

# **AZIENDA VINICOLA**

*Relazione del progetto di Basi di Dati*

Curata da

**Eugenio Militerno, MAT: 2454**

**Pasquale Ruotolo, MAT: 2322**

Sotto la supervisione di

**ANTONIO MARATEA**

Docente di Basi di Dati (INF/01)



**CdL di INFORMATICA**

UNIVERSITA' DEGLI STUDI DI NAPOLI PARTHENOPE

(DIPARTIMENTO DI SCIENZE E TECNOLOGIA)

Napoli, Centro direzionale isola C4, **Giugno, 2022**

# Sommario

<b>1</b>	<b>Traccia</b>	<b>2</b>
1.1	Descrizione . . . . .	4
1.2	Obiettivo . . . . .	4
<b>2</b>	<b>Ideazione</b>	<b>6</b>
<b>3</b>	<b>Realizzazione</b>	<b>7</b>
<b>4</b>	<b>Diagramma ER</b>	<b>8</b>
<b>5</b>	<b>Modello Relazionale (UML)</b>	<b>9</b>
<b>6</b>	<b>Diagramma Casi d'Uso</b>	<b>10</b>
<b>7</b>	<b>Codice</b>	<b>12</b>
<b>8</b>	<b>Codice di Trigger e Procedure</b>	<b>24</b>
8.1	Trigger . . . . .	24
8.2	Procedure . . . . .	30

# 1 Traccia

L'azienda vinicola produce e imbottiglia diversi tipi di **vini** . I vini si differenziano tra loro per la propria ricetta e per le **proprietà organolettiche** come *il colore, l'acidità, ecc...*; altri parametri concorrono a definire le caratteristiche di un vino: *alcool, sensazione di astringenza* (dovuta al numero di tannini).

Costituisce ulteriore distinzione il contenuto in **zuccheri non fermentati del vino** (secco, semisecco, dolce...). Inoltre ogni vino è caratterizzato da una **temperatura di servizio** (temperatura ideale per la consumazione). I vini sono immessi al consumo dopo aver subito il **processo di vinificazione** (quindi senza interventi tecnici successivi o aggiunte di altri componenti). L'azienda richiede per i propri vini la certificazione attestante la qualità del prodotto. L'analisi del vino viene effettuata da uno degli **enti predisposti** e si conclude con un giudizio di idoneità o di non idoneità. L'azienda utilizza le uve provenienti da **diversi vigneti**. Ogni vigneto è destinato alla coltivazione di uno o più **vitigni**, ha un'estensione specifica ed è situato in un preciso comune. Per ogni vino prodotto, si deve memorizzare la composizione, ovvero *l'insieme delle uve utilizzate e la relativa percentuale e i vigneti di provenienza*. L'azienda vuole memorizzare anche lo storico di tutti gli **interventi** realizzati sulle viti. I trattamenti contro le avversità possono essere applicati in più fasi e richiedono l'utilizzo di diversi prodotti chimici, dei quali si memorizza *il nome commerciale e il nome del principio attivo*.

Per ogni vigneto, si vuole conoscere il **numero di trattamenti e interventi** a cui è stato sottoposto. I vini vengono imbottigliati presso le varie sedi dell'azienda.

Di ogni sede si intende memorizzare *il comune in cui si trova, il nome e il nu-*

*mero di differenti vini che imbottiglia.* Per i vini è obbligatorio indicare **l'anno di produzione**. Si vuole inoltre conoscere *la quantità di vino imbottiglia ogni anno.* L'azienda vende i propri prodotti all'**ingrosso** a *catene alberghiere, ristoranti e negozi alimentari* (per i quali memorizza *il nome e la ragione sociale, indirizzo e comune*). Per le vendite, l'azienda prepara confezioni contenenti un certo numero di bottiglie dello stesso vino.

## 1.1 Descrizione

La nostra idea e' stata quella di seguire l'azienda vinicola nel raccolto, immagazzinamento, imbottigliamento e vendita del suo vino. L'azienda puo' imbottigliare ( *presso sue sedi o sedi esterne* ), il vino (anche di altre aziende), ed e' interessata a sapere se i suoi vini ottengono una certificazione. Inoltre, e' interessata a sapere il quantitativo di uva raccolta alla vendemmia, a conoscere quali e quante altre aziende fanno imbottigliare il loro vino presso le sue sedi, e tenere traccia delle sue vendite.

## 1.2 Obiettivo

Il progetto e' partito con l'idea di creare un DB per un'azienda vinicola che seguisse tutte le fasi della produzione e della vendita del vino. Sono state implementate inizialmente le parti riguardanti l'uva e i vigneti, seguite poi dalla trasformazione in mosto e successivamente in vino, e infine l'aspetto delle Certificazioni e della vendita del vino ad un cliente. Le query che ci siamo posti sono:

- 1-Trovare le sedi presso cui e' stato imbottigliato uno specifico tipo di vino
- 2-Trovare le sedi nelle quali e' stato imbottigliato un vino di un'altra azienda
- 3-Trovare i vini prodotti con l'uva vendemmiata dal vigneto con piu' trattamenti subiti
- 4-Visualizzare la quantita' media di uva raccolta, e di vini imbottigliati



5-Trovare i vigneti da cui provengono vini prodotti esclusivamente con l'uva ivi coltivata.

6-Visualizzare, per ciascuna certificazione, la lista e l'annata dei vini che hanno ottenuto tale certificazione

7-Visualizzare le vendite, differenziando tra "Al dettaglio", e "All'ingrosso", e visualizzare gli ordini fatti.

8-Trovare tutti i vitigni che hanno subito un preciso numero di trattamenti con uno specifico prodotto.

## 2 Ideazione

L'ideazione del DataBase è partita dal **Diagramma ER**, che ci è servito per idealizzare concettualmente i vari passaggi necessari all'azienda per eseguire tutte le fasi nella produzione, imbottigliamento e vendita del proprio vino. Successivamente, siamo passati a creare un diagramma UML per il **Modello Relazionale**, che conteneva anche le *Tabelle di Transizione*, utili per gestire le associazioni di grado M:N del DataBase. Infine, siamo passati ad SQL, per implementare tutto ciò che avevamo svolto nel progetto. Si allega qui: [8](#) un hyperlink alla pagina contenente il Modello ER, qui: [9](#) un hyperlink per il Modello Relazionale, e qui: [10](#) l'hyperlink alla pagina del Diagramma degli Attori.

### 3 Realizzazione

Abbiamo cominciato ad implementare il diagramma ER, partendo dal lato logico, e poi seguendo quello pratico:

Come prima entità abbiamo il Vigneto, con chiave {NOME,COMUNE}, che può subire Interventi, di chiave {TIPO,DATA}, e Trattamenti, con chiave {NOME PRODOTTO,DATA}. Il vigneto produce l'entità Tipo\_Uva, con chiave {TIPO}, ed è associato all'entità Raccolto\_Vigneto, che ha chiave {DATA}.

L'entità Tipo\_Uva compone il Tipo\_Vino, identificato dalla chiave {NOME\_VINO}, che rappresenta solamente la tipologia del vino.

Il raccolto del vigneto viene poi trasformato in Mosto, identificato da {NUM.O LOTTO}. Il Mosto viene poi Fermentato in una Sede\_Vinificazione, identificata da {NOME,INDIRIZZO}, e genera quindi il Lotto\_Vino.

Ciascun tipo di vino subisce un Controllo\_Certificazione, con chiave {NUM.O PRATICA}, che viene effettuato da un Ente idoneo, identificato da {NOME}.

Ciascun Lotto di vino viene Invecchiato in una Cantina, di chiave {NOME,INDIRIZZO}, e in una Botte, identificata da {NUM.O BOTTE}, con un'associazione ternaria. Successivamente verrà Imbottigliato da un Imbottigliatore, identificato da {NOME,INDIRIZZO} e diventa parte di una Confezione, con chiave {NUMERO,PREZZO}

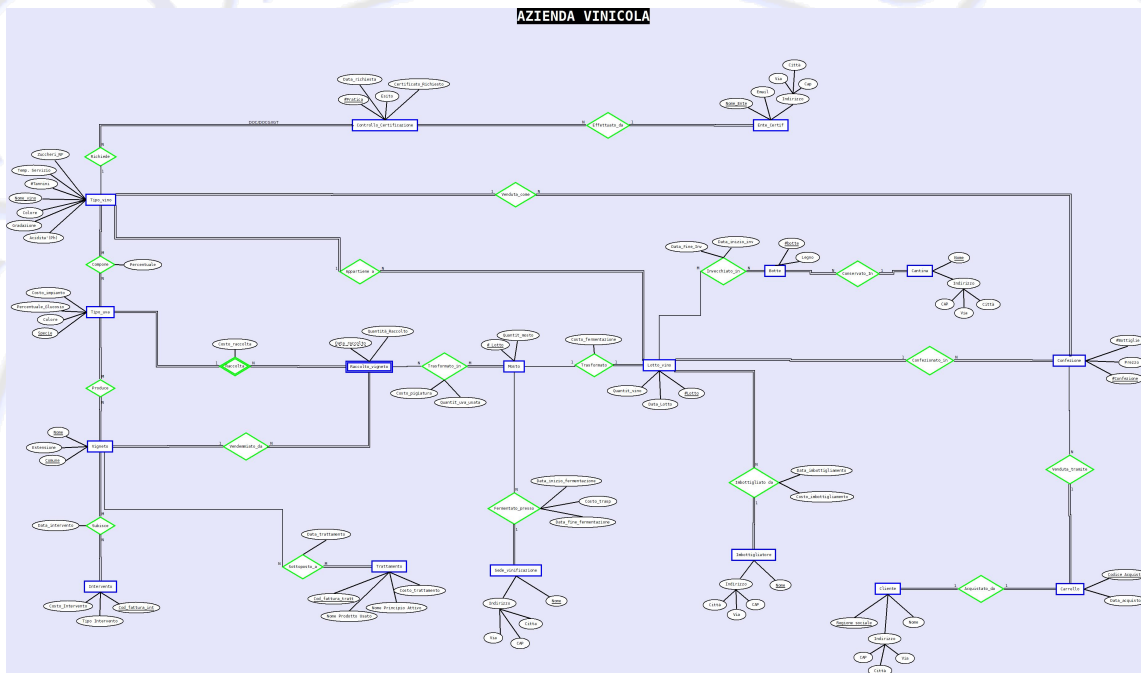
La vendita viene effettuata tramite *Carrello*, grazie al quale il Cliente può acquistare una o più Confezioni (anche differenti) di vino). Il presso è definito come attributo dell'associazione Vendita.



## 4 Diagramma ER

Hyperlink al jpeg (Meglio Visibile): [Diagramma ER](#)

Di seguito, l'Immagine JPEG del diagramma ER

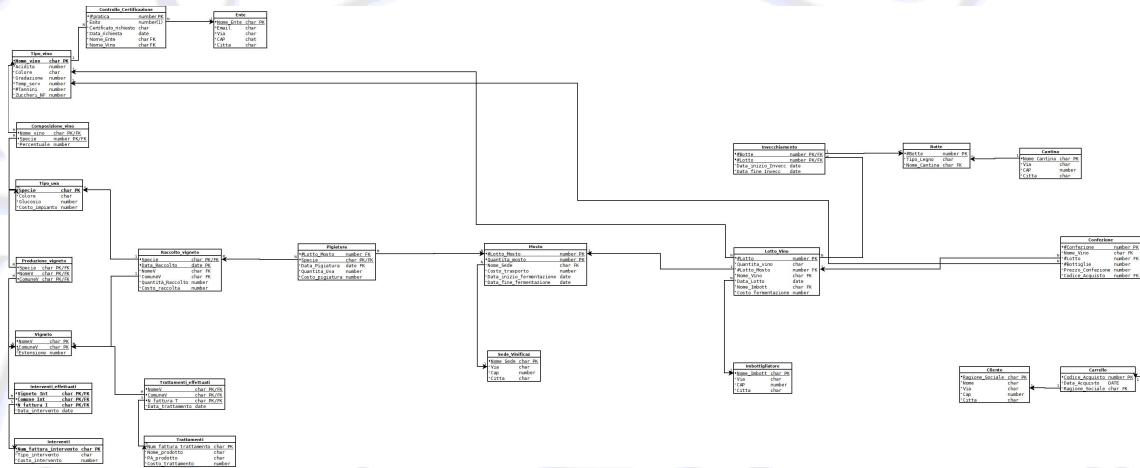


Il Diagramma ER rappresenta lo schema concettuale (*creato usando la **Model-lazione Concettuale***) del nostro DataBase. Esso fornisce lo scheletro, e le principali informazioni sui vari "oggetti" da trattare, ovvero le tabelle in UML. Inoltre, ci consente di sapere quali entità si interfacciano con quali altre, e in che modalità (*Associazioni 1:1, 1:N, M:N*), e infine i parametri di identificazione di ciascuna entità.

## 5 Modello Relazionale (UML)

Hyperlink al jpeg (Meglio Visibile): [Diagramma UML](#)

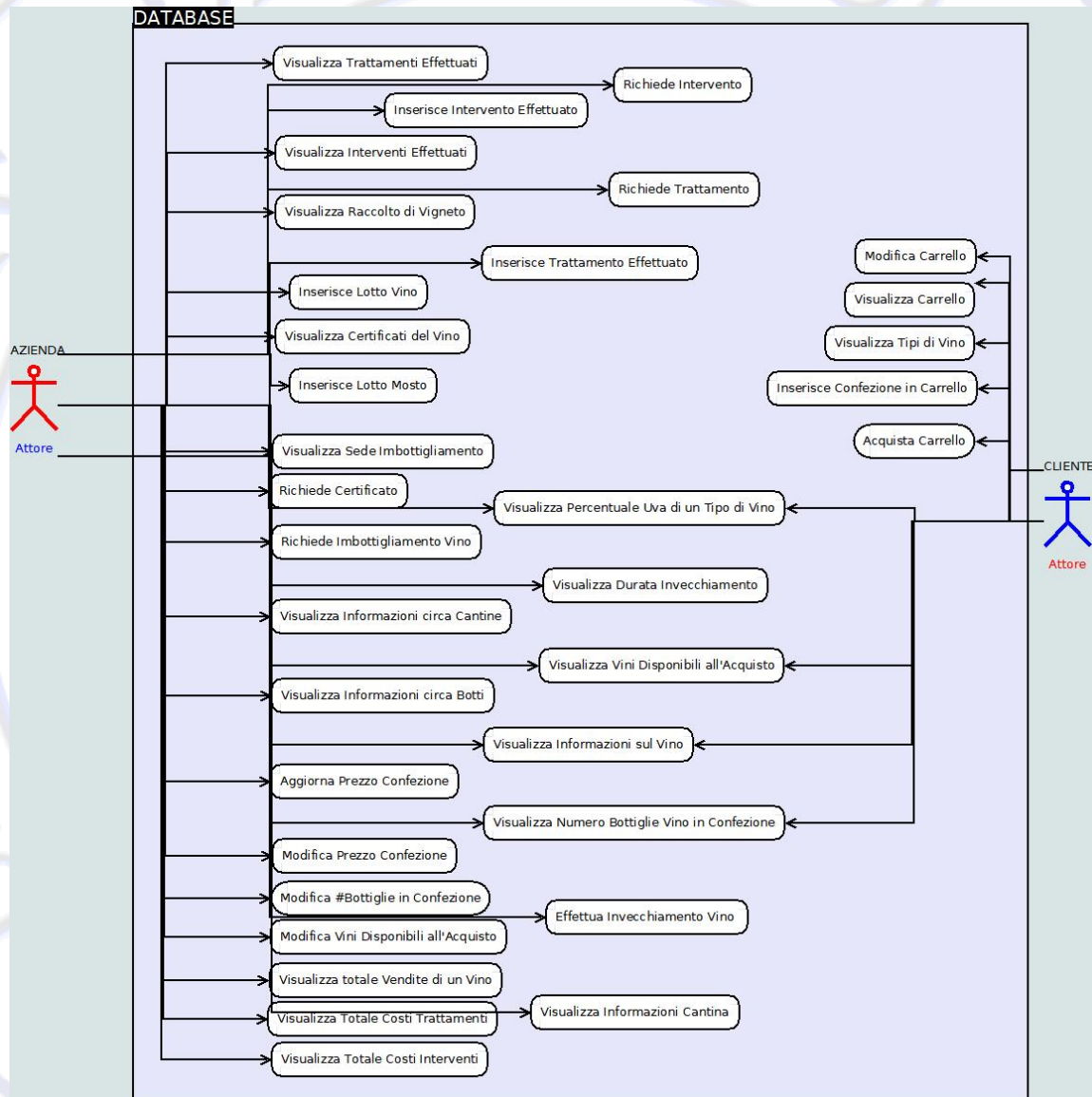
Di seguito, il JPEG del Diagramma:



## 6 Diagramma Casi d'Uso

Hyperlink al jpeg (Meglio Visibile): [CASI D'USO](#)

Di seguito, il Diagramma dei Casi d'Uso



In questo grafico, mostriamo gli utenti del Database, e le loro possibili interazioni

con esso. Ci sono due tipi di utenti:

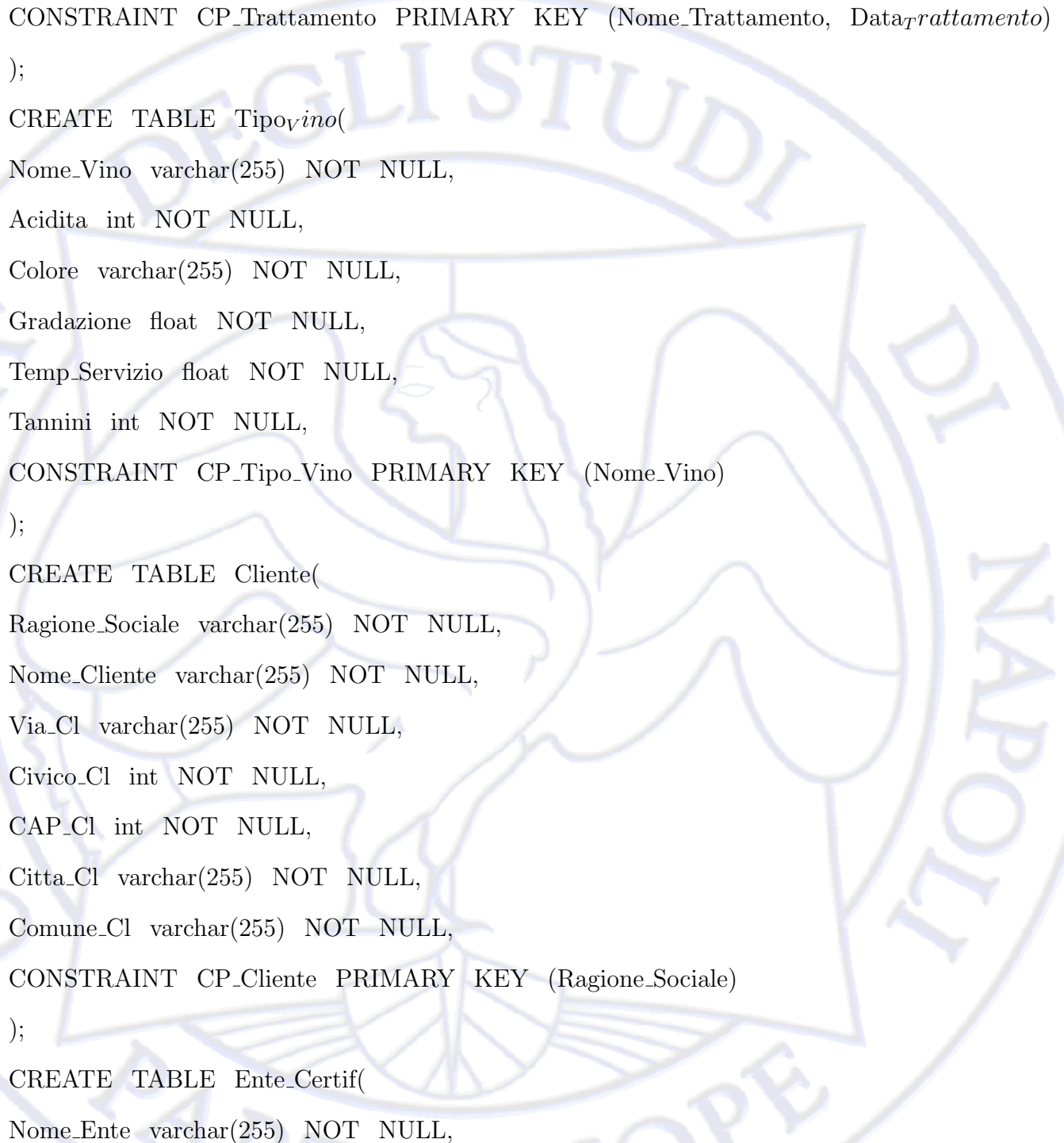


## 7 Codice

**CODICE RIGUARDANTE LE TABELLE CREATE**, comprese quelle *di transizione*

```
CREATE TABLE Vigneto(  
NomeV varchar(255) NOT NULL,  
ComuneV varchar(255) NOT NULL,  
Estensione float NOT NULL,  
CONSTRAINT CP_Vigneto PRIMARY KEY (NomeV, ComuneV)  
);  
  
CREATE TABLE Tipo_Uva(  
Specie varchar(255) NOT NULL,  
Glucosio float NOT NULL,  
CONSTRAINT CP_Tipo_Uva PRIMARY KEY (Specie)  
);  
  
CREATE TABLE Intervento(  
Tipo_Intervento varchar(255) NOT NULL,  
Data_Intervento DATE NOT NULL,  
CONSTRAINT CP_Intervento PRIMARY KEY (Tipo_Intervento, Data_Intervento)  
);  
  
CREATE TABLE Trattamento(  
Nome_Trattamento varchar(255) NOT NULL,  
Data_Trattamento DATE NOT NULL,  
Princ_Att_Prod varchar(255) NOT NULL,
```

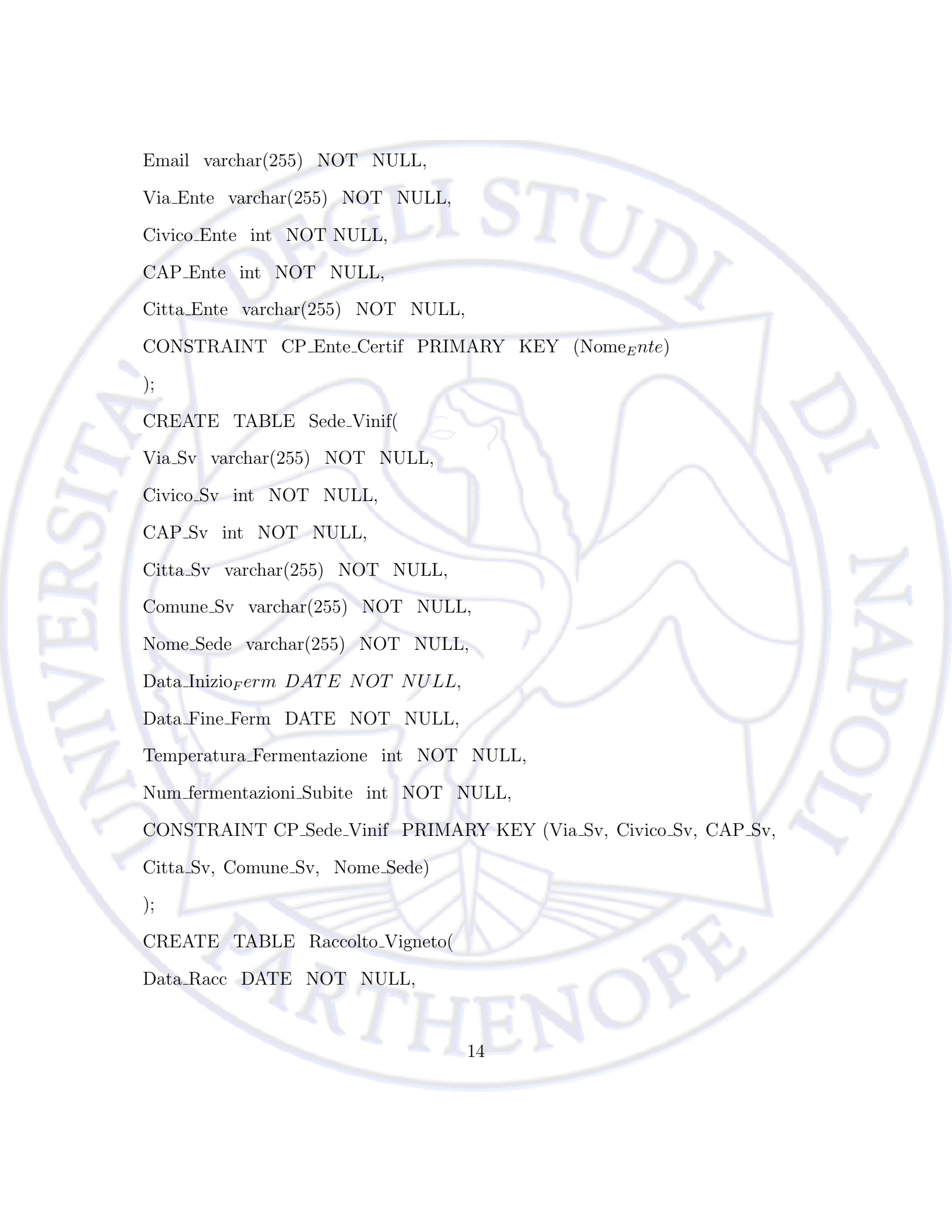




```

CONSTRAINT CP_Trattamento PRIMARY KEY (Nome_Trattamento, DataTrattamento)
);
CREATE TABLE TipoVino(
Nome_Vino varchar(255) NOT NULL,
Acidita int NOT NULL,
Colore varchar(255) NOT NULL,
Gradazione float NOT NULL,
Temp_Servizio float NOT NULL,
Tannini int NOT NULL,
CONSTRAINT CP_Tipo_Vino PRIMARY KEY (Nome_Vino)
);
CREATE TABLE Cliente(
Ragione_Sociale varchar(255) NOT NULL,
Nome_Cliente varchar(255) NOT NULL,
Via_Cl varchar(255) NOT NULL,
Civico_Cl int NOT NULL,
CAP_Cl int NOT NULL,
Citta_Cl varchar(255) NOT NULL,
Comune_Cl varchar(255) NOT NULL,
CONSTRAINT CP_Cliente PRIMARY KEY (Ragione_Sociale)
);
CREATE TABLE Ente_Certif(
Nome_Ente varchar(255) NOT NULL,

```



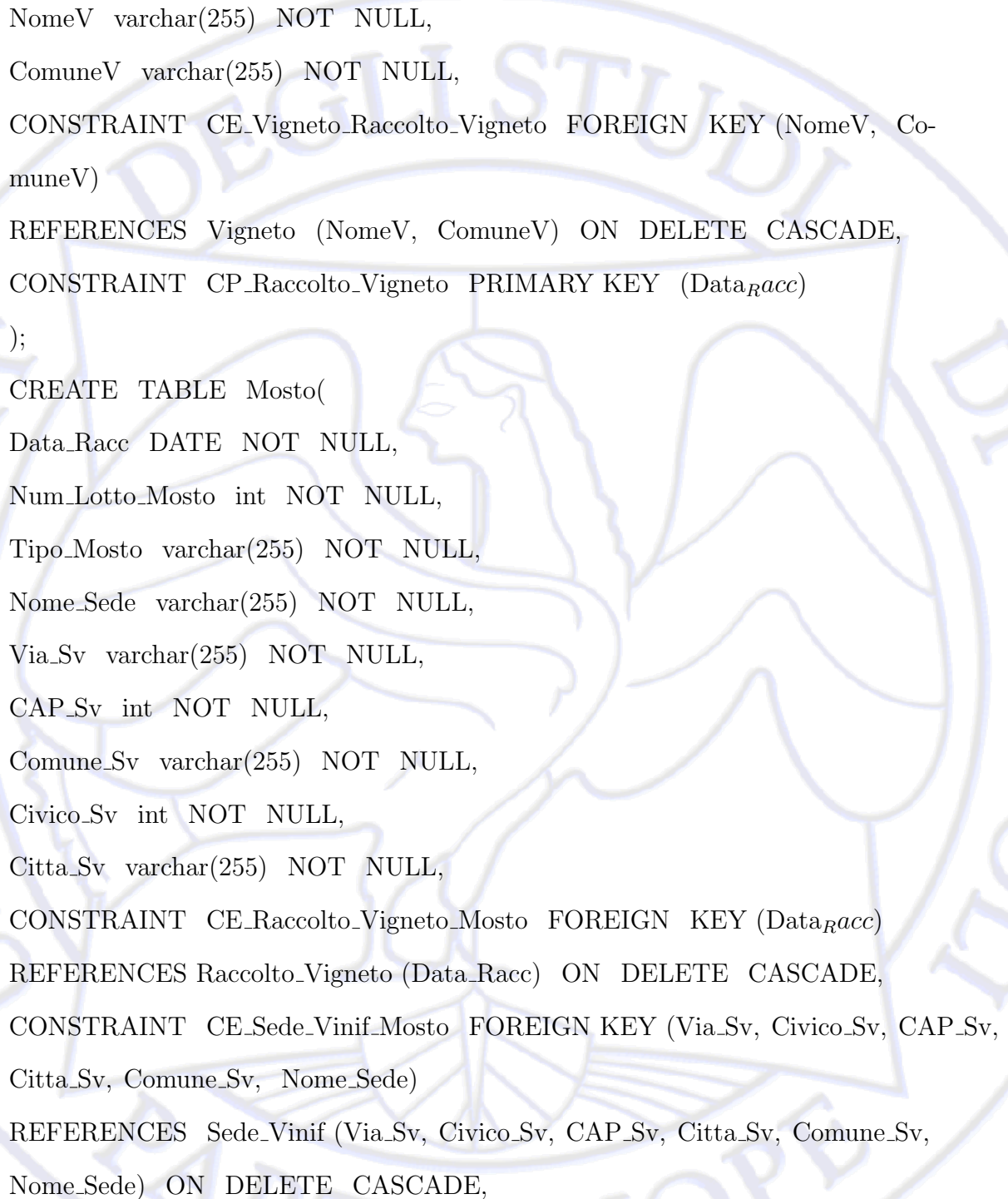
```

Email varchar(255) NOT NULL,
Via_Ente varchar(255) NOT NULL,
Civico_Ente int NOT NULL,
CAP_Ente int NOT NULL,
Citta_Ente varchar(255) NOT NULL,
CONSTRAINT CP_Ente_Certif PRIMARY KEY (NomeEnte)
);

CREATE TABLE Sede_Vinif(
Via_Sv varchar(255) NOT NULL,
Civico_Sv int NOT NULL,
CAP_Sv int NOT NULL,
Citta_Sv varchar(255) NOT NULL,
Comune_Sv varchar(255) NOT NULL,
Nome_Sede varchar(255) NOT NULL,
Data_InizioFerm DATE NOT NULL,
Data_Fine_Ferm DATE NOT NULL,
Temperatura_Fermentazione int NOT NULL,
Num_fermentazioni_Subite int NOT NULL,
CONSTRAINT CP_Sede_Vinif PRIMARY KEY (Via_Sv, Civico_Sv, CAP_Sv,
Citta_Sv, Comune_Sv, Nome_Sede)
);

CREATE TABLE Raccolto_Vigneto(
Data_Racc DATE NOT NULL,

```



```

NomeV varchar(255) NOT NULL,
ComuneV varchar(255) NOT NULL,
CONSTRAINT CE_Vigneto_Raccolto_Vigneto FOREIGN KEY (NomeV, Co-
muneV)
REFERENCES Vigneto (NomeV, ComuneV) ON DELETE CASCADE,
CONSTRAINT CP_Raccolto_Vigneto PRIMARY KEY (DataRacc)
);
CREATE TABLE Mosto(
Data_Racc DATE NOT NULL,
Num_Lotto_Mosto int NOT NULL,
Tipo_Mosto varchar(255) NOT NULL,
Nome_Sede varchar(255) NOT NULL,
Via_Sv varchar(255) NOT NULL,
CAP_Sv int NOT NULL,
Comune_Sv varchar(255) NOT NULL,
Civico_Sv int NOT NULL,
Citta_Sv varchar(255) NOT NULL,
CONSTRAINT CE_Raccolto_Vigneto_Mosto FOREIGN KEY (DataRacc)
REFERENCES Raccolto_Vigneto (Data_Racc) ON DELETE CASCADE,
CONSTRAINT CE_Sede_Vinif_Mosto FOREIGN KEY (Via_Sv, Civico_Sv, CAP_Sv,
Citta_Sv, Comune_Sv, Nome_Sede)
REFERENCES Sede_Vinif (Via_Sv, Civico_Sv, CAP_Sv, Citta_Sv, Comune_Sv,
Nome_Sede) ON DELETE CASCADE,

```

```
CONSTRAINT CP_Mosto PRIMARY KEY (Num_Lotto_Mosto)
);
CREATE TABLE Carrello(
Ragione_Sociale varchar(255) NOT NULL,
Codice_Acquisto int NOT NULL,
Num_Conf_Vendute int NOT NULL,
Metodo_Pagamento varchar(255) NOT NULL,
Prezzo_Vendita float NOT NULL,
Data_Acquisto DATE NOT NULL,
CONSTRAINT CE_Cliente_Carrello FOREIGN KEY (Ragione_Sociale)
REFERENCES Cliente (Ragione_Sociale) ON DELETE CASCADE,
CONSTRAINT CP_Carrello PRIMARY KEY (Codice_Acquisto)
);
CREATE TABLE Lotto_Vino(
Num_Lotto int NOT NULL,
Nome_Vino varchar(255) NOT NULL,
Data_Lotto DATE NOT NULL,
Num_Lotto_Mosto int NOT NULL,
CONSTRAINT CE_Lotto_M FOREIGN KEY(Num_Lotto_Mosto)
REFERENCES Mosto(Num_Lotto_Mosto) ON DELETE CASCADE,
CONSTRAINT CE_Tipo_Vino_Controllo_Certificazione FOREIGN KEY (Nome_Vino)
REFERENCES Tipo_Vino (Nome_Vino) ON DELETE CASCADE,
CONSTRAINT CP_Lotto_Vino PRIMARY KEY (Num_Lotto)
```



```
);  
  
CREATE TABLE Botte(  
    Num_Botte int NOT NULL,  
    Num_Lotto int NOT NULL,  
    Tipo_Legno varchar(255) NOT NULL,  
    CONSTRAINT CE_Lotto_Vino_Botte FOREIGN KEY (Num_Lotto)  
    REFERENCES Lotto_Vino (Num_Lotto) ON DELETE CASCADE,  
    CONSTRAINT CP_Botte PRIMARY KEY (Num_Botte)  
);  
  
CREATE TABLE Imbottigliatore(  
    Nome_Imbott varchar(255) NOT NULL,  
    Num_Lotto int NOT NULL,  
    Via_Imbott varchar(255) NOT NULL,  
    Civico_Imbott int NOT NULL,  
    CAP_Imbott int NOT NULL,  
    Citta_Imbott varchar(255) NOT NULL,  
    Comune_Imbott varchar(255) NOT NULL,  
    Data_Imbott DATE NOT NULL,  
    Capacita_Bottiglia int NOT NULL,  
    Metodo_Imbott varchar(255) NOT NULL,  
    CONSTRAINT CP_Imbottigliatore PRIMARY KEY (Nome_Imbott, Via_Imbott,  
    Civico_Imbott, Citta_Imbott, CAP_Imbott, Comune_Imbott)  
);
```



```

CREATE TABLE Confezione(
Num_Conf int NOT NULL,
Codice_Acquisto int NULL,
Prezzo_Conf float NOT NULL,
CONSTRAINT CE_Carrello_Conf FOREIGN KEY (Codice_Acquisto)
REFERENCES Carrello(Codice_Acquisto) ON DELETE CASCADE,
CONSTRAINT CP_Confezione PRIMARY KEY (Num_Conf)
);

CREATE TABLE Controllo_Certificazione(
Nome_Vino varchar(255) NOT NULL,
Nome_Ente varchar(255) NOT NULL,
Num_Pratica int NOT NULL,
Data_Certif DATE NOT NULL,
Esito NUMBER(1) NOT NULL,
Certificato_Richiesto varchar(255) NOT NULL,
CONSTRAINT CE_Tipo_Vino_CC FOREIGN KEY (Nome_Vino)
REFERENCES Tipo_Vino (Nome_Vino) ON DELETE CASCADE,
CONSTRAINT CE_Ente_CC FOREIGN KEY (Nome_Ente)
REFERENCES Ente_Certif (Nome_Ente) ON DELETE CASCADE,
CONSTRAINT CP_CC PRIMARY KEY (Num_pratica)
);

CREATE TABLE Composizione_vino(Nome_vinovarchar(255)NOT NULL,Specievarchar(255)NOT NULL,
CONSTRAINT CE_Tipo_vino_CC FOREIGN KEY (Specie) REFERENCES Tipo_vino(Specie))

```

```

CREATE TABLE Vendemmia(
Specie varchar(255) NOT NULL,
Data_Racc DATE NOT NULL,
Quantita int NOT NULL,
CONSTRAINT CE_Tipo_Uva_Vendemmia FOREIGN KEY (Specie)
REFERENCES Tipo_Uva (Specie) ON DELETE CASCADE,
CONSTRAINT CE_Raccolto_Vigneto_Vendemmia FOREIGN KEY(Data_Racc)
REFERENCES Raccolto_Vigneto (Data_Racc) ON DELETE CASCADE
);
CREATE TABLE Interventi_Subiti(
NomeV varchar(255) NOT NULL,
ComuneV varchar(255) NOT NULL,
Tipo_Intervento varchar(255) NOT NULL,
Data_Intervento DATE NOT NULL,
Costo float NOT NULL,
CONSTRAINT CE_Vigneto_Interventi_Subiti FOREIGN KEY (NomeV, Co-
muneV)
REFERENCES Vigneto (NomeV, ComuneV) ON DELETE CASCADE,
CONSTRAINT CE_Intervento_Interventi_Subiti FOREIGN KEY (Tipo_Intervento,
Data_Intervento)
REFERENCES Intervento (Tipo_Intervento, Data_Intervento) ON DELETE
CASCADE
);

```

```

CREATE TABLE Coltivazione(
NomeV varchar(255) NOT NULL,
ComuneV varchar(255 ) NOT NULL,
Specie varchar(255) NOT NULL,
Comune_Provenienza varchar(255) NOT NULL,
CONSTRAINT CE_Vigneto_Coltivazione FOREIGN KEY (NomeV, ComuneV)
REFERENCES Vigneto (NomeV, ComuneV) ON DELETE CASCADE,
CONSTRAINT CE_Tipo_Uva_Coltivazione FOREIGN KEY (Specie)
REFERENCES Tipo_Uva (Specie) ON DELETE CASCADE
);
CREATE TABLE Composizione_Lotto(
Nome_Vino varchar(255) NOT NULL,
Num_Lotto int NOT NULL,
Azienda_Imbott varchar(255) NOT NULL,
Volume_Nominale int NOT NULL,
CONSTRAINT CE_Tipo_Vino_Composizione_Lotto FOREIGN KEY (Nome_Vino)
REFERENCES Tipo_Vino (Nome_Vino) ON DELETE CASCADE,
CONSTRAINT CE_Lotto_Vino_Composizione_Lotto FOREIGN KEY (Num_Lotto)
REFERENCES Lotto_Vino (Num_Lotto) ON DELETE CASCADE
);
CREATE TABLE Trattamento_Subito(
NomeV varchar(255) NOT NULL,
ComuneV varchar(255) NOT NULL,

```

```
Nome_Trattamento varchar(255) NOT NULL,  
Data_Trattamento DATE NOT NULL,  
Costo_Trattamento float NOT NULL,  
CONSTRAINT CE_Vigneto_Trattamento_Subito FOREIGN KEY (NomeV,  
ComuneV)  
REFERENCES Vigneto (NomeV, ComuneV) ON DELETE CASCADE,  
CONSTRAINT CE_Trattamento_Trattamento_Subito FOREIGN KEY (Nome_Trattamento,  
Data_Trattamento)  
REFERENCES Trattamento(Nome_Trattamento, Data_Trattamento) ON DELETE  
CASCADE  
);  
CREATE TABLE Cantina(  
Nome_Cantina varchar(255) NOT NULL,  
Via_Cantina varchar(255) NOT NULL,  
Civico_Cantina int NOT NULL,  
CAP_Cantina int NOT NULL,  
Citta_Cantina varchar(255) NOT NULL,  
Comune_Cantina varchar(255) NOT NULL,  
Nome_Sede varchar(255) NOT NULL,  
Num_Lotto int NOT NULL,  
Temperatura_Cantina int NOT NULL,  
Percentuale_Umidita int NOT NULL,  
Metratura_Cantina int NOT NULL,
```



```
CONSTRAINT CE_Lotto_Vino_Stoccaggio FOREIGN KEY (Num_Lotto)
REFERENCES Lotto_Vino (Num_Lotto) ON DELETE CASCADE,
CONSTRAINT CP_Cantina PRIMARY KEY (Nome_Cantina, Citta_Cantina,
Via_Cantina)
);
CREATE TABLE Confezionamento(
Num_Conf int NOT NULL,
Num_Lotto int NOT NULL,
Data_Confezionamento DATE NOT NULL,
Num_Bottiglie_Confezionate int NOT NULL,
Formato_Confezionatura varchar(255) NOT NULL,
CONSTRAINT CE_Lotto_Vino_Confezionamento FOREIGN KEY (Num_Lotto)
REFERENCES Lotto_Vino (Num_Lotto) ON DELETE CASCADE,
CONSTRAINT CE_Confezione_Confezionamento FOREIGN KEY (Num_Conf)
REFERENCES Confezione (Num_Conf) ON DELETE CASCADE
);
CREATE TABLE Vendita(
Num_Lotto int NOT NULL,
Ragione_Sociale varchar(255) NOT NULL,
Codice_Acquisto int NOT NULL,
Metodo_Pagamento varchar(255) NOT NULL,
Num_Confezioni_Vendute int NOT NULL,
Categoria_Vino_Venduta varchar(255) NOT NULL,
```



Prezzo\_Vendita int NOT NULL,  
CONSTRAINT CE\_Lotto\_Vino\_Vendita FOREIGN KEY (Num\_Lotto)  
REFERENCES Lotto\_Vino (Num\_Lotto) ON DELETE CASCADE,  
CONSTRAINT CE\_Cliente\_Vendita FOREIGN KEY (Ragione\_Sociale)  
REFERENCES Cliente (Ragione\_Sociale) ON DELETE CASCADE,  
CONSTRAINT CE\_Carrello\_Vendita FOREIGN KEY (Codice\_Acquisto)  
REFERENCES Carrello (Codice\_Acquisto) ON DELETE CASCADE  
);

## 8 Codice di Trigger e Procedure

### 8.1 Trigger

*Trigger che controlla che il numero delle bottiglie in confezione sia multiplo di 3*

---

```
CREATE OR REPLACE TRIGGER Numero_Bottiglie_Conf_Div_Tre
BEFORE INSERT on Confezionamento
FOR EACH ROW
DECLARE
CONTATORE NUMBER;
ERRORE EXCEPTION;
BEGIN
IF :New.Num.Bottiglie_Confezionate MOD 3 <> 0 THEN
RAISE ERRORE;
END IF;
EXCEPTION
WHEN ERRORE THEN
RAISE_APPLICATION_ERROR(-20000, 'Numero Bottiglie per confezione
dev'essere divisibile per 3!');
END;
```

---

*Trigger che controlla se un certificato richiesto era gia' presente*

```
CREATE OR REPLACE TRIGGER Certificato_presente
BEFORE INSERT on Controllo_Certificazione
FOR EACH ROW
DECLARE
CONTATORE NUMBER;
ERRORE EXCEPTION;
BEGIN
SELECT (COUNT(*)) INTO CONTATORE FROM Controllo_Certificazione
WHERE (Certificato_Richiesto = :NEW.Certificato_Richiesto AND Nome_Vino
= :NEW.Nome_Vino);
IF CONTATORE > 1 THEN
RAISE ERRORE;
END IF;
EXCEPTION
WHEN ERRORE THEN
RAISE_APPLICATION_ERROR(-20000, 'Gia Inserito');
END;
```

—  
*Trigger che controlla che il totale della percentuale delle uve inserite per la composizione di un vino non superi il 100%*  
—

```
CREATE OR REPLACE TRIGGER Supera_100_PC
before INSERT or update ON Composizione_Vino
```

```
FOR EACH ROW
DECLARE
pragma autonomous_transaction;
Somma_Percentuali float;
Out_Percentuale float;
BEGIN
SELECT(
SUM(Percentuale)
)
INTO (Somma_Percentuali) FROM (Composizione_Vino) where (Nome_Vino =
:NEW.Nome_Vino);
IF(Somma_Percentuali + :NEW.Percentuale > 100) THEN
DBMS_OUTPUT.PUT_LINE( 'Supera 100%, Correzione in corso');
Out_Percentuale := 100 - Somma_Percentuali;
IF Out_Percentuale > 0 THEN
:NEW.Percentuale := Out_Percentuale;
ELSE
RAISE_APPLICATION_ERROR( -2000, 'Percentuale gia piena');
END IF;
END IF;
COMMIT;
END;
```

–

*Trigger che controlla che un vino non possa richiedere piu' di 3 certificazioni*

---

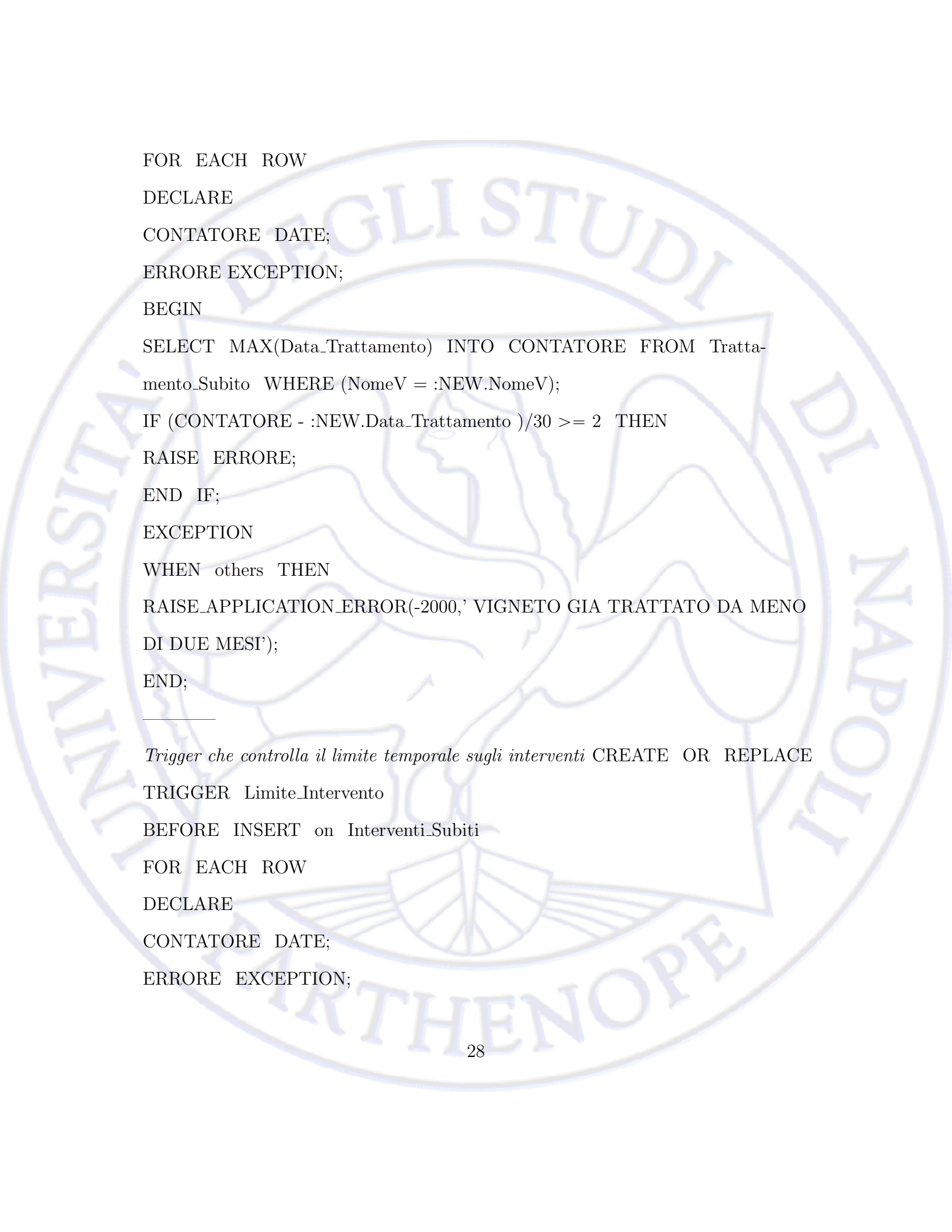
```
CREATE OR REPLACE TRIGGER Limite_Certificazioni
BEFORE INSERT on Controllo_Certificazione
FOR EACH ROW DECLARE
CONTATORE NUMBER;
ERRORE EXCEPTION;
BEGIN
SELECT (COUNT(*)) INTO CONTATORE FROM Controllo_Certificazione
WHERE (Nome_Vino = :NEW.Nome_Vino);
IF CONTATORE >= 3 THEN
RAISE ERRORE;
END IF;
EXCEPTION
WHEN others THEN
RAISE_APPLICATION_ERROR( -2000,' TROPPI CERTIFICATI');
END;
```

*Trigger che controlla che un trattamento venga svolto almeno dopo 2 mesi*

---

```
Trigger che controlla il limite temporale sui trattamenti CREATE OR REPLACE
TRIGGER Limite_Trattamento
BEFORE INSERT on Trattamento_Subito
```





```
FOR EACH ROW
DECLARE
CONTATORE DATE;
ERRORE EXCEPTION;
BEGIN
SELECT MAX(Data_Trattamento) INTO CONTATORE FROM Tratta-
mento_Subito WHERE (NomeV = :NEW.NomeV);
IF (CONTATORE - :NEW.Data_Trattamento )/30 >= 2 THEN
RAISE ERRORE;
END IF;
EXCEPTION
WHEN others THEN
RAISE_APPLICATION_ERROR(-2000,' VIGNETO GIA TRATTATO DA MENO
DI DUE MESI');
END;
```

---

*Trigger che controlla il limite temporale sugli interventi* CREATE OR REPLACE

```
TRIGGER Limite_Intervento
BEFORE INSERT on Interventi_Subiti
FOR EACH ROW
DECLARE
CONTATORE DATE;
ERRORE EXCEPTION;
```

```

BEGIN
SELECT  MAX(Data_Intervento) INTO  CONTATORE  FROM  Interventi_Subiti
WHERE  (NomeV = :NEW.NomeV);
IF  ((CONTATORE - :NEW.Data_Intervento))  <=  21  THEN
RAISE  ERRORE;
END  IF;
EXCEPTION
WHEN  others  THEN
RAISE_APPLICATION_ERROR(-2000,' Gia Fatto Intervento DA MENO DI Tre
Settimane');
END;

```

—————  
*Trigger che assegna Sconto/Sovrapprezzo ad uno specifico cliente* —————

```

create or replace trigger Sconto_Cliente
before insert on Cliente
for each row
declare
NomeCL  varchar(255);
Sovrapprezzo  int;
Costo int;
begin
select  Nome_Cliente into  NomeCL  from  Cliente  where  (:OLD.Ragione_Sociale
= :NEW.Ragione_Sociale);

```

```
select Prezzo_Vendita into Costo from Vendita;
Sovrapprezzo := (Costo * 40)/100;
if :new.Nome_Cliente = 'Camastra' then
Costo := Costo + Sovrapprezzo;
end if;
end;
```

---

## 8.2 Procedure

*Procedura che assegna automaticamente la certificazione "Vino da Tavola", una volta non superati 3 controlli* ———

```
CREATE OR REPLACE PROCEDURE Vino_Tavola (In_Nome_Vino var-
char2)
IS
CONTATORE NUMBER;
BEGIN
SELECT (COUNT(*)) INTO CONTATORE FROM Controllo_Certificazione
WHERE (Nome_Vino = In_Nome_Vino AND Esito = '0');
IF CONTATORE >= 3 THEN
dbms_output.put_line('Questo Vino è catalogato come vino da tavola');
ELSE
dbms_output.put_line('Questo Vino non è un vino da tavola');
END IF;
```

END;

---

CREATE OR REPLACE PROCEDURE Trova\_Sedi\_Imbott (In\_Lotto\_Vino  
int)

IS

Nome\_Imb varchar(255);

BEGIN

dbms\_output.put\_line('Questo lotto è stato imbottigliato presso:');

select (Nome\_Imbott) into Nome\_Imb from Imbottigliatore where(Num\_Lotto  
= In\_Lotto\_Vino);

dbms\_output.put\_line(Nome\_Imb);

END;