



1. Abstract

Se ami la moda e vuoi scoprire le ultime tendenze, Svelasca è l'app che fa per te! Fondata nel 2015 dal suo ideatore Steve Wolf, Svelasca è un'app che ti offre una vasta selezione di vestiti, accessori e abbigliamento di lusso, con un'apposita sezione scelta con cura dai nostri esperti stilisti.

Dopo aver effettuato la registrazione e inserito i propri dati è possibile acquistare i propri capi preferiti con un semplice tap e riceverli a casa in pochi giorni.

Per le imprese, la procedura è facile e basterà inviare il proprio catalogo prodotti e i dati dell'azienda per iniziare a commercializzare le proprie creazioni.

Il servizio di spedizione si avvale di personale qualificato e di collaboratori esterni.

L'utente ha la possibilità di dare un giudizio sui prodotti acquistati con un punteggio e, se lo desidera, con un commento scritto.

2. Analisi dei requisiti

2.1. Descrizione testuale dei requisiti

Per ogni **utente** che si è iscritto all'applicazione, la base di dati conserva le sue informazioni:

- Nome
- Cognome
- Indirizzo e-mail
- Password
- Indirizzi di consegna (stato, CAP, città, via, numero civico)
- Metodi di pagamento
- Numero di telefono

Per usufruire della consegna gratuita, un utente ha la possibilità di sottoscrivere Svelasca Plus, un servizio che ha una durata di un anno e che si rinnova automaticamente.

L'utente deve indicare dei **metodi di pagamento** che rispettino:

- Circuito
- Intestatario
- Numero
- Scadenza

Shopitfy (un servizio esterno) si occupa di gestire il pagamento

Ogni **azienda** che desidera vendere le proprie creazioni su Svelasca deve fornire:

- Nome
- Partita iva
- Indirizzo (stato, CAP, città, via, numero civico)
- Indirizzo e-mail
- Numero di telefono

- Prodotti offerti
- Data e orario di apertura e chiusura

Un marchio ritenuto particolarmente importante per Svelasca ha la possibilità di diventare partner se soddisfa determinati standard. Questa è un'opportunità unica e vantaggiosa per entrambe le parti, che potranno collaborare in modo efficace e innovativo. Per accedere alla partnership, il marchio dovrà dimostrare di condividere i valori, la visione e la missione, oltre a rispettare i requisiti di qualità che contraddistinguono Svelasca.

I brand partner ottengono un miglior posizionamento dei prodotti quando mostrati all'utente.

Ogni **prodotto** ha:

- Nome
- Prezzo
- Brand
- Recensione (valutazione + testo)
- Recensione media

Alcuni prodotti di alta qualità, realizzati con materiali e tecniche esclusive sono disponibili solo in un numero limitato e certificato di esemplari, che li rende ancora più preziosi e distintivi. Sono i prodotti in edizione limitata.

Ogni prodotto appartiene ad una sola **categoria**, ma una categoria ha più prodotti. La categoria è identificata da:

- Nome categoria

Ogni **corriere** fornisce:

- Nome
- Cognome
- Codice Fiscale
- Indirizzo e-mail
- Numero di telefono
- IBAN

Un corriere può essere dipendente di Svelasca e quindi è necessario conoscere lo stipendio, oppure può essere padroncino e quindi deve fornire la partita iva.

Un **ordine** consiste nella selezione di uno o più prodotti disponibili, indicando la quantità desiderata per ciascuno. Per ogni ordine, si devono registrare:

- Utente
- Prodotti ordinati
- Metodo di pagamento utilizzato
- Data e ora della transazione
- Prezzo totale (escluso costo spedizione)

Per garantire la qualità e l'efficienza del servizio di **spedizione**, è indispensabile disporre delle seguenti informazioni:

- Indirizzo di consegna
- Indirizzi di ritiro

- Corriere impiegato
- Costo spedizione
- Giorno di consegna previsto
- Giorno e orario effettivi di consegna

Un modo per esprimere la propria opinione sui prodotti acquistati è lasciare una **recensione**: l'utente può valutare il prodotto con un punteggio da 0 a 5 e, se lo desidera, aggiungere un commento di massimo 1000 caratteri.

2.2. Glossario dei termini

Termine	Descrizione	Collegamenti
Utente	Cliente registrato all'applicazione Svelasca	Carta, Indirizzo, Ordine
Plus	Utente iscritto a Svelasca Plus, servizio a pagamento annuale che garantisce consegne più rapide e gratuite	Entità figlia di Utente
Carta	Metodo di pagamento utilizzato per pagare gli ordini.	Utente, Ordine
Circuito	Circuito di pagamento utilizzato dalla carta di credito	Attributo di Carta
Indirizzo di consegna	Indirizzo fornito dall'utente per ricevere le spedizioni	Utente, ordine
Prezzo totale	Il totale dell'ordine, al netto delle spese di trasporto	Attributo di Ordine
Costo spedizione	Costo della spedizione che va aggiunto al totale. Se l'utente è plus viene registrato lo stesso per calcolare il guadagno del corriere padroncino	Attributo di spedizione
Giorno di consegna previsto	Data stimata per la consegna dell'ordine	Attributo di spedizione
Giorno e orario effettivi di consegna	Data e orario di quando avviene realmente la consegna	Attributo di spedizione
Azienda	Luogo di produzione e ritiro del prodotto	Ordine, prodotto
Azienda partner	Azienda che ha stretto un rapporto di partnership con Svelasca	Entità figlia di azienda
Orario	Giorni e orari di apertura di una azienda	Azienda
Corriere	Lavoratore che effettua la consegna degli ordini richiesti	Spedizione
Dipendente	Corriere alle dipendenze di Svelasca che percepisce uno stipendio	Entità figlia di corriere

Padroncino	Corriere autonomo con una propria partita IVA	Entità figlia di corriere
Recensione	Recensione relativa alla spedizione e al prodotto	Ordine, prodotto

2.3. Operazioni

OPERAZIONE	TIPO	FREQUENZA
Inserimento nuovo utente	S	5000 al giorno
Inserimento nuovo ordine	S	50000 al giorno
Rimozione ordine	S	800 al giorno
Inserimento nuovo prodotto	S	100 al giorno
Inserimento nuova azienda	S	10 al giorno
Visualizzare giorno e orari apertura azienda	L	50000 al giorno
Visualizzare il punteggio medio di un prodotto	L	80000 al giorno
Visualizzare lo storico degli acquisti di un utente	L	1000 al giorno
Visualizzare l'elenco dei prodotti di una categoria	L	200000 al giorno
Calcolare le entrate mensili di un padroncino	L	3000 al mese
Calcolare le entrate giornaliere di una azienda	L	12000 al giorno

3. Progettazione concettuale

3.1. Lista entità

Gli attributi sono considerati NOT NULL

- Utente:
 - ID: int primary key (poteva essere usata la mail come chiave ma avrebbe impedito all'utente di cambiare l'indirizzo)
 - Nome: varchar (50)
 - Cognome: varchar (50)
 - E-mail: varchar (60)
 - Password: varchar (12)
 - Telefono: varchar (20)

Utente si specializza nella sottocategoria Utente plus creando una generalizzazione parziale.
- Indirizzo:
 - Stato: varchar (2)
 - CAP: varchar (5) primary key
 - Città: varchar (20)
 - Via: varchar (30) primary key
 - Numero civico: varchar (10) primary key
 - Id Utente: int primary key
- Carta:
 - Circuito: varchar (30)
 - Intestatario: varchar (100)
 - Numero: char (20) primary key
 - Scadenza: date
- Ordine:
 - ID: int primary key
 - DataOraPagamento: datetime
 - Totale: decimal (9,2)
 - Corriere: char(16)

- Spedizione:
 - DataArrivo: date l'attributo può essere null
 - DataOraEffettiva: datetime l'attributo può essere null
 - ID_ordine: int primary key
 - Costo: decimal (6,2) l'attributo può essere null
- Azienda:
 - Nome: varchar (50)
 - Partita iva: varchar (15) primary key
 - E-mail: varchar (60)
 - Telefono: varchar (20)
 - Indirizzo è attributo composto:
 - Stato: varchar (2)
 - CAP: varchar (5)
 - Città: varchar (20)
 - Via: varchar (30)
 - Numero civico: varchar (10)

Azienda si specializza in una sottocategoria Azienda Partner creando una generalizzazione parziale.

- Prodotto:
 - Azienda: int primary key
 - nome: varchar(50) primary key
 - Prezzo: decimal (8,2)
 - Categoria: varchar (30) primary key
 - RecensioneMedia: decimal (3,2) l'attributo può essere null

Prodotto si specializza in una sottocategoria Prodotto Limited Edition creando una generalizzazione parziale.

- Orario:
 - ID_Azienda: int primary key
 - Data: date primary key
 - Orario_aperta: time primary key
 - Orario_chiusa: time, >Orario_aperta

- Categoria:
 - Nome: varchar (30) primary key

- Corriere
 - Nome: varchar (50)
 - Cognome: varchar (50)
 - Codice Fiscale: char (16) primary key
 - E-mail: varchar (60)
 - Telefono: varchar (20)
 - IBAN: char (27)

Corriere si specializza in due sottocategorie Dipendente e Padroncino creando una generalizzazione totale:

- Dipendente:
 - Stipendio: decimal (7,2)
- Padroncino
 - P.IVA: char (11)
- Recensione:
 - ID_prodotto: int primary key

- ID_ordine: int primary key
- Voto: int, >=0, <=5
- Testo: varchar (280) l'attributo può essere null

3.2. Lista delle relazioni

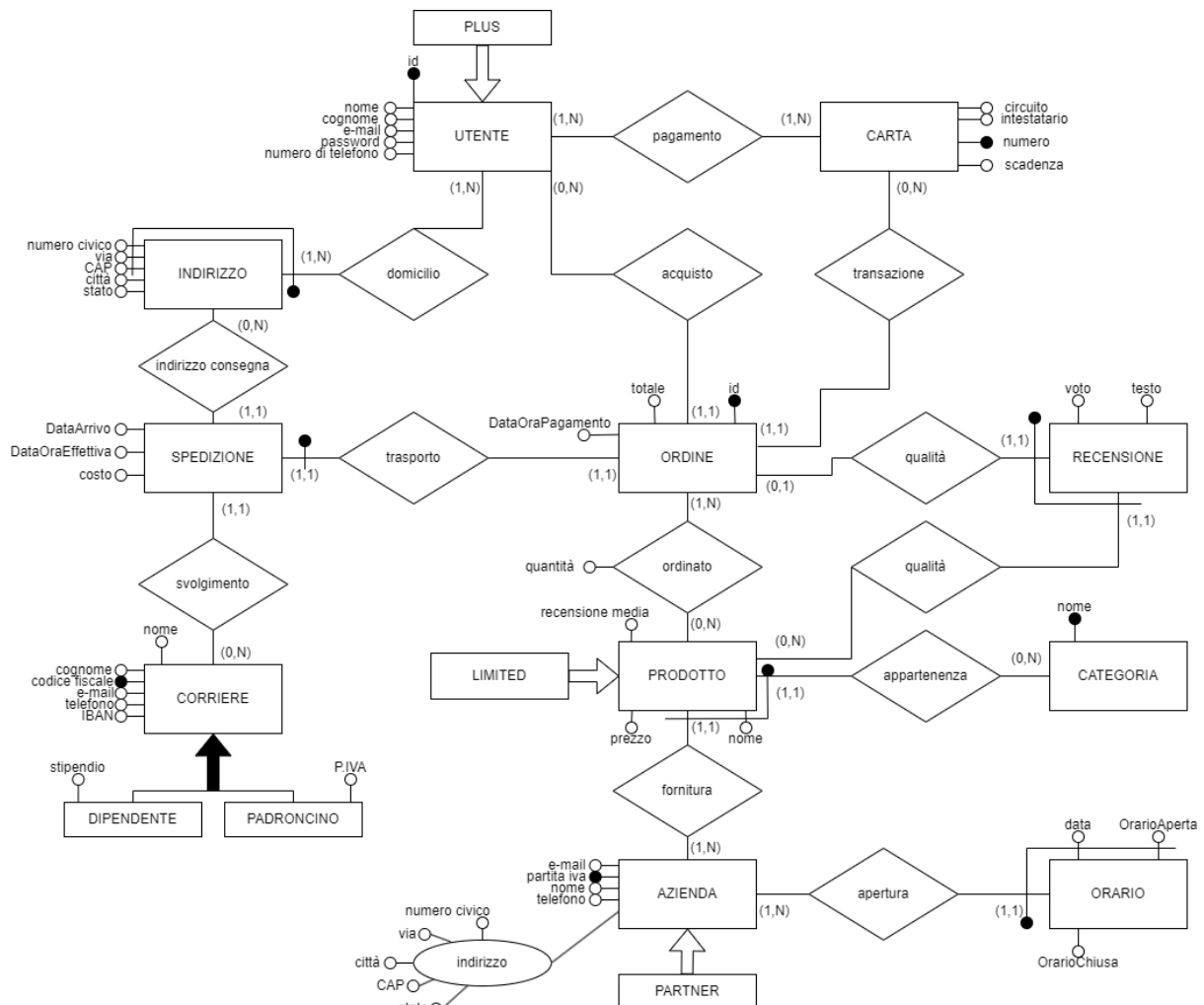
RELAZIONE	ENTITÀ	DESCRIZIONE
Pagamento	Carta (1,N) Utente (1,N)	Una carta può essere collegata ad uno o più utenti. Un utente deve registrare una o più carte.
Domicilio	Utente (1,N) Indirizzo (1,N)	Un utente deve inserire almeno un indirizzo di consegna. Un indirizzo appartiene ad uno o più utenti.
Acquisto	Utente (0,N) Indirizzo (1,1)	Un utente può avere zero o più ordini. Un ordine è fatto da un unico utente.
Transazione	Carta (0,N) Ordine (1,1)	Una carta può essere utilizzata per zero o più ordini. Un ordine è sempre pagato con una sola carta.
Ordinato	Prodotto (0,N) Ordine (1,N)	Un prodotto può essere ordinato zero o più volte. Un ordine è composto da uno o più prodotti.
	La relazione ha l'attributo quantità (int > 0) e indica la possibilità per un prodotto ordinato di averne una quantità superiore ad uno	
Trasporto	Ordine (1,1) Spedizione (1,1)	Un ordine è spedito una sola volta. Una spedizione si occupa di un singolo ordine.
Indirizzo consegna	Indirizzo (0,N) Spedizione (1,1)	Un indirizzo è usato zero o più volte come destinazione di spedizione. Una spedizione ha un solo indirizzo.
Svolgimento	Spedizione (1,1) Corriere (0,N)	Una spedizione è gestita da un unico corriere. Un corriere gestisce da zero a più spedizioni.
Valuta	Ordine (0,1) Recensione (1,1)	Un ordine può avere zero o una recensione. Una recensione si riferisce sempre ad un unico ordine.
Qualità	Prodotto (0,N) Recensione (1,1)	Un prodotto ha zero o più recensioni. Una recensione si riferisce sempre ad un unico prodotto.
Appartenenza	Categoria (0,N) Prodotto (1,1)	Una categoria ha zero o più prodotti. Un prodotto appartiene sempre ad un'unica categoria.
Fornitura	Prodotto (1,1) Azienda (1,N)	Un prodotto viene fornito da una sola azienda. Un'azienda fornisce almeno un prodotto.
Apertura	Azienda (1,N) Orario (1,1)	Un'azienda ha almeno un orario. Un orario è di un'unica azienda.

Vincoli di derivazione:

- L'attributo Totale dell'entità Ordine è dato dal prodotto dell'attributo prezzo dell'entità Prodotto per la Quantità che è attributo della relazione Ordinato

- L'attributo RecensioneMedia dell'entità Prodotto è dato dalla media aritmetica dell'attributo Voto appartenente all'entità Recensione

Schema Concettuale



4. Progettazione logica

4.1. Ristrutturazione

4.1.1. Analisi delle ridondanze

L'attributo **"totale"** dell'entità "ordine" è ridondante poiché può essere ricavato sommando la quantità di prodotti ordinati moltiplicata per il loro prezzo specifico.

Questo attributo viene utilizzato in alcune operazioni ovvero:

- 1) Calcolo dell'incasso giornaliero di una azienda: effettuato una volta al giorno dall'azienda
- 2) Creazione di un nuovo ordine: eseguito una sola volta quando viene concluso l'ordine
- 3) Visione storico degli ordini di un utente: viene effettuata raramente

Dunque, l'attributo "totale" occuperebbe memoria inutilmente, pertanto andrà eliminato.

L'attributo "RecensioneMedia" è ridondante perché può essere ricavato facendo la media aritmetica del voto dato nelle recensioni.

Concetto	Tipo	Volume
Prodotto	E	250000
Recensione	E	1000000
Qualità	R	1000000

- OPERAZIONE 1: inserimento di una recensione (7000 al giorno)
- OPERAZIONE 2: visualizzare la valutazione media di un prodotto (80000 al giorno)

ASSENZA DI RECENSIONEMEDIA

Operazione 1:

Un prodotto ha in media $1000000 / 250000 = 4$ recensioni

Concetto	Costrutto	Accessi	Tipo	Quantità
Qualità	Relazione	1	S	x 7000

Operazione 2:

Concetto	Costrutto	Accessi	Tipo	Quantità
Prodotto	Entità	4	L	x 80000

Considerando il costo degli accessi in scrittura il doppio di quelli in lettura otteniamo un costo giornaliero di: $14000 + 320000 = 334000$

PRESENZA DI RECENSIONEMEDIA

Operazione 1:

Concetto	Costrutto	Accessi	Tipo	Quantità
Qualità	Relazione	1	S	x 7000
Qualità	Relazione	4	L	x 7000
Prodotto	Entità	1	S	x 7000

Operazione 2:

Concetto	Costrutto	Accessi	Tipo	Quantità
Prodotto	Entità	1	L	x 80000

costo giornaliero di: $14000 + 28000 + 14000 + 80000 = 136000$

Analizzando i risultati ottenuti, conviene lasciare l'attributo "RecensioneMedia" rispetto al calcolo effettuato per ogni prodotto

4.1.2. Eliminazione delle generalizzazioni

GENERALIZZAZIONE	SOLUZIONE
Utente ⇌ Plus	L'entità Plus viene accorpata in Utente, usando al suo posto un attributo di tipo booleano per esprimere l'appartenenza dell'utente a Svelasca plus.
Azienda ⇌ Azienda Partner	L'entità Azienda esclusiva viene accorpata in Azienda, usando un attributo di tipo booleano.

Prodotto ⇌ Limited Edition	L'entità Limited Edition viene accorpata in Prodotto con un attributo booleano.
Corriere ← Dipendente, Padroncino	Le entità Dipendente, Padroncino vengono accorpate in Corriere. Viene utilizzato un attributo booleano dipendente. In caso risulti vero ciò implica che il corriere sia un dipendente, altrimenti indica che è padroncino. Si tratta infatti di una generalizzazione totale. Vengono aggiunti a Corriere gli attributi Stipendio e Partita Iva entrambi opzionali.

4.1.3. Accorpamento entità

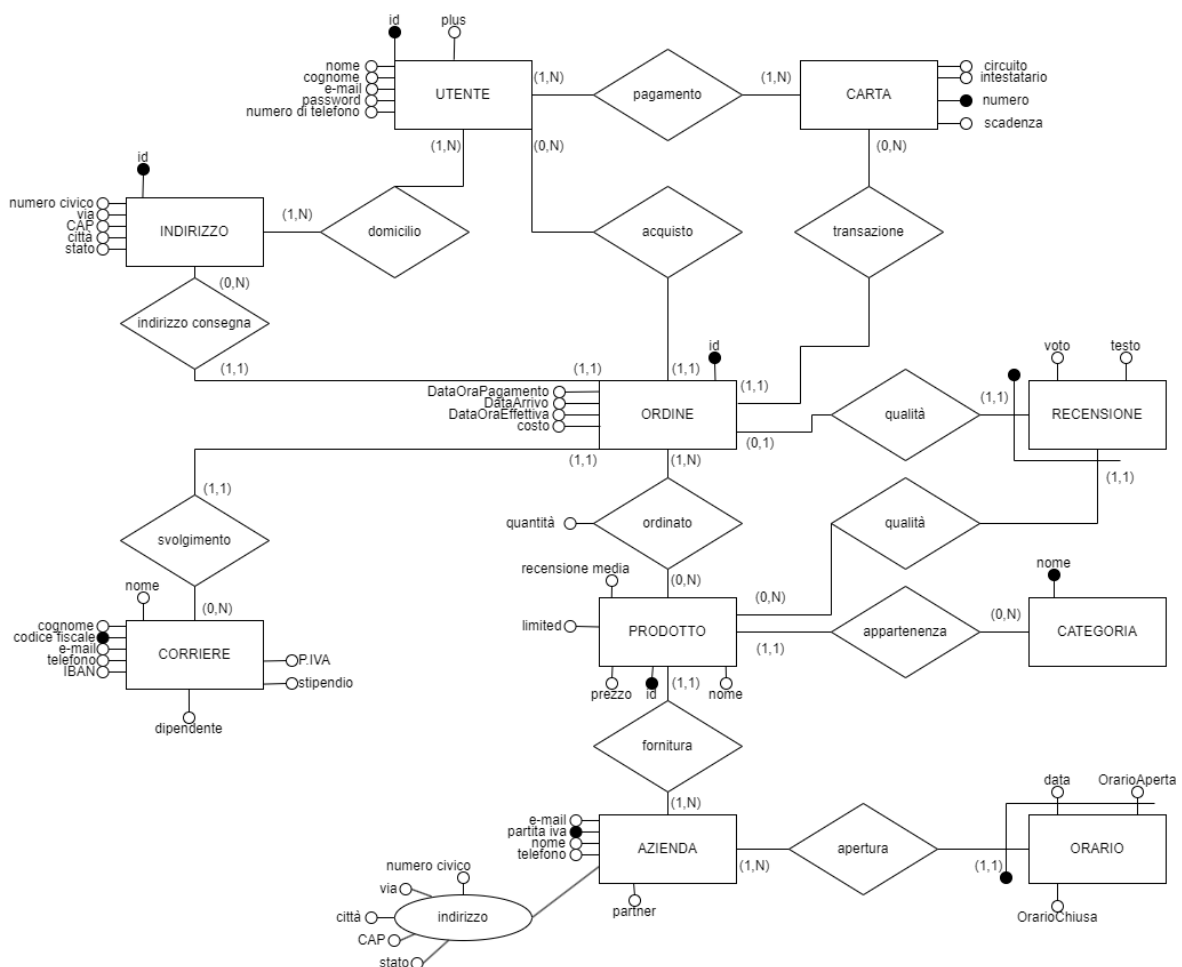
L'entità Spedizione viene accorpata nell'entità Ordine. Vengono quindi incorporati gli attributi costo, DataArrivo e DataOraEffettivo all'interno di Ordine.

4.1.4. Scelta degli identificatori primari

Per l'entità Indirizzo la chiave primaria ha utilizzato 4 attributi ciò comporta la necessità per l'entità Ordine di dover memorizzare tutti e 4 gli attributi come chiave esterna. Questa soluzione non risulta vantaggiosa per cui verrà utilizzato un ID come chiave primaria per l'entità Indirizzo.

L'entità Prodotto usa 3 attributi per la chiave primaria causando uno spreco di memoria quando questi vengono usati come chiave in Ordine e Recensione: verrà utilizzato un ID come chiave primaria di Prodotto.

Schema E-R Ristrutturato



4.2 Creazione delle tabelle (A → B indica che B è chiave esterna di A)

Utente (ID, Nome, Cognome, Mail, Password, Telefono, Plus)

Indirizzo (ID, Numero_Civico, Via, CAP, Città, Stato, Utente → Utente.ID)

Carta (Numero, Circuito, Intestatario, Scadenza)

Pagamento (Utente → Utente.ID, Carta → Carta.numero)

Azienda (Partita_IVA, Nome, Mail, Telefono, Stato, CAP, Via, Città, Numero_Civico, Partner)

Categoria (Nome)

Prodotto (ID, Azienda → Azienda.Partita_IVA, Categoria → Categoria.Nome, Nome, Prezzo, RecensioneMedia, Limited)

Orario (Azienda → Azienda.Partita_IVA, Data, Orario_Aperta, Orario_Chiusa)

Corriere (Codice_Fiscale, Nome, Cognome, Mail, Telefono, IBAN, Dipendente, Plva, Stipendio)

Recensione (Ordine → Ordine.ID, Prodotto → Prodotto.ID, Voto, Testo)

Ordine (ID, Utente → Utente.ID, Carta → Carta.numero, DataOraPagamento, DataArrivo, DataOraEffettiva, Totale, Costo, Corriere → Corriere.Codice_Fiscale)

Ordinato (Prodotto → Prodotto.ID, Ordine → Ordine.ID, Quantità)

5. Query e indici

5.1 Query

1. Visualizzare la partita IVA delle aziende che hanno incassato più di 7500 euro in ordine decrescente

```
1 - Visualizzare la partita IVA delle aziende che hanno incassato piu' di 7500 euro in ordine decrescente

Risultato:

azienda                                     totale_guadagni
526805160763184                             13407.91
080805508450412                             13153.88
316727298518131                             10865.97
597420858537600                             9658.02
275713044716836                             9076.61
472156754919827                             7951.77
466998191608686                             7529.94
```

	azienda character varying (15)	totale_guadagni numeric
1	526805160763184	13407.91
2	080805508450412	13153.88
3	316727298518131	10865.97
4	597420858537600	9658.02
5	275713044716836	9076.61
6	472156754919827	7951.77
7	466998191608686	7529.94

```
-- 1 Visualizzare la partita IVA delle aziende che hanno incassato più di 7500€ in ordine decrescente --
drop view if exists quantità_prodotti;
create view quantità_prodotti as
select ord.Quantità*p.Prezzo as totale_prodotto, p.Azienda
from Ordinato ord join Prodotto p on ord.Prodotto = p.ID;

select Azienda, sum(totale_prodotto) AS totale_guadagni
from quantità_prodotti
group by Azienda
having sum(totale_prodotto) > 7500
order by totale_guadagni desc;
```

2. I 5 prodotti più venduti in assoluto mostrandone la categoria e l'azienda

2 - I 5 prodotti più venduti in assoluto mostrandone la categoria e l'azienda

Risultato:

id	categoria	nome	quantità
70	Tuta	Bahringer, Stiedemann and Sporer	24
86	Jeans	Runolfsdottir, Leuschke and Kuvalis	22
230	Pigiama	Hermann-Parker	21
13	Reggiseno	Mertz Group	21
21	Calze	Hackett-Nitzsche	18

	id integer	nome character varying (50)	quantità bigint	categoria character varying (30)	nome character varying (50)
1	70	Bahringer, Stiedemann and Sporer	24	Tuta	Dabvine
2	86	Runolfsdottir, Leuschke and Kuvalis	22	Jeans	Yakidoo
3	230	Hermann-Parker	21	Pigiama	LiveZ
4	13	Mertz Group	21	Reggiseno	Rhyloo
5	21	Hackett-Nitzsche	18	Calze	Roomm

```
-- 2 I 5 prodotti più venduti in assoluto mostrandone la categoria e l'azienda --
DROP view IF EXISTS num_ord_prod;
CREATE view num_ord_prod AS
SELECT Prodotto, SUM(Quantità) as Quantità
FROM Ordinato
GROUP BY Prodotto;

SELECT p.ID, p.nome, nop.Quantità, p.Categoria, a.nome
FROM (Prodotto p JOIN num_ord_prod nop ON p.ID = nop.Prodotto)
JOIN Azienda a ON p.Azienda = a.Partita_IVA
ORDER BY nop.Quantità desc limit 5;
```

3. I 25 corrieri che hanno effettuato più consegne

3 - I 25 corrieri che hanno effettuato più consegne

Risultato:

codice_fiscale	nome	cognome	num_ord
02jHd2eIfeyD2Bc1	Saleem	Heselwood	7
1XAfm9D3jK8SGvcG	Joly	Eringey	7
2m4NLI6DAW1xMVGx	Allsun	Matchitt	7
3dEtS6iNf1vu26Lz	Lonee	McCumesky	7
3Z176EQ5ZWA6100J	Edmund	Muehler	7
53tmE8xx0EhYnmD	Cathy	Goom	7
6DUTN27W5f4fMB68	Charyl	Stiffell	7
7Cul1bfzmzWjg831	Yovonnda	Greensitt	7
7M1L372aGS16Ej9a	Vivia	Skupinski	7
7xLl7r907LPCE0I1	Donavon	Scimoni	7
8XILBS0oZJ7yvK2Y	Etta	Siviour	7
970r3IKc7WusSs8I	Lutero	Taig	7
9KWU4KT11Q5HRE15	Katine	Caress	7
9tia38cTdNM4153v	Hartwell	Walworth	7
aCbe01JX5njFtne8	Georgi	Gallichicchio	7
CTKgR96i7dI49r8f	Dore	Otham	7
hq3HI931ndt20G9x	Lucais	Seden	7
j74GN279J82DT3q8	Gibb	McGarvey	7
JmXr2j39N1Jc625W	Myrtie	Firpi	7
Jo3TSTo1SZm39MHP	Dore	Cumberledge	7
lHU5PLsBRMbrABXq	Grethel	Wynter	6
naEazA5vIB821Fi9	Leta	McLeish	6
OH7a5oD7Y02ujvw0	Amber	Polak	6
rzekaL6v4RJX2oGS	Martguerita	Madgett	6
TopgUdCb4x4tVUZR	Rickard	Ochterlonie	6

	codice_fiscale character	nome character varying (50)	cognome character varying (50)	num_ord bigint
1	02jHd2elfeyD2Bc1	Saleem	Heselwood	7
2	1XAfm9D3jK8SGvcG	Joly	Eringey	7
3	2m4NII6DAW1xMV...	Allsun	Matchitt	7
4	3dEtS6iNf1vu26Lz	Lonee	McCumesky	7
5	3Z176EQ5ZWA610...	Edmund	Muehler	7
6	53tmE8xx0EhYnmD	Cathy	Goom	7
7	6DUTN27W5f4fMB...	Charyl	Stiffell	7
8	7Cul1bfzmzWjg931	Yovonnda	Greensitt	7
9	7M1L372aGS16Ej9a	Vivia	Skupinski	7
10	7xLl7r907IPCE0I1	Donavon	Scimoni	7
11	8XILBS0oZJ7yvK2Y	Etta	Siviour	7
12	970r3IKc7WusSs8I	Lutero	Taig	7
13	9KWU4KT11Q5HRE...	Katine	Caress	7
14	9tia38cTdNM4153v	Hartwell	Walworth	7
15	aCbe01JX5njFtne8	Georgi	Gallichicchio	7
16	CTKgR96i7dI49r8f	Dore	Otham	7
17	hq3HI931ndt20G9x	Lucais	Seden	7
18	j74GN279J82DT3q8	Gibb	McGarvey	7
19	JmXr2j39N1Jc625W	Myrtie	Firpi	7
20	Jo3TSTo1SZm39MHP	Dore	Cumberledge	7
21	lHU5PLsBRMbrABXq	Grethel	Wynter	6
22	naEazA5vIB821Fi9	Leta	McLeish	6
23	OH7a5oD7Y02ujvw0	Amber	Polak	6
24	rzekaL6v4RJX2oGS	Martguerita	Madgett	6
25	TopgUdCb4x4tVUZR	Rickard	Ochterlonie	6

```
-- 3 I 25 corrieri che hanno effettuato più consegne --
DROP view IF EXISTS num_ord_cor;
CREATE view num_ord_cor AS
    SELECT Corriere, COUNT(*) as num_ord
    FROM Ordine
    GROUP BY Corriere;

SELECT c.codice_fiscale, c.Nome, c.Cognome, noc.num_ord
FROM Corriere c JOIN num_ord_cor noc ON c.codice_fiscale = noc.Corriere
ORDER BY noc.num_ord desc limit 25;
```

4. Visualizzare la mail degli utenti che hanno comprato almeno un prodotto limited edition

4 - Visualizzare la mail dei clienti che hanno comprato almeno un prodotto limited edition

Risultato:

nome	cognome	mail	num_ordini_limited
Randal	Barensen	rbarensen58@goo.ne.jp	2
Kiah	Springer	kspringere@weebly.com	1
Mick	Tytcomb	mtytcomb@cargo collective.com	1
Scotty	Weaben	sweabenk@typepad.com	1
Zebadiah	Tamlett	ztamlett@disqus.com	1
Valera	Mapes	vmapesw@example.com	1
Dedra	Ambrogi	dambrogil8@hc360.com	1
Amalle	Crosoer	acrosoer1f@go.com	1
Carlen	Cobelli	ccobelli1u@multiply.com	1
Libbey	Pirkis	lpirkis24@friendfeed.com	1
Mufinella	Turgoose	mturgoose26@newyorker.com	1
Gaylor	Tonner	gtonner2o@ow.ly	1
Elianora	Waugh	ewaugh2p@cloudflare.com	1
Brooke	Bister	bbister2q@miibeian.gov.cn	1
Liesa	Brookesbie	lbrookesbie30@nyu.edu	1
Patrizia	Sparling	psparling34@google.co.uk	1
Les	Nester	lnester3b@salon.com	1
Royal	Batcock	rbatcock3k@tmall.com	1
Jarvis	Hayball	jhayball3m@youtu.be	1
Torrie	Medlen	tmedlen3p@sakura.ne.jp	1
Rubin	Catcheside	rcatcheside3q@cyberchimps.com	1
Francoise	Splevins	fsplevins41@mozilla.org	1
Nicolai	Dartan	ndartan42@netlog.com	1
Ozzie	Fury	ofury48@dagondesign.com	1
Pat	Lauchlan	plauchlan4b@gnu.org	1
Geordie	Sellors	gsellors4d@dion.ne.jp	1
Avrom	Swanston	aswanston4k@wordpress.org	1
Egor	Carlile	ecarlile53@bravesites.com	1
Ania	Bines	abines5d@tumblr.com	1
Myrna	Seear	mseear7@redcross.org	1
Tallie	Tsar	ttsar5g@spiegel.de	1

	nome character varying (50)	cognome character varying (50)	mail character varying (60)	num_ordini_limited bigint
1	Randal	Barensen	rbarensen58@goo.ne.jp	2
2	Kiah	Springer	kspringere@weebly.com	1
3	Mick	Tytcomb	mtytcomb@cargo collective.com	1
4	Scotty	Weaben	sweabenk@typepad.com	1
5	Zebadiah	Tamlett	ztamlett@disqus.com	1
6	Valera	Mapes	vmapesw@example.com	1
7	Dedra	Ambrogi	dambrogil8@hc360.com	1
8	Amalle	Crosoer	acrosoer1f@go.com	1
9	Carlen	Cobelli	ccobelli1u@multiply.com	1
10	Libbey	Pirkis	lpirkis24@friendfeed.com	1
11	Mufinella	Turgoose	mturgoose26@newyorker.com	1
12	Gaylor	Tonner	gtonner2o@ow.ly	1
13	Elianora	Waugh	ewaugh2p@cloudflare.com	1
14	Brooke	Bister	bbister2q@miibeian.gov.cn	1
15	Liesa	Brookesbie	lbrookesbie30@nyu.edu	1
16	Patrizia	Sparling	psparling34@google.co.uk	1
17	Les	Nester	lnester3b@salon.com	1
18	Royal	Batcock	rbatcock3k@tmall.com	1
19	Jarvis	Hayball	jhayball3m@youtu.be	1
20	Torrie	Medlen	tmedlen3p@sakura.ne.jp	1
21	Rubin	Catcheside	rcatcheside3q@cyberchimps.com	1
22	Francoise	Splevins	fsplevins41@mozilla.org	1
23	Nicolai	Dartan	ndartan42@netlog.com	1
24	Ozzie	Fury	ofury48@dagondesign.com	1
25	Pat	Lauchlan	plauchlan4b@gnu.org	1
26	Geordie	Sellors	gsellors4d@dion.ne.jp	1
27	Avrom	Swanston	aswanston4k@wordpress.org	1
28	Egor	Carlile	ecarlile53@bravesites.com	1
29	Ania	Bines	abines5d@tumblr.com	1
30	Myrna	Seear	mseear7@redcross.org	1
31	Tallie	Tsar	ttsar5g@spiegel.de	1

```
-- 4 Visualizzare la mail degli utenti che hanno comprato almeno un prodotto limited edition --
DROP view IF EXISTS ordini_lim;
CREATE view ordini_lim AS
SELECT p.Id as cod_prodotto, p.nome, o.id as cod_ordine, o.utente
FROM (Ordinato x JOIN Prodotto p ON x.Prodotto = p.Id)
    JOIN Ordine o ON x.Ordine = o.id
WHERE p.limited = TRUE;

DROP view IF EXISTS tot_ordinilim_utente;
CREATE view tot_ordinilim_utente AS
SELECT u.id, COUNT(*) as num_ordini_limited
FROM Utente u JOIN ordini_lim os ON u.id = os.Utente
GROUP BY u.id;

SELECT u.nome, u.cognome, u.mail, tou.num_ordini_limited
FROM Utente u JOIN tot_ordinilim_utente tou ON u.id = tou.id
ORDER BY tou.num_ordini_limited DESC;
```

5. Recensioni positive (> 3) di ordini consegnati dopo le 12:00 del 17 Agosto 2023

5 - Recensioni positive (> 3) di ordini consegnati dopo le 12:00 del 17 Agosto 2023

Risultato:

prodotto	voto	testo	dataoraeffettiva
5	4	ok	2023-08-17 14:35:13
9	5	ok	2023-08-17 12:03:46
14	4	ottimo	2023-08-17 19:49:35
16	4	buono	2023-08-17 13:38:14
18	4	ottimo	2023-08-17 15:22:30
30	4	ottimo	2023-08-17 12:01:36
32	4	ok	2023-08-17 13:49:11
33	5	ok	2023-08-17 13:53:34
42	5	ottimo	2023-08-17 13:27:02
45	5	buono	2023-08-17 12:37:39
47	5	buono	2023-08-17 14:21:17
51	4	buono	2023-08-17 15:32:50
56	5	ok	2023-08-17 15:57:12
60	4	ok	2023-08-17 14:24:37
62	4	ok	2023-08-17 15:24:01
63	5	mi piace	2023-08-17 15:15:15
64	5	buono	2023-08-17 21:17:20
70	4	ottimo	2023-08-17 14:47:23
73	4	mi piace	2023-08-17 18:34:58
80	5	ok	2023-08-17 18:39:52
83	5	ottimo	2023-08-17 13:26:41
90	4	ottimo	2023-08-17 12:10:40
91	5	mi piace	2023-08-17 18:47:42
96	4	ottimo	2023-08-17 18:33:37
99	4	buono	2023-08-17 13:09:05

	prodotto integer	voto integer	testo character varying (280)	dataoraeffettiva timestamp without time zone
1	5	4	ok	2023-08-17 14:35:13
2	9	5	ok	2023-08-17 12:03:46
3	14	4	ottimo	2023-08-17 19:49:35
4	16	4	buono	2023-08-17 13:38:14
5	18	4	ottimo	2023-08-17 15:22:30
6	30	4	ottimo	2023-08-17 12:01:36
7	32	4	ok	2023-08-17 13:49:11
8	33	5	ok	2023-08-17 13:53:34
9	42	5	ottimo	2023-08-17 13:27:02
10	45	5	buono	2023-08-17 12:37:39
11	47	5	buono	2023-08-17 14:21:17
12	51	4	buono	2023-08-17 15:32:50
13	56	5	ok	2023-08-17 15:57:12
14	60	4	ok	2023-08-17 14:24:37
15	62	4	ok	2023-08-17 15:24:01
16	63	5	mi piace	2023-08-17 15:15:15
17	64	5	buono	2023-08-17 21:17:20
18	70	4	ottimo	2023-08-17 14:47:23
19	73	4	mi piace	2023-08-17 18:34:58
20	80	5	ok	2023-08-17 18:39:52
21	83	5	ottimo	2023-08-17 13:26:41
22	90	4	ottimo	2023-08-17 12:10:40
23	91	5	mi piace	2023-08-17 18:47:42
24	96	4	ottimo	2023-08-17 18:33:37
25	99	4	buono	2023-08-17 13:09:05

```
--5 Recensioni positive (> 3) di ordini consegnati dopo le 12:00 del 17 Agosto 2023 --
SELECT r.Prodotto, r.Voto, r.Testo, o.DataOraEffettiva
FROM Ordine o JOIN Recensione r ON o.ID = r.Ordine
WHERE o.DataOraEffettiva > '2023-08-17 12:00:00' and r.Voto > 3;
```

5.2 Indici

Le recensioni di un prodotto vengono visualizzate spesso dagli utenti in fase di lettura, in quanto vengono visualizzate nella pagina relativa al prodotto. L'inserimento di una recensione è un evento più raro. Pertanto, ipotizzando uno sviluppo massivo del nostro servizio è stato deciso di indicizzare gli attributi Prodotto e Voto nell'indice Prodotti_recensioni.

```
create index Prodotti_recensioni on Recensione (Prodotto, Voto);
```

6 Codice C++

6.1 Descrizione dell'utilizzo del codice

Il codice C++ per l'esecuzione delle query si trova nel file Codice.cpp, il quale deve essere compilato per mezzo del codice

```
g++ Codice.cpp -L dependencies\lib -lpq -o Codice
```

Nella cartella dependencies sono presenti i file necessari alla compilazione del programma.

Per eseguire il programma è sufficiente eseguire il file eseguibile del programma compilato.

Il programma chiederà quale query si desidera visualizzare, inserendo un numero da 1 a 5 sarà possibile visualizzare le serie corrispondenti.

6.2 Documentazione del codice

```
char conninfo [250];

sprintf ( conninfo , " user =%s password=%s dbname=%s hostaddr=%s port=%d",
PG_USER , PG_PASS , PG_DB , PG_HOST , PG_PORT ) ;

PGconn * conn = PQconnectdb ( conninfo );
```

Ritorna una connessione al database utilizzando le credenziali di accesso definite come parametri. Se la connessione non va a buon fine mostra un messaggio di errore altrimenti mostra un messaggio di corretta connessione

```
PGresult * res ;

res = PQexec ( conn , "select Azienda, sum(totale_prodotto)
```

La query viene passata come stringa e ne ritorna il risultato

```
void print(PGresult * res)
```

Funzione che stampa il risultato res di una query