



# PROGETTO DI BASI DI DATI

Progettazione e realizzazione di un Sistema Informativo  
di supporto alla gestione di un'Azienda Vinicola

Anno Accademico 2018/2019  
Università degli Studi di Trieste  
Professore Alfredo Cuzzocrea

Massimo Palmisano, Matteo Merz

# Indice

Introduzione al Caso di Studio.....	1
Analisi dei requisiti .....	2
Conferimento dell'uva.....	2
Lavorazione dei mosti.....	3
Analisi dei prodotti in lavorazione.....	3
Specifica delle operazioni .....	4
Schema Concettuale e documentazione associata .....	5
Schema Concettuale.....	5
Dizionario dei Dati .....	7
Tavola dei volumi.....	10
Regole Aziendali .....	11
Ristrutturazione dello Schema Concettuale.....	12
Analisi delle ridondanze .....	13
Eliminazione delle generalizzazioni .....	15
Partizionamento o Accorpamento di entità e associazioni .....	15
Eliminazione Attributi Composti .....	15
Scelta Identificatori Principali.....	15
Schema Concettuale Ristrutturato.....	16
Traduzione verso il modello relazionale.....	17
Schema logico.....	17
Documentazione Associata allo Schema Logico.....	18
Codifica SQL-DDL .....	19
Popolamento delle tabelle .....	26
Ruoli e utenti .....	39
Funzionalità applicativo.....	41
Segreteria .....	42
Cantina.....	57
Analisi .....	72
Amministrazione .....	77
Osservazioni.....	78
Query .....	79
Query elementari .....	79
Query complesse .....	84
Procedure Oracle PL/SQL .....	93

## Introduzione al Caso di Studio

Si intende progettare e realizzare un sistema informativo dedicato al supporto della gestione di un'azienda produttrice di vino, che opera su scala regionale ed è attiva dal 1° gennaio 2010. L'obiettivo è quello di automatizzare e gestire gli ambiti del conferimento dell'uva, della lavorazione e dell'analisi dei mosti, nonché di mantenere un'anagrafica di produttori e vigneti.

L'azienda si compone di più cantine dislocate sul territorio, cui sono affiliati diversi produttori. Questi ultimi sono proprietari di vigneti, che in base alla loro posizione e resa qualificano il vino con varie denominazioni. Per poter consegnare l'uva all'azienda, i produttori devono effettuare una prenotazione presso la cantina alla quale sono affiliati.

Le cantine presentano al loro interno vasche atte alla lavorazione dei lotti che, in base al prodotto finito che si vuole ottenere, verranno sottoposti a particolari processi. È infatti possibile a partire da uno stesso lotto ottenere prodotti finiti distinti. Essendo sottoposto a numerosi processi, un lotto può attraversare fasi diverse, di cui è importante tenere traccia. Inoltre, per assicurare la qualità del prodotto, da ogni lotto vengono estratti periodicamente dei campioni, di cui viene svolta un'attenta analisi dei parametri di interesse.

## Analisi dei requisiti

In questa sezione saranno esposti nel dettaglio i requisiti della base di dati. La trattazione sarà divisa in paragrafi, al fine di facilitarne la comprensione.

### Conferimento dell'uva

Ogni **cantina** dell'azienda è identificata univocamente da un codice ed è caratterizzata dal nome e dagli indirizzi dello stabilimento produttivo e della sede sociale. Ad una cantina possono essere affiliati più produttori.

Per ogni **produttore**, identificato da un codice univoco nell'ambito della base di dati, si intende memorizzare la partita IVA, il codice fiscale e i dati anagrafici (nome, cognome, sesso, data e luogo di nascita, residenza e, nel caso fosse diverso dalla residenza, domicilio). Un produttore è in ogni momento affiliato a una singola cantina, ma può decidere di cambiarla nel corso del tempo; si vuole tenere traccia sia delle affiliazioni attuali che di quelle passate.

Ogni produttore gestisce almeno un vigneto e non è consentita la gestione condivisa. Un **vigneto** è identificato da un codice univoco nell'ambito della base di dati ed è caratterizzato dal nome del vitigno piantato, dall'età, dagli anni di impianto e di inizio produzione, dall'estensione e dalla posizione. Si intende suddividere le varietà d'uva coltivata non solo in base al vitigno ma anche in base alla denominazione, in modo che, per esempio, il tipo di uva Prosecco DOGC, pur essendo prodotta dallo stesso vitigno, sia considerato diverso dal Prosecco DOC.

Il produttore deve dichiarare, prima di ogni vendemmia, la denominazione con cui porterà l'**uva** prodotta da ognuno dei suoi vigneti. Tale scelta deve essere ristretta alle denominazioni possibili del singolo vigneto, determinate sulla base del vitigno impiantato e della posizione, ed è in ogni caso vincolata a una resa per ettaro massima e ad un grado zuccherino minimo, come specificato nel disciplinare.

I produttori possono conferire l'uva esclusivamente nella cantina cui sono affiliati. Ogni conferimento, inoltre, deve essere preceduto da una **prenotazione**, da effettuarsi almeno tre giorni prima dello stesso, in cui vanno specificati il colore dell'uva da portare, la quantità, la modalità di raccolta (manuale o meccanica) e la data del conferimento.

Le varie cantine presentano una quantità massima di uva che possono giornalmente accettare ed è perciò necessario evitare di concedere ulteriori prenotazioni nel caso in cui tale quantità sia già stata raggiunta per un determinato giorno di vendemmia.

Il produttore che ha prenotato il conferimento di un certo quantitativo di uva può effettuarlo in uno o più carichi; è compito del sistema informativo sommare quanto è stato conferito nei singoli viaggi e controllare la corrispondenza del risultato con il quantitativo complessivo prenotato.

Per ogni **carico** di uva in entrata è necessario memorizzare il peso, il tipo di uva e alcune proprietà derivate dalle prime analisi: PH, grado, quantità di materiale esterno (da misurare in una scala da 1 a 5) e integrità degli acini (un carattere binario da considerare solo nel caso in cui la modalità di raccolta sia meccanica).

## Lavorazione dei mosti

La produzione è organizzata in **lotti**, per ognuno dei quali si è interessati alle caratteristiche del prodotto (famiglia di vino, denominazione, colore, annata), la cantina di origine e la possibilità di essere usato per tagli. I lotti possono essere in vari stadi (denominati **fasi**) di lavorazione: si intende facilitare il compito del capo cantina suggerendo per ogni lotto, oltre alla fase di lavorazione corrente, quella successiva, in base al tipo di **vinificazione** scelta.

Ogni cantina contiene una o più vasche per la conservazione e/o la lavorazione dei lotti; una **vasca** è identificata da un codice, univoco nell'ambito della cantina in cui è posta, ed è caratterizzata da capienza e tipologia. Si desidera poter sapere in ogni momento che lotto è contenuto in una particolare vasca, quali sono le sue caratteristiche e qual è la sua storia.

La storia di un lotto consiste nell'elenco dei processi ai quali quel determinato lotto è stato sottoposto. Per ogni lavorazione interessano in particolare le quantità di prodotto in gioco (quantità iniziale, quantità finale e perdita) e le date di inizio e di fine lavorazione.

Un **processo** è caratterizzato da nome e durata tipica e può richiedere l'aggiunta di una certa quantità di uno o più **prodotti**. Si vuole mantenere un catalogo di tali prodotti, in modo che, di ognuno, si conosca il nome, la categoria, una breve descrizione e, eventualmente, alcuni commenti o consigli utili per l'utilizzo.

Non tutte le cantine appartenenti all'azienda predispongono delle attrezzature necessarie per attuare tutti i processi: è dunque da implementare anche la possibilità di trasferire i lotti da una cantina a un'altra.

Si intende infine mantenere un registro in cui inserire le quantità di **scarto** prodotte al termine di ogni giornata. Lo scarto può essere sia liquido (feccia liquida), che solido (raspi, vinaccia, feccia solida): è quindi necessario inserire un parametro che specifichi l'unità di misura (chilogrammi o litri) adeguata.

## Analisi dei prodotti in lavorazione

Da ogni lotto possono essere estratti uno o più **campioni**, per ognuno dei quali si intende memorizzare la data e l'ora in cui è stato prelevato, quelle in cui sono stati resi disponibili i risultati, la matricola dell'operatore che ha eseguito le analisi e un'eventuale descrizione.

Le analisi sui campioni possono riguardare vari parametri, in base alla fase di lavorazione del lotto e/o a situazioni particolari che si desidera monitorare. Per ogni **parametro** analizzato è necessario conoscere il nome (univoco), l'esito della misurazione, corredata dell'opportuna unità di misura e del numero di cifre significative, e lo strumento utilizzato.

## Specifica delle operazioni

Le operazioni principali che il sistema informativo dovrà essere in grado di supportare sono le seguenti.

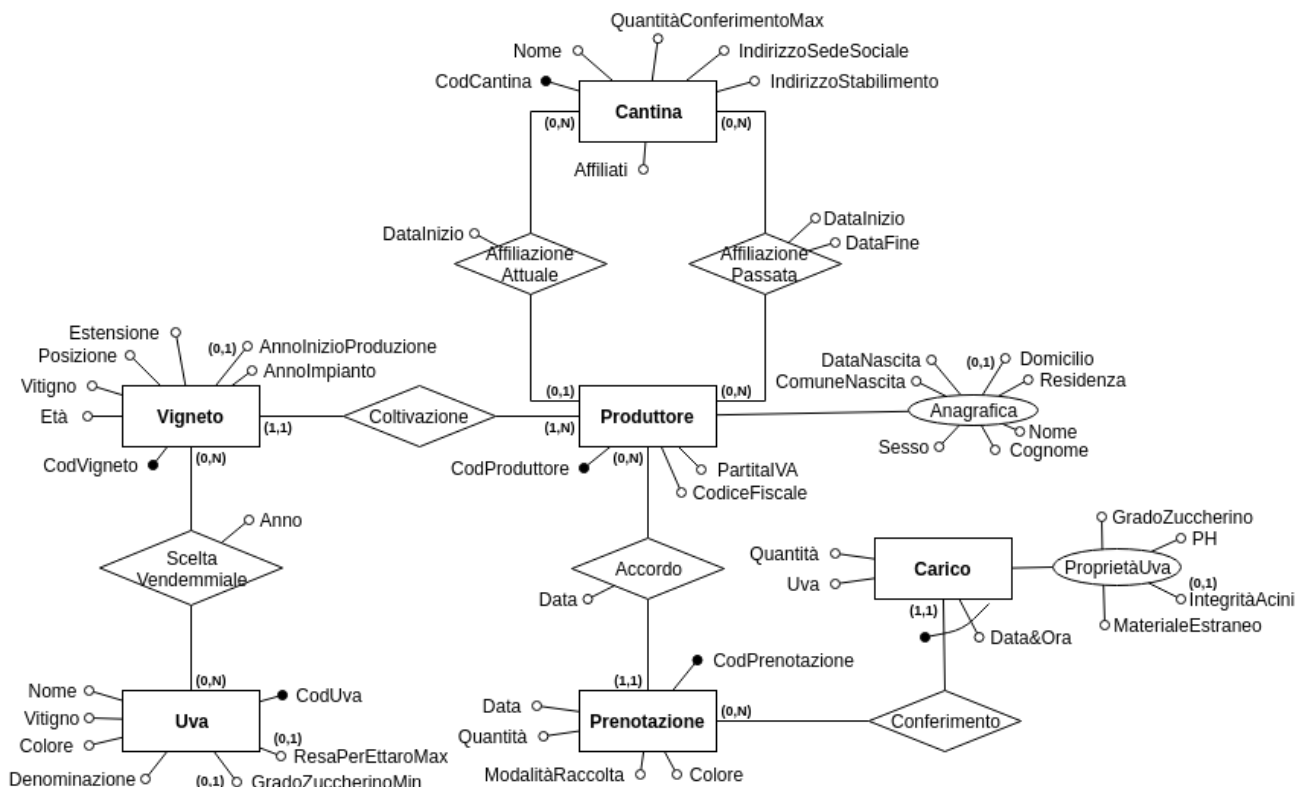
- Operazione 1:** registrare un nuovo vigneto;
- Operazione 2:** registrare la dichiarazione vendemmiale di un certo anno per un particolare vigneto;
- Operazione 3:** inserire una prenotazione;
- Operazione 4:** registrare l'arrivo di un carico di uva;
- Operazione 5:** determinare la quantità di tutti i tipi di uva conferita in un giorno;
- Operazione 6:** determinare le quantità di scarto prodotte in un giorno;
- Operazione 7:** modificare l'affiliazione di un produttore;
- Operazione 8:** creare un nuovo lotto;
- Operazione 9:** determinare il lotto contenuto in una vasca e il suo stato;
- Operazione 10:** determinare i processi eseguiti su un lotto;
- Operazione 11:** determinare la quantità di perdite durante la lavorazione di un lotto;
- Operazione 12:** cambiare il prodotto da usare per un determinato processo;
- Operazione 13:** trasferire un lotto da una cantina ad un'altra;
- Operazione 14:** modificare lo stato di un lotto;
- Operazione 15:** determinare la successiva fase di lavorazione di un lotto;
- Operazione 16:** registrare un nuovo campione, con tutti i dati ricavati dall'analisi;
- Operazione 17:** registrare un nuovo produttore affiliato a una determinata cantina;
- Operazione 18:** stampare il numero di produttori affiliati a una particolare cantina;
- Operazione 19:** registrare un nuovo lotto;
- Operazione 20:** determinare il nome della cantina di origine di un lotto.

# Schema Concettuale e documentazione associata

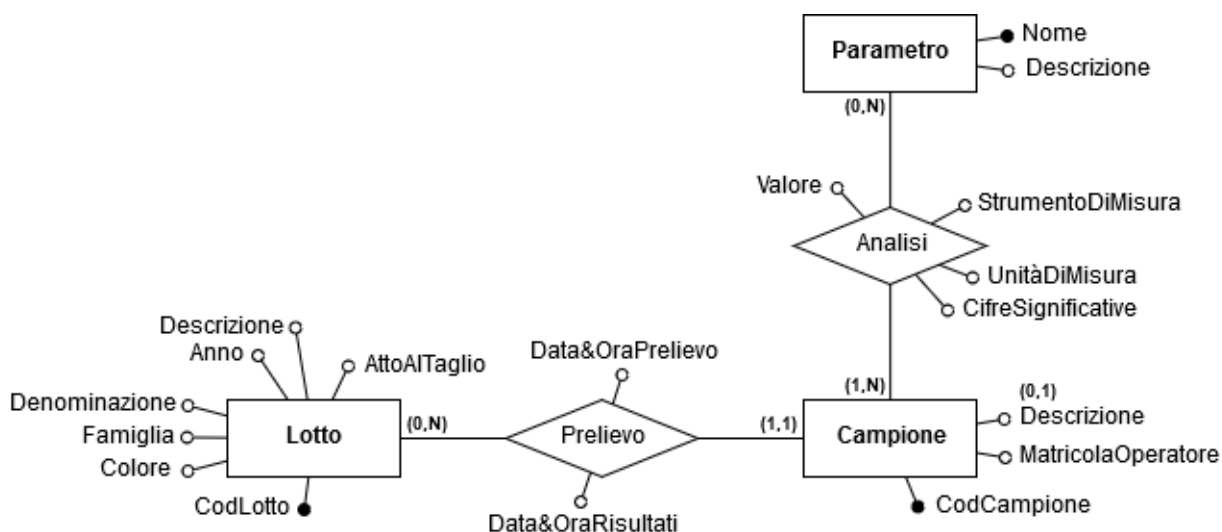
## Schema Concettuale

Sulla base dei requisiti precedentemente elencati, la progettazione concettuale ha prodotto il seguente schema. Esso è stato diviso in tre sezioni, in modo da migliorarne la comprensione.

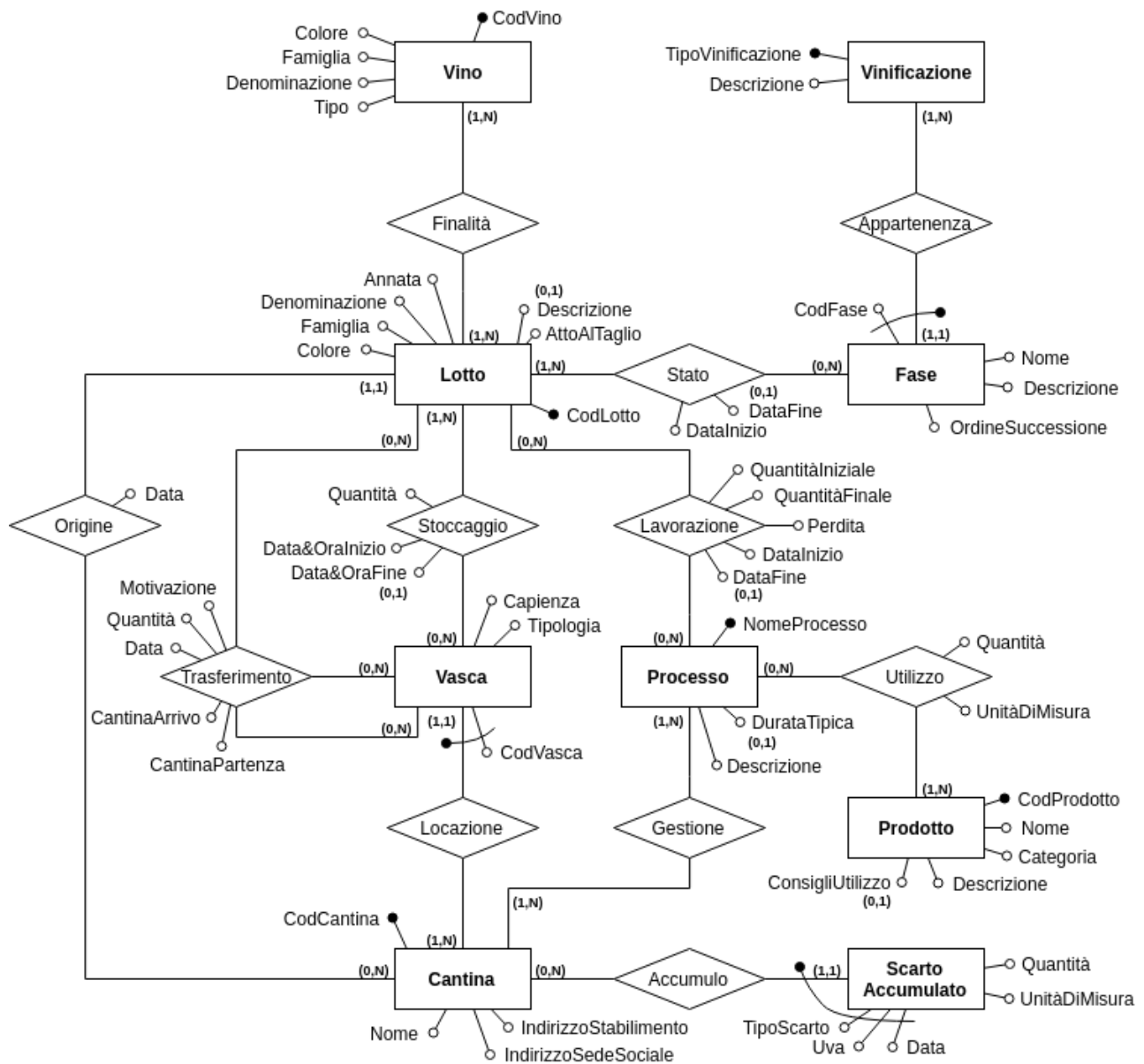
### Conferimento dell'uva



### Analisi dei prodotti in lavorazione



## Lavorazione dei mosti





## Dizionario dei Dati

Segue una breve descrizione di tutte le entità e tutte le relazioni dello schema concettuale.

### Elenco delle ENTITÀ

Entità	Descrizione	Attributi	Identificatori
<b>Uva</b>	Contiene i tipi di uva coltivati. Si ricordi come lo stesso vitigno possa produrre più tipi di uva in base alla posizione del vigneto (e quindi alla denominazione).	CodUva Nome Vitigno Denominazione Colore GradoZuccherinoMin* ResaPerEttaroMax*	CodUva
<b>Vigneto</b>	Rappresenta il luogo in cui viene prodotta l'uva. L'attributo <i>Estensione</i> è espresso in m <sup>2</sup> .	CodVigneto Vitigno AnnoImpianto AnnoInizioProduzione* Età Estensione Posizione	CodVigneto
<b>Produttore</b>	Raccoglie i dati dei coltivatori affiliati all'azienda.	CodProduttore PartitaIVA CodiceFiscale Anagrafica (Nome, Cognome, Sesso, Data di nascita, Luogo di nascita, Domicilio, Residenza)	CodProduttore PartitaIVA CodiceFiscale
<b>Prenotazione</b>	Necessaria affinché un produttore possa conferire l'uva raccolta. L'attributo <i>Quantità</i> è espresso in kg.	CodPrenotazione Data Quantità Colore Modalità raccolta	CodPrenotazione
<b>Carico</b>	Raccoglie i dati relativi ai carichi di uva conferiti dai produttori. L'attributo <i>Quantità</i> è espresso in kg.	Ora Data Uva Quantità ProprietàUva (PH, Grado, IntegritàAcini*, MaterialeEstraneo)	CodPrenotazione, Data, Ora
<b>Cantina</b>	Struttura adibita alla raccolta dell'uva e/o alla sua lavorazione. L'attributo <i>Quantità di conferimento massima</i> è espresso in kg.	CodCantina Nome Pianta Affiliati QuantitàConferimentoMax IndirizzoSedeSociale IndirizzoStabilimento	CodCantina
<b>Scarto Accumulato</b>	Registra le quantità di scarti di lavorazione accumulati giorno per giorno. L'attributo <i>UnitàDiMisura</i> può assumere i valori "kg" oppure "l".	TipoScarto Uva Data Quantità UnitàDiMisura	TipoScarto, Uva, Data

<b>Vasca</b>	Contenitore in cui sono stoccati i prodotti in lavorazione. L'attributo <i>Capienza</i> è espresso in m <sup>3</sup> .	CodVasca Capienza Tipologia	CodVasca CodCantina
<b>Processo</b>	Operazioni per far passare il prodotto in lavorazione da una fase all'altra.	NomeProcesso DurataTipica Descrizione	NomeProcesso
<b>Prodotto</b>	Registro dei prodotti che possono essere aggiunti durante i vari processi.	CodProdotto Nome Descrizione Categoria ConsigliUtilizzo	CodProdotto
<b>Lotto</b>	Una porzione uniforme di materia prima da sottoporre alle lavorazioni.	CodLotto Colore Famiglia Denominazione Annata Descrizione* Atto al taglio	CodLotto
<b>Vino</b>	Raccoglie i vari tipi di vino che l'azienda può produrre.	CodVino Colore Famiglia Denominazione Tipo	CodVino
<b>Fase</b>	Contiene gli stati in cui i lotti possono essere durante la lavorazione.	CodFase Nome OrdineSuccessione Descrizione	CodFase, TipoVinificazione
<b>Vinificazione</b>	Elenca i vari tipi di vinificazione che l'azienda è in grado di svolgere.	TipoVinificazione Descrizione	TipoVinificazione
<b>Campione</b>	Descrive i campioni prelevati dai lotti per effettuare le analisi.	CodCampione Matricola operatore Descrizione*	CodCampione
<b>Parametro</b>	Parametri biochimici analizzabili.	NomeParametro Descrizione	NomeParametro

Sono stati indicati con \* quegli attributi che possono avere valori nulli.

## Elenco delle RELAZIONI

Relazione	Descrizione	Entità Coinvolte		Attributi
<b>Scelta Vendemmiale</b>	Scelta del tipo di uva che sarà conferire in un certo anno.	Uva Vigneto	(0,N) (0,N)	Anno
<b>Coltivazione</b>	Associa ogni vigneto al produttore che lo gestisce.	Produttore Vigneto	(1,N) (1,1)	
<b>Accordo</b>	Associa ogni prenotazione al relativo produttore.	Produttore Prenotazione	(0,N) (1,1)	Data
<b>Conferimento</b>	Associa i carichi di uva alla relativa prenotazione effettuata.	Prenotazione Carico	(0,N) (1,1)	
<b>Affiliazione Attuale</b>	Associa i produttori alle cantine cui sono attualmente affiliati.	Produttore Cantina	(0,1) (0,N)	DataInizio
<b>Affiliazione Passata</b>	Associa i produttori alle cantine cui sono stati in passato affiliati.	Produttore Cantina	(0,N) (0,N)	DataInizio Data Fine
<b>Accumulo Scarto</b>	Associa lo scarto alla cantina in cui è stato accumulato.	Cantina ScartoAccumulato	(0,N) (1,1)	
<b>Locazione</b>	Associa ogni vasca alla cantina a cui appartiene.	Vasca Cantina	(1,1) (1,N)	
<b>Gestione</b>	Associa ad ogni cantina i processi che è in grado di offrire.	Cantina Processo	(1,N) (1,N)	
<b>Utilizzo</b>	Associa ad ogni processo i prodotti necessari per la sua esecuzione.	Processo Prodotto	(0,N) (1,N)	Quantità UnitàDiMisura
<b>Trasferimento</b>	Descrive il trasferimento di un lotto da una cantina a un'altra. L'attributo <i>Quantità</i> è espresso in litri.	Lotto Vasca Vasca	(0,N) (0,N) (0,N)	CantinaPartenza CantinaArrivo Data Quantità Motivazione
<b>Lavorazione</b>	Associa ad ogni lotto i processi ai quali è stato sottoposto.	Lotto Processo	(0,N) (0,N)	QuantitàIniziale QuantitàFinale Perdita DataInizio DataFine
<b>Stoccaggio</b>	Associa ad ogni vasca il lotto contenuto. L'attributo <i>Quantità</i> è espresso in litri.	Vasca Lotto	(0,N) (1,N)	DataInizio DataFine Quantità
<b>Origine</b>	Associa ad ogni lotto la cantina in cui è stato creato.	Cantina Lotto	(0,N) (1,1)	Data
<b>Prelievo</b>	Associa ad ogni campione il lotto da cui è stato prelevato.	Lotto Campione	(0,N) (1,1)	Data&OraPrelievo Data&OraRisultati
<b>Analisi</b>	Associa ad ogni campione i parametri che sono stati analizzati e i relativi valori risultanti.	Campione Parametro	(1,N) (0,N)	Valore StrumentoDiMisura UnitàDiMisura CifreSignificative
<b>Finalità</b>	Associa ogni lotto ai prodotti finiti da esso ottenibili.	Lotto Vino	(1,N) (1,N)	
<b>Stato</b>	Associa i lotti alle relative fasi di lavorazione.	Lotto Fase	(1,N) (0,N)	Data
<b>Appartenenza</b>	Associa una fase al processo di vinificazione di cui fa parte.	Fase Vinificazione	(1,1) (1,N)	

## Tavola dei volumi

Segue la tavola dei volumi associata allo schema concettuale presentato.

Concetto	Tipo	Volume
Produttore	E	10.000
Coltivazione	R	50.000
Vigneto	E	50.000
SceltaVendemmiale	R	500.000
Uva	E	200
Accordo	R	150.000
Prenotazione	E	150.000
Conferimento	R	300.000
Carico	E	300.000
AffiliazionePassata	R	100
AffiliazioneAttuale	R	10.000
Cantina	E	15
Accumulo	R	400
ScartoAccumulato	E	400
Locazione	R	750
Vasca	E	750
Trasferimento	R	10.000
Stoccaggio	R	5.000.000
Gestione	R	200
Processo	E	50
Impiego	R	500
Prodotto	E	200
Lavorazione	R	5.000.000
Origine	R	1.000.000
Lotto	E	1.000.000
Finalità	R	3.000.000
Vino	E	300
Stato	R	5.000.000
Fase	E	50
Appartenenza	R	50
Vinificazione	E	10
Prelievo	R	10.000.000
Campione	E	10.000.000
Analisi	R	30.000.000
Parametro	E	100

## Regole Aziendali

Riportiamo le regole aziendali relative ai vincoli di integrità, denominate **Regole di Vincolo**. Le asserzioni che la base di dati deve soddisfare sono:

- RV1. L'attributo "vitigno" di Vigneto deve corrispondere all'attributo vitigno dell'uva che viene coltivata in esso.
- RV2. L'attributo "colore" deve assumere uno ed uno solo dei valori {Bianco, Rosso}
- RV3. Ogni data, tranne quelle relative ai produttori e ai vigneti, devono essere successive al 01/01/2010
- RV4. La quantità iniziale di lavorazione deve essere minore alla quantità finale sommata alla perdita
- RV5. Il trasferimento di un lotto deve avere la cantina di partenza diversa da quella di arrivo
- RV6. La data di "Accordo" deve essere precedente di almeno due giorni la data di "Prenotazione"
- RV7. La quantità di uva di un certo colore totale conferita in una data deve essere minore di una quantità

## Ristrutturazione dello Schema Concettuale

Per poter efficacemente portare a termine la ristrutturazione dello schema concettuale, è necessario focalizzarsi sulle operazioni presentate dopo l'analisi dei requisiti. Tra quelle elencate, sono presenti delle operazioni la cui frequenza si concentra nel periodo della vendemmia. Per questo motivo, riportiamo due Tavole delle Operazioni: la prima relativa alle operazioni la cui frequenza si distribuisce in maniera approssimativamente uniforme durante tutto l'arco dell'anno, la seconda relativa alle operazioni la cui frequenza si concentra durante il periodo vendemmiale.

Operazione	Tipo Operazione	Frequenza
Operazione 01	Interattiva	5 alla settimana
Operazione 06	Batch	2 al giorno
Operazione 07	Interattiva	10 all'anno
Operazione 09	Interattiva	5.000 al giorno
Operazione 10	Interattiva	6.000 al giorno
Operazione 11	Interattiva	5.000 al giorno
Operazione 12	Interattiva	10 all'anno
Operazione 16	Interattiva	100 al giorno
Operazione 17	Interattiva	100 all'anno
Operazione 18	Interattiva	10 al giorno
Operazione 19	Interattiva	500 al giorno
Operazione 20	Interattiva	10 al giorno

*Tavola delle Operazioni - Periodo Vendemmiale*

Operazione	Tipo Operazione	Frequenza
Operazione 02	Interattiva	6.000 al giorno
Operazione 03	Interattiva	2.000 al giorno
Operazione 04	Interattiva	4.000 al giorno
Operazione 05	Batch	2 al giorno
Operazione 08	Interattiva	1.000 al giorno
Operazione 13	Interattiva	20 al giorno
Operazione 14	Interattiva	1.000 al giorno
Operazione 15	Interattiva	1.000 al giorno

*Tavola delle Operazioni - Periodo Annuale*

Si articola la ristrutturazione dello schema concettuale nei seguenti punti:

1. Analisi delle Ridondanze;
2. Eliminazione delle Generalizzazioni;
3. Partizionamento o Accorpamento di entità e associazioni;
4. Scelta degli identificatori principali.

## Analisi delle ridondanze

Si osservi che l'attributo *Produttori Affiliati* di *Cantina* è una ridondanza, in quanto ricavabile dalla relazione *Affiliazione Attuale*. È opportuno quindi svolgere un'analisi di carico per valutare se la presenza di tale ridondanza sia opportuna, in quanto coinvolge l'Operazione 17 "aggiungere una nuova affiliazione a una determinata cantina" e l'Operazione 18 "stampare il numero di produttori affiliati a una particolare cantina".

Riportiamo la Tavola degli Accessi relativa all'operazione 17 nel caso di **presenza della ridondanza**

Concetto	Costrutto	Numero Accessi	Tipo
Produttore	Entità	1	Scrittura
Affiliazione Attuale	Relazione	1	Scrittura
Cantina	Entità	1	Lettura
Cantina	Entità	1	Scrittura

*Tavola Accessi Operazione 17 in presenza di ridondanza*

Mentre per l'Operazione 18 abbiamo, sempre in presenza di ridondanza

Concetto	Costrutto	Numero Accessi	Tipo
Cantina	Entità	1	Lettura

*Tavola Accessi Operazione 18 in presenza di ridondanza*

Supponendo ragionevolmente che gli accessi in scrittura abbiano un costo doppio rispetto ai costi in lettura, il costo complessivo delle due operazioni è

$$(2 + 2 + 1 + 2) \cdot 100 + 1 \cdot 10 \cdot 365 = 4350 \text{ Accessi all'anno}$$

In **assenza di ridondanza** invece abbiamo, per l'operazione 17

Concetto	Costrutto	Numero Accessi	Tipo
Produttore	Entità	1	Scrittura
Affiliazione Attuale	Relazione	1	Scrittura

*Tavola Accessi Operazione 17 in assenza di ridondanza*

mentre per l'operazione 18

Concetto	Costrutto	Numero Accessi	Tipo
Cantina	Entità	1	Lettura
Affiliazione Attuale	Relazione	650	Lettura

*Tavola Accessi Operazione 18 in assenza di ridondanza*

da cui si evince

$$(2 + 2) \cdot 100 + (650 + 1) \cdot 10 \cdot 356 = 2.317.960 \text{ accessi all'anno}$$

Quindi, considerando che ci sono in media meno di 1.000 produttori per cantina e ci sono 15 cantine, l'attributo ridondante *Produttori Affiliati* necessiterebbe di al più, supponendo di utilizzare 4 byte di memoria (più che sufficienti per i nostri scopi), 60 byte di memoria. Essendo l'occupazione di spazio necessario per la ridondanza trascurabile e si avrebbe un risparmio di circa 2.313.610 accessi all'anno, la ridondanza sarà mantenuta.

Aggiungendo l'attributo *data origine* all'entità *Lotto*, la relazione *Origine* costituisce una ridondanza. La presenza o meno di questa coinvolge l'operazione 19 "registrare un nuovo lotto" e l'operazione 20 "determinare il nome della cantina di origine di un lotto": è opportuno quindi svolgere un'analisi di carico che stabilisca se la relazione *Origine* vada eliminata.

Trattiamo per primo il caso di **presenza di ridondanza**: riportiamo quindi le Tavole degli Accessi relative alle operazioni 19 e 20.

Concetto	Costrutto	Tipo Accesso	Numero Accessi
Lotto	Entità	Scrittura	1
Origine	Relazione	Scrittura	1

*Tavola Accessi Operazione 19 in presenza di ridondanza*

Concetto	Costrutto	Tipo Accesso	Numero Accessi
Lotto	Entità	Lettura	1
Origine	Relazione	Lettura	1
Cantina	Entità	Lettura	1

*Tavola Accessi Operazione 20 in presenza di ridondanza*

abbiamo quindi un totale di  $(1 \cdot 2 + 1 \cdot 2) \cdot 500 + (1 + 1 + 1) \cdot 10 = 2.030$  accessi al giorno.

Trattiamo ora il caso di assenza di ridondanza: riportiamo per questo caso le relative Tavole degli Accessi. Siccome i lotti che hanno subito un trasferimento sono in rapporto 1:100 rispetto a quelli che non lo hanno subito, nella seconda Tavola degli Accessi non si è fatto il passaggio per la relazione Trasferimento in quanto il numero degli accessi sarebbe trascurabile.

Concetto	Costrutto	Tipo Accesso	Numero Accessi
Lotto	Entità	Scrittura	1

*Tavola degli Accessi Operazione 19 in assenza di ridondanza*

Concetto	Costrutto	Tipo Accesso	Numero Accessi
Lotto	Entità	Lettura	1
Stoccaggio	Relazione	Lettura	1
Vasca	Entità	Lettura	1
Locazione	Relazione	Lettura	1
Cantina	Entità	Lettura	1

*Tavola degli Accessi Operazione 20 in assenza di ridondanza*

abbiamo quindi un costo complessivo di  $(1 \cdot 2) \cdot 500 + (1 + 1 + 1 + 1 + 1) \cdot 10 = 1.050$  accessi al giorno. In presenza di ridondanza quindi non solo avremmo un costo in termini di memoria, ma anche in termini di accessi, in particolare di quasi 1000 accessi. Tale ridondanza quindi sarà eliminata.



## Eliminazione delle generalizzazioni

Non sono presenti generalizzazioni.

## Partizionamento o Accorpamento di entità e associazioni

L'analisi di carico presentata non porta ad alcun partizionamento e/o accorpamento di alcuna entità o associazione.

## Eliminazione Attributi Composti

Siccome tutti gli attributi composti non sono né opzionali né multivalore, li trattiamo come attributi singoli associati alle loro relative entità

## Scelta Identificatori Principali

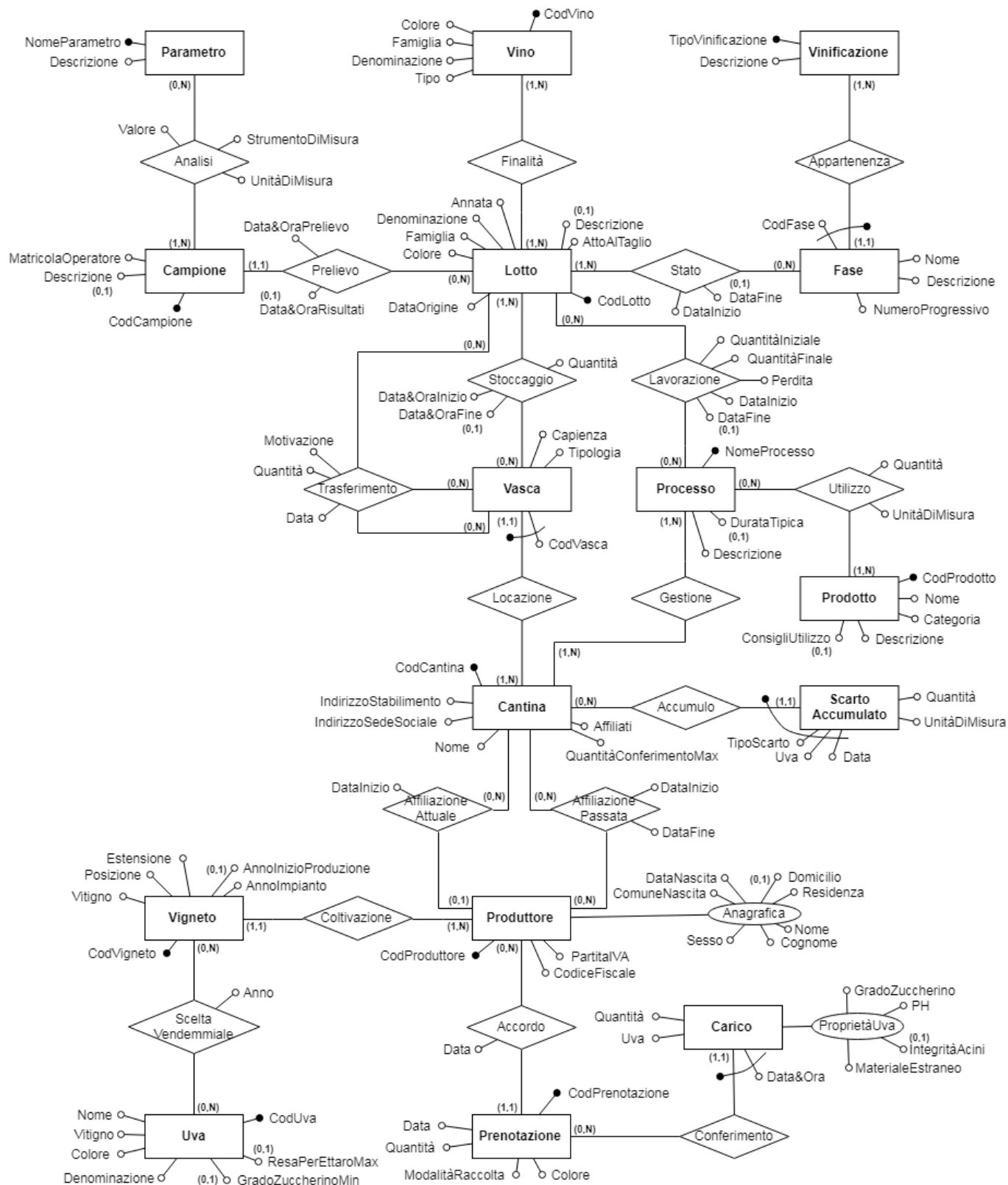
L'entità *Produttore* presenta tre chiavi: *Partita Iva*, *Codice Fiscale* e *CodProduttore*. Per questioni di efficienza di implementazione scegliamo come chiave l'attributo *CodProduttore*.

L'entità *Cantina* presenta due chiavi: *IndirizzoStabilimento* e *CodCantina*. È evidentemente più comodo e utile scegliere come identificatore principale *CodCantina*.

## Schema Concettuale Ristrutturato

Sarà ora presentato lo schema concettuale ristrutturato, prodotto delle considerazioni esposte nei paragrafi precedenti.

Gli attributi multipli *Anagrafica* in *Produttore* e *ProprietàUva* in *Carico* non sono stati al livello grafico cancellati (a livello logico invece sì) per non inficiare la leggibilità dello schema.



## Traduzione verso il modello relazionale

La progettazione logica ha prodotto come risultato il seguente schema logico, a cui bisogna affiancare la documentazione esposta nel paragrafo successivo.

### Schema logico

**Cantina** (CodCantina, Nome, Affiliati, QuantitaConferimentoMax, IndirizzoStabilimento, IndirizzoSedeSociale)

**Produttore** (CodProduttore, PartitaIVA, CodiceFiscale, Nome, Cognome, Sesso, DataNascita, ComuneNascita, Residenza, Domicilio, CantinaAffiliazione, DataInizioAffiliazione)

**AffiliazionePassata** (CodProduttore, CodCantina, DataInizio, DataFine)

**Vigneto** (CodVigneto, Estensione, Posizione, AnnoImpianto, AnnoInizioProduzione, Vitigno, CodProduttore)

**Uva** (CodUva, Nome, Vitigno, Colore, Denominazione, ResaPerEttaroMax, GradoZuccherinoMin)

**SceltaVendemmiale** (CodVigneto, CodUva, Anno)

**Prenotazione** (CodPrenotazione, DataConferimento, Quantita, Colore, ModalitaRaccolta, DataAccordo, CodProduttore)

**Carico** (Prenotazione, DataEOra, Uva, Quantita, PH, GradoZuccherino, IntegritaAcini, MaterialeEstraneo)

**ScartoAccumulato** (CodCantina, TipoScarto, Uva, Data, Quantita, UnitàDiMisura)

**Vasca** (CodVasca, CodCantina, Tipologia, Capienza)

**Lotto** (CodLotto, Annata, Denominazione, Famiglia, Colore, AttoAlTaglio, DataOrigine, Descrizione)

**Stoccaggio** (CodLotto, CodVasca, CodCantina, DataEOraInizio, DataEOraFine, Quantita)

**Trasferimento** (VascaPartenza, CantinaDestinazione, VascaPartenza, CantinaDestinazione, Lotto, Data, Quantita, Motivazione)

**Processo** (NomeProcesso, DurataTipica, Descrizione)

**Gestione** (Cantina, Processo)

**Lavorazione** (Lotto, Processo, QuantitaIniziale, QuantitaFinale, Perdita, DataInizio, DataFine)

**Prodotto** (CodProdotto, Nome, Categoria, Descrizione, ConsigliUtilizzo)

**Utilizzo** (Processo, Prodotto, Quantita, UnitàDiMisura)

**Vino** (CodVino, Colore, Famiglia, Denominazione, Tipo)

**Finalita** (Lotto, Vino)

**Vinificazione** (TipoVinificazione, Descrizione)

**Fase** (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)

**Stato** (Fase, Vinificazione, Lotto, DataInizio, DataFine)

**Campione** (CodCampione, MatricolaOperatore, DataEOraPrelievo, DataEOraRisultati, Descrizione, Lotto)

**Parametro** (NomeParametro, Descrizione)

**Analisi** (Campione, Parametro, StrumentoDiMisura, Valore, UnitàDiMisura)

## Documentazione Associata allo Schema Logico

Oltre alle Regole Aziendali, alla documentazione associamo la parte relativa ai vincoli di integrità referenziale che il nostro database deve soddisfare.

- *CodProduttore* nella relazione **AffiliazionePassata** e *CodProduttore* nella relazione **Produttore**
- *CodCantina* nella relazione **AffiliazionePassata** e *CodCantina* nella relazione **Cantina**
- *CodProduttore* nella relazione **Vigneto** e *CodProduttore* nella relazione **Produttore**
- *CodProduttore* nella relazione **Prenotazione** e *CodProduttore* nella relazione **Produttore**
- *CodVigneto* nella relazione **SceltaVendemmiale** e *CodVigneto* nella relazione **Vigneto**
- *CodUva* nella relazione **SceltaVendemmiale** e *CodUva* nella relazione **Uva**
- *CodPrenotazione* nella relazione **Carico** e *CodPrenotazione* nella relazione **Prenotazione**
- *Uva* nella relazione **Carico** e *CodUva* nella relazione **Uva**
- *CodCantina* nella relazione **ScartoAccumulato** e *CodCantina* nella relazione **Cantina**
- *CodCantina* nella relazione **Vasca** e *CodCantina* nella relazione **Cantina**
- *Vino* nella relazione **Lotto** e *CodVino* nella relazione **Vino**
- *CodCantinaPart* nella relazione **Trasferimento** e *CodCantina* nella relazione **Cantina**
- *CodCantinaDest* nella relazione **Trasferimento** e *CodCantina* nella relazione **Cantina**
- *CodVascaPart* nella relazione **Trasferimento** e *CodVasca* nella relazione **Vasca**
- *CodVascaDest* nella relazione **Trasferimento** e *CodVasca* nella relazione **Vasca**
- *CodLotto* nella relazione **Trasferimento** e *CodLotto* nella relazione **Lotto**
- *CodLotto* nella relazione **Stoccaggio** e *CodLotto* nella relazione **Lotto**
- *CodCantina* nella relazione **Stoccaggio** e *CodCantina* nella relazione **Cantina**
- *CodVasca* nella relazione **Stoccaggio** e *CodVasca* nella relazione **Vasca**
- *CodCantina* nella relazione **Gestione** e *CodCantina* nella relazione **Cantina**
- *NomeProcesso* nella relazione **Gestione** e *NomeProcesso* nella relazione **Processo**
- *NomeProcesso* nella relazione **Utilizzo** e *NomeProcesso* nella relazione **Processo**
- *CodProdotto* nella relazione **Utilizzo** e *CodProdotto* nella relazione **Prodotto**
- *CodLotto* nella relazione **Lavorazione** e *CodLotto* nella relazione **Lotto**
- *NomeProcesso* nella relazione **Lavorazione** e *NomeProcesso* nella relazione **Processo**
- *CodLotto* nella relazione **Finalita** e *CodLotto* nella relazione **Lotto**
- *CodVino* nella relazione **Finalita** e *CodVino* nella relazione **Vino**
- *CodFase* nella relazione **Stato** e *CodFase* nella relazione **Fase**
- *CodLotto* nella relazione **Stato** e *CodLotto* nella relazione **Lotto**
- *TipoVinificazione* nella relazione **Stato** e *TipoVinificazione* nella relazione **Vinificazione**
- *TipoVinificazione* nella relazione **Fase** e *TipoVinificazione* nella relazione **Vinificazione**
- *CodLotto* nella relazione **Campione** e *CodLotto* nella relazione **Lotto**
- *CodCampione* nella relazione **Analisi** e *CodCampione* nella relazione **Campione**
- *NomeParametro* nella relazione **Analisi** e *NomeParametro* nella relazione **Parametro**

# Codifica SQL-DDL

In questo paragrafo è riportato il codice SQL-DDL utilizzato per la creazione delle tabelle nella base di dati.

```
CREATE TABLE Cantina(  
    CodCantina          CHAR(2),  
    Nome                VARCHAR(50)    NOT NULL,  
    Affiliati            INTEGER        NOT NULL,  
    QuantitaConferimentoMax INTEGER    NOT NULL,  
    IndirizzoStabilimento VARCHAR(100) NOT NULL,  
    IndirizzoSedeSociale VARCHAR(100)  NOT NULL,  
  
    PRIMARY KEY (CodCantina),  
    CHECK (QuantitaConferimentoMax > 0),  
    CHECK (Affiliati > 0)  
);  
  
CREATE TABLE Produttore(  
    CodProduttore       CHAR(4),  
    PartitaIva           CHAR(11)    UNIQUE NOT NULL,  
    CodiceFiscale        CHAR(16)    UNIQUE NOT NULL,  
    Nome                 VARCHAR(20)  NOT NULL,  
    Cognome              VARCHAR(20)  NOT NULL,  
    Sesso                CHAR(1)      NOT NULL,  
    DataNascita          DATE         NOT NULL,  
    ComuneNascita        VARCHAR(20)  NOT NULL,  
    Residenza            VARCHAR(100) NOT NULL,  
    Domicilio           VARCHAR(100) DEFAULT NULL,  
    CantinaAffiliazione  CHAR(2)      DEFAULT NULL,  
    DataInizioAffiliazione DATE       DEFAULT NULL,  
  
    PRIMARY KEY (CodProduttore),  
    FOREIGN KEY (CantinaAffiliazione) REFERENCES Cantina(CodCantina),  
    CHECK (Sesso IN ('M', 'F')),  
    CHECK (DataNascita > TO_DATE('01/01/1900', 'dd/mm/yyyy')),  
    CHECK (Residenza <> Domicilio OR Domicilio IS NULL),  
    CHECK (DataInizioAffiliazione > TO_DATE('01/01/2010', 'dd/mm/yyyy'))  
);  
  
CREATE TABLE AffiliazionePassata(  
    CodProduttore  CHAR(4),  
    CodCantina     CHAR(2),  
    DataInizio     DATE,  
    DataFine       DATE    NOT NULL,  
  
    PRIMARY KEY (CodProduttore, CodCantina, DataInizio),  
    FOREIGN KEY (CodProduttore) REFERENCES Produttore(CodProduttore),  
    FOREIGN KEY (CodCantina)     REFERENCES Cantina(CodCantina),  
    CHECK (DataInizio > TO_DATE('01/01/2010', 'dd/mm/yyyy')),  
    CHECK (DataInizio < DataFine)  
);  
  
CREATE TABLE Vigneto(  
    CodVigneto      CHAR(5),  
    Estensione      INTEGER    NOT NULL,  
    Posizione       VARCHAR(100) NOT NULL,
```

```

AnnoImpianto          NUMERIC(4,0)   NOT NULL,
AnnoInizioProduzione  NUMERIC(4,0)   NOT NULL,
Vitigno                VARCHAR(15)    NOT NULL,
CodProduttore         CHAR(4)        NOT NULL,

PRIMARY KEY (CodVigneto),
FOREIGN KEY (CodProduttore) REFERENCES Produttore(CodProduttore),
CHECK (Estensione > 0),
CHECK (AnnoImpianto > 1900),
CHECK (AnnoInizioProduzione > 1900),
CHECK (AnnoImpianto <= AnnoInizioProduzione)
);

CREATE TABLE Uva(
    CodUva              CHAR(3),
    Nome                VARCHAR(20)    NOT NULL,
    Vitigno              VARCHAR(20)    NOT NULL,
    Colore               VARCHAR(6)     NOT NULL,
    Denominazione        VARCHAR(15)    NOT NULL,
    ResaPerEttaroMax     INTEGER        DEFAULT NULL,
    GradoZuccherinoMin   NUMERIC(3,1)   DEFAULT NULL,

    PRIMARY KEY (CodUva),
    CHECK (Colore IN ('Bianco', 'Rosso')),
    CHECK (ResaPerEttaroMax > 0 OR ResaPerEttaroMax IS NULL),
    CHECK (GradoZuccherinoMin > 0 OR GradoZuccherinoMin IS NULL)
);

CREATE TABLE SceltaVendemmiale(
    CodUva              CHAR(3)   NOT NULL,
    CodVigneto          CHAR(5),
    Anno                NUMERIC(4,0),

    PRIMARY KEY (CodVigneto, Anno),
    FOREIGN KEY (CodUva) REFERENCES Uva(CodUva),
    FOREIGN KEY (CodVigneto) REFERENCES Vigneto(CodVigneto),
    CHECK (Anno >= 2010)
);

CREATE TABLE Prenotazione(
    CodPrenotazione     CHAR(7),
    DataConferimento    DATE      NOT NULL,
    Quantita             INTEGER    NOT NULL,
    Colore               VARCHAR(6) NOT NULL,
    ModalitaRaccolta     VARCHAR(9) NOT NULL,
    DataAccordo          DATE      NOT NULL,
    CodProduttore        CHAR(4)   NOT NULL,

    PRIMARY KEY (CodPrenotazione),
    FOREIGN KEY (CodProduttore) REFERENCES Produttore(CodProduttore),
    CHECK (Quantita > 0),
    CHECK (Colore IN ('Bianco', 'Rosso')),
    CHECK (DataAccordo > TO_DATE('01/01/2010', 'dd/mm/yyyy')),
    CHECK (DataConferimento > DataAccordo+3),
    CHECK (ModalitaRaccolta IN ('Manuale', 'Meccanica'))
);

```

```

CREATE TABLE Carico(
    Prenotazione      CHAR(7),
    DataEOra          DATE,
    Uva                CHAR(3)          NOT NULL,
    Quantita           INTEGER          NOT NULL,
    PH                 NUMERIC(4,2)     NOT NULL,
    GradoZuccherino    NUMERIC(3,1)     NOT NULL,
    IntegritaAcini     VARCHAR(5)       NOT NULL,
    MaterialeEstraneo  NUMERIC(1,0)     NOT NULL,

    PRIMARY KEY (Prenotazione, DataEOra),
    FOREIGN KEY (Prenotazione) REFERENCES Prenotazione(CodPrenotazione),
    FOREIGN KEY (Uva)           REFERENCES Uva(CodUva),
    CHECK (Quantita > 0),
    CHECK (DataEOra > TO_DATE('01/01/2010', 'dd/mm/yyyy')),
    CHECK (PH >= 0 AND PH <= 14),
    CHECK (GradoZuccherino >= 0),
    CHECK (IntegritaAcini IN ('True', 'False')),
    CHECK (MaterialeEstraneo >= 1 AND MaterialeEstraneo <=5)
);

CREATE TABLE ScartoAccumulato(
    Cantina            CHAR(2),
    TipoScarto         VARCHAR(14),
    Uva                CHAR(3),
    Data               DATE,
    Quantita           INTEGER          NOT NULL,
    UnitaDiMisura       VARCHAR(2)      NOT NULL,

    PRIMARY KEY (Cantina, TipoScarto, Uva, Data),
    FOREIGN KEY (Cantina) REFERENCES Cantina(CodCantina),
    FOREIGN KEY (Uva)       REFERENCES Uva(CodUva),
    CHECK (TipoScarto IN ('Feccia Liquida', 'Raspi', 'Vinaccia', 'Feccia Solida')),
    CHECK (Data > TO_DATE('01/01/2010', 'dd/mm/yyyy')),
    CHECK (Quantita > 0),
    CHECK (UnitaDiMisura IN ('kg', 'l'))
);

CREATE TABLE Vasca(
    CodVasca           NUMERIC(3,0),
    Cantina             CHARACTER(2),
    Tipologia           VARCHAR(40)     NOT NULL,
    Capienza            INTEGER         NOT NULL,

    PRIMARY KEY (CodVasca, Cantina),
    FOREIGN KEY (Cantina) REFERENCES Cantina(CodCantina),
    CHECK (Capienza > 0)
);

CREATE TABLE Lotto(
    CodLotto           CHAR(10),
    Annata              NUMERIC(4,0)    NOT NULL,
    Denominazione       VARCHAR(15)     NOT NULL,
    Famiglia            VARCHAR(20)     NOT NULL,
    Colore              VARCHAR(6)      NOT NULL,
    AttoAlTaglio        VARCHAR(5)     NOT NULL,
    DataOrigine         DATE            NOT NULL,

```

```

        Descrizione      VARCHAR(200)  DEFAULT NULL,

        PRIMARY KEY (CodLotto),
        CHECK (Annata >= 2010),
        CHECK (AttoAlTaglio IN ('True', 'False')),
        CHECK (Colore IN ('Rosa', 'Rosso', 'Bianco')),
        CHECK (DataOrigine >= TO_DATE('01/01/2010', 'dd/mm/yyyy'))
    );

CREATE TABLE Stoccaggio(
    Lotto          CHAR(10),
    Vasca          NUMERIC(3,0),
    Cantina        CHAR(2),
    DataEOraInizio DATE,
    DataEOraFine   DATE          DEFAULT NULL,
    Quantita       INT           NOT NULL,

    PRIMARY KEY (Lotto, Vasca, Cantina, DataEOraInizio),
    FOREIGN KEY (Lotto) REFERENCES Lotto(CodLotto),
    FOREIGN KEY (Vasca, Cantina) REFERENCES Vasca(CodVasca, Cantina),
    CHECK (DataEOraInizio > TO_DATE('01/01/2010 01:00:00', 'dd/mm/yyyy hh:mi:ss')),
    CHECK (DataEOraInizio < DataEOraFine OR DataEOraFine IS NULL),
    CHECK (Quantita > 0)
);

CREATE TABLE Trasferimento(
    VascaPartenza    NUMERIC(3,0),
    VascaDestinazione NUMERIC(3,0),
    CantinaPartenza  CHAR(2),
    CantinaDestinazione CHAR(2),
    Lotto           CHAR(10),
    Data            DATE,
    Quantita        INTEGER      NOT NULL,
    Motivazione     VARCHAR(100) NOT NULL,

    PRIMARY KEY (VascaPartenza, VascaDestinazione, CantinaPartenza,
        CantinaDestinazione, Lotto, Data),
    FOREIGN KEY (VascaPartenza, CantinaPartenza) REFERENCES Vasca(CodVasca, Cantina),
    FOREIGN KEY (VascaDestinazione, CantinaDestinazione)
        REFERENCES Vasca(CodVasca, Cantina),
    CHECK (Data > TO_DATE('01/01/2010', 'dd/mm/yyyy')),
    CHECK (Quantita > 0)
);

CREATE TABLE Processo(
    NomeProcesso    VARCHAR(50),
    Descrizione     VARCHAR(50) NOT NULL,
    DurataTipica    VARCHAR(30) DEFAULT NULL,

    PRIMARY KEY (NomeProcesso)
);

CREATE TABLE Gestione(
    Cantina    CHAR(2),
    Processo   VARCHAR(50),

```



```

PRIMARY KEY (Cantina, Processo),
FOREIGN KEY (Cantina) REFERENCES Cantina(CodCantina),
FOREIGN KEY (Processo) REFERENCES Processo(NomeProcesso)
);

CREATE TABLE Lavorazione(
    Lotto CHAR(10),
    Processo VARCHAR(50),
    QuantitaIniziale INTEGER NOT NULL,
    QuantitaFinale INTEGER NOT NULL,
    Perdita INTEGER NOT NULL,
    DataInizio DATE NOT NULL,
    DataFine DATE DEFAULT NULL,

    PRIMARY KEY (Lotto, Processo),
    FOREIGN KEY (Lotto) REFERENCES Lotto(CodLotto),
    FOREIGN KEY (Processo) REFERENCES Processo(NomeProcesso),
    CHECK (QuantitaIniziale > 0),
    CHECK (QuantitaFinale > 0),
    CHECK (Perdita >= 0),
    CHECK (QuantitaFinale <= QuantitaIniziale + Perdita),
    CHECK (DataInizio >= TO_DATE('01/01/2010', 'dd/mm/yyyy')),
    CHECK (DataFine IS NULL OR DataFine > DataInizio)
);

CREATE TABLE Prodotto(
    CodProdotto CHAR(6),
    Nome VARCHAR(15) NOT NULL,
    Categoria VARCHAR(20) NOT NULL,
    Descrizione VARCHAR(250) NOT NULL,
    ConsigliUtilizzo VARCHAR(100) DEFAULT NULL,

    PRIMARY KEY (CodProdotto)
);

CREATE TABLE Utilizzo(
    Prodotto CHAR(6),
    Processo VARCHAR(20),
    Quantita INTEGER NOT NULL,
    UnitàDiMisura VARCHAR(10) NOT NULL,

    PRIMARY KEY (Prodotto, Processo),
    FOREIGN KEY (Prodotto) REFERENCES Prodotto(CodProdotto),
    FOREIGN KEY (Processo) REFERENCES Processo(NomeProcesso),
    CHECK (Quantita > 0),
    CHECK (UnitàDiMisura like '%/hl')
);

CREATE TABLE Vino(
    CodVino CHAR(4),
    Colore VARCHAR(6) NOT NULL,
    Famiglia VARCHAR(20) NOT NULL,
    Denominazione VARCHAR(15) NOT NULL,
    Tipo VARCHAR(20) NOT NULL,

    PRIMARY KEY (CodVino),
    CHECK (Colore IN ('Bianco', 'Rosso', 'Rosa'))

```

```

);

CREATE TABLE Finalita(
    Vino    CHAR(4),
    Lotto   CHAR(10),

    PRIMARY KEY (Vino, Lotto),
    FOREIGN KEY (Vino) REFERENCES Vino(CodVino),
    FOREIGN KEY (Lotto) REFERENCES Lotto(CodLotto)
);

CREATE TABLE Vinificazione(
    TipoVinificazione VARCHAR(30),
    Descrizione        VARCHAR(150) NOT NULL,

    PRIMARY KEY (TipoVinificazione)
);

CREATE TABLE Fase(
    CodFase          CHAR(4),
    Vinificazione    VARCHAR(30),
    Nome             VARCHAR(30) NOT NULL,
    Descrizione      VARCHAR(100) NOT NULL,
    NumeroProgressivo SMALLINT NOT NULL,

    PRIMARY KEY (CodFase, Vinificazione),
    FOREIGN KEY (Vinificazione) REFERENCES Vinificazione(TipoVinificazione),
    CHECK (NumeroProgressivo > 0)
);

CREATE TABLE Stato(
    Fase            CHAR(4),
    Vinificazione   VARCHAR(20),
    Lotto           CHAR(10),
    DataInizio      DATE,
    DataFine        DATE DEFAULT NULL,

    PRIMARY KEY (Fase, Vinificazione, Lotto, DataInizio),
    FOREIGN KEY (Fase, Vinificazione) REFERENCES Fase(CodFase, Vinificazione),
    FOREIGN KEY (Lotto) REFERENCES Lotto(CodLotto),
    CHECK (DataInizio > TO_DATE('01/01/2010', 'dd/mm/yyyy')),
    CHECK (DataFine IS NULL OR DataFine > DataInizio)
);

CREATE TABLE Campione(
    CodCampione     CHAR(10),
    MatricolaOperatore CHAR(5) NOT NULL,
    DataEOraPrelievo DATE NOT NULL,
    DataEOraRisultati DATE NOT NULL,
    Descrizione     VARCHAR(100) DEFAULT NULL,
    Lotto           CHAR(10) NOT NULL,

    PRIMARY KEY (CodCampione),
    FOREIGN KEY (Lotto) REFERENCES Lotto(CodLotto),
    CHECK (DataEOraPrelievo > TO_DATE('01/01/2010', 'dd/mm/yyyy')),
    CHECK (DataEOraRisultati IS NULL OR DataEOraPrelievo < DataEOraRisultati)
);

```

```

CREATE TABLE Parametro(
    NomeParametro VARCHAR(30),
    Descrizione   VARCHAR(100) NOT NULL,

    PRIMARY KEY (NomeParametro)
);

CREATE TABLE Analisi(
    Campione          CHAR(10),
    Parametro         VARCHAR(30),
    StrumentoDiMisura VARCHAR(30) NOT NULL,
    Valore            NUMERIC(10,5) NOT NULL,
    UnitàDiMisura     VARCHAR(15) NOT NULL,

    PRIMARY KEY (Campione, Parametro),
    FOREIGN KEY (Campione) REFERENCES Campione(CodCampione),
    FOREIGN KEY (Parametro) REFERENCES Parametro(NomeParametro)
);

CREATE TABLE Utenti(
    Username   VARCHAR(20),
    Password   VARCHAR(10) NOT NULL,
    Ruolo      VARCHAR(20) NOT NULL,

    PRIMARY KEY (Username),
    CHECK(Ruolo IN ('Cantina', 'Amministrazione', 'Analisi', 'Segreteria'))
);

```

# Popolamento delle tabelle

## Popolamento tabella *Cantina*:

```
INSERT INTO Cantina (CodCantina, Nome, Affiliati, QuantitaConferimentoMax,
                    IndirizzoStabilimento, IndirizzoSedeSociale)
VALUES ('01', 'Cantina Vittorio Veneto', 1520, 5000000, 'Via Canton 1 TV',
        'Via Lombardia 5');

INSERT INTO Cantina (CodCantina, Nome, Affiliati, QuantitaConferimentoMax,
                    IndirizzoStabilimento, IndirizzoSedeSociale)
VALUES ('02', 'L'Antica Vite', 753, 2000000, 'Via Da Vinci 65 VI', 'Via Lombardia 5');

INSERT INTO Cantina (CodCantina, Nome, Affiliati, QuantitaConferimentoMax,
                    IndirizzoStabilimento, IndirizzoSedeSociale)
VALUES ('03', 'La Collina del Sole', 986, 3000000, 'Via Europa 5 VR', 'Via Lombardia 5');

INSERT INTO Cantina (CodCantina, Nome, Affiliati, QuantitaConferimentoMax,
                    IndirizzoStabilimento, IndirizzoSedeSociale)
VALUES ('04', 'Associazione Viticoltori di Rovigo', 920, 2000000, 'Via Da Vinci 102 RO',
        'Via Lombardia 5');

INSERT INTO Cantina (CodCantina, Nome, Affiliati, QuantitaConferimentoMax,
                    IndirizzoStabilimento, IndirizzoSedeSociale)
VALUES ('05', 'Cantina Dal Cin', 201, 500000, 'Via Veneto 12 VE', 'Via Lombardia 5');

INSERT INTO Cantina (CodCantina, Nome, Affiliati, QuantitaConferimentoMax,
                    IndirizzoStabilimento, IndirizzoSedeSociale)
VALUES ('06', 'Cantina Viticoltori di Padova', 875, 1500000, 'Via Pascoli 78 PD',
        'Via Lombardia 5');
```

## Popolamento tabella *Produttore*:

```
INSERT INTO Produttore (CodProduttore, PartitaIva, CodiceFiscale, Nome, Cognome, Sesso,
                        DataNascita, ComuneNascita, Residenza, CantinaAffiliazione,
                        DataInizioAffiliazione)
VALUES ('FR3D', '26482559785', 'DLCMTT60E12P095G', 'Matteo', 'Dal Cin', 'M',
        TO_DATE('12/05/1960', 'dd/mm/yyyy'), 'Cozzuolo', 'Conegliano', '01',
        TO_DATE('01/02/2010', 'dd/mm/yyyy'));

INSERT INTO Produttore (CodProduttore, PartitaIva, CodiceFiscale, Nome, Cognome, Sesso,
                        DataNascita, ComuneNascita, Residenza, CantinaAffiliazione,
                        DataInizioAffiliazione)
VALUES ('Gr5t', '8671563324', 'DNAFRN55C30L985T', 'Francesco', 'Dan', 'M',
        TO_DATE('30/03/1955', 'dd/mm/yyyy'), 'Tarzo', 'Tarzo', '01',
        TO_DATE('10/10/2012', 'dd/mm/yyyy'));

INSERT INTO Produttore (CodProduttore, PartitaIva, CodiceFiscale, Nome, Cognome, Sesso,
                        DataNascita, ComuneNascita, Residenza, CantinaAffiliazione,
                        DataInizioAffiliazione)
VALUES ('Ag8y', '58734092785', 'MRCBSG55C30L985T', 'Marco', 'Bisaglia', 'M',
        TO_DATE('30/03/1955', 'dd/mm/yyyy'), 'Oderzo', 'Vittorio Veneto', '01',
        TO_DATE('10/10/2015', 'dd/mm/yyyy'));
```

```

INSERT INTO Produttore (CodProduttore, PartitaIva, CodiceFiscale, Nome, Cognome, Sesso,
                        DataNascita, ComuneNascita, Residenza, CantinaAffiliazione,
                        DataInizioAffiliazione)
VALUES ('K19o', '09138462548', 'FDRSPN65C30L985T', 'Federica', 'Spina', 'M',
        TO_DATE('30/03/1965', 'dd/mm/yyyy'), 'Mestre', 'Vittorio Veneto', '01',
        TO_DATE('10/10/2017', 'dd/mm/yyyy'));

INSERT INTO Produttore (CodProduttore, PartitaIva, CodiceFiscale, Nome, Cognome, Sesso,
                        DataNascita, ComuneNascita, Residenza, CantinaAffiliazione,
                        DataInizioAffiliazione)
VALUES ('Yg45', '63451897525', 'PSSDNI82B24L938G', 'Diana', 'Possamai', 'F',
        TO_DATE('24/02/1982', 'dd/mm/yyyy'), 'Pieve Di Soligo', 'Carpesica', '01',
        TO_DATE('03/04/2011', 'dd/mm/yyyy'));

INSERT INTO Produttore (CodProduttore, PartitaIva, CodiceFiscale, Nome, Cognome, Sesso,
                        DataNascita, ComuneNascita, Residenza, CantinaAffiliazione,
                        DataInizioAffiliazione)
VALUES ('Ki90', '33125778966', 'GRNTNO22L06P945F', 'Toni', 'Grinton', 'M',
        TO_DATE('06/12/1922', 'dd/mm/yyyy'), 'Carpesica', 'Vittorio Veneto', '02',
        TO_DATE('06/01/2010', 'dd/mm/yyyy'));

INSERT INTO Produttore (CodProduttore, PartitaIva, CodiceFiscale, Nome, Cognome, Sesso,
                        DataNascita, ComuneNascita, Residenza, CantinaAffiliazione,
                        DataInizioAffiliazione)
VALUES ('Oe12', '31245698722', 'RSSGSP68H31M152B', 'Giuseppe', 'Rossi', 'M',
        TO_DATE('31/08/1968', 'dd/mm/yyyy'), 'Trento', 'Bassano Del Grappa', '02',
        TO_DATE('03/10/2015', 'dd/mm/yyyy'));

INSERT INTO Produttore (CodProduttore, PartitaIva, CodiceFiscale, Nome, Cognome, Sesso,
                        DataNascita, ComuneNascita, Residenza, CantinaAffiliazione,
                        DataInizioAffiliazione)
VALUES ('AS93', '20164530075', 'RSSMRA68J25V682I', 'Mario', 'Rossi', 'M',
        TO_DATE('25/09/1968', 'dd/mm/yyyy'), 'Verona', 'Verona', '03',
        TO_DATE('06/06/2018', 'dd/mm/yyyy'));

INSERT INTO Produttore (CodProduttore, PartitaIva, CodiceFiscale, Nome, Cognome, Sesso,
                        DataNascita, ComuneNascita, Residenza, Domicilio,
                        CantinaAffiliazione, DataInizioAffiliazione)
VALUES ('LKMB', '18254513958', 'MTTBVL130B10K65F', 'Matteo', 'Bevilacqua', 'M',
        TO_DATE('10/03/1975', 'dd/mm/yyyy'), 'Tarzo', 'Tarzo', NULL, '01',
        TO_DATE('10/02/2013', 'dd/mm/yyyy'));

INSERT INTO Produttore (CodProduttore, PartitaIva, CodiceFiscale, Nome, Cognome, Sesso,
                        DataNascita, ComuneNascita, Residenza, Domicilio)
VALUES ('LK8I', '03147244961', 'MRADLL60A11S612F', 'Maria', 'Della Fune', 'F',
        TO_DATE('11/01/1960', 'dd/mm/yyyy'), 'Catanzaro', 'Catanzaro', 'Vittorio Veneto');

INSERT INTO Produttore (CodProduttore, PartitaIva, CodiceFiscale, Nome, Cognome, Sesso,
                        DataNascita, ComuneNascita, Residenza)
VALUES ('PO9S', '10327845352', 'LCADLL47J06S341F', 'Luca', 'Della Vedova', 'M',
        TO_DATE('06/08/1947', 'dd/mm/yyyy'), 'Bologna', 'Bologna');

```

#### Popolamento tabella *AffiliazionePassata*:

```
INSERT INTO AffiliazionePassata (CodProduttore, CodCantina, DataInizio, DataFine)
VALUES ('LK8I', '01', TO_DATE('02/05/2012', 'dd/mm/yyyy'), TO_DATE('14/08/2019',
'dd/mm/yyyy'));
```

```
INSERT INTO AffiliazionePassata (CodProduttore, CodCantina, DataInizio, DataFine)
VALUES ('PO9S', '02', TO_DATE('05/02/2010', 'dd/mm/yyyy'), TO_DATE('07/11/2018',
'dd/mm/yyyy'));
```

#### Popolamento tabella *Vigneto*:

```
INSERT INTO Vigneto (CodVigneto, Estensione, Posizione, AnnoImpianto,
AnnoInizioProduzione, Vitigno, CodProduttore)
VALUES ('AXGHE', 1000, '15.31254 65.123456', 1945, 1946, 'Chardonnay', 'FR3D');
```

```
INSERT INTO Vigneto (CodVigneto, Estensione, Posizione, AnnoImpianto,
AnnoInizioProduzione, Vitigno, CodProduttore)
VALUES ('BGFD', 3000, '31.1456 87.1345156', 1956, 1958, 'Barbera', 'FR3D');
```

```
INSERT INTO Vigneto (CodVigneto, Estensione, Posizione, AnnoImpianto,
AnnoInizioProduzione, Vitigno, CodProduttore)
VALUES ('PLIC', 4000, '45.235489 12.34568', 1964, 1964, 'Pinot Nero', 'AS93');
```

```
INSERT INTO Vigneto (CodVigneto, Estensione, Posizione, AnnoImpianto,
AnnoInizioProduzione, Vitigno, CodProduttore)
VALUES ('VCTG', 2000, '0.124584 786.12347', 1971, 1971, 'Merlot', 'LK8I');
```

```
INSERT INTO Vigneto (CodVigneto, Estensione, Posizione, AnnoImpianto,
AnnoInizioProduzione, Vitigno, CodProduttore)
VALUES ('SDWL', 5000, '45.15648 56.4894173', 1981, 1990, 'Glera', 'Oe12');
```

```
INSERT INTO Vigneto (CodVigneto, Estensione, Posizione, AnnoImpianto,
AnnoInizioProduzione, Vitigno, CodProduttore)
VALUES ('EVSD', 7000, '78.16489 99.123417875', 1999, 1999, 'Glera', 'AS93');
```

```
INSERT INTO Vigneto (CodVigneto, Estensione, Posizione, AnnoImpianto,
AnnoInizioProduzione, Vitigno, CodProduttore)
VALUES ('AMJCG', 5, '20.13245185 52.02121025', 1980, 1981, 'Glera', 'Ki90');
```

```
INSERT INTO Vigneto (CodVigneto, Estensione, Posizione, AnnoImpianto,
AnnoInizioProduzione, Vitigno, CodProduttore)
VALUES ('HBGKK', 3, '21.18245032 50.02023365', 1980, 1982, 'Glera', 'Ki90');
```

```
INSERT INTO Vigneto (CodVigneto, Estensione, Posizione, AnnoImpianto,
AnnoInizioProduzione, Vitigno, CodProduttore)
VALUES ('HLMJS', 7, '19.10245185 49.02120020', 1979, 1980, 'Glera', 'Ki90');
```

#### Popolamento tabella *Uva*:

```
INSERT INTO Uva (CodUva, Nome, Vitigno, Colore, Denominazione, ResaPerEttaroMax,
GradoZuccherinoMin)
VALUES ('284', 'Prosecco', 'Glera', 'Bianco', 'DOC', 180, 0.1);
```

```

INSERT INTO Uva (CodUva, Nome, Vitigno, Colore, Denominazione, ResaPerEttaroMax,
                GradoZuccherinoMin)
VALUES ('311', 'Prosecco', 'Glera', 'Bianco', 'DOCG', 130, 0.2);

INSERT INTO Uva (CodUva, Nome, Vitigno, Colore, Denominazione, ResaPerEttaroMax,
                GradoZuccherinoMin)
VALUES ('145', 'Prosecco', 'Glera', 'Bianco', 'DOCG Superiore', 120, 1.0);

```

#### Popolamento tabella *SceltaVendemmiale*:

```

INSERT INTO SceltaVendemmiale (CodUva, CodVigneto, Anno)
VALUES ('284', 'AXGHE', 2011);

INSERT INTO SceltaVendemmiale (CodUva, CodVigneto, Anno)
VALUES ('311', 'BGFD', 2012);

INSERT INTO SceltaVendemmiale (CodUva, CodVigneto, Anno)
VALUES ('145', 'PLIC', 2012);

INSERT INTO SceltaVendemmiale (CodUva, CodVigneto, Anno)
VALUES ('145', 'PLIC', 2013);

INSERT INTO SceltaVendemmiale (CodUva, CodVigneto, Anno)
VALUES ('311', 'AMJCG', 2011);

INSERT INTO SceltaVendemmiale (CodUva, CodVigneto, Anno)
VALUES ('311', 'AMJCG', 2012);

INSERT INTO SceltaVendemmiale (CodUva, CodVigneto, Anno)
VALUES ('311', 'AMJCG', 2013);

```

#### Popolamento tabella *Prenotazione*:

```

INSERT INTO Prenotazione (CodPrenotazione, DataConferimento, Quantita, Colore,
                          ModalitaRaccolta, DataAccordo, CodProduttore)
VALUES ('LGJEJK3', TO_DATE('03/09/2011', 'dd/mm/yyyy'), 357, 'Bianco', 'Manuale',
        TO_DATE('26/08/2011', 'dd/mm/yyyy'), 'FR3D');

INSERT INTO Prenotazione (CodPrenotazione, DataConferimento, Quantita, Colore,
                          ModalitaRaccolta, DataAccordo, CodProduttore)
VALUES ('MGJEJK3', TO_DATE('03/09/2011', 'dd/mm/yyyy'), 757, 'Bianco', 'Manuale',
        TO_DATE('26/08/2011', 'dd/mm/yyyy'), 'AS93');

INSERT INTO Prenotazione (CodPrenotazione, DataConferimento, Quantita, Colore,
                          ModalitaRaccolta, DataAccordo, CodProduttore)
VALUES ('FMCDF45', TO_DATE('20/10/2013', 'dd/mm/yyyy'), 150, 'Rosso', 'Manuale',
        TO_DATE('01/01/2013', 'dd/mm/yyyy'), 'LK8I');

INSERT INTO Prenotazione (CodPrenotazione, DataConferimento, Quantita, Colore,
                          ModalitaRaccolta, DataAccordo, CodProduttore)
VALUES ('HJD894D', TO_DATE('02/10/2014', 'dd/mm/yyyy'), 200, 'Bianco', 'Meccanica',
        TO_DATE('20/09/2014', 'dd/mm/yyyy'), 'LKMB');

```

```

INSERT INTO Prenotazione (CodPrenotazione, DataConferimento, Quantita, Colore,
                          ModalitaRaccolta, DataAccordo, CodProduttore)
VALUES ('HCLDU78', TO_DATE('01/11/2014', 'dd/mm/yyyy'), 210, 'Bianco', 'Meccanica',
       TO_DATE('18/10/2014', 'dd/mm/yyyy'), 'LKMB');

INSERT INTO Prenotazione (CodPrenotazione, DataConferimento, Quantita, Colore,
                          ModalitaRaccolta, DataAccordo, CodProduttore)
VALUES ('LLMD456', TO_DATE('10/12/2014', 'dd/mm/yyyy'), 200, 'Bianco', 'Meccanica',
       TO_DATE('20/11/2014', 'dd/mm/yyyy'), 'LKMB');

INSERT INTO Prenotazione (CodPrenotazione, DataConferimento, Quantita, Colore,
                          ModalitaRaccolta, DataAccordo, CodProduttore)
VALUES ('KMGBHE6', TO_DATE('25/10/2014', 'dd/mm/yyyy'), 300, 'Rosso', 'Manuale',
       TO_DATE('20/10/2014', 'dd/mm/yyyy'), 'FR3D');

INSERT INTO Prenotazione (CodPrenotazione, DataConferimento, Quantita, Colore,
                          ModalitaRaccolta, DataAccordo, CodProduttore)
VALUES ('KMGBHA7', TO_DATE('25/11/2014', 'dd/mm/yyyy'), 350, 'Rosso', 'Manuale',
       TO_DATE('20/11/2014', 'dd/mm/yyyy'), 'FR3D');

INSERT INTO Prenotazione (CodPrenotazione, DataConferimento, Quantita, Colore,
                          ModalitaRaccolta, DataAccordo, CodProduttore)
VALUES ('AFCDE46', TO_DATE('04/11/2019', 'dd/mm/yyyy'), 150, 'Rosso', 'Manuale',
       TO_DATE('01/10/2019', 'dd/mm/yyyy'), 'LK8I');

```

#### Popolamento tabella *Carico*:

```

INSERT INTO Carico (Prenotazione, DataEOra, Uva, Quantita, PH, GradoZuccherino,
                   IntegritaAcini, MaterialeEstraneo)
VALUES ('LGJEJK3', TO_DATE('26/08/2011 13:12:45', 'dd/mm/yyyy hh24:mi:ss'), '284', 78,
       3.2, 4.0, 'True', 3);

INSERT INTO Carico (Prenotazione, DataEOra, Uva, Quantita, PH, GradoZuccherino,
                   IntegritaAcini, MaterialeEstraneo)
VALUES ('MGJEJK3', TO_DATE('26/08/2011 15:02:37', 'dd/mm/yyyy hh24:mi:ss'), '311', 100,
       3.5, 4.0, 'False', 5);

INSERT INTO Carico (Prenotazione, DataEOra, Uva, Quantita, PH, GradoZuccherino,
                   IntegritaAcini, MaterialeEstraneo)
VALUES ('FMCDF45', TO_DATE('01/01/2013 10:20:33', 'dd/mm/yyyy hh24:mi:ss'), '145', 75,
       3.2, 3.9, 'True', 1);

INSERT INTO Carico (Prenotazione, DataEOra, Uva, Quantita, PH, GradoZuccherino,
                   IntegritaAcini, MaterialeEstraneo)
VALUES ('AFCDE46', TO_DATE('04/11/2019 10:20:33', 'dd/mm/yyyy hh24:mi:ss'), '284', 90,
       3.2, 3.9, 'True', 1);

INSERT INTO Carico (Prenotazione, DataEOra, Uva, Quantita, PH, GradoZuccherino,
                   IntegritaAcini, MaterialeEstraneo)
VALUES ('AFCDE46', TO_DATE('04/11/2019 14:20:33', 'dd/mm/yyyy hh24:mi:ss'), '311', 214,
       3.2, 3.9, 'True', 1);

INSERT INTO Carico (Prenotazione, DataEOra, Uva, Quantita, PH, GradoZuccherino,
                   IntegritaAcini, MaterialeEstraneo)
VALUES ('AFCDE46', TO_DATE('04/11/2019 09:20:33', 'dd/mm/yyyy hh24:mi:ss'), '145', 148,
       3.2, 3.9, 'True', 1);

```



#### Popolamento tabella *ScartoAccumulato*:

```
INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('01', 'Feccia Liquida', '284', TO_DATE('10/10/2012', 'dd/mm/yyyy'), 20, 'l');

INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('02', 'Feccia Solida', '311', TO_DATE('10/11/2013', 'dd/mm/yyyy'), 30, 'kg');

INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('03', 'Raspi', '284', TO_DATE('12/11/2014', 'dd/mm/yyyy'), 10, 'kg');

INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('01', 'Vinaccia', '145', TO_DATE('01/12/2015', 'dd/mm/yyyy'), 15, 'l');

INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('01', 'Raspi', '284', TO_DATE('02/10/2018', 'dd/mm/yyyy'), 52, 'kg');

INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('01', 'Feccia Solida', '145', TO_DATE('01/02/2012', 'dd/mm/yyyy'), 200, 'kg');

INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('01', 'Feccia Solida', '311', TO_DATE('02/05/2012', 'dd/mm/yyyy'), 300, 'kg');

INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('02', 'Raspi', '284', TO_DATE('02/02/2018', 'dd/mm/yyyy'), 100, 'kg');

INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('02', 'Raspi', '284', TO_DATE('03/08/2018', 'dd/mm/yyyy'), 200, 'kg');

INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('02', 'Raspi', '311', TO_DATE('13/10/2018', 'dd/mm/yyyy'), 200, 'kg');

INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('03', 'Raspi', '284', TO_DATE('23/05/2018', 'dd/mm/yyyy'), 230, 'kg');

INSERT INTO ScartoAccumulato (Cantina, TipoScarto, Uva, Data, Quantita, UnitaDiMisura)
VALUES ('03', 'Raspi', '311', TO_DATE('23/05/2018', 'dd/mm/yyyy'), 430, 'kg');
```

#### Popolamento tabella *Vasca*:

```
INSERT INTO Vasca (CodVasca, Cantina, Tipologia, Capienza)
VALUES (1, '01', 'Acciaio', 100);

INSERT INTO Vasca (CodVasca, Cantina, Tipologia, Capienza)
VALUES (2, '01', 'Legno', 1000);

INSERT INTO Vasca (CodVasca, Cantina, Tipologia, Capienza)
VALUES (50, '02', 'Acciaio', 200);

INSERT INTO Vasca (CodVasca, Cantina, Tipologia, Capienza)
VALUES (53, '02', 'Cemento', 500);

INSERT INTO Vasca (CodVasca, Cantina, Tipologia, Capienza)
```

```
VALUES (24, '02', 'Legno', 100);
```

```
INSERT INTO Vasca (CodVasca, Cantina, Tipologia, Capienza)
```

```
VALUES (10, '03', 'Acciaio', 400);
```

Popolamento tabella *Lotto*:

```
INSERT INTO Lotto(CodLotto, Annata, Denominazione, Famiglia, Colore, AttoAlTaglio,  
DataOrigine)
```

```
VALUES ('KDJNH78463', 2011, 'DOC', 'Prosecco', 'Bianco', 'True',  
TO_DATE('01/10/2011', 'dd/mm/yyyy'));
```

```
INSERT INTO Lotto(CodLotto, Annata, Denominazione, Famiglia, Colore, AttoAlTaglio,  
DataOrigine)
```

```
VALUES ('RDGHH75471', 2011, 'DOC', 'Prosecco', 'Bianco', 'False',  
TO_DATE('02/10/2011', 'dd/mm/yyyy'));
```

```
INSERT INTO Lotto(CodLotto, Annata, Denominazione, Famiglia, Colore, AttoAlTaglio,  
DataOrigine)
```

```
VALUES ('WQJNH65410', 2011, 'DOC', 'Prosecco', 'Bianco', 'False',  
TO_DATE('03/10/2011', 'dd/mm/yyyy'));
```

```
INSERT INTO Lotto(CodLotto, Annata, Denominazione, Famiglia, Colore, AttoAlTaglio,  
DataOrigine)
```

```
VALUES ('LKCSR16425', 2012, 'VDT', 'Chardonnay', 'Bianco', 'False',  
TO_DATE('02/11/2012', 'dd/mm/yyyy'));
```

```
INSERT INTO Lotto(CodLotto, Annata, Denominazione, Famiglia, Colore, AttoAlTaglio,  
DataOrigine)
```

```
VALUES ('MHNCD45715', 2013, 'IGT', 'Merlot', 'Rosso', 'False',  
TO_DATE('10/10/2013', 'dd/mm/yyyy'));
```

```
INSERT INTO Lotto(CodLotto, Annata, Denominazione, Famiglia, Colore, AttoAlTaglio,  
DataOrigine)
```

```
VALUES ('CSQWW12415', 2014, 'DOCG', 'Merlot', 'Rosso', 'True',  
TO_DATE('11/11/2014', 'dd/mm/yyyy'));
```

```
INSERT INTO Lotto(CodLotto, Annata, Denominazione, Famiglia, Colore, AttoAlTaglio,  
DataOrigine)
```

```
VALUES ('MGVRE44682', 2015, 'DOCG Superiore', 'Prosecco', 'Bianco', 'False',  
TO_DATE('02/09/2015', 'dd/mm/yyyy'));
```

Popolamento tabella *Stoccaggio*:

```
INSERT INTO Stoccaggio (Lotto, Vasca, Cantina, DataEOraInizio, DataEOraFine, Quantita)
```

```
VALUES ('KDJNH78463', 1, '01', TO_DATE('01/10/2011 13:35:24', 'dd/mm/yyyy hh24:mi:ss'),  
TO_DATE('11/10/2011 10:12:00', 'dd/mm/yyyy hh24:mi:ss'), 90);
```

```
INSERT INTO Stoccaggio (Lotto, Vasca, Cantina, DataEOraInizio, DataEOraFine, Quantita)
```

```
VALUES ('LKCSR16425', 2, '01', TO_DATE('02/11/2012 12:20:20', 'dd/mm/yyyy hh24:mi:ss'),  
TO_DATE('02/11/2012 21:10:30', 'dd/mm/yyyy hh24:mi:ss'), 70);
```

```
INSERT INTO Stoccaggio (Lotto, Vasca, Cantina, DataEOraInizio, DataEOraFine, Quantita)
```

```
VALUES ('MHNCD45715', 50, '02', TO_DATE('10/10/2013 14:10:00', 'dd/mm/yyyy hh24:mi:ss'),  
TO_DATE('12/10/2013 14:18:30', 'dd/mm/yyyy hh24:mi:ss'), 80);
```

```
INSERT INTO Stoccaggio (Lotto, Vasca, Cantina, DataEOraInizio, DataEOraFine, Quantita)
```

```
VALUES ('CSQWW12415', 53, '02', TO_DATE('11/11/2014 16:26:25', 'dd/mm/yyyy hh24:mi:ss'),
```

```
TO_DATE('12/11/2014 10:10:00', 'dd/mm/yyyy, hh24:mi:ss'), 100);
```

#### Popolamento tabella *Trasferimento*:

```
INSERT INTO Trasferimento (VascaPartenza, VascaDestinazione, CantinaPartenza,
                           CantinaDestinazione, Lotto, Data, Quantita, Motivazione)
VALUES (1, 50, '01', '02', 'KDJNH78463', TO_DATE('11/10/2011 10:13:56',
        'dd/mm/yyyy hh24:mi:ss'), 30, 'Tutte le vasche erano occupate');

INSERT INTO Trasferimento (VascaPartenza, VascaDestinazione, CantinaPartenza,
                           CantinaDestinazione, Lotto, Data, Quantita, Motivazione)
VALUES (1, 50, '01', '02', 'RDGHH75471', TO_DATE('04/03/2014 10:13:56',
        'dd/mm/yyyy hh24:mi:ss'), 40, 'Vasche adeguate non disponibili');

INSERT INTO Trasferimento (VascaPartenza, VascaDestinazione, CantinaPartenza,
                           CantinaDestinazione, Lotto, Data, Quantita, Motivazione)
VALUES (53, 2, '02', '01', 'WQJNH65410', TO_DATE('06/06/2015 10:13:56',
        'dd/mm/yyyy hh24:mi:ss'), 65, 'Tutte le vasche erano occupate');

INSERT INTO Trasferimento (VascaPartenza, VascaDestinazione, CantinaPartenza,
                           CantinaDestinazione, Lotto, Data, Quantita, Motivazione)
VALUES (53, 2, '02', '01', 'LKCSR16425', TO_DATE('03/11/2012 10:11:46',
        'dd/mm/yyyy hh24:mi:ss'), 50, 'Vasche adeguate non disponibili');
```

#### Popolamento tabella *Processo*:

```
INSERT INTO Processo (NomeProcesso, Descrizione, DurataTipica)
VALUES ('Diraspatura', 'Separazioni degli acini dal raspo', NULL);

INSERT INTO Processo (NomeProcesso, Descrizione, DurataTipica)
VALUES ('Fermentazione', 'Gli zuccheri del mosto diventano alcol etilico', '10 gg ');

INSERT INTO Processo (NomeProcesso, Descrizione, DurataTipica)
VALUES ('Solfitazione', 'Funzione antisettica', NULL);

INSERT INTO Processo (NomeProcesso, Descrizione, DurataTipica)
VALUES ('Invecchiamento', 'Aumenta la qualità del vino', 'Variabile');
```

#### Popolamento tabella *Gestione*:

```
INSERT INTO Gestione (Cantina, Processo) VALUES ('01', 'Diraspatura');
INSERT INTO Gestione (Cantina, Processo) VALUES ('01', 'Fermentazione');
INSERT INTO Gestione (Cantina, Processo) VALUES ('01', 'Solfitazione');
INSERT INTO Gestione (Cantina, Processo) VALUES ('01', 'Invecchiamento');
INSERT INTO Gestione (Cantina, Processo) VALUES ('02', 'Diraspatura');
INSERT INTO Gestione (Cantina, Processo) VALUES ('02', 'Fermentazione');
INSERT INTO Gestione (Cantina, Processo) VALUES ('02', 'Solfitazione');
INSERT INTO Gestione (Cantina, Processo) VALUES ('02', 'Invecchiamento');
INSERT INTO Gestione (Cantina, Processo) VALUES ('03', 'Diraspatura');
INSERT INTO Gestione (Cantina, Processo) VALUES ('03', 'Fermentazione');
INSERT INTO Gestione (Cantina, Processo) VALUES ('03', 'Solfitazione');
INSERT INTO Gestione (Cantina, Processo) VALUES ('03', 'Invecchiamento');
INSERT INTO Gestione (Cantina, Processo) VALUES ('04', 'Diraspatura');
```

```

INSERT INTO Gestione (Cantina, Processo) VALUES ('04', 'Fermentazione');
INSERT INTO Gestione (Cantina, Processo) VALUES ('04', 'Solfitazione');
INSERT INTO Gestione (Cantina, Processo) VALUES ('04', 'Invecchiamento');

```

#### Popolamento tabella *Lavorazione*:

```

INSERT INTO Lavorazione (Lotto, Processo, QuantitaIniziale, QuantitaFinale, Perdita,
                        DataInizio, DataFine)
VALUES ('LKCSR16425', 'Fermentazione', 300, 290, 1, TO_DATE('02/10/2015', 'dd/mm/yyyy'),
        TO_DATE('15/10/2015', 'dd/mm/yyyy'));

INSERT INTO Lavorazione (Lotto, Processo, QuantitaIniziale, QuantitaFinale, Perdita,
                        DataInizio, DataFine)
VALUES ('LKCSR16425', 'Invecchiamento', 250, 240, 3, TO_DATE('10/11/2015', 'dd/mm/yyyy'),
        TO_DATE('15/04/2017', 'dd/mm/yyyy'));

INSERT INTO Lavorazione (Lotto, Processo, QuantitaIniziale, QuantitaFinale, Perdita,
                        DataInizio, DataFine)
VALUES ('MGVRE44682', 'Diraspatura', 300, 100, 200, TO_DATE('10/10/2012 20:02:20',
        'dd/mm/yyyy hh24:mi:ss'), TO_DATE('11/10/2012 10:22:22', 'dd/mm/yyyy hh24:mi:ss'));

```

#### Popolamento tabella *Prodotto*:

```

INSERT INTO Prodotto (CodProdotto, Nome, Categoria, Descrizione, ConsigliUtilizzo)
VALUES ('CFA784', 'Enoferm R', 'Lievito', 'Lievito per mosti d''uva rossa',
        'Per rapido avviamento della fermentazione dei mosti d''uva');

INSERT INTO Prodotto (CodProdotto, Nome, Categoria, Descrizione, ConsigliUtilizzo)
VALUES ('GER525', 'Acido Malico', 'Attivante', 'Regolatore del pH',
        'Conferisce un aumento di acidità in relazione alla quantità utilizzata');

INSERT INTO Prodotto (CodProdotto, Nome, Categoria, Descrizione, ConsigliUtilizzo)
VALUES ('BIR438', 'Polylact', 'Chiarificante', 'Chiarificante preventivo/curativo',
        'Aumentare la dose nel caso di uve con problemi sanitari');

```

#### Popolamento tabella *Utilizzo*:

```

INSERT INTO Utilizzo (Prodotto, Processo, Quantita, UnitàDiMisura)
VALUES ('CFA784', 'Fermentazione', 2000, 'g/hl');

INSERT INTO Utilizzo (Prodotto, Processo, Quantita, UnitàDiMisura)
VALUES ('CFA784', 'Invecchiamento', 10, 'g/hl');

INSERT INTO Utilizzo (Prodotto, Processo, Quantita, UnitàDiMisura)
VALUES ('CFA784', 'Solfitazione', 60, 'g/hl');

INSERT INTO Utilizzo (Prodotto, Processo, Quantita, UnitàDiMisura)
VALUES ('GER525', 'Fermentazione', 500, 'g/hl');

INSERT INTO Utilizzo (Prodotto, Processo, Quantita, UnitàDiMisura)
VALUES ('GER525', 'Solfitazione', 50, 'g/hl');

INSERT INTO Utilizzo (Prodotto, Processo, Quantita, UnitàDiMisura)
VALUES ('GER525', 'Invecchiamento', 20, 'g/hl');

INSERT INTO Utilizzo (Prodotto, Processo, Quantita, UnitàDiMisura)

```

```
VALUES ('BIR438', 'Fermentazione', 1000, 'g/hl');

INSERT INTO Utilizzo (Prodotto, Processo, Quantita, UnitaDiMisura)
VALUES ('BIR438', 'Invecchiamento', 5, 'g/hl');

INSERT INTO Utilizzo (Prodotto, Processo, Quantita, UnitaDiMisura)
VALUES ('BIR438', 'Solfitazione', 80, 'g/hl');
```

#### Popolamento tabella *Vino*:

```
INSERT INTO Vino (CodVino, Colore, Famiglia, Denominazione, Tipo)
VALUES ('LFCG', 'Bianco', 'Prosecco', 'DOC', 'Frizzante');

INSERT INTO Vino (CodVino, Colore, Famiglia, Denominazione, Tipo)
VALUES ('MKDS', 'Bianco', 'Prosecco', 'DOCG', 'Frizzante');

INSERT INTO Vino (CodVino, Colore, Famiglia, Denominazione, Tipo)
VALUES ('FGER', 'Bianco', 'Prosecco', 'DOCG Superiore', 'Frizzante');

INSERT INTO Vino (CodVino, Colore, Famiglia, Denominazione, Tipo)
VALUES ('DFES', 'Rosso', 'Merlot', 'IGT', 'Liscio');

INSERT INTO Vino (CodVino, Colore, Famiglia, Denominazione, Tipo)
VALUES ('FDES', 'Bianco', 'Chardonnay', 'DOC', 'Liscio');
```

#### Popolamento tabella *Finalita*:

```
INSERT INTO Finalita (Vino, Lotto) VALUES ('LFCG', 'KDJNH78463');
INSERT INTO Finalita (Vino, Lotto) VALUES ('FDES', 'LKCSR16425');
```

#### Popolamento tabella *Vinificazione*:

```
INSERT INTO Vinificazione (TipoVinificazione, Descrizione)
VALUES ('In bianco', 'La fermentazione Avviene senza macerazione. E' prevista la pigiatura e quasi sempre la diraspapigiatura');

INSERT INTO Vinificazione (TipoVinificazione, Descrizione)
VALUES ('In rosso', 'La fermentazione avviene con macerazione e avviene il contatto tra vinacce e mosto in fermentazione');

INSERT INTO Vinificazione (TipoVinificazione, Descrizione)
VALUES ('Termovinificazione', 'Si scalda parzialmente la vinaccia aggiungendo l'anidride solforosa. Si fermenta come fosse in bianco');

INSERT INTO Vinificazione (TipoVinificazione, Descrizione)
VALUES ('Macerazione Carbonica', 'Si fermenta l'intero grappolo e sono necessarie uve perfettamente sane');

INSERT INTO Vinificazione (TipoVinificazione, Descrizione)
VALUES ('Vinificazione in Riduzione', 'Utilizzo di gas antiossidanti per proteggere dall'ossigeno il mosto esaltando gli aromi tipici varietali');
```

### Popolamento tabella *Campione*:

```
INSERT INTO Campione (CodCampione, MatricolaOperatore, DataEOraPrelievo,
                      DataEOraRisultati, Descrizione, Lotto)
VALUES ('FGARC12456', 'LFMGS', TO_DATE('02/11/2010 16:02:11', 'dd/mm/yyyy hh24:mi:ss'),
        TO_DATE('02/12/2010 10:45:12', 'dd/mm/yyyy hh24:mi:ss'), 'Analisi Qualità',
        'KDJNH78463');
```

```
INSERT INTO Campione (CodCampione, MatricolaOperatore, DataEOraPrelievo,
                      DataEOraRisultati, Lotto)
VALUES ('ZAPRC82445', 'HYNUU', TO_DATE('22/11/2011 17:22:11', 'dd/mm/yyyy hh24:mi:ss'),
        TO_DATE('03/12/2011 11:55:59', 'dd/mm/yyyy hh24:mi:ss'), 'LKCSR16425');
```

```
INSERT INTO Campione (CodCampione, MatricolaOperatore, DataEOraPrelievo,
                      DataEOraRisultati, Lotto)
VALUES ('JBGYH12456', 'LMHJU', TO_DATE('02/11/2012 16:02:11', 'dd/mm/yyyy hh24:mi:ss'),
        TO_DATE('02/12/2012 08:10:22', 'dd/mm/yyyy hh24:mi:ss'), 'MHNCD45715');
```

```
INSERT INTO Campione (CodCampione, MatricolaOperatore, DataEOraPrelievo,
                      DataEOraRisultati, Descrizione, Lotto)
VALUES ('ACEFD98752', 'AMJCF', TO_DATE('01/10/2015 10:10:33', 'dd/mm/yyyy hh24:mi:ss'),
        TO_DATE('10/10/2015 15:03:12', 'dd/mm/yyyy hh24:mi:ss'), 'analisi grado alcolico',
        'LKCSR16425');
```

```
INSERT INTO Campione (CodCampione, MatricolaOperatore, DataEOraPrelievo,
                      DataEOraRisultati, Descrizione, Lotto)
VALUES ('RREFD98752', 'AMJCF', TO_DATE('02/11/2015 11:12:22', 'dd/mm/yyyy hh24:mi:ss'),
        TO_DATE('12/11/2015 17:22:52', 'dd/mm/yyyy hh24:mi:ss'), 'analisi grado alcolico',
        'MHNCD45715');
```

```
INSERT INTO Campione (CodCampione, MatricolaOperatore, DataEOraPrelievo,
                      DataEOraRisultati, Descrizione, Lotto)
VALUES ('CFADS10236', 'QWEWS', TO_DATE('12/09/2015 15:22:01', 'dd/mm/yyyy hh24:mi:ss'),
        TO_DATE('13/09/2015 18:22:01', 'dd/mm/yyyy hh24:mi:ss'), 'analisi glicerolo',
        'CSQWW12415');
```

```
INSERT INTO Campione (CodCampione, MatricolaOperatore, DataEOraPrelievo,
                      DataEOraRisultati, Descrizione, Lotto)
VALUES ('MLPOP12017', 'QWEWS', TO_DATE('10/08/2015, 14:22:11', 'dd/mm/yyyy hh24:mi:ss'),
        TO_DATE('15/08/2015', 'dd/mm/yyyy hh24:mi:ss'), 'analisi glicerolo straordinaria',
        'MGVRE44682');
```

### Popolamento tabella *Parametro*:

```
INSERT INTO Parametro (NomeParametro, Descrizione)
VALUES ('Acido Acetico', 'La quantità di acido acetico presente è indice dello stato di
        sanità dell''uva');
```

```
INSERT INTO Parametro (NomeParametro, Descrizione)
VALUES ('Grado Alcolico', 'Indice del grado alcolico del mosto');
```

```
INSERT INTO Parametro (NomeParametro, Descrizione)
VALUES ('Glicerolo', 'Deve essere dentro un range per dare un senso di rotondità al
        vino');
```

### Popolamento tabella *Analisi*:

```
INSERT INTO Analisi (Campione, Parametro, StrumentoDiMisura, Valore, UnitaDiMisura)
VALUES ('FGARC12456', 'Acido Acetico', 'lettore fotometrico', 4.2, 'grammi/litro');

INSERT INTO Analisi (Campione, Parametro, StrumentoDiMisura, Valore, UnitaDiMisura)
VALUES ('ZAPRC82445', 'Grado Alcolico', 'Alcolometro', 12, '%');

INSERT INTO Analisi (Campione, Parametro, StrumentoDiMisura, Valore, UnitaDiMisura)
VALUES ('JBGYH12456', 'Acido Acetico', 'lettore fotometrico', 4.5, 'grammi/litro');
INSERT INTO Analisi (Campione, Parametro, StrumentoDiMisura, Valore, UnitaDiMisura)
VALUES ('ACEFD98752', 'Grado Alcolico', 'lettore fotometrico', 4.0, 'grammi/litro');

INSERT INTO Analisi (Campione, Parametro, StrumentoDiMisura, Valore, UnitaDiMisura)
VALUES ('RREFD98752', 'Grado Alcolico', 'lettore fotometrico', 3.7, 'grammi/litro');

INSERT INTO Analisi (Campione, Parametro, StrumentoDiMisura, Valore, UnitaDiMisura)
VALUES ('CFADS10236', 'Glicerolo', 'misuratore glicerolo', 10, 'grammi/litro');

INSERT INTO Analisi (Campione, Parametro, StrumentoDiMisura, Valore, UnitaDiMisura)
VALUES ('MLPOP12017', 'Glicerolo', 'misuratore glicerolo', 11, 'grammi/litro');

INSERT INTO Analisi (Campione, Parametro, StrumentoDiMisura, Valore, UnitaDiMisura)
VALUES ('FGARC12456', 'Glicerolo', 'CDR WineLab', 12, 'grammi/litro');

INSERT INTO Analisi (Campione, Parametro, StrumentoDiMisura, Valore, UnitaDiMisura)
VALUES ('JBGYH12456', 'Glicerolo', 'CDR WineLab', 13, 'grammi/litro');

INSERT INTO Analisi (Campione, Parametro, StrumentoDiMisura, Valore, UnitaDiMisura)
VALUES ('ZAPRC82445', 'Glicerolo', 'CDR WineLab', 8, 'grammi/litro');
```

### Popolamento tabella *Fase*:

```
INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('AHJD', 'In bianco', 'Fermentazione Alcolica', 'Forma di metabolismo energetico
che permette la trasformazione del mosto in vino', 5);

INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('AHJD', 'In rosso', 'Fermentazione Alcolica', 'Forma di metabolismo energetico che
permette la trasformazione del mosto in vino', 5);

INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('BVRW', 'In bianco', 'Chiarifica', 'Si priva il vino di diverse sostanze, così da
portarlo a una maggiore limpidezza', 6);

INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('BRRW', 'In rosso', 'Chiarifica', 'Si priva il vino di diverse sostanze, così da
portarlo a una maggiore limpidezza', 6);

INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('FDGT', 'Vinificazione in Riduzione', 'Ammostamento', 'Schiacciare l'uva per ot-
tenere il mosto attraverso la pigiadiraspatura', 1);

INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('LMHB', 'Vinificazione in Riduzione', 'Sgrondatura', 'seperazione del mosto dalle
bucce', 2);
```

```

INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('VGFA', 'Vinificazione in Riduzione', 'Fermentazione', 'inoculo dei lieviti', 3);

INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('YHER', 'Vinificazione in Riduzione', 'Trattamento Riducente', 'uso gas inerti antiossidanti', 4);

INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('QQLK', 'Vinificazione in Riduzione', 'Svinatura', 'separazione della parte buona del vino dai residui di lievi e bucce', 5);

INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('FGFS', 'Vinificazione in Riduzione', 'Pressatura delle vinacce', 'estrazione mosto residuo', 6);

INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('HNLS', 'Vinificazione in Riduzione', 'Affinamento', 'passaggio in botte per la maturazione del vino', 7);

INSERT INTO Fase (CodFase, Vinificazione, Nome, Descrizione, NumeroProgressivo)
VALUES ('MLAJ', 'Vinificazione in Riduzione', 'Imbottigliamento', 'travaso dalle botti alle bottiglie', 8);

```

#### Popolamento tabella *Stato*:

```

INSERT INTO Stato (Fase, Vinificazione, Lotto, DataInizio, DataFine)
VALUES ('AHJD', 'In rosso', 'MHNCD45715', TO_DATE('10/10/2011', 'dd/mm/yyyy'),
        TO_DATE('15/10/2011', 'dd/mm/yyyy'));

INSERT INTO Stato (Fase, Vinificazione, Lotto, DataInizio, DataFine)
VALUES ('BRRW', 'In rosso', 'MHNCD45715', TO_DATE('16/10/2011', 'dd/mm/yyyy'), NULL);

```

#### Popolamento tabella *Utenti*:

```

INSERT INTO Utenti (Username, Password, Ruolo)
VALUES ('MARIOROSSI', 'MRpassword', 'Amministrazione');

INSERT INTO Utenti (Username, Password, Ruolo)
VALUES ('MARIABIANCHI', 'MBpassword', 'Segreteria');

INSERT INTO Utenti (Username, Password, Ruolo)
VALUES ('ANTONIOVERDI', 'AVpassword', 'Cantina');

INSERT INTO Utenti (Username, Password, Ruolo)
VALUES ('LUCAROSSI', 'LRpassword', 'Analisi');

```



## Ruoli e utenti

Sono stati innanzitutto creati quattro ruoli, ognuno associato a una determinata categoria di utenti.

```
CREATE ROLE Cantina;  
CREATE ROLE Amministrazione;  
CREATE ROLE Analisi;  
CREATE ROLE Segreteria;
```

Analizziamoli nel dettaglio uno per uno:

- Il ruolo *Cantina* permette la gestione della sezione del sistema informativo dedicato alla trasformazione dell'uva. A seguire i privilegi che tale ruolo riassume.

```
GRANT SELECT          ON Cantina          TO Cantina;  
GRANT ALL PRIVILEGES ON ScartoAccumulato TO Cantina;  
GRANT SELECT          ON Vasca            TO Cantina;  
GRANT ALL PRIVILEGES ON Lotto             TO Cantina;  
GRANT ALL PRIVILEGES ON Stoccaggio        TO Cantina;  
GRANT ALL PRIVILEGES ON Trasferimento     TO Cantina;  
GRANT SELECT          ON Processo         TO Cantina;  
GRANT ALL PRIVILEGES ON Gestione          TO Cantina;  
GRANT ALL PRIVILEGES ON Lavorazione       TO Cantina;  
GRANT ALL PRIVILEGES ON Prodotto         TO Cantina;  
GRANT ALL PRIVILEGES ON Utilizzo          TO Cantina;  
GRANT SELECT          ON Vino             TO Cantina;  
GRANT ALL PRIVILEGES ON Finalita          TO Cantina;  
GRANT SELECT          ON Vinificazione    TO Cantina;  
GRANT ALL PRIVILEGES ON Fase              TO Cantina;  
GRANT ALL PRIVILEGES ON Stato             TO Cantina;
```

- Il ruolo *Amministrazione* è pensato per concedere una gestione completa del sistema informativo: per questo motivo ad esso sono associati i massimi privilegi su tutte le tabelle create.

```
GRANT ALL PRIVILEGES ON Cantina          TO Amministrazione;  
GRANT ALL PRIVILEGES ON Produttore       TO Amministrazione;  
GRANT ALL PRIVILEGES ON AffiliazionePassata TO Amministrazione;  
GRANT ALL PRIVILEGES ON Vigneto          TO Amministrazione;  
GRANT ALL PRIVILEGES ON Uva              TO Amministrazione;  
GRANT ALL PRIVILEGES ON SceltaVendemmiale TO Amministrazione;  
GRANT ALL PRIVILEGES ON Prenotazione     TO Amministrazione;  
GRANT ALL PRIVILEGES ON Carico            TO Amministrazione;  
GRANT ALL PRIVILEGES ON ScartoAccumulato TO Amministrazione;  
GRANT ALL PRIVILEGES ON Vasca            TO Amministrazione;  
GRANT ALL PRIVILEGES ON Lotto            TO Amministrazione;  
GRANT ALL PRIVILEGES ON Stoccaggio        TO Amministrazione;  
GRANT ALL PRIVILEGES ON Trasferimento     TO Amministrazione;  
GRANT ALL PRIVILEGES ON Processo         TO Amministrazione;  
GRANT ALL PRIVILEGES ON Gestione          TO Amministrazione;  
GRANT ALL PRIVILEGES ON Lavorazione       TO Amministrazione;  
GRANT ALL PRIVILEGES ON Prodotto         TO Amministrazione;  
GRANT ALL PRIVILEGES ON Utilizzo          TO Amministrazione;  
GRANT ALL PRIVILEGES ON Vino             TO Amministrazione;  
GRANT ALL PRIVILEGES ON Finalita          TO Amministrazione;  
GRANT ALL PRIVILEGES ON Vinificazione    TO Amministrazione;  
GRANT ALL PRIVILEGES ON Fase              TO Amministrazione;
```

```
GRANT ALL PRIVILEGES ON Stato TO Amministrazione;
GRANT ALL PRIVILEGES ON Campione TO Amministrazione;
GRANT ALL PRIVILEGES ON Parametro TO Amministrazione;
GRANT ALL PRIVILEGES ON Analisi TO Amministrazione;
```

- Il ruolo *Analisi* è invece ristretto al controllo delle tabelle riguardanti le analisi e la gestione dei campioni. A seguire i privilegi che tale ruolo riassume.

```
GRANT ALL PRIVILEGES ON Campione TO Analisi;
GRANT SELECT ON Parametro TO Analisi;
GRANT ALL PRIVILEGES ON Analisi TO Analisi;
GRANT SELECT ON Lotto TO Analisi;
```

- Il ruolo *Segreteria*, infine, permette la gestione di tutti quegli aspetti che non sono ancora stati menzionati, quindi l'anagrafica dei produttori, quella dei vigneti, le affiliazioni e tutta la parte riguardante le prenotazioni e il conferimento dell'uva. A seguire i privilegi che tale ruolo riassume.

```
GRANT SELECT ON Cantina TO Segreteria;
GRANT ALL PRIVILEGES ON Produttore TO Segreteria;
GRANT ALL PRIVILEGES ON AffiliazionePassata TO Segreteria;
GRANT ALL PRIVILEGES ON Vigneto TO Segreteria;
GRANT SELECT ON Uva TO Segreteria;
GRANT ALL PRIVILEGES ON SceltaVendemmiale TO Segreteria;
GRANT ALL PRIVILEGES ON Prenotazione TO Segreteria;
GRANT ALL PRIVILEGES ON Carico TO Segreteria;
```

Sono successivamente stati creati quattro utenti, ognuno esemplificativo di una delle quattro categorie associate ai ruoli appena presentati.

```
CREATE USER MarioRossi IDENTIFIED BY MRpassword;
CREATE USER MariaBianchi IDENTIFIED BY MBpassword;
CREATE USER AntonioVerdi IDENTIFIED BY AVpassword;
CREATE USER LucaRossi IDENTIFIED BY LRpassword;
```

```
GRANT Amministrazione TO MarioRossi;
GRANT Segreteria TO MariaBianchi;
GRANT Cantina TO AntonioVerdi;
GRANT Analisi TO LucaRossi;
```

Nell'applicativo, ogni categoria di utenti avrà accesso ad alcune maschere piuttosto che ad altre, come sarà chiaramente spiegato nel prossimo paragrafo.

## Funzionalità applicativo

Non appena l'applicazione viene avviata, compare la seguente schermata di caricamento. Essa rimane fino a che non è avvenuto il collegamento alla Base di Dati Oracle 11g XE.

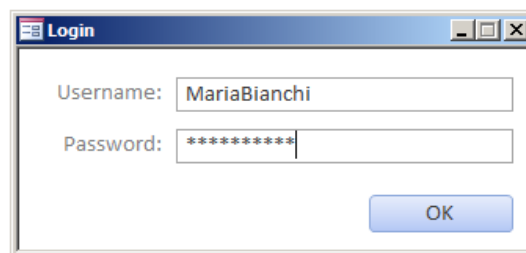


```
Option Compare Database
Private Sub Form_Timer()
    On Error GoTo Err_Form_Load
    Dim ws As Workspace
    Dim db As Database
    Dim Conn As String

    Conn = "ODBC;DSN=progettoDB;UID=SYSTEM;PWD=password"
    Set ws = DBEngine.Workspaces(0)
    Set db = ws.OpenDatabase("", False, False, Conn)
    DoCmd.Close
    DoCmd.OpenForm "Login"
    Exit Sub

Err_Form_Load:
    MsgBox "ERRORE: " & Err.Description
End Sub
```

Una volta concluso il caricamento compare una schermata di accesso in cui inserire le credenziali per autenticarsi nell'applicativo.



```
Option Compare Database
Private db As DAO.Database

Private Sub OK_Click()
    'Gestisce l'accesso all'applicativo
    Set db = CurrentDb
    Dim Data As Recordset

    sql = "SELECT RUOLO " & _
        "FROM PROGETTODB_SCHEMA_UTENTI " & _
        "WHERE USERNAME = '" & Me.Username & "' AND " & _
        "PASSWORD = '" & Me.Password & "';"
```

```

Set Data = db.OpenRecordset(sql)
'Controlla l'esistenza delle credenziali inserite
If Data.EOF Then
    MsgBox "Credenziali non valide"
    Me.Username = ""
    Me.Password = ""
    Exit Sub
End If

'Apri il form che corrisponde al ruolo dell'utente
Data.MoveFirst
Select Case Data.Fields!Ruolo
    Case "AMMINISTRAZIONE": DoCmd.OpenForm "Direzione"
    Case "SEGRETERIA":      DoCmd.OpenForm "SettoreSEGRETERIA"
    Case "CANTINA":        DoCmd.OpenForm "GestioneLOTTI"
    Case "ANALISI":        DoCmd.OpenForm "SettoreANALISI"
    Case Else: MsgBox "Ruolo non riconosciuto"
End Select
DoCmd.Close
End Sub

```

Come si intuisce dalla lettura del codice e da quanto affermato alla fine del precedente paragrafo, ad ogni utente è associato un determinato ruolo. Dopo avere effettuato l'accesso, per ogni utente viene aperta una maschera diversa, sulla base del ruolo ricoperto.

Le funzionalità riservate ai diversi ruoli saranno approfondite in paragrafi distinti.

## Segreteria

Se l'utente che ha effettuato l'accesso ha come ruolo "Segreteria", gli sarà mostrata la seguente maschera.

SettoreSEGRETERIA

La collina del Sole  
Settore SEGRETERIA

Conferimenti Logout Utente: MARIABIANCHI  
Ruolo: SEGRETERIA

Seleziona il produttore:  Cerca [Inserisci un nuovo produttore](#)

Codice Fiscale: -  
Partita IVA: -  
Affiliazione: - Dal: -

Affiliazioni passate:

CODCANTINA	INIZIO	FINE

Dati anagrafici:

Nome: -  
Cognome: -  
Sesso: -  
Data nascita: -  
Comune nascita: -  
Residenza: -  
Domicilio: -

Modifica dati

Vigneti associati al produttore:



Dettaglio Inserisci vigneto

Essa permette la gestione completa delle informazioni legate ai produttori, compresi i vigneti che coltivano e le scelte vendemmiali.

L'analisi inizia dall'intestazione:



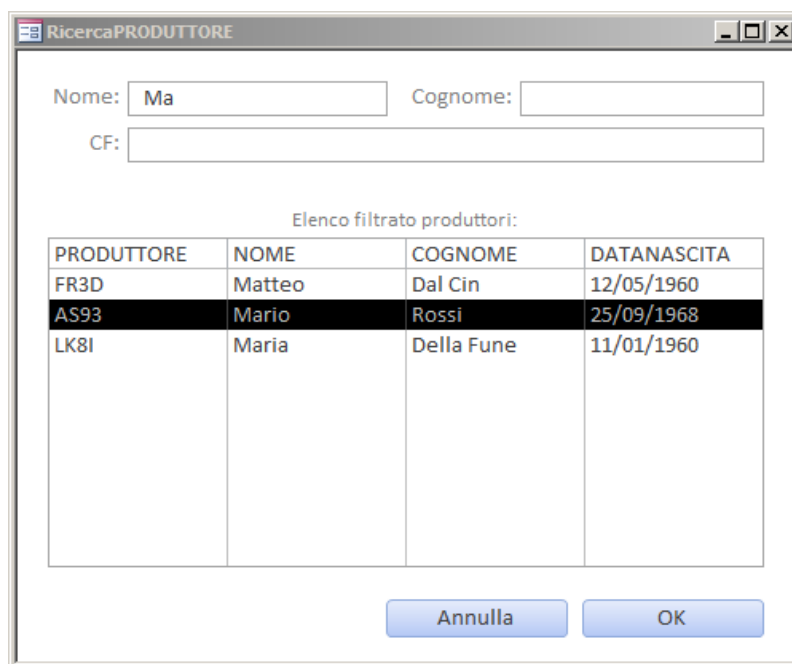
All'estrema destra sono indicati il nome e il ruolo dell'utente autenticato, preceduti da un pulsante che permette di annullare l'accesso e che rimanda alla maschera di login. Vi è poi un secondo pulsante che apre la maschera dedicata all'inserimento dei conferimenti (sarà analizzata in seguito).

```
Option Compare Database
Private Sub Logout_Click()
    For Each frm In Form
        DoCmd.Close
    Next frm
    DoCmd.OpenForm "Login"
End Sub

Private Sub GoToConferimenti_Click()
    DoCmd.Close
    DoCmd.OpenForm "SettoreCONFERIMENTI"
End Sub
```

Per quanto riguarda il corpo della maschera, si nota immediatamente come quasi tutti i pulsanti siano disattivati: la prima operazione da fare consiste infatti nel selezionare un produttore. Tale scelta può avvenire attraverso un menu a tendina, contenente tutti i codici dei produttori, oppure attraverso una maschera apposita di ricerca avanzata, a cui si accede premendo il pulsante *Cerca*.

La maschera si presenta così (in alto i campi in cui specificare i filtri e in basso l'elenco filtrato dei produttori).



PRODUTTORE	NOME	COGNOME	DATANASCITA
FR3D	Matteo	Dal Cin	12/05/1960
AS93	Mario	Rossi	25/09/1968
LK8I	Maria	Della Fune	11/01/1960

In base ai valori inseriti nei campi *Nome*, *Cognome* e *CF* (Codice Fiscale), la lista si aggiorna automaticamente: nell'immagine è stata inserita la stringa 'Ma' in *Nome* e sono comparsi 'Mario', 'Maria' e 'Matteo'.

Una volta trovato il produttore che si stava cercando, è sufficiente selezionarlo e premere il tasto *OK* (esso rimane disabilitato fino a che non viene selezionato qualcuno).

```
Option Compare Database
Private Sub Filtro()
'Aggiorna il contenuto di ElencoProduttori
    Dim sql As String
    sql = "SELECT CODPRODUTTORE AS PRODUTTORE, NOME, COGNOME, DATANASCITA " & _
        "FROM PROGETTODB_SCHEMA_PRODUTTORE " & _
        "WHERE NOME LIKE '%" & Me.Nome & "%' AND " & _
        "COGNOME LIKE '%" & Me.Cognome & "%' AND " & _
        "CODICEFISCALE LIKE '%" & Me.CF & "%';"
    Me.ElencoProduttori.RowSource = sql
End Sub

Private Sub Form_Load()
    Me.OK.Enabled = False
    Filtro
End Sub

'Aggiornamento di ElencoProduttori in base ai valori inseriti in Nome, Cognome e CF
Private Sub Nome_AfterUpdate()
    Filtro
End Sub
Private Sub Cognome_AfterUpdate()
    Filtro
End Sub
Private Sub CF_AfterUpdate()
    Filtro
End Sub

Private Sub ElencoProduttori_Click()
    Me.OK.Enabled = True
End Sub
Private Sub Annulla_Click()
    DoCmd.Close
End Sub
```

Una volta selezionato il produttore, secondo una delle due modalità, la maschera si popola con le informazioni ad esso associate.

SettoreSEGRETERIA

La collina del Sole  
Settore SEGRETERIA

Conferimenti Logout Utente: MARIABIANCHI  
Ruolo: SEGRETERIA

Seleziona il produttore: AS93 Cerca [Inserisci un nuovo produttore](#)

Codice Fiscale: RSMRA68J25V682I  
Partita IVA: 20164530075  
Affiliazione: 03 Dal: 06/06/2018

Affiliazioni passate:

CODCANTINA	INIZIO	FINE

Dati anagrafici:

Nome: Mario  
Cognome: Rossi  
Sesso: M  
Data nascita: 25/09/1968  
Comune nascita: Verona  
Residenza: Verona  
Domicilio: -

Modifica dati

Vigneti associati al produttore:

VIGNETO	VITIGNO	ESTENSIONE	ANNOIMPIANTO
PLIC	Pinot Nero	4	1964
EVSD	Glera	7	1999

Dettaglio Inserisci vigneto

Premendo su *Modifica dati* i campi di testo diventano modificabili e compaiono i pulsanti *Salva* e *Annulla* (con essi è evidentemente possibile salvare o annulla le modifiche apportate).

SettoreSEGRETERIA

La collina del Sole  
Settore SEGRETERIA

Conferimenti Logout Utente: MARIABIANCHI  
Ruolo: SEGRETERIA

Seleziona il produttore: AS93 Cerca [Inserisci un nuovo produttore](#) Annulla Salva

Codice Fiscale: RSMRA68J25V682I  
Partita IVA: 20164530075  
Affiliazione: 03 Dal: 06/06/2018

Affiliazioni passate:

CODCANTINA	INIZIO	FINE

Dati anagrafici:

Nome: Mario  
Cognome: Rossi  
Sesso: M  
Data nascita: 25/09/1968  
Comune nascita: Verona  
Residenza: Verona  
Domicilio: -

Modifica dati

Vigneti associati al produttore:

VIGNETO	VITIGNO	ESTENSIONE	ANNOIMPIANTO
PLIC	Pinot Nero	4	1964
EVSD	Glera	7	1999

Dettaglio Inserisci vigneto

Nella parte bassa sono invece elencati i vigneti gestiti dal produttore e per ognuno le informazioni più importanti.

Vigneti associati al produttore:

VIGNETO	VITIGNO	ESTENSIONE	ANNOIMPIANTO
PLIC	Pinot Nero	4	1964
EVSD	Glera	7	1999

Per visualizzare la scheda completa di un vigneto è sufficiente selezionarlo nella lista e premere il tasto *Dettaglio*. Si apre così una maschera contenente tutte le informazioni legate al particolare vigneto, comprese le scelte vendemmiali registrate fino a quel momento.

**DettaglioVIGNETO**

Produttore: AS93

Vitigno: Pinot Nero

Estensione: 4

Posizione: 45.235489 12.34568

Impianto: 1964      Inizio produzione: 1964

Scelta vendemmiale (2019)

Scelte vendemmiali precedenti:

UVA	ANNO
145	2013

**DettaglioVIGNETO**

Produttore: AS93

Vitigno: Pinot Nero

Estensione: 4

Posizione: 45.235489 12.34568

Impianto: 1964      Inizio produzione: 1964

Scelta vendemmiale (2019)

Scelte vendemmiali precedenti:

UVA	ANNO
145	2019
145	2013

Nel caso nella lista non sia presente la scelta vendemmiale per l'anno corrente (immagine a destra), in questo caso il 2019, è possibile inserirla scegliendo l'uva dal menu a tendina e poi premendo il tasto *Salva*. Nel caso in cui invece fosse già presente (immagine a sinistra), il campo viene disabilitato

```

Option Compare Database
Private db As DAO.Database
Private CodVigneto As String
Private Sub Form_Load()
    CodVigneto = Me.OpenArgs

    'Ricava le informazioni associate al vigneto selezionato
    Set db = CurrentDb
    Dim Data As Recordset
    sql = "SELECT * " & _
        "FROM PROGETTODB_SCHEMA_VIGNETO " & _
        "WHERE CODVIGNETO = '" & CodVigneto & "';"

    Set Data = db.OpenRecordset(sql)
    Data.MoveFirst

```



```

'Compila i campi in base al produttore selezionato
Me.Produttore.Caption = Data.Fields!CodProduttore
Me.Vitigno.Caption = Data.Fields!Vitigno
Me.Estensione.Caption = Data.Fields!Estensione
Me.Posizione.Caption = Data.Fields!Posizione
Me.AnnoImpianto.Caption = Data.Fields!AnnoImpianto
Me.AnnoInizioProduzione.Caption = Data.Fields!AnnoInizioProduzione

sql = "SELECT CODUVA, NOME, DENOMINAZIONE " & _
      "FROM PROGETTODB_SCHEMA_UVA " & _
      "ORDER BY NOME;"
Me.SceltaVendemmiale.RowSource = sql

Me.Salva.Enabled = False
Me.Label_SceltaVendemmiale.Caption = "Scelta vendemmiale (" & Format(Now(), "yyyy") & ")"

sql = "SELECT CODUVA AS UVA, ANNO " & _
      "FROM PROGETTODB_SCHEMA_SCELTAVENDEMMIALE " & _
      "WHERE CODVIGNETO = '" & CodVigneto & "' " & _
      "ORDER BY ANNO DESC;"
Me.ScelteVendemmialiPrecedenti.RowSource = sql
Me.SceltaVendemmiale.Enabled = True

Set Data = db.OpenRecordset(sql)
If IsNull(Data) Then
    Exit Sub
End If
Data.MoveFirst
If Data.Fields!Anno = 2019 Then
    Me.SceltaVendemmiale.Enabled = False
End If
End Sub

Private Sub SceltaVendemmiale_AfterUpdate()
    Me.Salva.Enabled = True
End Sub

Private Sub Salva_Click()
    On Error GoTo Err_Salva_Click
    sql = "INSERT INTO PROGETTODB_SCHEMA_SCELTAVENDEMMIALE(CODUVA, CODVIGNETO, ANNO) " & _
          "VALUES ('" & Me.SceltaVendemmiale & "', '" & _
          CodVigneto & "', " & _
          Format(Now(), "yyyy") & ");"
    DoCmd.RunSQL (sql)
    MsgBox "Inserimento effettuato con successo"
    DoCmd.Close
    Exit Sub

Err_Salva_Click:
    MsgBox "ERRORE: " & Err.Description
    DoCmd.Close
End Sub

Private Sub Annulla_Click()
    DoCmd.Close
End Sub

```

Oltre alla visualizzazione del dettaglio di un particolare vigneto, è anche possibile inserirne uno nuovo, sempre associato al produttore selezionato, premendo il pulsante *Inserisci vigneto*. Si aprirà in questo caso una maschera di inserimento, qui sotto riportata.

The image shows a Windows-style dialog box titled "InserimentoVIGNETO". Inside the dialog, there are several input fields and buttons. The "Produttore" field is pre-filled with "AS93". Below it are empty text boxes for "Vitigno", "Estensione", and "Posizione". Further down are two smaller text boxes for "Impianto" and "Inizio produzione". At the bottom of the dialog, there are two blue buttons: "Annulla" on the left and "Inserisci" on the right.

```
Option Compare Database
Private Sub Form_Load()
    MsgBox Form.OpenArgs
    Me.Produttore.Caption = Form.OpenArgs
    Me.Vitigno.SetFocus
End Sub

Private Sub Inserisci_Click()
    'Gestisce l'inserimento di un nuovo vigneto
    On Error GoTo Err_Inserisci_Click
    CodVigneto = "XXXXXX"

    sql = "INSERT INTO PROGETTODB_SCHEMA VIGNETO (CODVIGNETO, ESTENSIONE, POSIZIONE, " & _
        "ANNOIMPIANTO, ANNOINIZIOPRODUZIONE, VITIGNO, CODPRODUTTORE) " & _
        "VALUES ('" & CodVigneto & "', '" & _
        Me.Estensione & "', '" & _
        Me.Posizione & "', '" & _
        Me.AnnoImpianto & "', '" & _
        Me.AnnoInizioProduzione & "', '" & _
        Me.Vitigno & "', '" & _
        Me.Produttore.Caption & "');"

    DoCmd.RunSQL sql
    MsgBox "Inserimento effettuato con successo"
Exit Sub

Err_Inserisci_Click:
    MsgBox "ERRORE: " & Err.Description
End Sub

Private Sub Annulla_Click()
    DoCmd.Close
End Sub
```

L'analisi della maschera si conclude con la funzionalità di inserimento di un nuovo produttore, attivabile dalla schermata principale attraverso il relativo pulsante, presente a destra del pulsante *Cerca*.

A seguire il codice VBA utilizzato per la realizzazione di tutte le funzionalità esposte.

```
Option Compare Database
Private db As DAO.Database
Private CodProduttore As String
Dim Attributo(10) As String
Dim Stato As String

Private Sub PopolaAttributi()
'Definizione dei campi potenzialmente modificabili
    Attributo(0) = "CF"
    Attributo(1) = "PartitaIVA"
    Attributo(2) = "Affiliazione"
    Attributo(3) = "Nome"
    Attributo(4) = "Cognome"
    Attributo(5) = "Sesso"
    Attributo(6) = "DataNascita"
    Attributo(7) = "ComuneNascita"
    Attributo(8) = "Residenza"
    Attributo(9) = "Domicilio"
    Attributo(10) = "AffiliazioneDal"
End Sub

Private Sub Form_Load()
    PopolaAttributi
    Clear Me, Attributo, "-"
    ReadMode Me, Attributo
    Stato = "Visualizzazione"
```

```

sql = "SELECT CODPRODUTTORE " & _
      "FROM PROGETTODB_SCHEMA_PRODUTTORE " & _
      "ORDER BY CODPRODUTTORE"
Me.Produttore.RowSource = sql
Me.Produttore.Enabled = True
Me.Produttore.SetFocus
Me.Produttore = ""

Me.Cerca.Enabled = True
Me.Modifica.Enabled = False
Me.Salva.Visible = False
Me.Annulla.Visible = False

Me.VignetiProduttore.RowSource = ""
Me.VignetiProduttore.Enabled = True
Me.DettaglioVigneto.Enabled = False
Me.InserisciVigneto.Enabled = False
End Sub

Private Sub InserisciProduttore_Click()
'Attiva la modalit  di inserimento di un nuovo produttore
Clear Me, Attributo, ""
WriteMode Me, Attributo
Stato = "Inserimento"

Me.Produttore = "XXXX"
Me.CF.SetFocus

Me.Salva.Visible = True
Me.Annulla.Visible = True
Me.Modifica.Enabled = False
Me.Produttore.Enabled = False
Me.Cerca.Enabled = False
Me.DettaglioVigneto.Enabled = False
Me.InserisciVigneto.Enabled = False
Me.VignetiProduttore.Enabled = False
End Sub

Private Sub Modifica_Click()
'Attiva la modalit  di modifica di un produttore esistente
WriteMode Me, Attributo
Stato = "Modifica"

Me.CF.SetFocus

Me.Salva.Visible = True
Me.Annulla.Visible = True
Me.Modifica.Enabled = False
Me.Produttore.Enabled = False
Me.Cerca.Enabled = False
Me.DettaglioVigneto.Enabled = False
Me.InserisciVigneto.Enabled = False
Me.VignetiProduttore.Enabled = False
End Sub

```

```

Private Sub Salva_Click()
'Salva l'inserimento/la modifica di un utente, in base alla variabile Stato
    If Stato = "Inserimento" Then
        InserimentoProduttore
    ElseIf Stato = "Modifica" Then
        ModificaProduttore
    End If
End Sub

Private Sub Annulla_Click()
'Annulla l'inserimento/la modifica di un utente
    Form_Load
End Sub

Private Sub InserimentoProduttore()
'Gestisce l'inserimento di un nuovo utente
    On Error GoTo Err_InserimentoProduttore

    sql = "INSERT INTO PROGETTODB_SCHEMA_PRODUTTORE(CodProduttore, PartitaIva, " & _
        "CodiceFiscale, Nome, Cognome, Sesso, DataNascita, " & _
        "ComuneNascita, Residenza, Domicilio, CantinaAffiliazione, " & _
        "DataInizioAffiliazione) " & _
        "VALUES ('" & _
            Me.Produttore & "', '" & _
            Me.PartitaIVA & "', '" & _
            Me.CF & "', '" & _
            Me.Nome & "', '" & _
            Me.Cognome & "', '" & _
            Me.Sesso & "', #" & _
            Format(Me.DataNascita, "dd/mm/yyyy") & "#, '" & _
            Me.ComuneNascita & "', '" & _
            Me.Residenza & "', '" & _
            Me.Domicilio & "', '" & _
            Me.Affiliazione & "', #" & _
            Format(Me.AffiliazioneDal, "dd/mm/yyyy") & "#);"

    DoCmd.RunSQL sql
    MsgBox "Inserimento effettuato con successo"
    Form_Load
    Exit Sub

Err_InserimentoProduttore:
    MsgBox "ERRORE: " & Err.Description
End Sub

Private Sub ModificaProduttore()
'Gestisce la modifica dei dati di un utente
    On Error GoTo Err_ModificaProduttore

    sql = "UPDATE PROGETTODB_SCHEMA_PRODUTTORE " & _
        "SET PARTITAIVA = '" & Me.PartitaIVA & "', " & _
        "CODICEFISCALE = '" & Me.CF & "', " & _
        "NOME = '" & Me.Nome & "', " & _
        "COGNOME = '" & Me.Cognome & "', " & _
        "SESSO = '" & Me.Sesso & "', " & _
        "DATANASCITA = #" & Format(Me.DataNascita, "dd/mm/yyyy") & "#, " & _
        "COMUNENASCITA = #" & Me.ComuneNascita & "', " & _
        "RESIDENZA = '" & Me.Residenza & "', " & _

```

```

        "DOMICILIO = '" & Me.Domicilio & "', " & _
        "CANTINAAFFILIAZIONE = '" & Me.Affiliazione & "', " & _
        "DATAINIZIOAFFILIAZIONE='" & Format(Me.AffiliazioneDal, "dd/mm/yyyy") & "# " & _
        "WHERE CODPRODUTTORE = '" & CodProduttore & "';"
DoCmd.RunSQL sql
MsgBox "Modifica effettuata con successo"
Form_Load
Exit Sub

Err_ModificaProduttore:
MsgBox "ERRORE: " & Err.Description
End Sub

Private Sub Produttore_AfterUpdate()
On Error GoTo Exit_Produttore_AfterUpdate

CodProduttore = Me.Produttore

'Ricava le informazioni associate al produttore selezionato
Set db = CurrentDb
Dim Data As Recordset
sql = "SELECT * " & _
      "FROM PROGETTODB_SCHEMA_PRODUTTORE " & _
      "WHERE CODPRODUTTORE = '" & CodProduttore & "';"
Set Data = db.OpenRecordset(sql)
Data.MoveFirst

Me.CF = Data.Fields!CODICEFISCALE
Me.PartitaIVA = Data.Fields!PartitaIVA
Me.Nome = Data.Fields!Nome
Me.Cognome = Data.Fields!Cognome
Me.Sesso = Data.Fields!Sesso
Me.DataNascita = Data.Fields!DataNascita
Me.ComuneNascita = Data.Fields!ComuneNascita
Me.Residenza = Data.Fields!Residenza

If Not IsNull(Data.Fields!Domicilio) Then
    Me.Domicilio = Data.Fields!Domicilio
Else: Me.Domicilio = "-"
End If

If Not IsNull(Data.Fields!CantinaAffiliazione) Then
    Me.Affiliazione = Data.Fields!CantinaAffiliazione
    Me.AffiliazioneDal = Data.Fields!DataInizioAffiliazione
Else
    Me.Affiliazione = "-"
    Me.AffiliazioneDal = "-"
End If

'Riempie la tabella delle affiliazioni passate
sql = "SELECT CODCANTINA, DATAINIZIO AS INIZIO, DATAFINE AS FINE " & _
      "FROM PROGETTODB_SCHEMA_AFFILIAZIONEPASSATA " & _
      "WHERE CODPRODUTTORE = '" & CodProduttore & "' " & _
      "ORDER BY DATAINIZIO"
Me.AffiliazioniPassate.RowSource = sql

```

```

'Riempie la tabella dei vigneti
    sql = "SELECT CODVIGNETO AS VIGNETO, VITIGNO, ESTENSIONE, ANNOIMPIANTO " & _
          "FROM PROGETTODB_SCHEMA_VIGNETO " & _
          "WHERE CODPRODUTTORE = '" & CodProduttore & "';"
    Me.VignetiProduttore.RowSource = sql

    Me.Modifica.Enabled = True
    Me.InserisciVigneto.Enabled = True
    Me.DettaglioVigneto.Enabled = False
    Exit Sub

Exit_Produttore_AfterUpdate:
    MsgBox "ERRORE: " & Err.Description
    Form_Load
    Exit Sub
End Sub

Private Sub VignetiProduttore_Click()
    Me.DettaglioVigneto.Enabled = True
End Sub

Private Sub DettaglioVigneto_Click()
    'Apre il form DettaglioVIGNETO passandogli il codice vigneto
    DoCmd.OpenForm "DettaglioVIGNETO", , , , , Me.VignetiProduttore
End Sub

Private Sub InserisciVigneto_Click()
    'Apre il form InserimentoVIGNETO passandogli il codice produttore
    DoCmd.OpenForm "InserimentoVIGNETO", , , , , Me.Produttore
End Sub

```

Si passa dunque all'analisi della maschera, già citata, dedicata alla memorizzazione dei conferimenti.

**SettoreCONFERIMENTI**

La collina del Sole  
Settore CONFERIMENTI

Segreteria Logout Utente: MARIABIANCHI  
Ruolo: SEGRETERIA

Produttore:  Cerca

Prenotazione:

Uva:

Quantità:  q.li

Proprietà uva conferita:

pH:

Grado zuccherino:

Materiale estraneo:

Integrità acini:

Ultimi conferimenti inseriti:

PRODUTTORE	UVA	QUANTITA	DATAEORA
LK8I	145	75	01/01/2013 10:20:33
AS93	311	100	26/08/2011 15:02:37
FR3D	284	78	26/08/2011 13:12:45

Annulla Aggiungi conferimento

L'intestazione è del tutto analoga a quella della precedente e non sarà perciò approfondita.

Si passa dunque all'analisi del corpo della maschera. Il primo passo consiste nella selezione del produttore (le modalità di selezione sono uguali a quelle precedentemente presentate). La lista *Prenotazione* viene riempita in base al produttore selezionato, così da facilitare il compito dell'operatore che deve controllare se il produttore, prima di portare il carico d'uva, ha effettuato una regolare prenotazione. Nella lista vengono elencate le prenotazioni a nome del produttore selezionato previste per il giorno corrente.

**SettoreCONFERIMENTI**

La collina del Sole  
Settore CONFERIMENTI

Segreteria Logout Utente: MARIABIANCHI  
Ruolo: SEGRETERIA

Produttore: AS93 Cerca

Prenotazione: MGJEK3 Bianco

Uva: 145

Quantità: 200 q.li

Proprietà uva conferita:

pH: 9,22

Grado zuccherino: 14,6

Materiale estraneo: False

Integrità acini:

Ultimi conferimenti inseriti:

PRODUTTORE	UVA	QUANTITA	DATAEORA
LK8I	145	75	01/01/2013 10:20:33
AS93	311	100	26/08/2011 15:02:37
FR3D	284	78	26/08/2011 13:12:45

Annulla Aggiungi conferimento

Premendo il pulsante *Aggiungi conferimento* si inseriscono le informazioni inserite nei vari campi, mentre con il tasto *Annulla* si ritorna alla situazione iniziale.



Nella tabella di destra sono riassunti gli ultimi conferimenti registrati.

```
Option Compare Database
Private Sub Form_Load()
    sql = "SELECT CODPRODUTTORE FROM PROGETTODB_SCHEMA_PRODUTTORE"
    Me.Produttore.RowSource = sql

    Me.Produttore.Enabled = True
    Me.CercaProduttore.Enabled = True
    Me.Prenotazione.Enabled = False
    Me.Uva.Enabled = False
    Me.Quantita.Enabled = False
    Me.pH.Enabled = False
    Me.GradoZuccherino.Enabled = False
    Me.MaterialeEstraneo.Enabled = False
    Me.IntegritaAcini.Enabled = False
    Me.AggiungiConferimento.Enabled = False

    'Pulisce il contenuto dei campi di inserimento
    Me.Produttore = Null
    Me.Prenotazione.RowSource = ""
    Me.Uva = Null
    Me.Quantita = Null
    Me.pH = Null
    Me.GradoZuccherino = Null
    Me.MaterialeEstraneo = Null
    Me.IntegritaAcini = Null

    'Refresh della lista UltimiConferimenti
    sql = "SELECT P.CODPRODUTTORE AS PRODUTTORE, C.UVA, C.QUANTITA, C.DATAEORA " & _
        "FROM PROGETTODB_SCHEMA_PRENOTAZIONE AS P INNER JOIN " & _
        "PROGETTODB_SCHEMA_CARICO AS C ON P.CODPRENOTAZIONE = C.PRENOTAZIONE " & _
        "ORDER BY C.DATAEORA DESC;"
    Me.UltimiConferimenti.RowSource = sql
End Sub

'Abilita la ricerca avanzata di un produttore
Private Sub CercaProduttore_Click()
    DoCmd.OpenForm "RicercaPRODUTTORE"
End Sub

'Abilita e riempie il campo prenotazione
Private Sub Produttore_AfterUpdate()
    Me.Produttore.Enabled = False
    Me.CercaProduttore.Enabled = False

    Me.Prenotazione.Enabled = True
    Me.Uva.Enabled = True
    Me.Quantita.Enabled = True
    Me.pH.Enabled = True
    Me.GradoZuccherino.Enabled = True
    Me.MaterialeEstraneo.Enabled = True
    Me.IntegritaAcini.Enabled = True
    Me.AggiungiConferimento.Enabled = True

    sql = "SELECT CODPRENOTAZIONE, COLORE " & _
        "FROM PROGETTODB_SCHEMA_PRENOTAZIONE " & _
```

```

        "WHERE CODPRODUTTORE = '" & Me.Produttore & "';"
    Me.Prenotazione.RowSource = sql
End Sub

Private Sub AggiungiConferimento_Click()
    'Gestisce l'inserimento di un nuovo conferimento
    On Error GoTo Err_AggiungiConferimento_Click
    sql = "INSERT INTO PROGETTODB_SCHEMA_CARICO(PRENOTAZIONE, DATAEORA, UVA, " & _
        "QUANTITA, PH, GRADOZUCCHERINO, INTEGRITAACINI, " & _
        "MATERIALEESTRANEO) " & _
        "VALUES('" & Me!Prenotazione & "', #" & _
        Format(Now(), "dd/mm/yyyy HH:MM:SS") & "#, '" & _
        Me!Uva & "', " & _
        Me!Quantita & ", [" & _
        Me!pH & "], [" & _
        Me!GradoZuccherino & "], '" & _
        Me!IntegritaAcini & "', " & _
        Me!MaterialeEstraneo & ");"

    DoCmd.RunSQL sql
    MsgBox "Inserimento effettuato con successo"
    Form_Load
Exit Sub

Err_AggiungiConferimento_Click:
    MsgBox "ERRORE: " & Err.Description
End Sub

Private Sub Annulla_Click()
    Form_Load
End Sub

```

## Cantina

Se l'utente che ha effettuato l'accesso ha come ruolo "Cantina", gli sarà mostrata la seguente maschera.

The screenshot shows the 'Gestione LOTTI' application window. The title bar reads 'Gestione LOTTI'. The header area includes a logo of a bunch of grapes, the text 'La collina del Sole Settore CANTINA', a 'Logout' button, and user information: 'Utente: ANTONIOVERDI' and 'Ruolo: CANTINA'. The main area contains a form for selecting a lot. It starts with 'Seleziona il lotto:' followed by a dropdown menu, a 'Cerca' button, and a link 'Crea un nuovo lotto'. Below this are several fields: 'Colore: -', 'Fase corrente: -', 'Famiglia: -', 'Denominazione: -', 'Annata: -', 'Data origine: -', 'Fase successiva: -', 'AttoAlTaglio: -', and 'Descrizione: -'. A 'Passa alla fase successiva' button is located to the right of the 'Descrizione' field. At the bottom, there is a section titled 'Lavorazioni svolte:' with a large empty table. Below the table are two buttons: 'Fasi' and 'Processi'. At the bottom right, there is a button 'Aggiungi lavorazione'.

Essa permette la gestione e la visualizzazione dello storico dei lotti.

L'intestazione è analoga a quelle già presentate e non sarà perciò ulteriormente approfondita. Si noti solo come sia cambiato, ovviamente, il nome e il ruolo dell'utente

Si passa dunque all'analisi del corpo della maschera. Si osserva immediatamente come quasi tutti i pulsanti siano disattivati: la prima operazione consiste infatti nella selezione di un lotto. Tale scelta può avvenire o attraverso il menu a tendina, contenente tutti i codici dei lotti, oppure attraverso una maschera di ricerca avanzata appositamente creata, a cui si accede premendo il pulsante *Cerca*.

The screenshot shows the 'Ricerca LOTTO' application window. It contains search filters: 'Famiglia: Pro', 'Denominazione:', 'Colore:', 'Data origine:', 'Annata:', and 'Atto al taglio:'. Below the filters is a table titled 'Elenco filtrato produttori:'. The table has four columns: 'CODLOTTO', 'FAMIGLIA', 'DENOMINAZIONE', and 'DATAORIGINE'. The first two rows are visible: 'KDJNH78463' with 'Prosecco' and 'DOC' and '01/10/2011', and 'MGVRE44682' with 'Prosecco' and 'DOCG Superiore' and '02/09/2015'. At the bottom, there are two buttons: 'Annulla' and 'OK'.

CODLOTTO	FAMIGLIA	DENOMINAZIONE	DATAORIGINE
KDJNH78463	Prosecco	DOC	01/10/2011
MGVRE44682	Prosecco	DOCG Superiore	02/09/2015

Tale maschera, al pari di quella dedicata alla ricerca avanzata dei produttori, è formata da un'intestazione, contenente i campi secondo i quali è possibile filtrare i lotti, e da una lista, contenente un elenco dei lotti filtrati. In base ai valori inseriti nei campi *Famiglia*, *Denominazione*, *Colore*, *Data origine*, *Annata* e *Atto al taglio*, la lista si aggiorna automaticamente.

Una volta trovato il lotto è sufficiente selezionarlo e premere il tasto *OK* (esso è disabilitato fino al momento in cui non viene selezionato un lotto).

```
Option Compare Database
Private Sub Filtro()
'Aggiorna il contenuto di ElencoLotti
    Dim sql As String
    sql = "SELECT CODLOTTO, FAMIGLIA, DENOMINAZIONE, DATAORIGINE " & _
        "FROM PROGETTODB_SCHEMA_LOTTO " & _
        "WHERE FAMIGLIA LIKE '%" & Me.Famiglia & "%' AND " & _
        "DENOMINAZIONE LIKE '%" & Me.Denominazione & "%' AND " & _
        "COLORE LIKE '%" & Me.Colore & "%' AND " & _
        "ATTOALTAGLIO LIKE '%" & Me.AttoAlTaglio & "%' "
    If Not IsNull(Me.Annata) Then
        sql = sql & "AND ANNATA = " & Me.Annata
    End If
    If Not IsNull(Me.DataOrigine) Then
        sql = sql & "AND DATAORIGINE = #" & Format(Me.DataOrigine, "dd/mm/yyyy") & "#"
    End If
    sql = sql & ";"
    Me.ElencoLotti.RowSource = sql
End Sub

Private Sub Form_Load()
    Me.OK.Enabled = False
    Filtro
End Sub

'Aggiornamento di ElencoLotti in base ai valori inseriti nei vari campi
Private Sub Famiglia_AfterUpdate()
    Filtro
End Sub
Private Sub Denominazione_AfterUpdate()
    Filtro
End Sub
Private Sub Colore_AfterUpdate()
    Filtro
End Sub
Private Sub Annata_AfterUpdate()
    Filtro
End Sub
Private Sub AttoAlTaglio_AfterUpdate()
    Filtro
End Sub
Private Sub DataOrigine_AfterUpdate()
    Filtro
End Sub

Private Sub ElencoLotti_Click()
    Me.OK.Enabled = True
End Sub
```

```
Private Sub Annulla_Click()
    DoCmd.Close
End Sub
```

Una volta selezionato il produttore, la maschera si popola con le informazioni ad esso associate.

Seleziona il lotto:   [Crea un nuovo lotto](#)

Colore: Bianco      Fase corrente: Fermentazione alcolica

Famiglia: Chardonnay      Descrizione:

Denominazione: VDT      Forma di metabolismo energetico che permette la trasformazione del mosto in vino.

Annata: 2012      Fase successiva: Chiarifica

Data origine: 02/11/2012

AttoAlTaglio: False

Descrizione: -

Lavorazioni svolte:

LOTTO	PROCESSO	QTA_INIZIALE	QTA_FINALE	PERDITA	INIZIO	FINE
LKCSR16425	Fermentazione	300	290	1	02/10/2015	15/10/2015
LKCSR16425	Invecchiamento	250	240	3	10/11/2015	15/04/2017

Nella sezione a destra viene mostrata la fase di lavorazione in cui il lotto si trova, con relativa descrizione, e la fase successiva, così da facilitare il lavoro del capo cantina. Premendo il pulsante *Passa alla fase successiva* la fase successiva sarà considerata come la nuova fase corrente.

Nella sezione in basso sono invece elencate le lavorazioni cui il lotto è stato sottoposto fino ad ora, con tutte le relative caratteristiche. Premendo il pulsante *Aggiungi lavorazione* si apre una maschera che permette l'inserimento di una nuova lavorazione.

Seleziona il processo:

Prodotti utilizzati:

CODICE	NOME
CFA785	Enoferm R
GER526	Acido Malico

Q.tà iniziale:  |

Q.tà finale:  |

Perdita:  |

Data inizio:       Data fine:

Seleziona il processo:

Prodotti utilizzati:

CODICE	NOME
CFA785	Enoferm R
GER526	Acido Malico

Q.tà iniziale:  |

Q.tà finale:  |

Perdita:  |

Data inizio:       Data fine:

Il primo passo per l'inserimento consiste nella selezione del processo (immagine a sinistra), dopodiché anche gli altri campi vengono abilitati e lo si può concludere (immagine a destra). La lista che compare appena sotto la riga dedicata all'inserimento del processo ricorda quanti e quali prodotti debbano essere utilizzati.

```
Option Compare Database
Private CodLotto As String
Private Sub Form_Load()
    CodLotto = OpenArgs
    sql = "SELECT PROCESSO " & _
        "FROM PROGETTODB_SCHEMA_GESTIONE " & _
        "WHERE CANTINA = '01';"
    Me.Processo.RowSource = sql
    Me.Processo.Enabled = True
    Me.Processo.SetFocus

    Me.QuantitaIniziale.Enabled = False
    Me.QuantitaFinale.Enabled = False
    Me.Perdita.Enabled = False
    Me.DataInizio.Enabled = False
    Me.DataFine.Enabled = False
    Me.ElencoProdotti.RowSource = ""
End Sub

Private Sub Processo_AfterUpdate()
    sql = "SELECT P.CODPRODOTTO AS CODICE, P.NOME " & _
        "FROM PROGETTODB_SCHEMA_PRODOTTO P INNER JOIN " & _
        "PROGETTODB_SCHEMA_UTILIZZO U ON P.CODPRODOTTO = U.PRODOTTO " & _
        "WHERE U.PROCESSO = '" & Me.Processo & "';"
    Me.ElencoProdotti.RowSource = sql

    Me.QuantitaIniziale.Enabled = True
    Me.QuantitaFinale.Enabled = True
    Me.Perdita.Enabled = True
    Me.DataInizio.Enabled = True
    Me.DataFine.Enabled = True
    Me.Processo.Enabled = False
End Sub

Private Sub Inserisci_Click()
    'Gestisce l'inserimento di una nuova lavorazione
    On Error GoTo Err_Inserisci_Click
    sql = "INSERT INTO PROGETTODB_SCHEMA_LAVORAZIONE (LOTTO, PROCESSO, " & _
        "QUANTITA INIZIALE, QUANTITA FINALE, PERDITA, DATA INIZIO, DATA FINE) " & _
        "VALUES ('" & CodLotto & "', '" & _
        Me.Processo & "', " & _
        Me.QuantitaIniziale & ", " & _
        Me.QuantitaFinale & ", " & _
        Me.Perdita & ", #" & _
        Me.DataInizio & "#, #" & _
        Me.DataFine & "#);"

    DoCmd.RunSQL (sql)
    MsgBox "Inserimento effettuato con successo"
    DoCmd.Close
Exit Sub
Err_Inserisci_Click:
    MsgBox "Errore durante l'inserimento: " & Err.Description
    DoCmd.OpenForm "Form1"
End Sub
```

```

Err_Inserisci_Click:
    MsgBox "ERRORE: " & Err.Description
End Sub

Private Sub Annulla_Click()
    DoCmd.Close
End Sub

```

L'ultima funzionalità della sezione dedicata alla gestione dei lotti consiste nell'inserimento di un nuovo lotto. Per accedervi è sufficiente premere il relativo pulsante, posizionato a destra del pulsante *Cerca*.

Ci sono infine due pulsanti, posizionati in basso a sinistra, che permettono di accedere alla gestione delle fasi e alla gestione dei processi, attraverso le rispettive maschere. In seguito saranno analizzate nel dettaglio.

Intanto ecco il codice VBA utilizzato per la realizzazione di questa maschera.

```

Option Compare Database
Private db As DAO.Database
Private CodLotto As String
Dim Attributo(6) As String

Private Sub PopolaAttributi()
'Definizione dei campi potenzialmente modificabili
    Attributo(0) = "Colore"
    Attributo(1) = "Famiglia"
    Attributo(2) = "Denominazione"
    Attributo(3) = "Annata"
    Attributo(4) = "DataOrigine"
    Attributo(5) = "AttoAlTaglio"

```

```

        Attributo(6) = "Descrizione"
End Sub

Private Sub Form_Load()
    PopolaAttributi
    Clear Me, Attributo, "-"
    ReadMode Me, Attributo
    Me.Lotto = ""

    sql = "SELECT CODLOTTO " & _
        "FROM PROGETTODB_SCHEMA_LOTTO " & _
        "ORDER BY CODLOTTO;"
    Me.Lotto.RowSource = sql
    Me.Lotto.Enabled = True
    Me.Lotto.SetFocus

    Me.ElencoLavorazioni.RowSource = ""
    Me.ElencoLavorazioni.Enabled = True
    Me.Cerca.Enabled = True
    Me.Salva.Visible = False
    Me.Annulla.Visible = False
    Me.PassaFaseSuccessiva.Enabled = False
    Me.AggiungiLavorazione.Enabled = False
End Sub

Private Sub CreaLotto_Click()
    'Attiva la modalità di inserimento di un nuovo lotto
    Clear Me, Attributo, ""
    WriteMode Me, Attributo

    Me.Lotto = "XXXXXXXXXX"
    Me.ElencoLavorazioni.RowSource = ""
    Me.Colore.SetFocus

    Me.Lotto.Enabled = False
    Me.Cerca.Enabled = False
    Me.Salva.Visible = True
    Me.Annulla.Visible = True
    Me.ElencoLavorazioni.Enabled = False
End Sub

Private Sub Salva_Click()
    'Gestisce la creazione di un lotto
    On Error GoTo Err_Salva_Click

    sql = "INSERT INTO PROGETTODB_SCHEMA_LOTTO (CODLOTTO, ANNATA, DENOMINAZIONE, " & _
        "FAMIGLIA, COLORE, ATTOALTAGLIO, DATAORIGINE) " & _
        "VALUES ('" & Me.Lotto & "', " & _
        Me.Annata & ", '" & _
        Me.Denominazione & "', '" & _
        Me.Famiglia & "', '" & _
        Me.Colore & "', '" & _
        Me.AttoAlTaglio & "', #" & _
        Format(Me.DataOrigine, "dd/mm/yyyy") & "#');"

    DoCmd.RunSQL (sql)
    MsgBox "Inserimento effettuato con successo"

```



```

ReadMode Me, Attributo
Me.Lotto.Enabled = True
Me.Lotto.SetFocus

Me.Cerca.Enabled = True
Me.Salva.Visible = False
Me.Annulla.Visible = False
Exit Sub

Err_Salva_Click:
MsgBox "ERRORE: " & Err.Description
End Sub

Private Sub Annulla_Click()
Form_Load
End Sub
Private Sub Cerca_Click()
DoCmd.OpenForm "RicercaLOTTO"
End Sub

Private Sub Lotto_AfterUpdate()
On Error GoTo Exit_Lotto_AfterUpdate
CodLotto = Me.Lotto

'Ricava le informazioni associate al lotto selezionato
Set db = CurrentDb
Dim Data As Recordset

sql = "SELECT * " & _
      "FROM PROGETTODB_SCHEMA_LOTTO " & _
      "WHERE CODLOTTO = '" & CodLotto & "';"
Set Data = db.OpenRecordset(sql)
Data.MoveFirst

'Compila i vari campi in base al lotto selezionato
Me.Colore = Data.Fields!Colore
Me.Famiglia = Data.Fields!Famiglia
Me.Denominazione = Data.Fields!Denominazione
Me.Annata = Data.Fields!Annata
Me.DataOrigine = Data.Fields!DataOrigine
Me.AttoAlTaglio = Data.Fields!AttoAlTaglio

If Not IsNull(Data.Fields!Descrizione) Then
Me.Descrizione = Data.Fields!Descrizione
Else: Me.Descrizione = "-"
End If

'Riempie la lista con le lavorazioni effettuate sul lotto
sql = "SELECT LOTTO, PROCESSO, QUANTITAINIZIALE AS QTA_INIZIALE, " & _
      "QUANTITAFINALE AS QTA_FINALE, PERDITA, DATAINIZIO AS INIZIO, " & _
      "DATAFINE AS FINE " & _
      "FROM PROGETTODB_SCHEMA_LAVORAZIONE " & _
      "WHERE LOTTO = '" & CodLotto & "';"
Me.ElencoLavorazioni.RowSource = sql

Me.AggiungiLavorazione.Enabled = True

```

```

'Riempimento della sezione riguardante la fase
sql = "SELECT S.LOTTO, F.NOME, F.DESCRIZIONE, F.NUMEROPROGRESSIVO " & _
      "FROM PROGETTODB_SCHEMA_FASE F INNER JOIN " & _
      "PROGETTODB_SCHEMA_STATO S ON (F.CODFASE = S.FASE AND " & _
      "F.VINIFICAZIONE = S.VINIFICAZIONE)" & _
      "WHERE S.LOTTO = '" & CodLotto & "' " & _
      "ORDER BY F.NUMEROPROGRESSIVO;"
Set Data = db.OpenRecordset(sql)
If Data.EOF Then
    Me.FaseCorrente.Caption = "-"
    Me.InfoFaseCorrente.Caption = "-"
    Me.FaseSeccessiva.Caption = "-"
    Exit Sub
End If

Data.MoveFirst
Me.FaseCorrente.Caption = Data.Fields!Nome
Me.InfoFaseCorrente.Caption = Data.Fields!Descrizione

Data.MoveNext
If Data.EOF Then
    Me.FaseSeccessiva.Caption = "-"
    Exit Sub
End If

Me.FaseSeccessiva = Data.Fields!Nome
Me.PassaFaseSuccessiva.Enabled = True
Exit Sub

Exit_Lotto_AfterUpdate:
MsgBox "ERRORE: " & Err.Description
End Sub

Private Sub AggiungiLavorazione_Click()
'Apri il form InserimentoLAVORAZIONE passandogli come parametro il lotto selezionato
DoCmd.OpenForm "InserimentoLAVORAZIONE", , , , , , Me.Lotto
End Sub

Private Sub GoToFasi_Click()
DoCmd.OpenForm "GestioneFASI"
End Sub

Private Sub GoToProcessi_Click()
DoCmd.OpenForm "GestionePROCESSI"
End Sub

```

Si passa ora all'analisi della maschera dedicata alla gestione delle fasi.

Selezione il tipo di vinificazione: In bianco

Fasi della vinificazione:

CODFASE	NOME	NUM
AHJD	Fermentazione Alcoli	5
BVRW	Chiarifica	6

Nome: Chiarifica  
Progressivo: 6  
Descrizione:  
Si priva il vino di diverse sostanze, che lo portano a una maggiore limpidezza

Elimina Aggiungi fase

Si seleziona innanzitutto il tipo di vinificazione (per esempio, si supponga di selezionare la vinificazione in bianco): automaticamente si compila una lista con tutte le fasi associate a tale vinificazione. Selezionandone una è possibile visualizzarne il dettaglio.

I due pulsanti in basso a destra permettono o l'eliminazione della fase selezionata oppure l'aggiunta di una nuova fase. L'immagine rappresenta il comportamento della maschera una volta selezionata l'opzione di aggiunta di una nuova fase.

Selezione il tipo di vinificazione: In bianco

Fasi della vinificazione:

CODFASE	NOME	NUM
AHJD	Fermentazione Alcoli	5
BVRW	Chiarifica	6

Nome: Invecchiamento  
Progressivo: 10  
Descrizione:  
...

Elimina Aggiungi fase

Annulla Salva

```
Option Compare Database
Private db As DAO.Database
Private Attributo(2) As String
Private CodFase As String

Private Sub PopolaAttributi()
'Definizione dei campi potenzialmente modificabili
    Attributo(0) = "Nome"
    Attributo(1) = "Progressivo"
    Attributo(2) = "Descrizione"
End Sub
```

```

Private Sub Form_Load()
    sql = "SELECT TIPOVINIFICAZIONE FROM PROGETTODB_SCHEMA_VINIFICAZIONE"
    Me.Vinificazione.RowSource = sql

    PopolaAttributi
    Clear Me, Attributo, "-"
    ReadMode Me, Attributo

    Me.Vinificazione.Enabled = True
    Me.Vinificazione.SetFocus
    Me.FasiVinificazione.Enabled = True
    Me.FasiVinificazione.RowSource = ""

    Me.Salva.Visible = False
    Me.Annulla.Visible = False
    Me.Elimina.Enabled = False
    Me.Aggiungi.Enabled = False
End Sub

Private Sub Vinificazione_AfterUpdate()
    sql = "SELECT CODFASE, NOME, NUMEROPROGRESSIVO AS NUM " & _
        "FROM PROGETTODB_SCHEMA_FASE " & _
        "WHERE VINIFICAZIONE = '" & Me.Vinificazione & "' " & _
        "ORDER BY NUMEROPROGRESSIVO;"
    Me.FasiVinificazione.RowSource = sql

    Me.Aggiungi.Enabled = True
End Sub

Private Sub FasiVinificazione_Click()
    CodFase = Me.FasiVinificazione

    Set db = CurrentDb
    Dim Data As Recordset

    sql = "SELECT NOME, DESCRIZIONE, NUMEROPROGRESSIVO " & _
        "FROM PROGETTODB_SCHEMA_FASE " & _
        "WHERE CODFASE = '" & CodFase & "' AND " & _
        "VINIFICAZIONE = '" & Me.Vinificazione & "';"
    Set Data = db.OpenRecordset(sql)
    Data.MoveFirst

    'Compila i vari campi in base alla fase selezionata
    Me.Nome = Data.Fields!Nome
    Me.Descrizione = Data.Fields!Descrizione
    Me.Progressivo = Data.Fields!NumeroProgressivo

    Me.Elimina.Enabled = True
End Sub

Private Sub Aggiungi_Click()
    Clear Me, Attributo, ""
    WriteMode Me, Attributo

    CodFase = "XXXX"
    Me.Nome.SetFocus
    Me.Salva.Visible = True

```

```

    Me.Annulla.Visible = True
    Me.Elimina.Enabled = False
    Me.Vinificazione.Enabled = False
    Me.FasiVinificazione.Enabled = False
    Me.Aggiungi.Enabled = False
End Sub

Private Sub Salva_Click()
'Gestisce l'inserimento di una nuova fase per il tipo di vinificazione selezionato
    On Error GoTo Err_Salva_Click
    sql = "INSERT INTO PROGETTODB_SCHEMA_FASE (CODFASE, VINIFICAZIONE, NOME, " & _
        "DESCRIZIONE, NUMEROPROGRESSIVO) " & _
        "VALUES (" & CodFase & ", '" & _
            Me.Vinificazione & "', '" & _
            Me.Nome & "', '" & _
            Me.Descrizione & "', " & _
            Me.Progressivo & ");"

    DoCmd.RunSQL (sql)
    MsgBox "Inserimento effettuato con successo"
    Form_Load
Exit Sub

Err_Salva_Click:
    MsgBox "ERRORE: " & Err.Description
End Sub

Private Sub Annulla_Click()
    Form_Load
End Sub

```

L'ultima maschera associata a questo ruolo riguarda la gestione dei processi

Processi gestiti dalla cantina:

PROCESSO	DURATATIPICA	DESCRIZIONE
Solfitazione		Funzione antisettica
Diraspatura		Separazioni degli acini dal raspo
Fermentazione	10 gg	Gli zuccheri del mosto diventano alcol etilico
Invecchiamento	Variabile	Aumenta la qualità del vino

Prodotti utilizzati nel processo:

Nome: -  
 Categoria: -  
 Descrizione: -  
 Quantità: - Udm: -  
 Consigli per l'utilizzo: -

In essa sono specificati quali processi la cantina in questione può svolgere e, per ognuno di essi, che prodotti devono essere utilizzati. Per aggiungere un nuovo processo è sufficiente premere il tasto aggiungi, presente sotto la prima lista. Si apre la seguente schermata, il cui funzionamento è di immediata comprensione.

Nome:   
 Durata tipica:   
 Descrizione:   
 Annulla Inserisci

```
Option Compare Database
Private Attributo(2) As String
```

```
Private Sub Inserisci_Click()
'Gestisce l'inserimento di un nuovo processo
On Error GoTo Err_Inserisci_Click
sql = "INSERT INTO PROGETTODB_SCHEMA_PROCESSO(" & _
      "NOMEPROCESSO, DESCRIZIONE, DURATATIPICA) " & _
      "VALUES('" & Me.Nome & "', '" & Me.Descrizione
If Not IsNull(Me.DurataTipica) Then
  sql = sql & "', '" & Me.DurataTipica & " ');"
Else: sql = sql & "', NULL);"
End If
DoCmd.RunSQL sql
```

```

MsgBox "Inserimento effettuato con successo"
Exit Sub

Err_Inserisci_Click:
MsgBox "ERRORE: " & Err.Description
End Sub

Private Sub Annulla_Click()
DoCmd.Close
End Sub

```

Selezionando un processo dalla lista compare un pulsante che ne permette l'eliminazione; si popola inoltre una seconda lista con i prodotti da utilizzare in quel particolare processo. Per ogni prodotto è possibile vedere in che quantità deve essere utilizzato e dei consigli per l'utilizzo.

**GestionePROCESSI**

Processi gestiti dalla cantina:

PROCESSO	DURATA TIPICA	DESCRIZIONE
Solfitazione		Funzione antisettica
Diraspatura		Separazioni degli acini dal raspo
Fermentazione	10 gg	Gli zuccheri del mosto diventano alcol etilico
Invecchiamento	Variabile	Aumenta la qualità del vino

Elimina Aggiungi

Prodotti utilizzati nel processo:

CODICE	NOME
CFA785	Enoferm R
GER526	Acido Malico

Nome: Enoferm R  
 Categoria: Lievito  
 Descrizione: Lievito per mosti d'uva rossa  
 Quantità: 2000 UdM: g/hl

Consigli per l'utilizzo:  
 Per rapido avviamento della fermentazione dei mosti d'uva

Elimina Aggiungi

Come nel caso dei processi, anche per i prodotti è possibile sia l'inserimento (pulsante *Aggiungi*) che l'eliminazione (pulsante *Elimina*).

A seguire è riportato il codice VBA utilizzato per questa maschera.

```

Option Compare Database
Private db As DAO.Database
Dim NomeProcesso As String
Dim CodProdotto As String
Private Attributo(5) As String

Private Sub PopolaAttributi()
'Definizione dei campi potenzialmente modificabili
Attributo(0) = "Nome"
Attributo(1) = "Categoria"

```

```

    Attributo(2) = "Descrizione"
    Attributo(3) = "Quantita"
    Attributo(4) = "UdM"
    Attributo(5) = "ConsigliUtilizzo"
End Sub

Private Sub Form_Load()
    PopolaAttributi
    Clear Me, Attributo, "-"
    ReadMode Me, Attributo

    sql = "SELECT G.PROCESSO, P.DURATATIPICA, P.DESCRIZIONE " & _
        "FROM PROGETTODB_SCHEMA_GESTIONE G INNER JOIN " & _
        "PROGETTODB_SCHEMA_PROCESSO P ON G.PROCESSO = P.NOMEPROCESSO " & _
        "WHERE G.CANTINA = '01';"
    Me.ElencoProcessi.RowSource = sql
    Me.ElencoProcessi.Enabled = True
    Me.ElencoProcessi.SetFocus

    Me.ElencoProdotti.RowSource = ""
    Me.ElencoProdotti.Enabled = True
    Me.AggiungiProcesso.Enabled = True
    Me.EliminaProcesso.Visible = False
    Me.EliminaProdotto.Visible = False
    Me.AggiungiProdotto.Enabled = False
End Sub

Private Sub ElencoProcessi_Click()
    NomeProcesso = Me.ElencoProcessi

    sql = "SELECT P.CODPRODOTTO AS CODICE, P.NOME " & _
        "FROM PROGETTODB_SCHEMA_PRODOTTO P INNER JOIN " & _
        "PROGETTODB_SCHEMA_UTILIZZO U ON P.CODPRODOTTO = U.PRODOTTO " & _
        "WHERE U.PROCESSO = '" & NomeProcesso & "';"
    Me.ElencoProdotti.RowSource = sql

    Me.EliminaProcesso.Visible = True
    Me.EliminaProcesso.Enabled = True
    Me.AggiungiProdotto.Enabled = True
End Sub

Private Sub AggiungiProcesso_Click()
    'Permette l'aggiunta di un nuovo processo gestito dalla cantina in questione
    DoCmd.OpenForm "InserimentoPROCESSO"
End Sub

Private Sub EliminaProcesso_Click()
    'Permette di eliminare il processo selezionato da quelli gestiti dalla cantina
    sql = "DELETE FROM PROGETTODB_SCHEMA_GESTIONE " & _
        "WHERE G.CANTINA = '01' AND PROCESSO = '" & NomeProcesso & "';"
    DoCmd.RunSQL (sql)
    Form_Load
End Sub

Private Sub ElencoProdotti_Click()
    On Error GoTo Exit_ElencoProdotti_Click
    CodProdotto = Me.ElencoProdotti

```



```

Set db = CurrentDb
Dim Data As Recordset
sql = "SELECT * " & _
      "FROM PROGETTODB_SCHEMA_PRODOTTO " & _
      "WHERE CODPRODOTTO = '" & CodProdotto & "';"
Set Data = db.OpenRecordset(sql)
Data.MoveFirst

'Compila i campi relativi al prodotto selezionato
Me.Nome = Data.Fields!Nome
Me.Categoria = Data.Fields!Categoria
Me.Descrizione = Data.Fields!Descrizione
Me.ConsigliUtilizzo = Data.Fields!ConsigliUtilizzo

sql = "SELECT * " & _
      "FROM PROGETTODB_SCHEMA_UTILIZZO " & _
      "WHERE PRODOTTO = '" & CodProdotto & "' AND " & _
      "PROCESSO = '" & NomeProcesso & "';"
Set Data = db.OpenRecordset(sql)
Data.MoveFirst

'Compila i campi relativi all'utilizzo del prodotto
Me.Quantita = Data.Fields!Quantita
Me.UdM = Data.Fields!UnitaDiMisura

Me.EliminaProdotto.Visible = True
Exit Sub

Exit_ElencoProdotti_Click:
MsgBox "ERRORE: " & Err.Description
Form_Load
Exit Sub
End Sub

Private Sub AggiungiProdotto_Click()
'Aggiunge un nuovo prodotto legato al processo selezionato
Clear Me, Attributo, ""
WriteMode Me, Attributo
Me.Nome.SetFocus

Me.EliminaProdotto.Visible = False
Me.ElencoProcessi.Enabled = False
Me.EliminaProcesso.Enabled = False
Me.AggiungiProcesso.Enabled = False
End Sub

Private Sub EliminaProdotto_Click()
'Elimina l'associazione del prodotto al processo selezionato
sql = "DELETE FROM TABLE PROGETTODB_SCHEMA_UTILIZZO " & _
      "WHERE PRODOTTO = '" & CodProdotto & "' AND " & _
      "PROCESSO = '" & NomeProcesso & "';"
DoCmd.RunSQL (sql)
ElencoProcessi_Click
End Sub

```

## Analisi

Se l'utente che ha effettuato l'accesso ha come ruolo "Analisi", gli verrà mostrata la seguente maschera.

SettoreANALISI

La collina del Sole  
Settore ANALISI

Logout Utente: LUCAROSS  
Ruolo: ANALISI

Seleziona il lotto:  Cerca

Campioni registrati:

CODICE	PRELIEVO
--------	----------

Prelievo: Data: - Ora: - Descrizione: -

Risultato: Data: - Ora: -

Operatore: - Aggiungi campione

Parametri analizzati:

PARAMETRO	VALORE	UdM
-----------	--------	-----

Aggiungi parametro

Al solito si eviterà di parlare dell'intestazione in quanto analoga a quelle già presentate.

È innanzitutto necessario inserire il codice del lotto da analizzare. Come avveniva in un'altra maschera precedentemente descritta, tale inserimento può avvenire sia in modo diretto che attraverso il meccanismo di ricerca avanzata, attivabile premendo il pulsante *Cerca*.

Dopo aver selezionato il lotto, la lista si popola automaticamente con tutti i campioni prelevati dal lotto in questione. Selezionando uno di essi è possibile vederne le caratteristiche nel dettaglio.

SettoreANALISI

La collina del Sole  
Settore ANALISI

Logout Utente: LUCAROSS  
Ruolo: ANALISI

Seleziona il lotto: KDJNH78463 Cerca

Campioni registrati:

CODICE	PRELIEVO
FGARC12456	02/11/2010 16:02:11

Prelievo: Data: 02/11/2010 Ora: 16:02 Descrizione: -

Risultato: Data: 02/12/2010 Ora: 10:45

Operatore: LFMGS Aggiungi campione

Parametri analizzati:

PARAMETRO	VALORE	UdM
Acido Acetico	42	grammi/litro

Aggiungi parametro

Premendo il pulsante *Aggiungi campione* si ha la possibilità di memorizzare un nuovo campione prelevato dal lotto selezionato in alto.

Premendo invece su *Aggiungi parametro* si ha la possibilità di aggiungere la misurazione di un parametro ai dati associati a un determinato campione. Tale operazione è possibile attraverso la seguente maschera.

**Option** Compare Database

**Private** CodCampione As String

**Private Sub** Form\_Load()

sql = "SELECT NOMEPARAMETRO FROM PROGETTODB\_SCHEMA\_PARAMETRO;"

Me.Nome.RowSource = sql

CodCampione = Me.OpenArgs

**End Sub**

**Private Sub** Inserisci\_Click()

'Gestisce l'inserimento di un nuovo parametro per un determinato campione

On Error GoTo Err\_Inserisci\_Click

sql = "INSERT INTO PROGETTODB\_SCHEMA\_ANALISI (CAMPIONE, PARAMETRO, " & \_  
"STRUMENTODIMISURA, VALORE, UNITADIMISURA) " & \_  
"VALUES ('" & CodCampione & "', '" & \_

```

        Me.Nome & ", " & _
        Me.Strumento & ", " & _
        Me.Valore & ", " & _
        Me.UdM & "');"

DoCmd.RunSQL (sql)
MsgBox "Inserimento eseguito con successo"
DoCmd.Close
Exit Sub

Err_Inserisci_Click:
MsgBox "ERRORE: " & Err.Description
End Sub

Private Sub Annulla_Click()
DoCmd.Close
End Sub

```

Qui sotto è invece riportato il codice che determina le funzionalità della maschera adibita alla gestione delle analisi.

```

Option Compare Database
Private db As DAO.Database
Private Attributo(5) As String
Private CodCampione As String

Private Sub PopolaAttributi()
'Definizione dei campi potenzialmente modificabili
Attributo(0) = "Operatore"
Attributo(1) = "DataPrelievo"
Attributo(2) = "OraPrelievo"
Attributo(3) = "DataRisultati"
Attributo(4) = "OraRisultati"
Attributo(5) = "Descrizione"
End Sub

Private Sub Form_Load()
PopolaAttributi
Clear Me, Attributo, "-"
ReadMode Me, Attributo

sql = "SELECT CODLOTTO FROM PROGETTODB_SCHEMA_LOTTO ORDER BY CODLOTTO;"
Me.Lotto.RowSource = sql
Me.Lotto.Enabled = True
Me.Lotto = ""
Me.Lotto.SetFocus

Me.ElencoCampioni.Enabled = True
Me.ElencoCampioni.RowSource = ""
Me.ElencoParametri.Enabled = True
Me.ElencoParametri.RowSource = ""
Me.Annulla.Visible = False
Me.Salva.Visible = False
Me.AggiungiCampione.Enabled = False
Me.AggiungiParametro.Enabled = False
End Sub

```

```

Private Sub Cerca_Click()
'Abilita la ricerca avanzata di un lotto
    DoCmd.OpenForm "RicercaLOTTO"
End Sub

Private Sub Lotto_AfterUpdate()
    sql = "SELECT CODCAMPIONE AS CODICE, DATAEORAPRELIEVO AS PRELIEVO " & _
        "FROM PROGETTODB_SCHEMA_CAMPIONE " & _
        "WHERE LOTTO = '" & Me.Lotto & "' " & _
        "ORDER BY DATAEORAPRELIEVO DESC;"
    Me.ElencoCampioni.RowSource = sql

    Me.AggiungiCampione.Enabled = True
End Sub

Private Sub ElencoCampioni_Click()
    CodCampione = Me.ElencoCampioni

    'Riempimento dei campi associati al campione selezionato
    Set db = CurrentDb
    Dim Data As Recordset

    sql = "SELECT MATRICOLAOPERATORE, DATAEORAPRELIEVO, DATAEORARISULTATI, DESCRIZIONE " & _
        "FROM PROGETTODB_SCHEMA_CAMPIONE " & _
        "WHERE CODCAMPIONE = '" & CodCampione & "';"

    Set Data = db.OpenRecordset(sql)
    Data.MoveFirst

    Me.Operatore = Data.Fields!MATRICOLAOPERATORE
    Me.DataPrelievo = Format(Data.Fields!DATAEORAPRELIEVO, "dd/mm/yyyy")
    Me.OraPrelievo = Format(Data.Fields!DATAEORAPRELIEVO, "hh:mm")
    Me.DataRisultati = Format(Data.Fields!DATAEORARISULTATI, "dd/mm/yyyy")
    Me.OraRisultati = Format(Data.Fields!DATAEORARISULTATI, "hh:mm")

    'Riempimento della lista dei parametri associati al campione
    sql = "SELECT PARAMETRO, VALORE, UNITADIMISURA AS Udm " & _
        "FROM PROGETTODB_SCHEMA_ANALISI " & _
        "WHERE CAMPIONE = '" & CodCampione & "';"
    Me.ElencoParametri.RowSource = sql

    Me.AggiungiParametro.Enabled = True
End Sub

Private Sub AggiungiCampione_Click()
'Permette l'inserimento di un nuovo campione legato al lotto selezionato
    Clear Me, Attributo, ""
    WriteMode Me, Attributo
    Me.Operatore.SetFocus

    Me.Annulla.Visible = True
    Me.Salva.Visible = True
    Me.Lotto.Enabled = False
    Me.ElencoCampioni.Enabled = False
    Me.ElencoParametri.Enabled = False
    Me.ElencoParametri.RowSource = ""

```

```

    Me.AggiungiCampione.Enabled = False
    Me.AggiungiParametro.Enabled = False
End Sub

Private Sub Salva_Click()
'Gestisce l'inserimento di un nuovo campione
    On Error GoTo Err_Salva_Click
    CodCampione = "XXXXXXXXXX"

    sql = "INSERT INTO PROGETTODB_SCHEMA_CAMPIONE(CODCAMPIONE, MATRICOLAOPERATORE, " & _
        "DATAEORAPRELIEVO, DATAEORARISULTATI, DESCRIZIONE, LOTTO) " & _
        "VALUES ('" & CodCampione & "', '" & _
            Me.Operatore & "', #" & _
            Format(Me.DataPrelievo & " " & _
                Me.OraPrelievo, "dd/mm/yyyy hh:mm:ss") & "#, #" & _
            Format(Me.DataRisultati & " " & _
                Me.OraRisultati, "dd/mm/yyyy hh:mm:ss") & "#, '" & _
            Me.Descrizione & "', '" & _
            Me.Lotto & "');"

    DoCmd.RunSQL (sql)
    MsgBox "Inserimento effettuato con successo"
    Form_Load
Exit Sub

Err_Salva_Click:
    MsgBox "ERRORE: " & Err.Description
End Sub

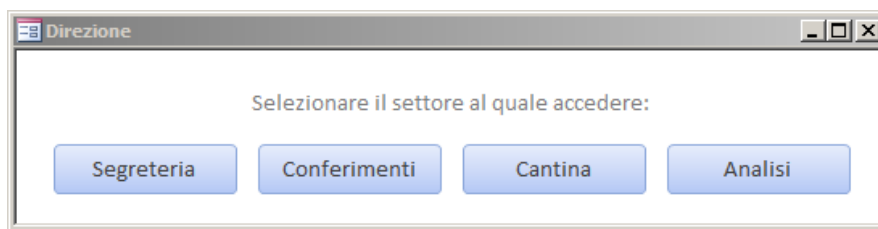
Private Sub Annulla_Click()
    Form_Load
End Sub

Private Sub AggiungiParametro_Click()
'Apri il form InserimentoPARAMETRO passandogli il codice campione
    DoCmd.OpenForm "InserimentoPARAMETRO", , , , , CodCampione
End Sub

```

## Amministrazione

Se l'utente che ha effettuato l'accesso ha come ruolo "Amministrazione", gli verrà mostrata la seguente maschera.



Selezionando il settore l'amministratore assumerà automaticamente i privilegi concessi al determinato ruolo e potrà usufruire delle varie funzionalità che sono state espone fino ad ora.

**Option** Compare Database

```
Private Sub Analisi_Click()  
'Apre il form dedicato al settore analisi  
    DoCmd.OpenForm "SettoreANALISI"  
End Sub  
  
Private Sub Cantina_Click()  
'Apre il form dedicato al settore cantina  
    DoCmd.OpenForm "SettoreCANTINA"  
End Sub  
  
Private Sub Conferimenti_Click()  
'Apre il form dedicato al settore conferimenti  
    DoCmd.OpenForm "SettoreCONFERIMENTI"  
End Sub  
  
Private Sub Segreteria_Click()  
'Apre il form dedicato al settore segreteria  
    DoCmd.OpenForm "SettoreSEGRETERIA"  
End Sub
```

Non è evidentemente necessario spiegare una seconda volta tutte le funzionalità cui l'amministratore ha accesso, proprio perché sono state appena espone nei paragrafi precedenti.

## Osservazioni

Come si nota fin dalla prima schermata presentata, l'applicazione è riferita a una singola cantina, "La collina del Sole". Questo non significa però che sia stata perso il concetto della distribuzione: l'applicazione richiede infatti una configurazione iniziale nella quale bisogna specificare la cantina di appartenenza.

Da una lettura attenta del codice risulta inoltre evidente come molto spesso vengano chiamati dei metodi che non sono presenti nei moduli fino ad ora presentati. Questo perché è stato creato un modulo esterno contenente alcune funzioni ricorrenti. Il codice contenuto nel modulo è il seguente.

**Option** Compare Database

'Riempie con la stringa c i campi del form frm specificati in attr()

**Public Sub** Clear(frm As Form, attr() As String, c As String)

**For Each** a In attr

        frm(a) = c

**Next**

**End Sub**

'Rende modificabili i campi del form frm specificati in attr()

**Public Sub** WriteMode(frm As Form, attr() As String)

**For Each** a In attr

        frm(a).Locked = **False**

        frm(a).BorderStyle = 1

**Next**

**End Sub**

'Rende NON modificabili i campi del form frm specificati in attr()

**Public Sub** ReadMode(frm As Form, attr() As String)

**For Each** a In attr

        frm(a).Locked = **True**

        frm(a).BorderStyle = 0

**Next**

**End Sub**



## Query

In questo paragrafo saranno presentate e risolte alcune interrogazioni. La trattazione sarà divisa tra query elementari e query complesse, alle quali sarà dedicato maggior spazio.

### Query elementari

#### Query 1

Determinare tutte le informazioni (Codice, Tipologia, Capienza) riguardanti le vasche maggiormente usate per lo stoccaggio dei lotti all'interno della cantina "L'antica vite".

```
CREATE VIEW UtilizziVascheAnticaVite(CodVasca, NumUtilizzi) AS (  
    SELECT S.Vasca, COUNT(*)  
    FROM Stoccaggio S INNER JOIN  
        Cantina C ON S.Cantina = C.CodCantina  
    WHERE C.Nome = 'L'Antica Vite'  
    GROUP BY S.Vasca  
);  
  
CREATE VIEW MaxUtilizzoVasche(NumMaxUtilizzi) AS (  
    SELECT MAX(V1.NumUtilizzi)  
    FROM UtilizziVascheAnticaVite V1  
);  
  
SELECT V1.CodVasca AS Codice, V1.Tipologia, V1.Capienza  
FROM UtilizziVascheAnticaVite V1 INNER JOIN  
    Vasca V ON V1.CodVasca = V.CodVasca  
WHERE V1.NumUtilizzi = (SELECT V2.NumMaxUtilizzi FROM MaxUtilizzoVasche V2);
```

L'interrogazione restituisce il seguente risultato.

Codice	Tipologia	Capienza
50	Acciaio	200
53	Cemento	500

#### Query 2

Elenco { Codice\_Vigneto, Estensione\_Vigneto, Anno\_Impianto\_Vigneto, Uva\_Scelta } per tutti i vigneti associati al produttore "AS93". La scelta vendemmiale deve essere riferita all'anno corrente.

```
SELECT V.CodVigneto AS Codice_Vigneto, V.Estensione AS Estensione_Vigneto,  
    V.AnnoImpianto AS Anno_Impianto_Vigneto, SV.CodUva AS Uva_Scelta  
FROM Vigneto V INNER JOIN  
    SceltaVendemmiale SV ON V.CodVigneto = SV.CodVigneto  
WHERE SV.Anno = EXTRACT(YEAR FROM CURRENT_DATE) AND V.CodProduttore = 'AS93';
```

L'interrogazione restituisce il seguente risultato.

Codice_Vigneto	Estensione_Vigneto	Anno_Impianto_Vigneto	Uva_scelta
PLIC	4	1964	145

### Query 3

Elenco { Nome, Descrizione, NumeroProgressivo } delle fasi che il lotto “KDJNH78463” ha attraversato fino ad ora.

```
SELECT F.Nome, F.Descrizione, F.NumeroProgressivo
FROM Fase F INNER JOIN
    Stato S ON (F.CodFase = S.Fase AND F.Vinificazione = S.Vinificazione)
WHERE S.Lotto = 'MHNCD45715'
ORDER BY F.NumeroProgressivo;
```

L'interrogazione restituisce il seguente risultato.

Nome	Descrizione	NumeroProgressivo
Fermentazione alcolica	Forma di metabolismo energetico che permette la trasformazione del mosto in vino.	5
Chiarifica	Si priva il vino di diverse sostanze, così da portarlo a una maggiore limpidezza.	6

### Query 4

Elenco { CodProdotto, Nome } dei prodotti utilizzati durante il processo di solfitazione.

```
SELECT P.CodProdotto, P.Nome
FROM Prodotto P INNER JOIN
    Utilizzo U ON P.CodProdotto = U.Prodotto
WHERE U.Processo = 'Solfitazione';
```

L'interrogazione restituisce il seguente risultato.

CodProdotto	Nome
CFA785	Enoferm R
GER526	Acido Malico

### Query 5

Elenco { Nome\_Processo, Descrizione\_Processo } dei processi che fanno uso del prodotto Acido Malico

```
SELECT Pc.NomeProcesso AS Nome_Processo, Pc.Descrizione AS Descrizione_Processo
FROM Processo Pc INNER JOIN
    Utilizzo U ON Pc.NomeProcesso = U.Processo INNER JOIN
    Prodotto Pd ON U.Prodotto = Pd.CodProdotto
WHERE Pd.Nome = 'Acido Malico';
```

L'interrogazione produce il seguente risultato.

Nome_Processo	Descrizione_Processo
Fermentazione	Gli zuccheri del mosto diventano alcol etilico
Invecchiamento	Aumenta la qualità del vino

## Query 6

Elenco { Nome\_Produttore, Cognome\_Produttore, Comune\_Nascita } dei produttori vittoriosi che sono stati almeno una volta affiliati alla cantina di Vittorio Veneto.

```
SELECT P.Nome AS Nome_Produttore, P.Cognome AS Cognome_Produttore, P.ComuneNascita
FROM Produttore P INNER JOIN
    AffiliazionePassata A ON P.CodProduttore = A.CodProduttore INNER JOIN
    Cantina C1 ON A.CodCantina = C1.CodCantina INNER JOIN
    Cantina C2 ON P.CantinaAffiliazione = C2.CodCantina
WHERE P.Residenza = 'Vittorio Veneto' AND
    (C2.Nome = 'Cantina Vittorio Veneto' OR C1.Nome = 'Cantina Vittorio Veneto');
```

L'interrogazione produce il seguente risultato.

Nome_Produttore	Cognome_Produttore	ComuneNascita
Marco	Bisaglia	Oderzo
Federica	Spina	Mestre
Toni	Grinton	Vittorio Veneto

## Query 7

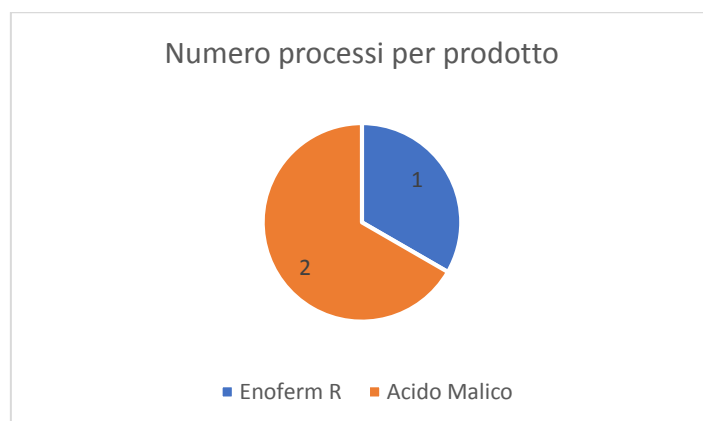
Contare, per ogni prodotto presente nella base di dati, il numero di processi che lo impiega.

```
SELECT Pd.Nome AS Nome_Prodotto, COUNT(U.Processo) AS Num_Processi
FROM Prodotto Pd INNER JOIN
    Utilizzo U ON Pd.CodProdotto = U.Prodotto
GROUP BY Pd.Nome;
```

L'interrogazione produce il seguente risultato.

Nome_Prodotto	Num_Processi
Enoferm R	1
Acido Malico	2

È possibile rappresentare il risultato dell'interrogazione in un grafico. Essendo in questo caso di esempio la mole di dati non importante, esso potrebbe non apparire significativo; anche se in realtà evidentemente lo è.



## Query 8

Elenco { Codice\_Lotto, Famiglia, Denominazione } dei lotti da cui sono stati estratti dei campioni per i quali è stato misurata la quantità di Glicerolo.

```
SELECT L.CodLotto AS Codice_Lotto, L.Famiglia, L.Denominazione
FROM Lotto L INNER JOIN
    Campione C ON L.CodLotto = C.Lotto INNER JOIN
    Analisi A ON C.CodCampione = A.Campione
WHERE Parametro = 'Glicerolo';
```

L'interrogazione restituisce il seguente risultato.

Codice_Lotto	Famiglia	Denominazione
KDJNH78463	Prosecco	DOC
MHNCD45715	Merlot	IGT
LKCSR16425	Chardonnay	VDT

## Query 9

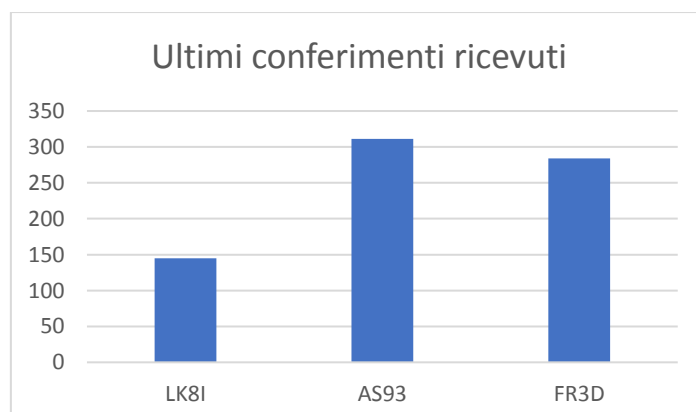
Elenco { CodProduttore, Uva, Quantità, DataEOra } degli ultimi carichi conferiti. Il risultato dovrà essere in ordine decrescente rispetto alla data di conferimento.

```
SELECT P.CodProduttore, C.Uva, C.Quantita, C.DataEOra
FROM Prenotazione P INNER JOIN
    Carico C ON P.CodPrenotazione = C.Prenotazione
ORDER BY C.DataEOra DESC;
```

L'interrogazione restituisce il seguente risultato.

CodProduttore	Uva	Quantita	DataEOra
LK8I	145	75	01-GEN-13
AS93	311	100	26-AGO-11
FR3D	284	78	26-AGO-11

Il risultato può essere riassunto in un grafico significativo. Quello che segue è stato realizzato in MS Access usando la query appena presentata.



## Query 10

Determinare { Nome\_Produttore, Cognome\_Produttore, CantinaAffiliazione } dei produttori che hanno conferito, durante il 2011, la maggior quantità di Prosecco.

```
CREATE VIEW QuantitaProseccoConferita2011 (CodProduttore, Quantita) AS (  
    SELECT P.CodProduttore, SUM( C.Quantita)  
    FROM Carico C INNER JOIN  
        Prenotazione P ON C.Prenotazione = P.CodPrenotazione INNER JOIN  
        Uva U ON C.Uva = U.CodUva  
    WHERE U.Nome = 'Prosecco' AND  
        C.DataEOra >= TO_DATE('01/01/2011', 'dd/mm/yyyy') AND  
        C.DataEOra < TO_DATE('01/01/2012', 'dd/mm/yyyy')  
    GROUP BY P.CodProduttore  
);  
  
CREATE VIEW QuantitaMaxConferita2011(QuantitaMax) AS (  
    SELECT MAX(V1.Quantita)  
    FROM QuantitaProseccoConferita2011 V1  
);  
  
SELECT P.Nome AS Nome_Produttore, P.Cognome AS Cognome_Produttore, P.CantinaAffiliazione  
FROM Produttore P INNER JOIN  
    QuantitaProseccoConferita2011 V1 ON P.CodProduttore = V1.CodProduttore  
WHERE V1.Quantita = (SELECT V2.QuantitaMax FROM QuantitaMaxConferita2011 V2);
```

L'interrogazione restituisce il seguente risultato.

Nome_Produttore	Cognome_Produttore	CantinaAffiliazione
Mario	Rossi	03

## Query complesse

### Query 1

Elenco {Nome, Cognome, Data Affiliazione} dei produttori senza affiliazioni passate che hanno già effettuato almeno 3 prenotazioni.

```
CREATE VIEW ProdPass(CodProduttore) AS (  
    SELECT CodProduttore  
    FROM AffiliazionePassata  
);  
  
CREATE VIEW ProdConNumPren(CodProduttore, NumPrenotazioni) AS (  
    SELECT P.CodProduttore, COUNT(P.CodPrenotazione)  
    FROM Prenotazione P  
    GROUP BY P.CodProduttore  
);  
  
SELECT P.Nome AS Nome_Produttore, P.Cognome AS Cognome_Produttore,  
       P.DataInizioAffiliazione AS Data_Inizio_Affiliazione  
FROM Produttore P INNER JOIN ProdConNumPren PCNP ON P.CodProduttore=PCNP.CodProduttore  
WHERE PCNP.NumPrenotazioni>=3 AND P.CodProduttore NOT IN (SELECT PP.CodProduttore  
                                                           FROM ProdPass PP);
```

L'interrogazione restituisce il seguente risultato.

Nome_Produttore	Cognome_Produttore	Data_Inizio_Affiliazione
Matteo	Bevilacqua	10-FEB-13
Matteo	Dal Cin	01-GEN-10

### Query 2

Elenco {Nome, Cognome, Cantina Affiliazione} dei produttori di residenza vittoriese che hanno scelto di coltivare il Prosecco DOCG almeno due volte in uno stesso vigneto.

```
CREATE VIEW ProdConNumProsDOCG(CodProduttore, NumVolte) AS (  
    SELECT V.CodProduttore, COUNT(U.CodUva)  
    FROM Vigneto V INNER JOIN SceltaVendemmiale SV ON V.CodVigneto=SV.CodVigneto  
    INNER JOIN Uva U ON SV.CodUva=U.CodUva  
    WHERE U.Nome='Prosecco' AND U.Denominazione='DOCG'  
    GROUP BY V.CodProduttore, V.CodVigneto  
);  
  
SELECT P.Nome AS Nome_Produttore, P.Cognome AS Cognome_Produttore,  
       P.CantinaAffiliazione AS Cantina_Affiliazione  
FROM Produttore P INNER JOIN ProdConNumProsDOCG P2 ON P.CodProduttore=P2.CodProduttore  
WHERE P.Residenza='Vittorio Veneto';
```

L'interrogazione restituisce il seguente risultato.

Nome_Produttore	Cognome_Produttore	Cantina_Affiliazione
Toni	Grinton	01

### Query 3

Determinare la disponibilità totale e la disponibilità rimanente di uva che una determinata cantina può ricevere in una determinata data (si intende ottenere una tabella con le colonne { Nome Cantina, Data, Totale, Rimanente }).

```
CREATE VIEW SumQuanCant (CodCantina, Data, Quantita) AS (
    SELECT Pd.CantinaAffiliazione, TO_DATE(Co.DataEOra, 'dd/mm/yyyy'), sum(Co.Quantita)
    FROM    Produttore Pd INNER JOIN Prenotazione Pn ON Pd.CodProduttore=Pn.CodProduttore
           INNER JOIN Carico      Co ON Co.Prenotazione=Pn.CodPrenotazione
           INNER JOIN Cantina     Ca ON Ca.CodCantina=Pd.CantinaAffiliazione
    GROUP BY PD.CantinaAffiliazione, TO_DATE(Co.DataEOra, 'dd/mm/yyyy')

UNION

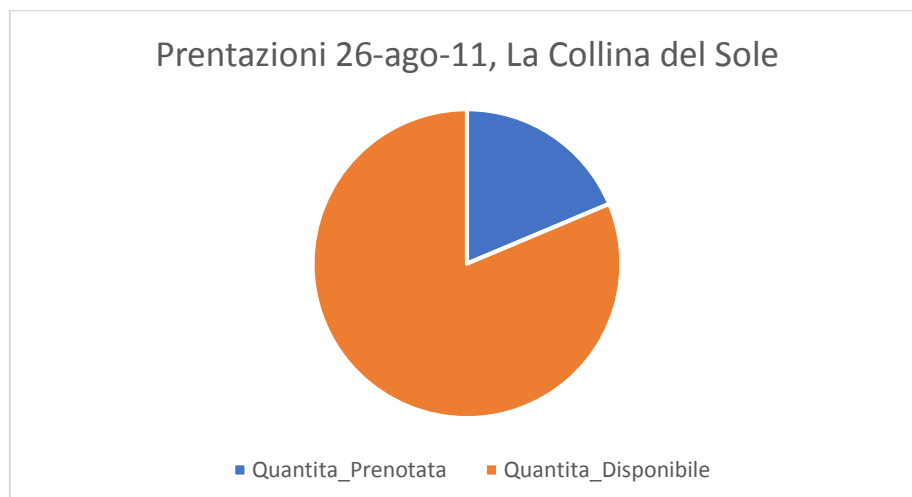
    SELECT Ca.CodCantina, TO_DATE(Co.DataEOra, 'dd/mm/yyyy'), sum(Co.Quantita)
    FROM    Cantina Ca INNER JOIN AffiliazionePassata AP ON Ca.CodCantina=AP.CodCantina
           INNER JOIN Produttore Pd ON AP.CodProduttore=Pd.CodProduttore
           INNER JOIN Prenotazione Pn ON Pd.CodProduttore=Pn.CodProduttore
           INNER JOIN Carico      Co ON Co.Prenotazione=Pn.CodPrenotazione
    GROUP BY CA.CodCantina, TO_DATE(CO.DataEOra, 'dd/mm/yyyy')
);

SELECT C.Nome AS Nome_Cantina, SQC.Data AS Data,
       C.QuantitaConferimentoMax AS Quantita_Totale,
       C.QuantitaConferimentoMax-SQC.Quantita AS Quantita_Disponibile
FROM    SumQuanCant SQC INNER JOIN Cantina C ON SQC.CodCantina=C.CodCantina
```

L'interrogazione restituisce il seguente risultato.

Nome_Cantina	Data	Quantita_Totale	Quantita_Disponibile
Cantina Vittorio Veneto	26-AGO-11	5.000.000	4.999.922
Cantina Vittorio Veneto	01-GEN-13	5.000.000	4.999.925
La Collina del Sole	26-AGO-11	3.000.000	2.439.900

Per esempio, il seguente grafico, creato con MS Access, mostra la situazione delle prenotazioni della cantina "La collina del Sole" per il giorno 26 Agosto 2011. Per realizzarlo è stata usata la query appena presentata.



#### Query 4

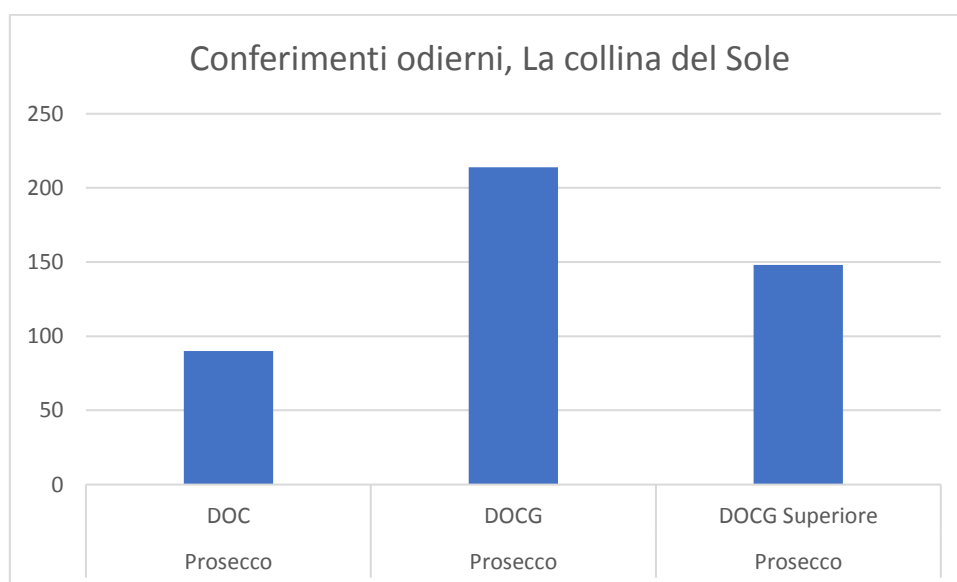
Elenco {Nome, Denominazione, Quantità} dell'uva conferita in una determinata giornata (quella odierna) in una determinata cantina.

```
CREATE VIEW QuantitaOdiernaUvaConferita(CodUva, Quantita) AS (  
    SELECT U.CodUva, sum(C.Quantita)  
    FROM Carico C INNER JOIN Uva U ON C.Uva=U.CodUva  
    WHERE TO_DATE(C.DataEOra, 'dd/mm/yyyy')=TO_DATE(CURRENT_DATE, 'dd/mm/yyyy')  
    GROUP BY U.CodUva  
);  
  
SELECT U.Nome AS Nome_Uva, U.Denominazione AS Denominazione_Uva,  
    QOUC.Quantita AS Quantita_Odierna  
FROM Uva U INNER JOIN QuantitaOdiernaUvaConferita QOUC ON U.CodUva=QOUC.CodUva;
```

L'interrogazione restituisce il seguente risultato.

Nome_Uva	Denominazione_Uva	Quantita_Odierna
Prosecco	DOC	90
Prosecco	DOCG	214
Prosecco	DOCG Superiore	148

Il risultato di questa interrogazione può essere efficacemente riassunto in un grafico. Quello che segue è stato realizzato in MS Access utilizzando la query appena presentata.





## Query 5

Elenco {Nome, IndirizzoStabilimento} delle cantine, aventi almeno due vasche, che sono state delle cantine di arrivo per almeno un lotto finalizzato al vino bianco prosecco DOC.

```
CREATE VIEW CantinaConNumVasche (CodCantina, NumVasche) AS (  
    SELECT C.CodCantina, sum(V.CodVasca)  
    FROM   Cantina C INNER JOIN Vasca V ON C.CodCantina=V.CodVasca  
    GROUP BY C.CodCantina  
);  
  
CREATE VIEW CantinaPartenzaProseccoDOC (CodCantina) AS (  
    SELECT T.CantinaPartenza  
    FROM   Vino V INNER JOIN Finalita F      ON V.CodVino=F.Vino  
           INNER JOIN Lotto L        ON F.Lotto=L.CodLotto  
           INNER JOIN Trasferimento T ON L.CodLotto=T.Lotto  
    WHERE  V.Colore='Bianco' AND V.famiglia='Prosecco' AND V.Denominazione='DOC'  
);  
  
SELECT C.Nome, C.IndirizzoStabilimento AS  
FROM   Cantina C INNER JOIN CantinaConNumVasche C1 ON C.CodCantina=C1.CodCantina  
WHERE  C1.NumVasche>=2 AND C.CodCantina IN (SELECT C2.CodCantina  
                                           FROM   CantinaPartenzaProseccoDOC C2);
```

L'interrogazione restituisce il seguente risultato.

Nome	IndirizzoStabilimento
Cantina Vittorio Veneto	Via Canton 1 TV
L'Antica Vite	Via Da Vinci 65 VI

## Query 6

Elenco {Annata, Denominazione, Famiglia, Colore, Descrizione} di tutti i lotti stoccati per un periodo di tempo superiore alla media dei giorni di stoccaggio dei lotti della cantina “Cantina Vittorio Veneto”.

```
CREATE VIEW GiorniStoccaggio (CodLotto, NumGiorni) AS (  
    SELECT Lotto, TO_DATE(DataEOraFine, 'dd/mm/yyyy') -  
        TO_DATE(DataEOraInizio, 'dd/mm/yyyy')  
    FROM Stoccaggio  
    WHERE DataEOraFine is not null  
);  
  
CREATE VIEW MediaStoccaggioCantinaVittorio (MediaStoccaggio) AS (  
    SELECT avg(GS.NumGiorni)  
    FROM GiorniStoccaggio GS INNER JOIN Lotto L ON GS.CodLotto=L.CodLotto  
        INNER JOIN Stoccaggio S ON L.CodLotto=S.Lotto  
        INNER JOIN Cantina C ON S.Cantina=C.CodCantina  
    WHERE C.Nome='Cantina Vittorio Veneto'  
);  
  
SELECT L.Annata AS Annata, L.Denominazione AS Denominazione, L.Famiglia AS Famiglia,  
    L.Colore AS Colore, L.Descrizione AS Descrizione  
FROM Lotto L INNER JOIN GiorniStoccaggio GS ON L.CodLotto=GS.CodLotto  
WHERE GS.NumGiorni > (SELECT MediaStoccaggio  
    FROM MediaStoccaggioCantinaVittorio);
```

L'interrogazione restituisce il seguente risultato.

Annata	Denominazione	Famiglia	Colore	Descrizione
2011	DOC	Prosecco	Bianco	-

## Query 7

Elenco {Nome, IndirizzoStabilimento} di tutte le cantine che hanno accumulato nel 2018 uno scarto complessivo di “Raspi” superiore al 10% della media relativa al 2012 degli scarti accumulati “Feccia Solida” della “Cantina Vittorio Veneto”.

```
CREATE VIEW ScartoMedioFecciaSolidaVV2012 (valore_medio) AS (  
    SELECT 0.1*avg(Quantita)  
    FROM ScartoAccumulato SA INNER JOIN Cantina C ON SA.Cantina=C.CodCantina  
    WHERE TipoScarto='Feccia Solida' AND Data>=TO_DATE('01/01/2012', 'dd/mm/yyyy') AND  
          Data<=TO_DATE('31/12/2012', 'dd/mm/yyyy') AND C.Nome='Cantina Vittorio Veneto'  
);  
  
CREATE VIEW ScartoAccumulato2018 (Cantina, Quantita) AS (  
    SELECT Cantina, sum(Quantita)  
    FROM ScartoAccumulato  
    WHERE TipoScarto='Raspi' AND Data>=TO_DATE('01/01/2018', 'dd/mm/yyyy') AND  
          Data<=TO_DATE('31/12/2018', 'dd/mm/yyyy')  
    GROUP BY Cantina  
);  
  
SELECT C.Nome, C.IndirizzoStabilimento  
FROM Cantina C INNER JOIN ScartoAccumulato2018 SC ON C.CodCantina=SC.Cantina  
WHERE SC.Quantita>(SELECT SMFS.valore_medio  
                   FROM ScartoMedioFecciaSolidaVV2012 SMFS);
```

L'interrogazione restituisce il seguente risultato.

Nome	IndirizzoStabilimento
Cantina Vittorio Veneto	Via Canton 1 TV
L'Antica Vite	Via Da Vinci 65 VI
La Collina Del Sole	Via Europa 5 VR

## Query 8

Elenco {Nome, Descrizione} di tutti i parametri che sono stati analizzati nel 2015 almeno il 20% delle volte in cui è stato analizzato il parametro 'Acido Acetico' nel biennio 2012-2013.

```
CREATE VIEW NumAnalisiAcidoAcetico (TotAnalisiAA) AS (  
    SELECT 0.2*COUNT(Campione)  
    FROM   Analisi A INNER JOIN Campione C ON A.Campione=C.CodCampione  
    WHERE  Parametro='Acido Acetico' AND  
           C.DataEOraPrelievo>=TO_DATE('01/01/2012', 'dd/mm/yyyy') AND  
           C.DataEOraPrelievo<=TO_DATE('31/12/2013', 'dd/mm/yyyy')  
);  
  
CREATE VIEW NumAnalisiParametri (Parametro, TotAnalisi) AS (  
    SELECT A.Parametro, count(C.CodCampione)  
    FROM   Analisi A INNER JOIN Campione C ON A.Campione=C.CodCampione  
    WHERE  DataEOraPrelievo>=TO_DATE('01/01/2015', 'dd/mm/yyyy') AND  
           C.DataEOraPrelievo<=TO_DATE('31/12/2015', 'dd/mm/yyyy')  
    GROUP BY A.Parametro  
);  
  
SELECT P.NomeParametro AS Parametro, P.Descrizione  
FROM   Parametro P INNER JOIN NumAnalisiParametri NAP ON P.NomeParametro=NAP.Parametro  
WHERE  NAP.TotAnalisi>= (SELECT TotAnalisiAA  
                        FROM   NumAnalisiAcidoAcetico);
```

L'interrogazione restituisce il seguente risultato:

Parametro	Descrizione
Grado Alcolico	Indice del grado alcolico del mosto
Glicerolo	Deve essere dentro un range per dare un senso di rotondità al vino

## Query 9

Elenco {Nome, Categoria, Descrizione} di tutti i prodotti che sono impiegati in almeno tre processi gestiti da almeno quattro cantine.

```
CREATE VIEW ProcessiDaAlmenoQuattroCantine (Processo, NumCantine) AS (  
    SELECT Processo, count(Cantina)  
    FROM Gestione  
    GROUP BY Processo  
);  
  
CREATE VIEW ProdottiInNumProcessi (Prodotto, NumProcessi) AS (  
    SELECT U.Prodotto, count(U.Processo)  
    FROM Utilizzo U INNER JOIN ProcessiDaAlmenoQuattroCantine PDAQC  
        ON U.Processo=PDAQC.Processo  
    WHERE PDAQC.NumCantine>=4  
    GROUP BY U.Prodotto  
);  
  
SELECT P.Nome, P.Categoria, P.Descrizione  
FROM Prodotto P INNER JOIN ProdottiInNumProcessi PINP ON P.CodProdotto=PINP.Prodotto  
WHERE PINP.NumProcessi>=3;
```

L'interrogazione produce il seguente risultato.

Nome	Categoria	Descrizione
Polylact	Chiarificante	Chiarificante preventivo/curativo
Enoferm R	Lievito	Lievito per mosti d'uva rossa
Acido Malico	Attivante	Regolatore del pH

## Query 10

Elenco {CodLotto, Annata, Denominazione, Famiglia, Colore} di tutti i lotti atti al taglio tali che hanno subito una perdita media nelle lavorazioni inferiore al 20% della media di tutte le perdite relative alle lavorazioni nell'anno 2012

```
CREATE VIEW LottiAttiAlTaglio (CodLotto) AS (  
    SELECT CodLotto  
    FROM Lotto  
    WHERE AttoAlTaglio='True'  
);  
  
CREATE VIEW MediaPerdite2012 (media_perdite) AS (  
    SELECT 0.2*avg(Perdita)  
    FROM Lavorazione  
    WHERE DataInizio>=TO_DATE('01/01/2012', 'dd/mm/yyyy') AND  
           DataFine<=TO_DATE('31/12/2012', 'dd/mm/yyyy')  
);  
  
CREATE VIEW MediaPerditaLotto (Lotto, MediaPerdita) AS (  
    SELECT Lotto, avg(Perdita)  
    FROM Lavorazione  
    GROUP BY Lotto  
);  
  
SELECT L.CodLotto, L.Annata, L.Denominazione, L.Famiglia, L.Colore  
FROM Lotto L INNER JOIN MediaPerditaLotto MPL ON L.CodLotto=MPL.Lotto  
WHERE L.CodLotto IN (SELECT CodLotto  
                     FROM LottiAttiAlTaglio) AND  
       MPL.MediaPerdita<(SELECT MP.media_perdite  
                        FROM MediaPerdite2012 MP);
```

L'interrogazione produce il seguente risultato.

CodLotto	Annata	Denominazione	Famiglia	Colore
KDJNH78463	2011	DOC	Prosecco	Bianco

# Procedure Oracle PL/SQL

## Procedura 1: FineAffiliazione

La procedura FineAffiliazione pone fine alla affiliazione di un produttore, ponendo come data di fine affiliazione il momento in cui viene avviata la procedura. Nello specifico, viene preso come parametro un CodProduttore e, se tale valore non genera eccezioni, vengono aggiornate le tabelle Produttore e Affiliazione-Passata, per registrare la fine affiliazione e la tabella Cantina per aggiornare il relativo numero di affiliati.

```
create or replace PROCEDURE FineAffiliazione(p_codProduttore IN CHAR)
IS
    l_codCantina produttore.cantinaAffiliazione%TYPE;
    l_dataInizio produttore.dataInizioAffiliazione%TYPE;
    l_count      NUMERIC(1,0);

BEGIN
    -- verifichiamo che il produttore esista
    SELECT COUNT(CodProduttore)
    INTO    l_count
    FROM    Produttore
    WHERE   CodProduttore=p_codProduttore;

    IF( l_count=0 ) THEN
        DBMS_OUTPUT.put_line('farmer doesn't exists');
        RETURN;
    END IF;

    -- selezioniamo la cantina di affiliazione e la data inizio affiliazione
    SELECT CantinaAffiliazione, DataInizioAffiliazione
    INTO    l_codCantina, l_dataInizio
    FROM    Produttore
    WHERE   CodProduttore=p_codProduttore AND DataInizioAffiliazione IS NOT NULL;

    -- poniamo fine alla affiliazione
    UPDATE Produttore
    SET     CantinaAffiliazione=NULL, DataInizioAffiliazione=NULL
    WHERE   CodProduttore=p_codProduttore;

    -- inseriamo il record in AffiliazionePassata
    INSERT INTO AffiliazionePassata (CodProduttore, CodCantina, DataInizio, DataFine)
    VALUES (p_codProduttore, l_codCantina, l_dataInizio, SYSDATE);

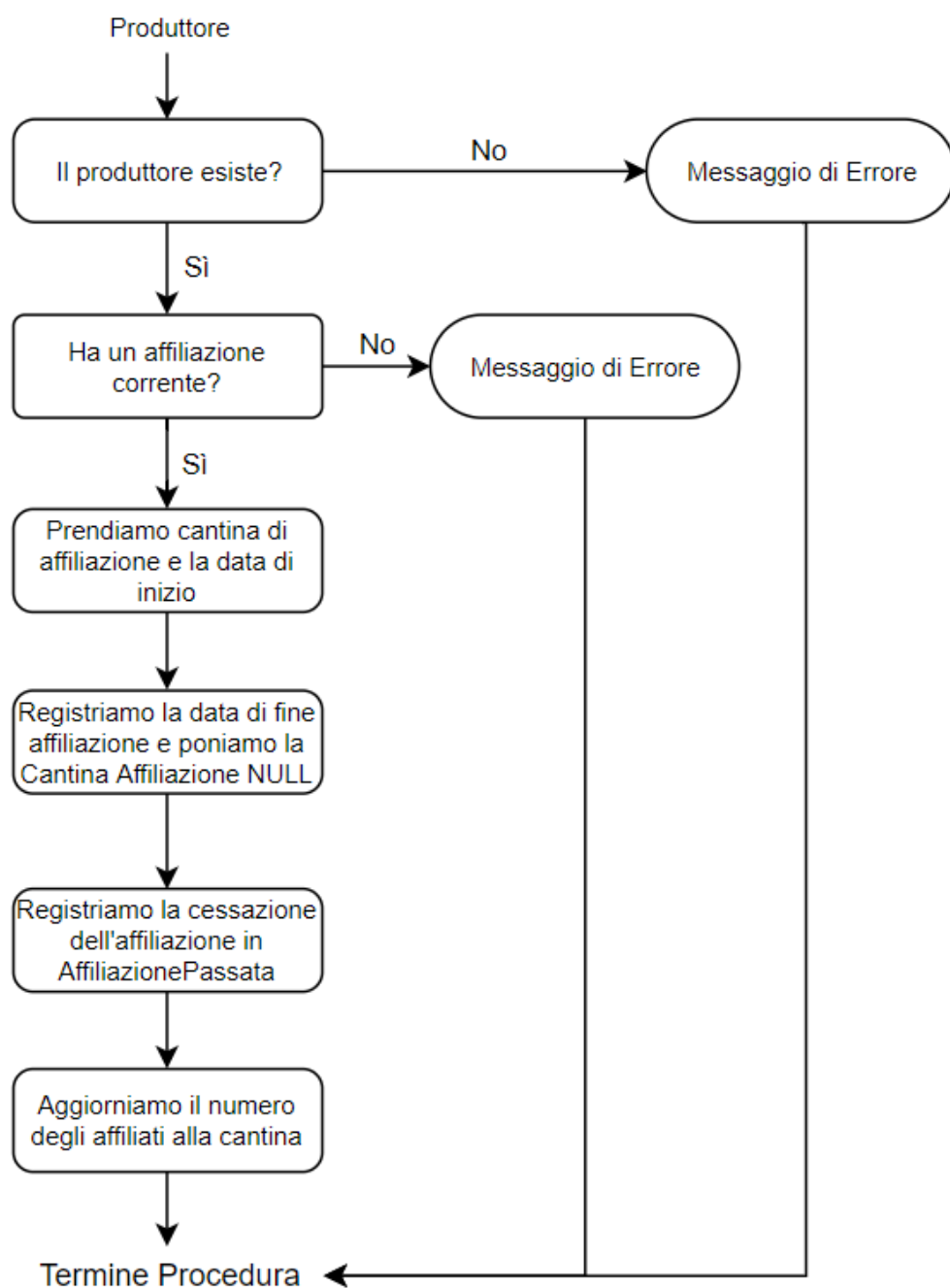
    --diminuiamo il numero di produttori affiliati
    UPDATE Cantina
    SET     Affiliati=Affiliati-1
    WHERE   CodCantina=l_codCantina;

    DBMS_OUTPUT.put_line('produttore rimosso dalla affiliazione');

EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.put_line('the farmer actually doesn't have
                                                an as-sociated winery');
    WHEN OTHERS THEN DBMS_OUTPUT.put_line( SQLERRM );

END FineAffiliazione;
```

Riportiamo il Diagramma di Flusso relativo alla procedura, che esplica in maniera più intuitiva il meccanismo della stessa.





## Procedura 2: InserisciSceltaVendemmiale

La procedura `InserisciSceltaVendemmiale` consente di inserire una scelta vendemmiale relativa all'anno corrente. Prende in ingresso la coppia uva e vigneto e, qualora questi non generino eccezioni, inserisce la coppia nella tabella `SceltaVendemmiale`. Qualora la scelta sia già presente nel database, questa viene aggiornata con i parametri dati in ingresso alla procedura.

```
create or replace PROCEDURE InserisciSceltaVendemmiale (p_uva IN CHAR, p_vigneto IN CHAR)
IS
    l_count          NUMERIC(1,0);
    l_currentYear    NUMERIC(4,0):=EXTRACT(YEAR FROM SYSDATE);

BEGIN
    -- verifichiamo che il vigneto esista
    SELECT COUNT(*)
    INTO    l_count
    FROM    Vigneto
    WHERE   CodVigneto=p_vigneto;

    IF( l_count=0 ) THEN
        DBMS_OUTPUT.put_line('vineyard inserted doesn't exists');
        RETURN;
    END IF;

    -- verifichiamo che l'uva esista
    SELECT COUNT(*)
    INTO    l_count
    FROM    Uva
    WHERE   CodUva=p_uva;

    IF( l_count=0 ) THEN
        DBMS_OUTPUT.put_line('grapes inserted doesn't exists');
        RETURN;
    END IF;

    -- verifichiamo che non esista già una scelta vendemmiale associata all'anno corrente
    SELECT COUNT(*)
    INTO    l_count
    FROM    SceltaVendemmiale
    WHERE   CodVigneto=p_vigneto AND Anno = l_currentYear;

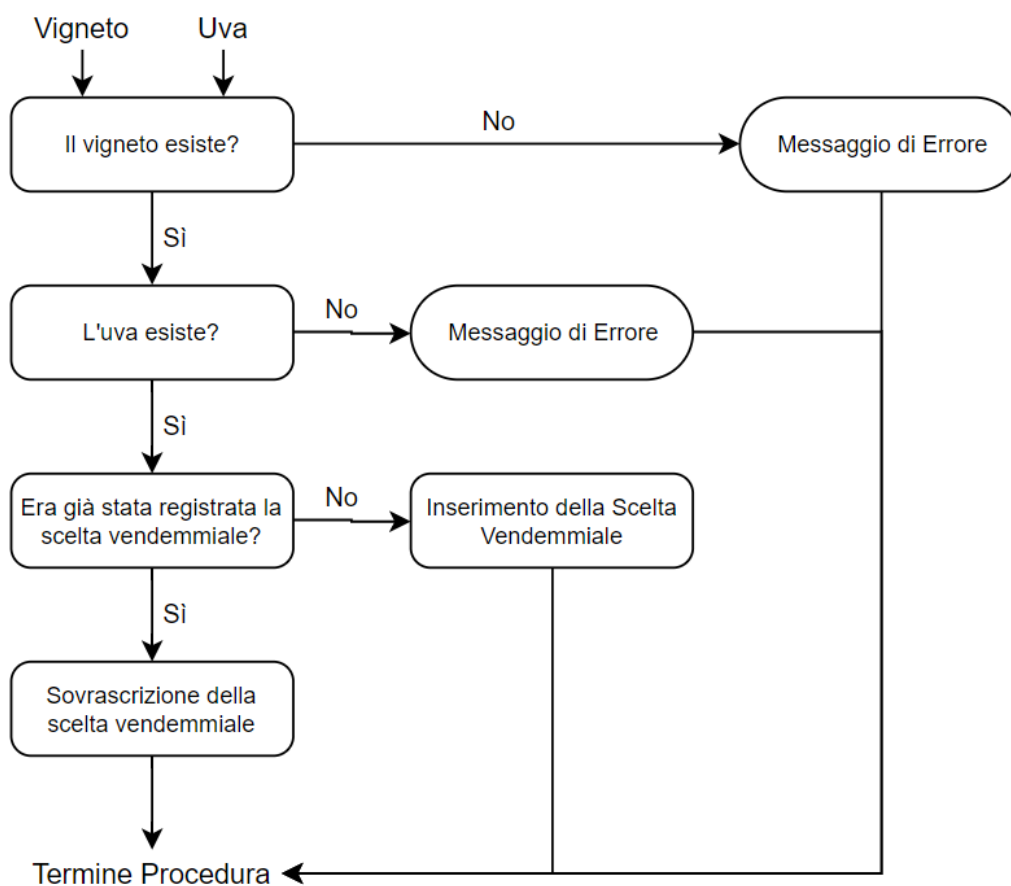
    IF( l_count=1 ) THEN
        UPDATE SceltaVendemmiale
        SET     CodUva=p_uva
        WHERE   CodVigneto=p_vigneto AND Anno=l_currentYear;
        DBMS_OUTPUT.put_line('choice overwritten');

    ELSE
        INSERT INTO SceltaVendemmiale VALUES (p_uva, p_vigneto, l_currentYear);
        DBMS_OUTPUT.put_line('choice inserted');
    END IF;

EXCEPTION
    WHEN OTHERS THEN DBMS_OUTPUT.put_line( SQLERRM );

END InserisciSceltaVendemmiale;
```

Riportiamo il Diagramma di Flusso relativo alla procedura, che esplica in maniera più intuitiva il meccanismo della stessa.



### Procedura 3: MostraPrenotazioniFuture

La procedura `MostraPrenotazioniFuture` mostra tutte le prenotazioni future relative a un produttore. Nello specifico prende come parametri di ingresso un `CodProduttore` e, qualora tale valore non porti ad eccezioni (opportunamente gestite), mostra a video tutte le prenotazioni successive alla data corrente prese dal produttore identificato dal rispettivo codice.

```
create or replace PROCEDURE MostraPrenotazioniFuture (p_CodProduttore CHAR)
IS
    l_prenotazione prenotazione%ROWTYPE;

    CURSOR c_prenotazione IS SELECT *
                                FROM   Prenotazione
                                WHERE  CodProduttore=p_CodProduttore AND
                                       DataAccordo > SYSDATE;

BEGIN

    OPEN c_prenotazione;

    LOOP FETCH c_prenotazione INTO l_prenotazione;
        EXIT WHEN c_prenotazione%NOTFOUND;
        DBMS_OUTPUT.put_line (l_prenotazione.DataConferimento || ' ' ||
                               l_prenotazione.Quantita || ' ' || l_prenotazione.Colore
                               || ' ' || l_prenotazione.ModalitaRaccolta || ' '
                               || l_prenotazione.DataAccordo);

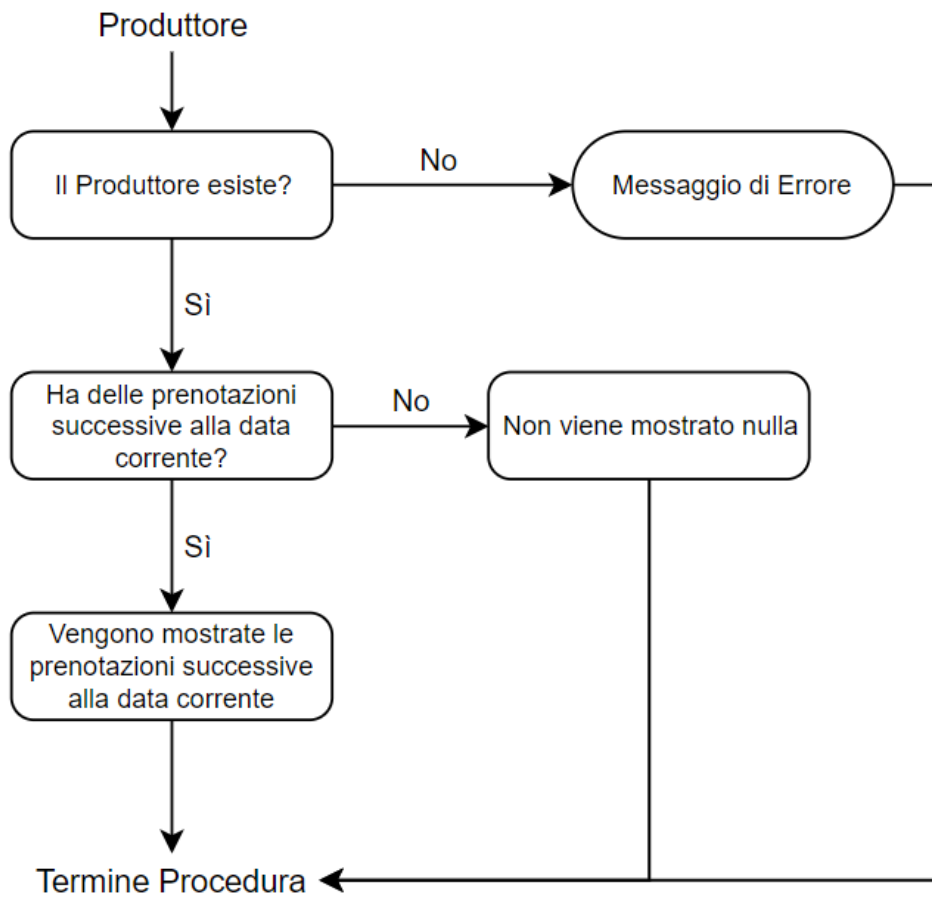
    END LOOP;

    CLOSE c_prenotazione;

EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.put_line('not match found');
    WHEN OTHERS THEN DBMS_OUTPUT.put_line( SQLERRM );

END MostraPrenotazioniFuture;
```

Riportiamo il Diagramma di Flusso relativo alla procedura, che esplica in maniera più intuitiva il meccanismo della stessa.



#### Procedura 4: ModificaDataConferimento

La procedura `ModificaDataConferimento` permette a un produttore di cambiare la data di conferimento qualora la richiesta venga presentata con almeno un giorno di anticipo. La procedura prende in ingresso il codice di prenotazione e la nuova data di conferimento desiderata dal produttore. Qualora questi parametri non generino eccezioni, la richiesta viene accolta e quindi il rispettivo record nella tabella Prenotazioni viene opportunamente aggiornata.

```
create or replace PROCEDURE ModificaDataConferimento (p_prenotazione IN CHAR,
                                                       p_newDate IN DATE)
IS
    l_oldDate prenotazione.dataConferimento%TYPE;

BEGIN
    -- verifichiamo che la data della nuova prenotazione non sia una data già passata
    IF (p_newDate <= SYSDATE ) THEN
        DBMS_OUTPUT.put_line('invalid date inserted');
        RETURN;
    END IF;

    -- selezioniamo la data di conferimento associata alla prenotazione
    SELECT DataConferimento
    INTO    l_oldDate
    FROM    Prenotazione
    WHERE   CodPrenotazione=p_prenotazione;

    -- verifichiamo che la data di conferimento non sia già scaduta
    IF( l_oldDate < SYSDATE )
        THEN DBMS_OUTPUT.put_line('The load date is already passed');
        RETURN;
    END IF;

    IF ( SYSDATE+1 > l_oldDate ) THEN
        DBMS_OUTPUT.put_line('it''s too late to change the load date');
        RETURN;
    END IF;

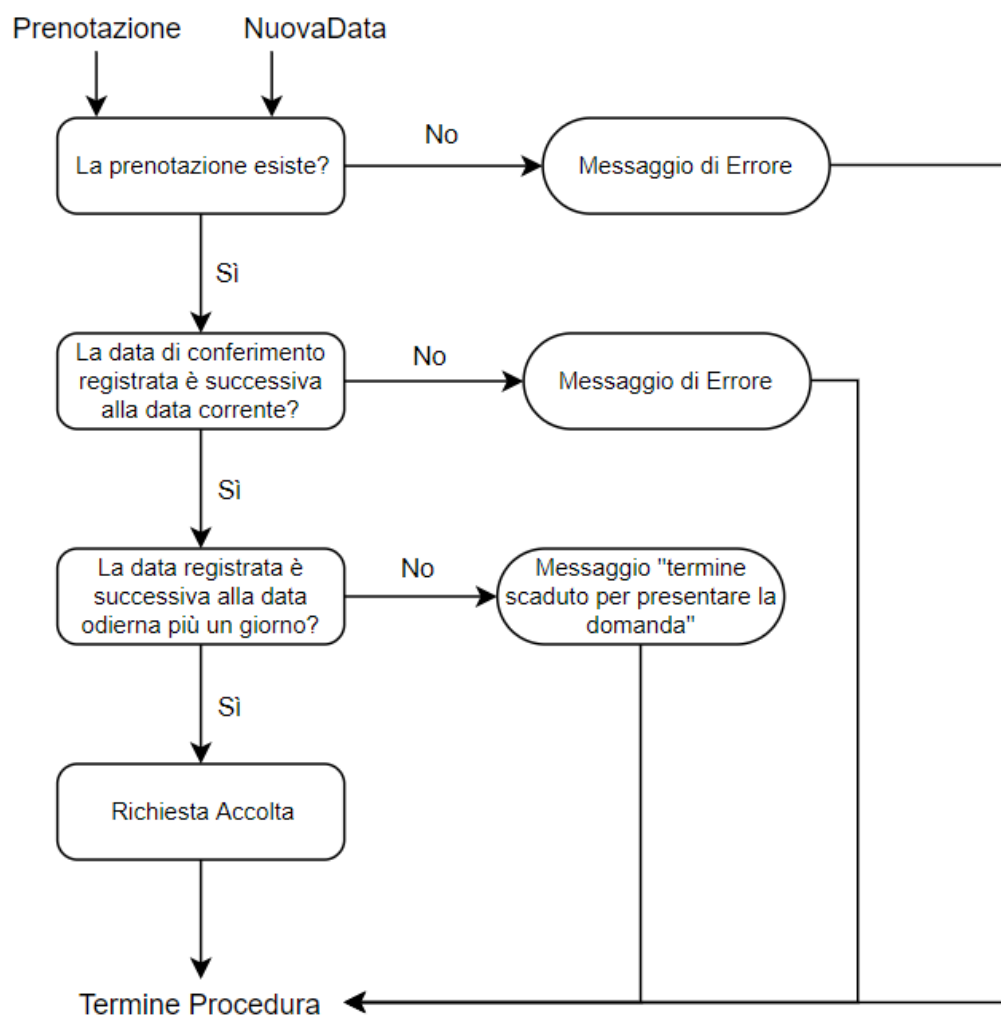
    -- la richiesta arrivata a questo punto viene accolta
    UPDATE Prenotazione
    SET     DataConferimento=p_newDate
    WHERE   CodPrenotazione=p_prenotazione;

    DBMS_OUTPUT.put_line('la richiesta e'' stata accolta');

EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.put_line('the reservation doesn''t exists');
    WHEN OTHERS THEN DBMS_OUTPUT.put_line( SQLERRM );

END ModificaDataConferimento;
```

Riportiamo il Diagramma di Flusso relativo alla procedura, che esplica in maniera più intuitiva il meccanismo della stessa.



## Procedura 5: MostraProcessi

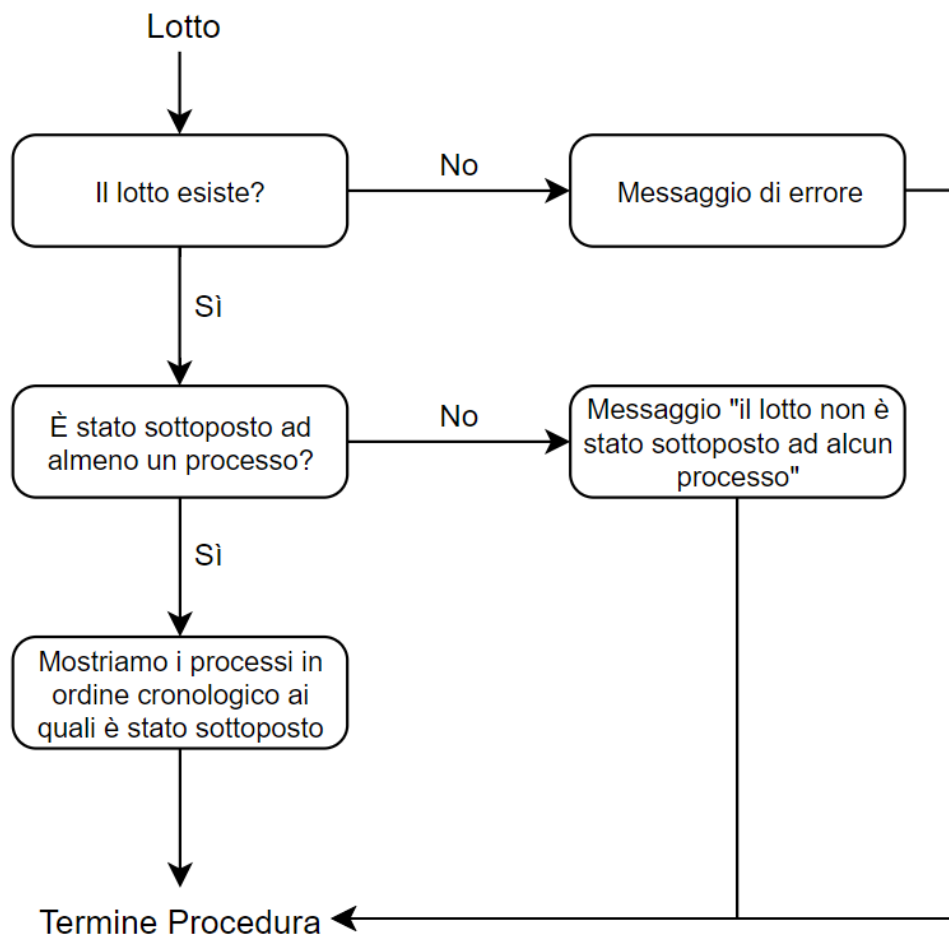
La procedura `MostraProcessi` restituisce tutti i processi cui è stato sottoposto un lotto. Prende in ingresso il codice di un lotto e, qualora questo corrisponda effettivamente a un lotto, mostra tutti i trasferimenti cui è stato sottoposto in ordine temporale.

```
create or replace PROCEDURE MostraProcessi (p_lotto IN CHAR)
IS
    l_nomeProcesso        processo.nomeProcesso%TYPE;
    l_descrizioneProcesso processo.descrizione%TYPE;
    l_dataInizio          lavorazione.dataInizio%TYPE;
    l_dataFine            lavorazione.dataFine%TYPE;
    l_numeroProcessi      NUMERIC(3,0):=1;
    l_count               NUMERIC(3,0);
    CURSOR c_processo IS SELECT  P.NomeProcesso, P.Descrizione, L.DataInizio, L.DataFine
                           FROM    Lavorazione L INNER JOIN Processo P
                           ON      L.Processo=P.NomeProcesso
                           WHERE   L.Lotto=p_lotto
                           ORDER BY L.DataInizio;
BEGIN
    -- verifichiamo che il lotto esista
    SELECT COUNT(CodLotto)
    INTO    l_count
    FROM    Lotto
    WHERE   CodLotto=p_lotto;
    IF( l_count=0 ) THEN
        DBMS_OUTPUT.put_line('lot doesn't exist');
        RETURN;
    END IF;
    -- verifichiamo che il lotto sia stato sottoposto a un qualche processo
    SELECT COUNT(Lotto)
    INTO    l_count
    FROM    Lavorazione
    WHERE   Lotto=p_lotto;
    IF( l_count=0 ) THEN
        DBMS_OUTPUT.put_line('lot hasn't any process associated');
        RETURN;
    END IF;

    OPEN c_processo;
    LOOP FETCH c_processo INTO l_nomeProcesso, l_descrizioneProcesso,
                               l_dataInizio, l_dataFine;
    EXIT WHEN c_processo%NOTFOUND;
        DBMS_OUTPUT.put_line('Processo ' || l_numeroProcessi || ': ' || l_nomeProcesso
                               || ' Descrizione Processo: ' || l_descrizioneProcesso ||
                               '');
        DBMS_OUTPUT.put_line('Data Inizio: ' || l_dataInizio || ' Data Fine: ' ||
                               l_dataFine);
        DBMS_OUTPUT.put_line('');
    END LOOP;
    CLOSE c_processo;

EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.put_line('the lot inserted doesn't exist');
    WHEN OTHERS THEN DBMS_OUTPUT.put_line( SQLERRM );
END MostraProcessi;
```

Riportiamo il Diagramma di Flusso relativo alla procedura, che esplica in maniera più intuitiva il meccanismo della stessa.





## Procedura 6: Trasferisci

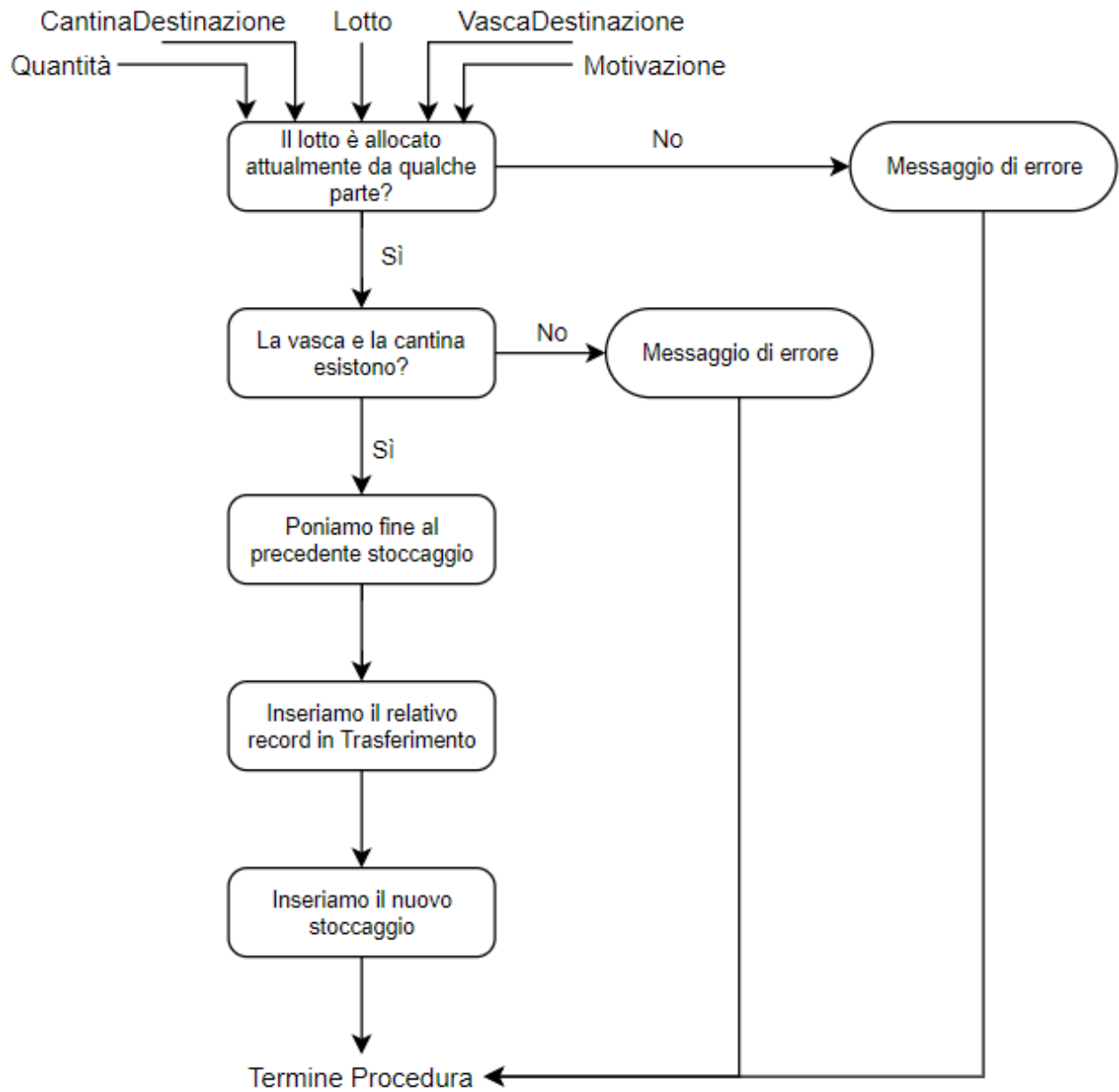
La procedura `Trasferisci` trasferisce un lotto da una vasca all'altra, anche di cantine diverse. Prende in ingresso una tupla avente lo stesso schema della tabella `Trasferimento`. Qualora tale tupla non generi eccezioni, la procedura si occupa di porre fine all'attuale stoccaggio del lotto e anche di aggiungere un nuovo stoccaggio, relativo alla nuova vasca in cui il lotto sarà riposto.

```
create or replace PROCEDURE Trasferisci (p_lotto IN CHAR, p_cantinaDestinazione IN CHAR,
                                         p_vascaDestinazione IN NUMERIC,
                                         p_quantita IN INT, p_motivazione IN VARCHAR)
IS
    l_count          NUMERIC;
    l_numVascaPartenza   vasca.codVasca%TYPE;
    l_numVascaDestinazione vasca.codVasca%TYPE;
    l_cantinaPartenza   cantina.codCantina%TYPE;
    l_cantinaDestinazione cantina.codCantina%TYPE;

BEGIN
    -- controlliamo che il lotto sia allocato da qualche parte
    SELECT COUNT(Lotto)
    INTO   l_count
    FROM   Stoccaggio
    WHERE  Lotto=p_lotto AND DataEOraFine IS NULL;
    IF( l_count=0 )
        THEN DBMS_OUTPUT.put_line('lot doesn't exists or isn't storage in any pool');
        RETURN;
    END IF;
    -- controlliamo che la vasca esista
    SELECT COUNT(CodVasca)
    INTO   l_count
    FROM   Vasca
    WHERE  CodVasca=p_vascaDestinazione AND Cantina=p_cantinaDestinazione;
    IF( l_count=0 ) THEN
        DBMS_OUTPUT.put_line('pool doesn't exists');
        RETURN;
    END IF;
    -- prendiamo il numero di vasca di partenza
    SELECT Vasca, Cantina
    INTO   l_numVascaPartenza, l_cantinaPartenza
    FROM   Stoccaggio
    WHERE  Lotto=p_lotto AND DataEOraFine IS NULL;
    -- poniamo fine all'ormai precedente stoccaggio
    UPDATE Stoccaggio
    SET    DataEOraFine=SYSDATE
    WHERE  Lotto=p_lotto;
    -- inseriamo un nuovo trasferimento
    INSERT INTO Trasferimento VALUES (l_numVascaPartenza, p_vascaDestinazione,
                                       l_cantinaPartenza, p_cantinaDestinazione, p_lotto,
                                       SYSDATE, p_quantita, p_motivazione);
    -- inseriamo il nuovo record di stoccaggio
    INSERT INTO Stoccaggio VALUES (p_lotto, p_vascaDestinazione, p_cantinaDestinazione,
                                    SYSDATE, NULL, p_quantita);
    DBMS_OUTPUT.put_line('lot trasnfered');

EXCEPTION
    WHEN OTHERS THEN DBMS_OUTPUT.put_line( SQLERRM );
END Trasferisci;
```

Riportiamo il Diagramma di Flusso relativo alla procedura, che esplica in maniera più intuitiva il meccanismo della stessa.



## Procedura 7: MostraTrasferimenti

La procedura `MostraTrasferimenti` mostra, a partire da un lotto, gli eventuali trasferimenti ai quali è stato sottoposto. Distingue e gestisce le eccezioni in cui il lotto non esista o che qualora esista non sia stato sottoposto a trasferimenti.

```
create or replace PROCEDURE MostraTrasferimenti (p_lotto IN CHAR)
IS
    l_CantinaPartenza      Trasferimento.CantinaPartenza%TYPE;
    l_CantinaDestinazione  Trasferimento.CantinaDestinazione%TYPE;
    l_VascaPartenza        Trasferimento.VascaPartenza%TYPE;
    l_VascaDestinazione    Trasferimento.VascaDestinazione%TYPE;
    l_NomeCantinaPartenza  Cantina.Nome%TYPE;
    l_NomeCantinaDestinazione Cantina.Nome%TYPE;
    l_count NUMERIC(1,0);
    -- cursore che contiene i vari trasferimenti
    CURSOR c_trasferimento IS SELECT      CantinaPartenza, VascaPartenza,
                                         CantinaDestinazione, VascaDestinazione
                                FROM        Trasferimento
                                WHERE       Lotto=p_lotto
                                ORDER BY   Data;

BEGIN
    SELECT COUNT (CodLotto)
    INTO    l_count
    FROM    Lotto
    WHERE   CodLotto=p_lotto;

    IF( l_count=0 ) THEN
        DBMS_OUTPUT.put_line('lot doesn't exist');
        RETURN;
    END IF;

    OPEN c_trasferimento;
    LOOP FETCH c_trasferimento INTO l_CantinaPartenza, l_VascaPartenza,
                                   l_CantinaDestinazione, l_VascaDestinazione;

        SELECT Nome
        INTO    l_NomeCantinaPartenza
        FROM    Cantina
        WHERE   CodCantina=l_CantinaPartenza;

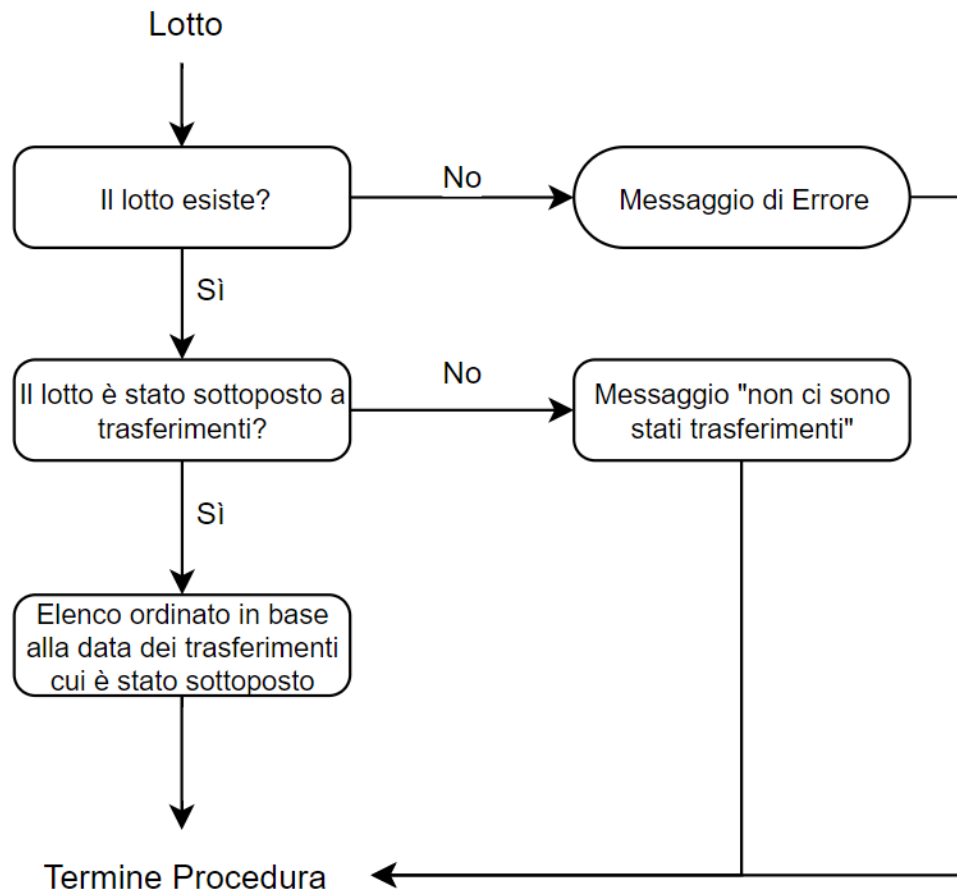
        SELECT Nome
        INTO    l_NomeCantinaDestinazione
        FROM    Cantina
        WHERE   CodCantina=l_CantinaDestinazione;

        DBMS_OUTPUT.put_line('Cantina di Partenza : ' || l_NomeCantinaPartenza || ' '
                             || 'Numero Vasca di Partenza: ' || l_VascaPartenza);
        DBMS_OUTPUT.put_line('Cantina di Destinazione: ' || l_NomeCantinaDestinazione || ' '
                             || 'Numero Vasca di Destinazione: ' || l_VascaDestinazione);

    EXIT WHEN c_trasferimento%NOTFOUND;
    END LOOP;
    CLOSE c_trasferimento;

EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.put_line('lot was never transfered');
    WHEN OTHERS THEN DBMS_OUTPUT.put_line(SQLERRM);
END MostraTrasferimenti;
```

Riportiamo il Diagramma di Flusso relativo alla procedura, che esplica in maniera più intuitiva il meccanismo della stessa.



## Procedura 8: InserisciFase

La procedura `InserisciFase` prende come parametro una fase relativa a una certa vinificazione. Qualora si occupa di inserire una fase all'interno della tabella `Fase`. Nel caso la fase non debba essere solo aggiunta ma si tratti di una fase intermedia, si preoccupa di aggiornare tutti i numeri progressivi delle fasi che ora la seguono. Distingue e gestisce le eccezioni in cui la fase non esista già (sovrascrivendola) e in cui non esista una vinificazione associata a tale fase.

```
create or replace PROCEDURE InserisciFase (p_codFase IN CHAR, p_vinificazione IN VARCHAR,
                                           p_nome IN VARCHAR, p_descrizione IN VARCHAR,
                                           p_numeroProgressivo IN SMALLINT)

IS
    l_count      NUMERIC(1,0);
    l_max        fase.numeroProgressivo%TYPE;

BEGIN
    -- verifichiamo che non esista già la fase
    SELECT COUNT(*)
    INTO    l_count
    FROM    Fase
    WHERE   CodFase=p_codFase AND Vinificazione=p_vinificazione;

    -- se l_count=1 allora la fase esiste già: sovrascriviamo la precedente
    IF ( l_count = 1 ) THEN
        UPDATE Fase
        SET     Nome=p_nome, Descrizione=p_descrizione,
              NumeroProgressivo=p_NumeroProgressivo
        WHERE   CodFase=p_codFase AND Vinificazione=p_vinificazione;

        DBMS_OUTPUT.put_line('phase overwritten');
        RETURN;
    END IF;

    -- Verifichiamo che esista la vinificazione associata
    SELECT count(TipoVinificazione)
    INTO    l_count
    FROM    Vinificazione
    WHERE   TipoVinificazione=p_Vinificazione;

    -- se l_count=0 allora non esiste la vinificazione associata
    IF ( l_count=0 ) THEN
        DBMS_OUTPUT.put_line('The vinification doesn't exists');
        RETURN;
    END IF;

    -- Distinguiamo due casi: quello in cui viene aggiunta in coda o inserita nel mezzo
    SELECT COUNT(*)
    INTO    l_count
    FROM    Fase
    WHERE   Vinificazione=p_vinificazione AND NumeroProgressivo=p_numeroProgressivo;

    -- se l_count=0 aggiungiamo la fase, altrimenti va riordinato il numero progressivo
    IF(l_count=0)
        THEN INSERT INTO Fase VALUES (p_CodFase, p_Vinificazione, p_Nome, p_Descrizione,
                                         p_NumeroProgressivo);
```

```

        DBMS_OUTPUT.put_line('phase added');

ELSE

    SELECT MAX(NumeroProgressivo)
    INTO    l_max
    FROM    Fase
    WHERE   Vinificazione=p_Vinificazione;

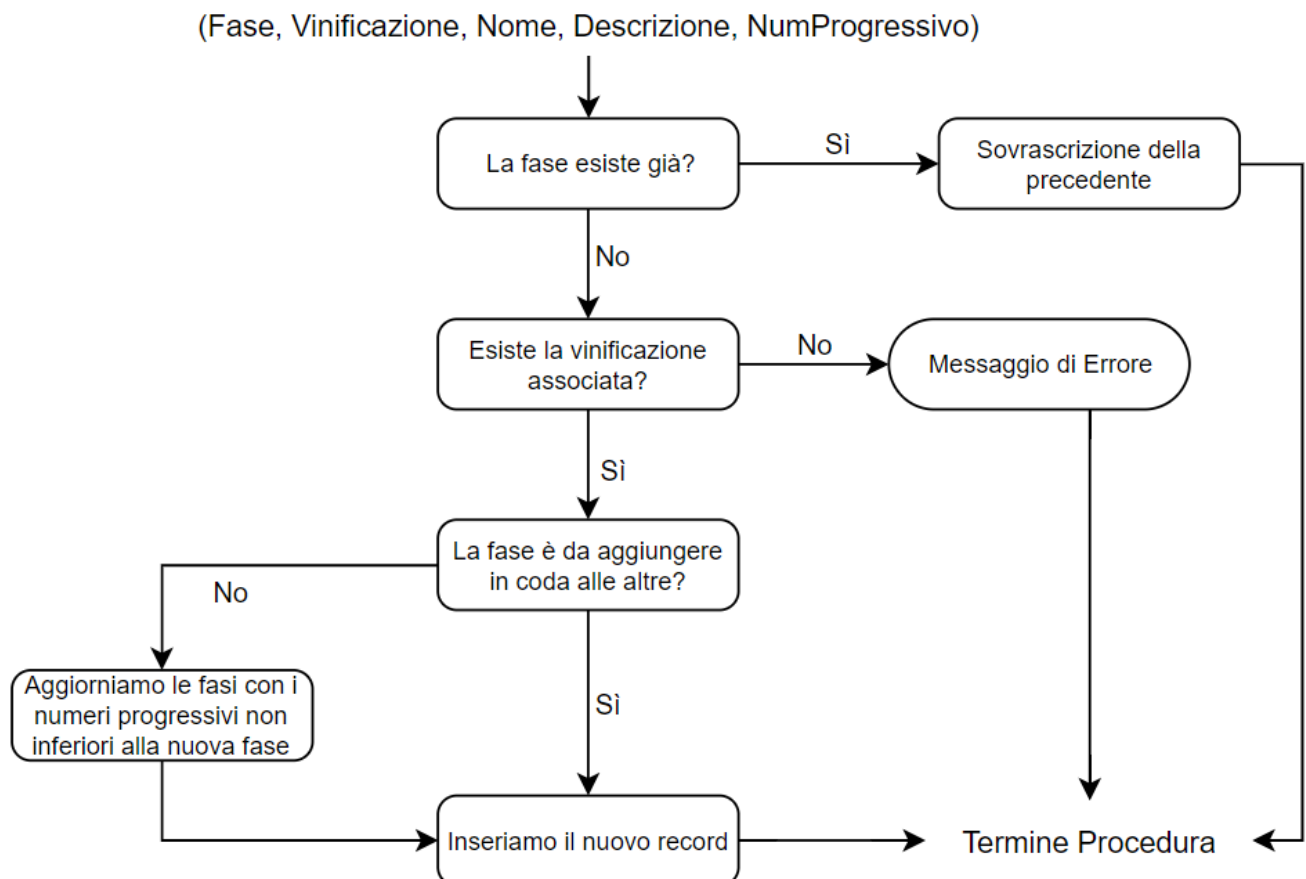
    UPDATE Fase
    SET     NumeroProgressivo = NumeroProgressivo+1
    WHERE   Vinificazione = p_Vinificazione AND
            NumeroProgressivo >= p_numeroProgressivo;

    INSERT INTO Fase
    VALUES (p_CodFase, p_Vinificazione, p_Nome, p_Descrizione, p_NumeroProgressivo);
    DBMS_OUTPUT.put_line('phase inserted');
END IF;

EXCEPTION
    WHEN OTHERS THEN DBMS_OUTPUT.put_line( SQLERRM );
END InserisciFase;

```

Riportiamo il Diagramma di Flusso relativo alla procedura, che esplica in maniera più intuitiva il meccanismo della stessa.



## Procedura 9: EliminaFase

La procedura `EliminaFase` si occupa di eliminare una fase, eventualmente riorganizzando i numeri progressivi delle fasi che fanno riferimento alla stessa vinificazione. Anche questa procedura gestisce le varie eccezioni che si possono verificare nell'esecuzione della stessa.

```
create or replace PROCEDURE EliminaFase (p_codFase IN CHAR, p_vinificazione IN VARCHAR)

IS
    l_maxNumProg  fase.numeroProgressivo%TYPE;
    l_count       NUMERIC(1,0);

BEGIN
    --verifichiamo che la fase esista
    SELECT count(*)
    INTO    l_count
    FROM    Fase
    WHERE   CodFase=p_codFase AND Vinificazione=p_vinificazione;

    IF( l_count=0 ) THEN
        DBMS_OUTPUT.put_line('the phase doesn't exists');
        RETURN;
    END IF;

    -- troviamo il numero progressivo massimo da aggiornare
    SELECT NumeroProgressivo
    INTO    l_maxNumProg
    FROM    Fase
    WHERE   codFase=p_codFase AND Vinificazione=p_vinificazione;

    -- cancelliamo la fase desiderata
    DELETE FROM Fase
    WHERE   Vinificazione=p_vinificazione AND CodFase=p_codFase;

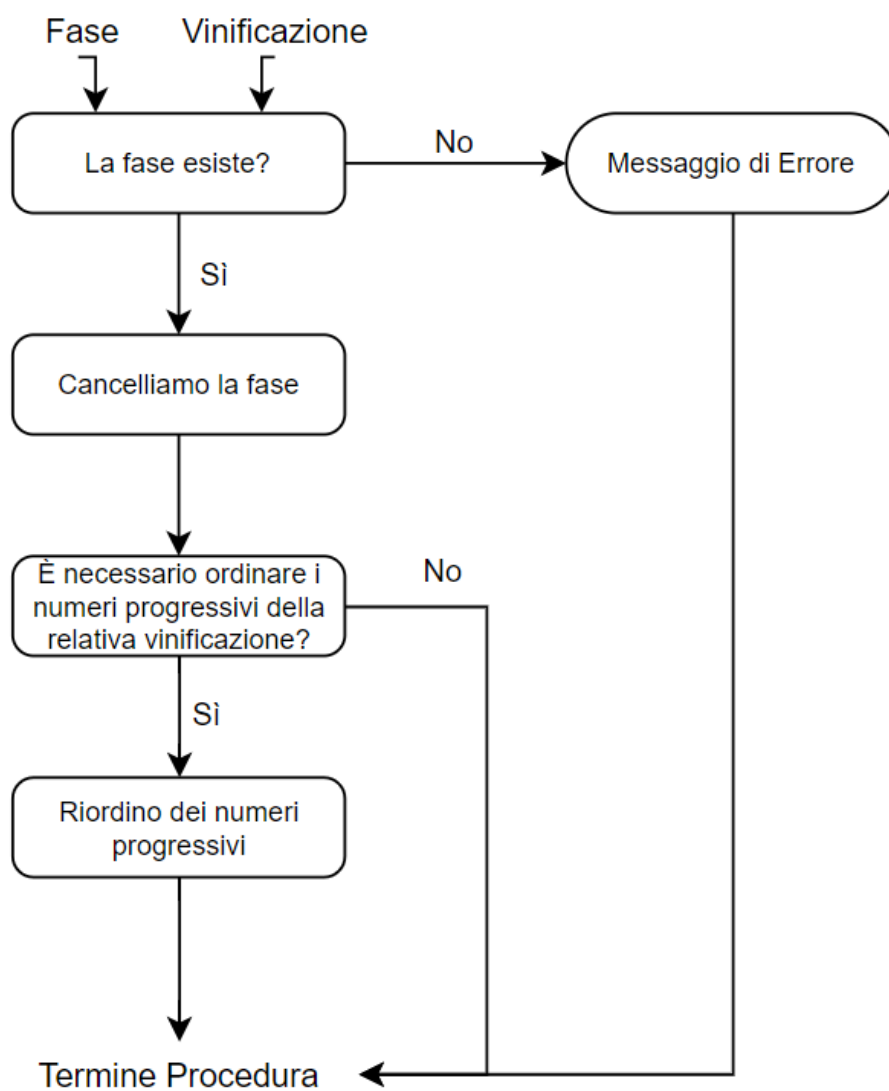
    -- riordiniamo i record
    UPDATE Fase
    SET     NumeroProgressivo=NumeroProgressivo-1
    WHERE   Vinificazione=p_vinificazione AND NumeroProgressivo >= l_maxNumProg;

    DBMS_OUTPUT.put_line('la fase è stata eliminata correttamente');

EXCEPTION
    WHEN OTHERS THEN DBMS_OUTPUT.put_line( SQLERRM );

END EliminaFase;
```

Riportiamo il Diagramma di Flusso relativo alla procedura, che esplica in maniera più intuitiva il meccanismo della stessa.





## Procedura 10: OrarioLavorativo

La procedura `OrarioLavorativo` si occupa di assicurarsi che le modifiche vengano apportate solo nei giorni feriali e solo in orario lavorativo, che qui si è supposto essere nella fascia oraria 07:00 – 20:00 dei soli giorni feriali.

```
create or replace PROCEDURE OrarioLavorativo
IS
BEGIN
    -- si è ipotizzato un'orario lavorativo vada dalle 7:00 alle 20:00 dei giorni feriali
    IF TO_CHAR (SYSDATE, 'HH24:MI') NOT BETWEEN '7:00' AND '20:00'
        OR TO_CHAR (SYSDATE, 'DY') IN ('SAT', 'SUN') THEN
        RAISE_APPLICATION_ERROR (-20205, 'You may only make change during normal office
hours');
    END IF;

EXCEPTION
    WHEN OTHERS THEN DBMS_OUTPUT.put_line( SQLERRM );
END OrarioLavorativo;
```

Riportiamo il Diagramma di Flusso relativo alla procedura, che esplica in maniera più intuitiva il meccanismo della stessa.

