

## **Relazione al Caso di Studio: E-commerce di Prodotti Cosmetici – Generoso Ferrara – Modulo SQL**

### **Introduzione:**

Nel contesto dell'evoluzione digitale e dell'ampia diffusione degli acquisti online, l'industria dei prodotti cosmetici ha sperimentato una trasformazione significativa nel modo in cui i consumatori accedono e acquistano i propri prodotti preferiti. In questo scenario, è fondamentale creare un ambiente di e-commerce efficace e coinvolgente che soddisfi le esigenze dei clienti e al contempo ottimizzi le operazioni aziendali.

Il presente documento si propone di esaminare in dettaglio la progettazione di una base di dati per un'ipotetica piattaforma di e-commerce specializzata nella vendita di prodotti cosmetici. L'obiettivo principale di questo sistema è consentire ai clienti di navigare tra una vasta gamma di prodotti, effettuare ordini in modo semplice e sicuro, e ricevere un'esperienza di shopping online gratificante. Allo stesso tempo, l'azienda mira a gestire con efficienza le operazioni di vendita, la gestione degli stock e il monitoraggio delle preferenze dei clienti.

Nella prima sezione, verrà presentato il modello ER (Entity-Relationship) del sistema, che identifica le principali entità coinvolte e le relazioni tra di loro. Il modello ER fornisce un'istantanea chiara della struttura del database e delle interazioni tra le diverse componenti del sistema.

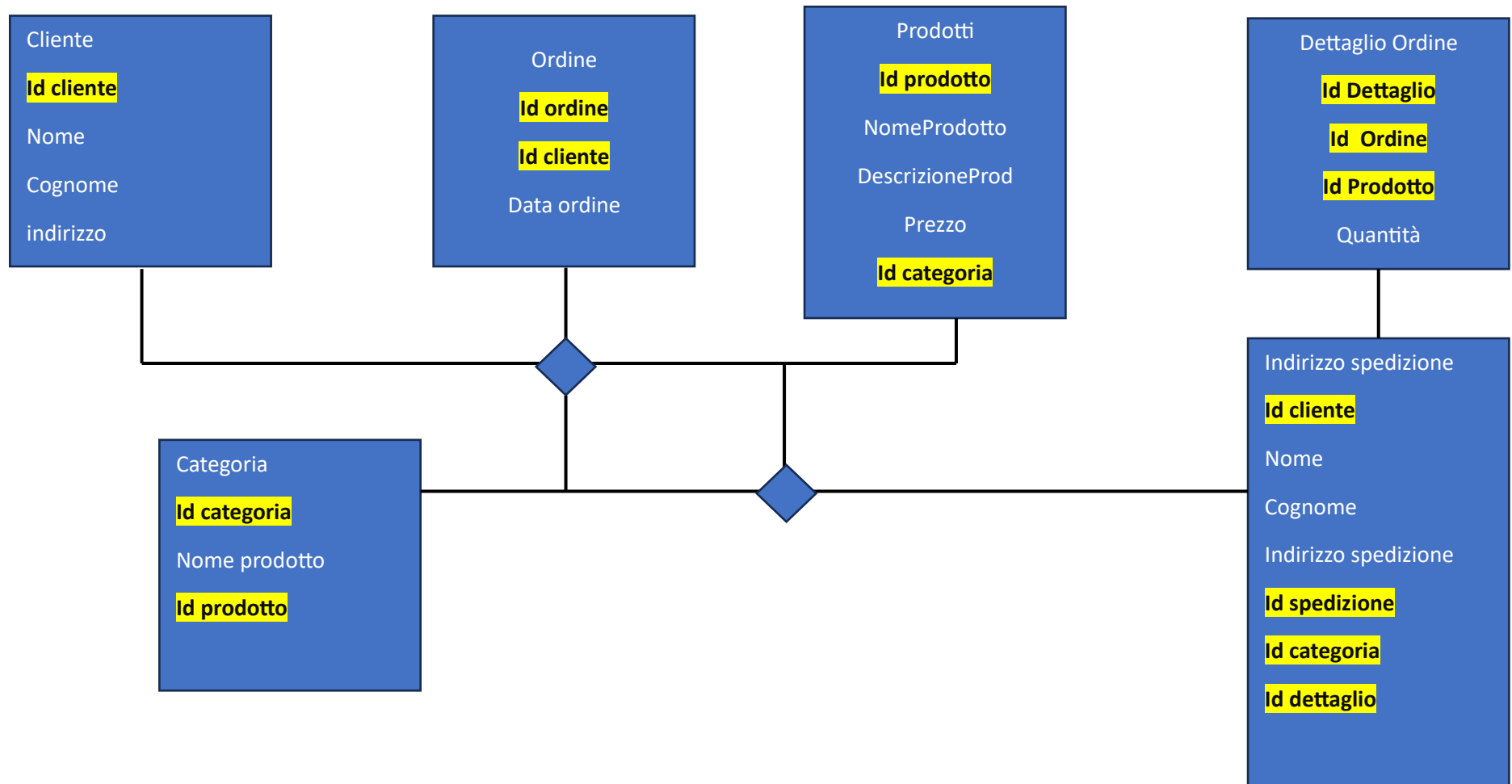
Successivamente, nella sezione dedicata al modello logico, verranno tradotte le entità e le relazioni del modello ER in tabelle concrete del database. Ogni tabella includerà gli attributi appropriati e le chiavi primarie e esterne necessarie per garantire l'integrità referenziale dei dati.

Infine, nella sezione di progettazione fisica, saranno forniti i comandi DDL (Data Definition Language) e DML (Data Manipulation Language) necessari per creare tabelle, definire chiavi primarie ed esterne, e popolare il database con dati di esempio. Inoltre, verranno introdotte alcune viste che agevolano la fruizione dei dati e semplificano le interrogazioni comuni.

Il documento si concluderà con una serie di interrogazioni di esempio, dimostrando come la base di dati possa essere sfruttata per ottenere informazioni rilevanti, come analisi delle vendite, prodotti più popolari e altro ancora. Inoltre, si sottolineerà come il sistema sia in grado di supportare richieste complesse attraverso l'uso di join, raggruppamenti e ordinamenti.

In definitiva, la progettazione di questa base di dati mira a rappresentare un'infrastruttura solida e scalabile che supporta l'esperienza di shopping online per i clienti e le operazioni aziendali per l'azienda di prodotti cosmetici. La visione complessiva è quella di creare un ambiente virtuale che unisca efficacemente le esigenze dei consumatori moderni con le dinamiche aziendali, per conseguire un successo reciproco e duraturo.

## Modello ER



### Descrizione delle entità:

- **Cliente:** Rappresenta i clienti registrati nel sistema.
- **Ordine:** Rappresenta un ordine effettuato da un cliente e collega i dettagli dell'ordine.
- **Dettaglio Ordine:** Contiene i dettagli specifici di ogni prodotto all'interno di un ordine.
- **Prodotti:** Rappresenta i prodotti in vendita nel negozio.
- **Categoria:** Categorie di prodotti a cui appartengono i vari prodotti.
- **Indirizzo Spedizione:** Gli indirizzi dei clienti per la spedizione.

### Descrizione delle relazioni:

- Un **Cliente** può effettuare **zero o molti Ordini**.
- Un **Ordine** deve essere collegato a **un e solo un Cliente**.
- Un **Ordine** può contenere **zero o molti Dettagli Ordine**.
- Un **Dettaglio Ordine** deve essere legato a **un e solo un Ordine** e **un e solo un Prodotto**.
- Un **Prodotti** può appartenere a **una sola Categoria**.
- Una **Categoria** può avere **zero o molti Prodotti**.
- Un **Cliente** può avere **zero o molti Indirizzi**, ma un **Indirizzo** deve essere legato a **un e solo un Cliente**.

### Modello Logico:

#### Tabella: Cliente

- IDCliente (PK)
- Nome
- Cognome
- Email

#### Tabella: Indirizzo spedizione

- IDIndirizzo (PK)
- IDCliente (FK verso Cliente)
- Via
- Città
- CAP
- Paese

#### Tabella: Ordine

- IDOrdine (PK)

- IDCliente (FK verso Cliente)
- DataOrdine

**Tabella: Categoria**

- IDCategoria (PK)
- NomeCategoria

**Tabella: Prodotti**

- IDProdotto (PK)
- NomeProdotto
- Descrizione
- Prezzo
- IDCategoria (FK verso Categoria)

**Tabella: DettaglioOrdine**

- IDDettaglio (PK)
- IDOrdine (FK verso Ordine)
- IDProdotto (FK verso Prodotto)
- Quantità

Nella progettazione logica, le tabelle rappresentano le entità e le relazioni del modello ER e includono le chiavi primarie (PK) e le chiavi esterne (FK) per stabilire le relazioni tra di loro. Questo modello può servire come punto di partenza per l'implementazione del database.

Le tabelle elencate sopra costituiscono il modello logico della base di dati. Ogni tabella ha le proprie chiavi primarie e chiavi esterne per stabilire relazioni tra di loro.

**Progettazione Fisica**

-- Creazione delle tabelle

```
CREATE TABLE Prodotti (
  IDProdotto INT AUTO_INCREMENT PRIMARY KEY,
  NomeProdotto VARCHAR(255),
  Descrizione TEXT,
  Prezzo DECIMAL(10, 2),
  IDCategoria INT,
  FOREIGN KEY (IDCategoria) REFERENCES Categoria(IDCategoria)
);
```

```
CREATE TABLE Categoria (  
    IDCategoria INT AUTO_INCREMENT PRIMARY KEY,  
    NomeCategoria VARCHAR(50)  
);
```

```
CREATE TABLE Clienti (  
    IDCliente INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(50),  
    Cognome VARCHAR(50),  
    Email VARCHAR(100),  
    Indirizzo TEXT  
);
```

```
CREATE TABLE Ordini (  
    IDOrdine INT AUTO_INCREMENT PRIMARY KEY,  
    DataOrdine DATE,  
    IDCliente INT,  
    FOREIGN KEY (IDCliente) REFERENCES Clienti(IDCliente)  
);
```

```
CREATE TABLE DettagliOrdine (  
    IDDettaglio INT AUTO_INCREMENT PRIMARY KEY,  
    IDOrdine INT,  
    IDProdotto INT,  
    Quantità INT,  
    FOREIGN KEY (IDOrdine) REFERENCES Ordini(IDOrdine),  
    FOREIGN KEY (IDProdotto) REFERENCES Prodotti(IDProdotto)  
);
```

-- Creazione di una vista per ottenere il dettaglio degli ordini

```
CREATE VIEW DettaglioOrdiniView AS
SELECT o.IDOrdine, o.DataOrdine, c.Nome AS NomeCliente, c.Cognome AS CognomeCliente,
       p.NomeProdotto, do.Quantità, p.Prezzo, (do.Quantità * p.Prezzo) AS Totale
FROM Ordini o
JOIN Clienti c ON o.IDCliente = c.IDCliente
JOIN DettaglioOrdine do ON o.IDOrdine = do.IDOrdine
JOIN Prodotti p ON do.IDProdotto = p.IDProdotto;
```

### **Interrogazioni:**

Ecco 10 interrogazioni SQL che potresti eseguire sulla base di dati dell'e-commerce di prodotti cosmetici:

#### **1. Elenco dei clienti:**

```
SELECT * FROM Clienti;
```

#### **2. Dettagli degli ordini di un cliente specifico:**

```
SELECT o.IDOrdine, o.DataOrdine, p.NomeProdotto, do.Quantità
FROM Ordine o
JOIN DettaglioOrdine do ON o.IDOrdine = do.IDOrdine
JOIN Prodotti p ON do.IDProdotto = p.IDProdotto
WHERE o.IDCliente = [ID_cliente];
```

#### **3. Totale delle vendite per ogni categoria:**

```
SELECT c.NomeCategoria, SUM(p.Prezzo * do.Quantità) AS TotaleVendite
FROM Categoria c
LEFT JOIN Prodotti p ON c.IDCategoria = p.IDCategoria
LEFT JOIN DettaglioOrdine do ON p.IDProdotto = do.IDProdotto
```

GROUP BY c.NomeCategoria;

**4. Prodotti più venduti (in base alla quantità):**

```
SELECT p.NomeProdotto, SUM(do.Quantità) AS QuantitàTotaleVenduta
FROM Prodotti p
LEFT JOIN DettaglioOrdine do ON p.IDProdotto = do.IDProdotto
GROUP BY p.NomeProdotto
ORDER BY QuantitàTotaleVenduta DESC
LIMIT 10;
```

**5. Elenco di ordini effettuati in una data specifica:**

```
SELECT o.IDOrdine, c.Nome, c.Cognome, o.DataOrdine
FROM Ordine o
JOIN Cliente c ON o.IDCliente = c.IDCliente
WHERE o.DataOrdine = '2023-08-24';
```

**6. Prodotti nella categoria "Trucco":**

```
SELECT p.NomeProdotto, p.Prezzo
FROM Prodotti p
JOIN Categoria c ON p.IDCategoria = c.IDCategoria
WHERE c.NomeCategoria = 'Trucco';
```

**7. Totale speso da un cliente specifico:**

```
SELECT c.Nome, c.Cognome, SUM(p.Prezzo * do.Quantità) AS TotaleSpeso
FROM Cliente c
JOIN Ordine o ON c.IDCliente = o.IDCliente
```

```
JOIN DettaglioOrdine do ON o.IDOrdine = do.IDOrdine
JOIN Prodotti p ON do.IDProdotto = p.IDProdotto
WHERE c.IDCliente = [ID_cliente]
GROUP BY c.Nome, c.Cognome;
```

#### 8. **Dettagli completi di un ordine specifico:**

```
SELECT o.IDOrdine, o.DataOrdine, c.Nome, c.Cognome, p.NomeProdotto, do.Quantità, p.Prezzo
FROM Ordine o
JOIN Cliente c ON o.IDCliente = c.IDCliente
JOIN DettaglioOrdine do ON o.IDOrdine = do.IDOrdine
JOIN Prodotti p ON do.IDProdotto = p.IDProdotto
WHERE o.IDOrdine = [ID_ordine];
```

#### 9. **Elenco dei clienti con più di un indirizzo:**

```
SELECT c.Nome, c.Cognome, COUNT(i.IDIndirizzo) AS NumeroIndirizzi
FROM Cliente c
JOIN Indirizzo i ON c.IDCliente = i.IDCliente
GROUP BY c.Nome, c.Cognome
HAVING COUNT(i.IDIndirizzo) > 1;
```

#### 10. **Miglior cliente (con il totale speso) in un intervallo di date:**

```
SELECT c.Nome, c.Cognome, SUM(p.Prezzo * do.Quantità) AS TotaleSpeso
FROM Cliente c
JOIN Ordine o ON c.IDCliente = o.IDCliente
JOIN DettaglioOrdine do ON o.IDOrdine = do.IDOrdine
JOIN Prodotti p ON do.IDProdotto = p.IDProdotto
WHERE o.DataOrdine BETWEEN '2023-01-01' AND '2023-08-24'
```



```
GROUP BY c.Nome, c.Cognome  
ORDER BY TotaleSpeso DESC  
LIMIT 1;
```