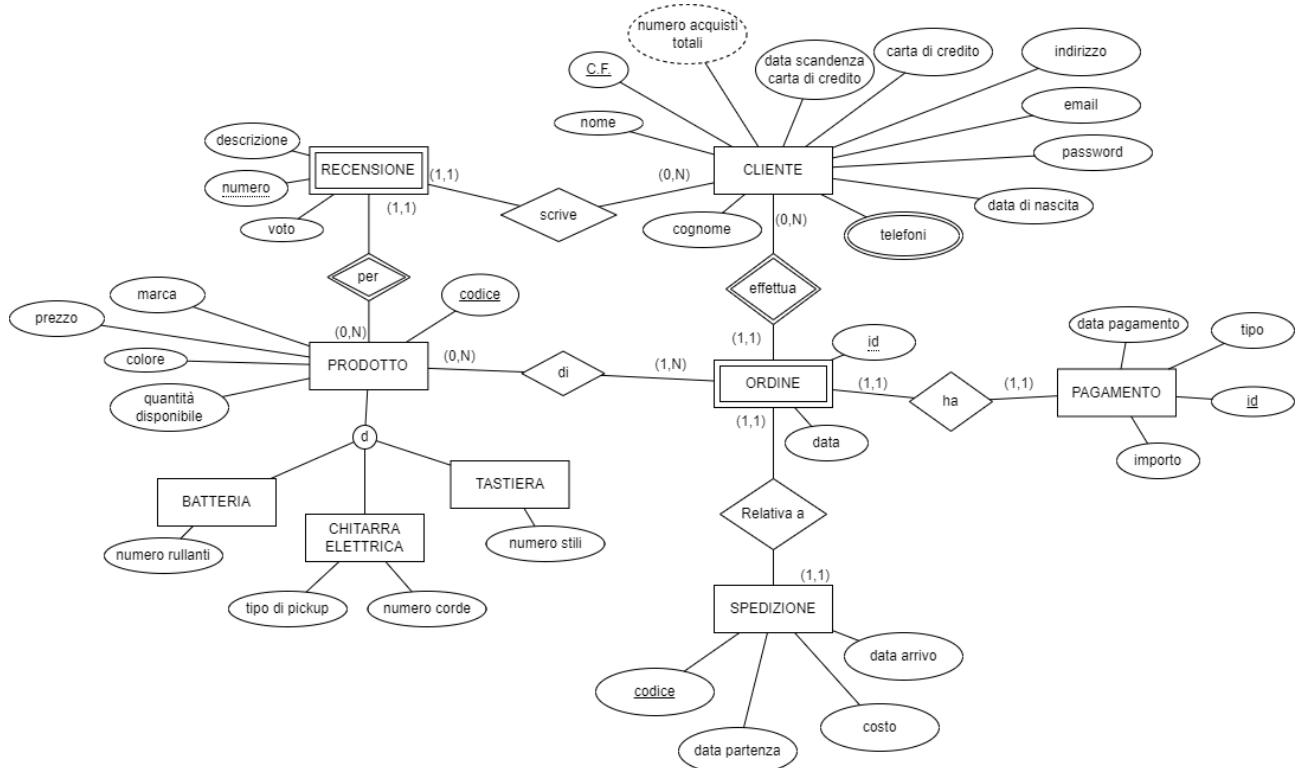
Glossario dei termini principali

Termine	Descrizione
Cliente	Cliente coinvolto nella realtà d'interesse
Ordine	L'elemento principale della realtà d'interesse che collega quasi tutte le entità tra di loro
Pagamento	Le informazioni riguardanti il pagamento

Prodotto	Tutte le informazioni relative ad un prodotto
Recensione	Informazioni relative alle recensioni che possono essere fornite dai clienti
Spedizione	L'insieme di informazioni riguardanti la spedizione
Batteria	L'insieme di informazioni riguardanti un tipo di prodotto
Chitarra Elettrica	L'insieme di informazioni riguardanti un tipo di prodotto
Tastiera	L'insieme di informazioni riguardanti un tipo di prodotto

2. Progettazione Concettuale

Nel mondo reale dati come la carta di credito non verrebbero salvati nel DB ma ci si affiderebbe ad un Payment Gateway (come PayPal). Qui è solo in ambito didattico/dimostrativo.



Dizionario delle entità

Legenda: **sotto-entità**, **attributo multivale**, **attributo ridondante**, **entità debole**, **chiave candidata**

Entità	Descrizione	Attributi	Identificatore
Cliente	Cliente coinvolto nella realtà d'interesse	Cf, indirizzo, email, password, nome, cognome, dataNascita, carta di credito, scadenza carta di credito, telefoni , numero acquisti totali	cf
Ordine	L'elemento principale della realtà d'interesse che collega quasi tutte le entità tra di loro	Id, data	id
Spedizione	L'insieme di informazioni riguardanti la spedizione	Codice, costo, data partenza, data arrivo	codice
Pagamento	Le informazioni riguardanti il pagamento	Id, importo, data, tipo	id
Prodotto	Tutte le informazioni relative ad un prodotto	Codice, marca, colore, quantità disponibile, prezzo	codice
Batteria	L'insieme di informazioni riguardanti un tipo di prodotto	Numero di rullanti	/
Chitarra Elettrica	L'insieme di informazioni riguardanti un tipo di prodotto	Numero di corde, tipo di pickup	/
Tastiera	L'insieme di informazioni riguardanti un tipo di prodotto	Numero di stili	/
Recensione	Informazioni relative alle recensioni che possono essere fornite dai clienti	Numero, descrizione, voto	numero

Dizionario delle relazioni

Relazione	Descrizione	Entità coinvolte	Attributi
Effettua	Un cliente effettua un ordine	cliente: (0, N) ordine: (1, 1)	/
Di	Un ordine è di un prodotto	ordine: (1, N) prodotto: (0, N)	/
Scrive	Il cliente scrive una recensione	recensione: (1, 1) cliente: (0, N)	/
Ha	Un ordine ha un pagamento	ordine: (1, 1) pagamento: (1, 1)	/
Relativa A	Una spedizione è relativa a un ordine	ordine: (1, 1) spedizione: (1, 1)	/
Per	Una recensione è per un prodotto	recensione: (1, 1) prodotto: (0, N)	/

Vincoli non esprimibili nello schema

- Il numero di ruotanti va da 3 a 7;
- I prezzi non superano i 15.000 euro;
- La spedizione è pagabile solo in contante e quindi non integrata in importo.

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

Tavola dei volumi

Concetto	Tipo	Carico Applicativo
Cliente	E	100
Ordine	E	400
Pagamento	E	400
Recensione	E	200
Prodotto	E	40
Batteria	E	5
Chitarra Elettrica	E	10
Tastiera	E	5

Spedizione	E	400
Effettua	R	400
Scrive	R	200
Per	R	200
Di	R	400
Ha	R	400
Relativo A	R	400

Tavola delle operazioni

Operazione	Tipo	Frequenza
1. Aggiungi un nuovo cliente	I	2/settimana
2. Aggiungi un nuovo ordine	I	5/settimana
3. Rimuovere una recensione	I	1/settimana
4. Selezionare il numero di acquisti totali di ogni cliente	I	3/mese
5. Selezionare tutte le informazioni delle chitarre elettriche con tipo di pickup Single Coil	I	1/settimana
6. Selezionare tutti i prodotti con un prezzo inferiore a 350 euro	I	10/settimana
7. Selezionare l'id di tutti gli ordini effettuati in data 20 Giugno 2019 che hanno un importo superiore a 100 euro e sono relativi a prodotti di colore blu	B	1/anno
8. Selezionare id, marca e voto di tutti i prodotti di colore bianco, con almeno un voto dal 7 in su e disponibili.	I	3/settimana
9. Selezionare l'id tutti gli ordini con data arrivo pari a 23 febbraio 2022 i cui prodotti ordinati non hanno una recensione	I	2/settimana
10. Rimuovere un ordine erroneo e non più valido	I	1/settimana

4. Progettazione Logica

ELIMINAZIONE DELL'ATTRIBUTO MULTIVALORE

Nello schema iniziale è presente l'attributo multivaleore telefoni nell'entità Cliente. In fase di progettazione logica questo attributo va rimosso e ristrutturato, trasformandolo in un'entità debole e poi collegandolo alla entità alla quale apparteneva tramite una relazione. Si otterrà quindi lo schema seguente:



Il dato ridondante è numero acquisti totali dell'entità Cliente, dato deducibile dalla relazione “Di” tra Ordine e Prodotto. Per decidere se mantenere o no tale attributo facciamo il calcolo degli accessi, relativi alle operazioni precedentemente elencate, con e senza tale attributo.

Notiamo inoltre che, essendo l'attributo di valore possibilmente inferiore a abbiamo bisogno di 1 byte per memorizzarlo. Contando che i Clienti sono in media 100 allora $1*100 = 100$ byte totali.

TAVOLA DEGLI ACCESSI

Valutiamo ora gli accessi delle operazioni 2, 4 e 10 con e senza l'attributo ridondante.

Calcolo con ridondanza – Operazione 2			
Tabella	Tipo	Accessi	Tipo
Ordine	E	1	S
Effettua	R	1	S
Ha	R	1	S
Pagamento	E	1	S
Di	R	1	S
Relativo A	R	1	S
Spedizione	E	1	S
Cliente	E	1	L
Cliente	E	1	S
TOTALE:		$(8S + 1L) = 17 * 5 = 85/\text{settimana} + 100 \text{ byte}$	

Calcolo senza ridondanza – Operazione 2			
Tabella	Tipo	Accessi	Tipo
Ordine	E	1	S
Effettua	R	1	S
Ha	R	1	S
Pagamento	E	1	S
Di	R	1	S
Relativo A	R	1	S
Spedizione	E	1	S
TOTALE:		$(7S) = 14 * 5 = 70/\text{settimana}$	

Calcolo con ridondanza – Operazione 4

Tabella	Tipo	Accessi	Tipo
Cliente	E	100	L
TOTALE:		$(100L) = 100*3= 300/\text{mese}= 75/\text{settimana}$ + 100 byte	

Calcolo senza ridondanza – Operazione 4

Tabella	Tipo	Accessi	Tipo
Cliente	E	100	L
Effettua	R	4*100	L
TOTALE:		$(500L) = 500*3= 1500/\text{mese}= 375/\text{settimana}$	

Calcolo con ridondanza – Operazione 10

Tabella	Tipo	Accessi	Tipo
Ordine	E	1	L
Ordine	E	1	S
Relativo A	R	1	L
Relativo A	R	1	S
Spedizione	E	1	L
Spedizione	E	1	S
Ha	R	1	L
Ha	R	1	S
Pagamento	E	1	L
Pagamento	E	1	S
Di	R	1	L

Di	R	1	S
Effettua	R	1	L
Effettua	R	1	S
Cliente	E	1	L
Cliente	E	1	S
TOTALE:		$(8S+8L) = 24 * 1 = 24/\text{settimana} + 100 \text{ byte}$	

Calcolo senza ridondanza – Operazione 10			
Tabella	Tipo	Accessi	Tipo
Ordine	E	1	L
Ordine	E	1	S
Relativo A	R	1	L
Relativo A	R	1	S
Spedizione	E	1	L
Spedizione	E	1	S
Ha	R	1	L
Ha	R	1	S
Pagamento	E	1	L
Pagamento	E	1	S
Di	R	1	L
Di	R	1	S
Effettua	R	1	L
Effettua	R	1	S
TOTALE:		$(7S+7L) = 21 * 1 = 21/\text{settimana}$	

Il totale degli accessi con attributo ridondante è 184/settimana + 100 byte.

Il totale degli accessi senza dato ridondante è 466/settimana.

Conviene quindi mantenere l'attributo ridondante.

Eliminazione delle gerarchie

La specializzazione che divide Prodotto nelle sue entità figlie Batteria, Chitarra Elettrica e Tastiera va eliminata nella fase di ristrutturazione. Ora, i metodi sono tre ma, essendo in partecipazione parziale, utilizzeremo la trasformazione in entità debole che permetterà di evitare diversi attributi nulli a discapito di un incremento di accessi all'interno delle operazioni relative alle entità figlie.

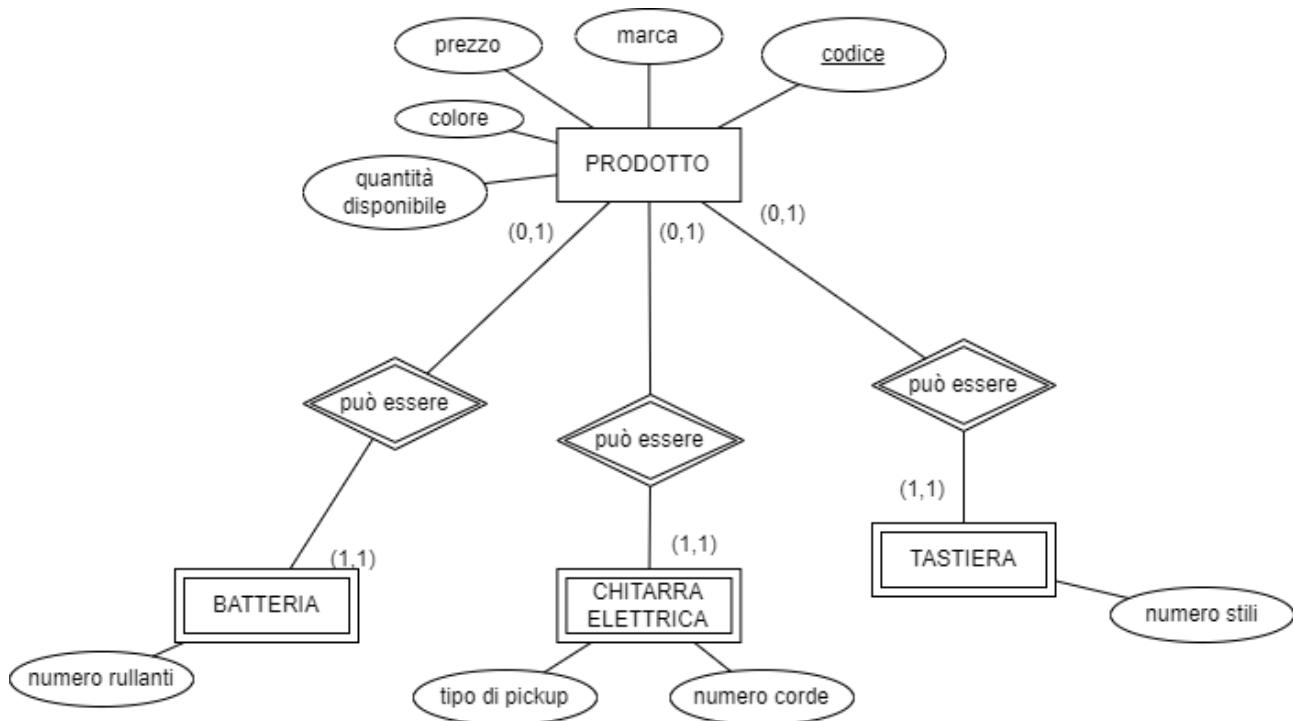
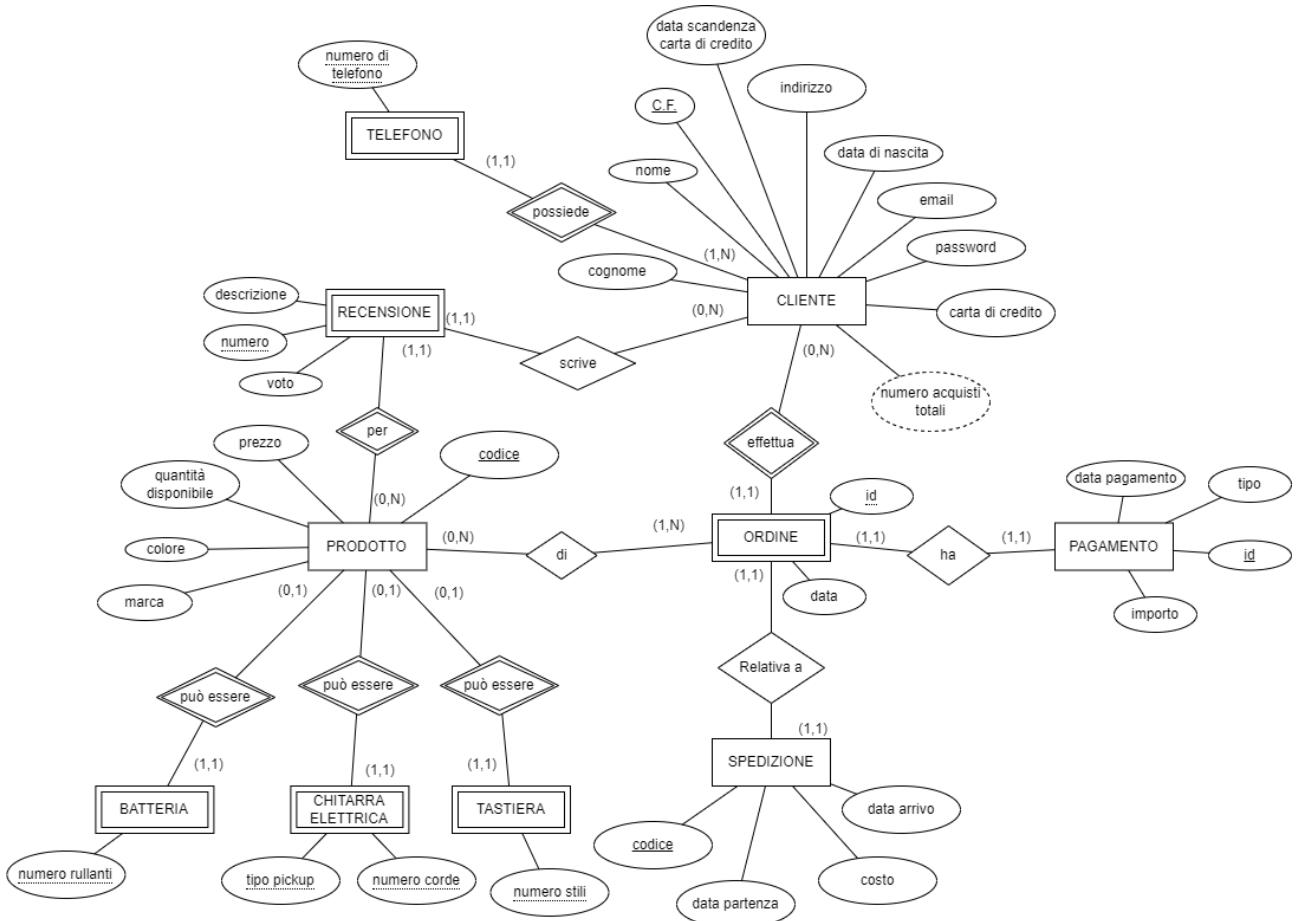


Diagramma ristrutturato



Schema relazionale

- ❖ **Cliente** (C.F., nome, cognome, dataNascita, indirizzo, email, password, carta di credito, scadenzaCarta, numero totale acquisti)
- ❖ **Ordine** (id, Cliente \uparrow , data, Spedizione \uparrow , Pagamento \uparrow)
- ❖ **Pagamento** (id, dataPagamento, tipo, importo)
- ❖ **Prodotto** (codice, prezzo, quantità disponibile, marca, colore)
- ❖ **Recensione** (numero, Prodotto \uparrow , descrizione, voto, Cliente \uparrow)
- ❖ **Spedizione** (codice, data partenza, data arrivo, costo)
- ❖ **Batteria** (numero rullanti, Prodotto \uparrow)
- ❖ **Chitarra Elettrica** (tipo di pickup, numero di corde, Prodotto \uparrow)
- ❖ **Tastiera** (numero di stili, Prodotto \uparrow)
- ❖ **Telefono** (numero di telefono, Cliente \uparrow)
- ❖ **Di** (Prodotto \uparrow , Ordine.id \uparrow , Ordine.Cliente.cf \uparrow)

Normalizzazione

Il database EStore si presenta già normalizzato.

- È in **prima forma normale** perché, una volta ristrutturato, lo schema EER non presenta né attributi composti, né attributi multivalore (è stato eliminato l'attributo telefono).
- È in **seconda forma normale** perché non esistono dipendenze parziali dalle diverse chiavi.
- È in **terza forma normale** perché non esistono dipendenze transitive dalle diverse chiavi.

5. Realizzazione Database con MySQL

Script per creare il database

```
1 •  use mysql;
2 •  DROP USER IF EXISTS 'theuser'@'localhost';
3 •  CREATE USER 'theuser'@'localhost' IDENTIFIED BY 'theuser';
4 •  DROP DATABASE IF EXISTS estore;
5 •  CREATE DATABASE estore;
6 •  GRANT ALL ON estore.* TO 'theuser'@'localhost';
7
8 •  USE estore;
9
10 •  DROP TABLE IF EXISTS Cliente;
11 •  CREATE TABLE Cliente(
12     cf CHAR(16) NOT NULL,
13     nome VARCHAR(20) NOT NULL,
14     cognome VARCHAR(20) NOT NULL,
15     dataNascita DATE NOT NULL,
16     indirizzo varchar(40) not null,
17     email varchar(70) not null,
18     passw varchar(40) not null,
19     cartadiCredito varchar(30) not null,
20     scadenzaCarta date not null,
21     numTotaleAcquisti int not null default 0,
22     PRIMARY KEY(cf)
23 );
24
```

```
25 • DROP TABLE IF EXISTS Pagamento;
26 • Ⓜ CREATE TABLE Pagamento(
27     id INT NOT NULL auto_increment,
28     dataPagamento DATE not null,
29     tipo varchar(20) not null,
30     importo double(7,2) not null,
31     PRIMARY KEY(id)
32 );
33
34 • DROP TABLE IF EXISTS Telefono;
35 • Ⓜ CREATE TABLE Telefono(
36     numTelefono char(10) not null,
37     idCliente CHAR(16) NOT NULL,
38     PRIMARY KEY(numTelefono,idCliente),
39     FOREIGN KEY(idCliente) REFERENCES Cliente(cf) ON UPDATE cascade ON DELETE cascade
40 );
41
42 • DROP TABLE IF EXISTS Spedizione;
43 • Ⓜ CREATE TABLE Spedizione(
44     codice int not null auto_increment,
45     dataPartenza date not null,
46     dataArrivo date,
47     costo double(5,2) not null,
48     primary key(codice)
49 );
50
```

```
51 •   DROP TABLE IF EXISTS Ordine;
52 •   CREATE TABLE Ordine(
53     id int auto_increment,
54     idCliente char(16) not null,
55     idPagamento int not null,
56     dataOrdine date not null,
57     idSpedizione int not null,
58     primary key(id,idCliente),
59     FOREIGN KEY(idCliente) REFERENCES Cliente(cf) ON UPDATE cascade ON DELETE cascade,
60     FOREIGN KEY(idSpedizione) REFERENCES Spedizione(codice) ON UPDATE cascade ON DELETE cascade,
61     FOREIGN KEY(idPagamento) REFERENCES Pagamento(id) ON UPDATE cascade ON DELETE cascade
62 );
63
64 •   DROP TABLE IF EXISTS Prodotto;
65 •   CREATE TABLE Prodotto(
66     codice int not null auto_increment,
67     prezzo double(7,2) not null,
68     quantitaDisponibile int not null default 0,
69     marca varchar(20) not null,
70     colore varchar(20) not null,
71     primary key(codice)
72 );
73
```

```

74 •   DROP TABLE IF EXISTS Recensione;
75 •   CREATE TABLE Recensione(
76     numero int not null auto_increment,
77     idProdotto int not null,
78     idCliente char(16) not null,
79     descrizione text(1000) not null,
80     voto enum('1','2','3','4','5','6','7','8','9','10') not null,
81     primary key(numero,idProdotto),
82     FOREIGN KEY(idProdotto) REFERENCES Prodotto(codice) ON UPDATE cascade ON DELETE cascade,
83     FOREIGN KEY(idCliente) REFERENCES Cliente(cf) ON UPDATE cascade ON DELETE cascade
84 );
85
86 •   DROP TABLE IF EXISTS Batteria;
87 •   CREATE TABLE Batteria(
88     numeroRullanti int unsigned not null,
89     idProdotto int not null,
90     primary key(numeroRullanti,idProdotto),
91     FOREIGN KEY(idProdotto) REFERENCES Prodotto(codice) ON UPDATE cascade ON DELETE cascade
92 );
93
94 •   DROP TABLE IF EXISTS ChitarraElettrica;
95 •   CREATE TABLE ChitarraElettrica(
96     tipoPickup varchar(30) not null,
97     numeroCorde int unsigned not null,
98     idProdotto int not null,
99     primary key(tipoPickup,numeroCorde,idProdotto),
100    FOREIGN KEY(idProdotto) REFERENCES Prodotto(codice) ON UPDATE cascade ON DELETE cascade
101 );
102
102
103 •   DROP TABLE IF EXISTS Tastiera;
104 •   CREATE TABLE Tastiera(
105     numeroStili int unsigned not null,
106     idProdotto int not null,
107     primary key(numeroStili,idProdotto),
108     FOREIGN KEY(idProdotto) REFERENCES Prodotto(codice) ON UPDATE cascade ON DELETE cascade
109 );
110
111 •   DROP TABLE IF EXISTS Di;
112 •   CREATE TABLE Di(
113     idProdotto int not null,
114     idOrdine int not null,
115     idCliente char(16) not null,
116     primary key(idProdotto,idOrdine,idCliente),
117     FOREIGN KEY(idProdotto) REFERENCES Prodotto(codice) ON UPDATE cascade ON DELETE cascade,
118     FOREIGN KEY(idOrdine) REFERENCES Ordine(id) ON UPDATE cascade ON DELETE cascade,
119     FOREIGN KEY(idCliente) REFERENCES Ordine(idCliente) ON UPDATE cascade ON DELETE cascade
120 );

```

Script per popolare il database

```
1 •  SET SQL_SAFE_UPDATES = 0;
2 •  use estore;
3
4 •  DELETE FROM Cliente;
5 •  INSERT INTO Cliente VALUES
6 ("BNCPLA98H05A717B", "Paolo", "Bianchi", '1998-02-05', "Via Vivaldi no.5 Batttipaglia SA", "pbianchi@gmail.com", "pbianchi98", "Visa", '2022-07-09', 1),
7 ("BNCPLA72H19A717A", "Paolo", "Bosi", '1995-06-09', "Via Saturno no.7 Batttipaglia SA", "pbosi@gmail.com", "pbosi95", "Visa", '2022-08-10', 2),
8 ("MRTVLL88B45A717I", "Martina", "Volli", '1988-02-05', "Via Garibaldi no.9 Batttipaglia SA", "mvolli@gmail.com", "mvolli88", "Postepay", '2022-06-08', 3),
9 ("JMSGNN95P05A717F", "James", "Gunn", '1995-09-05', "Via Guardiani no.3 Batttipaglia SA", "jgunn@gmail.com", "jgunn95", "Hype", '2023-05-07', 2),
10 ("LNEMTN99C49A717E", "Elena", "Martini", '1999-03-09', "Via Costanzo no.1 Batttipaglia SA", "emartini@gmail.com", "emartini99", "Visa", '2023-03-10', 0);
11
12 •  DELETE FROM Pagamento;
13 •  INSERT INTO Pagamento VALUES
14 (1,'2021-04-15','Bonifico',10000.00),
15 (2,'2021-03-10','Bonifico',10000.00),
16 (3,'2019-06-20','Bonifico',300.00),
17 (4,'2019-06-20','Bonifico',700.00),
18 (5,'2019-06-20','Bonifico',1000.00),
19 (6,'2022-02-21','Bonifico',6000.00),
20 (7,'2022-02-21','Bonifico',3500.00),
21 (8,'2021-05-09','Bonifico',250.00);
22
23
24 •  DELETE FROM Telefono;
25 •  INSERT INTO Telefono VALUES
26 ("3383415121", "BNCPLA98H05A717B"),
27 ("3285612975", "BNCPLA98H05A717B"),
28 ("3486751290", "BNCPLA72H19A717A"),
29 ("3385649102", "MRTVLL88B45A717I"),
30 ("3450988129", "MRTVLL88B45A717I"),
31 ("3386574390", "MRTVLL88B45A717I"),
32 ("3487690123", "JMSGNN95P05A717F"),
33 ("3485423907", "JMSGNN95P05A717F"),
34 ("3456398129", "LNEMTN99C49A717E"),
35 ("3457890123", "LNEMTN99C49A717E");
36
37 •  DELETE FROM Spedizione;
38 •  INSERT INTO Spedizione VALUES
39 (1,'2021-04-15','2021-04-18',10.00),
40 (2,'2021-03-10','2021-03-12',30.00),
41 (3,'2019-06-20','2019-06-25',10.00),
42 (4,'2019-06-20','2019-06-25',20.00),
43 (5,'2019-06-20','2019-06-25',10.00),
44 (6,'2022-02-21','2022-02-23',50.00),
45 (7,'2022-02-21','2022-02-23',5.00),
46 (8,'2021-05-09','2021-05-15',30.00);
47
```

```
48 •    DELETE FROM Ordine;
49 •    INSERT INTO Ordine VALUES
50      (1,"BNCPLA98H05A717B",1,'2021-04-15',1),
51      (2,"BNCPLA72H19A717A",2,'2021-03-10',2),
52      (3,"BNCPLA72H19A717A",3,'2019-06-20',3),
53      (4,"MRTVLL88B45A717I",4,'2019-06-20',4),
54      (5,"MRTVLL88B45A717I",5,'2019-06-20',5),
55      (6,"MRTVLL88B45A717I",6,'2022-02-21',6),
56      (7,"JMSGNN95P05A717F",7,'2022-02-21',7),
57      (8,"JMSGNN95P05A717F",8,'2021-05-09',8);
58
59 •    DELETE FROM Prodotto;
60 •    INSERT INTO Prodotto VALUES
61      (1,10000.00,5,"Fender","bianco"),
62      (2,7890.00,2,"Fender","rosso"),
63      (3,300.00,80,"Ibanez","bianco"),
64      (4,700.00,80,"Startone","blu"),
65      (5,1000.00,10,"Vintage","giallo ocra"),
66      (6,6000.00,2,"Gibson","rosso"),
67      (7,3500.00,10,"Yamaha","blu"),
68      (8,150.00,50,"Ibanez","marrone"),
69      (9,100.00,20,"Squier","rosso"),
70      (10,250.00,20,"Startone","bianco"),
71      (11,100.00,150,"Yamaha","blu"),
72      (12,300.00,20,"Yamaha","blu"));
73
```

```

74 •    DELETE FROM Recensione;
75 •    INSERT INTO Recensione VALUES
76     (1,1,"BNCPLA98H05A717B","Chitarra sorprendente e professionale",'10'),
77     (1,4,"MRTVLL88B45A717I","Batteria discreta, arrivata leggermente graffiata in un rullante",6),
78     (1,7,"JMSGNN95P05A717F","Tastiera sorprendente e professionale anche se mancano molti stili",8),
79     (1,10,"JMSGNN95P05A717F","Batteria di bassa qualità ma utile nel suo scopo, i materiali fanno presagire una bassa durata",5),
80     (1,12,"BNCPLA72H19A717A","Ottima tastiera per il prezzo",7);
81
82 •    DELETE FROM Batteria;
83 •    INSERT INTO Batteria VALUES
84     (3,4),
85     (5,10);
86
87 •    DELETE FROM ChitarraElettrica;
88 •    INSERT INTO ChitarraElettrica VALUES
89     ("Single Coil",6,1),
90     ("Humbucker",12,2),
91     ("Single Coil",6,3),
92     ("P90",8,5),
93     ("Single Coil",6,6),
94     ("Humbucker",8,8),
95     ("Single Coil",6,9);
96
97 •    DELETE FROM Tastiera;
98 •    INSERT INTO Tastiera VALUES
99     (20,7),
100    (30,11),
101    (80,12);
102
103 •    DELETE FROM Di;
104 •    INSERT INTO Di VALUES
105     (1,1,"BNCPLA98H05A717B"),
106     (1,2,"BNCPLA72H19A717A"),
107     (12,3,"BNCPLA72H19A717A"),
108     (4,4,"MRTVLL88B45A717I"),
109     (5,5,"MRTVLL88B45A717I"),
110     (6,6,"MRTVLL88B45A717I"),
111     (7,7,"JMSGNN95P05A717F"),
112     (10,8,"JMSGNN95P05A717F");

```

6.Implementazione query SQL

Le query sono state rese indipendenti in modo da poter essere riutilizzabili pur dando la possibilità di completare le operazioni.

Script per eseguire le operazioni:

Operazione 1:

```
INSERT INTO Cliente VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
```

Operazione 2:

```
INSERT INTO Pagamento VALUES(0, ?, ?, ?, ?);
```

```
INSERT INTO Spedizione VALUES(0, ?, ?, ?, ?);
```

```
INSERT INTO Ordine VALUES(0,?, ?, ?, ?);  
INSERT INTO Di VALUES(?, ?, ?);
```

Operazione 3:

```
DELETE FROM Recensione WHERE numero = ? AND idProdotto = ?;
```

Operazione 4:

```
SELECT nome, cognome, indirizzo, cf, numTotaleAcquisti FROM Cliente;
```

Operazione 5:

```
SELECT * FROM ChitarraElettrica AS ce, Prodotto AS p WHERE ce.idProdotto = p.codice AND ce.tipoPickup = ?;
```

Operazione 6:

Di questa operazione è possibile selezionarne di tre diverse:

```
SELECT * FROM Prodotto WHERE prezzo < ?;
```

```
SELECT * FROM Prodotto WHERE prezzo = ?;
```

```
SELECT * FROM Prodotto WHERE prezzo > ?;
```

Operazione 7:

```
SELECT o.id from Ordine AS o, Di AS d, Prodotto AS p, Pagamento AS pa  
where d.idOrdine = o.id AND d.idProdotto = p.codice AND o.idPagamento =  
pa.id AND o.dataOrdine = ? AND pa.importo > ? AND p.colore = ?;
```

Operazione 8:

```
SELECT p.codice, p.marca, r.voto FROM Prodotto AS p, Recensione AS r  
WHERE r.idProdotto = p.codice AND p.quantitaDisponibile > ? AND p.colore  
= ? GROUP BY p.codice HAVING ? < (SELECT avg(r.voto) FROM Recensione AS r  
WHERE r.idProdotto = p.codice);
```

Operazione 9:

```
SELECT o.id FROM Prodotto AS p, Di AS d, Spedizione AS s, Ordine AS o  
WHERE d.idOrdine = o.id AND d.idProdotto = p.codice AND o.idSpedizione =  
s.codice AND s.dataArrivo = ? AND NOT EXISTS (SELECT * FROM Recensione AS r  
WHERE r.idProdotto = p.codice);
```

Operazione 10:

```
SET FOREIGN_KEY_CHECKS = 0;
```

```
DELETE FROM Ordine WHERE id = ?;
```

```
DELETE FROM Spedizione WHERE codice = ?;
```

```
DELETE FROM Pagamento WHERE id = ?;
```

```
SET FOREIGN_KEY_CHECKS = 1;
```

```
DELETE FROM Di WHERE idOrdine = ?;
```

7.Applicazione Java

L'applicazione non utilizza alcuna interfaccia grafica ma usa uno switch che fa da selettore tra le 10 operazioni all'interno della console. Inoltre per la maggior parte delle operazioni viene richiesto l'input dell'utente nella console.

I risultati vengono mostrati direttamente nella console di Eclipse tranne per alcune eccezioni che saranno visibili solo sulla workbench di MySQL.

- SELECT nome, cognome, indirizzo, cf, numTotaleAcquisti FROM Cliente;

The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the results of a MySQL query:

```
BNCPLU72H14P471A | Paolo | Bosi | null | Via Saturno no.7 Battipaglia SA | null | null | null | null | 2
BNCPUA89H09A717B | Paolo | Bianchi | null | Via Savigliano no.5 Battipaglia SA | null | null | null | null | 1
JHSGH99P06A717F | James | Green | null | Via Guaridani no.3 Battipaglia SA | null | null | null | null | 2
LNEHTH99C49A717E | Elena | Martin | null | Via Costanzo no.1 Battipaglia SA | null | null | null | null | 0
MRTVLL88B45A717I | Martina | Volli | null | Via Garibaldi no.9 Battipaglia SA | null | null | null | null | 3
```

- SELECT o.id FROM Prodotto AS p, Di AS d, Spedizione AS s, Ordine AS o WHERE d.idOrdine = o.id AND d.idProdotto = p.codice AND o.idSpedizione = s.codice AND s.dataArrivo = '2022-02-23' AND NOT EXISTS (SELECT * FROM Recensione AS r WHERE r.idProdotto = p.codice);

The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the results of a complex MySQL query:

```
9
2022-02-23
6 | null | 0 | null | 0
```