



Università degli Studi di Salerno



Dipartimento di Ingegneria dell'Informazione ed Elettrica e
Matematica Applicata

Corso di Laurea in Ingegneria Informatica

Basi di Dati 2021/2022 **Canale A-H**

Project Work

Traccia N. 1 – Coltivatori e allevatori

Gruppo n. **01 – AH**

WP	Cognome e Nome	Matricola	e-mail	Responsabile
1	Amendola Miriam	0612704718	m.amendola21@studenti.unisa.it	
2	Battipaglia Valerio	0612704828	v.battipaglia12@studenti.unisa.it	
3	Dell'Orto Giuseppe Maria	0612704679	g.dellorto1@studenti.unisa.it	
4	Caso Antonio	0612704841	a.caso35@studenti.unisa.it	X

Anno accademico 2021-2022

Sommario

1. Descrizione della realtà di interesse	3
1.1. Analisi della realtà di interesse	3
2. Analisi delle specifiche	4
2.1. Glossario dei termini	4
2.2. Strutturazione dei requisiti in frasi	4
2.2.1. Frasi di carattere generale	4
2.2.2. Frasi relative a Viticoltore	4
2.2.3. Frasi relative a Vigneto	4
2.2.4. Frasi relative a Prodotto	4
2.2.5. Frasi relative ad Ordine	5
2.2.6. Frasi relative ad Abbonamento	5
2.2.7. Frasi relative a Cliente (specializzazione)	5
2.3. Identificazione delle operazioni principali	5
3. Progettazione Concettuale	6
3.1. Schema Concettuale	6
3.1.1. Note sullo schema E-R	6
3.2. Design Pattern	7
3.2.1. Pattern Evoluzione di concetto	7
3.3. Dizionario dei Dati	8
3.4. Regole Aziendali	11
4. Progettazione Logica	12
4.1. Ristrutturazione Schema Concettuale	12
4.1.1. Analisi delle Prestazioni	12
4.1.1.1. Tavola dei volumi	12
4.1.1.2. Tavola delle operazioni	12
4.2. Analisi delle ridondanze	13
4.2.1. Analisi della ridondanza 1: Vendita	14
4.2.1.1. Valutazione della ridondanza 1	14
4.2.2. Analisi della ridondanza 2: Subtotale	15
4.2.2.1. Valutazione della ridondanza 2	15
4.3. Eliminazione delle generalizzazioni	17
4.3.1. Generalizzazione <i>Utente</i>	17
4.3.2. Generalizzazione Prodotto	18
4.3.3. Generalizzazione Abbonamento	19
4.3.4. Generalizzazione Ordine	20
4.4. Partizionamento/Accorpamento Entità e Associazioni	21
4.5. Scelta degli identificatori principali	22
4.6. Schema ristrutturato finale	23

4.7.	Schema logico	24
4.8.	Documentazione dello schema logico	25
5.	Normalizzazione	26
	Script Creazione e Popolamento Database	28
6.	Query SQL	41
6.1.	Query con operatore di aggregazione e join: Abbonamenti per viticoltore	41
6.2.	Query nidificata complessa: Articoli correlati	42
6.3.	Query insiemistica: Numero vendite per prodotto	43
7.	Viste	44
7.1.	Vista Vendite Trimestrali	44
7.1.1.	Query con Vista: Vendite Trimestrali per viticoltore	44
8.	Trigger	45
8.1.	Trigger inizializzazione: <i>Partecipazione obbligatoria Composizione</i>	45
8.2.	Trigger per vincoli aziendali	46
8.2.1.	Trigger: Ricevuta	46

1. Descrizione della realtà di interesse

Titolo: **Coltivatori e allevatori**

Una start-up ha creato un sito web per la vendita dei prodotti da diversi coltivatori (orticoltori, frutticoltori, bachicoltori, funghicoltori, ecc.) e offre in particolare la possibilità agli utenti di adottare uno spazio di coltura attraverso un abbonamento annuale, ricevendo un set di prodotti ottenuti da quello spazio. La start-up ha richiesto la progettazione e la realizzazione di un database a supporto di tale servizio.

- Esistono diversi **coltivatori**. Per ognuno è necessario memorizzare almeno i dati anagrafici, la partita iva, i recapiti. Ogni coltivatore gestisce e cura più **spazi di coltura** (orti, pezzo d'orto, frutteto, pezzone, ecc.) di cui è proprietario. Ogni spazio di coltura produttivo è caratterizzato da un codice (univoco rispetto al coltivatore), dalla dimensione, dalla locazione.
- La locazione in cui si trova lo spazio è particolarmente importante per le caratteristiche dei prodotti. Pertanto, vanno descritti le caratteristiche della locazione, in termini di posizione geografica, esposizione, altitudine, etc. e in termini di tipologia di colture disponibili tenendo conto anche dei dintorni.
- Talvolta (in presenza di avversità) è necessario fare uso di insetticidi naturali, macerati ed estratti vegetali che hanno azione insetticida e permettono di intervenire con trattamenti non dannosi all'ambiente. Si vuole tenere traccia dell'uso di questi insetticidi naturali da parte del coltivatore.
- Periodicamente, il coltivatore raccoglie i prodotti da ogni spazio. Si vuole tenere traccia dei raccolti, della data e orario in cui avviene, della quantità raccolta da ogni spazio. Inoltre, devono essere indicati i prodotti raccolti. I prodotti così raccolti vengono poi confezionati in contenitori e preparati per la vendita. Questi **contenitori** possono essere acquistati dagli utenti. Sono caratterizzati dal tipo, dal peso/quantità, dal costo.
- Gli utenti possono acquistare i singoli prodotti oppure abbonarsi ad uno spazio di coltura. Nel caso di **acquisto** del singolo prodotto, sono di interesse i dati relativi all'acquisto (data, ora, costo, quantità, etc.). Per gli **abbonamenti**, è previsto un costo semestrale o annuale. L'abbonamento è caratterizzato dalla data in cui viene sottoscritto, dalla durata, dal costo, dai dati del cliente, dello specifico spazio a cui ci si abbona, dal set di prodotti che si riceveranno periodicamente.

Esempi di possibili variazioni: Per ogni progetto, è possibile scegliere il tipo di coltivatore (e.g., orticoltore, frutticoltore, bachicoltore, etc.), specializzando il database su una di tali coltivazioni. Conseguentemente, varieranno le caratteristiche della coltivazione, degli spazi, dei trattamenti, della raccolta, etc. Inoltre, è possibile curare con maggior dettaglio la sezione relativa alla coltivazione, oppure ai trattamenti, oppure alle vendite e abbonamenti.

1.1. Analisi della realtà di interesse

L'obiettivo del progetto è di realizzare un database per la gestione di un sito web mirato alla vendita di prodotti forniti da aziende vitivinicole. In particolare, verrà curata con maggior dettaglio la sezione relativa alle vendite e agli abbonamenti fornendo informazioni di supporto ai clienti della piattaforma, ad esempio la visualizzazione delle caratteristiche dei prodotti disponibili per la vendita, delle caratteristiche dei vigneti di provenienza, dei dati di riepilogo degli ordini (con maggiore attenzione per le informazioni legate all'emissione di ricevute e dello stato dell'ordine), degli abbonamenti attivi di un determinato cliente.

2. Analisi delle specifiche

Workpackage	Task	Responsabile
WP0	Analisi delle specifiche	Intero Gruppo

2.1. Glossario dei termini

	Termine	Descrizione	Sinonimi	Collegamenti
1	Viticoltore	Persona proprietaria di uno o più vigneti e intestataria di una azienda agricola oppure vitivinicola.	Coltivatore	Vigneto
2	Vigneto	Luogo adibito alla coltivazione di vite per la produzione di uva da tavola oppure uva da vino.	Spazio di coltura	Viticoltore, Abbonamento, Prodotto
3	Prodotto	Confezione di prodotti preparati per la vendita. Nel caso di vino e aceto si considera una bottiglia, mentre nel caso dell'uva da tavola si considera una cassetta di un certo peso.	Contenitore, Vino, Aceto, Uva	Vigneto, Ordine, Abbonamento
4	Ordine	Un acquisto avvenuto tramite sito web.	Acquisto	Prodotto
5	Abbonamento	Contratto per ricevere periodicamente un insieme di prodotti da un determinato vigneto.	-	Vigneto, Prodotto

Tabella 1. Glossario dei Termini

2.2. Strutturazione dei requisiti in frasi

2.2.1. Frasi di carattere generale

Una start-up ha creato un sito web per la vendita dei prodotti da diversi coltivatori (orticoltori, frutticoltori, bachicoltori, funghicoltori, ecc.) e offre in particolare la possibilità agli utenti di adottare uno spazio di coltura attraverso un abbonamento annuale, ricevendo un set di prodotti ottenuti da quello spazio. La start-up ha richiesto la progettazione e la realizzazione di un database a supporto di tale servizio.

2.2.2. Frasi relative a Viticoltore

Esistono diversi coltivatori. Per ognuno è necessario memorizzare almeno i dati anagrafici, la partita iva, i recapiti. Ogni coltivatore gestisce e cura più spazi di coltura (orti, pezzo d'orto, frutteto, pezzone, ecc.) di cui è proprietario. *Un viticoltore può anche iscriversi alla piattaforma come cliente.*

2.2.3. Frasi relative a Vigneto

Ogni spazio di coltura produttivo è caratterizzato da un codice (univoco rispetto al coltivatore), dalla dimensione, dalla locazione. Talvolta (in presenza di avversità) è necessario fare uso di insetticidi naturali, macerati ed estratti vegetali che hanno azione insetticida e permettono di intervenire con trattamenti non dannosi all'ambiente. Si vuole tenere traccia dell'uso di questi insetticidi naturali da parte del coltivatore.

2.2.4. Frasi relative a Prodotto

I prodotti così raccolti vengono poi confezionati in contenitori e preparati per la vendita. Questi contenitori possono essere acquistati dagli utenti. Sono caratterizzati dal tipo, dal peso/quantità,

dal costo. *È possibile vendere bottiglie di vino, di aceto (di vari formati) o cassette di uva di un certo peso. Di ogni prodotto si vogliono conoscere le caratteristiche di interesse per l'utente, come la gradazione alcolica oppure la denominazione.*

2.2.5. Frasi relative ad Ordine

Gli utenti possono acquistare i singoli prodotti oppure abbonarsi ad uno spazio di coltura. Nel caso di acquisto del singolo prodotto, sono di interesse i dati relativi all'acquisto (data, ora, costo, quantità, etc.). *È di interesse per il cliente determinare lo stato dell'ordine. Un ordine può trovarsi in tre stati: in lavorazione, spedito e consegnato. Per questi stati si vogliono memorizzare informazioni di interesse al cliente (il numero di tracking, il corriere, la data di spedizione e la data di consegna). Si vogliono inoltre mantenere le informazioni relative al documento fiscale che certifica l'avvenuto pagamento dell'ordine.*

2.2.6. Frasi relative ad Abbonamento

Per gli abbonamenti, è previsto un costo semestrale o annuale. L'abbonamento è caratterizzato dalla data in cui viene sottoscritto, dalla durata, dal costo, dai dati del cliente, dello specifico spazio a cui ci si abbona, dal set di prodotti che si riceveranno periodicamente.

2.2.7. Frasi relative a Cliente (specializzazione)

Del cliente è obbligatorio conoscere l'indirizzo di spedizione. Un cliente può effettuare acquisti, sottoscrivere abbonamenti e controllare le informazioni sullo stato dell'ordine.

2.3. Identificazione delle operazioni principali

Operazione 1: Inserisci un cliente specificando i dati anagrafici e l'indirizzo di spedizione (in media 1 volta al mese)

Operazione 2: Stampa i prodotti disponibili al momento con i relativi dettagli limitando la visualizzazione a 15 prodotti. (in media 500 volte al giorno)

Operazione 3: Esegui un acquisto inserendo i dati relativi all'ordine (circa 105 volte a settimana)

Operazione 4: Per ogni viticoltore, stampa il numero delle vendite trimestrali raggruppandole per prodotto. (in media 4 volte all'anno)

Operazione 5: Stampa i vini provenienti da una certa regione di una certa qualità e con una certa gradazione alcolica (in media 15 volte al mese)

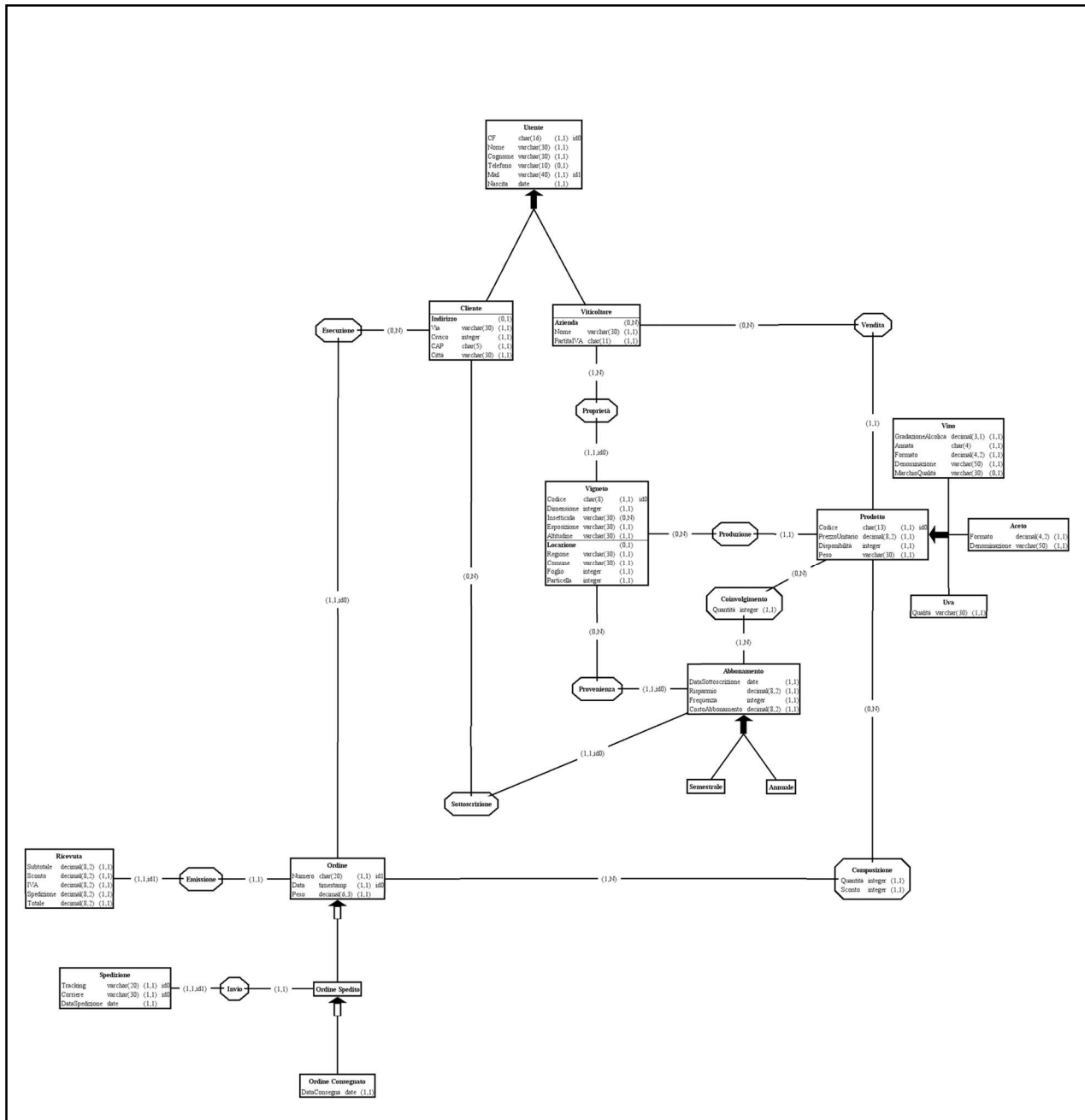
Operazione 6: Registra un nuovo abbonamento per un cliente su un certo vigneto (circa 500 volte all'anno)

Operazione 7: Stampa il riepilogo degli ordini di un cliente, visualizzando per ciascuno di essi lo stato e i dati della ricevuta. (circa 130 volte a settimana)

3. Progettazione Concettuale

Workpackage	Task	Responsabile
WP1	Progettazione Concettuale	Amendola Miriam

3.1. Schema Concettuale



3.1.1. Note sullo schema E-R

Gli attributi composti Indirizzo e Locazione sono obbligatori, anche se nello schema risultano essere opzionali. La generalizzazione Utente è sovrapposta.

3.2. Design Pattern

3.2.1. Pattern Evoluzione di concetto

Il pattern è stato applicato per semplificare la gestione dell'evoluzione dello stato dell'ordine. Un ordine inserito nel database inizialmente si trova in lavorazione, successivamente viene spedito in una certa data e tramite una certa spedizione, diventando un ordine spedito ed infine diventa un ordine consegnato e ne viene memorizzata la data di consegna.

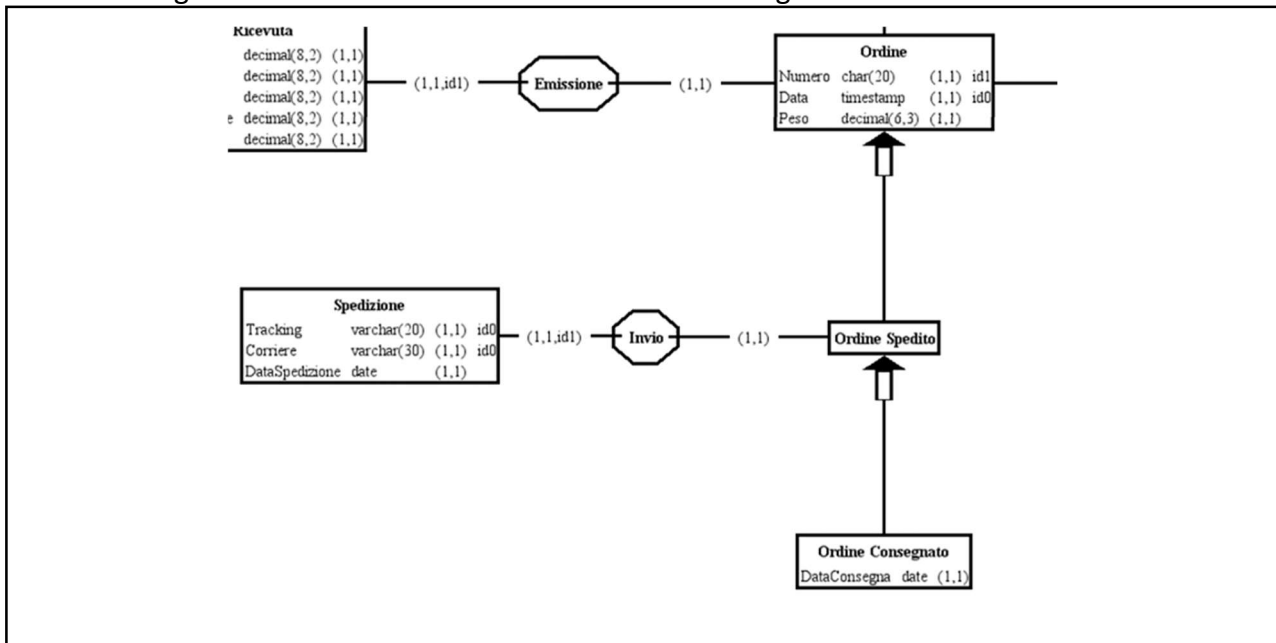


Figura 3 Schema successiva all'applicazione del Pattern Evoluzione di concetto.

3.3. Dizionario dei Dati

Entità	Descrizione	Attributi	Identificatore
Utente	Generico utente iscritto alla piattaforma web.	CF, Nome, Cognome, Telefono, Mail, Nascita	CF, Mail
Cliente	Specializzazione di Utente, in grado di acquistare uno o più prodotti oppure abbonarsi ad un vigneto.	Indirizzo(Via, Civico, CAP, Città)	CF, Mail
Viticoltore	Specializzazione di utente. Utente in possesso di uno o più vigneti e che mette in vendita prodotti.	Azienda(PartitaIVA, Nome)	CF, Mail
Vigneto	Spazio di coltura di proprietà di un viticoltore.	Codice, Dimensione, Insetticida, Altitudine, Esposizione, Locazione(Regione, Comune, Foglio, Particella)	(Codice, Vitiscoltore.CF)
Prodotto	Articolo acquistabile sulla piattaforma.	Codice, Prezzo Unitario, Disponibilità, Peso	Codice
Vino	Specializzazione di Prodotto.	GradazioneAlcolica, Annata, Formato, MarchioQualità, Denominazione	Codice
Aceto	Specializzazione di Prodotto.	Formato, Denominazione	Codice
Uva	Specializzazione di Prodotto.	Qualità	Codice
Ordine	Ordine relativo ad un acquisto eseguito tramite la piattaforma.	Numero, Data, Peso	Numero, (Data, Cliente.CF)
Ordine spedito	Specializzazione di ordine che rappresenta uno stadio evolutivo dell'ordine in cui esso è stato spedito.		Numero, (Data, Cliente.CF)
Ordine consegnato	Specializzazione di ordine che rappresenta uno	DataConsegna	Numero, (Data, Cliente.CF)

	stadio evolutivo dell'ordine in cui esso dopo essere stato spedito è stato anche consegnato.		
Abbonamento	Abbonamento attivo sottoscritto da un cliente per un certo vigneto e un certo insieme di prodotti.	Sottoscrizione, Risparmio, Frequenza, CostoAbbonamento	(Cliente.CF, Vigneto.Codice, Viticoltore.CF)
Annuale	Specializzazione di Abbonamento la cui tariffa viene liquidata con cadenza annuale.		Cliente.CF, Vigneto.Codice, Viticoltore.CF)
Semestrale	Specializzazione di Abbonamento la cui tariffa viene liquidata con cadenza ogni sei mesi.		Cliente.CF, Vigneto.Codice, Viticoltore.CF)
Spedizione	Storico delle spedizioni legate agli ordini.	Tracking, Corriere, DataSpedizione	Ordine.Numero
Ricevuta	Documento che certifica l'ordine, contenente informazioni su come è stato calcolato l'importo totale dovuto.	Subtotale, Sconto, IVA, Spedizione, Totale	Ordine.Numero

Tabella 2. Dizionario dei dati – Entità

Relazioni	Descrizione	Entità Coinvolte	Attributi
Coinvolgimento	Specifica quali prodotti sono l'oggetto dell'abbonamento.	Abbonamento, Prodotto	Quantità
Composizione	Indica che il dato prodotto fa parte dell'ordine con una certa quantità alla quale è stato applicato un certo sconto.	Ordine, Prodotto	Quantità, Sconto
Emissione	Collega un ordine alla ricevuta che contiene i dati riepilogativi del pagamento.	Ricevuta, Ordine	

Esecuzione	Specifica quale cliente ha effettuato l'ordine.	Ordine, Cliente	
Invio	Specifica quale spedizione si occupa dell'ordine.	Ordine, Spedizione	
Produzione	Indica il vigneto di provenienza del prodotto.	Vigneto, Prodotto	
Proprietà	Indica il proprietario del vigneto.	Viticolto, Vigneto	
Provenienza	Indica il vigneto cui si sceglie di abbonarsi.	Abbonamento, Vigneto	
Sottoscrizione	Specifica quale cliente ha richiesto l'abbonamento.	Abbonamento, Cliente	
Vendita	Specifica chi sia il produttore per il dato prodotto.	Viticolto, Prodotto	

Tabella 3. Dizionario dei dati - Relazioni

Workpackage	Task	Responsabile
WP4	Regole Aziendali	Caso Antonio

3.4. Regole Aziendali

Regole di Vincolo
<p>(RV1) Non è possibile inserire un utente non maggiorenne.</p> <p>(RV2) Non è possibile effettuare un ordine il cui peso è maggiore di 80kg.</p> <p>(RV3) Se si sceglie di registrare un abbonamento semestrale, si avrà uno sconto pari al 15% dell'importo totale, mentre se si sceglie un abbonamento annuale, si avrà uno sconto pari al 30% dell'importo totale.</p> <p>(RV4) L'abbonamento semestrale garantisce la consegna solo ogni 1, 2 o 3 mesi, mentre l'abbonamento annuale garantisce la consegna solo ogni 1, 2, 3, 4 o 6 mesi.</p>

Tabella 4. Regole di vincolo

Regole di derivazione
<p>(RD1) L'attributo Totale di Ricevuta è dato dalla somma dei suoi attributi Subtotale e Spedizione.</p> <p>(RD2) L'attributo Subtotale dell'entità Ricevuta è ricavabile a partire dalla somma degli attributi PrezzoUnitario delle occorrenze di Prodotto coinvolte nella relazione Composizione moltiplicate per l'attributo Quantità di Composizione, comprensivi di IVA e di sconto.</p> $subtotale = \sum (prezzo_{unitario} \cdot quantità + IVA_{calcolata} - sconto)$ <p>(RD3) Se il costo di un acquisto è maggiore di €50, la spedizione è gratuita, altrimenti è pari a €7.</p> <p>(RD4) L'attributo IVA di Ricevuta è ottenuto calcolando l'aliquota del subtotale destinata all'IVA. Se il prodotto è una istanza di Vino oppure Aceto, si assume che l'IVA sia al 22%:</p> $IVA = subtotale - subtotale \cdot 0.22$ <p>Se il prodotto è una istanza di Uva, si assume che l'IVA sia al 4%:</p> $IVA = subtotale - subtotale \cdot 0.04$ <p>(RD5) L'attributo Peso di Ordine è ricavabile a partire dalla somma degli attributi Peso delle occorrenze dell'entità Prodotto coinvolte nella relazione Composizione moltiplicate per la rispettiva quantità indicata nell'attributo Quantità di Composizione.</p> <p>(RD6) Quando viene eseguito un nuovo acquisto, l'attributo Disponibilità di ciascun Prodotto coinvolto nella relazione Composizione deve diminuire in accordo alla quantità acquistata.</p> <p>(RD7) L'attributo CostoAbbonamento di Abbonamento è dato dal PrezzoUnitario di ciascun Prodotto (comprensivo di IVA) coinvolto nella relazione Coinvolgimento moltiplicato per l'attributo Quantità e la sua frequenza relativa.</p> $costoabbonamento_{base} = \left(\sum (prezzo_{unitario} \cdot quantità + IVA_{calcolata}) \cdot \frac{tipologia}{frequenza} \right)$ $costoabbonamento = costoabbonamento_{base} - costoabbonamento_{base} \cdot \frac{sconto}{100}$ $sconto = \begin{cases} 15 & \text{semestrale} \\ 30 & \text{annuale} \end{cases}$

Tabella 5. Regole di derivazione

4. Progettazione Logica

Workpackage	Task	Responsabile
WP2	Progettazione Logica	Battipaglia Valerio

4.1. Ristrutturazione Schema Concettuale

4.1.1. Analisi delle Prestazioni

4.1.1.1. Tavola dei volumi

Concetto	Tipo	Volume
Utente	E	370
Viticoltore	E	20
Cliente	E	350
Vigneto	E	40
Prodotto	E	60
Vino	E	40
Aceto	E	10
Uva	E	10
Ordine	E	5500
Ordine Spedito	E	60
Spedizione	E	5500
Ordine Consegnato	E	5440
Ricevuta	E	5500
Abbonamento	E	570
Vendita	R	60
Proprietà	R	40
Produzione	R	60
Provenienza	R	570
Sottoscrizione	R	570
Coinvolgimento	R	7110
Composizione	R	22000
Esecuzione	R	5500
Emissione	R	5500
Invio	R	60

Tabella 6. Tavola dei volumi

4.1.1.2. Tavola delle operazioni

Operazione	Tipo	Frequenza
Operazione 1: Inserisci un cliente specificando i dati anagrafici e l'indirizzo di spedizione	I	12 volte l'anno
Operazione 2: Stampa i prodotti disponibili al momento con i relativi	I	182500 volte l'anno

dettagli limitando la visualizzazione a 15 prodotti.		
Operazione 3: Esegui un acquisto inserendo i dati relativi all'ordine	I	5460 volte l'anno
Operazione 4: Per ogni viticoltore, stampa il numero delle vendite trimestrali raggruppandole per prodotto.	I	1 volte l'anno
Operazione 5: Stampa i vini provenienti da una certa regione di una certa qualità e con una certa gradazione alcolica	B	780 volte l'anno
Operazione 6: Registra un nuovo abbonamento per un cliente su un certo vigneto	I	26000 volte l'anno
Operazione 7: Stampa il riepilogo degli ordini di un cliente, visualizzando per ciascuno di essi lo stato e i dati della ricevuta.	I	6760 volte l'anno
Operazione 8: Inserimento di un nuovo prodotto.	I	50 volte l'anno

Tabella 7. Tavola delle operazioni

4.2. Analisi delle ridondanze

- Ridondanza 1: Vendita (Prodotto, Viticoltore). L'associazione Vendita è formata dalla composizione delle associazioni Proprietà e Produzione.
TIPO: Associazione derivabile dalla composizione di altre associazioni.
- Ridondanza 2: Subtotale(Ordine). L'attributo Subtotale di Ordine si ottiene a partire dal prezzo unitario di ogni prodotto coinvolto nell'associazione Composizione e dall'attributo Quantità di Composizione. *TIPO:* Attributo derivabile da attributi di altre entità

4.2.1. Analisi della ridondanza 1: Vendita

- **Operazione 4:** Per ogni viticoltore, stampa il numero delle vendite trimestrali raggruppandole per prodotto.

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
VITICOLTORE	E	1	L
VENDITA	R	3	L
PRODOTTO	E	3	L
COMPOSIZIONE	R	1098	L

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
VITICOLTORE	E	1	L
PROPRIETA'	R	2	L
VIGNETO	E	2	L
PRODUZIONE	R	2	L
PRODOTTO	E	2	L
COMPOSIZIONE	R	732	L

- **Operazione 8:** Inserimento di un nuovo prodotto nel database

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
PRODOTTO	E	1	S
PRODUZIONE	R	1	S
VENDITA	R	1	S

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
PRODOTTO	E	1	S
PRODUZIONE	R	1	S

4.2.1.1. Valutazione della ridondanza 1

Dopo aver analizzato le operazioni che coinvolgono la ridondanza si osserva che, con il carico considerato:

- In presenza di ridondanza il costo della operazione 4 è di 1105 accessi, mentre il costo dell'operazione 8 è di $6 * 50 = 300$ accessi, per un totale di 1405 accessi.
- L'occupazione di memoria è di circa $16 * 60 = 960$ byte (essendo tradotta come l'attributo CF di Viticoltore all'interno di prodotto)
- In assenza di ridondanza il costo dell'operazione 4 è di 714 accessi, mentre il costo dell'operazione 8 è di $4 * 50 = 200$ accessi, per un totale di 914 accessi.

Il costo delle operazioni in presenza di ridondanza è maggiore di quello in assenza di ridondanza; pertanto, si decide di rimuovere la ridondanza dallo schema.

4.2.2. Analisi della ridondanza 2: Subtotale

- **Operazione 7: Stampa il riepilogo degli ordini di un cliente, visualizzando per ciascuno di essi lo stato e i dati della ricevuta.**

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
CLIENTE	E	1	L
ESECUZIONE	R	15	L
ORDINE	E	15	L
EMISSIONE	R	15	L
RICEVUTA	E	15	L

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
CLIENTE	E	1	L
ESECUZIONE	R	15	L
ORDINE	E	15	L
COMPOSIZIONE	R	60	L
PRODOTTO	E	60	L

- **Operazione 3: Esegui un acquisto inserendo i dati relativi all'ordine.**

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
COMPOSIZIONE	R	4	S
PRODOTTO	E	4	L
PRODOTTO	E	4	S
ORDINE	E	1	S
COMPOSIZIONE	R	4	L
EMISSIONE	R	4	S
RICEVUTA	E	4	S

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
COMPOSIZIONE	R	4	S
PRODOTTO	E	4	L
PRODOTTO	E	4	S
ORDINE	E	1	S

4.2.2.1. Valutazione della ridondanza 2

Dopo aver analizzato le operazioni che coinvolgono la ridondanza si osserva che, con il carico considerato:

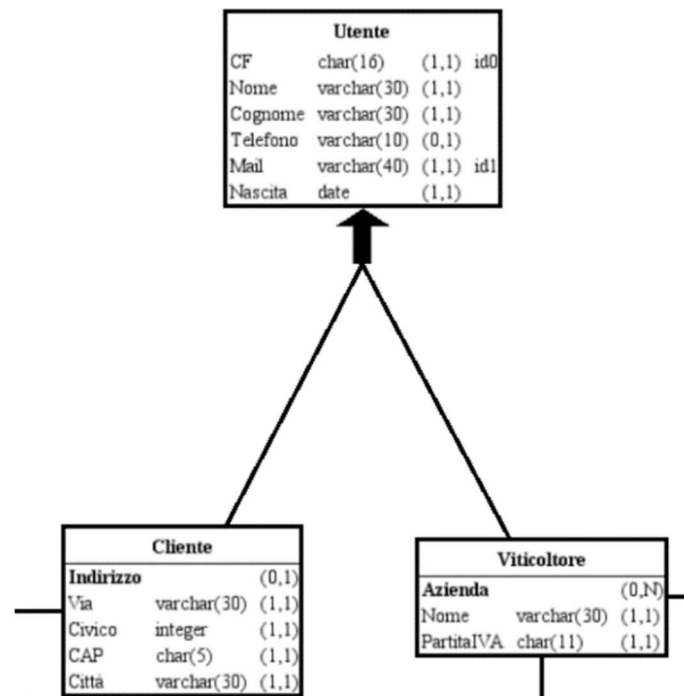
- *In presenza di ridondanza il costo della operazione 3 è di $42 \cdot 5640 = 236.880$ accessi, mentre il costo dell'operazione 7 è di $61 \cdot 5640 = 344.040$ accessi, per un totale di 580.920 accessi.*
- *L'occupazione di memoria è di $5500 \cdot 5 = 27500$*

- In assenza di ridondanza il costo dell'operazione 3 è di $151 \cdot 6760 = 1.020.760$ accessi, mentre il costo dell'operazione 7 è di $22 \cdot 6760 = 148720$ accessi, per un totale di 1.169.480 accessi.

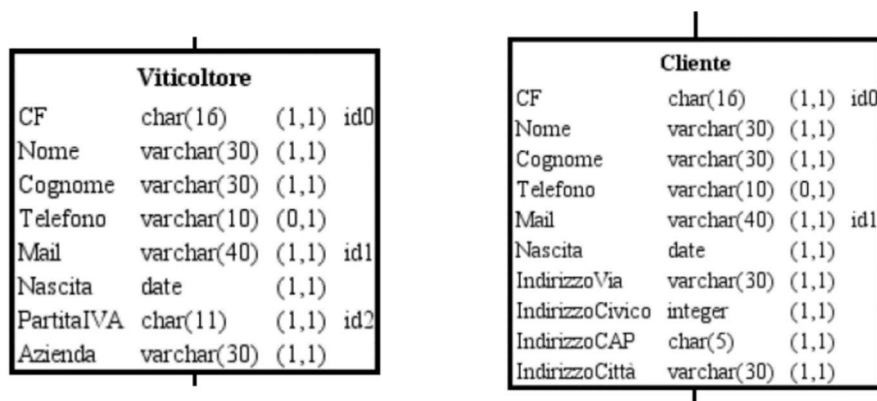
Il costo delle operazioni in assenza di ridondanza è maggiore di quello in presenza di ridondanza; pertanto, si decide di mantenere la ridondanza.

4.3. Eliminazione delle generalizzazioni

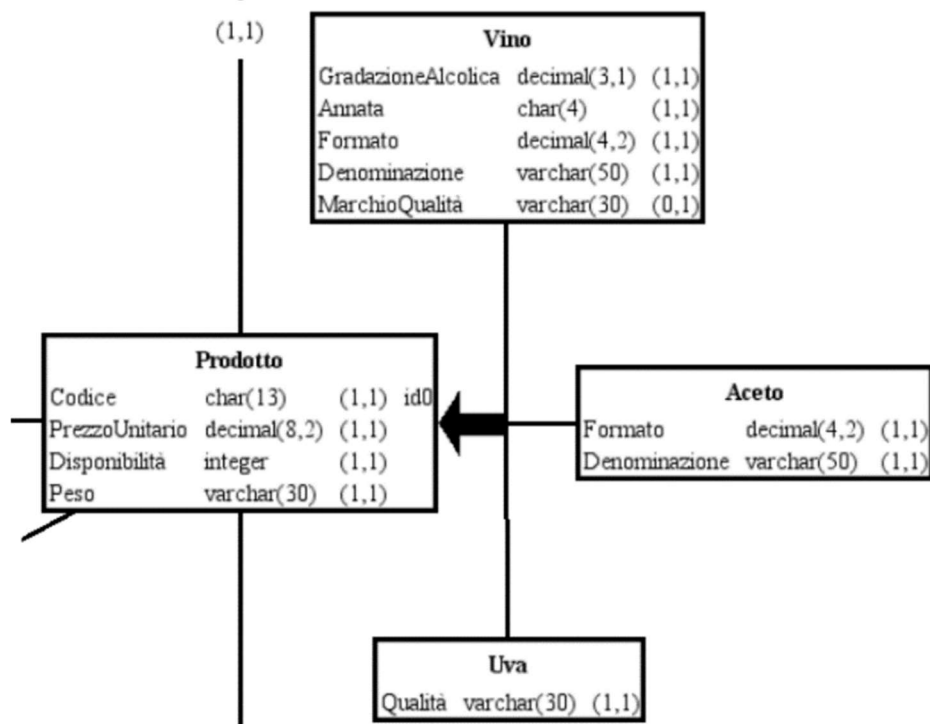
4.3.1. Generalizzazione *Utente*



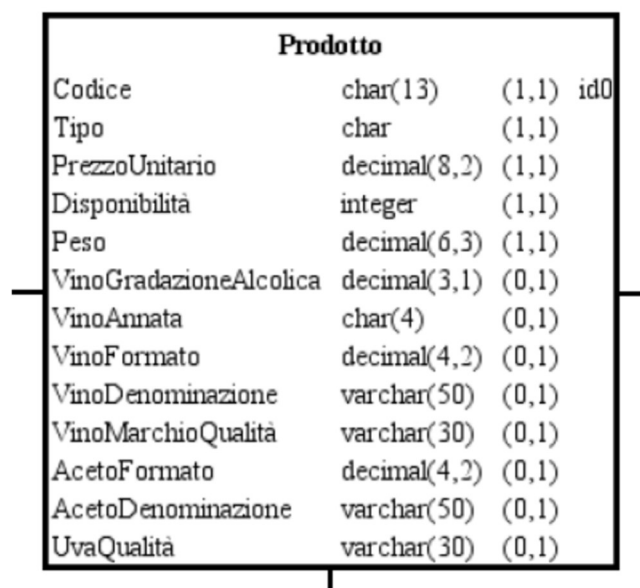
La generalizzazione *Utente* viene eliminata accorpendo l'entità genitore *Utente* nelle entità figlie, *Cliente* e *Viticoltore*. Questa scelta è stata motivata dal fatto che le operazioni fanno distinzione tra il cliente e il viticoltore e anche dal fatto che le entità figlie sono coinvolte in più di una relazione con altre entità.



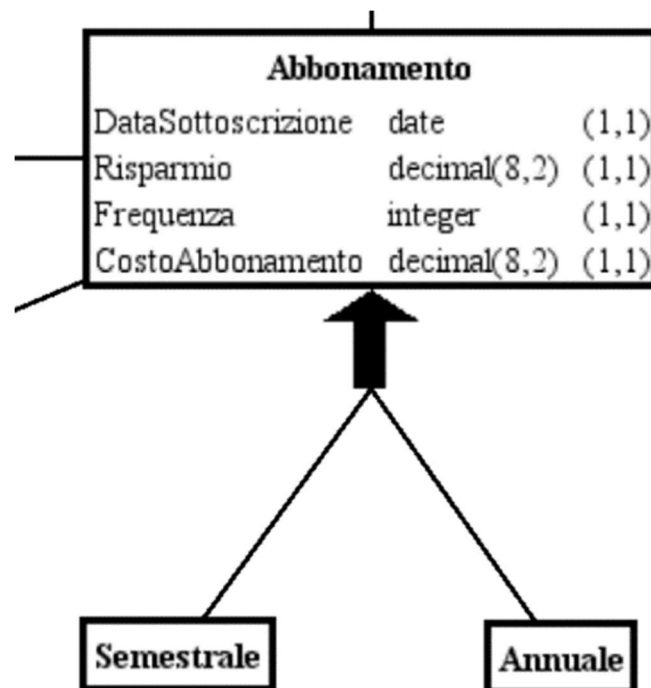
4.3.2. Generalizzazione Prodotto



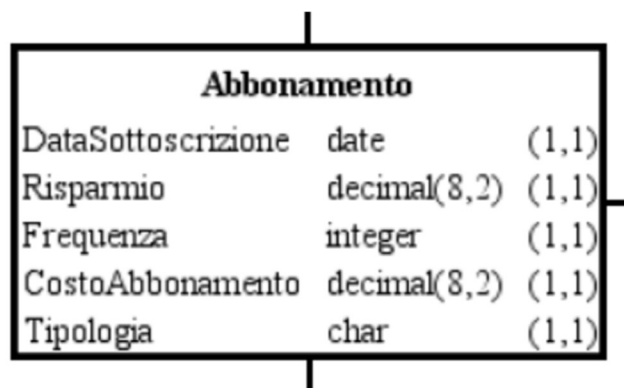
La generalizzazione *Prodotto* viene eliminata accorpendo le entità figlie *Vino*, *Aceto* e *Uva* nell'entità padre *Prodotto*. Questa scelta è stata motivata dal fatto che le operazioni riguardano i prodotti nel loro insieme senza fare differenza in base alla categoria.



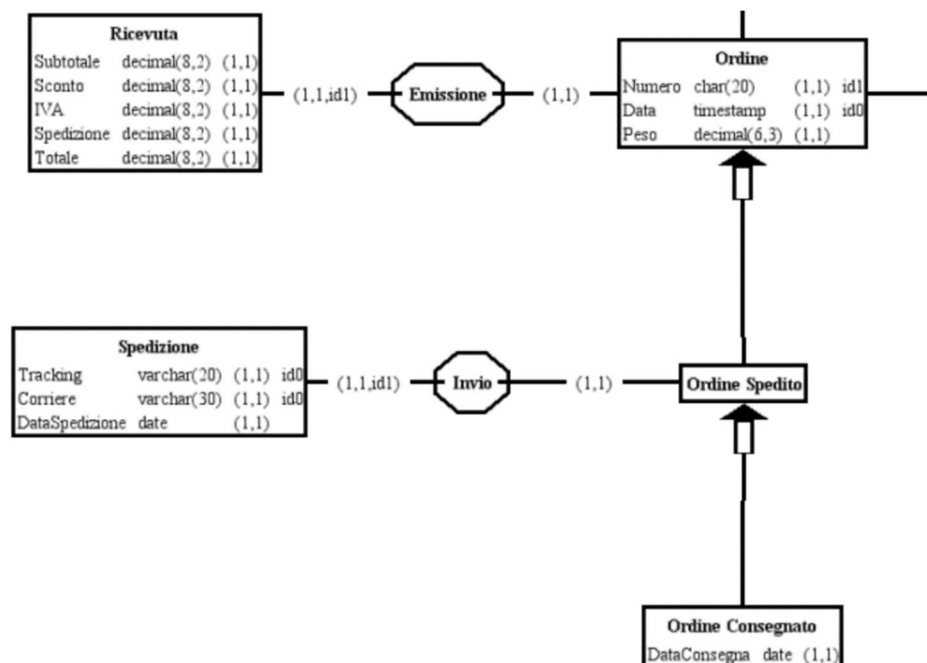
4.3.3. Generalizzazione Abbonamento



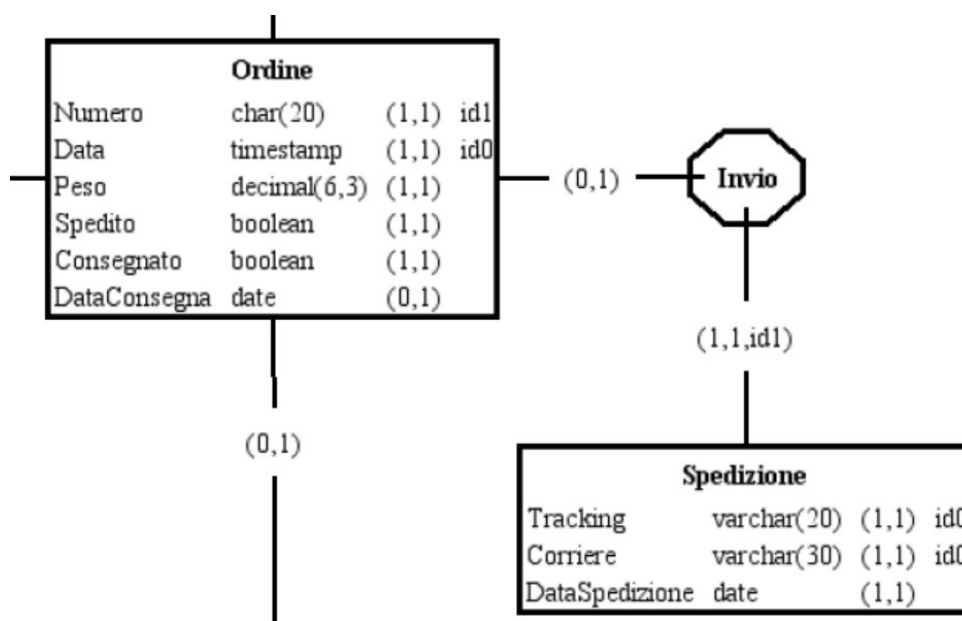
La generalizzazione Abbonamento viene eliminata accorpendo le entità figlie Semestrale e Annuale nell'entità padre Abbonamento, aggiungendo un attributo tipologia. Questa scelta è stata motivata dal fatto che le operazioni riguardano gli abbonamenti nel loro insieme senza fare differenza in base alla categoria.



4.3.4. Generalizzazione Ordine

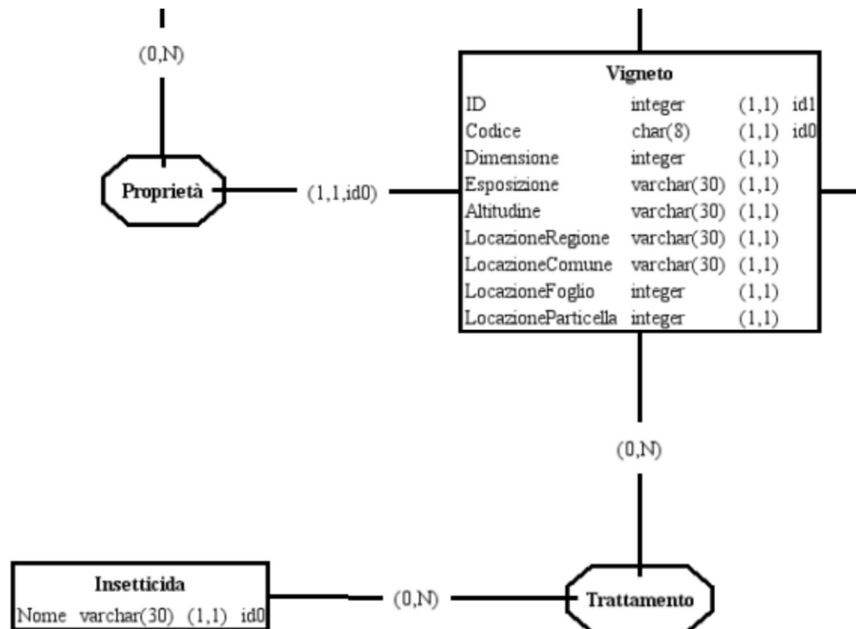


La generalizzazione Ordine viene eliminata accorpendo l'entità figlia Ordine Spedito e aggiungendo l'attributo booleano Spedito e successivamente accorpendo l'entità Ordine Consegnato nell'entità padre Ordine, aggiungendo un attributo booleano Consegnato. Si può avere consegnato solo se prima era spedito. Questa scelta è stata motivata dal fatto che essendo la generalizzazione parziale non era possibile impiegare la seconda soluzione ovvero quella che prevede il partizionamento del padre nelle figlie.

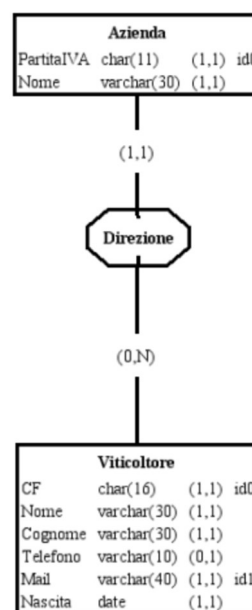


4.4. Partizionamento/Accorpamento Entità e Associazioni

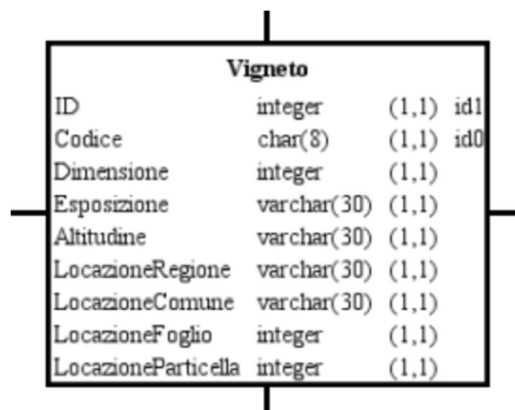
L'attributo multivalore Insetticida di Vigneto è stato reificato come in figura:



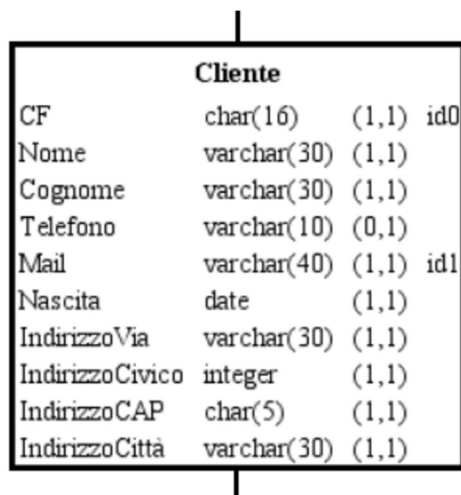
L'attributo composto e multivalore Azienda è stato reificato come in figura:



L'attributo composto Locazione è stato scomposto in attributi semplici come in figura:



L'attributo composto Indirizzo è stato scomposto in attributi semplici come in figura:



4.5. Scelta degli identificatori principali

Per le entità *Cliente* e *Viticoltore*, entrambe dotate di due identificatori, si sceglie di adottare come identificatore principale il Codice Fiscale, essendo un identificatore impiegato frequentemente e di dimensione fissa rispetto all'Email.

Per l'entità *Vigneto*, l'identificatore misto, composto dal codice fiscale del proprietario e da un codice alfanumerico di dimensione fissa, viene anch'esso sostituito da un codice numerico denominato ID perché l'identificatore misto verrebbe tradotto come una chiave composta da due attributi, la quale verrebbe "propagata" anche in altre relazioni, diminuendo l'efficienza dello schema logico.

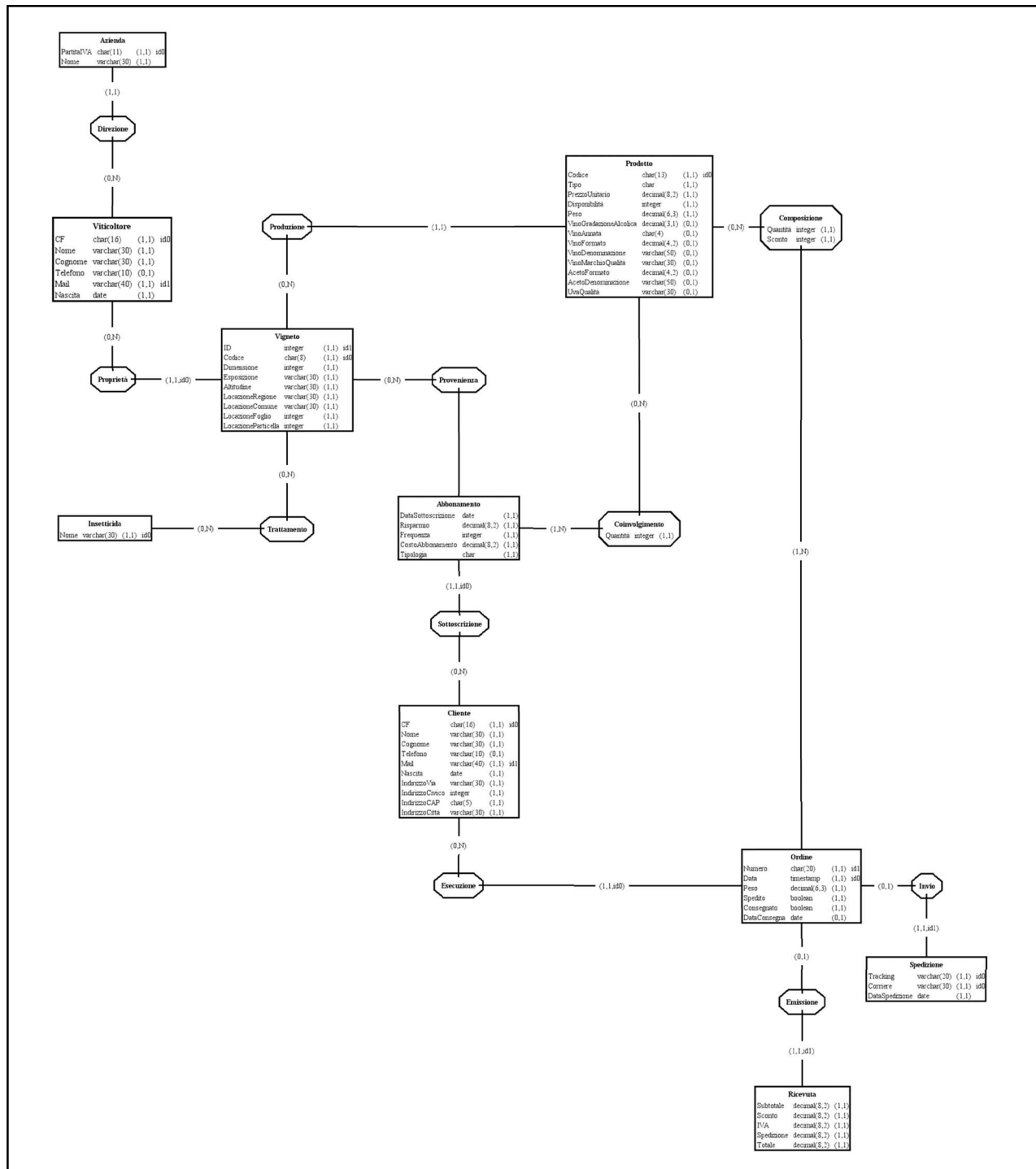
Per l'entità *Ordine* si sceglie di adottare come identificatore principale il Numero dell'ordine perché ha una dimensione fissa ed è più semplice.

Per l'entità *Spedizione* si preferisce mantenere l'identificatore esterno perché composto da un numero minore di attributi.

Per le restanti entità si preferisce utilizzare gli identificatori indicati in fase di progettazione concettuale.

4.6. Schema ristrutturato finale

Inserire lo schema ristrutturato finale



4.7. Schema logico

Abbonamento(Cliente:Cliente.CF, Vigneto:Vigneto.ID, DataSottoscrizione, Risparmio, Frequenza, CostoAbbonamento, Tipologia)

Cliente(CF, Nome, Cognome, Telefono*, Mail, Nascita, IndirizzoVia, IndirizzoCivico, IndirizzoCAP, IndirizzoCittà)

Insetticida(Nome)

Viticoltore(CF, Nome, Cognome, Telefono*, Mail, Nascita)

Azienda(PartitaIVA, Nome, Direttore)

Ordine(Numero, Peso, Spedito, Consegnato, DataConsegna*, Data, Cliente:Cliente.CF)

Spedizione(Ordine:Ordine.Numero, Tracking, Corriere, DataSpedizione)

Ricevuta(Ordine:Ordine.Numero, Subtotale, Sconto, IVA, Spedizione, Totale)

Vigneto(ID, Codice, Proprietario:Viticoltore.CF, Dimensione, Esposizione, Altitudine, LocazioneRegione, LocazioneComune, LocazioneFoglio, LocazioneParticella)

Trattamento(Insetticida:Insetticida.Nome, Vigneto:Vigneto.ID)

Coinvolgimento(Cliente:Abbonamento.Cliente, Vigneto:Abbonamento.Vigneto, Prodotto:Prodotto.Codice, Quantità)

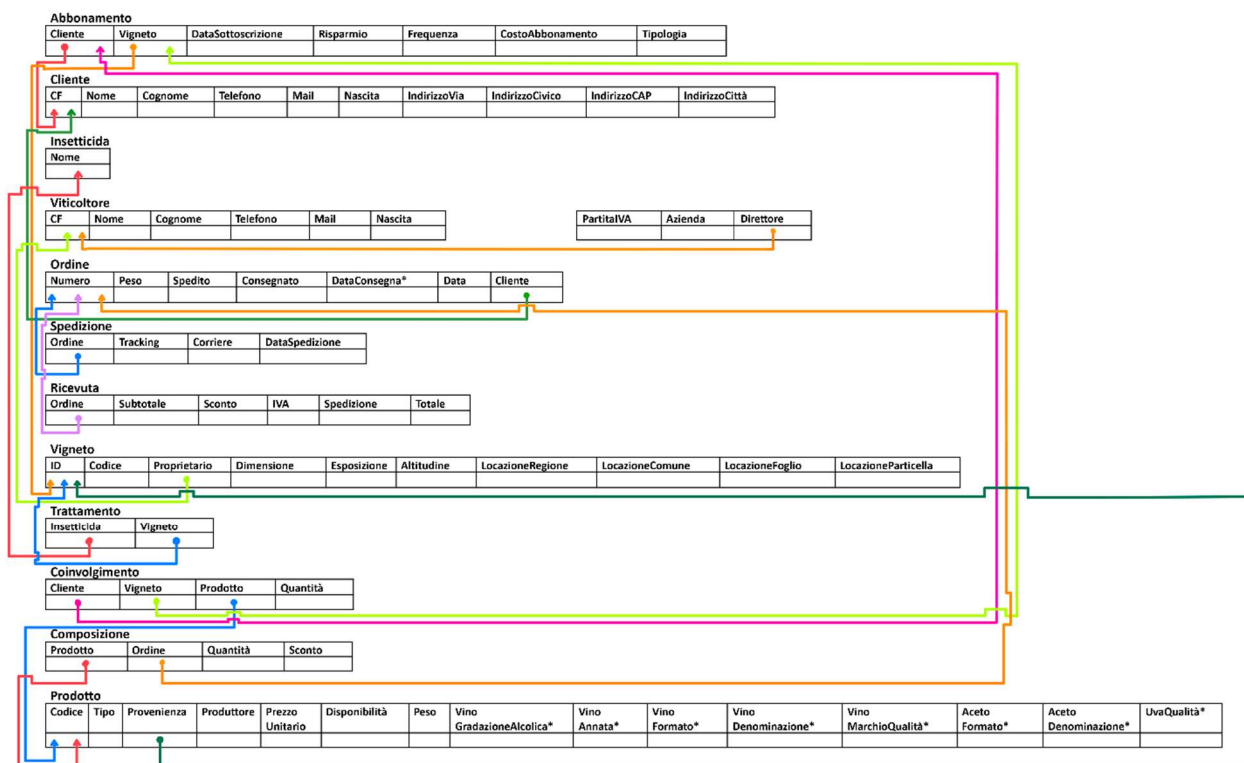
Composizione(Prodotto:Prodotto.Codice, Ordine:Ordine.Numero , Quantità, Sconto)

Prodotto(Codice, Tipo, Provenienza:Vigneto.ID, PrezzoUnitario, Disponibilità, Peso, VinoGradazioneAlcolica*, VinoAnnata*, VinoFormato*, VinoDenominazione*, VinoMarchioQualità*, AcetoFormato*, AcetoDenominazione*, UvaQualità*)

4.8. Documentazione dello schema logico

Vincoli introdotti in fase di ristrutturazione:

- L'attributo Consegnato può essere true solo se l'attributo Spedito è true.
- L'attributo DataConsegna è obbligatorio solo se Consegnato è true.
- L'attributo Tipo di Prodotto è 'U' se il prodotto è una istanza di Uva, 'V' se il prodotto è una istanza di Vino, 'A' se il prodotto è una istanza di Aceto.
- Gli attributi VinoGradazioneAlcolica, VinoAnnata, VinoFormato, VinoDenominazione, VinoMarchioQualità sono obbligatori solo se il prodotto è una istanza di Vino.
- Gli attributi AcetoFormato, AcetoDenominazione sono obbligatori solo se prodotto è una istanza di Aceto.
- L'attributo UvaQualità è obbligatorio solo se prodotto è una istanza di Uva.



5. Normalizzazione

Workpackage	Task	Responsabile
WP3	Normalizzazione	Dell'Orto Giuseppe Maria

Le varie entità analizzate rispettano tutte la forma normale di Boyce and Codd (BCNF) e di conseguenza sono anche in seconda e in prima forma normale.

Abbonamento(Cliente:Cliente.CF, Vigneto:Vigneto.ID, DataSottoscrizione, Risparmio, Frequenza, CostoAbbonamento, Tipologia)

Cliente, Vigneto -> DataSottoscrizione, Risparmio, Frequenza, CostoAbbonamento, Tipologia

Cliente(CF, Nome, Cognome, Telefono*, Mail, Nascita, IndirizzoVia, IndirizzoCivico, IndirizzoCAP, IndirizzoCittà)

CF->Nome, Cognome, Telefono, Mail, Nascita, IndirizzoVia, IndirizzoCivico, IndirizzoCAP, IndirizzoCittà*

Si osservi che IndirizzoCAP non determina IndirizzoCittà in quanto alcuni CAP sono associati a città diverse.

Azienda(PartitaIVA, Nome)

PartitaIva -> Nome

Insetticida(Nome)

Nome -> Nome (Banale)

Non essendoci dipendenze funzionali si considera in BCNF.

Ordine(Numero, Peso, Spedito, Consegnato, DataConsegna*, Data, Cliente:Cliente.CF)

Numero -> Peso, Spedito, Consegnato, DataConsegna, Data, Cliente*

Spedizione(Ordine:Ordine.Numero, Tracking, Corriere, DataSpedizione)

Ordine-> Tracking, Corriere, DataSpedizione

Chiave alternativa:

Tracking, Corriere -> Ordine, DataSpedizione

Ricevuta(Ordine:Ordine.Numero, Subtotale, Sconto, IVA, Spedizione, Totale)

Ordine -> Subtotale, Sconto, IVA, Spedizione, Totale

Vigneto(ID, Codice, Proprietario:Viticoltore.CF, Dimensione, Esposizione, Altitudine, LocazioneRegione, LocazioneComune, LocazioneFoglio, LocazioneParticella)

ID -> Codice, Proprietario, Dimensione, Esposizione, Altitudine, LocazioneRegione, LocazioneComune, LocazioneFoglio, LocazioneParticella

Chiave alternativa:

Codice, Proprietario->ID, Dimensione, Esposizione, Altitudine, LocazioneRegione, LocazioneComune, LocazioneFoglio, LocazioneParticella

Trattamento(Insetticida:Insetticida.Nome, Vigneto:Vigneto.ID)

Insetticida, Vigneto -> Insetticida, Vigneto (Banale)

Non essendoci dipendenze funzionali non banali si considera in BCNF.

Coinvolgimento(Cliente, Vigneto, Prodotto, Quantità)

Cliente,Vigneto,Prodotto -> Quantità

Composizione(Prodotto, Ordine, Quantità, Sconto)

Prodotto,Ordine -> Quantità,Sconto

Prodotto(Codice, Tipo, Provenienza:Vigneto.ID, PrezzoUnitario, Disponibilità, Peso, VinoGradazioneAlcolica*, VinoAnnata*, VinoFormato*, VinoDenominazione*, VinoMarchioQualità, AcetoFormato, AcetoDenominazione*, UvaQualità*)

Codice -> Tipo, Provenienza, PrezzoUnitario, Disponibilità, Peso, VinoGradazioneAlcolica, VinoAnnata*, VinoFormato*, VinoDenominazione*, VinoMarchioQualità, AcetoFormato, AcetoDenominazione*, UvaQualità**

Script Creazione e Popolamento Database

Workpackage	Task	Responsabile
WP2	SQL: Script creazione e popolamento	Battipaglia Valerio

```
--Creazione database
/*DROP DATABASE IF EXISTS "Gruppo01";
CREATE DATABASE "Gruppo01" WITH OWNER = postgres;*/
--Creazione tabelle

begin transaction;

drop table if exists abbonamento cascade;
drop table if exists cliente cascade;
drop table if exists insetticida cascade;
drop table if exists viticoltore cascade;
drop table if exists ordine cascade;
drop table if exists spedizione cascade;
drop table if exists ricevuta cascade;
drop table if exists vigneto cascade;
drop table if exists trattamento cascade;
drop table if exists coinvolgimento cascade;
drop table if exists composizione cascade;
drop table if exists prodotto cascade;
drop table if exists azienda cascade;
drop domain if exists tipo_prodotto cascade;

create table cliente (
    cf char(16) primary key,
    nome varchar(30) not null,
    cognome varchar(30) not null,
    telefono varchar(10),
    mail varchar(40) unique not null,
    nascita date not null,
    indirizzo_via varchar(30) not null,
    indirizzo_civico integer not null,
    indirizzo_cap char(5) not null,
    indirizzo_città varchar(30) not null,
    check(age(nascita) > '18 years')
);

create table insetticida (
    nome varchar(30) primary key
);

create table viticoltore (
    cf char(16) primary key,
```

```

    nome varchar(30) not null,
    cognome varchar(30) not null,
    telefono varchar(10),
    mail varchar(40) unique not null,
    nascita date not null
);

create table azienda (
    partita_iva char(11) primary key,
    nome varchar(30) not null,
    direttore char(16) not null,
    constraint fk_direzione foreign key(direttore) references viticoltore(cf)
    on update cascade on delete cascade
);

create table ordine (
    numero char(20) primary key,
    data timestamp default current_timestamp,
    peso decimal(6,3) not null,
    spedito boolean not null,
    consegnato boolean not null,
    data_consegna date null,
    cliente char(16) not null,
    constraint fk_esecuzione foreign key (cliente) references cliente(cf)
    on update cascade on delete restrict,
    unique(data, cliente),
    check((data_consegna is not null and consegnato = 'true') or (consegnato =
'false' and data_consegna is null)),
    check((consegnato = 'true' and spedito = 'true') or consegnato = 'false')
);

create table spedizione (
    ordine char(20) primary key,
    tracking varchar(20) not null,
    corriere varchar(30) not null,
    data_spedizione date not null,
    constraint fk_invio foreign key(ordine) references ordine(numero)
    on update cascade on delete restrict
);

create table ricevuta (
    ordine char(20) primary key,
    subtotale decimal(8,2) not null,
    sconto decimal(8,2) not null,
    iva decimal(8,2) not null,

```

```

    spedizione decimal(8,2) not null,
    totale decimal(8,2) not null,
    constraint fk_emissione foreign key(ordine) references ordine(numero)
    on update cascade on delete cascade
    deferrable initially deferred
);

create table vigneto (
    id serial primary key,
    codice char(8) not null,
    dimensione integer not null,
    esposizione varchar(30) not null,
    altitudine varchar(30) not null,
    locazione_regione varchar(30) not null,
    locazione_comune varchar(30) not null,
    locazione_foglio integer not null,
    locazione_particella integer not null,
    proprietario char(16) not null,
    constraint fk_proprieta foreign key (proprietario) references viticoltore(cf)
    on update cascade on delete restrict,
    unique(proprietario, codice)
);

create domain tipo_prodotto as char check
    (value = 'U' or value = 'V' or value = 'A');

create table prodotto (
    codice char(13) primary key,
    provenienza integer not null,
    tipo tipo_prodotto not null,
    prezzo_unitario decimal(8,2),
    disponibilità integer not null,
    peso decimal(6,3) not null,
    vino_gradazione_alcolica decimal(3,1),
    vino_annata char(4),
    vino_formato decimal(4,2),
    vino_denominazione varchar(50),
    vino_marchio_qualità varchar(30),
    aceto_formato decimal(4,2),
    aceto_denominazione varchar(50),
    uva_qualità varchar(30),
    constraint fk_provenienza foreign key(provenienza) references vigneto(id)
    on update cascade on delete restrict,
    check((tipo = 'U' and uva_qualità is not null
        and vino_gradazione_alcolica is null and vino_annata is null and
vino_formato is null
        and vino_denominazione is null and vino_marchio_qualità is null
        and aceto_formato is null and aceto_denominazione is null) or

```

```

        (tipo = 'V' and vino_gradazione_alcolica is not null and vino_annata is not
null and vino_formato is not null
            and vino_denominazione is not null and vino_marchio_qualità is not null
            and aceto_formato is null and aceto_denominazione is null and uva_qualità
is null) or
        (tipo = 'A' and uva_qualità is null
            and vino_gradazione_alcolica is null and vino_annata is null and
vino_formato is null
            and vino_denominazione is null and vino_marchio_qualità is null
            and aceto_formato is not null and aceto_denominazione is not null))
);

create table trattamento (
    insetticida varchar(30) references insetticida(nome)
    on update cascade on delete restrict,
    vigneto integer references vigneto(id)
    on update cascade on delete restrict,
    primary key(insetticida, vigneto)
);

create table composizione (
    prodotto char(13),
    ordine char(20),
    quantità integer not null,
    sconto integer not null,
    constraint fk_composizione_prodotto foreign key (prodotto) references
prodotto(codice)
    on update cascade on delete restrict
    deferrable initially deferred,
    constraint fk_composizione_ordine foreign key(ordine) references ordine(numero)
    on update cascade on delete restrict
    deferrable initially deferred,
    primary key(prodotto, ordine)
);

create table abbonamento (
    cliente char(16),
    vigneto integer,
    data_sottoscrizione date not null,
    risparmio decimal(8,2) default 0.00,
    frequenza integer not null,
    costo_abbonamento decimal(8,2) not null,
    tipologia char not null,
    primary key(cliente, vigneto),
    constraint fk_sottoscrizione foreign key (cliente) references cliente(cf)
    on update cascade on delete cascade,

```



```

        constraint fk_provenienza foreign key (vigneto) references vigneto(id)
        on update cascade on delete cascade,
        check((tipologia = 'A' and (frequenza = 1 or frequenza = 2 or frequenza = 3 or
frequenza = 4 or frequenza = 6)) or
            tipologia = 'S' and (frequenza = 1 or frequenza = 2 or frequenza = 3))
    );

create table coinvolgimento (
    cliente char(16),
    vigneto integer,
    prodotto char(13),
    quantità integer not null,
    primary key(cliente, vigneto, prodotto),
    constraint fk_coinvolgimento_abbonamento foreign key(cliente, vigneto)
references abbonamento(cliente, vigneto)
    on update cascade on delete restrict
    deferrable initially deferred,
    constraint fk_coinvolgimento_prodotto foreign key(prodotto) references
prodotto(codice)
    on update cascade on delete restrict
    deferrable initially deferred
);

--Popolamento

delete from ricevuta cascade;
delete from coinvolgimento cascade;
delete from abbonamento cascade;
delete from spedizione cascade;
delete from cliente cascade;
delete from composizione cascade;
delete from prodotto cascade;
delete from ordine cascade;
delete from insetticida cascade;
delete from vigneto cascade;
delete from viticoltore cascade;
delete from trattamento cascade;
delete from azienda cascade;

ALTER SEQUENCE vigneto_id_seq RESTART WITH 1;

insert into viticoltore(cf, nome, cognome, mail, nascita)
values('PRLSGS36P49B8130', 'Mario', 'Rui', 'mariorui@gmail.com', '1966-05-11'),
      ('VVRLVA87B13H916I', 'Maria', 'Della Vecchia', 'mariadv@gmail.com', '1956-05-
11'),
      ('RGPPZJ27M68G9770', 'Giorgio', 'Chiellini', 'gichi@gmail.com', '1994-05-
11'),

```

```

        ('SFPLHG62M28A625K', 'Luca', 'Rossi', 'lucarossi@gmail.com', '1994-07-11'),
        ('PBMLCL53P05B989C', 'Matteo', 'Bianchi', 'matteobianchi@gmail.com', '1994-
06-11'),
        ('DPHNTV37E71H347W', 'Ernesto', 'Iannaccone', 'ernestoianaccone@gmail.com',
'1956-05-11'),
        ('KLMPVI79E02A427C', 'Pasquale', 'De Crescenzo', 'pdscrescenzo@gmail.com',
'1978-05-11'),
        ('ZTVTVM63B21C452E', 'Amilcare', 'Esposito', 'amilcareesp@gmail.com', '1932-
09-13'),
        ('PYTRFV92H47C348D', 'Ciro', 'Esposito', 'ciroesp@gmail.com', '1978-05-23'),
        ('RVHPSY44B15M147B', 'Gennaro', 'Savastano', 'gennycar@gmail.com', '1988-02-
16');

insert into azienda(partita_iva, nome, direttore)
values('75685900823', 'Torre dei Chiusi', 'PRLSGS36P49B8130'),
      ('52224190570', 'Torre Venere', 'VRLVA87B13H916I'),
      ('14348440885', 'Borgo Giulia', 'RGPPZJ27M68G9770'),
      ('10188390149', 'Alepa', 'SFPLHG62M28A625K'),
      ('48766970122', 'Cantine Ruggiero', 'PBMLCL53P05B989C'),
      ('59909760353', 'Bosco de Medici', 'DPHNTV37E71H347W'),
      ('56275190157', 'Cantina del Angelo', 'KLMPVI79E02A427C'),
      ('66788950278', 'Cantine del mare', 'ZTVTVM63B21C452E'),
      ('67971160154', 'Cantina Sociale Solopaca', 'PYTRFV92H47C348D'),
      ('79606460594', 'San Giovanni', 'RVHPSY44B15M147B');

insert into cliente(cf, nome, cognome, telefono, mail, nascita, indirizzo_via,
indirizzo_civico, indirizzo_cap, indirizzo_città)
values('SPLCQZ32S62I400K', 'Andrea', 'Verdi', '3384778956', 'averdi@gmail.com',
'1967-04-11', 'Roma', 420, 69000, 'Genova');
insert into cliente(cf, nome, cognome, mail, nascita, indirizzo_via,
indirizzo_civico, indirizzo_cap, indirizzo_città)
values('QRRSVC71D59F865Z', 'Luca', 'Massi', 'lmassi@gmail.com', '1955-05-11',
'Garibali', 124, 86000, 'Trieste');
insert into cliente(cf, nome, cognome, telefono, mail, nascita, indirizzo_via,
indirizzo_civico, indirizzo_cap, indirizzo_città)
values('ZZHWP90B08G358A', 'Pino', 'Insegno', '3456778996', 'pinse@gmail.com',
'1945-08-11', 'Isonzo', 167, 87540, 'Gorizia');
insert into cliente(cf, nome, cognome, telefono, mail, nascita, indirizzo_via,
indirizzo_civico, indirizzo_cap, indirizzo_città)
values('MTMZBN40H24C595L', 'Giuseppe', 'Brambilla', '3453445657',
'gbramb@gmail.com', '1967-05-12', 'Trieste', 189, 83500, 'Pordenone');
insert into cliente(cf, nome, cognome, mail, nascita, indirizzo_via,
indirizzo_civico, indirizzo_cap, indirizzo_città)
values('XZXNG84C24L864S', 'Pasquale', 'Borsellin', 'borsep@gmail.com', '1967-09-
24', 'Monte Grappa', 245, 88700, 'Pordenone');
insert into cliente(cf, nome, cognome, mail, nascita, indirizzo_via,
indirizzo_civico, indirizzo_cap, indirizzo_città)
values('YLMNGD73B61F533I', 'Fausto', 'Bisognin', 'fbisogn@gmail.com', '1977-05-30',
'Brennero', 42, 86340, 'Udine');

```

```

insert into cliente(cf, nome, cognome, telefono, mail, nascita, indirizzo_via,
indirizzo_civico, indirizzo_cap, indirizzo_città)
values('CLGWTV39E17G914M', 'Lucia', 'Marzullo', '3308989555', 'lmarz@gmail.com',
'1988-07-26', 'Vittorio Veneto', 78, 82999, 'Udine');
insert into cliente(cf, nome, cognome, mail, nascita, indirizzo_via,
indirizzo_civico, indirizzo_cap, indirizzo_città)
values('KHRGDZ86R53A200C', 'Rosa', 'Mauriello', 'mmaur@gmail.com', '1999-03-11',
'Caduti In Guerra', 69, 89780, 'Venezia');
insert into cliente(cf, nome, cognome, mail, nascita, indirizzo_via,
indirizzo_civico, indirizzo_cap, indirizzo_città)
values('GTGGFS76D12I663C', 'Paola', 'Iannicelli', 'pian@gmail.com', '1962-01-19',
'Eroi di Patria', 56, 81666, 'Verona');
insert into cliente(cf, nome, cognome, mail, nascita, indirizzo_via,
indirizzo_civico, indirizzo_cap, indirizzo_città)
values('GTSTJP48C11H862F', 'Carmine', 'Petraglia', 'cpetr@gmail.com', '1998-01-11',
'Patria', 22, 81420, 'Varese');

insert into insetticida(nome)
values('Verderame'),
('Naturalis'),
('Prodatix');

insert into vigneto(codice, dimensione, esposizione, altitudine, locazione_regione,
locazione_comune, locazione_foglio, locazione_particella, proprietario)
values('PRLSGS1', 1000, 'nord', 500, 'Liguria', 'La Spezia', '20', '100',
'PRLSGS36P49B8130'),
('PRLSGS2', 2000, 'sud', 1300, 'Lombardia', 'Varese', '10', '56',
'PRLSGS36P49B8130'),
('VURLVA1', 1400, 'sud', 800, 'Emilia-Romagna', 'Forli', '24', '120',
'VURLVA87B13H916I'),
('VURLVA2', 2400, 'est', 1500, 'Piemonte', 'Novara', '36', '145',
'VURLVA87B13H916I'),
('RGPPZJ1', 3000, 'est', 700, 'Sardegna', 'Nuoro', '43', '176',
'RGPPZJ27M68G9770'),
('RGPPZJ2', 3200, 'west', 300, 'Emilia-Romagna', 'Ravenna', '12', '65',
'RGPPZJ27M68G9770'),
('SFPLHG1', 2200, 'est', 300, 'Toscana', 'Grosseto', '3', '12',
'SFPLHG62M28A625K'),
('PBMLCL1', 1700, 'nord', 355, 'Piemonte', 'Vercelli', '45', '187',
'PBMLCL53P05B989C'),
('DPHNTV1', 3400, 'nord', 700, 'Lombardia', 'Brescia', '34', '134',
'DPHNTV37E71H347W'),
('KLMPVI1', 5000, 'sud', 450, 'Liguria', 'Genova', '37', '162',
'KLMPVI79E02A427C'),
('ZTVTVM1', 1500, 'west', 700, 'Molise', 'Isernia', '29', '119',
'ZTVTVM63B21C452E'),
('ZTVTVM2', 1500, 'sud', 100, 'Sicilia', 'Palermo', '18', '75',
'ZTVTVM63B21C452E'),

```

```

        ('PYTRFV1', 5000, 'nord', 600, 'Lombardia', 'Lodi', '26', '104',
'PYTRFV92H47C348D'),
        ('PYTRFV2', 2700, 'sud', 95, 'Toscana', 'Arezzo', '42', '169',
'PYTRFV92H47C348D'),
        ('RVHPSY1', 3000, 'est', 740, 'Lazio', 'Rieti', '7', '30',
'RVHPSY44B15M147B'),
        ('RVHPSY2', 3800, 'west', 467, 'Marche', 'Macerata', '43', '167',
'RVHPSY44B15M147B');

insert into trattamento(insetticida, vigneto)
values('Verderame', 1),
      ('Verderame', 2),
      ('Verderame', 3),
      ('Verderame', 4),
      ('Verderame', 5),
      ('Naturalis', 6),
      ('Naturalis', 7),
      ('Naturalis', 8),
      ('Naturalis', 9),
      ('Naturalis', 10),
      ('Prodatix', 11),
      ('Prodatix', 12),
      ('Prodatix', 13),
      ('Prodatix', 14);

/*VINO*/
insert into prodotto(codice,provenienza,
tipo,prezzo_unitario,disponibilità,peso,vino_gradazione_alcolica, vino_annata,
vino_formato, vino_marchio_qualità, vino_denominazione)
values('00000000aaaa',4 , 'V', 13.00,100,1, 12, 2021 ,0.75, 'igp', 'Aglianico'),
      ('00000000aaab',2 , 'V', 12.00,150,1.250, 14, 2019 ,1.00, 'dop',
'Solopaca'),
      ('00000000aaac',3 , 'V', 12.50,100,1, 12.5, 2018 ,0.75, 'igp', 'Fiano'),
      ('00000000aaad',3 , 'V', 18.00,200,1.750, 13.6, 2021 ,1.50, 'dop',
'Montepulciano'),
      ('00000000aaan',4 , 'V', 13.50,100,1, 12.5, 2019 ,0.75, 'igp', 'Fiano'),
      ('00000000aaao',4, 'V', 20.00,100,5.50, 16, 2021 ,5, 'stg', 'Primitivo di
Manudria'),
      ('00000000aaas',5, 'V', 11.50,200,1, 12.5, 2018 ,0.75, 'dop', 'Lambrusco'),
      ('00000000aaat',5 , 'V', 20.00,100,1.50, 12, 2021 ,1, 'stg', 'Chianti');

/*Aceti*/
insert into prodotto(codice,provenienza,tip0,prezzo_unitario,
disponibilità,peso,aceto_formato, aceto_denominazione)
values('00000000aaae', 7 , 'A' ,5.00,100,1.00,0.75, 'Di Vino'),
      ('00000000aaaf', 7 , 'A' ,6.00,200,1.00,0.75, 'Di Vino Bianco'),
      ('00000000aaag', 3 , 'A' ,4.00,150,1.00,0.75, 'Di Vino Rosso'),
      ('00000000aaap', 4 , 'A' ,4.70,300,1.00,0.75, 'Di Vino Decolorato');

```

```

/*UVA*/
insert into prodotto(codice,provenienza,tipo,prezzo_unitario,
disponibilità,peso,uva_qualità)
values('000000000aaah',2, 'U', 3.30, 20,5.50 , 'Scarlotta Seedless'),
      ('000000000aaai',2, 'U', 3.50, 50,6.50 , 'Red Globe'),
      ('000000000aaal',3, 'U', 2.50, 50,5.50 , 'Zibibbo'),
      ('000000000aaam',4, 'U', 1.50, 50,5.50 , 'Fragola'),
      ('000000000aaaq',4 , 'U', 2.80, 50,5.00 , 'Regina'),
      ('000000000aaar',4 , 'U', 1.90, 50,5.00 , 'Italia');

commit;

--abbonamento di 'GTSTJP48C11H862F'
begin transaction;
  insert into coinvolgimento(cliente, vigneto, prodotto, quantità)
  values('GTSTJP48C11H862F', 1, '000000000aaaa', 1),
        ('GTSTJP48C11H862F', 1, '000000000aaab', 3),
        ('GTSTJP48C11H862F', 1, '000000000aaac', 1);

  insert into abbonamento(cliente, vigneto, tipologia, data_sottoscrizione,
frequenza, costo_abbonamento)
  values('GTSTJP48C11H862F', 1, 'A', '2021-07-14', 2, 12.0);
commit;

--abbonamento di 'GTGGFS76D12I663C'
begin transaction;
  insert into coinvolgimento(cliente, vigneto, prodotto, quantità)
  values('GTGGFS76D12I663C', 2, '000000000aaab', 2),
        ('GTGGFS76D12I663C', 2, '000000000aaad', 1);

  insert into abbonamento(cliente, vigneto, tipologia, data_sottoscrizione,
frequenza, costo_abbonamento)
  values('GTGGFS76D12I663C', 2, 'S', '2021-06-15', 2, 15.0);
commit;

--abbonamento di 'KHRGDZ86R53A200C'
begin transaction;
  insert into coinvolgimento(cliente, vigneto, prodotto, quantità)
  values('KHRGDZ86R53A200C', 3, '000000000aaaa', 2);

  insert into abbonamento(cliente, vigneto, tipologia, data_sottoscrizione,
frequenza, costo_abbonamento)
  values('KHRGDZ86R53A200C', 3, 'A', '2021-05-16', 3, 16.0);
commit;

```

```

--abbonamento di 'CLGWT39E17G914M'
begin transaction;
    insert into coinvolgimento(cliente, vigneto, prodotto, quantità)
    values('CLGWT39E17G914M', 4, '00000000aaad', 5);

    insert into abbonamento(cliente, vigneto, tipologia, data_sottoscrizione,
frequenza, costo_abbonamento)
    values('CLGWT39E17G914M', 4, 'S', '2021-08-17', 1, 12.0);
commit;

--abbonamento di 'YLMNGD73B61F533I'
begin transaction;
    insert into coinvolgimento(cliente, vigneto, prodotto, quantità)
    values('YLMNGD73B61F533I', 5, '00000000aaaf', 1);

    insert into abbonamento(cliente, vigneto, tipologia, data_sottoscrizione,
frequenza, costo_abbonamento)
    values('YLMNGD73B61F533I', 5, 'A', '2021-04-18', 1, 100.0);
commit;

--ORDINI

--ordine di 'SPLCQZ32S62I400K'
begin transaction;
    --Primo ordine composto da due vini diversi: primo vino è ...a, 5 bottiglie;
secondo vino è ...b, 5 bottiglie.
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('00000000aaaa', '8ZIXCQNTXMIUIBETXAJY', 5, 10),
        ('00000000aaab', '8ZIXCQNTXMIUIBETXAJY', 5, 10),
        ('00000000aaac', '8ZIXCQNTXMIUIBETXAJY', 5, 10),
        ('00000000aaad', '8ZIXCQNTXMIUIBETXAJY', 5, 10);

    --Problema nella consegna
    insert into ordine(numero, peso, spedito, consegnato, data_consegna, cliente,
data)
    values('8ZIXCQNTXMIUIBETXAJY', 2.6, 'true', 'false', null, 'SPLCQZ32S62I400K',
'2022-01-01');

    --spedizione
    insert into spedizione(ordine, tracking, corriere, data_spedizione)
    values('8ZIXCQNTXMIUIBETXAJY', 'AAV0XJOI67CJGSUV0E', 'UPS', '2021-09-15');
commit;

```

```

--ordine di 'SPLCQZ32S62I400K'
begin transaction;
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('00000000aaaa', '8ZIXCQNTXMIUIBETXAJJ', 5, 10);
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('00000000aaab', '8ZIXCQNTXMIUIBETXAJJ', 5, 10);

    --Problema nella consegna
    insert into ordine(numero, peso, spedito, consegnato, data_consegna, cliente,
data)
    values('8ZIXCQNTXMIUIBETXAJJ', 2.6, 'false', 'false', null, 'SPLCQZ32S62I400K',
'2022-01-02');
commit;

--ordine di QRRSVC71D59F865Z
begin transaction;
    --Secondo ordine composto da un solo vino, 3 bottiglie e nessuno sconto.
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('00000000aaab', 'VE8URM7NEMTE6VZF6CQZ', 3, 0);
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('00000000aaac', 'VE8URM7NEMTE6VZF6CQZ', 3, 0);
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('00000000aaad', 'VE8URM7NEMTE6VZF6CQZ', 3, 0);
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('00000000aaae', 'VE8URM7NEMTE6VZF6CQZ', 3, 0);

    insert into ordine(numero, peso, spedito, consegnato, data_consegna, cliente,
data)
    values('VE8URM7NEMTE6VZF6CQZ', 15.4, 'true', 'true', '2021-09-19',
'QRRSVC71D59F865Z', '2022-01-03'); --Consegnato con successo

    insert into spedizione(ordine, tracking, corriere, data_spedizione)
    values('VE8URM7NEMTE6VZF6CQZ', 'MD3DACGXXFES7CE2JGRG', 'BRT', '2021-09-16');
commit;

--ordine di ZZHWPP90B08G358A
begin transaction;
    --Terzo ordine composto da un solo vino, 2 bottiglie e sconto totale ( parente
del gestore del sito ).
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('00000000aaab', 'BST7501007DQ8334YMQ6', 2, 80),
        ('00000000aaai', 'BST7501007DQ8334YMQ6', 2, 10),
        ('00000000aaag', 'BST7501007DQ8334YMQ6', 2, 10);

    insert into ordine(numero, peso, spedito, consegnato, data_consegna, cliente)

```

```

    values('BST7501007DQ8334YMQ6', 14, 'true', 'false', null, 'ZZHWPP90B08G358A'); -
--Non ancora consegnato

    insert into spedizione(ordine, tracking, corriere, data_spedizione)
    values('BST7501007DQ8334YMQ6', 'ZNEIZDR1XBP1', 'SDA', '2021-09-17');
commit;

--ordine di MTMZBN40H24C595L
begin transaction;
    --Quarto ordine composto da due aceti diversi: primo aceto è ...e, 2 bottiglie;
    secondo aceto è ...f, 15 bottiglie.
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('000000000aaae', 'CDV20CARAJ1IRK3142F2', 2, 4);
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('000000000aaai', 'CDV20CARAJ1IRK3142F2', 15, 7.67);

    insert into ordine(numero, peso, spedito, consegnato, data_consegna, cliente,
data)
    values('CDV20CARAJ1IRK3142F2', 23.20, 'false', 'false', null,
'MTMZBN40H24C595L', '2022-01-04'); --Non ancora consegnato
commit;

--ordine di MTMZBN40H24C595L
begin transaction;
    --Quinto ordine composto da un solo acetp, 8 bottiglie e sconto 50%.
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('000000000aaae', 'IJJE0HVD2RYNS36AJ4SD', 8, 50);

    insert into ordine(numero, peso, spedito, consegnato, data_consegna, cliente,
data)
    values('IJJE0HVD2RYNS36AJ4SD', 3, 'true', 'false', null, 'MTMZBN40H24C595L',
'2022-01-05'); --Non ancora consegnato

    insert into spedizione(ordine, tracking, corriere, data_spedizione)
    values('IJJE0HVD2RYNS36AJ4SD', '7581XQAUI8F1W1I1RS', 'UPS', '2021-09-19');
commit;

--ordine di MTMZBN40H24C595L
begin transaction;
    insert into composizione(prodotto, ordine, quantità, sconto)
    values('000000000aaaa', 'IJJE0HVD2RYNS36AK4SD', 8, 50);

    insert into ordine(numero, peso, spedito, consegnato, data_consegna, cliente,
data)
    values('IJJE0HVD2RYNS36AK4SD', 3, 'false', 'false', null, 'MTMZBN40H24C595L',
'2022-01-06'); --Non ancora consegnato

```



```
commit;

/*begin transaction;
  --Ordine senza prodotti fa fallire la transazione
  insert into ordine(numero, peso, spedito, consegnato, data_consegna, cliente,
data)
    values('IJJE0HVD2RYNS36A4SD', 2.6, 'false', 'false', null, 'SPLCQZ32S62I400K',
'2010-01-10');
commit;*/

/*
begin transaction;
  -- Abbonamento privo di prodotti fa fallire la transazione
  insert into abbonamento(cliente, vigneto, tipologia, data_sottoscrizione,
frequenza, costo_abbonamento)
    values('YLMNGD73B61F533I', 1, 'A', '2021-04-18', 1, 104.0);
commit;
*/
```

6. Query SQL

<i>Workpackage</i>	<i>Task</i>	<i>Responsabile</i>
WP3	SQL: Query	Dell'Orto Giuseppe Maria

6.1. Query con operatore di aggregazione e join: Abbonamenti per viticoltore

Per ottenere il numero totale di abbonamenti attivi per viticoltore si è dovuta eseguire una join fra abbonamento, vigneto, viticoltore per ottenere sia la corrispondenza fra abbonamento e vigneto che fra vigneto e viticoltore.

Si è poi selezionato le informazioni del viticoltore accostando ad esse le informazioni sul numero degli abbonamenti ottenuta utilizzando l'operatore aggregato *count*.

```
-- ABBONAMENTI PER VITICOLTORE
select
    viticoltore.cf as "Viticoltore",
    viticoltore.nome as "Nome Viticoltore",
    viticoltore.cognome as "Cognome Viticoltore",
    count(*) as "Abbonamenti Attivi"
from abbonamento, vigneto, viticoltore
where
    abbonamento.vigneto = vigneto.id and
    vigneto.proprietario = viticoltore.cf
group by
    viticoltore.cf, viticoltore.nome, viticoltore.cognome;
```

6.2. Query nidificata complessa: Articoli correlati

La query permette di suggerire al cliente, che acquista un certo prodotto, gli articoli acquistati da altri clienti che hanno comprato a loro volta il medesimo articolo.

Si definisce prima la vista *'ordini_per_cliente'* che permette di ottenere tutti i prodotti acquistati e il relativo cliente. Nella query interna si va a selezionare tutte le righe che hanno codice fiscale diverso dal cliente a cui vogliamo consigliare i prodotti e che allo stesso tempo hanno acquistato il prodotto in base al quale se ne consigliano altri.

Dalla query interna si selezionano i clienti che hanno acquistato il nostro stesso prodotto e la tabella ottenuta si mette in join con la tabella *ordine_per_cliente* per ottenere gli altri articoli comprati da altre persone che hanno acquistato lo stesso articolo di interesse; ci si assicura di non rilesionare il prodotto di riferimento e allo stesso tempo di controllare la consistenza dei dati tramite uguaglianza su codice fiscale.

```
--- ARTICOLI CORRELATI
create or replace view ordini_per_cliente(cf, codice) as
  select cliente.cf, prodotto.codice
  from
    cliente, ordine, composizione, prodotto
  where
    ordine.cliente = cliente.cf and
    ordine.numero = composizione.ordine and
    composizione.prodotto = prodotto.codice;

select distinct J.codice as "Correlati"
from ordini_per_cliente J, (select distinct *
  from ordini_per_cliente J1
  where
    J1.cf <> 'MTMZBN40H24C595L' and
    exists (select *
      from ordini_per_cliente J2
      where J2.codice = J1.codice and
            J2.cf <> J1.cf and
            J2.codice = '00000000aaaa')) as correlati
where J.codice <> correlati.codice and J.cf = correlati.cf;
```

6.3. Query insiemistica: Numero vendite per prodotto

Si è diviso il problema in tre insiemi: uva, vino, aceto. Per ogni insieme si è fatta una join fra le tabelle ordine, composizione e prodotto e si è controllato il tipo.

Una volta ottenuta questa tabella si è raggruppato per codice e si è contato il numero di occorrenze del prodotto per avere quindi il numero di vendite. Infine, si sono uniti i vari insiemi per avere la visione completa delle vendite per qualsiasi tipologia di prodotto ordinate in maniera decrescente in vendite.

```
-- NUMERO DI VENDITE PER PRODOTTO
(select prodotto.uva_qualità as "Nome Prodotto", count(prodotto.codice) as
"Vendite"
from ordine, composizione, prodotto
where
    ordine.numero = composizione.ordine and
    composizione.prodotto = prodotto.codice and
    prodotto.tipo = 'U'
group by
    prodotto.codice)

union

(select prodotto.vino_denominazione as "Nome Prodotto", count(prodotto.codice) as
"Vendite"
from ordine, composizione, prodotto
where
    ordine.numero = composizione.ordine and
    composizione.prodotto = prodotto.codice and
    prodotto.tipo = 'V'
group by
    prodotto.codice)

union

(select prodotto.aceto_denominazione as "Nome Prodotto", count(prodotto.codice) as
"Vendite"
from ordine, composizione, prodotto
where
    ordine.numero = composizione.ordine and
    composizione.prodotto = prodotto.codice and
    prodotto.tipo = 'A'
group by
    prodotto.codice)

order by "Vendite" desc;
```

7. Viste

<i>Workpackage</i>	<i>Task</i>	<i>Responsabile</i>
WP4	Viste	Caso Antonio

7.1. Vista *Vendite Trimestrali*

La vista *vendite_trimestrali* permette, tramite la join fra prodotto, composizione e ordine, di visualizzare per ogni prodotto quanto si è guadagnato negli ultimi tre mesi.

```
-- VISTA: VENDITE TRIMESTRALI
create or replace view vendite_trimestrali(prodotto,prezzo) as
  select
    P.codice as prodotto,
    sum(C.quantità*(p.prezzo_unitario-(p.prezzo_unitario*sconto/100)))
  from
    prodotto as P
    join composizione as C on (P.codice = C.prodotto)
    join ordine as O on (C.ordine=O.numero)
  where current_date - O.data < interval '3 months'
  group by p.codice;
```

7.1.1. Query con Vista: Vendite Trimestrali per viticoltore

Nella query sottostante si sfrutta questa vista per poter vedere i guadagni di ogni viticoltore in quanto ogni prodotto è associato ad un viticoltore.

```
-- VENDITE TRIMESTRALI PER VITICOLTORE
select
  v.nome,
  v.cognome,
  v.azienda,
  sum(vt.prezzo)
from
  vendite_trimestrali as vt
  join prodotto as p on(vt.prodotto=p.codice)
  join vigneto as vg on(p.provenienza=vg.id)
  join viticoltore as v on(vg.proprietario=v.cf)
group by
  v.cf,
  v.nome,
  v.cognome,
  v.azienda;
```

8. Trigger

8.1. Trigger inizializzazione: *Partecipazione obbligatoria* *Composizione*

Workpackage	Task	Responsabile
WP1	Trigger inizializzazione/popoloamento database	Amendola Miriam

Il trigger *check_composizione* garantisce che per ogni ordine inserito vi sia una corrispondenza in composizione e che quindi l'ordine non sia vuoto implementando il vincolo di partecipazione obbligatoria della relazione composizione.

```
-- TRIGGER INIZIALIZZAZIONE (IMPLEMENTAZIONE PARTECIPAZIONE OBBLIGATORIA)
-- PARTECIPAZIONE OBBLIGATORIA RELAZIONE COMPOSIZIONE
create or replace function at_least_one_composizione() returns trigger as $$
begin
    if (exists (select numero from ordine
                where numero not in (select ordine from composizione)
                )) then
        raise exception 'Ordine privo di prodotti, operazione
annullata';
    end if;
    return new;
end $$ language plpgsql;

create trigger check_composizione
after insert on ordine
for each row
execute procedure at_least_one_composizione();

create trigger check_composizione
after update or delete on composizione
for each row
execute procedure at_least_one_composizione();
```

8.2. Trigger per vincoli aziendali

<i>Workpackage</i>	<i>Task</i>	<i>Responsabile</i>
WP4	Trigger per vincoli aziendali	Caso Antonio

8.2.1. Trigger: Ricevuta

Il trigger *Ricevuta* implementa la logica di calcolo del totale di un ordine.

Per ogni articolo inserito nell'ordine (a cui si accede tramite composizione) se ne calcola l'IVA da pagare e si applica eventualmente uno sconto. Inoltre, totale viene determinato anche considerando la spedizione.

Per ogni composizione facente parte dell'ordine si aggiorna la ricevuta inserendo l'iva pagata, il risparmio (lo sconto totale ricevuto sull'acquisto), le spese di spedizione e il totale. **Questo trigger implementa i vincoli aziendali RD1, RD2, RD3, RD4.**

```
-- TRIGGER RICEVUTA
create or replace function calcolo_ricevuta() returns trigger as $$
    declare subtot decimal(8,2);
    declare percentuale_iva INTEGER;
    declare sped decimal(8,2);
    declare scorporo_iva decimal(8,2);
    declare sconto_tipo_prodotto INTEGER;

    begin
        sconto_tipo_prodotto = new.sconto;
        if (not exists (select * from ricevuta where ordine = new.ordine)) then

            if((select tipo from prodotto where new.prodotto = prodotto.codice) =
            'U') then
                percentuale_iva = 4;
            else
                percentuale_iva = 22;
            end if;
            select ((percentuale_iva*prezzo_unitario/100) + prezzo_unitario) *
            new.quantità, ((percentuale_iva*prezzo_unitario/100)) * new.quantità from prodotto
            where new.prodotto = prodotto.codice into subtot, scorporo_iva;

            if (subtot<50.00) then
                sped = 7.00;
            else
                sped = 0.00;
            end if;

            insert into ricevuta (ordine, subtotale, sconto, iva, spedizione,
            totale)
                values(new.ordine, subtot, (sconto_tipo_prodotto*subtot/100),
            scorporo_iva, sped, ((subtot)-(sconto_tipo_prodotto*subtot/100)) + sped);

        else
```

```

        if((select tipo from prodotto where new.prodotto = prodotto.codice) =
'U') then
            percentuale_iva = 4;
        else
            percentuale_iva = 22;
        end if;

        select ((percentuale_iva*prezzo_unitario/100) + prezzo_unitario) *
new.quantità, ((percentuale_iva*prezzo_unitario/100)) * new.quantità from prodotto
where new.prodotto = prodotto.codice into subtot, scorporo_iva;

        if((select subtotale+subtot from ricevuta where ricevuta.ordine =
new.ordine)<50) then
            sped = 7.00;
        else
            sped = 0.00;
        end if;

        update ricevuta set
            subtotale = ricevuta.subtotale+subtot,
            sconto = ricevuta.sconto + (sconto_tipo_prodotto*subtot/100),
            iva=ricevuta.iva+scorporo_iva,
            spedizione = sped,
            totale=(ricevuta.subtotale- ricevuta.sconto) + ((subtot)-
(sconto_tipo_prodotto*subtot/100)) + sped
        where ricevuta.ordine = new.ordine;

        if (subtot<50.00) then
            sped = 7.00;
        else
            sped = 0.00;
        end if;
    end if;

    return new;
end $$ language plpgsql;

create trigger ricevuta
    before insert on composizione
    for each row
    execute procedure calcolo_ricevuta();

```

Nota: nel file trigger.sql sono stati riportati anche altri trigger banali e simili a quelli descritti in precedenza che sono già stati implementati.