

Università degli Studi di Trieste  
Ingegneria Elettronica e Informatica

Progetto di Basi di Dati  
IN0500900  
“Web Store”

Stefano Chen

30/05/2023

# Indice

1.	<b>Presentazione Progetto</b>	<b>2</b>
2.	<b>Glossario dei Termini</b>	<b>4</b>
3.	<b>Schema Entity-Relationship</b>	<b>5</b>
4.	<b>Vincoli Non Esprimibili Graficamente</b>	<b>7</b>
5.	<b>Tabella dei Volumi</b>	<b>8</b>
6.	<b>Ristrutturazione</b>	<b>10</b>
7.	<b>Schema Entity-Relationship Ristrutturato</b>	<b>12</b>
8.	<b>Schema Logico</b>	<b>13</b>
9.	<b>Normalizzazione</b>	<b>14</b>
10.	<b>Query SQL</b>	<b>16</b>

# Presentazione Progetto

Un negozio specializzato nella vendita di componenti hardware e software per computer, decide di aprire un web store per permettere ai clienti di acquistare comodamente da casa.

I clienti, collegandosi alla web page, avranno la possibilità di:

- Effettuare il Login/Logout o il SignUp
- Modificare i propri dati
- Navigare nel catalogo dei Prodotti
- Cercare un prodotto per nome o per tag
- Visualizzare le informazioni relative a un prodotto
- Aggiungere un prodotto al proprio carrello
- Confermare l'ordine
- Effettuare il Pagamento
- Controllare lo stato/dettagli dell'ordine
- Visualizzare la cronologia degli ordini

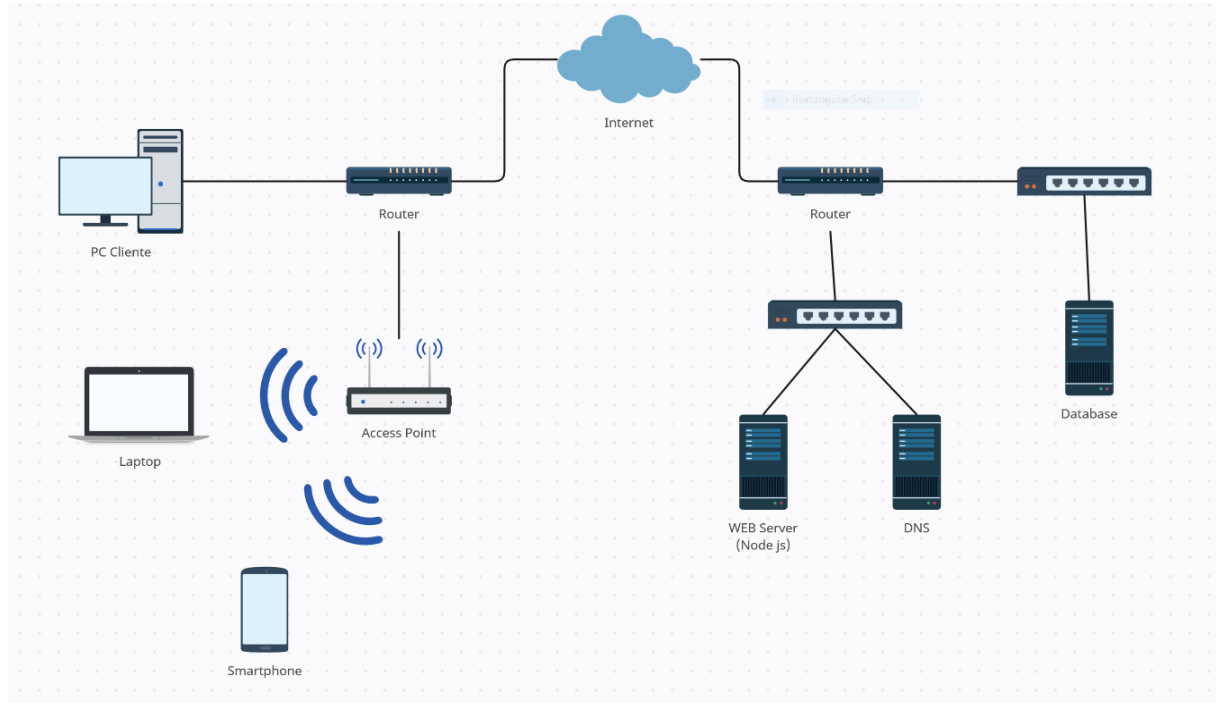
I dipendenti del negozio avranno a loro disposizione un software gestionale che si collegherà al database aziendale per effettuare le seguenti operazioni:

- Visualizzare tutti gli ordini effettuati in una specifica data
- Visualizzare tutti gli ordini ancora da elaborare
- Visualizzare i dettagli di un ordine
- Aggiungere/Modificare un ordine
- Inserire o modificare il codice di tracking di un ordine
- Modificare lo stato di un ordine
- Aggiungere/Modificare un prodotto
- Aggiungere/Modificare i tags di un prodotto

Il negozio ci richiede di progettare un database in grado di permettere le azioni sopra nominate.

# Analisi del Progetto

Supponiamo che il negozio abbia una infrastruttura di rete simile all'immagine sottostante:



Durante la procedura di SignUp, per ogni utente verranno memorizzati il nome, il cognome, un numero di telefono, l'indirizzo email, una password e l'indirizzo di consegna (Via, Città, CAP). Ogni cliente sarà identificato in modo univoco da un numero cliente; la password dell'account **non** dovrà essere salvata in chiaro (la fase di crittografia sarà responsabilità del Web server).

L'operazione di Login sarà composta da due step:

1. Il Cliente invia una POST Request, contenente email e password, al WEB Server
2. Il WEB Server interroga il Database per verificare che esista la entry corrispondente.

Se esiste allora il Web Server invia al PC del Cliente una Response contenente un token di autenticazione. Altrimenti restituisce una Response con lo Status Code 401/ Unauthorized.

Sul database dovranno essere salvate le informazioni dei prodotti in vendita (nome, modello, descrizione, quantità, prezzo). Ogni prodotto avrà dei tag assegnati, così da facilitare l'operazione di ricerca. I Tag indicano il gruppo di appartenenza di un prodotto e/o le sue caratteristiche (e.g. monitor, gaming, tastiera, OS, ecc...)

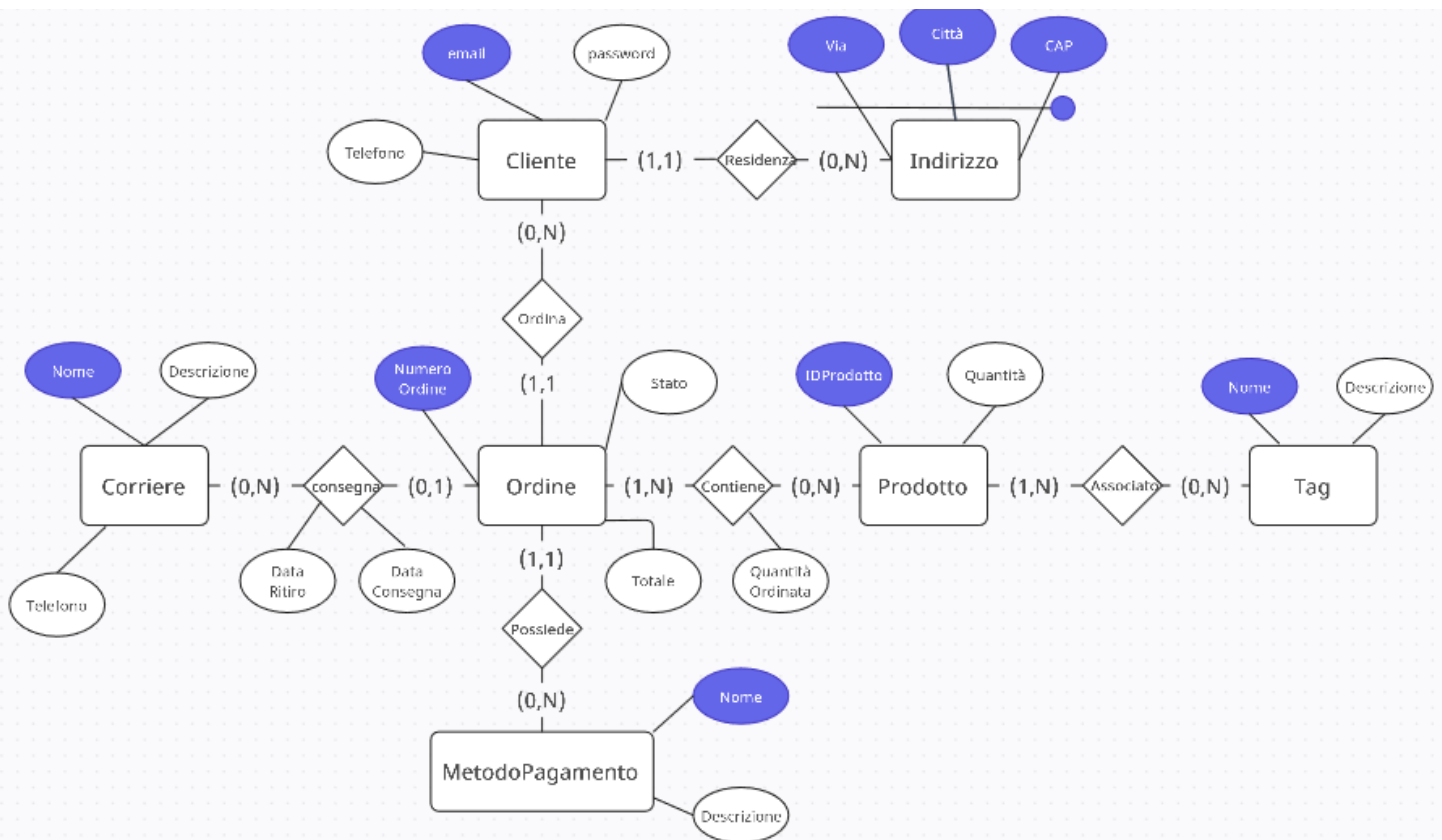
Per motivi di prestazioni si decide che le funzionalità del carrello saranno implementate dal WEB Server, usando i Cookies e le Session; nel database

verranno inseriti solo i dettagli degli ordini confermati. Ogni ordine avrà un metodo di pagamento assegnato (e.g. Bonifico, PayPal, Carta di Credito, ecc...) ciò implica che per ogni ordine verrà salvato anche il numero della ricevuta di pagamento. La consegna degli ordini avverrà tramite corrieri di terze parti (e.g. GLS, BRT, DHL, ecc...) questi ultimi generano un codice di tracking che verrà memorizzato nel database e dovrà essere messo a disposizione del cliente.

## Glossario dei Termini

Termine	Definizione	Sinonimi	Collegamento
Cliente	Colui che acquista servizi e prodotti	Acquirente, Compratore	Ordine, Pagamento
Indirizzo di Consegna	l'identificativo del luogo in cui avverrà la consegna		Cliente, Ordine
Prodotto	Rappresenta il bene per cui il cliente paga	Bene, Servizio	Ordine
Pagamento	l'azione di scambiare soldi per un prodotto		Ordine, Cliente
Codice di Tracking	è un codice alfanumerico che identifica in modo univoco ogni spedizione, ogni corriere ha dei formati differenti	Codice di Spedizione, Identificativo della Spedizione, Numero del pacco	Ordine
Password	è una sequenza alfanumerica scelta arbitrariamente dal cliente	Passcode, Parola d'accesso	Cliente
Crittografia	è la conversione dei dati da un formato leggibile a un formato codificato. per impedire che le informazioni vengano lette da chi vuole servirsene per scopi malevoli.		Password

# Schema Entity-Relationship



## Entity

Entity	Descrizione	Attributi	Identificatore
Cliente	Colui che acquista servizi e prodotti	Nome, Cognome, Email, Password, Telefono	Email
Indirizzo	l'identificativo del luogo in cui avverrà la consegna dell'ordine	Via, CAP, Città	Via,Citta,CAP
Ordine	Un insieme di prodotti ordinati dal cliente	Numero Ordine, Stato, Data Ordine, Totale, RicevutaPagamento, CodiceTracking	Numero Ordine

Entity	Descrizione	Attributi	Identificatore
Metodo Pagamento	Rappresenta le diverse tipologie di pagamento disponibili (e.g. Bonifico, Paypal, MasterCard, ecc...)	Nome, Descrizione	Nome
Corriere	Rappresenta l'azienda che prenderà in carico la consegna (e.g. GLS, BRT, DHL, ecc...)	Nome, Descrizione, Telefono	Nome
Prodotto	Rappresenta il prodotto in vendita	IDProdotto, Nome, Modello, Descrizione, Quantità, Prezzo Unitario	IDProdotto
Tag	indica il gruppo di appartenenza di un prodotto (e.g. monitor, gaming, tastiera, OS, ecc...)	Nome, Descrizione	Nome

## Relationship

Relationship	Descrizione	Componenti	Attributi
Residenza	Ogni cliente possiede un indirizzo di consegna	Cliente, Indirizzo,	
Ordina	Ogni cliente può generare uno o più ordini	Cliente, Ordine	
Consegna	Ogni ordine è consegnato da un corriere	Ordine, Corriere	DataRitiro, DataConsegna
Possiede	Ogni ordine possiede un metodo di pagamento	Ordine, MetodoPagamento	

Relationship	Descrizione	Componenti	Attributi
Contiene	Ogni ordine contiene una lista di prodotti	Ordine, Prodotto	QuantitàOrdinata
Associato	Ogni prodotto è associato a diversi tag	Prodotto, Tag	

## Vincoli non esprimibili graficamente

1. La data di consegna non può essere precedente alla data di ritiro
2. La quantità ordinata non può essere minore o uguale a zero
3. Il prezzo unitario di ogni prodotto non può essere un numero negativo
4. La password (in chiaro) deve essere lunga almeno 8 caratteri, deve contenere almeno una maiuscola e una minuscola
5. la quantità disponibile di un prodotto non deve essere un numero negativo
6. Ogni qualvolta un ordine viene confermato (viene inserita una nuova entry), bisogna diminuire la quantità disponibile, nel magazzino, per ogni prodotti presenti nell'ordine.

## Considerazioni generali

Si osserva che l'attributo Telefono relativo all'entità Cliente potrebbe essere un'entità distinta. Si decide di non considerare tale opzione poiché si è interessati a memorizzare solo il numero di telefono principale e non anche quelli secondari.

Si osserva che l'attributo Stato dell'entità Ordine potrebbe essere un'entità distinta così da ottenere un database con una granularità maggiore. Si sceglie di implementare questa possibilità durante il processo di ristrutturazione del modello E-R.



# Tabella dei Volumi

Suppongo che ci siano all'incirca 500 clienti iscritti all'online store.

Suppongo che esistano 15 diversi corrieri.

Suppongo che esistano 10 diversi metodi di pagamento.

Suppongo che esistano 100 tag differenti

Si stima che ogni mese il negozio si rifornisce con 4000 prodotti.

Si stima che ogni prodotto è associato in media a 2 tag

Si stima una media di 2 prodotti per ordine.

Si stima che ogni cliente effettui 3 ordini ogni mese.

Concetto	Tipo	Volume
Cliente	E	500
Indirizzo	E	500
Corriere	E	15
MetodoPagamento	E	10
Prodotto	E	4000
Tag	E	100
Ordine	E	1500 (=500x3)
Residenza	R	500
Ordina	R	1500
Consegna	R	1500
Possiede	R	1500
Contiene	R	3000 (=500x3x2)
Associato	R	8000 (=4000x2)

## Operazioni di interesse

Operazione	Tipo	Frequenza
Visualizzare tutti gli ordini effettuati in una specifica data	Batch	10/giorno
Visualizzare tutti gli ordini effettuati da un cliente	Batch	100/giorno
Visualizzare tutti gli ordini ancora da elaborare	Batch	100/giorno
Visualizzare i dettagli di un ordine	Interattiva	50/giorno
Aggiungere/Modificare un ordine	Interattiva	50/giorno
Inserire o modificare il codice di tracking di un ordine	Interattiva	50/giorno
Modificare lo stato di un ordine	Interattiva	50/giorno
Aggiungere/Modificare un prodotto	Interattiva	100/settimana
Aggiungere/Modificare i tags di un prodotto	Interattiva	100/settimana
Ricerca di un prodotto per nome e/o modello	Interattiva	250/giorno
Ricerca di prodotti in base ai tags	Batch	400/giorno

# Ristrutturazione

## Analisi delle ridondanze

L'entità Ordine presenta l'attributo Totale. Questo risulta essere un attributo calcolabile. In seguito all'analisi delle azioni da eseguire sul database, risulta che l'attributo in questione sia un appesantimento non necessario per il sistema. Si sceglie quindi di eliminare l'attributo Totale dato che è facilmente ricavabile.

## Eliminazione delle generalizzazioni

Non sono presenti generalizzazioni.

## Partizionamento di entità

Si osserva che l'attributo Stato dell'entità Ordine potrebbe essere un'entità distinta così da ottenere un database con granularità maggiore. Di conseguenza si decide di eliminare tale attributo e di creare una nuova entità Stato. Questa entità sarà collegata tramite una relazione "Appartenenza" all'Ordine.

Analogamente, dall'entità Cliente verrà estratto l'entità Account contenente gli attributi email e password.

## Partizionamento di relationship

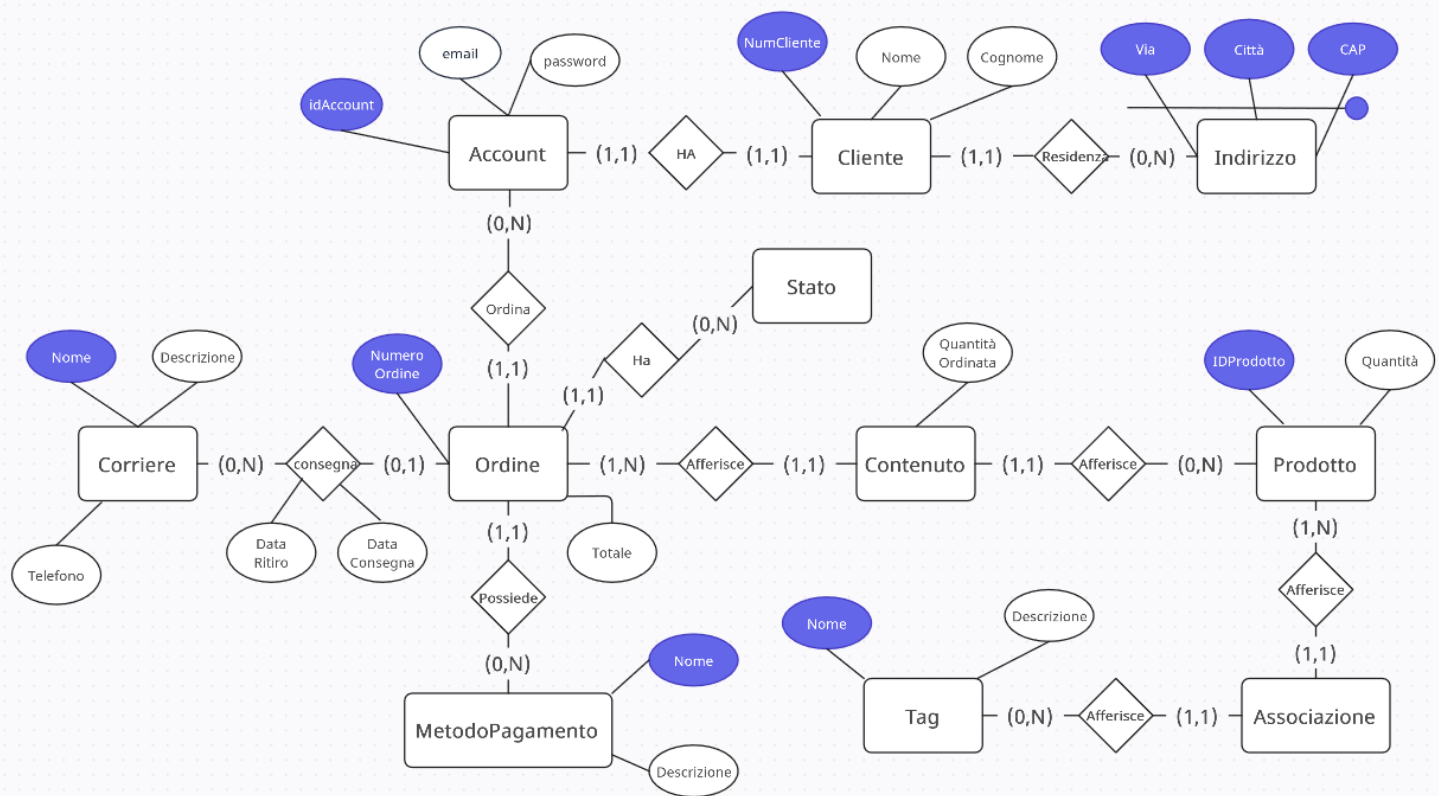
Osservo che la Relazione "Contiene" che collega le entità Ordine e Prodotto è una relazione molti a molti. La cardinalità (0,N) da parte del Prodotto è giustificata dal fatto che un prodotto può essere presente in tanti ordini, mentre la cardinalità (1,N) da parte dell'entità Ordine è giustificato dal fatto che un ordine è composto da almeno un prodotto. Quindi, si decide di eseguire una partizione della relazione "Contiene" introducendo l'entità Contenuto, che sarà in relazione (1,N) - (1,1) con Ordine e in relazione (1,1) - (0,N) con Prodotto.

Si noti che anche la Relazione “Associato” che collega Prodotto e Tag è una relazione molti a molti, per cui si effettua una decisione analoga introducendo l’entità Associazione

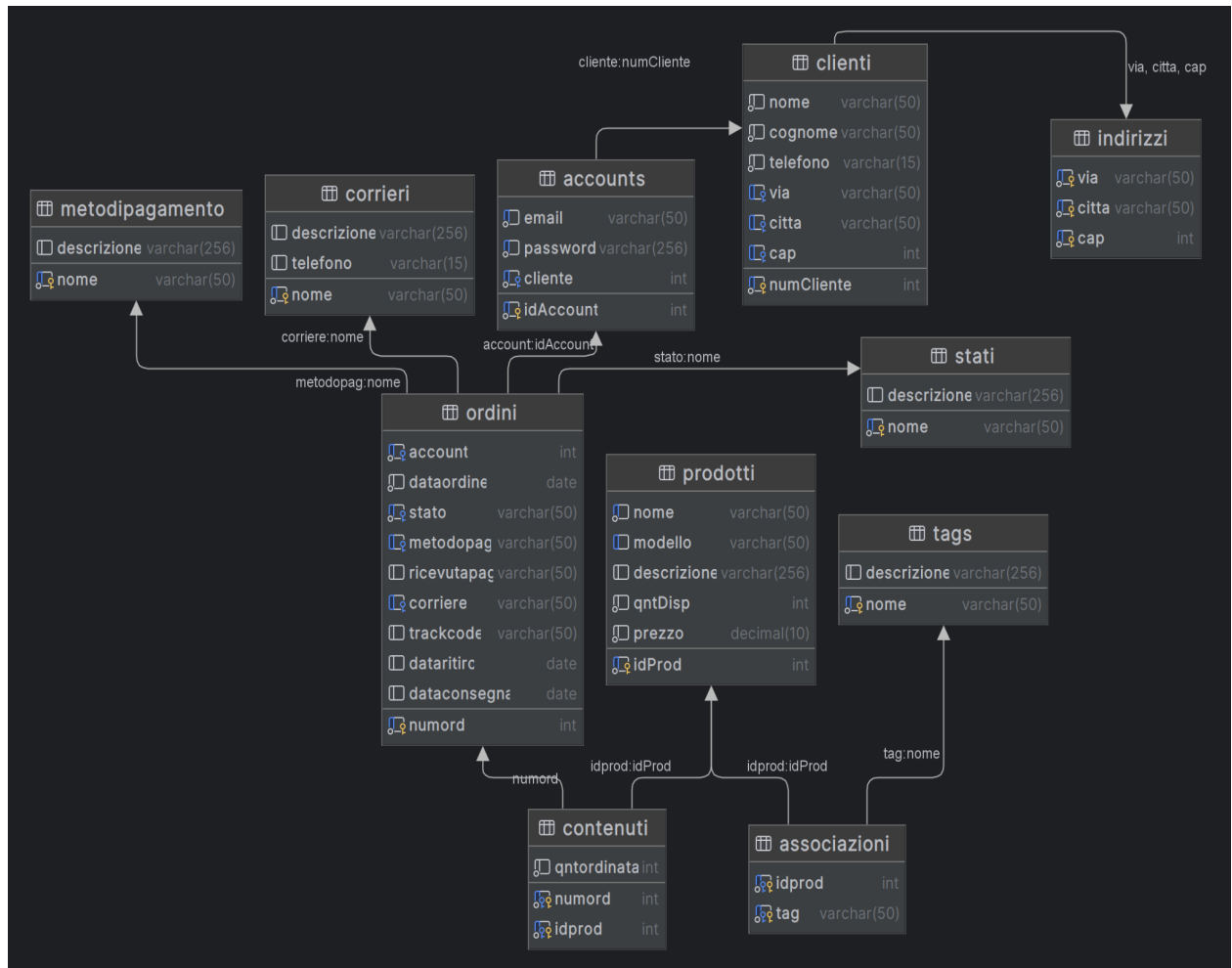
## Scelta degli identificatori

Entità	Identificatore
Cliente	NumCliente
Account	idAccount
Indirizzo	Via, Città, CAP
Corriere	Nome
MetodoPagamento	Nome
Prodotto	IDProdotto
Tag	Nome
Ordine	NumeroOrdine
Contenuto	NumeroOrdine, IDProdotto
Associazione	IDProdotto, Tag

# Schema Entity-Relationship Ristrutturato



# Schema Logico



- Cliente (NumCliente, Nome, Cognome, Telefono, Via, Citta, CAP)
- Account (idAccount, Email, Password, Cliente)
- Indirizzo (Via, Citta, CAP)
- Ordine (NumOrdine, Account, DataOrdine, Stato, MetodoPag, RicevutaPag, Corriere, TrackCode, DataRitiro, DataConsegna)
- Stato (Nome, Descrizione)
- MetodoPagamento (Nome, Descrizione)
- Corriere (Nome, Descrizione, Telefono)
- Prodotto (IDProd, Nome, Modello, Descrizione, QntDisp, Prezzo)

- Tag (Nome, Descrizione)
- Contenuto (NumOrd, IDProd, QntOrdinata)
- Associazione (IDProd, Tag)

# Normalizzazione

## Prima Forma Normale

La prima forma normale (1FN) dice che tutte le colonne devono essere atomiche:

- Un valore per colonna
- Non ci sono colonne duplicate

La prima forma è rispettata.

## Seconda Forma Normale

Una tabella è in seconda forma normale (2FN) se è in 1FN e ciascuna colonna dipende dall'intera primary key

L'unica tabella che contiene chiavi composte è indirizzi e ogni colonna dipende dall'intera primary key. Quindi anche la seconda forma è rispettata

## Terza Forma Normale

Una tabella è in terza forma normale (3FN) se è in 2FN e ogni attributo dipende solo dalla primary key. Cioè:

- tutte le colonne sono indipendenti tra loro
- non ci sono colonne calcolate

Dopo la fase di ristrutturazione, grazie all'eliminazione del campo Totale, nessuna tabella contiene campi calcolati.

La terza forma è rispettata.

# Progettazione Fisica

## Scelta del DBMS

Considerando il fatto che il database verrà usato da un negozio di piccola-media dimensione, e il numero di clienti online è relativamente piccolo. Si decide di adottare un database relazionale (e.g. MySQL) invece che uno NoSQL (e.g. MongoDB).

## Indici per la Ricerca

Si può osservare che le ricerche effettuate sulla tabella Prodotti avverranno spesso tramite gli attributi Nome e/o Modello. Siccome tali ricerche presentano una frequenza molto elevata, risulta conveniente aggiungere un indice secondario per questi campi.

Si noti che lo stesso ragionamento può essere effettuato per la ricerca tramite tags, ma in questo caso non c'è la necessità di inserire un indice secondario perché i tags fanno parte della chiave primaria della tabella Associazioni, ciò implica che esiste già un indice primario riguardante il suddetto campo.



# Query SQL

## Realizzazione delle operazioni

Di seguito verranno mostrati pezzi di codice che permettono la realizzazione delle operazioni più frequenti. Si è deciso di realizzarle tutte tramite delle Stored Procedure poiché quest'ultime rimangono memorizzate all'interno del Database, permettendo il loro riutilizzo anche in sessioni future.

## Operazioni Cliente

- Visualizzare tutti gli ordini effettuati da un account

```
DELIMITER $$

CREATE PROCEDURE allOrdersByAccount(IN idAccount int)
BEGIN
    SELECT numord,dataordine,stato,corriere,trackcode,dataconsegna
    from ordini where account = idAccount
    order by numord DESC;
end $$

DELIMITER ;
```

- Visualizza i prodotti presenti nell'ordine; dato in input il numero d'ordine

```
DELIMITER $$

CREATE PROCEDURE orderDetail(IN numeroOrd int, OUT totale decimal(10,2))
BEGIN
    SELECT sum(qntordinata * prezzo) into totale from contenuti c inner join prodotti p on c.idprod = p.idProd
    group by numOrd having numOrd = numeroOrd;

    SELECT nome,modello,qntordinata,prezzo
    from contenuti c inner join prodotti p on c.idprod = p.idProd
    where numord = numeroOrd;
end $$

DELIMITER ;
```

## Operazioni Dipendenti Negozio

- Visualizza tutti gli ordini con un determinato stato

```
DELIMITER $$

CREATE PROCEDURE allOrdersByStatus(IN status varchar(50))
BEGIN
    SELECT numord, dataordine, corriere, trackcode, email,nome,cognome,telefono,via,citta,cap
    from ordini o inner join accounts a on o.account = a.idAccount
    | inner join clienti c on a.cliente = c.numCliente
    where stato = status;
end $$

DELIMITER ;
```

- Inserimento di un Prodotto

```
DELIMITER $$

CREATE PROCEDURE addProduct(IN name varchar(50), IN model varchar(50), IN descr varchar(256),
                             IN qnt int, IN prz decimal(10,2))
BEGIN
    declare x int;
    select count(nome) into x from prodotti p where p.nome = name AND p.modello = model;
    IF (x > 0) THEN
        UPDATE prodotti p SET qntDisp = qntDisp + qnt where p.nome = name and p.modello = model;
    ELSE
        INSERT INTO prodotti(nome, modello, descrizione, qntDisp, prezzo) VALUE (name,model,descr,qnt,prz);
    end if;
end $$

DELIMITER ;
```

# Trigger

Per implementare il sesto vincolo non esprimibile graficamente, indicato nella parte di progettazione, si decide di usare un trigger.

## Vincolo:

“Ogni qualvolta un ordine viene confermato (viene inserita una nuova entry), bisogna diminuire la quantità disponibile, nel magazzino, per ogni prodotti presenti nell’ordine”.

```
DELIMITER $$

CREATE TRIGGER trg_GestioneMagazzino
AFTER INSERT ON online_shop.contenuti FOR EACH ROW
BEGIN
    declare qnt int;
    set qnt = NEW.qntordinata;

    UPDATE prodotti SET qntDisp = qntDisp - qnt WHERE idProd = NEW.idprod;

end $$

DELIMITER ;
```